



UNIVERSIDAD DE GUAYAQUIL
Facultad de Ciencias Matemáticas y Físicas
Carrera de Ingeniería en Sistemas Computacionales

“Servidor SMS”

TESIS DE GRADO

Previo la Obtención del Título de:

**INGENIERO EN SISTEMAS
COMPUTACIONALES**

Autores:

GIOMAR PAOLA CHÓEZ CATUTO
JULIO CÉSAR CHÓEZ CATUTO
WASHINGTON SAÚL ZAMBRANO RIVAS

GUAYAQUIL – ECUADOR
Año: 2009

AGRADECIMIENTO

Agradezco a Dios por darme la oportunidad de cumplir uno de mis sueños, el graduarme como Ingeniera en Sistemas Computacionales, y por ser quién ha estado a mi lado en todo momento dándome las fuerzas necesarias para continuar luchando día tras día y seguir adelante.

A mis tutores Ing. Mario Robayo e Ing. Eduardo Alvarado por sus sugerencias, orientación y apoyo para la realización de este proyecto.

A mis maestros por compartir sus conocimientos que con mucho esmero han brindado en toda mi carrera y por incentivarnos a seguir adelante.

Y a todas aquellas personas que de una u otra forma colaboraron en la realización de este proyecto, hago extenso mi más sincero agradecimiento.

Giomar Chóez C.

AGRADECIMIENTO

A la carrera por sus avances tecnológicos y por su prestigio, formando profesionales capacitados para un medio muy competitivo como es el área de sistemas.

A los tutores y profesores del seminario, el Ing. Mario Robayo y el Ing. Eduardo Alvarado que con sus experiencias y conocimientos hicieron posible dar paso a la terminación del proyecto de grado.

Al coordinador del paralelo el Ing. Abel Alarcón que por sus reuniones indicaba pautas del seminario y siempre presto al dialogo y colaboración.

A mis compañeros y amigos, que por medio de las discusiones y preguntas, me hacen crecer en conocimiento.

Julio Chóez C.

AGRADECIMIENTO

A Dios ante todo, a mi Madre que siempre está aquí a mi lado, mis hermanos.

A la institución educativa, Carrera de Ingeniería de Sistemas Computaciones, de la universidad de Guayaquil que me brindo la oportunidad para desarrollarme como un profesional y como una persona de bien.

A todos mis tíos que me dieron la oportunidad de estar aquí en esta bella ciudad, que no lo voy a defraudar.

A los catedráticos de la carrera que siempre están ahí enseñando y dando a conocer sus experiencias en el campo laboral.

A mi Tío Político que él no vive en nuestro país, que siempre estuvo dando su apoyo, pendiente de los pasos que estaba dando en los estudios.

Saúl Zambrano R.

DEDICATORIA

El esfuerzo y dedicación que he puesto en este proyecto va con mucho cariño a mis padres Sr. Pablo Chóez y Sra. Vicenta Catuto por todo lo que me han dado en esta vida, por todo lo que soy como persona gracias a sus nobles consejos y por estar a mi lado en los momentos difíciles, apoyándome en todas las decisiones que he tomado en la vida.

A mi hermano Julio César quien me ha acompañado durante todo este trayecto con paciencia y optimismo.

A mi hija Milena Paola por ser la fuente de mi inspiración para superarme cada día más y así poder luchar para que la vida nos depare un futuro mejor.

Giomar Chóez C.

DEDICATORIA

No hay palabras para describir mi profundo agradecimiento a mis Padres, por todos sus consejos y ayuda comprendiendo que nada se consigue de la noche a la mañana, que la perseverancia y el esfuerzo son el camino para lograr objetivos.

A mi hermana quien es parte del grupo de seminario y también como compañera de la universidad con su ayuda y apoyo.

A mi sobrina que es parte de mi vida y una de las razones de mi inspiración y deseos de superación.

Este seminario es parte de un nuevo comienzo por ello la dedico a Dios por darme la oportunidad de llegar a estas instancias con base de esfuerzo y sacrificio.

Julio Chóez C.

DEDICATORIA

Le dedico este trabajo especial de grado, ante todo, a Dios, el cual me han acompañado a lo largo de toda mi vida, dándome salud y fortaleza.

A mis padres, especialmente a mi Madre que siempre ha estado aquí a mi lado en todo momento, a mis Abuelitas que ahora no están a mi lado dedico este trabajo. A mi hermano y mi hermana porque siempre vieron un gran potencial en mí y siempre me lo hicieron saber.

A las personas que siempre han confiado en mí, a los verdaderos amigos que siempre te dan la mano. Y por último, a la persona que siempre esta ayudándome moralmente, que ahora forma parte de esta lucha, para ti con mucho amor
Ingrid

Saúl Zambrano R.

TRIBUNAL DE GRADUACIÓN

PRESIDENTE DEL TRIBUNAL

PRIMER VOCAL

SEGUNDO VOCAL

SECRETARIO

DECLARACIÓN EXPRESA

“La autoría de la tesis de grado corresponde exclusivamente a los suscritos,
Perteneciendo a la Universidad de Guayaquil los derechos que generen la
Aplicación de la misma”

Según el Reglamento de Graduación de la Carrera de Ingeniería en Sistemas
Computacionales, Art. 26.

Giomar Paola Chóez Catuto
C.I. # 091959834-2

Julio César Chóez Catuto
C.I. # 091959718-7

Washington Saúl Zambrano Rivas
C.I. # 131000275-1

RESUMEN

El mundo va cambiando cada día más y por ende las necesidades de las personas, de acuerdo a las nuevas tecnologías que cada vez van avanzando con el pasar del tiempo, el **BLUETOOTH** no solo se adapta a los cambios sino que surge de las necesidades propias del ser humano.

Servidor Bluetooth ofrecerá a los estudiantes de la Carrera de Ingeniería de Sistemas Computacionales acceder a información personal, recibir avisos de manera eficaz, además de recibir publicidad de manera rápida con tan solo tener un teléfono celular con bluetooth.

Servidor Bluetooth se convierte en una herramienta útil en cualquier tipo de mercado, debido a que la mayoría de las personas poseen un teléfono celular con bluetooth que ha dejado de ser un lujo para ser una herramienta necesaria de comunicación, este mercado potencial hará que el Servidor Bluetooth se convierta en el boom que hará que las instituciones que lo

adquieran aumenten sus ingresos, adquieran más estudiantes y proporcionen confianza, seguridad y veracidad en tiempo real.

Nuestra compañía contará con:

- Líder de proyecto.
- Soporte usuario.
- Programador conocimientos base datos

Para realizar este gran proyecto se requiere una inversión inicial de \$770 dólares, los cuales serán aportados por los integrantes del proyecto.

Este proyecto no tendrá ningún costo, será un beneficio tanto para los estudiantes como para la carrera, puesto que va a satisfacer las necesidades del estudiante de conocer sus notas y la Carrera de Ingeniería de Sistemas innovará un producto utilizando una de las tecnologías que existen en la actualidad tratando de esta manera mejorar sus servicios a nivel académico dando una nueva imagen y ganando un prestigio cada vez más notorio a nivel de universidades.

ÍNDICE GENERAL

ÍNDICE	PÁGINAS
AGRADECIMIENTOS.....	II – IV
DEDICATORIAS	V – VII
TRIBUNAL DE GRADUACIÓN	VIII
DECLARACIÓN EXPRESA	IX
RESUMEN	X
INTRODUCCIÓN.....	XI

CAPITULO 1

1. PRELIMINARES

1.1. Antecedentes	1
1.2. Ámbito y problemática	5
1.3. Solución a la problemática	6
1.4. Visión y Misión del proyecto	7
1.5. Estudio de factibilidad	8
1.5.1.¿Qué es y en qué consiste el negocio?	8
1.5.2.¿Quién dirigirá el negocio?	9
1.5.3.¿Cuáles son las causas y razones del éxito?	9
1.5.4.¿Cuáles son los mecanismos y las estrategias que se van a utilizar para lograr las metas previstas?	10
1.6. Definición del proyecto	11
1.7. Gráfica del proyecto	11
1.8. Objetivos del proyecto	12
1.8.1.Objetivos generales del proyecto	12
1.8.2.Objetivos específicos del proyecto	13

1.9.	Alcance del proyecto	14
1.10.	Arquitectura del proyecto	16
1.11.	Metodología del proyecto	18
1.12.	Recursos necesarios para el proyecto	21
1.13.	Análisis del mercado	24
1.14.	Análisis técnico	25
1.15.	Análisis administrativo	25
1.15.1.	Roles de cada uno de los cargos	26
1.16.	Matriz de riesgos	31
1.17.	Matriz FODA	31
1.18.	Cronograma de actividades del proyecto	33

CAPITULO 2

2. ANÁLISIS

2.1.	Levantamiento de la información.....	34
2.2.	Análisis de requerimientos	35
2.3.	Análisis de riesgos	36
2.4.	Diagrama Entidad – Relación	38
2.4.1.	Diagrama DER	39
2.5.	Diagrama de flujo de procesos	40
2.5.1.	Flujo de procesos	40
2.5.2.	Procesos de portal Web	40
2.5.2.1.	Proceso de ingreso de usuarios	40
2.5.2.2.	Proceso de respuesta del servidor ante una petición	42
2.5.2.3.	Proceso de envío de mensajes	43
2.5.2.4.	Proceso de envío de publicidad	44
2.5.2.5.	Proceso de reportes	45
2.5.2.6.	Proceso de ingreso de usuarios externos	46

2.6.	Diagrama de casos de uso	47
2.6.1.	Primer nivel	47
2.6.2.	Segundo nivel	48
2.6.3.	Tercer nivel	49
2.6.4.	Descripción de casos de uso primer nivel	49
2.7.	Diagrama de contexto de arquitectura y funcionalidad...	52

CAPITULO 3

3. DISEÑO

3.1.	Diseño de interfaz	53
3.2.	Pagina Login	55
3.3.	Pagina principal	55
3.4.	Página ingreso de usuarios del sistema	56
3.5.	Página consulta de usuarios del sistema	57
3.6.	Página ingreso de usuarios externos del sistema	57
3.7.	Página consulta de usuarios externos del sistema ...	58
3.8.	Página ingreso de privilegios	58
3.9.	Página consulta de privilegios	59
3.10.	Página envío de mensajes	59
3.11.	Página consulta de envío de publicidad	60
3.12.	Página reporte dispositivos detectados	60
3.13.	Página reporte buzón de entrada	61
3.14.	Página reporte buzón de salida	61
3.15.	Página reporte buzón no recibidos	62
3.16.	Página reporte buzón de borradores	62
3.17.	Página reporte contador	63
3.18.	Página auditoria	64
3.19.	Página más usados.....	64

3.20.	Página de ingreso de peticiones	65
3.21.	Página de consulta de peticiones	65
3.22.	Diagrama de flujo de ventanas	66

CAPITULO 4

4. DESARROLLO DEL SOFTWARE

4.1.	Antecedentes	67
4.2.	Procesos principales.....	68
4.2.1.	Clase Aplicación Bluetooth.....	69
4.2.1.1.	Clase Conexión.java	69
4.2.1.2.	Clase Dispositivo.java.....	69
4.2.2.	Aplicación Bluetooth	70
4.2.2.1.	Buscar archivo.java	70
4.2.2.2.	Buscar servicios.java	70
4.2.2.3.	Detectar dispositivos.java	70
4.2.2.4.	Envío mensajes.java	71
4.2.2.5.	Envía publicidad.java	71
4.2.2.6.	Main.java	72
4.2.2.7.	Remote device discovery.java	72
4.2.2.8.	Remote device discovery1.java	72
4.2.3.	Clases Web Bluetooth.....	73
4.2.3.1.	Borradores.java	73
4.2.3.2.	Conexión.java	73
4.2.3.3.	Dispositivos.java	73
4.2.3.4.	Externos.java	74
4.2.3.5.	Fechas.java	74
4.2.3.6.	Funciones.java	74
4.2.3.7.	Mensajes.java	74

4.2.3.8.	Permisos.java	75
4.2.3.9.	Peticiones.java	75
4.2.3.10.	Publicidad.java	75
4.2.3.11.	Usuarios.java	75
4.2.4.	Clases Dispositivos	76
4.2.4.1.	Busca servicio.java	76
4.2.4.2.	Busca servicios.java	76
4.2.4.3.	Buscar servicios.java	76
4.2.4.4.	Enviar mensajes.java	77
4.2.4.5.	Remote device discovery.java	77
4.2.5.	Paquetes com.metaparadigm.jsonrpc	77
4.2.6.	Paquetes org.jfree.chart.demo.servlet	78
4.2.7.	Paquetes org.json	78
4.2.8.	Funciones y procedimientos en la base de datos ..	78
4.2.8.1.	Función funciones	78
4.2.8.2.	Procedimiento consultas	78
4.2.8.3.	Procedimiento contador mensajes	79
4.2.8.4.	Procedimiento mantenimiento borradores ...	79
4.2.8.5.	Procedimiento mantenimiento dispositivos ..	79
4.2.8.6.	Procedimiento mantenimiento externos	80
4.2.8.7.	Procedimiento mantenimiento permisos	80
4.2.8.8.	Procedimiento mantenimiento peticiones	80
4.2.8.9.	Procedimiento mantenimiento publicidad	81
4.2.8.10.	Procedimiento mantenimiento usuarios	81
4.2.8.11.	Procedimiento vistas	81

CAPITULO 5

5. PRUEBAS

5.1.	Método de prueba del software.....	82
5.2.	Técnicas en aplicación de pruebas de software.....	84
5.2.1.	Técnica de caja negra	84
5.2.1.1.	Partición equivalente.....	86
5.2.1.2.	Análisis de valores límite.....	87
5.2.2.	Técnica de caja blanca.....	87
5.3.	Estrategias de pruebas	88
5.4.	Pruebas realizadas al sistema	90
5.4.1.	Prueba de ingreso al sistema	90
5.4.2.	Prueba de envío de mensaje de texto a los dispositivos	91
5.4.2.1.	Pruebas del sistema	91
5.4.2.2.	Pruebas en los dispositivos	94
5.4.3.	Ingreso de usuarios del sistema	96

CAPITULO 6

6. CONCLUSIONES Y RECOMENDACIONES

6.1.	Conclusiones	101
6.2.	Recomendaciones.....	103
6.3.	Bibliografía	105

TABLAS

ÍNDICE	PÁGINAS
Tabla 1.1 Recurso Humano.....	22
Tabla 1.2 Matriz de Riesgo.....	32
Tabla 1.3 Fortalezas del proyecto.....	32
Tabla 1.4 Oportunidades del proyecto.....	33
Tabla 1.5 Debilidades del proyecto	33
Tabla 1.6 Amenazas del proyecto	33
Tabla 1.7 Cronograma de actividades	34
Tabla 2.1 Administración de usuarios	49
Tabla 2.2 Administración de privilegios	50
Tabla 2.3 Administración de envíos	50
Tabla 2.4 Visualización de reportes	51
Tabla 2.5 Configuración de peticiones	51

FIGURAS

ÍNDICE	PÁGINAS
Fig. 1.1 Bluetooth	2
Fig. 1.2 Ejemplo de un servidor bluetooth.....	3
Fig. 1.3 Gráfica del proyecto	11
Fig. 1.4 Arquitectura de N-CAPAS	17
Fig. 1.5 Dispositivo Bluetooth	23
Fig. 1.6 Administración del personal	26
Fig. 2.1 Diagrama Entidad – Relación	39
Fig. 2.2 Diagrama de ingreso de usuarios.....	41
Fig. 2.3 Diagrama de respuesta de servidor	42
Fig. 2.4 Diagrama de envío de mensajes	43
Fig. 2.5 Diagrama de envío de publicidad	44
Fig. 2.6 Diagrama de reportes	45
Fig. 2.7 Diagrama de ingreso de usuarios externos	46
Fig. 2.8 Diagrama de primer nivel	47
Fig. 2.9 Diagrama de segundo nivel	48
Fig. 2.10 Diagrama de tercer nivel	49
Fig. 2.11 Diagrama de funcionamiento del sistema	52
Fig. 3.1 Página de ingreso al sistema	55
Fig. 3.2 Página principal	56
Fig. 3.3 Ingreso de usuarios del sistema	56
Fig. 3.4 Consulta de usuarios del sistema	57
Fig. 3.5 Ingreso de usuarios externos	57
Fig. 3.6 Consulta de usuarios externos	58
Fig. 3.7 Ingreso de privilegios	58
Fig. 3.8 Consulta de privilegios	59

Fig. 3.9 Envío de mensajes	59
Fig. 3.10 Envío de publicidad	60
Fig. 3.11 Dispositivos detectados	60
Fig. 3.12 Reporte de buzón de entrada	61
Fig. 3.13 Reporte de buzón de salida	61
Fig. 3.14 Reporte de buzón no recibidos	62
Fig. 3.15 Reporte de borradores	63
Fig. 3.16 Reporte de contador de mensajes	63
Fig. 3.17 Auditoría	64
Fig. 3.18 Más usados.....	64
Fig. 3.19 Ingreso de peticiones	65
Fig. 3.20 Consulta de peticiones	65
Fig. 3.21 Diagrama de flujo de ventanas	66
Fig. 5.1 Identificación del usuario	90
Fig. 5.2 Tabla de usuarios del sistema	90
Fig. 5.3 Error de usuario.....	91
Fig. 5.4 Error al no escribir un mensaje	91
Fig. 5.5 Selección de la lista de borradores	92
Fig. 5.6 Dispositivos detectados	92
Fig. 5.7 Dispositivos detectados por el sistema	93
Fig. 5.8 Mensaje de espera	93
Fig. 5.9 Información en la consola	94
Fig. 5.10 Consulta de dispositivos	95
Fig. 5.11 Recepción de mensajes en los dispositivos	95
Fig. 5.12 Visualización del mensaje en los dispositivos	96
Fig. 5.13 Ingreso erróneo en un campo de sólo letras	96
Fig. 5.14 Ingreso erróneo en un campo de sólo números	97
Fig. 5.15 Validación de cédula del usuario	97
Fig. 5.16 Consulta del número de cédula	98

Fig. 5.17 Validación del correo electrónico	98
Fig. 5.18 Claves no coinciden	99
Fig. 5.19 Confirmación para almacenar los datos	99
Fig. 5.20 Ingreso del nuevo usuario	100
Fig. 5.21 Consulta de usuarios del sistema	100

INTRODUCCIÓN

En los últimos tiempos se ha podido presenciar el auge de las comunicaciones móviles. Las personas se encuentran en constante movimiento y gran parte de ellos poseen un teléfono celular. El mensaje por bluetooth es gratis para la persona que lo recibe y con alta capacidad de impacto y retención de los mensajes.

CAPITULO 1

PRELIMINARES

1.1. Antecedente

Desde la llegada de los celulares apareció la herramienta de comunicación más avanzada en la actualidad de la telefonía móvil. Los SMS estaban diseñados para que se comunicasen entre ellos pero con el tiempo ha surgido que se envíen mensajes a otros usuarios cercanos que se encuentren dentro del radio de acción que tengan su terminal con bluetooth activados.

Bluetooth es una especificación industrial para Redes Inalámbricas de Área Personal (WPANs) que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia segura y globalmente libre.



Figura 1.1 Bluetooth

De hecho, los móviles habituales no tienen función de enviar mensajes vía Bluetooth, esta funcionalidad se suple habitualmente enviando el mensaje como si fuera una tarjeta de contacto. Es por esto que el “Servidor Bluetooth” pretende dar un pequeño giro ofreciendo un software disponible para que el usuario pueda enviar mensajes masivos e imágenes de publicidad, cuya funcionabilidad será realmente útil.

El proceso consiste, el Servidor Bluetooth comprende una base de datos de almacenamiento de los campos relevantes de los mensajes, así como de los datos de los usuarios externos, como por ejemplo son los datos de identificación de los usuarios, número de teléfono, dispositivo asociado, etc.

Comprende además una interfaz que permite la comunicación del Servidor Bluetooth con los teléfonos celulares con bluetooth a través de la herramienta JAVA, además dicha interfaz permite realizar una serie de funciones como por ejemplo de consulta, almacenamiento, envío de mensajes, envío de publicidad de una manera rápida y eficaz.

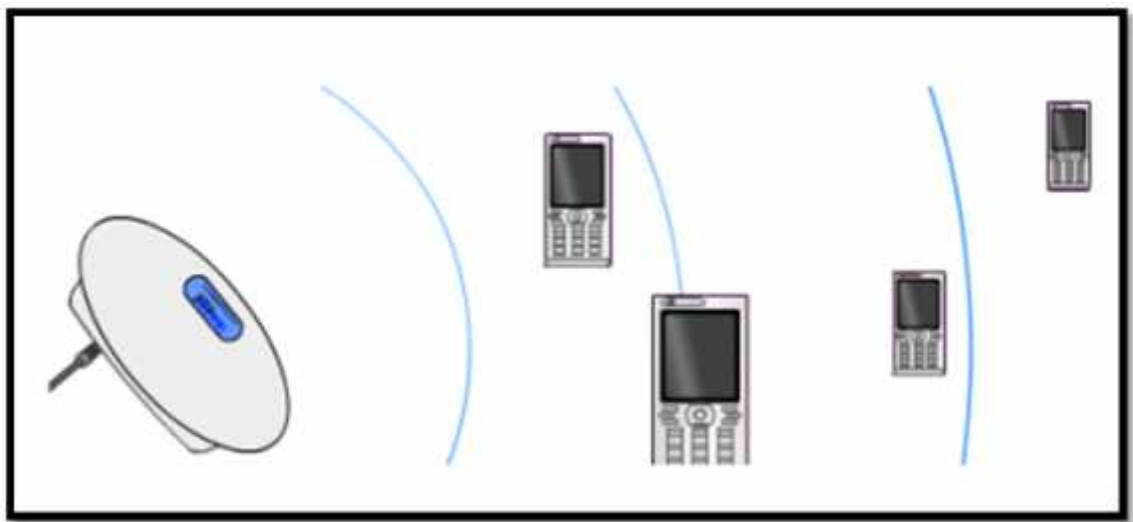


Figura 1.2 Ejemplo de un servidor bluetooth

Servidor Bluetooth será una aplicación destinada a la Universidad de Guayaquil, Facultad de Ciencias Matemáticas y Físicas Carrera de Ingeniería de Sistemas Computacionales. En la cual se enfocara en las notas de los semestres de la carrera y además en el envío de publicidad.

El estudiante registrado plenamente por la carrera podrá realizar consultas de notas como las materias que este cursando en ese momento ya sea que el estudiante este en un solo nivel de la carrera, como también en algunos paralelos como cursos de diferentes niveles.

Para ingresar a la aplicación se necesita que el estudiante ingrese el número de cédula más los parámetros para su acceso, enviando un mensaje vía bluetooth con la respectiva consulta desde un celular al Servidor Bluetooth y se hará la consulta respectiva de las notas. Los estudiantes una vez validado el ingreso con un parámetro podrán realizar consultas de sus notas tanto del primero o segundo parcial, como también el de recuperación.

La aplicación a su vez podrá enviar mensajes a los estudiantes y profesores, luego de haber escrito el texto, se activa el bluetooth y envía el mensaje a todos los dispositivos dotados de bluetooth que se encuentran a su alcance. Por ejemplo, comentarios, anuncios, mensajes publicitarios como de la carrera indicando desde una suspensión de clases como la de indicar un evento en particular. Así como también permitirá el envío de imágenes publicitarias, sin costo alguno en la transmisión y sin importar la compañía celular a la que pertenezca.

Con el fin de que el estudiante aproveche esta aplicación y con un celular que hoy en día es el primer medio de comunicación como herramienta para obtener la información.

1.2. Ámbito y problemática

El Servidor Bluetooth es una tecnología basada en la recepción-emisión de mensajes cortos por medio del bluetooth de un celular.

La propuesta de realizar este proyecto es para tener un contacto más directo con nuestros clientes o usuarios, en la actualidad las personas siempre quieren estar al tanto y bien informados, tener la información veraz y rápida al instante de todo lo que sucede dentro de una actividad, sea comercial, de servicio, inclusive hasta para un control bi-direccional de parte y parte, (cliente-proveedor).

Es muy útil para nuestra sociedad, prestaremos un servicio de punta, está dirigido para la unidad educativa (Universidad de Guayaquil), contará con todos los servicios adquiridos para que los miembros que estén involucrados dentro de la unidad tengan una fácil comunicación, y a su vez, los alumnos y profesores estén al tanto de notas, promedios, convocatorias, actividades.

También nos permite en el enfoque dar a conocer muchas cosas que se puede realizar por medio de este servicio, oferta de trabajo, ventas de cualquier artículo basado en la carrera, promocionar algún producto en particular.

La facilidad de este medio, es porqué no utiliza las redes de los proveedores de telefonía móvil, puede enviar una cantidad ilimitada de mensajes a teléfonos celulares de cualquier empresa proveedora y siempre sin costo alguno.

1.3. Solución a la Problemática

Nuestra propuesta es ofrecer un servicio de mensajería que facilite al usuario obtener información requerida de una manera rápida y eficaz a través de su teléfono celular con bluetooth, así como también de permitir a la empresa poder enviar mensajes informativos, avisos etc. e imágenes de publicidad a grupos de personas.

1.4. Visión y Misión del Proyecto

Visión: Reconocemos que en una empresa los clientes son cruciales para éxito y por ello nos esforzamos para proporcionarles el nivel óptimo de servicio y dar el soporte con el mayor número de beneficios posible que ofrecen soluciones completas a los clientes.

Servidor Bluetooth proporcionará el servicio de entrega de múltiples mensajes de textos e imágenes integrado a sus aplicaciones, es una evolución de la solución básica de mensajes que las operadoras del servicio ofrecen en la actualidad, aportando mayor valor a la gestión de sus negocios.

Misión: Conectar gente y organizaciones por completo mediante un paquete de soluciones que ofrece este servicio en la red desarrollando y ofreciendo la tecnología a bajo costo.

Especializada en soluciones empresariales y el ámbito educativo, tendrá capacidad para dar asistencia a clientes en todo el Ecuador.

Este producto tiene acogida en otros países que está siendo explotada en sus recursos y actualmente en el país se están desarrollando empresas dedicadas a este tipo de negocios.

Bancos, universidades, buscan tener informado al cliente de una manera ágil, también el cliente pueda acceder a una información oportuna en el menor tiempo.

1.5. Estudio de Factibilidad

1.5.1. ¿Qué es y en qué consiste el negocio?

Es un servidor de transmisión de mensajes donde puede enviar mensajes de texto e imágenes, y resolver una consulta, llamado Servidor Bluetooth.

El negocio consiste en prestar un servicio de comunicación eficaz y rápida por medio de la tecnología bluetooth del celular.

El proceso consiste en una aplicación Web donde tendrá acceso las personas indicadas para enviar los respectivos mensajes a sus clientes, estos mensajes son transmitidos a través del dispositivo bluetooth del servidor hacia los diferentes dispositivos que se encuentren a su alrededor, a su vez, el usuario (estudiante) puede hacer una petición por medios de consultas que se predestinara por las peticiones realizada por el servicio adquirido, y se realiza de la misma manera de una forma inversa al proceso ya mencionado.

1.5.2. ¿Quién dirigirá el negocio?

El Administrador del servidor tendrá la responsabilidad de dirigir el negocio de una manera eficaz y operativa el mayor tiempo posible que dure el contrato del servicio. El administrador debe estar al tanto de lo que suceda y tener la capacidad de resolver los problemas que algún día de presente.

En este caso, Los Administradores, serán los Estudiantes que están realizando este proyecto.

1.5.3. ¿Cuáles son las causas y razones de éxito?

Reducción de los procesos y el tiempo de respuesta con que se obtiene la información es corta ganando así la satisfacción del estudiante.

El éxito del Servidor Bluetooth es la facilidad de manejo de envío y recepción de mensajes mediante el dispositivo bluetooth del servidor hacia un celular con esta tecnología sin ningún costo para el estudiante como para el cliente en este caso la Universidad de Guayaquil.

1.5.4. ¿Cuáles son los mecanismo y las estrategias que se van a utilizar para lograr las metas previstas?

Mecanismo

- Implementar el servicio en nuestra unidad educativa.
- Dar cobertura a todos los estudiantes que estén ingresados en una base de datos.
- Que sea fácil de utilizar, sencillo y veraz los requerimientos o peticiones que el usuario necesite.

Estrategias

- Envío de publicidad, compra y venta
- Envío de anuncios de ofertas de trabajo
- Envío de avisos de seminarios o cursos
- Envío de imágenes publicitarias.
- Acceso ilimitado para realizar consultas

1.6. Definición del proyecto

El Proyecto Servidor Bluetooth permite enviar mensajes masivos y responder las solicitudes de los usuarios a través de un celular con bluetooth, desde la aplicación web. Permitiendo administrar la información de los usuarios y ofreciendo reportes sobre el uso de la aplicación.

1.7. Gráfica del Proyecto

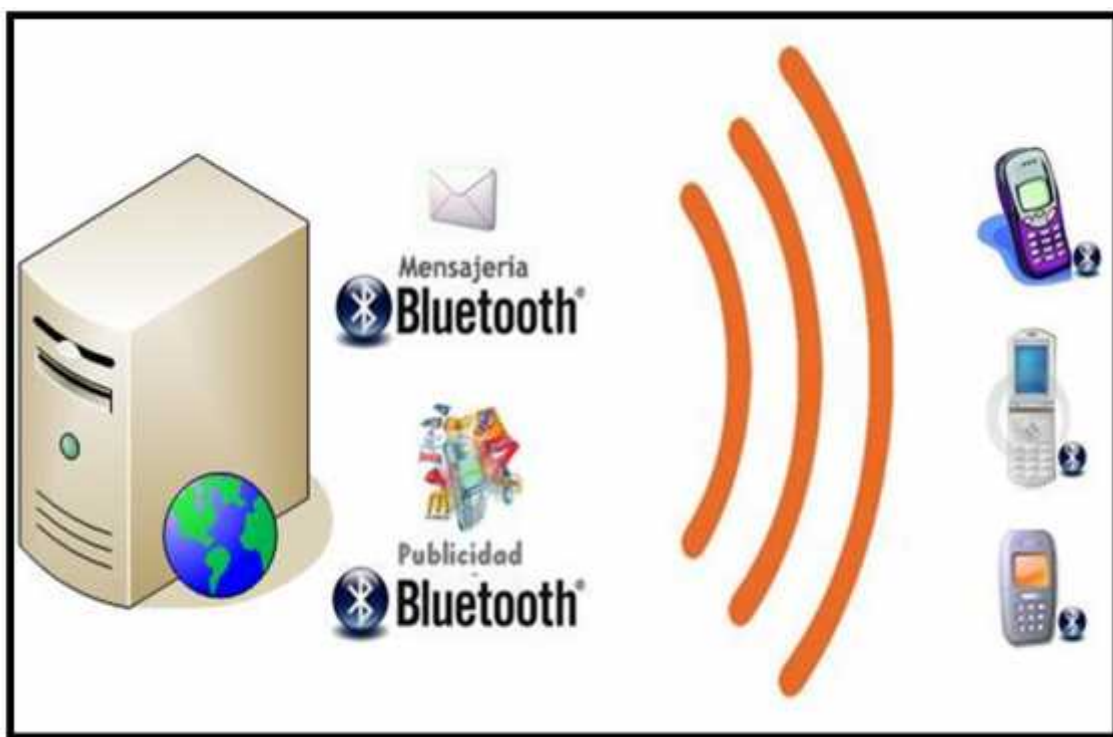


Figura 1.3 Gráfica del proyecto

1.8. Objetivos del Proyecto

1.8.1. Objetivos Generales del Proyecto

En la actualidad se ha hecho necesario buscar algo novedoso con lo cual las empresas estén comunicadas directamente con sus clientes, razón por la cual se busca interactuar directamente la relación Empresa-cliente ya no dependiendo de que el cliente acceda a Internet en un computador, ni que el cliente posea un dispositivo móvil con capacidad de navegar en Internet. Ahora con esta herramienta y con información del cliente pondremos a consideración el envío de mensajes desde un computador a un móvil y viceversa.

El servidor Bluetooth, permite a la Empresa obtener un servicio para que esté en contacto con sus clientes y futuros clientes de manera rápida y directa a través de un dispositivo móvil con bluetooth, logrando así la interacción del usuario con los procesos ligados de un Servidor Bluetooth.

1.8.2. Objetivos Específicos del Proyecto

- Desarrollar una Aplicación en Java que permita manejar información por medio del puerto de Comunicaciones.
- Desarrollar una Aplicación en Java que permita el envío y recepción de mensajes de textos.
- Desarrollar una Aplicación en Java que permita el envío de imágenes publicitarias.
- Integración de Java y una base de datos My-SQL.
- Optimizar tiempo en las consultas de los usuarios por medio del celular.
- Va a estar dirigido a una unidad educativa (Carrera Ingeniería de Sistemas Computacionales) para coordinar a los profesores y a los alumnos, enviándoles calificaciones y cualquier otro aviso.
- Envío y recepción de información de una manera clara, transparente y veraz.
- Control de envío y recepción de mensajes de texto.
- Como canal de comunicación para hacer llegar publicidad en cualquier sector de actividad.

1.9. Alcance del Proyecto

El proyecto contiene los elementos técnicos requeridos para la implantación de los servicios de envío de mensajes de texto e imágenes de publicidad, en modalidad unidireccional y bidireccional de información. El desarrollo e implantación de los servicios de mensajería, permitirá al cliente recibir información de interés y desde el equipo móvil el cliente (estudiante) podrá enviar una petición.

Es por esto que el esquema básico que va a contener nuestra aplicación Web es el siguiente:

- Comunicación bluetooth; Envíos masivos de mensajes de textos e imágenes a personas que se encuentran dentro del radio de acción del dispositivo Bluetooth. El sistema de envíos deberá estar instalado en los servidores de la empresa así como también el dispositivo bluetooth adquirido para el funcionamiento del sistema.
- Avisos programados bajo demanda; Permite enviar mensajes a varias personas a la vez y puede recibir peticiones de usuarios en este caso la de los estudiantes.

- Servicio de forma Bidireccional; Permite al usuario poder comunicarse con el sistema y cubrir un requerimiento de solicitud y confirmación de los servicios disponibles vía bluetooth.
- Servicio de acceso cliente – servidor bluetooth; El usuario puede ingresar con un único código que puede ser un código asignado o su número de cedula desde cualquier teléfono móvil con tecnología bluetooth y el servidor envía información al usuario al equipo celular inscrito en la base de datos.
- Aplicación Web; Se desarrollará una aplicación Web donde se tendrá:
 - Registro y consulta de usuarios del sistema y externos
 - Envío y recepción de mensajes a un determinado usuario o a un grupo.
 - Administración del contenido de mensajes de texto.
 - Administración de nuevas recepciones de mensajes.
 - Programación de envío de mensajes
 - Envío de publicidad.
 - Administración del contenido de las imágenes de publicidad.
 - Programación de envío de imágenes.
 - Programación de privilegios de usuario.

- Registro de usuarios del sistema y externos (estudiantes)
- Administración de usuarios.
- TOP TEN de los dispositivos conectados a nuestro sistema.
- Reporte de mensajes enviados.
- Reporte de mensajes recibidos.
- Reporte de mensajes que no fueron recibidos por el usuario.
- Reporte para la auditoría del sistema.
- Reportes diarios, mensuales, quincenales o de acuerdo al rango que el usuario seleccione de todas las consultas o envíos.
- Reportes estadísticos. Contador de mensajes, enviados, recibidos y no enviados.
- Impresión de todos los reportes.
- Exportación de todos los reportes a Excel o PDF.

1.10. Arquitectura del Proyecto

Para nuestro proyecto utilizaremos la implementación N-CAPAS.

La programación por capas es un estilo de programación en el que el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño; un ejemplo básico de esto consiste en separar la capa de datos de la capa de presentación al usuario.



Figura 1.4 Arquitectura de N-CAPAS

La ventaja principal de este estilo es que el desarrollo se puede llevar a cabo en varios niveles y, en caso de que sobrevenga algún cambio, sólo se ataca al nivel requerido sin tener que revisar entre código mezclado.

Un buen ejemplo de este método de programación sería el modelo de interconexión de sistemas abiertos. Además, permite distribuir el trabajo de creación de una aplicación por niveles; de este modo, cada grupo de trabajo está totalmente abstraído del resto de niveles, de forma que basta con conocer la API que existe entre niveles.

1.11. Metodología del Proyecto

La metodología que usaremos en nuestro proyecto es el Lenguaje de Modelamiento Unificado (UML – Unified Modeling Language) un lenguaje gráfico que permitirá visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software.

El Lenguaje Unificado de Modelado es un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial que los diagramas y símbolos significan.

UML se puede usar para modelar distintos tipos de sistemas de software, ofrece nueve diagramas en los cuales modelar sistemas, para nuestro análisis usaremos:

- **Diagramas de Casos de Uso para modelar los procesos.**

Un diagrama Uso-Caso describe lo que hace un sistema desde el punto de vista de un observador externo, debido a esto, un diagrama de este tipo generalmente es de los más sencillos de interpretar en UML, ya que su razón de ser se concentra en un Que hace el sistema, a diferencia de otros diagramas UML que intentan dar respuesta a un Como logra su comportamiento el sistema.

Un Uso-Caso está muy relacionado con lo que pudiera ser considerado un escenario en el sistema, esto es, lo que ocurre cuando alguien interactúa con el sistema: "Acude un mesero a colocar la orden, la orden es tomada por el cocinero, y posteriormente se abona a la cuenta del cliente el cargo".

Un Uso-Caso es empleado con más frecuencia en alguna de las siguientes etapas:

Determinación de Requerimientos: Por lo general nuevos requerimientos de sistema generan nuevos usos-casos, conforme es analizado y diseñado el sistema.

Comunicación con el Cliente: Debido a la sencillez de este tipo de diagramas, son fáciles de emplear para comunicarse con el cliente final del proyecto.

Generación de pruebas de Sistemas: A través de los diagramas uso-caso se pueden generar una serie de pruebas de sistema.

- **Diagramas de Secuencia para modelar el paso de mensajes entre objetos.**

Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada método de la clase.

Mientras que el diagrama de casos de uso permite el modelado de una vista business del escenario, el diagrama de secuencia contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario, y mensajes pasados entre los objetos.

Típicamente uno examina la descripción de un caso de uso para determinar qué objetos son necesarios para la implementación del escenario.

Si tienes modelada la descripción de cada caso de uso como una secuencia de varios pasos, entonces puedes "caminar sobre" esos pasos para descubrir qué objetos son necesarios para que se puedan seguir los pasos.

- **Diagramas de Colaboración para modelar el comportamiento de los objetos en el sistema.**

Un diagrama de colaboración es una forma alternativa al diagrama de secuencia de mostrar un escenario. Este tipo de diagrama muestra las interacciones entre objetos organizadas en torno a los objetos y los enlaces entre ellos.

Los diagramas de secuencia proporcionan una forma de ver el escenario en un orden temporal - qué pasa primero, qué pasa después. Los clientes entienden fácilmente este tipo de diagramas, por lo que resultan útiles en las primeras fases de análisis. UML es una consolidación de muchas de las notaciones y conceptos más usados orientados a objetos.

1.12. Recursos necesarios para el Proyecto

Para la elaboración de nuestro Proyecto contamos con los siguientes recursos:

- **Recurso Humano:** el personal que está a cargo de este Proyecto tendrá las siguientes funciones:

NOMBRE	RESPONSABILIDADES
JULIO CHÓEZ C.	Responsable de la parte de levantamiento de información y búsqueda de Clientes, Diseño y Desarrollo.
GIOMAR CHÓEZ C.	Responsable de Seguridades de la Base de Datos (DBA), Diseño y Desarrollo
SAÚL ZAMBRANO R.	Responsable de desarrollo y Distribución de Redes (levantamiento del Portal Web)

Tabla 1.1 Recurso Humano

- **Recurso de Software:** Para los diseños preliminares hemos utilizado Dreamweaver MX y Visio para lo que corresponde a los diagramas del proyecto. Además se utilizará lo siguiente:
 - Sistema operativo Windows XP
 - Leguaje de programación de java
 - Librería de java “.jar”
 - Base de datos MySQL 5
 - Tomcat 5 o glassfish para el servidor Server web
 - Maquina virtual de java JDK 6

- **Recurso de Hardware:** 1 CPU para servidor \$700. 1 dispositivo Bluetooth D-LINK \$35. Dos equipos que serán destinados para el desarrollo de la aplicación, levantamiento de la información, análisis, diagramas, creación de la base de datos, etc.



Figura 1.5 Dispositivo Bluetooth

Las características de los equipos son:

EQUIPO	CARACTERÍSTICAS
PORTÁTIL 1	AMD Turion X2, HP Pavilion dv6921la Notebook 2.00 Ghz, 2 GB de memoria, Disco Duro 160 GB.
PORTÁTIL 2	Intel Core 2 Duo. HP Pavilion dv4-1225la, 2.00 Ghz, 3 GB de memoria, Disco Duro 250 GB, con bluetooth.
SERVIDOR	Intel Core 2 Duo, 1.99 GB de RAM, 2.80 Ghz de memoria, Disco Duro de 250 GB.
DISPOSITIVO BLUETOOTH	D-LINK DBT-122 Bluetooth 2.0, Cobertura 10 metros.

Tabla 1.1 Características de los equipos

1.13. Análisis del Mercado

Actualmente este servicio no está muy explotada dentro de del área del territorio Ecuatoriano, la información se mueve de una manera rápida, ligera y veraz, con este servicio damos a nuestros usuarios la mayor cobertura posible sin necesidad de estar conectado a Internet, porque todas la personas no posee la disponibilidad del servicio de Internet, pero si contamos con un 80% de servicio de telefonía móvil con tecnología bluetooth, el cual es la razón que proyecto, sacar provecho de este servicio y complementar con la mensajería SMS. Este servicio actualmente lo están implementando los centros comerciales e instituciones educativas.

Dentro de una Unidad educativa va ser de mucha importancia este servicio, le prestamos todas las facilidades para que tenga acceso a consultas de varias peticiones, el beneficio de contar con un servicio de mensajes vía bluetooth y de envío de imágenes de publicidad será el fácil acceso a una información personal para los estudiantes y generación de ingresos por la publicidad enviada para los directivos de la unidad educativa.

El costo de este servicio para los estudiantes este servicio será totalmente gratuito.

1.14. Análisis Técnico

El servidor Bluetooth contara con el software actualizado, con una infraestructura muy sofisticada, agradable para el usuario.

Contaremos con los siguientes equipos:

- Un servidor.
- Un dispositivo bluetooth que ofrece el servicio de comunicación con los teléfonos de los usuarios que tengan la misma tecnología.

1.15. Análisis Administrativo

Para poder llevar una buena administración del personal contamos con 6 personas que ocupan los siguientes cargos:

- 1 jefe o líder de proyectos.
- 1 Soporte de usuario.
- 1 programador con conocimientos de base de datos.

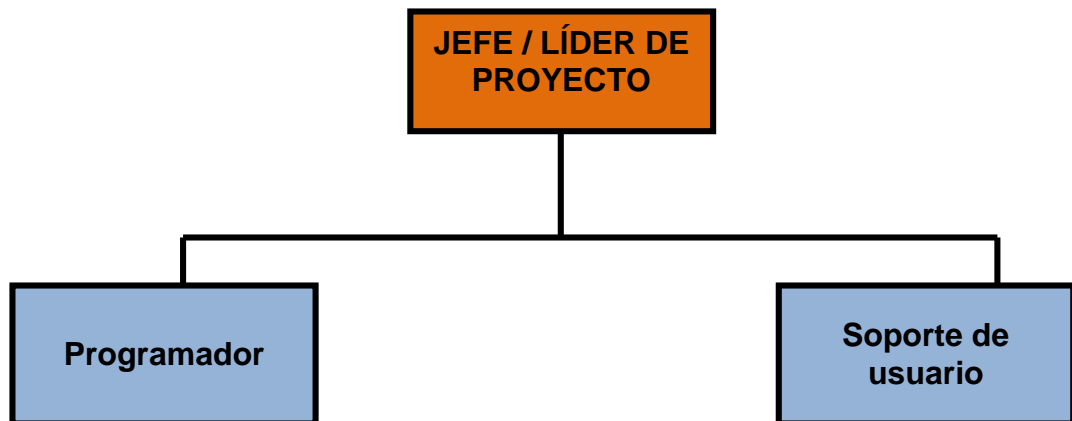


Figura 1.3. Administración del personal

1.15.1. Roles de cada uno de los cargos

- **Jefe / Líder del proyecto**

Es la figura encargada de coordinar, dirigir, controlar y responsabilizarse en último término de la ejecución del proyecto. El jefe de proyecto debe tener una autoridad real, una capacidad para tomar decisiones sobre la realización del proyecto siempre respetando los objetivos del proyecto.

La falta de autoridad o el exceso de independencia de algunas áreas informáticas son la causa de muchas dificultades en los proyectos.

La misión del jefe de proyecto tiene dos vertientes, por un lado la técnica, ha de concretar objetivos, adecuar el proyecto a las posibilidades de la empresa, gestionar los recursos, cambiar los métodos, etc., y por otro lado la de gestión de recursos humanos, materiales y de relación con otros departamentos y con el usuario.

En definitiva y como resumen las funciones del jefe de proyecto podrían resumirse como tres:

- **Planificar**, que es la única forma de asegurar la finalización del proyecto en el plazo adecuado sin dejarlo en manos de la improvisación.
- **Coordinar**, para que todas las acciones de los implicados en el proyecto tanto propios (componentes del equipo de proyecto) como externos (usuarios u otros departamentos), sean eficaces y no den lugar a conflictos.
- **Explotación de sistemas o aplicaciones**. Consta de previsión de fechas de realización de trabajos, operación general del sistema, control seguridad del sistema, supervisión de trabajos, etc.

- **Controlar**, para asegurar que se cumple lo planificado y se obtiene la calidad requerida.
- **Gestión y administración del propio Centro de Procesamiento de Datos.** Las funciones de gestión y administración de un Centro de Procesamiento de Datos engloban operaciones de supervisión, planificación y control de proyectos, seguridad y control de proyectos, seguridad general de las instalaciones y equipos financiera y gestión de los propios recursos humanos.

- **Programador**

- Se encarga de elaborar los programas que se ejecutan en las computadoras, modifican los existentes y vigilan que todos los procesos se ejecuten correctamente.
- Toman especificaciones de los sistemas realizados por los analistas y las transforman en programas eficientes y bien documentados para las computadoras.
- Deben tener los conocimientos suficientes del hardware para poder optimizar la utilización del equipo.

- Su función es extremadamente técnica y especializada deben poseer un nivel alto de conocimiento en la herramienta de desarrollo con la que se va a trabajar.

Además deberá tener conocimientos sobre:

- Administrar la estructura de la Base de Datos.
- Administrar la actividad de los datos.
- Administrar el Sistema Manejador de Base de Datos.
- Confirmar la seguridad de la Base de Datos.

- **Soporte de usuario**

El soporte de usuarios es una tarea que no muchas personas desean asumir, se necesita de mucha sensibilidad ante los usuarios, psicología, paciencia ya que todos los requerimientos de soporte están acompañados de críticas al producto y de necesidad urgente de solución, es por ello que la respuesta debe ser ágil, rápida y segura.

Debe tener el tacto suficiente para comprender la situación de quien lo necesita y ayudarlo en cualquier momento.

El soporte se extiende a quienes fabrican el producto (son los arquitectos del mismo). En ese caso, la persona que brinda soporte, se convierte en un canal de comunicación que a su vez debe captar la solución para multiplicarla con los demás usuarios del sistema.

Debe poseer una buena capacidad de comunicación, cierto carisma y mucho tacto para tratar cada tema. Teniendo presente de que quien solicita soporte tiene un problema, que tal vez sea de simple solución, pero justo en ese momento y para esa persona es uno de los más graves.

Un buen soporte de usuarios no es necesariamente la persona que mas conoce de tecnología, pero si debe ser quien más conoce a los mas eruditos en cada tecnología para aprovechar su conocimiento y que, como siempre, el cliente o usuario quede satisfecho.

1.16. Matriz de Riesgo

RIESGOS
Radio de transmisión
Rechazo de los usuarios
Tiempo de respuestas por envíos de mensajes masivos
Teléfonos con bluetooth no compatibles

Tabla 1.2 Matriz de riesgo

1.17. Matriz FODA

FORTALEZAS
1. Facilidad de manejo de contenidos
2. Envíos de imágenes publicitarias a dispositivos con bluetooth.
3. Envíos de mensajería a dispositivos con bluetooth.
4. Envíos de notas a los estudiante de la carrera
5. Envíos de materias a los estudiante de la carrera
6. Programación de múltiples contenidos

Tabla 1.3 Fortalezas del proyecto

OPORTUNIDADES
1. Comerciable
2. Falibilidad de datos

Tabla 1.4 Oportunidades del Proyecto

DEBILIDADES
1. Rango de cobertura
2. Trabaja en una sola arquitectura

Tabla 1.5 Debilidades del Proyecto

AMENAZAS
1. Virus
2. Voltaje de la máquina
3. Agentes externos

Tabla 1.6 Amenazas del Proyecto

1.18. Cronograma de Actividades del Proyecto

N° DE ACTIVIDAD	ACTIVIDAD	DIAS
1	Análisis y Definición de Requerimientos	8
2	Análisis y Definición de Objetivos y Alcances del proyecto	20
3	Definición de Misión, Visión y cronograma de trabajo del proyecto	5
4	Definición de Arquitectura y Metodologías a emplearse	7
5	Análisis de Casos de Uso, Diagrama de Objetos, Flujo de Información entre Objetos.	25
6	Diseño de la pantalla principal para cada módulo del proyecto	25
7	Codificación de cada módulo del proyecto	103
8	Pruebas del proyecto	20
9	Documentación del tomo II (Manual de usuario y técnico)	25
TOTAL		240

Tabla 1.7 Cronograma de actividades

CAPITULO 2

ANÁLISIS

2.1. Levantamiento de la información

Haciendo un estudio de mercado en la carrera de ingeniería de sistemas de la facultad de Matemáticas y Físicas tenemos un nicho de mercado bien amplio, del cual se realiza un análisis de la relación Universidad – Estudiante en el cual se garantiza el éxito del proyecto.

Encontramos que el estudiante para averiguar sus notas de las materias que este cursando necesita ir a los laboratorios de la carrera, ir a un cyber o cualquier lugar que tenga una computadora con conexión a internet.

Variando los cambios y usando ahora celulares que tengan la tecnología bluetooth se pretende demostrar lo escrito anteriormente que es de mucha importancia tener esta aplicación en la facultad. Para optimizar y estar al día en los avances tecnológicos.

Para agilizar este proceso se realizara una alternativa que mediante el uso desde un celular que tenga la tecnología bluetooth y utilizando unos parámetros requeridos acceder a la información de una manera más optima y oportuna, y estrechar la cadena de valor llegando a obtener la información requerida.

2.2. Análisis de requerimientos

Debido a los avances de la tecnología y la demanda de los estudiantes en la carrera de sistemas ya que es netamente tecnológico es necesario cambiar el esquema de los procesos, para obtener la información veraz y eficaz.

Debemos realizar una página que mediante un administrador o usuario pueda manipular el contenido de los mensajes y publicidad, así como también administrar los usuarios que van a utilizar la aplicación.

Esta aplicación también podrá presentar varios reportes para tener un enfoque de cómo se están realizando los procesos de envío y recepción y además presenta reportes estadísticos.

2.3. Análisis de riesgos

En si el proyecto con el dispositivo bluetooth representa una vulnerabilidad por ser tecnología de ondas de radio de corto alcance (2.4 gigahertzios de frecuencia) cuyo objetivo es el simplificar las comunicaciones entre dispositivos informáticos, como ordenadores móviles, teléfonos móviles.

Como todo proyecto necesita la parte económica para estabilizarse en sus operaciones, una inestabilidad política y económica en el país afectaría también a nuestro proyecto, para lo cual la conexión que utilizaremos para el envío y recepción de mensajes como imágenes publicitarias se las realizara mediante un dispositivo bluetooth conectado al servidor.

Con esta tecnología que permite comunicaciones, incluso a través de obstáculos, a distancias de hasta unos 10 metros. Limita a que los estudiantes que necesitan realizar sus consultas tengan que estar en dicho rango con el servidor.

Otro riesgo que representa es el Bluejacking un término que se refiere a una técnica consistente en enviar mensajes no solicitados entre dispositivos [Bluetooth](#), como por ejemplo [teléfonos móviles](#), [PDAs](#) o portátiles.

La tecnología Bluetooth tiene un alcance limitado de unos 10 metros normalmente en dispositivos pequeños (como teléfonos móviles) aunque otros aparatos más grandes (como portátiles) con transmisores más potentes pueden alcanzar los 100 metros.

Bluejacking es bastante inofensivo, pero como mucha gente no sabe qué ocurre piensan que su teléfono móvil tiene un virus o alguien entró a él. Normalmente un bluejacker sólo enviará un mensaje de texto, aunque en los modelos de teléfonos más recientes es posible enviar también imágenes y sonido.

Actualmente hay varios programas utilizados para esta práctica (como Bluetooth Messenger, Easy Jack, etc.), aunque el más utilizado es Mobiluck.

2.4. Diagrama Entidad Relación

El Diagrama Entidad Relación representa las relaciones entre los Objetos de Datos, es la notación que se usa para realizar la actividad de Modelado de Datos, los atributos de cada objeto de datos señalados en el DER se pueden describir mediante una descripción de Objetos de Datos.

Su objetivo fundamental va a ser la creación, edición y verificación de diagramas Entidad/Relación y su posterior paso al modelo lógico estándar. Asimismo, se incluirá un diccionario de datos en el que el usuario podrá definir sus datos.

La herramienta tendrá también la posibilidad de detectar los posibles errores que los usuarios puedan cometer en su creación. También se podrá generar un modelo lógico correspondiente al modelo conceptual que se está realizando.

Para nuestro Diagrama Entidad Relación se ha considerado las siguientes Entidades con sus respectivos campos:

2.4.1. Diagrama DER

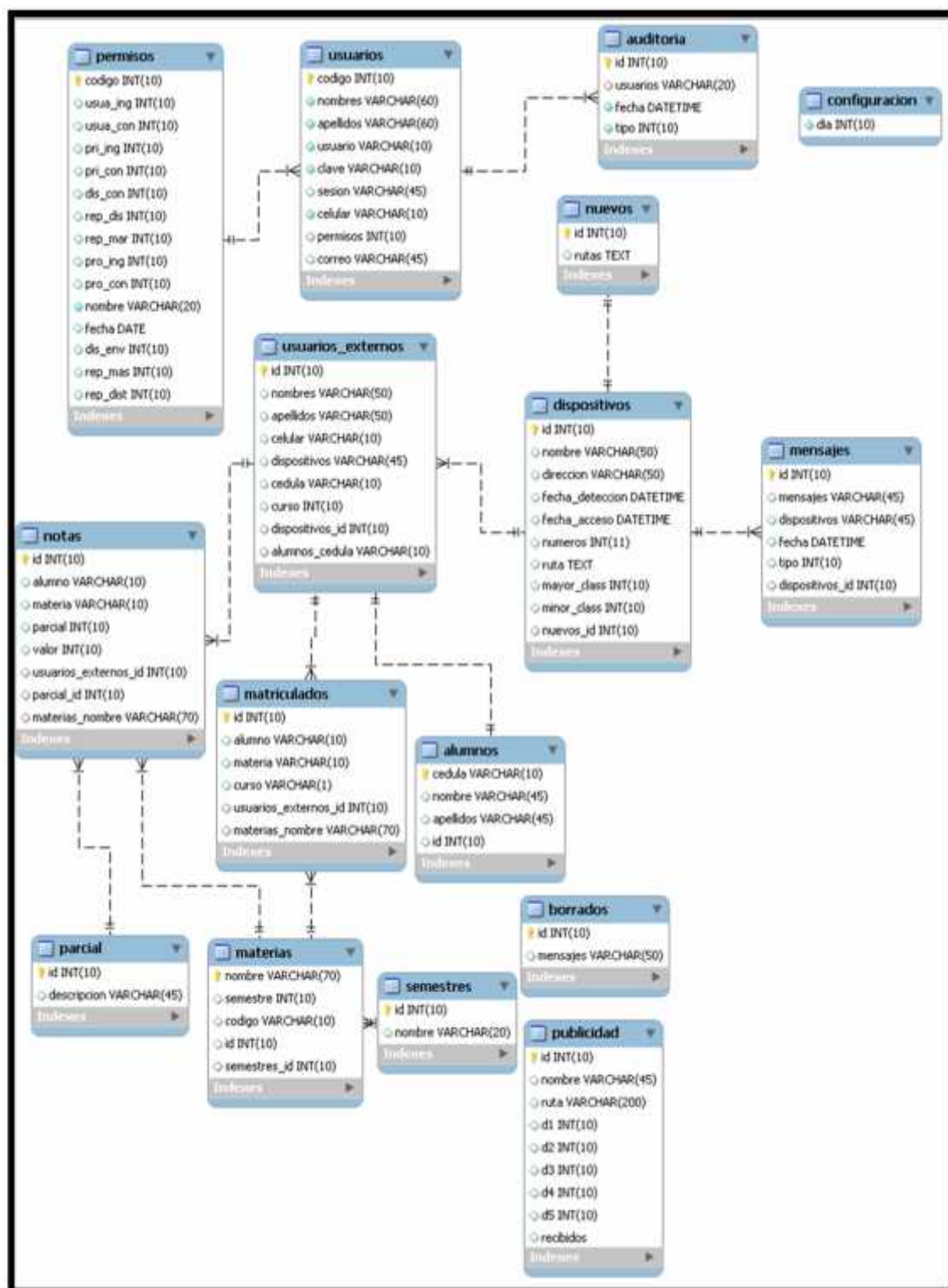


Figura 2.1 Diagrama de Entidad – Relación

2.5. Diagrama de flujo de procesos

2.5.1. Flujo de procesos

El Diagrama de Flujo de Procesos es una representación gráfica de la secuencia de todas las operaciones, secuencias, esperas y los almacenamientos que ocurren durante el proceso de ingreso, registro, suscripción, consultas e impresión de datos al visitar nuestro Portal. Incluye, además, la información que se considera deseable para el análisis, por ello se ha considerado importante los siguientes Flujos de Procesos:

2.5.2. Procesos del portal Web

2.5.2.1. Proceso de ingreso de usuario

- Ingreso al Portal Web
- Validación del usuario
- Verificación del Sitio con permisos asignados
- Acceso al menú principal
- Ingreso a paginas autorizadas

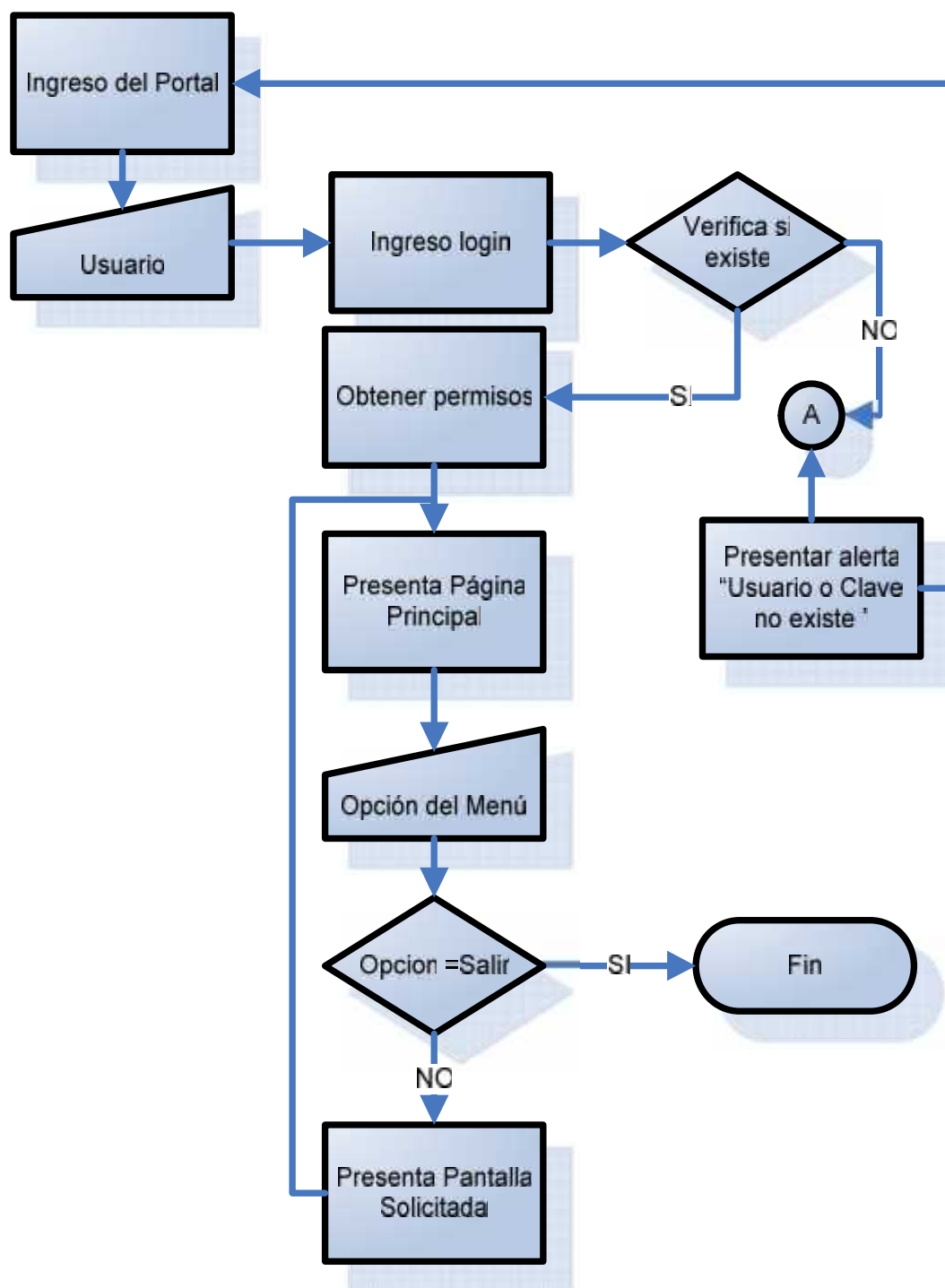


Figura 2.2 Diagrama de ingreso de usuarios

2.5.2.2. Proceso de respuesta del servidor ante una petición

- Esperando petición
- Validación de la petición
- Consulta con la base de datos
- Consulta del dispositivo a enviar respuesta
- Registro de la petición de envío

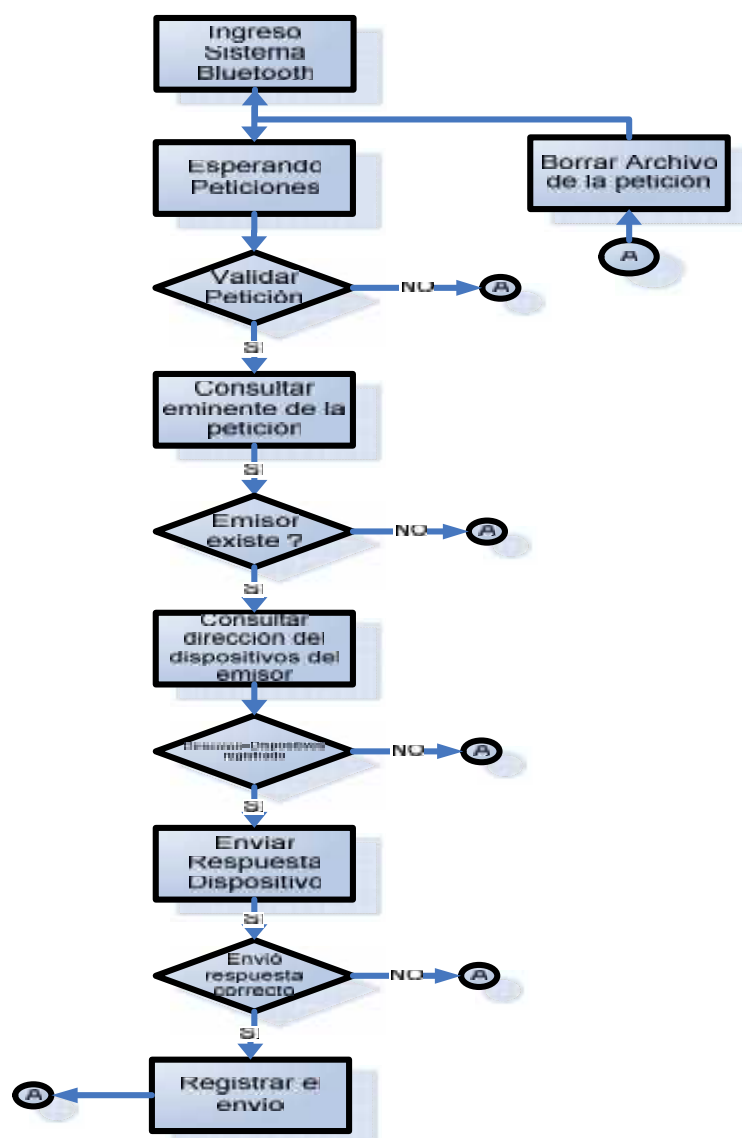


Figura 2.3 Diagrama de respuesta del servidor

2.5.2.3. Proceso de envío de mensajes

- Consultar dispositivos activos
- Seleccionar los dispositivos
- Enviar el mensaje de texto
- Presentar en la consola errores o éxito de envío

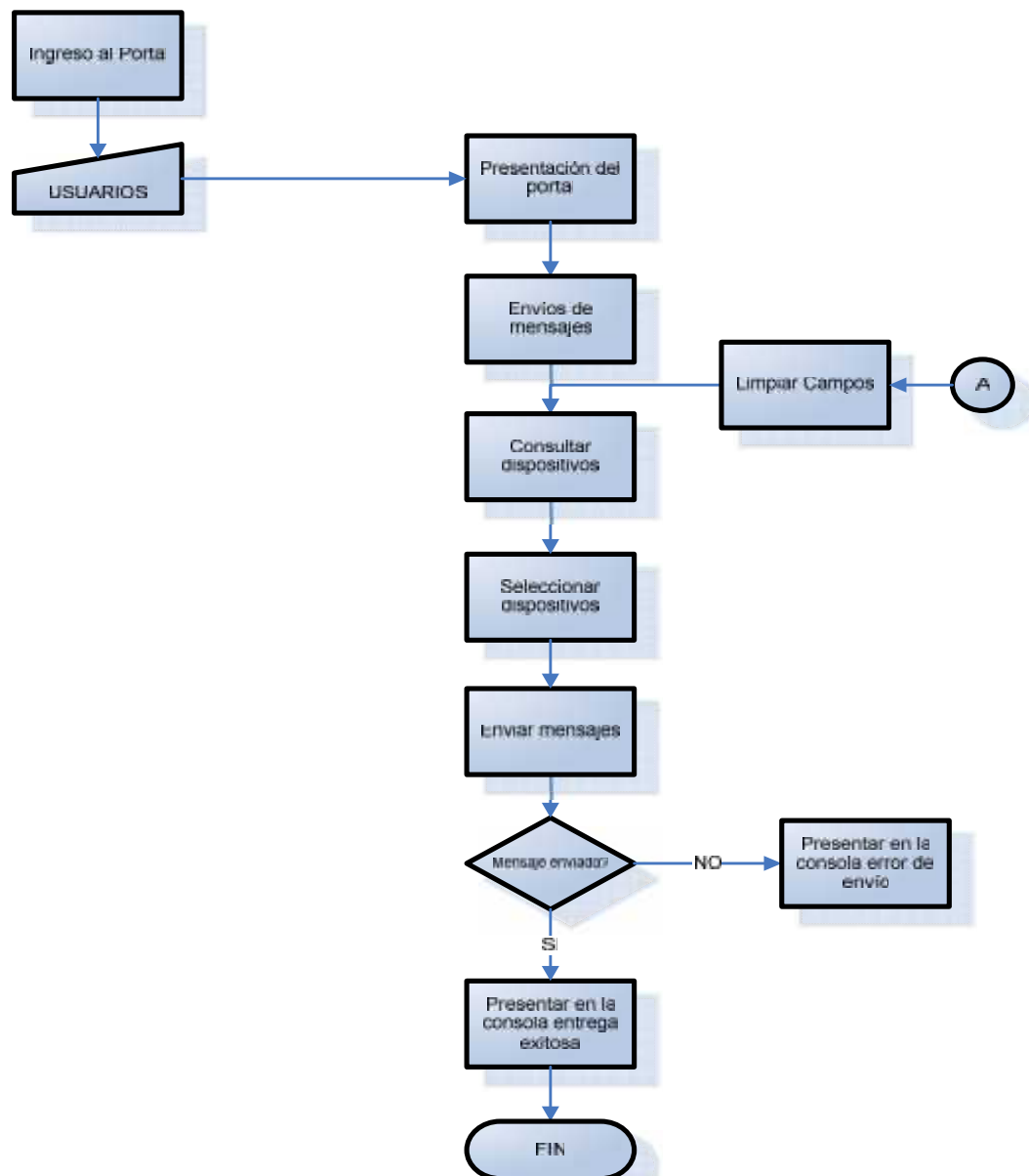


Figura 2.4 Diagrama de envío de mensajes

2.5.2.4. Proceso de envío de publicidad

- Activar el envío de publicidad
- Consulta de dispositivos activos
- Envío de publicidad
- Registro de publicidad ignorada o aceptada

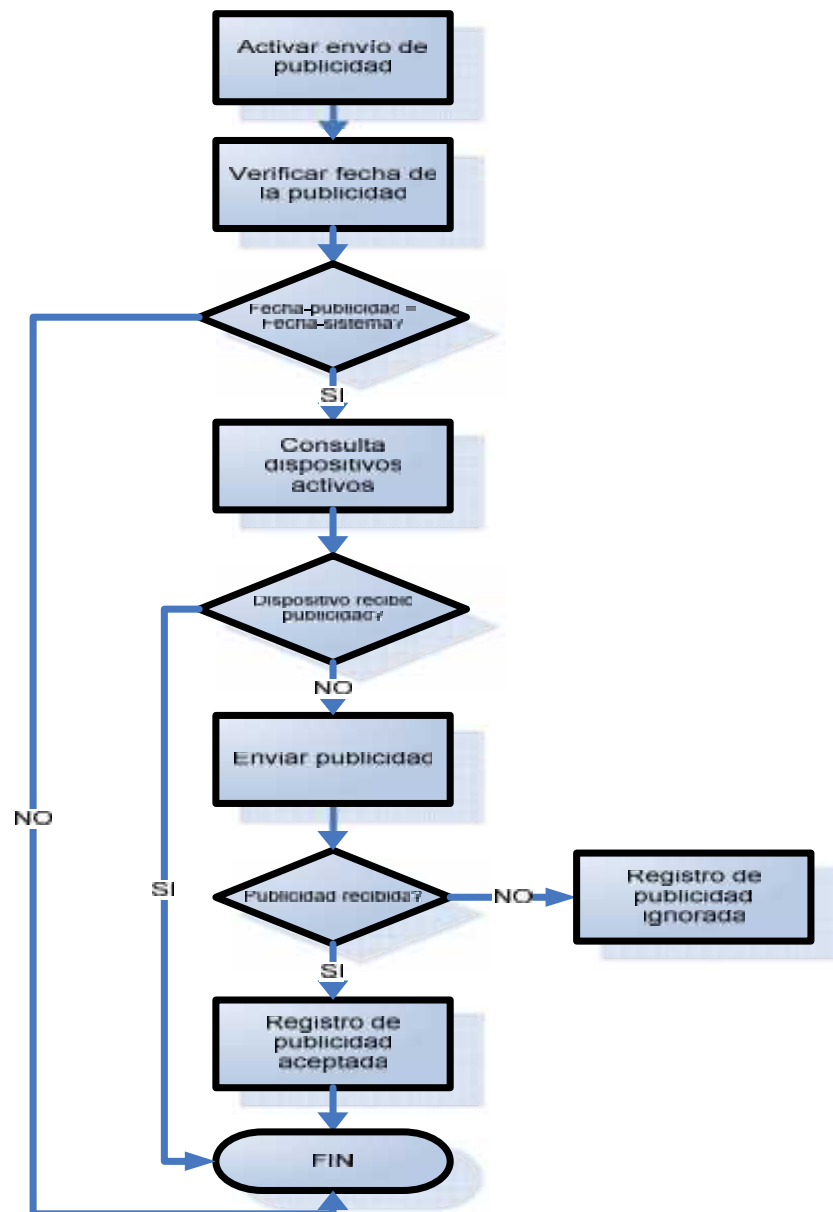


Figura 2.5 Diagrama de envío de publicidad

2.5.2.5. Proceso de reportes

- Ingreso al Portal Web
- Visualización del Sitio con permisos asignados
- Acceso al menú principal
- Presentación de reporte generados por el sistema
- Visualización de resultados

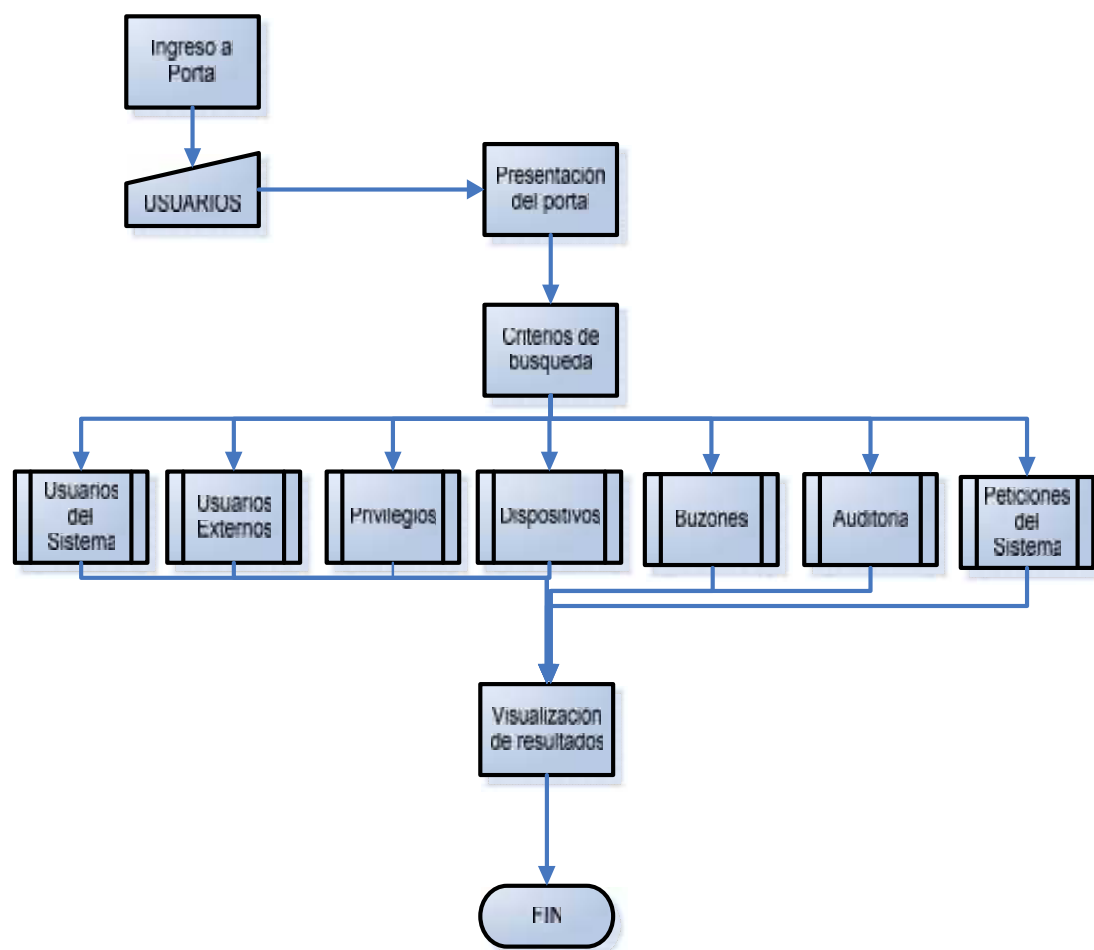


Figura 2.6 Diagrama de reportes

2.5.2.6. Proceso de ingreso de usuarios externos

- Ingreso al Portal Web
- Visualización del Sitio con permisos asignados
- Acceso al menú principal
- Ingresar datos del usuario
- Asignación de dispositivo móvil detectado

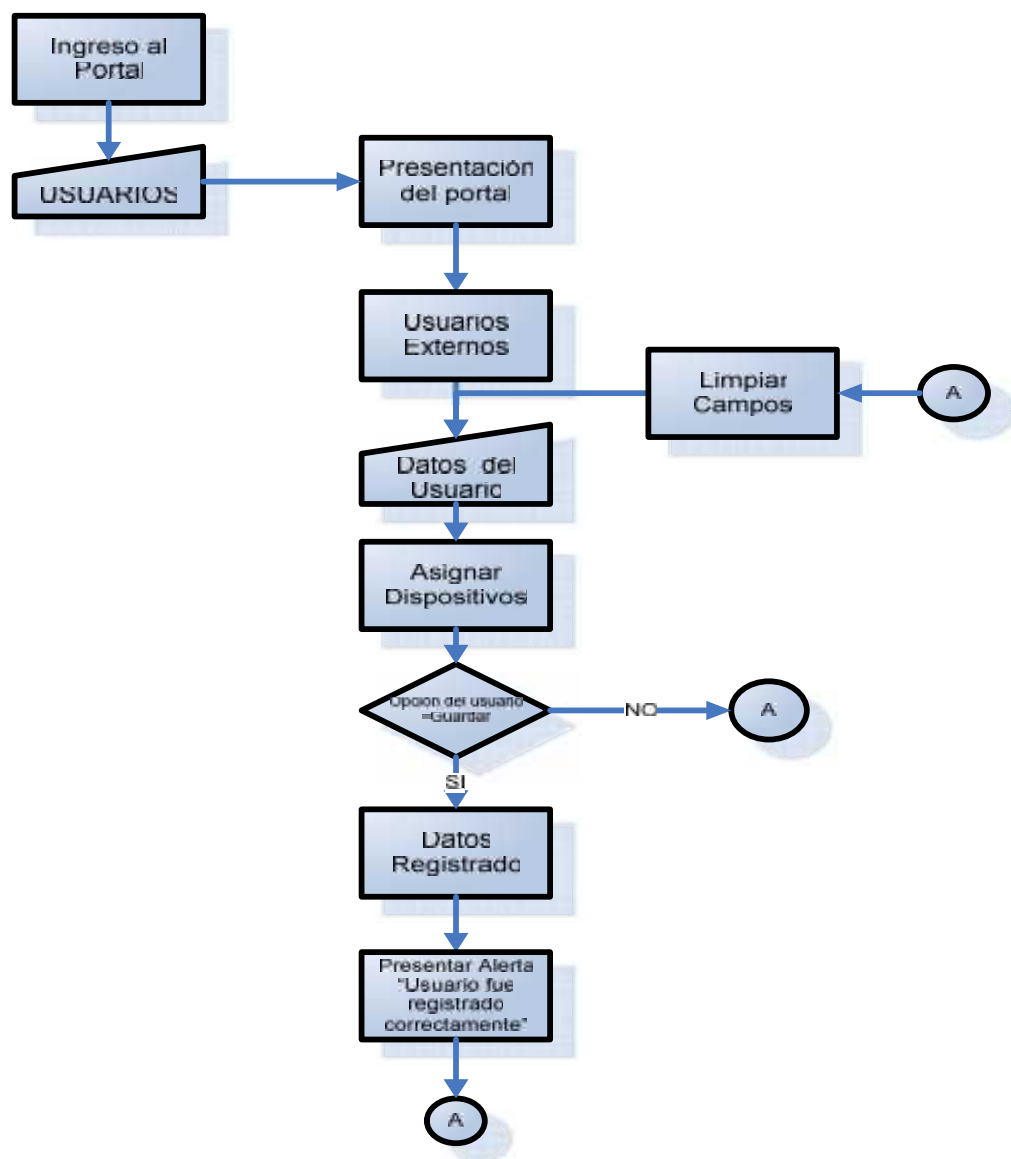


Figura 2.7 Diagrama de ingreso de usuarios externos

2.6. Diagrama de Casos de Uso

Ingreso al portal Web

2.6.1. Primer Nivel

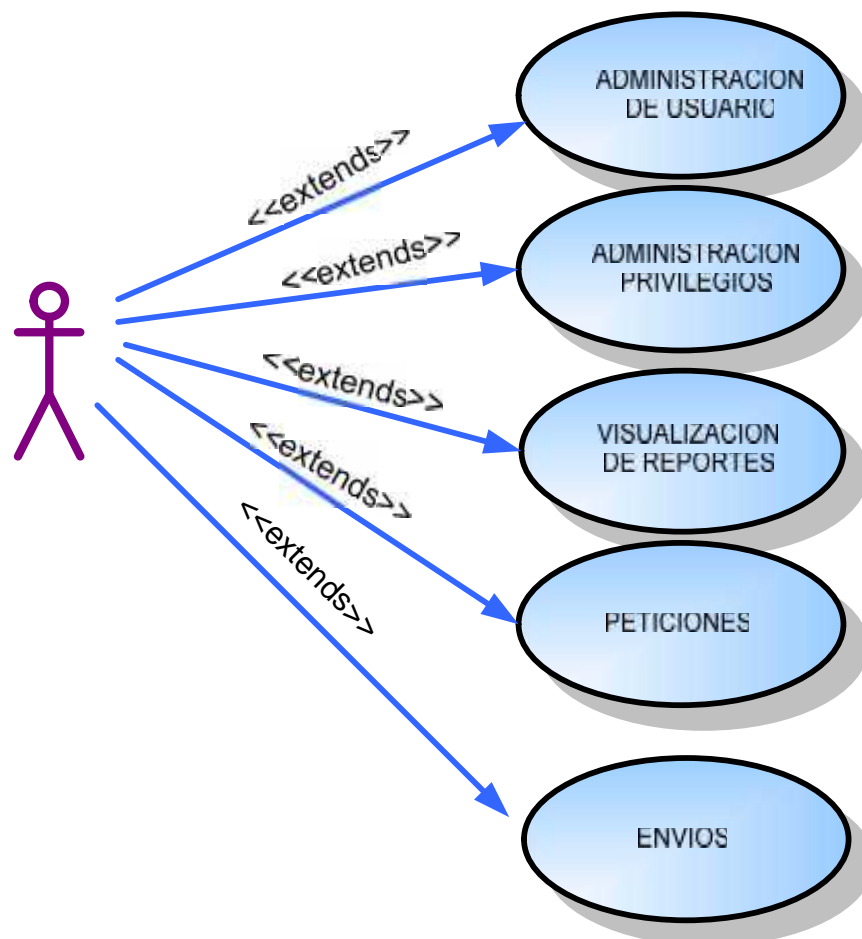


Figura 2.8 Diagrama de primer nivel

2.6.2. Segundo Nivel

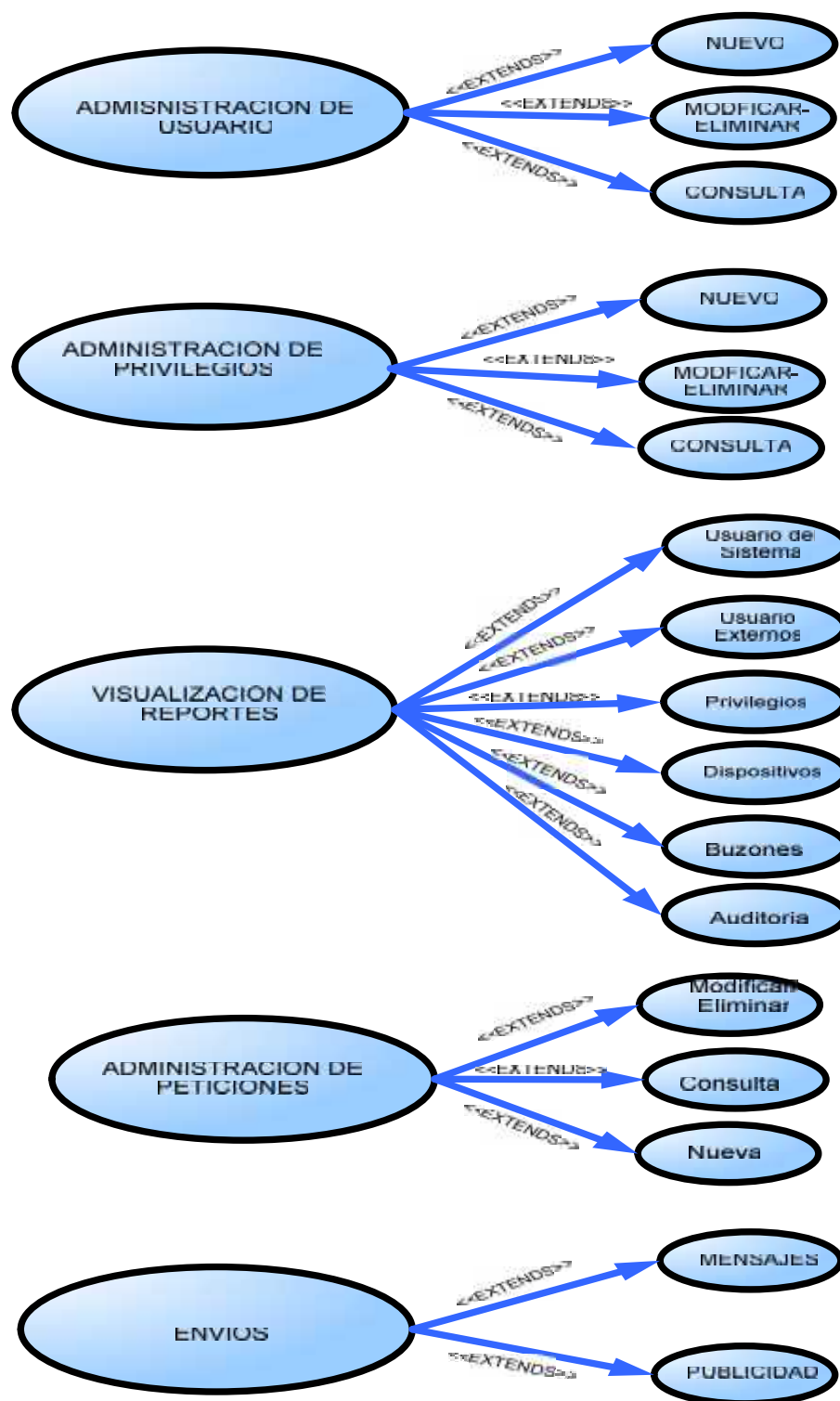


Figura 2.9 Diagrama de segundo nivel

2.6.3 Tercer Nivel

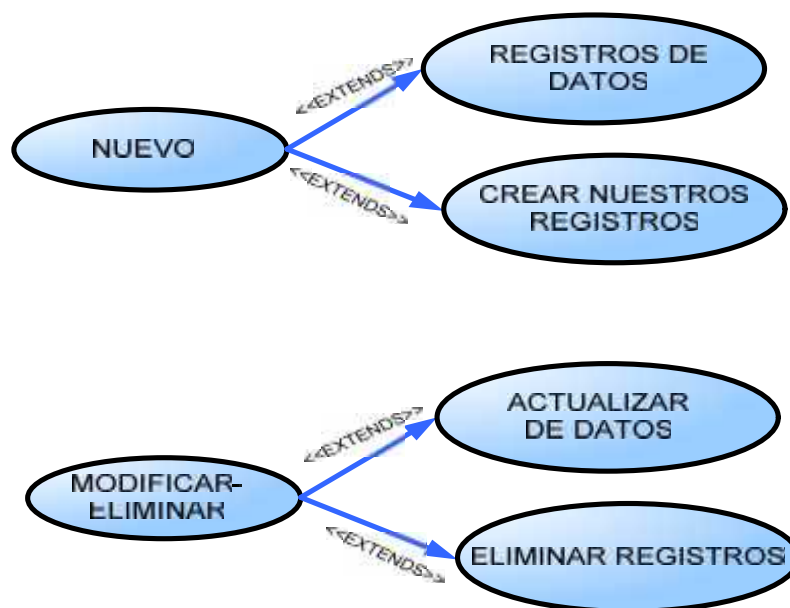


Figura 2.10 Diagrama de tercer nivel

2.6.4 Descripción de casos de uso 1er. Nivel

Nombre:	Administración de Usuarios
Alias:	
Actores:	Usuario del Sistema
Función:	<p>Ingresar, consultar, modificar y eliminación de registros de los usuarios externos y del sistema.</p> <p>El Usuario del Sistema tiene que registrar todos sus datos. El sistema debe validar lo siguiente:</p> <ol style="list-style-type: none"> 1. Que se ingrese el primer nombre y el primer apellido. 2. Que se ingrese número celular. 3. Se asigne un usuario y clave y confirmación. 4. Se le asigne el privilegio. <p>Los usuarios externos tienen que registrar sus datos. El sistema debe de validar:</p> <ol style="list-style-type: none"> 1. Que se ingrese nombre y apellidos. 2. Ingresar número de celular. 3. ingresar la cedula. 4. asignar el dispositivo al usuario.
Referencias:	

Tabla 2.1 Administración de usuarios

Nombre:	Administración de Privilegios
Alias:	
Actores:	Usuario del Sistema
Función:	Permitir ingresar, modificar y consultar privilegios para los usuarios del sistema por cada modulo.
	El Usuario administrador puede dar acceso a otro usuario a este modulo. Este puede seleccionar o deseleccionar las opciones por cada modulo.
	Consulta los permisos y dependiendo el tipo de usuario este podrá modificar o eliminar usuarios y privilegios.
Referencias:	

Tabla 2.2 Administración de Privilegios

Nombre:	Administración de Envíos
Alias:	
Actores:	Usuario del Sistema
Función:	Envíos de mensajes y publicidad a los dispositivos que utilicen tecnología bluetooth
	El Sistema realizará las siguiente opciones: <ul style="list-style-type: none"> – Detecta los dispositivos que están dentro del alcance del sistema. – Envíos de mensajes a los dispositivos que estén dentro del rango d frecuencia. – Respuesta al servidor sobre en envió del mensaje. – Envía imágenes publicitarias a los celulares con bluetooth dentro del alcance del sistema. – Se podrá crear, modificar y eliminar nuevas publicidades con las opciones por días de envió
Referencias:	

Tabla 2.3 Administración de envíos

Nombre:	Visualización de Reportes
Alias:	
Actores:	Usuario del Sistema
Función:	Presentación de varios reportes
	<p>El Sistema presentara los siguiente reportes:</p> <ul style="list-style-type: none"> – Reporte de dispositivos encontrados. – Reportes de mensajes de entrada, salida, no recibidos. – Exportar documentos en Excel y para adobe *.pdf. – Búsqueda de datos en los buzones por rango de fechas establecidas. – Imprimir reportes en todos las consultas – Reportes diarios de consultas de mensajes.
Referencias:	

Tabla 2.4 Visualización de Reportes

Nombre:	Configuración de peticiones
Alias:	
Actores:	Usuario del Sistema
Función:	Ingreso y modificación de peticiones del sistema
	<p>El usuario podrá ingresar y modificar las peticiones, argumento y las respuestas que dará el servidor.</p> <p>Podrá consultar las peticiones existentes en el sistema.</p>
Referencias:	

Tabla 2.5 Configuración de peticiones

2.7. Diagrama de Contexto de Arquitectura y Funcionalidad

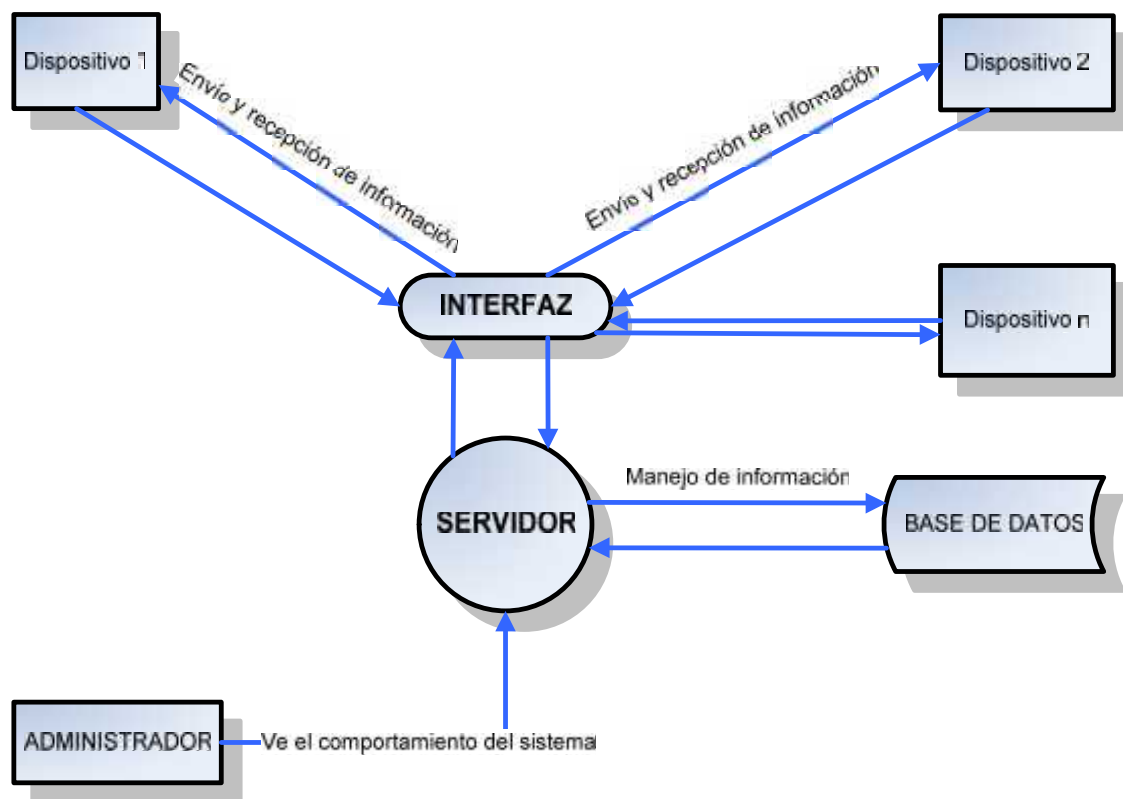


Figura 2.11 Diagrama de funcionamiento del sistema

CAPITULO 3

DISEÑO

3.1. Diseño de interfaz

El diseño del interfaz es uno de los elementos "claves" en la realización del programa. Es el conjunto de pasos que seguirá el usuario, durante todo el tiempo que se relacione con el sistema, detallando lo que envía y recibe mensajes así como el manejo que se le dará en cada momento, y las acciones que realizará, así como las respuestas que el sistema le dará.

Su objetivo es que las aplicaciones sean más atractivos y además, hacer que la interacción con el usuario sea lo más intuitiva posible. Para ello deberá de realizarse mediante patrones de calidad ya que es el éxito del proyecto la integración entre el usuario y el software. Las interfaces de usuario de aplicaciones Web son muy flexibles, permitiendo una gran variedad de recursos gráficos, lo cual nos hace ser más cuidadoso en la facilidad de uso, así como la funcionalidad.

Es muy frecuente encontrar interfaces para la Web en las que no se consideran al usuario, el sistema tendrá que desempeñar las funciones que fueron definidas en los alcances del proyecto. Siempre habrá cambios en el desarrollo del diseño del sistema, eso siempre se da con las peticiones del usuario pero no alteran con los objetivos de la aplicación.

No olvidar que es el usuario quien determina el éxito o fracaso de una aplicación de software, sin importa el nivel de experiencia que tenga y así logra la facilidad de uso requerida para el software. Lo ideal es que las interfaces de aplicaciones para la Web sean fáciles de usar, que sean agradables al usuario, que contengan elementos bien distribuidos, que la interfaz realmente contenga la información que el usuario espera de ella, ya que es una relación directa de mensajes de texto desde un celular que llegara al software que interactuara con la aplicación.

3.2 Página Login

Permite validar los usuarios que deseen ingresar al sistema, digitando su nombre de usuario y su clave de acceso los cuales serán analizados para luego permitir su acceso o denegarlo.



Figura 3.1 Página de ingreso al sistema

3.3 Página Principal

En esta página se encuentran todas las opciones de control de nuestro sistema presentando el usuario que ha ingresado a la aplicación con su nivel de privilegios.



Figura 3.2 Página Principal

3.4 Página ingreso de usuarios del sistema

En esta página se ingresarán los datos de los usuarios que podrán acceder al sistema, además se les otorgara el privilegio correspondiente.

Figura 3.3 Ingreso de usuarios del sistema

3.5 Página consulta de usuarios del sistema

Esta página visualiza todos los usuarios del sistema, además podrá modificar sus datos personales o eliminarlos e imprimir.



Figura 3.4 Consulta de usuarios del sistema

3.6 Página ingreso de usuarios externos del sistema

Esta página permite el ingreso de los usuarios que estarán registrados con su dispositivo detectado por el sistema.



Figura 3.5 Ingreso de usuarios externos

3.7 Página consulta de usuarios externos del sistema

Permite la consulta modificación y eliminación de usuarios externos así como el reporte y su impresión.



Bienvenido : JULIO CHÓEZ

Principal Usuarios Privilegios Envios Reportes 12 Peticiones

CONSULTA DE USUARIOS EXTERNOS

Reporte en : Pdf Modificar Eliminar Exportar Imprimir

	Nombres	Apellidos	Celular	Dispositivo	Nombre Dispositivo
<input type="radio"/>	GOMAR PAOLA	CHÓEZ CATUTO	090192403	0018AFC75A5E	Milenita
<input type="radio"/>	JULIO CÉSAR	CHÓEZ CATUTO	097344226	001D6E8EA947	Julian
<input type="radio"/>	PAOLA CAROLINA	CATUTO CEDERO	090192402	00219E74F015	Glomy-Love
<input type="radio"/>	TOMÁS LUIS	FERRARI MORA	088149568	00180FB4627B	Nokia 6131
<input type="radio"/>	JUAN SEBASTIAN	CASTRO ROBLES	088568944	001E3BAE3C07	Nokia 5200
<input type="radio"/>	FABRICIO ENRIQUE	MOREIRA JARA	097564854	001ADC8FC033	Federico
<input type="radio"/>	LOURDES ESTEFANIA	RUBIO ARMAS	097356246	001E3BAE3C07	Nokia 5200
<input type="radio"/>	MILENA PAOLA	FEBRES CORDERO	090195628	001E3BAE3C07	Nokia 5200

Figura 3.6 Consulta de usuarios externos

3.8 Página ingreso de privilegios

El usuario podrá ingresar el nombre del privilegio y asignar los permisos a diferentes módulos disponible en el sistema.



Bienvenido : JULIO CESAR CHÓEZ CATUTO

Principal Usuarios Privilegios Envios Reportes 12 Peticiones

INGRESO DE PRIVILEGIOS DE USUARIOS

Nombre :

Menu : SELECCIONE EL MENU

PRIVILEGIO PARA ESE MODULO

Figura 3.7 Ingreso de privilegios

3.9 Página consulta de privilegios

Permite la consulta así como su modificación, eliminación e impresión de los privilegios existentes en el sistema.



Figura 3.8 Consulta de privilegios

3.10 Página envío de mensajes

Permite el envío de mensajes desde la aplicación a todos los dispositivos dentro del rango de cobertura que el sistema ha detectado.



Figura 3.9 Envío de mensajes

3.11 Página consulta de envío de publicidad

Permite consultar las imágenes publicitarias que se pueden enviar a los dispositivos de los usuarios externos, así como también su modificación, eliminación y reporte.



Figura 3.10 Consulta de envío de publicidad

3.12 Página reporte dispositivos detectados

Permite la visualización de los dispositivos que fueron detectados por el sistema, su fecha de detección y la fecha de su ultimo acceso. Se podrá exportar en formato excel y pdf, así como su impresión.



Figura 3.11 Dispositivos detectados

3.13 Página reporte buzón de entrada

Permite apreciar todos los mensajes que desde los dispositivos de los usuarios externos han enviado hacia el sistema.

Dispositivo	Nombre	Mensaje	Fecha
001BAFC75A5E	Nokia 6120	0919598342.notes	2009-05-12 09:55:44.0
001BAFC75A5E	Nokia 6120	090192403.notes	2009-05-12 09:58:31.0
001BAFC75A5E	Nokia 6120	090192403.notes	2009-05-12 10:02:28.0
001D6E8EA947	Julian	0919597187.097344226	2009-05-13 21:36:29.0

Figura 3.12 Reporte de buzón de entrada

3.14 Página reporte buzón de salida

Permite visualizar los mensajes que desde el sistema envió a los dispositivos encontrados. Así como se podrá realizar reportes tanto en formato excel como en pdf y su impresión.

Dispositivo	Nombre	Mensaje	Fecha
001D6E8EA947	Julian	Por favor ingrese el mensaje a enviar...	2009-05-11 22:51:58.0
001BAF623C5D	Julioch	Por favor ingrese el mensaje a enviar...	2009-05-11 22:55:07.0
00210688FA26	BlackBerry 8100	Por favor ingrese el mensaje a enviar...	2009-05-11 22:55:22.0
001D6E8EA947	Julian	Por favor ingrese el mensaje a enviar...	2009-05-11 22:55:36.0
001BAFC75A5E	Nokia 6120	Por favor ingrese el mensaje a enviar...	2009-05-12 09:51:32.0
001BAFC75A5E	Nokia 6120	Por favor ingrese el mensaje a enviar...	2009-05-12 10:12:40.0

Figura 3.13 Reporte de buzón de salida

3.15 Página reporte buzón no enviados

Permite ver aquellos mensajes que ya sea porque los dispositivos estuvieron fuera del rango en el envío o fueron rechazados por parte del usuario no fueron entregados.



Bienvenido : julio choez

Principal Usuarios Privilegios Envios Reportes 12 Peticiones

Reporte en : Pdf Exportar Imprimir Buscar

Visualización : MENSUAL

Dispositivo	Nombre	Mensaje	Fecha
001BAFC75A5E	Nokia 6120	Por favor Ingrese el mensaje a enviar...	2009-05-12 09:50:58.0
001E3BAE3C07	Nokia 5200	Por favor Ingrese el mensaje a enviar...	2009-05-12 09:51:01.0
001BAFC75A5E	Nokia 6120	Por favor Ingrese el mensaje a enviar...	2009-05-12 09:51:04.0
001E3BAE3C07	Nokia 5200	Por favor Ingrese el mensaje a enviar...	2009-05-12 09:51:05.0
00180FB4627B	Nokia 6131	Por favor Ingrese el mensaje a enviar...	2009-05-14 10:40:41.0
00180FB4627B	Nokia 6131	Por favor Ingrese el mensaje a enviar...	2009-05-14 10:50:05.0

Figura 3.14 Reporte de buzón no recibidos

3.16 Página reporte buzón borradores

Permite ingresar, modificar, eliminar una especie de plantillas para facilitar la escritura de mensajes a los usuarios del sistema cuando requieran enviar un mensaje de texto.



Bienvenido : JULIO CHÓEZ

Principal Usuarios Privilegios Envios Reportes 12 Peticiones

BORRADORES

Ingresar Modificar Eliminar Imprimir

	Titulo	Contenido
<input type="radio"/>	OFERTAS LABORALES	SE SOLICITA ESTUDIANTE DE LA CARRERA DE SISTEMAS PARA CUBRIR UNA VACANTE EN
<input type="radio"/>	SUSPENSION DE CLASES	A TODOS LOS ESTUDIANTES SE LES COMUNICA QUE EL DIA DD DE MM QUEDAN SUSPENDIDAS LAS ACTIVIDADES DEBIDO A
<input type="radio"/>	REUNION	SE COMUNICA QUE EL DIA DD HABRA UNA REUNION PARA
<input type="radio"/>	SEMINARIOS	EL DIA DD DE MM HABRA UN SEMINARIO EN EL SALON DE ACTOS SOBRE ... DICTADO POR

Figura 3.15 Reporte de borradores

3.17 Página reporte contador

Permite ver un grafico en el cual se muestran los mensajes enviados, no enviados y recibidos por la aplicación de acuerdo a la opción que el usuario elija por día, fecha, mes, etc.

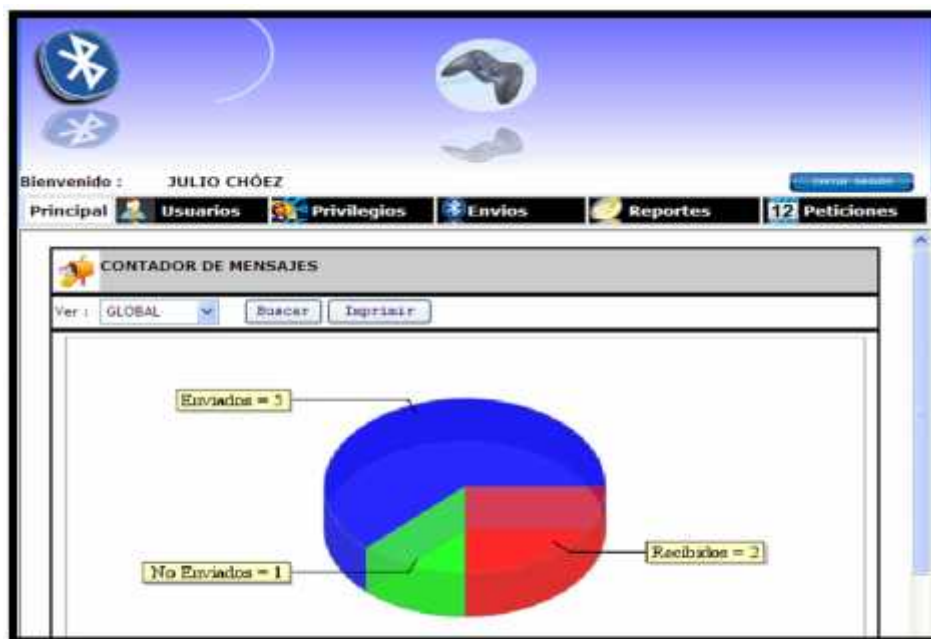


Figura 3.16 Reporte de contador de mensajes

3.18 Página auditoria

Permite realizar un control de los usuarios que ingresan al sistema y su manejo.

Nombres	Apellidos	Usuario	Fecha	Acción
JULIO CESAR	CHÓEZ CATUTO	admín	2009-06-05 21:03:29	INGRESO AL SISTEMA
JULIO CESAR	CHÓEZ CATUTO	admín	2009-06-05 21:48:39	INGRESO AL SISTEMA
JULIO CESAR	CHÓEZ CATUTO	admín	2009-06-05 21:52:53	SALIO DEL SISTEMA
JULIO CESAR	CHÓEZ CATUTO	admín	2009-06-05 21:53:00	INGRESO AL SISTEMA
JULIO CESAR	CHÓEZ CATUTO	admín	2009-06-05 22:33:18	SALIO DEL SISTEMA
JULIO CESAR	CHÓEZ CATUTO	admín	2009-06-05 22:41:50	INGRESO AL SISTEMA
JULIO CESAR	CHÓEZ CATUTO	admín	2009-06-06 18:24:24	INGRESO AL SISTEMA
JULIO CESAR	CHÓEZ CATUTO	admín	2009-06-06 18:26:00	SALIO DEL SISTEMA
JULIO CESAR	CHÓEZ CATUTO	admín	2009-06-06 18:26:20	INGRESO AL SISTEMA
JULIO CESAR	CHÓEZ CATUTO	admín	2009-06-06 18:28:45	SALIO DEL SISTEMA
JULIO CESAR	CHÓEZ CATUTO	admín	2009-06-06 18:28:59	INGRESO AL SISTEMA

Figura 3.17 Auditoria

3.19 Página Más Usados

Permite visualizar los 10 dispositivos que han utilizado más el sistema Servidor Bluetooth.



Figura 3.18 Más usados

3.20 Página de ingreso de peticiones

Permite ingresar los parámetros de las peticiones que el usuario externo podrá consultar desde su celular.

Figura 3.19 Ingreso de peticiones

3.21 Página de consulta de peticiones

Permite consultar las peticiones que se encuentran en la base de datos. Así como también eliminarlas o imprimir.

	Petición	Argumento	Respuesta
<input type="checkbox"/>	Ayudantía	Redes	Sábados y domingos de 10:00 - 12:00 en el Lab 1
<input type="checkbox"/>	Noticias	1 semestre	Sábado 20 de junio no habrá asistencia
<input type="checkbox"/>	Noticias	2 semestre	Profesor Eduardo Alvarado no asistirá durante esta semana
<input type="checkbox"/>	Novedades	Deportivas	Mes de Julio Campeonato Interno de Fútbol
<input type="checkbox"/>	Novedades	Seminarios	Seminario de CISCO informes al 2564765 - Ing. Abel Alarcón
<input type="checkbox"/>	Exámenes	1 parcial	del 22 de junio al 26 de junio
<input type="checkbox"/>	Exámenes	2 parcial	del 7 al 11 de septiembre
<input type="checkbox"/>	Laboratorio	LAB1	6:00 - 7:00 de lunes a viernes
<input type="checkbox"/>	Laboratorio	LAB2	8:00 - 9:00 martes y jueves
<input type="checkbox"/>	Laboratorio	LAB3	7:00 - 8:00 lunes y miércoles

Figura 3.20 Consulta de peticiones

3.22 Diagrama de Flujo de ventanas

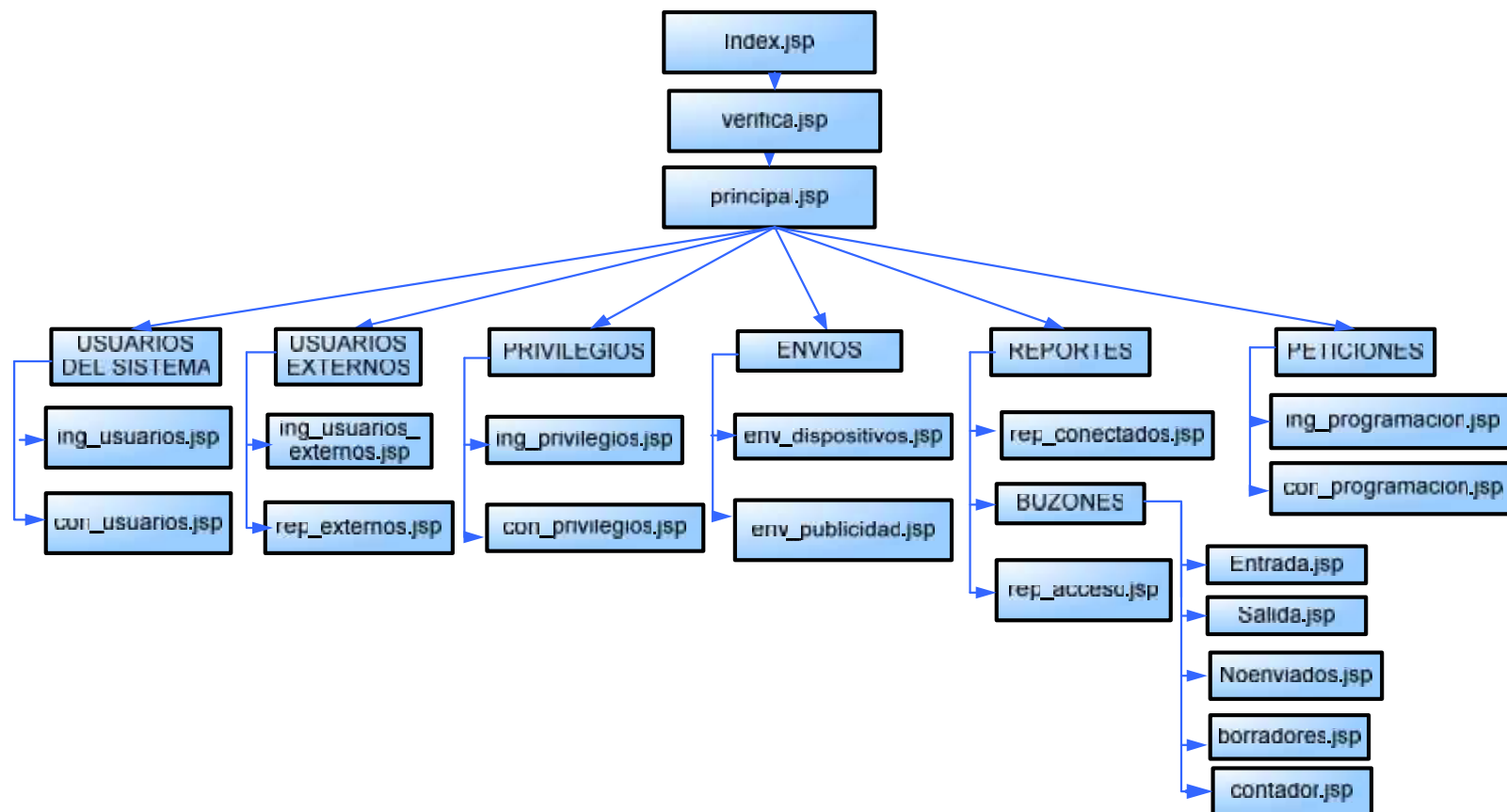


Figura 3.21 Diagrama de flujo de ventanas

CAPITULO 4

DESARROLLO DEL SOFTWARE

4.1. Antecedente

Para el desarrollo de esta herramienta utilizamos el método de cascada del desarrollo, es decir, dividimos la herramienta por módulos dependiendo de la importancia de los mismos y realizando las pruebas pertinentes.

4.2. Procesos Principales

La aplicación que desarrollamos tiene como proceso esencial la interacción con dispositivos BLUETOOTH. Tenemos como herramientas la base de datos MySQL Server 5.0, esta herramienta se la puede descargar www.mysql.com, es aquí donde se realizará todas las tablas del sistema que van interactuar con todas las aplicaciones.

Como es una aplicación en Java, utilizamos algunas librerías, las más importantes son: Free Chart, bluecove, jdom, itext, jsonrpc, sqlconector, cos.

La herramienta NET BEANS para programar en JAVA con el comando IDE NET BEANS IDE 5.0, para su descarga tenemos la página www.sun.com, aquí desarrollaremos todo el lenguaje en java con la Maquina Virtual JDK 6.0.

Con todos estos elementos desarrollamos la aplicación (BLUETOOTH APLICATION), aquí se realizará todas las funciones, procedimientos, clases, librerías y los paquetes más importantes y levantamos una aplicación WEB (WEB BLUETOOTH), levantamos la aplicación web donde está reflejado las aplicaciones que el usuario tendrá acceso y poder manejar todo es sistema con sus respectivas restricciones.

Utilizamos la herramienta de IREPORT, descargada en la página www.ireport.org, esta herramienta nos ayudara en todo los reportes que el sistema va a presentar a los usuarios.

4.2.1 Clase Aplicación Bluetooth

4.2.1.1. CLASE “Conexión.java”

Realizará la conexión a la base de datos, en este caso, interactúa con la base MySQL, las funciones y procedimientos de la aplicación Bluetooth, que ayudan a las demás clases para lograr un objetivo en común. Es una de las clases mas importante del sistema.

4.2.1.2. CLASE “Dispositivo.java”

Ésta clase se encarga de detectar todos los dispositivos que se encuentre en el área de cobertura del Bluetooth.

Tiene como objetivo llevar un mantenimiento de las direcciones de los dispositivos, esto son Ingresar, Consultar y Eliminar aquellos que son detectados por el sistema Bluetooth.

4.2.2 Aplicación Bluetooth

4.2.2.1 BUSCAR ARCHIVOS . JAVA

Es una de las clases más importante del sistema, ya que ella permite interactuar al sistema con el usuario por medios de peticiones la misma que retornará con una respuesta procesada por el sistema.

4.2.2.2 BUSCAR SERVICIOS.JAVA

Esta clase permite buscar el servicio de envíos de los dispositivos que se han conectados con el sistema, por medio del protocolo de comunicación OBDX. OBDX es un medio de envío de información que trabaja con el puerto 0X1105.

4.2.2.3 DETECTAR DISPOSITIVO.JAVA

Esta clase trabaja como hilo del sistema, la cual tiene la finalidad de realizar un texteo o un escaneo de los dispositivos Bluetooth que se encuentra dentro del perímetro establecido, dentro del área de los 10 metros a la redonda.

4.2.2.4 ENVIA MENSAJES.JAVA

Como su nombre lo dice, ésta clase permite enviar mensajes y responder dichos mensajes ya procesados por el sistema a los usuarios por medio de los dispositivos Bluetooth detectados, en este caso por medio del celular en un formato ya establecido y en forma de texto TXT, actualmente los celulares que poseen estos dispositivos Bluetooth permite hacer estos tipos de envíos.

4.2.2.5 ENVIO PUBLICIDAD.JAVA

Ésta clase permite enviar publicidad por medios de imágenes, tiene el mismo procedimiento que la clase anterior, la única diferencia es que el sistema sólo envía, el usuario sólo lo recibe y no puede reenviar, ya que esta restringida la recepción de imágenes en el servidor.

Estos tipos de imágenes sólo lo puede recibir el sistema con formato tipos GIF Y JPG. Estas imágenes son enviadas a los dispositivos que se encuentren dentro del rango de los 10 metros.

4.2.2.6 MAIN.JAVA

Ésta clase es la inicial del sistema, es la que inicializa los procesos por la cual van hacer interactuar con el ususario por medio de los dispositivos Bluetooth.

Ésta clase es donde arrancamos y está ligado todos los procesos, una vez que corremos esta clase se activan todos los servicios que van a ser utilizados por el sistema.

4.2.2.7 REMOTE DEVICE DISCOVERY. JAVA (DETECTAR DISPOSITIVOS REMOTO)

Esta clase activa los dispositivos Bluetooth que estan detectados dentro del perímetro de la Aplicación bluetooth de una forma de uno a uno, es decir, hace una activación de uno a uno.

4.2.2.8 REMOTE DEVICE DISCOVERY 1.JAVA (DETECTA DISPOSITIVO REMOTO)

Esta clase hace la misma función que la clase anterior, pero con la única diferencia que lo toma de una manera grupal a todos los dispositivos detectado por el sistema y automaticamente manda a escanear los servicios de los dispositivos.

4.2.3 CLASES WEB BLUETOOTH

4.2.3.1 BORRADORES.JAVA

Esta clase permite el mantenimiento, es decir:

- Ingresar
- Consultar
- Modificar

De los datos que se encuentra en la tabla Borrador de la base de datos MySQL.

4.2.3.2 CONEXIÓN.JAVA

Realizará la conexión a la base de datos, en este caso, interactúa con la base MySQL, las funciones y procedimientos de la aplicación Web (Web Bluetooth), que ayudan a las demás clases para lograr un objetivo en común.

4.2.3.3 DISPOSITIVOS.JAVA

Esta clase permite hacer mantenimiento. Ingresar, Consultar, Modificar, Eliminar, los dispositivos a la base de datos MySQL, asigna dispositivos a un usuario registrado al sistema, consulta de dispositivos detectado por el sistema, puede modificar y consultar la ruta a los dispositivos.

4.2.3.4 EXTERNOS.JAVA

Esta clase permite hacer mantenimiento, es decir, Ingresar, Consultar, Modificar y Eliminar de los usuarios externos para presentarlo por pantalla, en la página Web, tiene la función de imprimir, exportar en formato PDF o EXCEL

4.2.3.5 FECHAS.JAVA

Esta clase posee un formato de fecha que se utilizará en todas las consultas entre fechas en: los buzones de entradas y salidas, reportes y en los ingresos de los usuarios.

4.2.3.6 FUNCIONES.JAVA

Esta clase permite validar a los usuarios que ingresa a la página encriptando la clave que fue ingresada por el usuario por medio de un algoritmo MD5 y también guarda los registros para realizar las auditorias del sistema.

4.2.3.7 MENSAJES.JAVA

Esta clase permite realizar una consulta entre fechas a la tabla Mensajes de la base de datos, en la cual me retorna un objeto de tipo Vector que va ser utilizado en la página de la aplicación Web en las consultas de los buzones de entradas, salidas y borradores.

4.2.3.8 PERMISOS.JAVA

Esta clase realiza los mantenimiento, Ingresar, Modificar, Eliminar de la tabla Permiso de la base de Datos.

Una vez que tiene registro, tambien realiza la función de verificar los permisos, por medio del nombre y codigo del permiso.

4.2.3.9 PETICIONES.JAVA

Esta clase realiza los mantenimientos de Ingresar, Modificar y Eliminar de las peticiones en la base de Datos en la tabla Petición.

4.2.3.10 PUBLICIDAD.JAVA

Esta clase realiza los mantenimientos de Ingresar, Modificar y Eliminar de las publicidades en la base de Datos en la tabla Publicidad.

4.2.3.11 USUARIOS.JAVA

Esta clase realiza los mantenimientos de Ingresar, Modificar y Eliminar de los usuarios en la base de Datos en la tabla Usuario.

Con esta clase obtenemos todos los datos del usuario: Usuario, Apellido, numero de celular, y registra su clave.

4.2.4 CLASES DISPOSITIVOS

4.2.4.1 BUSCA SERVICIO.JAVA

Esta clase permite buscar la ruta de envío de los dispositivos Bluetooth que el sistema haya detectado, realiza un escaneo de uno a uno, es decir de dispositivo a dispositivo.

4.2.4.2 BUSCA SERVICIOS.JAVA

Esta clase permite buscar la ruta de envío de los dispositivos Bluetooth que el sistema haya detectado, hace un escaneo de forma grupal, analiza a todos los dispositivos Bluetooth detectados.

4.2.4.3 BUSCAR SERVICIOS.JAVA

Esta clase permite buscar la ruta de envío de los dispositivos Bluetooth que el sistema haya detectado. Aparte de la búsqueda, el servicio hace una Retroalimentación, está en un permanente escaneo a todos los dispositivos Bluetooth, se utiliza una clase Abstracta.

4.2.4.4 ENVIAR MENSAJES

En esta clase nos permite el envío de mensajes a los dispositivos Bluetooth que se encuentren detectados por el sistema, por medio de la aplicación Web, el usuario puede enviar mensajes a todos los dispositivos o en una selección de los mismos.

4.2.4.5 REMOTE DEVICE DISCOVERY.JAVA (DETECTAR DISPOSITIVOS REMOTOS)

Esta clase activa los dispositivos Bluetooth que estan detectados dentro del perimetro de la Aplicación bluetooth de una forma de uno a uno, y de una forma grupal, la cual los dispositivos detectados son colocados en una consola dentro de la aplicación Web.

4.2.5 PAQUETES “COM.METAPARADIGM.JSONRPC”

Este paquete se lo utiliza en toda la página Web, tiene como finalidad realizar un AJAX asincrónica desde la página web hacia las clases .JAVA

4.2.6 PAQUETES “ORG.JFREE.CHART.DEMO.SERVLET”

Este paquete permite realizar gráficos, paint, barras, free chart que son presentadas en la página para que se muestre mas vistoso y muy agradable para los usuarios.

4.2.7 PAQUETES “ORG.JSON”

Este paquete es el que nos permite trabajar en conjunto para realizar la interacción del cliente hacia el servidor por medio de un objeto AJAX.

4.2.8 FUNCIONES Y PROCEDIMIENTOS EN LA BASE DE DATOS

4.2.8.1 FUNCION FUNCIONES

Esta función permite realizar consulta a las base de datos dependiendo de la operación que el usuario desea realizar, retorna un solo dato de tipo varchar, los parámetros que utiliza, una operación de tipo entero y dos valores de tipo varchar.

4.2.8.2 PROCEDIMIENTO CONSULTAS

Este procedimiento permite realizar consulta a las tablas según la operación que el sistema ejecute, retorna un RESULTSET de la consulta realizada, este procedimiento utiliza cuatro parametros.

4.2.8.3 PROCEDIMIENTO CONTADOR MENSAJES

Este procedimiento permite realizar consulta dependiendo el estado de los mensajes y de las fechas ingresadas, lo estados están enumerado por estado: 1 entrada, estado: 2 salida, estado: 3 no envidado. Retorna el numero de registros de la consulta tipo VARCHAR.

4.2.8.4 PROCEDIMIENTO MANTENIMIENTO BORRADORES

Este procedimiento permite realizar Ingreso, Modificación y Eliminación de la tabla borrador en la base de datos, tiene como parametro operación, título, contenido y código. Las operaciones estan enumerada, 1: Ingreso, 2: Modificación, 3: Eliminación.

4.2.8.5 PROCEDIMIENTO MANTENIMIENTO DISPOSITIVOS

Este procedimiento permite realizar Ingreso, Modificación y Eliminación de la tabla Dispositivos en la base de datos, tiene como parámetro: el Nombre, Dirección y Operación, dentro de las operaciones tenemos, 1: Ingreso, 2: Modificación y 3: Eliminación.

4.2.8.6 PROCEDIMIENTO MANTENIMIENTO EXTERNOS

Este procedimiento permite realizar los Ingresos, Modificaciones y Eliminación de la tabla Usuarios Externo de la base de datos, toma como parámetro: Operación, Nombre, Apellido, Celular, Dispositivo, Cédula y el Código, dentro de los Parámetros tenemos, 1: Ingreso, 2: Modificación, 3: Eliminación.

4.2.8.7 PROCEDIMIENTO MANTENIMIENTO PERMISOS

Este procedimiento permite realizar Ingreso, Modificación y Eliminación de la tabla permisos en la base de datos, tiene como operación 1: Ingreso, 2: Modificación y 3: Eliminación.

4.2.8.8 PROCEDIMIENTO MANTENIMIENTO PETICIONES

Este procedimiento permite realizar Ingreso, Modificación y Eliminación de la tabla Peticiones en la base de datos, toma como parámetros: Petición, Argumento, Respuesta, Código y Operación. Tiene como operación 1: Ingreso, 2: Modificación y 3: Eliminación.

4.2.8.9 PROCEDIMIENTO MANTENIMIENTO PUBLICIDAD

Este procedimiento permite realizar Ingreso, Modificación y Eliminación de la tabla Publicidad en la base de datos. Los parámetros que posee son: Operación, Ruta, Estado y Código. Dentro de las operaciones tenemos 1: Ingreso, 2: Modificación y 3: Eliminación.

4.2.8.10 PROCEDIMIENTO MANTENIMIENTO USUARIOS

Este procedimiento permite realizar Ingreso, Modificación y Eliminación de la tabla Usuario en la base de datos, tiene como parámetro Nombre, Apellido, Usuario, Cédula, Clave, Permiso, Código, Operaciones, Correo.

Dentro de las operaciones tenemos 1: Ingreso, 2: Modificación y 3: Eliminación.

4.2.8.11 PROCEDIMIENTO VISTAS

Este procedimiento permite hacer todas las consultas en forma general y ayudará para realizar los reportes, tenemos como parámetro la Operación y el Usuario.

CAPITULO 5

PRUEBAS

5.1. Métodos de prueba del software

En el presente capítulo se detallarán las técnicas y estrategias de pruebas que se le aplicaran al sistema con el fin de garantizar la calidad de nuestro proyecto así como también de descubrir errores que nos permitan mejorar la aplicación que se está desarrollando.

Los objetivos principales de realizar una prueba son:

- Detectar un error
- Tener un buen caso de prueba
- Descubrir un error no descubierto antes.

Principios de la prueba:

- Hacer un seguimiento de las pruebas hasta los requisitos del cliente.
- Plantear y diseñar las pruebas antes de generar ningún código.
- El 80% de todos los errores se centran en sólo el 20% de los módulos.
- Empezar las pruebas en módulos individuales y avanzar hasta probar el sistema entero.

Atributos de una buena prueba

- Más alta probabilidad de encontrar un error.
- No debe ser redundante.
- No debería ser ni demasiado sencilla ni demasiado compleja.

5.2. Técnicas en aplicación de pruebas de software

El diseño de pruebas puede requerir tanto o más esfuerzo que el diseño inicial del producto. Se deben diseñar las pruebas con objeto de intentar capturar el mayor número posible de errores, con la mínima cantidad de esfuerzo y tiempo.

Teniendo en cuenta este objetivo, podemos programar y realizar diferentes clases de pruebas, siendo las más frecuentes en el ámbito de la ingeniería las siguientes:

5.2.1. Técnica de caja negra

Los datos de prueba se escogerán atendiendo a las especificaciones del problema, sin importar los detalles internos del programa, a fin de verificar que el programa corra bien.

Entre los criterios mínimos que guiarán al escoger los datos de prueba, están:

- Valores fáciles: el programa se depurará con datos de fácil comprobabilidad.

- Valores típicos realistas: se ensayará un programa con datos seleccionados para que representen como se aplicará. Los datos han de ser sencillos, de modo que los resultados sean verificables en forma manual.
- Valores extremos
- Valores ilegales: cuando en un programa entra algún valor inadecuado, su salida habrá de ser un mensaje de error adecuado. Es preferible que el programa ofrezca indicación de errores en la entrada y que realice cálculos que sigan siendo factibles luego de desechar la entrada equivocada.

El método de la caja negra se centra en los requisitos fundamentales del software y permite obtener entradas que prueben todos los requisitos funcionales del programa.

Con este equipo de pruebas se intenta encontrar:

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a la base de datos externa.
- Errores de rendimiento.
- Errores de inicialización y terminación.

Con la aplicación de esa técnica se obtiene un conjunto de pruebas que:

- Reduce el número de casos de pruebas y nos dicen algo sobre la presencia o ausencia de errores.

5.2.1.1. Partición equivalente

Una partición equivalente es un método de prueba de caja negra que divide el dominio de entrada de un programa en clases de datos. El diseño de casos de prueba para la partición equivalente se basa en la evaluación de las clases de equivalencia.

Se definen clases de equivalencia, para distintas condiciones de entrada, dirigidas a la definición de casos de prueba que descubran clases de errores. Las clases de equivalencia se pueden definir con arreglo a las siguientes directrices:

- Si una condición de entrada especifica un rango, entonces se definen una clase de equivalencia válida y dos inválidas.
- Si una condición de entrada requiere un valor específico, entonces se definen una clase de equivalencia válida y dos inválidas.

- Si una condición de entrada especifica un miembro de un conjunto, entonces se definen una clase de equivalencia válida y una inválida.

5.2.1.2. Análisis de valores límite

El método de Análisis de valores límite nos lleva a elegir las pruebas que nos ejecuten los valores límite; es decir, se ejecutan los valores de los extremos de un rango permisible, para la verificación de la existencia de algún error.

Además, con este método de prueba de caja negra, se pueden identificar los valores extremos de un campo determinado, en caso de no conocerse, con lo que se puede hacer una validación para no permitir el ingreso de valores fuera de rango, por ejemplo.

5.2.2. Técnica de caja blanca

Se inicia con la observación de que un programa difícilmente puede considerarse como probado por completo si su código contiene partes que nunca han sido ejecutadas.

Este método analiza la estructura lógica del programa y, para cada alternativa que puede presentarse, los datos de prueba ideados conducirán a ella. Se procura escoger los que verifiquen cada posibilidad en las proposiciones case, las cláusulas de cada proposición if y la condición de terminación de cada ciclo.

En un programa extenso, este método es impráctico, sin embargo en un módulo pequeño constituye un excelente medio de prueba y depuración.

5.3. Estrategias de pruebas

El realizar las pruebas tiene, como objetivo principal, verificar que el sistema opere de la forma que se ha requerido, bajo condiciones específicas. Un sistema complejo, puede tener miles e incluso millones de modos operativos potencialmente testeables, es decir, objetos de prueba.

Para el inicio de un adecuado plan de pruebas se recomienda la elección de una de las áreas en donde se implementará el sistema para nuestro caso escogeremos el ingreso de usuarios externos.

Se deberán realizar básicamente pruebas de caja negra, introduciendo valores correctos y esperados, valores límites, valores fuera de límites, entre otras pruebas.

El propósito de implementar todas las estrategias posibles es garantizar que en el uso inicial del Sistema este se encuentre libre de problemas lo cual se puede descubrir durante este proceso y llevar a cabo las correcciones de lugar para su buen funcionamiento.

La prueba de caja negra nos permite asegurar la confiabilidad de los datos que se comunican a través de los componentes de la aplicación. En este sentido, se ha comprobado que los valores correspondientes a los ingresos de datos esenciales sean congruentes.

Los nombres de los alumnos, profesores y demás que sean elementos textuales similares no contienen valores numéricos ni caracteres especiales.

Los valores numéricos no aceptan letras ni caracteres especiales pero si valores numéricos en rangos esperados. Las listas desplegables de selección sólo permiten elegir valores entre los datos mostrados.

5.4. Pruebas realizadas al sistema

A continuación se presentan tres de los casos de pruebas realizados al sistema:

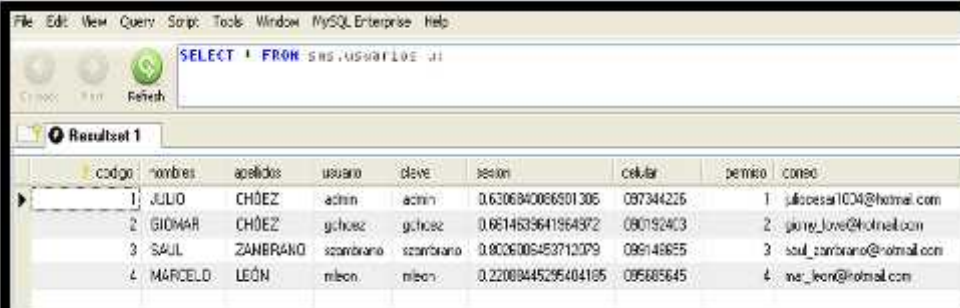
5.4.1. Prueba de ingreso al sistema

En la pantalla de identificación de usuarios se validaron aquellos que están registrados. Y aquellos que no lo están apareció un mensaje de error. Como podemos apreciar en la siguiente figura en donde ingresamos el usuario lpazmiño y su clave.



Figura 5.1 Identificación de usuario

Pero como podemos observar en la base de datos adjunta dicho usuario no se encuentra registrado.



	id	codigo	nombres	apellidos	usuario	clave	sesion	celular	correo	correo
1	1	JULIO	CHÓEZ	admin	admin	0.6306840066501305	087344226	1	juliocezar1004@hotmail.com	
2	2	GIOMAR	CHÓEZ	gchoez	gchoez	0.6614639641864392	080192403	2	giomaj_love@hotmail.com	
3	3	SAUL	ZAMBRANO	szambrano	szambrano	0.8006005453712079	095146625	3	saul_zambrano@hotmail.com	
4	4	MARCELO	LEÓN	mleon	mleon	0.22083445295404105	095685645	4	maz_leon@hotmail.com	

Figura 5.2 Tabla de usuarios del sistema

Es por esto que aparece el siguiente mensaje de error en el que indica que no existe.



Figura 5.3 Error de usuario

Entonces queda demostrado que si realiza la validación de usuarios del sistema.

5.4.2 Prueba de envío de mensajes de texto a los dispositivos

5.4.2.1 Pruebas del sistema

Como podemos apreciar ésta pantalla tiene tres botones, en el caso de no escribir ningún mensaje o no seleccione ningún borrador al hacer click en enviar no le permitirá realizar la acción y le presentará el siguiente mensaje:



Figura 5.4 Error al no escribir un mensaje

Escribimos el mensaje o seleccionamos de la lista desplegable la opción ofertas laborales como se muestra en la figura 5.5. Una vez seleccionado escribimos el resto de información.



Figura 5.5 Selección de la lista de borradores

Haciendo click en el botón actualizar aparecen los dispositivos detectados, para esta prueba son cuatro los dispositivos.



Figura 5.6 Dispositivos detectados

Seleccionamos todos los dispositivos detectados por el sistema como se aprecia en la figura 5.7.



Figura 5.7 Dispositivos detectados por el sistema

Al hacer click en enviar, el sistema nos presenta un mensaje de espera mientras se envían los mensajes a todos los dispositivos que hemos seleccionado.



Figura 5.8 Mensaje de espera

Una vez que ha finalizado el envío podemos observar que en la consola aparece el detalle de los envíos por lo tanto el programa sí nos indica que se efectúan los envíos de los mensajes a los dispositivos (figura 5.9).



Figura 5.9 Información en la Consola

5.4.2.2 Pruebas en los dispositivos

Una vez que se hace click en enviar, el sistema comienza a enviar a los dispositivos conectados un mensaje, esto lo realiza uno por uno. Los dispositivos reciben una solicitud como se aprecia en la figura 5.10 en el que indica si desea recibir mensaje del servidor.



Figura 5.10 Consulta al dispositivo

Cuando el usuario acepta recibir el mensaje, en el celular dependiendo de la configuración que tengan le llegará en forma de mensaje o en forma de nota como se muestra en la figura 5.11.



Figura 5.11 Recepción de mensajes en los dispositivos

Y al final en cada dispositivo se puede visualizar el mensaje enviado (figura 5.12), comprobando de esta manera que el sistema si realiza el envío de mensajes a los teléfonos celulares.



Figura 5.12 Visualización del mensaje en los dispositivos

5.4.3 Ingreso de usuarios del sistema

Al ingresar usuarios del sistema, Servidor Bluetooth no permitirá ingresar números en aquellos que han sido definidos como texto, ni texto en aquellos en los que sólo son números como se muestran en las figuras 5.13 y 5.14.



Figura 5.13 Ingreso erróneo en un campo de sólo letras



Figura 5.14 Ingreso erróneo en un campo de sólo números

Valida que el número celular no esté registrado en la base de datos, caso contrario nos presenta el siguiente mensaje:



Figura 5.15 Validación de cédula del usuario

En este caso nos presenta error porque en la base de datos ya existe el número celular registrado en otro usuario, como podemos observar en la siguiente figura:



Usuario	Nombres	Apellidos	Celular	Privilegio
admin	JULIO CESAR	CHÓEZ CATUTO	097544226	ADMINISTRADOR
gchoez	GIOMAR PAOLA	CHÓEZ CATUTO	090192403	OPERADOR
szambrano	WASHINGTON SAIUL	ZAMBRANO RIVAS	099148655	CONSULTOR
mleon	MARCELO JAVIER	LEÓN GONZALEZ	095685645	AUDITOR

Figura 5.16 Consulta del número de cédula

Valida también que el correo electrónico esté compuesto por @ y el punto. Caso contrario nos muestra el siguiente mensaje:



Nombre: JULIO CARLOS
 Apellido: CARDENAS VALDEZ
 Celular: 094921718
 Correo: Juan@hotmail
 Usuario: juanc
 Clave: *****
 Confirmar Clave: *****
 Privilegio: ADMINISTRADOR

Windows Internet Explorer
 Por favor verifique el correo del usuario
 Aceptar

Figura 5.17 Validación del correo electrónico

Además cuando se confirma la clave ésta debe ser la misma que se digitó anteriormente caso contrario aparece el siguiente mensaje:



Figura 5.18 Claves no coinciden

Una vez que se validen todos los campos podrá guardar el nuevo registro y aparecerá un mensaje en el que preguntará al usuario si desea ingresar los nuevos datos.



Figura 5.19 Confirmación para almacenar los datos

Al aceptar se ingresará el registro a la base de datos correctamente como lo indica en la figura 5.20.

Bienvenido : JULIO CESAR CHÓEZ CATUTO

Principal Usuarios Privilegios Envíos Reportes 12 Peticiones

INGRESO DE USUARIOS DEL SISTEMA

Nombres : JUAN CARLOS
 Apellidos : CARDENAS VALDEZ
 Celular : 094921718
 Correo : juan@hotmail.com
 Usuario : juanc
 Clave : *****
 Confirmar Clave : *****
 Privilegio : CONSULTOR

Guardar Limpiar

Windows Internet Explorer
Usuario fue ingresado con éxito...
Aceptar

Figura 5.20 Ingreso del nuevo usuario

Podemos comprobarlo al hacer una consulta de usuarios, de esta manera demostramos que el sistema realiza el ingreso con éxito.

Bienvenido : JULIO CESAR CHÓEZ CATUTO

Principal Usuarios Privilegios Envíos Reportes 12 Peticiones

CONSULTA DE USUARIOS DEL SISTEMA

Reporte en : Excel Modificar Eliminar Imprimir Exportar

Usuario	Nombres	Apellidos	Celular	Privilegio
<input type="radio"/> admin	JULIO CESAR	CHÓEZ CATUTO	097344226	ADMINISTRADOR
<input type="radio"/> gchoez	GIOMAR PAOLA	CHÓEZ CATUTO	090192403	OPERADOR
<input type="radio"/> szambrano	WASHINGTON SAUL	ZAMBRANO RIVAS	099148655	CONSULTOR
<input type="radio"/> mleon	MARCELO JAVIER	LEON GONZALEZ	005685635	AUDITOR
<input type="radio"/> juanc	JUAN CARLOS	CARDENAS VALDEZ	094921718	CONSULTOR

Figura 5.21 Consulta de usuarios del sistema

CAPITULO 6

CONCLUSIONES Y RECOMENDACIONES

6.1. Conclusiones

Este programa tiene el fin de facilitar a los estudiantes de la carrera de Ingeniería de Sistemas Computacionales conocer sus notas al enviar un mensaje desde su teléfono celular con tecnología bluetooth al Servidor Bluetooth.

El sistema interactúa con el usuario a través del dispositivo bluetooth enviando respuesta a la petición que el usuario requiera, porque además de sus notas, el estudiante podrá conocer información adicional como son cursos, seminarios, novedades que existen en la facultad, eventos deportivos o sociales, etc.

Además podrá recibir imágenes publicitarias de ofertas de equipos de computación, cursos CISCO, logos, etc.

El beneficio del sistema es grande, porque brinda a los estudiantes verificar la información con respecto a sus notas de manera ágil y rápida, utilizando un dispositivo con tecnología bluetooth, evitando de esta manera seguir con el actual proceso de consulta que resulta tedioso, en el que la información se encuentra disponible en un sistema de la carrera que sólo puede ser solicitado en ventanilla mediante una solicitud. Con este sistema se pretende evitar estos problemas y estar a la vanguardia de la tecnología.

Tomamos como base la información recabada por la compañía Bluepublicidad Cía. Ltda. en la que su estudio de mercado a 100 personas indicó que 7 de cada 10 entrevistados poseen celulares con Bluetooth, por lo cual nuestro proyecto tiene mucha expectativa al utilizar esta tecnología.

La tecnología Bluetooth es muy difundida en el área de los teléfonos celulares, cada vez más modelos la incluyen por lo que el campo de aplicación del proyecto sigue creciendo.

Nuestro sistema cumplió con los objetivos esperados por tal motivo concluimos que el Servidor Bluetooth es una buena alternativa de consulta y marketing publicitario el cual puede ser utilizado en diferentes campos no solo el educativo.

6.2. Recomendaciones

Nuestro proyecto tiene mejor funcionamiento en ambiente Windows porque ésta plataforma trabaja muy bien con la tecnología bluetooth a diferencia de Linux en la que no existe soporte para este tipo de tecnología pero en el caso que se la utilice sólo habría que configurar el servicio de comunicación bluetooth.

Se recomienda el uso del lenguaje Java, para la creación de este proyecto, por la versatilidad y la gran cantidad de equipos que lo soportan además porque la Máquina Virtual Java viene incluida en los teléfonos móviles desde su fabricación. Ésta debe contar con soporte para Bluetooth (JSR-82) para que no exista problema alguno en la recepción de los mensajes e imágenes de publicidad.

Se debe trabajar con la máquina virtual de java JDK 6.0 porque tiene herramientas muy esenciales para poder trabajar con la tecnología bluetooth como es el caso del jar bluecove el cual permite la comunicación entre dispositivos bluetooth.

Este proyecto puede ser referencia y guía para quienes deseen continuar desarrollando este tipo de aplicaciones. Por ejemplo, se recomienda mejorar este sistema para que a futuro se pueda realizar la matriculación de estudiantes, enviar videos, u otros documentos.

Se recomienda instalar el servidor en un lugar central, para que pueda cubrir un área lo más útil posible. Aunque la interferencia no afecta en la comunicación Bluetooth, se recomienda en lo posible instalar el servidor alejado de otros aparatos que trabajen en la banda de los 2.4GHz.

El Sistema Operativo en cada marca de teléfono celular puede ser diferente por lo que se recomienda los celulares Nokia, ya que en las pruebas que se realizaron, los resultados fueron los esperados. No tanto así en los blackberry cuyo sistema es un poco complejo.

A futuro se recomienda la instalación del bluetooth 3.0 que es el nuevo estándar inalámbrico cuya velocidad de transferencia de archivos entre dispositivos, será de 24 MB por segundo a diferencia de la versión 2.0 que solo llega a 3 MB por segundo.

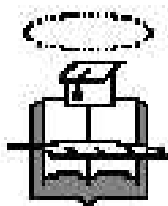
Se recomienda además que las imágenes de publicidad sean con extensión .jpg o .gif. para lo cual fue programado el sistema.

6.3. Bibliografía

Para la elaboración de nuestro proyecto nos basamos en información recogida de diferentes sitios web de interés.

WEBSITES:

- <http://www.expreso.ec/ediciones/2009/05/11/guayaquil/andres-usa-el-bluetooth-para-vender-productos/Default.asp>
- www.laneros.com/attachment.php?attachmentid=70414
- <http://www.tuexperto.com/2009/04/14/bluetooth-30-el-nuevo-estandar-inalambrico-podria-estar-listo-este-mes/>
- <http://internetng.dit.upm.es/2009/05/05/bluetooth-30-hs-un-nuevo-bluetooth-mucho-mas-rapido/>
- <http://es.wikipedia.org/wiki/Bluetooth>
- <http://www.osmosislatina.com/lenguajes/uml/casos.htm>
- <http://tvdι.det.uvigo.es/~avilas/UML/node43.html>
- http://es.wikipedia.org/wiki/Diagrama_de_secuencia
- <http://www.monografias.com/trabajos39/proyecto-informatico/proyecto-informatico2.shtml>
- <http://es.wikipedia.org/wiki/Bluejacking>
- <http://es.sun.com/>
- <http://www.todomercado.com/Internet-y-Comunicaciones/Marketing-de-proximidad-via-Bluetooth-Proximity-Marketing.view?id=1215292361700>
- <http://www.blue-tooth.es/>
- <http://java.sun.com/javame/technology/index.jsp>
- <http://bluetooth.com>
- <http://spanish.bluetooth.com>



UNIVERSIDAD DE GUAYAQUIL
Facultad de Ciencias Matemáticas y Físicas
Carrera de Ingeniería en Sistemas Computacionales

“Servidor SMS”

TESIS DE GRADO

Previo la Obtención del Título de:

**INGENIERO EN SISTEMAS
COMPUTACIONALES**

Autores:

GIOMAR PAOLA CHÓEZ CATUTO
JULIO CÉSAR CHÓEZ CATUTO
WASHINGTON SAÚL ZAMBRANO RIVAS

GUAYAQUIL – ECUADOR
Año: 2009

AGRADECIMIENTO

Agradezco a Dios por darme la oportunidad de cumplir uno de mis sueños, el graduarme como Ingeniera en Sistemas Computacionales, y por ser quién ha estado a mi lado en todo momento dándome las fuerzas necesarias para continuar luchando día tras día y seguir adelante.

A mis tutores Ing. Mario Robayo e Ing. Eduardo Alvarado por sus sugerencias, orientación y apoyo para la realización de este proyecto.

A mis maestros por compartir sus conocimientos que con mucho esmero han brindado en toda mi carrera y por incentivarnos a seguir adelante.

Y a todas aquellas personas que de una u otra forma colaboraron en la realización de este proyecto, hago extenso mi más sincero agradecimiento.

Giomar Chóez C.

AGRADECIMIENTO

A la carrera por sus avances tecnológicos y por su prestigio, formando profesionales capacitados para un medio muy competitivo como es el área de sistemas.

A los tutores y profesores del seminario, el Ing. Mario Robayo y el Ing. Eduardo Alvarado que con sus experiencias y conocimientos hicieron posible dar paso a la terminación del proyecto de grado.

Al coordinador del paralelo el Ing. Abel Alarcón que por sus reuniones indicaba pautas del seminario y siempre presto al dialogo y colaboración.

A mis compañeros y amigos, que por medio de las discusiones y preguntas, me hacen crecer en conocimiento.

Julio Chóez C.

AGRADECIMIENTO

A Dios ante todo, a mi Madre que siempre está aquí a mi lado, mis hermanos.

A la institución educativa, Carrera de Ingeniería de Sistemas Computaciones, de la universidad de Guayaquil que me brindo la oportunidad para desarrollarme como un profesional y como una persona de bien.

A todos mis tíos que me dieron la oportunidad de estar aquí en esta bella ciudad, que no lo voy a defraudar.

A los catedráticos de la carrera que siempre están ahí enseñando y dando a conocer sus experiencias en el campo laboral.

A mi Tío Político que él no vive en nuestro país, que siempre estuvo dando su apoyo, pendiente de los pasos que estaba dando en los estudios.

Saúl Zambrano R.

DEDICATORIA

El esfuerzo y dedicación que he puesto en este proyecto va con mucho cariño a mis padres Sr. Pablo Chóez y Sra. Vicenta Catuto por todo lo que me han dado en esta vida, por todo lo que soy como persona gracias a sus nobles consejos y por estar a mi lado en los momentos difíciles, apoyándome en todas las decisiones que he tomado en la vida.

A mi hermano Julio César quien me ha acompañado durante todo este trayecto con paciencia y optimismo.

A mi hija Milena Paola por ser la fuente de mi inspiración para superarme cada día más y así poder luchar para que la vida nos depare un futuro mejor.

Giomar Chóez C.

DEDICATORIA

No hay palabras para describir mi profundo agradecimiento a mis Padres, por todos sus consejos y ayuda comprendiendo que nada se consigue de la noche a la mañana, que la perseverancia y el esfuerzo son el camino para lograr objetivos.

A mi hermana quien es parte del grupo de seminario y también como compañera de la universidad con su ayuda y apoyo.

A mi sobrina que es parte de mi vida y una de las razones de mi inspiración y deseos de superación.

Este seminario es parte de un nuevo comienzo por ello la dedico a Dios por darme la oportunidad de llegar a estas instancias con base de esfuerzo y sacrificio.

Julio Chóez C.

DEDICATORIA

Le dedico este trabajo especial de grado, ante todo, a Dios, el cual me han acompañado a lo largo de toda mi vida, dándome salud y fortaleza.

A mis padres, especialmente a mi Madre que siempre ha estado aquí a mi lado en todo momento, a mis Abuelitas que ahora no están a mi lado dedico este trabajo. A mi hermano y mi hermana porque siempre vieron un gran potencial en mí y siempre me lo hicieron saber.

A las personas que siempre han confiado en mí, a los verdaderos amigos que siempre te dan la mano. Y por último, a la persona que siempre esta ayudándome moralmente, que ahora forma parte de esta lucha, para ti con mucho amor
Ingrid

Saúl Zambrano R.

TRIBUNAL DE GRADUACIÓN

PRESIDENTE DEL TRIBUNAL

PRIMER VOCAL

SEGUNDO VOCAL

SECRETARIO

DECLARACIÓN EXPRESA

“La autoría de la tesis de grado corresponde exclusivamente a los suscritos,
Perteneciendo a la Universidad de Guayaquil los derechos que generen la
Aplicación de la misma”

Según el Reglamento de Graduación de la Carrera de Ingeniería en Sistemas
Computacionales, Art. 26.

Giomar Paola Chóez Catuto
C.I. # 091959834-2

Julio César Chóez Catuto
C.I. # 091959718-7

Washington Saúl Zambrano Rivas
C.I. # 131000275-1

RESUMEN

El mundo va cambiando cada día más y por ende las necesidades de las personas, de acuerdo a las nuevas tecnologías que cada vez van avanzando con el pasar del tiempo, el **BLUETOOTH** no solo se adapta a los cambios sino que surge de las necesidades propias del ser humano.

Servidor Bluetooth ofrecerá a los estudiantes de la Carrera de Ingeniería de Sistemas Computacionales acceder a información personal, recibir avisos de manera eficaz, además de recibir publicidad de manera rápida con tan solo tener un teléfono celular con bluetooth.

Servidor Bluetooth se convierte en una herramienta útil en cualquier tipo de mercado, debido a que la mayoría de las personas poseen un teléfono celular con bluetooth que ha dejado de ser un lujo para ser una herramienta necesaria de comunicación, este mercado potencial hará que el Servidor Bluetooth se convierta en el boom que hará que las instituciones que lo

adquieran aumenten sus ingresos, adquieran más estudiantes y proporcionen confianza, seguridad y veracidad en tiempo real.

Nuestra compañía contará con:

- Líder de proyecto.
- Soporte usuario.
- Programador conocimientos base datos

Para realizar este gran proyecto se requiere una inversión inicial de \$770 dólares, los cuales serán aportados por los integrantes del proyecto.

Este proyecto no tendrá ningún costo, será un beneficio tanto para los estudiantes como para la carrera, puesto que va a satisfacer las necesidades del estudiante de conocer sus notas y la Carrera de Ingeniería de Sistemas innovará un producto utilizando una de las tecnologías que existen en la actualidad tratando de esta manera mejorar sus servicios a nivel académico dando una nueva imagen y ganando un prestigio cada vez más notorio a nivel de universidades.

ÍNDICE

ÍNDICE	PÁGINAS
AGRADECIMIENTOS.....	II – IV
DEDICATORIAS	V – VII
TRIBUNAL DE GRADUACIÓN	VIII
DECLARACIÓN EXPRESA	IX
RESUMEN	X

CAPITULO 1

MANUAL DE USUARIO

1. Interfaz de la aplicación	3
2. Página de inicio	6
3. Ingreso a usuarios	9
3.1. Sistema	9
3.1.1.Ingresar	10
3.1.2.Consultar	19
3.2. Externos	26
3.2.1.Ingresar	26
3.2.2.Consultar	34
4. Ingreso a privilegios	42
4.1. Ingresar	42
4.2. Consultar	46
5. Ingreso a envíos	52
5.1. Mensajes	52
5.2. Publicidad	58
6. Ingreso a Reportes	68

6.1. Dispositivos encontrados	69
6.2. Buzones	72
6.2.1.Entrada	73
6.2.2.Salida	78
6.2.3.No enviados	83
6.2.4.Borradores	87
6.2.5.Contador	95
6.3. Auditoría	97
6.4. Más usados	100
7. Ingreso a peticiones	102
7.1. Ingresar	102
7.2. Consultar	106

CAPITULO 2

MANUAL TÉCNICO

1. Clases	122
1.1. CLASE “Conexión.java”	122
1.2. CLASE “dispositivos.java”	128
2. Application_bluetooth	136
2.1. CLASE “Buscar_Archivos.java”	136
2.2. CLASE “Buscar_Servicios.java”	144
2.3. CLASE “Detectar_Dispositivo.java”	147
2.4. CLASE “Envia_Mensaje.java”	150
2.5. CLASE “Envio_publicidad.java”	154
2.6. CLASE “Main.java”	158
2.7. CLASE “RemoteDeviceDiscovery.java”	162
2.8. CLASE “RemoteDeviceDiscovery_1.java”	165
2.9. CLASE “ServicesSearch.java”	168
3. Web_bluetooth.....	171

3.1.	CLASE "Borradores.java"	171
3.2.	CLASE "Conexion.java"	174
3.3.	CLASE "Dispositivos.java"	178
3.4.	CLASE "externos.java"	187
3.5.	CLASE "fecha.java"	191
3.6.	CLASE "funciones.java"	194
3.7.	CLASE "mensajes.java"	197
3.8.	CLASE "permisos.java"	199
3.9.	CLASE "peticiones.java"	205
3.10.	CLASE "publicidad.java"	207
3.11.	CLASE "usuarios.java"	214
4.	Dispositivos.....	219
4.1.	CLASE "Busca_Servicio.java"	219
4.2.	CLASE "Busca_Servicios.java"	222
4.3.	CLASE "Buscar_Servicios.java"	223
4.4.	CLASE "Envia_Mensaje.java"	226
4.5.	CLASE "RemoteDeviceDiscovery.java"	229
4.6.	CLASE "ServicesSearch.java"	232
5.	Clase com.metaparadigm.jsonrpc.....	235
5.1.	CLASE "JSONRPC_CALLBACK.java"	235
5.2.	CLASE "JSONRPCServlet.java"	236
5.3.	CLASE org.jfree.chart.demo.servlet.....	241
5.4.	CLASE "WebHit.java"	242
5.5.	CLASE "WebHitChart.java"	245
5.6.	CLASE org.json.....	256
5.6.1.	CLASE "JSONArray.java" ARREGLO.....	256
5.7.	CLASE "JSONObject.java" STRING O UN OBJETO.....	268
5.8.	CLASE "JSONTokener.java" ESPECIE MATRIZ.....	284
6.	Páginas.....	291
6.1.	Página"index.jsp".....	291

6.2. Página"principal.jsp"	294
6.3. Página"abrir_reporte.JSP"	295
6.4. Página"acceder.jsp"	297
6.5. Página"bandejas.jsp"	298
6.6. Página"borradores.jsp"	301
6.7. Página"cabecera.jsp"	303
6.8. Página"centro.jsp"	304
6.9. Página"con_privilegios.JSP"	306
6.10. Página"con_programacion.JSP"	308
6.11. Página"con_usuarios.JSP"	310
6.12. Página"consulta_mensajes.JSP"	312
6.13. Página"contador.JSP"	313
6.14. Página"entrada.JSP"	316
6.15. Página"env_dispositivos.JSP"	318
6.16. Página"env_publicidad.JSP"	321
6.17. Página"ing_borradores.JSP"	323
6.18. Página"ing_privilegios.JSP"	325
6.19. Página"ing_programacion.JSP"	330
6.20. Página"ing_usuarios.JSP"	332
6.21. Página"ing_usuarios_externos.JSP"	334
6.22. Página"menu.JSP"	336
6.23. Página"mod_borradores.JSP"	339
6.24. Página"mod_externos.JSP"	341
6.25. Página"mod_privilegios.JSP"	343
6.26. Página"mod_publicidad.JSP"	349
6.27. Página"no_enviados.JSP"	350
6.28. Página"nueva_publicidad.JSP"	352
6.29. Página"rep_acceso.JSP"	354
6.30. Página"rep_conectados.JSP"	356
6.31. Página"rep_externos.JSP"	358

6.32. Página"rep_publicidad.JSP"	360
6.33. Página"rep_usados.JSP"	361
6.34. Página"salida.JSP"	363
7. Funciones	365
8. Procedimientos almacenados my_sql.....	367
8.1. Procedimiento almacenado "consultas".....	367
8.2. Procedimiento almacenado "contador mensajes"	372
8.3. Procedimiento almacenado "mantenimiento_borradores".	373
8.4. Procedimiento almacenado "mantenimiento_dispositivos"	373
8.5. Procedimiento almacenado "mantenimiento_externos"...	375
8.6. Procedimiento almacenado "mantenimiento_peticiones"..	376
8.7. Procedimiento almacenado "mantenimiento_publicidad"..	377
8.8. Procedimiento almacenado "mantenimiento_usuarios"....	378
8.9. Procedimiento almacenado "vista_mensajes"	379
8.10. Procedimiento almacenado "vista"	380

FIGURAS

ÍNDICE

PÁGINAS

MANUAL DE USUARIO

Fig. 2.1 Ingreso al sistema	6
Fig. 2.2 Página principal.....	8
Fig. 3.1 Menú usuario.....	9
Fig. 3.2 Submenú sistema.....	9
Fig. 3.3 Página de ingreso de usuarios del sistema.....	10
Fig. 3.4 Consulta de usuarios del sistema.....	19
Fig. 3.5 Modificación de datos usuarios del sistema.....	20
Fig. 3.6 Ventana de configuración de la impresora.....	24
Fig. 3.7 Impresión de usuarios del sistema.....	24
Fig. 3.8 Exportación de usuarios del sistema a PDF.....	25
Fig. 3.9 Exportación de usuarios del sistema a EXCEL.....	25
Fig. 3.10 Submenú externos.....	26
Fig. 3.11 Página de ingreso de usuarios externos.....	26
Fig. 3.12 Consulta de usuarios externos.....	35
Fig. 3.13 Modificación de usuarios externos.....	36
Fig. 3.14 Impresión de usuarios externos.....	40
Fig. 3.15 Exportación de usuarios externos a PDF.....	41
Fig. 3.16 Exportación de usuarios externos a EXCEL.....	41
Fig. 4.1 Menú privilegios.....	42
Fig. 4.2 Página de ingreso de privilegios.....	43
Fig. 4.3 Selección de un menú en privilegios.....	44
Fig. 4.4 Consulta de privilegios de usuarios.....	47
Fig. 4.5 Modificación de privilegios.....	48
Fig. 4.6 Impresión de privilegios.....	51

Fig. 5.1 Menú envíos	53
Fig. 5.2 Página envío de mensajes.....	53
Fig. 5.3 Selección de un borrador.....	54
Fig. 5.4 Presentación de la consola.....	55
Fig. 5.5 Página consulta de envío de publicidad.....	58
Fig. 5.6 Ingreso de publicidad.....	59
Fig. 5.7 Reporte de una publicidad.....	65
Fig. 5.8 Impresión del reporte de publicidad.....	66
Fig. 5.9 Modificación de una publicidad.....	67
Fig. 6.1 Menú Reportes.....	69
Fig. 6.2 Página envío de mensajes.....	69
Fig. 6.3 Exportación Reporte de Dispositivos a PDF.....	70
Fig. 6.4 Exportación Reporte de Dispositivos a EXCEL.....	71
Fig. 6.5 Impresión de Dispositivos detectados.....	72
Fig. 6.6 Submenú Buzones.....	72
Fig. 6.7 Buzón de entrada.....	73
Fig. 6.8 Exportación Buzón de entrada a PDF.....	74
Fig. 6.9 Exportación Buzón de entrada a EXCEL.....	75
Fig. 6.10 Impresión de buzón de entrada.....	76
Fig. 6.11 Búsqueda semanal buzón de entrada.....	77
Fig. 6.12 Buzón de salida.....	78
Fig. 6.13 Exportación buzón de salida a PDF.....	79
Fig. 6.14 Exportación buzón de salida a EXCEL.....	80
Fig. 6.15 Impresión de buzón de salida.....	81
Fig. 6.16 Búsqueda semanal buzón de salida.....	82
Fig. 6.17 Buzón de No enviados.....	83
Fig. 6.18 Exportación Buzón de no enviados a PDF.....	84
Fig. 6.19 Exportación Buzón de no enviados a EXCEL.....	85
Fig. 6.20 Impresión de buzón de no enviados.....	86

Fig. 6.21 Búsqueda semanal no enviados.....	87
Fig. 6.22 Borradores.....	87
Fig. 6.23 Página de ingreso de borrador.....	88
Fig. 6.24 Modificación de borradores.....	92
Fig. 6.25 Impresión de borradores.....	95
Fig. 6.26 Contador de mensajes.....	95
Fig. 6.27 Impresión de contador de mensajes.....	97
Fig. 6.28 Página de Reporte de auditoría.....	97
Fig. 6.29 Exportación Reporte de auditoría a PDF.....	98
Fig. 6.30 Exportación de Reporte de auditoría a EXCEL.....	99
Fig. 6.31 Impresión del Reporte de Auditoría.....	100
Fig. 6.32 Página de los dispositivos más usados.....	100
Fig. 6.33 Impresión de los dispositivos más usados.....	101
Fig. 7.1 Menú Peticiones.....	102
Fig. 7.2 Página de ingreso de peticiones.....	103
Fig. 7.3 Página de consulta de peticiones.....	107
Fig. 7.4 Modificación de una petición.....	108
Fig. 7.5 Impresión de peticiones.....	111
Fig. 7.6 Exportación de Reporte de peticiones a PDF.....	112
Fig. 7.7 Exportación de Reporte de peticiones a EXCEL.....	112

MANUAL TÉCNICO

Fig. 6.1 Acceso al sistema.....	291
Fig. 6.2 Página principal.....	294
Fig. 6.3 Reporte buzón de entrada.....	296
Fig. 6.4 Ventana de error de validación de usuario.....	297
Fig. 6.5 Bandeja de entrada.....	298
Fig. 6.6 Borradores.....	301
Fig. 6.7 Cabecera del sistema.....	303
Fig. 6.8 Centro del sistema.....	305

Fig. 6.9 Privilegios.....	306
Fig. 6.10 Peticiones.....	308
Fig. 6.11 Consulta de usuarios del sistema.....	310
Fig. 6.12 Consulta de mensajes.....	312
Fig. 6.13 Contador de mensajes.....	313
Fig. 6.14 Buzón de entrada.....	316
Fig. 6.15 Envío de mensajes.....	319
Fig. 6.16 Consulta de envío de publicidad.....	321
Fig. 6.17 Ingreso de borradores.....	323
Fig. 6.18 Ingreso de privilegios.....	326
Fig. 6.19 Ingreso de peticiones.....	331
Fig. 6.20 Ingreso de usuarios del sistema.....	332
Fig. 6.21 Ingreso de usuarios externos.....	334
Fig. 6.22 Menú.....	336
Fig. 6.23 Modificación de borradores.....	339
Fig. 6.24 Modificación de usuarios externos.....	341
Fig. 6.25 Modificación de privilegios.....	343
Fig. 6.26 Modificación de publicidad.....	349
Fig. 6.27 Buzón de mensajes no enviados.....	350
Fig. 6.28 Ingreso de publicidad.....	353
Fig. 6.29 Auditoría.....	355
Fig. 6.30 Conectados.....	356
Fig. 6.31 Reporte de usuarios externos.....	358
Fig. 6.32 Reporte de publicidad.....	360
Fig. 6.33 Reporte de los más usados.....	361
Fig. 6.34 Buzón de salida.....	363



CAPÍTULO 1

MANUAL DE USUARIO

SERVIDOR BLUETOOTH

Este sistema tiene como fin el envío de mensajes de texto e imágenes publicitarias a los estudiantes y profesores de la Carrera de Ingeniería en Sistemas Computacionales, desde el servidor bluetooth hacia los celulares con bluetooth activado y dentro del campo de acción.

En este proyecto nos vamos a enfocar en el envío de mensajes de texto, imágenes publicitarias, así como también la facilidad que le ofrece al estudiante para consultar sus notas enviando desde su teléfono celular una petición y recibiendo respuesta del servidor bluetooth.

El objetivo principal del proyecto es brindar un servicio a través del cual la Carrera de Ingeniería de Sistemas Computacionales esté en contacto con sus estudiantes y profesores de manera rápida y directa a través de un dispositivo móvil con Bluetooth.

La definición del proyecto es el envío de mensajes de textos, imágenes publicitarias y responder a las solicitudes de los estudiantes a través de un celular con bluetooth, administrar la información de los usuarios así como también facilitar el reporte del uso de la aplicación.

Una de las características que contiene el proyecto es que puede ser utilizada por usuarios diferentes cada uno con sus restricciones de acceso dependiendo de su cargo.

Los Cargos son los siguientes:

- Administrador.- Persona que podrá tener acceso a todo el sistema.
- Operador.- Persona que tendrá acceso a ciertos privilegios del sistema que se detallan más adelante de acuerdo a los permisos otorgados por el administrador.
- Consultor- Persona que podrá tener acceso a todo lo referente con consulta y reportes menos el Reporte de Auditoría.
- Auditor.- Persona que podrá tener acceso a todo lo referente a consulta y reporte pero en especial al Reporte de Auditoría.

1. INTERFAZ DE LA APLICACIÓN

Todas las páginas están orientadas a sus diversas actividades y de detallarán cada uno de los pasos a seguir para que los usuarios del sistema Servidor Bluetooth puedan tener un buen manejo de la aplicación.

El Administrador del Servidor Bluetooth podrá:

- Ingresar nuevos usuarios del sistema y usuarios externos.
- Consultar y modificar usuarios del sistema y externos.
- Ingresar nuevos privilegios.
- Consultar, modificar y eliminar los privilegios, menos el de administrador.
- Enviar mensajes de textos.
- Enviar imágenes de publicidad.
- Crear registros de las publicidades a enviar, definiendo los días en que se enviarán las imágenes.
- Modificar o eliminar una publicidad.
- Ver todos los reportes del sistema
- Ingresar peticiones, así como también su respectiva consulta, eliminación.
- Podrá imprimir los reportes y las consultas que se realice en el sistema.

El Usuario del sistema podrá:

- Dependiendo de los permisos otorgados por el administrador, tener acceso a los diferentes menús del sistema.
- No tendrá acceso a los usuarios del sistema.
- No tendrá acceso a los privilegios.

El Consultor podrá:

- Tener acceso a la consulta de los usuarios del sistema y externos.
- Tener acceso a la consulta de los privilegios.
- Tener acceso a los reportes del sistema, menos el reporte de auditoría.
- Tener acceso a la consulta de las peticiones.
- Impresión de reportes.

El Auditor podrá:

- Tener acceso a la consulta de los usuarios del sistema y externos.
- Tener acceso a la consulta de los privilegios.
- Tener acceso a los reportes del sistema.
- Tener acceso al reporte de auditoría.
- Tener acceso a la consulta de las peticiones.
- Impresión de reportes.

Antes de empezar a utilizar el sistema es necesario leer detenidamente este manual antes de interactuar con el Sistema Servidor Bluetooth.

2. PÁGINAS DE INICIO

A continuación se detallará cada una de las páginas.

Para ingresar al sistema, debe digitar el Usuario y la Clave, estos previamente deben estar registrados en la base de Datos para que sean usuarios válidos para el Sistema.



Figura 2.1 Ingreso al Sistema

- **Botón ingresar**



Una vez que se ha digitado el usuario y la clave, al hacer click en el botón ingresar, se permitirá ingresar al sistema una vez que hayan sido validado los campos con la base de datos.

En el caso que usted digite mal el nombre usuario o clave, al hacer click en el botón ingresar le aparecerá una ventana indicando el siguiente mensaje:



Y en el caso que usted haga click en el botón ingresar sin haber ingresado ningún dato en los campos solicitados, el sistema le pedirá que los ingrese enviándole el siguiente mensaje:



A continuación se muestra la página del Menú Principal del sistema para los usuarios, detallando cada uno de los Ítems. Para poder explicar todo el funcionamiento del sistema iniciaremos sesión como Administrador debido a que como se ha explicado anteriormente cada usuario tendrá privilegios que le permitirán o no acceder a un submenú.



Figura 2.2 Página principal

- **Link Cerrar Sesión:**



Permite cerrar la sesión actual y vuelve a la página de ingreso al sistema.

- **Link Principal:**



Al hacer click en principal le permitirá retornar como su nombre lo indica a la página principal del sistema en el caso que usted se encuentre en cualquier otra página.

3. INGRESO A USUARIOS

Haciendo click en Usuarios se despliega los submenús que facilitará al administrador ingresar y consultar usuarios del sistema o externos.



Figura 3.1 Menú usuarios

En el menú principal de Usuarios aparecen los sub menú los cuales son:

3.1. Sistema:

En este submenú tenemos a su vez dos submenús que nos permitirán ingresar y consultar usuarios del sistema.



Figura 3.2 Submenú Sistema

En el submenú Sistema aparecen los submenús:

3.1.1. Ingresar:

Haciendo click en Ingresar se despliega la página de ingreso de usuarios del sistema.



The screenshot shows a web application interface. At the top, there's a header with a logo on the left and a user profile icon on the right. Below the header, a navigation bar contains links: 'Principal', 'Usuarios', 'Privilegios', 'Envios', 'Reportes', and 'Peticiones'. The 'Usuarios' link is highlighted. The main content area is titled 'INGRESO DE USUARIOS DEL SISTEMA'. It contains a form with the following fields: 'Nombres:', 'Apellidos:', 'Celular:', 'Correo:', 'Usuario:', 'Clave:', 'Confirmar Clave:', and 'Privilegio:'. The 'Privilegio:' field is a dropdown menu with the text 'Seleccione el privilegio'. At the bottom of the form are two buttons: 'Guardar' and 'Limpiar'.

Figura 3.3 Página de ingreso de usuarios del sistema

En esta página ingresaremos los siguientes datos:

- **Nombres:** Es un campo de sólo texto, donde se ingresará el nombre del usuario del sistema.

Por ejemplo:

Nombres:

ESTEFAN CAROLINA

En el caso que no ingrese datos en este campo y haga click en el botón guardar, el sistema le enviará el siguiente mensaje, al aceptar usted podrá ingresar el nombre del usuario.



Como se indicó anteriormente, éste es un campo de sólo texto, si usted ingresa algún número y al hacer click en el botón guardar el sistema le enviará el siguiente mensaje para usted vuelva a revisar el nombre del usuario:



- **Apellidos:** Es un campo de sólo texto, donde se ingresará el apellido del usuario del sistema.

Por ejemplo:

Apellidos:

MEÑEZ PARRALES

En el caso de que usted no ingrese datos en este campo y usted hace click en el botón guardar, el sistema le enviará el siguiente mensaje, al aceptar usted podrá ingresar el apellido del usuario.



Como éste es un campo de sólo texto, si usted ingresa algún número y hace click en el botón guardar el sistema le enviará el siguiente mensaje para usted vuelva a revisar el apellido del usuario:



- **Celular:** Es un campo de sólo número, donde se ingresará el número celular del usuario del sistema.

Por ejemplo:

Celular:

095622558

En el caso que usted no ingrese datos en este campo y hace click en guardar, el sistema le enviará el siguiente mensaje, al aceptar usted podrá ingresar el número celular del usuario.



Cómo es un campo de sólo números, en el caso que usted ingrese letras y haga clic en el botón guardar el sistema le enviará el siguiente mensaje:



Y en el caso que el número de teléfono celular sea menor a 9 números y al hacer click en el botón guardar el sistema le enviará el siguiente mensaje para usted revise el número de teléfono y lo pueda corregir.



Y si el número celular ya está registrado en la base de datos y hace click en el botón guardar, el sistema le enviará el siguiente mensaje:



- **Correo:** Es un campo alfanumérico, donde se ingresará la dirección de correo electrónico del usuario del sistema.

Por ejemplo:

Correo:

5TFF17@hotmail.com

En el caso que usted no ingrese o digite mal la dirección de correo, al hacer click en el botón guardar el sistema le enviará el siguiente mensaje para que usted pueda revisar la dirección y corregirla:



- **Usuario:** Es un campo alfanumérico, donde se ingresará el nombre de usuario con que se identificará para ingresar al sistema.

Por ejemplo: Usuario:

En el caso que usted no ingrese ningún dato en éste campo, y al hacer click en el botón guardar el sistema le presentará el siguiente mensaje:



- **Clave:** Es un campo alfanumérico, donde se ingresará la clave que va a utilizar el usuario para ingresar al sistema.

En el caso que usted no ingrese ningún dato en éste campo y haga click en el botón guardar el sistema le presentará el siguiente mensaje:



- **Confirmar clave:** En este campo deberá ir lo mismo que se ha escrito en Clave para que no exista error.

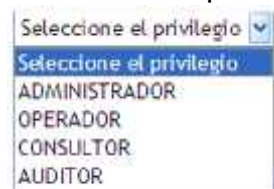
Estas claves no podrán ser visibles y aparecerán de la siguiente manera:

Clave:	<input type="password" value="....."/>
Confirmar Clave:	<input type="password" value="....."/>

En el caso que usted no ingrese ningún valor o la reconfirmación no es la misma, al hacer click en el botón guardar el sistema le presentará el siguiente mensaje:



- **Privilegio:** Es una lista desplegable del cual podrá seleccionar los siguientes privilegios:



Si usted no selecciona ningún dato y hace click en el botón guardar, el sistema le enviará el siguiente mensaje para que usted escoja un privilegio:



Además los siguientes botones:

- **Botón guardar:**

Si usted no ha ingresado ningún dato en los campos antes mencionados y hace click en éste botón, el sistema le pedirá que ingrese el nombre del usuario del sistema como primer elemento necesario para su registro y debe continuar ingresando el resto de información porque el sistema no registra campos vacíos.

Una vez que ha ingresado todos los datos, al hacer click en este botón le permitirá guardar los datos del nuevo usuario del sistema. Le aparece una ventana en la que le pregunta si está usted seguro de ingresar los datos.



Si usted está seguro haga click en aceptar caso contrario haga click en cancelar. Al aceptar le aparecerá el siguiente mensaje el cual indica que ha ingresado un nuevo usuario.



Al aceptar irá directamente a la página de consulta de usuarios del sistema para que usted verifique que se ha realizado el registro correctamente.

- **Botón limpiar:**



Al hacer click en este botón, podrá borrar todo lo que usted ha ingresado para volver a digitar datos.

3.1.2. Consultar:

Haciendo click en Consultar se despliega la página que permite consultar a los usuarios del sistema.



Figura 3.4 Consulta de usuarios del sistema

En donde tenemos los siguientes botones:

- **Botón modificar:**



Antes de hacer click en este botón debe primero seleccionar al usuario a modificar caso contrario le aparecerá el siguiente mensaje:



Una vez seleccionado haga click en este botón el cual le permite modificar los datos del usuario, el sistema le presentará un mensaje para verificar si usted en realidad desea modificar los datos:



Al aceptar se despliega la página en donde usted podrá modificar los datos del usuario:

A screenshot of a web application titled "SERVIDOR BLUETOOTH". The header includes a Bluetooth logo and the text "SERVIDOR BLUETOOTH". Below the header, a navigation bar contains links: "Principal", "Usuarios", "Privilegios", "Envios", "Reportes", and "12 Peticiones". The main content area is titled "MODIFICACION DE USUARIOS DEL SISTEMA" and contains a form for editing user data. The form fields are: "Nombres:" (JUAN CARLOS), "Apellidos:" (CARDENAS VALDEZ), "Celular:" (094921718), "Correo:" (juan@hotmail.com), "Usuario:" (juanc), "Clave:" (*****), and "Privilegio:" (CONSULTOR). At the bottom of the form are two buttons: "Modificar" and "Regresar".

Figura 3.5 Modificación de datos usuario del sistema

Como podemos observar en esta página usted podrá modificar cualquier dato del usuario del sistema, cada campo será validado como se explicó en el ingreso de un usuario.

Tenemos dos botones que son:

- **Modificar:** 

Al hacer click en este botón le aparecerá el siguiente mensaje.



Al aceptar el sistema le enviará un mensaje indicando que la modificación se ha realizado con éxito.



- **Regresar:** 

Al hacer click en éste botón, retornará a la página de consulta de usuarios del sistema.

- **Botón eliminar:**



Antes de hacer click en este botón debe primero seleccionar el usuario a eliminar caso contrario el sistema le presentará el siguiente mensaje:



Usted no podrá eliminar el usuario cuyo privilegio sea administrador, si usted intenta hacerlo el sistema le enviará el siguiente mensaje:



El sistema realiza esta validación debido a que el administrador es el único que puede tener acceso a los privilegios del sistema. Una vez seleccionado el usuario al hacer click en el botón eliminar le aparecerá el siguiente mensaje para confirmar que desea realizar la acción:



Al aceptar, el sistema le presenta el mensaje de confirmación indicando que la eliminación del usuario ha sido realizada:



- **Botón Imprimir:**



Al hacer click en éste botón usted podrá imprimir un detalle de todos los usuarios del sistema que se encuentran registrados en la base de datos, presentándole la ventana donde usted podrá indicar la impresora que va a utilizar:

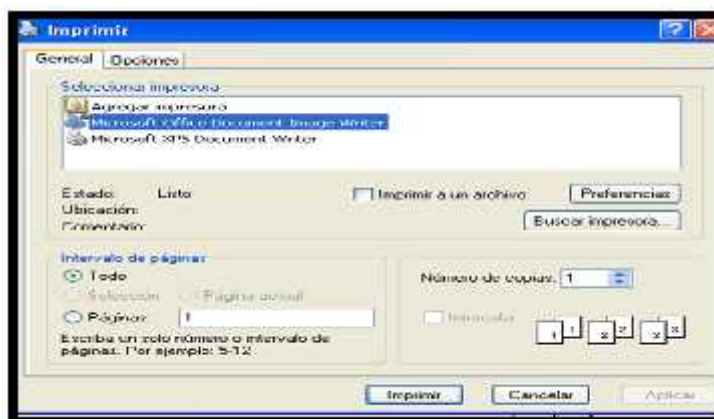


Figura 3.6 Ventana de configuración de la impresora

Al hacer click en imprimir se despliega una nueva ventana donde usted podrá visualizar la información y proceder a imprimir.



Figura 3.7 Impresión de usuarios del sistema

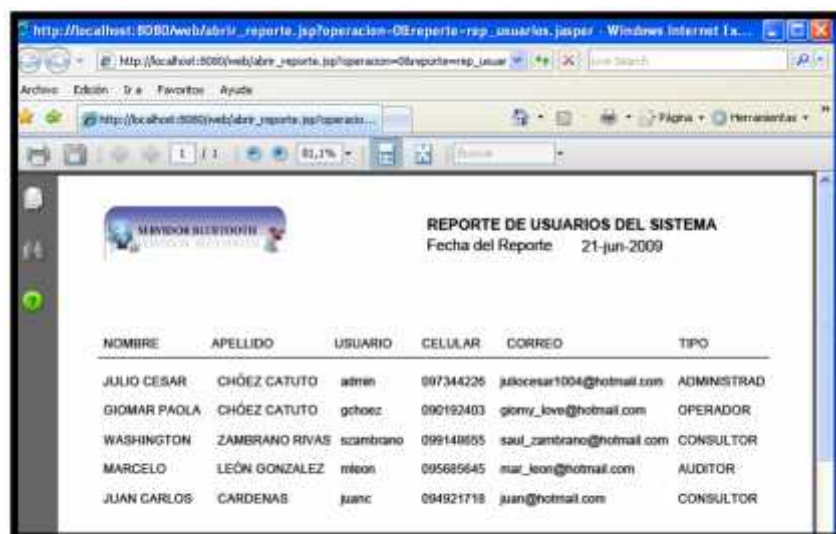
- **Botón Exportar:**



Este botón le permite exportar la información en las siguientes opciones:



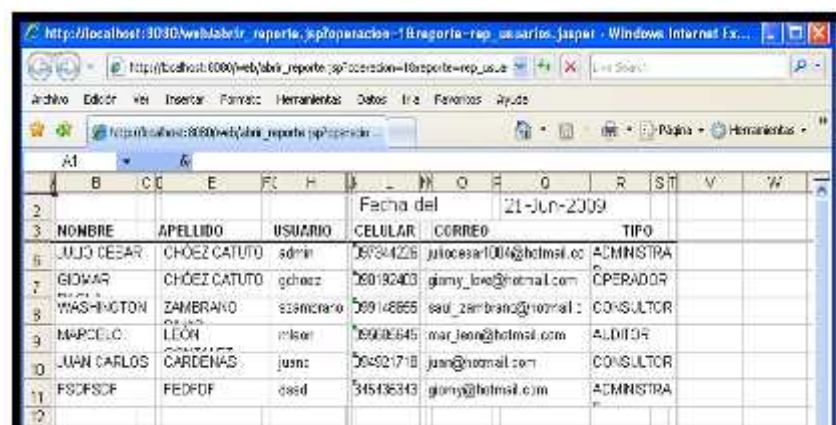
Si usted selecciona el formato PDF la información se le presentará de la siguiente manera:



NOMBRE	APELLIDO	USUARIO	CELULAR	CORREO	TIPO
JULIO CESAR	CHÓEZ CATUTO	admin	007344226	juliocezar1004@hotmail.com	ADMINISTRAD
GIOMAR PAOLA	CHÓEZ CATUTO	gchoez	090192403	giomy_love@hotmail.com	OPERADOR
WASHINGTON	ZAMBRANO RIVAS	scambrano	099148655	saul_zambrano@hotmail.com	CONSULTOR
MARCELO	LEÓN GONZALEZ	mleon	095685645	mar_leon@hotmail.com	AUDITOR
JUAN CARLOS	CARDENAS	juanc	094921718	juan@hotmail.com	CONSULTOR

Figura 3.8 Exportación de usuarios del sistema a PDF

Si usted selecciona el formato EXCEL la información se le presentará de la siguiente manera:



	NOMBRE	APELLIDO	USUARIO	CELULAR	CORREO	TIPO
2	JULIO CESAR	CHÓEZ CATUTO	admin	007344226	juliocezar1004@hotmail.com	ADMINISTRA
6	GIOMAR	CHÓEZ CATUTO	gchoez	090192403	giomy_love@hotmail.com	OPERADOR
7	WASHINGTON	ZAMBRANO	scambrano	099148655	saul_zambrano@hotmail.com	CONSULTOR
8	MARCELO	LEÓN	mleon	095685645	mar_leon@hotmail.com	AUDITOR
9	JUAN CARLOS	CARDENAS	juanc	094921718	juan@hotmail.com	CONSULTOR
10	FEDFDF	FEDFDF	asdd	345436343	giomy@hotmail.com	ADMINISTRA

Figura 3.9 Exportación de usuarios del sistema a EXCEL

3.2. Externos:

En este submenú tenemos a su vez 2 submenús que nos permitirán ingresar y consultar usuarios externos.



Figura 3.10 Submenú Externos

En el submenú Sistema aparecen los submenús:

3.2.1. Ingresar:

Haciendo click en Ingresar se despliega la página de ingreso de usuarios del sistema.



Figura 3.11 Página de ingreso de usuarios externos

En esta página ingresaremos los siguientes datos:

- **Nombres:** Es un campo de sólo texto, donde se ingresará el nombre del usuario del sistema.

Por ejemplo:

Nombres:

MILENA PAOLA

En el caso que no ingrese datos en este campo y hace click en el botón guardar, el sistema le enviará el siguiente mensaje, al hacer click en aceptar usted podrá ingresar el nombre del estudiante.



Como se indicó anteriormente, éste es un campo de sólo texto, si usted ingresa algún número y hace click en el botón guardar el sistema le enviará el siguiente mensaje para usted vuelva a revisar el nombre del alumno:



- **Apellidos:** Es un campo de sólo texto, donde se ingresará el apellido del alumno.

Por ejemplo:

Apellidos: FEBRES CONDERO

En el caso de que usted no ingrese datos en este campo y haga click en el botón guardar, el sistema le enviará el siguiente mensaje, al hacer click en aceptar usted podrá ingresar el apellido del alumno.



Como éste es un campo de sólo texto, si usted ingresa algún número y hace click en el botón guardar el sistema le enviará el siguiente mensaje para usted vuelva a revisar el apellido del alumno:



- **Celular:** Es un campo de sólo número, donde se ingresará el número celular del usuario externo.

Por ejemplo: Celular:

En el caso que usted no ingrese datos en este campo y haga click en el botón guardar, el sistema le enviará el siguiente mensaje, al hacer click en aceptar usted podrá ingresar el número celular del alumno.



Cómo es un campo de sólo números, en el caso que usted ingrese letras y haga click en el botón guardar el sistema le enviará el siguiente mensaje:



O en el caso de que el número de teléfono celular sea menor a 9 números y usted haga click en el botón guardar el sistema le enviará el siguiente mensaje para usted revise el número de teléfono y lo pueda corregir.



Y si el número celular ya está registrado en la base de datos y hace click en el botón guardar, el sistema le enviará el siguiente mensaje:



- **Cédula:** Es un campo de sólo número, donde se ingresará el número de cédula del alumno.

Por ejemplo:

Cédula :

1302608556

En el caso que usted no ingrese datos en este campo y hace click en el botón guardar, el sistema le enviará el siguiente mensaje, al hacer click en aceptar usted podrá ingresar el número de cédula del alumno.



Cómo es un campo de sólo números, en el caso que usted ingrese letras y haga click en el botón guardar el sistema le enviará el siguiente mensaje:



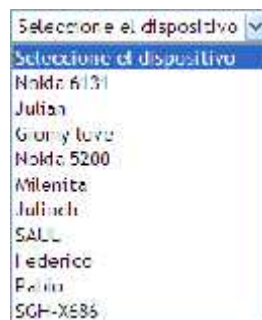
O en el caso de que el número de cédula sea menor a 10 números y hace click en el botón guardar el sistema le enviará el siguiente mensaje para usted revise el número de cédula y lo pueda corregir.



Y si el número celular ya está registrado en la base de datos y usted hace click en el botón guardar, el sistema le enviará el siguiente mensaje:



- **Dispositivo:** Es una lista desplegable del cual podrá seleccionar el dispositivo del usuario:



Si usted no selecciona ningún dispositivo y hace click en el botón guardar el sistema le enviará el siguiente mensaje:



Además en esta página podemos observar los siguientes botones:

- **Botón guardar:** 

Si usted no ha ingresado ningún dato en los campos antes mencionados y hace click en éste botón, el sistema le pedirá que ingrese el nombre del usuario externo en este caso el alumno como primer elemento necesario para su registro y debe continuar ingresando el resto de información porque el sistema no registra campos vacíos.

Una vez que ha ingresado todos los datos, al hacer click en este botón le permitirá guardar los datos del nuevo usuario externo.

Le aparece una ventana en la que le pregunta si está usted seguro de ingresar los datos.

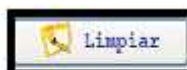


Si usted está seguro haga click en aceptar caso contrario haga click en cancelar. Al aceptar le aparecerá el siguiente mensaje el cual indica que ha ingresado un nuevo usuario.



Al aceptar irá directamente a la página de consulta de usuarios externos para que usted verifique que se ha realizado el registro correctamente.

- **Botón limpiar:**



Al hacer click en este botón, podrá borrar todo lo que usted ha ingresado para volver a digitar datos.

3.2.2. Consultar:

Haciendo click en Consultar se despliega la página que permite consultar a los usuarios externos.



Figura 3.12 Consulta de usuarios externos

En donde tenemos los siguientes botones:

- **Botón modificar:**



Antes de hacer click en este botón debe primero seleccionar al usuario a modificar caso contrario le aparecerá el siguiente mensaje:



Una vez seleccionado haga click en este botón el cual le permite modificar los datos del usuario externo, el sistema le presentará un mensaje para verificar si usted en realidad desea modificar los datos:



Al aceptar se despliega la página en donde usted podrá modificar los datos del usuario externo:

SERVIDOR BLUETOOTH

Bienvenido : JULIO CESAR CHÓEZ CATUTO

Principal Usuarios Privilegios Envios Reportes 12 Peticiones

MODIFICAR USUARIOS EXTERNOS

Nombres : JUAN FERNANDO

Apellidos : LOPEZ RAMIREZ

Celular : 090392005

Cedula : 0926589632

Dispositivo : Nokia 6131

Modificar Regresar

Figura 3.13 Modificación de datos usuario externo

Como podemos observar en esta página usted podrá modificar cualquier dato del usuario del sistema, cada campo será validado como se explicó en el ingreso de un usuario externo.

Tenemos dos botones que son:

- **Modificar:** 

Al hacer click en este botón le aparecerá el siguiente mensaje.



Al aceptar el sistema le enviará un mensaje indicando que la modificación se ha realizado con éxito.



- **Regresar:**



Al hacer click en éste botón, retornará a la página de consulta de usuarios externos.

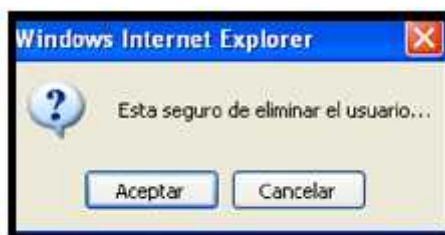
• **Botón eliminar:**



Antes de hacer click en este botón debe primero seleccionar el usuario a eliminar caso contrario el sistema le presentará el siguiente mensaje:



Una vez seleccionado el usuario al hacer click en el botón eliminar le aparecerá el siguiente mensaje para confirmar que desea realizar la acción:



Al aceptar, el sistema le presenta el mensaje de confirmación indicando que la eliminación del usuario ha sido realizada:

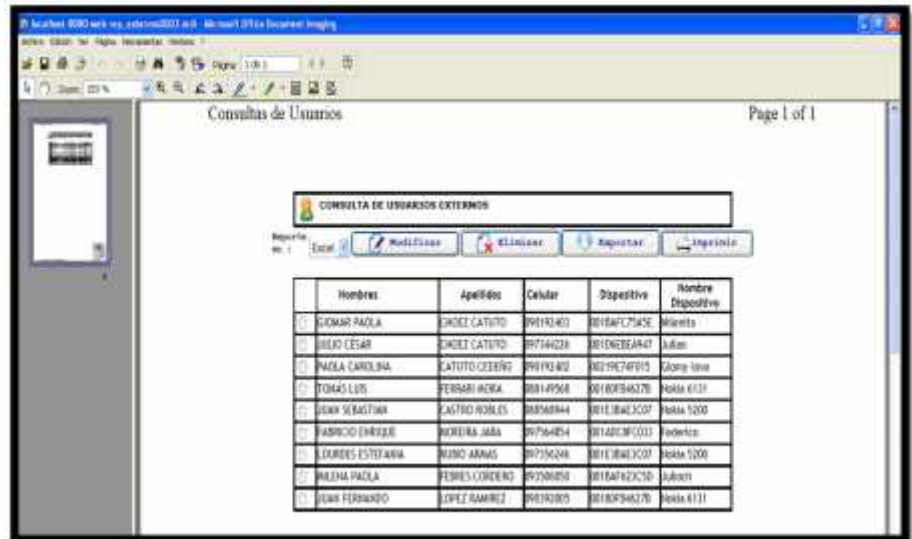


- **Botón Imprimir:**



Al hacer click en éste botón usted podrá imprimir un detalle de todos los usuarios externos que se encuentran registrados en la base de datos, presentándole la ventana donde usted podrá indicar la impresora que va a utilizar como se explico en la opción de imprimir usuarios del sistema figura 3.6.

Al hacer click en imprimir se despliega una nueva ventana donde usted podrá visualizar la información y proceder a imprimir.



Consulta de Usuarios

CONSULTA DE USUARIOS EXTERNOS

Reporte en: [Excel](#) [Modificar](#) [Eliminar](#) [Exportar](#) [Impresión](#)

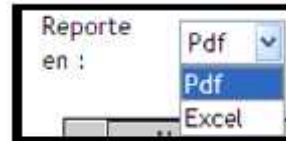
	Nombre	Apellido	Celular	Dispositivo	Nombre Dispositivo
1	DOMINI PAOLA	CHOCLET CATUTO	99193403	99193403	Android
2	JULIO CESAR	CHOCLET CATUTO	99194028	99194028	Android
3	PAOLA CAROLINA	CATUTO CEBERO	99193402	99193402	Android
4	TOMÁS LUIS	FERRARI ACRA	99149568	99149568	Android
5	JUAN SEBASTIAN	CASTRO ROBLES	99568844	99568844	Android
6	ABRILIO ENRIQUE	ACOSTA JARA	99568844	99568844	Android
7	JORDEN ESTEFANIA	NUÑO ARNAS	99756246	99756246	Android
8	PALENA PAOLA	FERRER CORDERO	99756246	99756246	Android
9	JUAN FERNANDO	LOPEZ RAMIREZ	99756246	99756246	Android

Figura 3.14 Impresión de usuarios externos

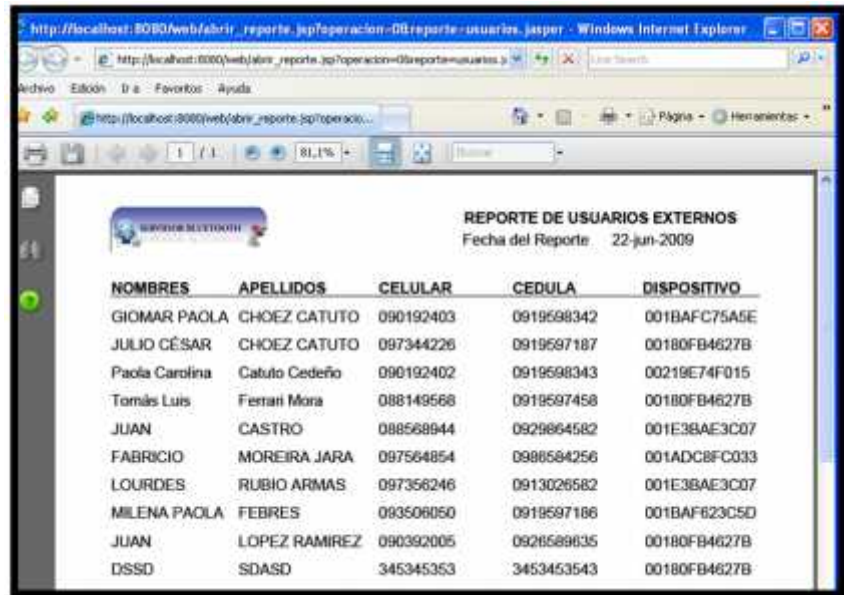
- Botón Exportar:**



Este botón le permite exportar la información en las siguientes opciones:



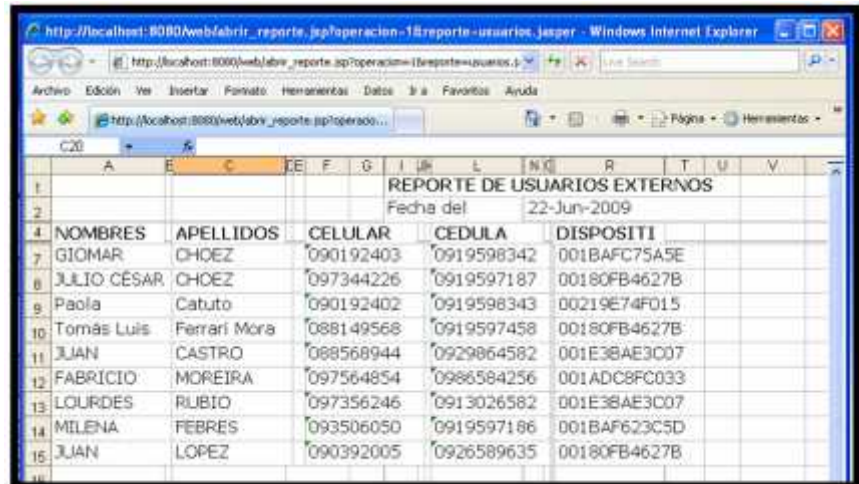
Si usted selecciona el formato PDF la información se le presentará de la siguiente manera:



NOMBRES	APELLIDOS	CELULAR	CEDULA	DISPOSITIVO
GIOMAR PAOLA	CHOEZ CATUTO	090192403	0919598342	001BAFC75A5E
JULIO CÉSAR	CHOEZ CATUTO	097344226	0919597187	00180FB4627B
Paola Carolina	Catuto Cedeño	090192402	0919598343	00219E74F015
Tomás Luis	Ferrari Mora	088149568	0919597458	00180FB4627B
JUAN	CASTRO	088568944	0929864582	001E38AE3C07
FABRICIO	MOREIRA JARA	097564854	0986584256	001ADC8FC033
LOURDES	RUBIO ARMAS	097356246	0913026582	001E38AE3C07
MILENA PAOLA	FEBRES	093506050	0919597186	001BAF623C5D
JUAN	LOPEZ RAMIREZ	090392005	0926589635	00180FB4627B
DSSD	SDASD	345345353	3453453543	00180FB4627B

Figura 3.15 Exportación de usuarios externos a PDF

Si usted selecciona el formato EXCEL la información se le presentará de la siguiente manera:



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
1																						
2																						
3																						
4																						
5																						
6																						
7																						
8																						
9																						
10																						
11																						
12																						
13																						
14																						
15																						
16																						

Figura 3.16 Exportación de usuarios del externos a EXCEL

4. INGRESO A PRIVILEGIOS

Haciendo click en Privilegios se despliega los submenús que facilitará al administrador ingresar y consultar los privilegios para los usuarios del sistema.



Figura 4.1 Menú privilegios

En el menú principal de privilegios aparecen los sub menú los cuales son:

4.1. Ingresar:

Haciendo click en Ingresar se despliega la página de ingreso de privilegios de usuarios del sistema.



Figura 4.2 Página de ingreso de privilegios

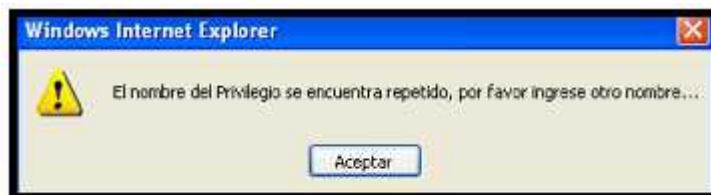
En esta página ingresaremos los siguientes datos:

- **Nombre:** Es un campo alfanumérico, donde se ingresará el nombre del privilegio.

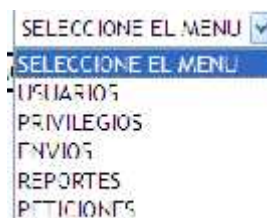
Por ejemplo:

Nombre:

En el caso que usted ingrese un nombre que ya se encuentre registrado en la base de datos y hace click en el botón guardar, el sistema le enviará el siguiente mensaje:



- **Menú:** Es una lista desplegable donde usted podrá seleccionar las páginas que tiene el sistema para indicar en cada una lo que desea que el usuario puede visualizar.



Cuando usted selecciona un menú, le aparecerá en la página los submenús en donde usted podrá elegir cual desea que el usuario puede tener acceso. Por ejemplo: al escoger menú usuarios le aparecerá lo siguiente:

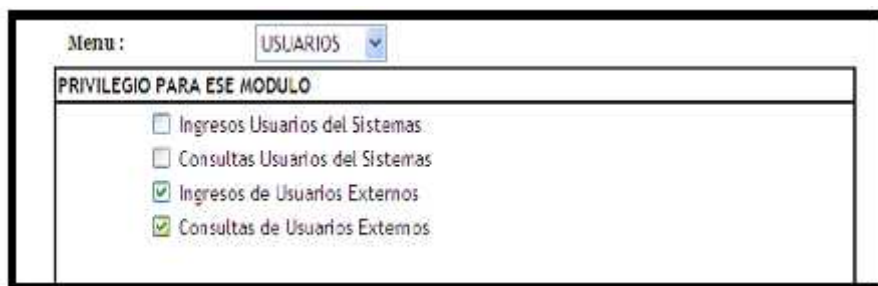


Figura 4.3 Selección de un menú en privilegios

En el caso de que usted no seleccione ningún menú y hace click en el botón guardar, el sistema le enviará el siguiente mensaje:



Y además encontramos los siguientes botones:

- **Botón guardar:**



Si usted no ha ingresado ningún dato en el campo nombre y hace click en éste botón, el sistema le pedirá que ingrese el nombre del privilegio como primer elemento necesario para su registro y continuar con el proceso de ingreso de la publicidad.



Una vez que ha ingresado el nombre y seleccionado las opciones de los menús, al hacer click en este botón le permitirá guardar el nuevo privilegio.

Le aparece una ventana en la que le pregunta si está usted seguro de ingresar los datos.



Si usted está seguro haga click en aceptar caso contrario haga click en cancelar. Al aceptar le aparecerá el siguiente mensaje el cual indica que ha ingresado un nuevo usuario.



Al aceptar irá directamente a la página de consulta de privilegios para que usted verifique que se ha realizado el registro correctamente.

4.2. Consultar:

Haciendo click en Consultar se despliega la página que permite consultar a los privilegios de los usuarios del sistema.



Figura 4.4 Consulta de privilegios de usuarios

En donde tenemos los siguientes botones:

- **Botón modificar:**



Antes de hacer click en este botón debe primero seleccionar al usuario a modificar caso contrario le aparecerá el siguiente mensaje:



Una vez seleccionado haga click en este botón el cual le permite modificar el privilegio, el sistema le presentará un mensaje para verificar si usted en realidad desea modificarlo:



Al aceptar se despliega la página en donde usted podrá modificar el privilegio que seleccionó:



Figura 4.5 Modificación de un privilegio

Como podemos observar en esta página usted podrá modificar el privilegio, cada campo será validado como se explicó en el ingreso del mismo.

Tenemos dos botones que son:

- **Modificar:**



Al hacer click en este botón le aparecerá el siguiente mensaje.



Al aceptar el sistema le enviará un mensaje indicando que la modificación se ha realizado con éxito.



- **Regresar:**



Al hacer click en éste botón, retornará a la página de consulta de privilegios de usuarios.

- **Botón eliminar:**



Antes de hacer click en este botón debe primero seleccionar el privilegio a eliminar caso contrario el sistema le presentará el siguiente mensaje:



Usted no podrá eliminar el privilegios Administrador, debido a que éste privilegio del sistema es el que contiene el acceso a todo el sistema Servidor Bluetooth, al querer realizar ésta acción el sistema le enviará el siguiente mensaje:



Una vez seleccionado el privilegio al hacer click en el botón eliminar le aparecerá éste mensaje de confirmación:



Al aceptar, el sistema le presente el mensaje de confirmación de que la eliminación del privilegio ha sido realizada:



- **Botón Imprimir:**



Al hacer click en éste botón usted podrá imprimir un detalle de todos los privilegios que se encuentran registrados en la base de datos, presentándole la ventana donde usted podrá indicar la impresora que va a utilizar como se mostro en la figura 3.6.

Al hacer click en imprimir se despliega una nueva ventana donde usted podrá visualizar la información y proceder a imprimir.



Figura 4.6 Impresión de privilegios

5. INGRESO A ENVÍOS

Haciendo click en Envíos se despliega los submenús que facilitará al administrador enviar mensajes y publicidad a los usuarios que el dispositivo detecte dentro de su rango de cobertura.



Figura 5.1 Menú Envíos

En el menú principal de envíos aparecen los sub menú los cuales son:

5.1. Mensajes:

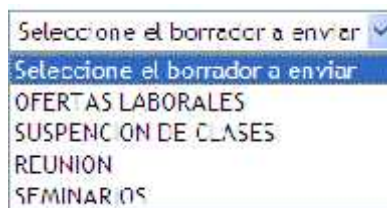
Haciendo click en Mensajes se despliega la página de envío de mensajes.



Figura 5.2 Página envío de mensajes

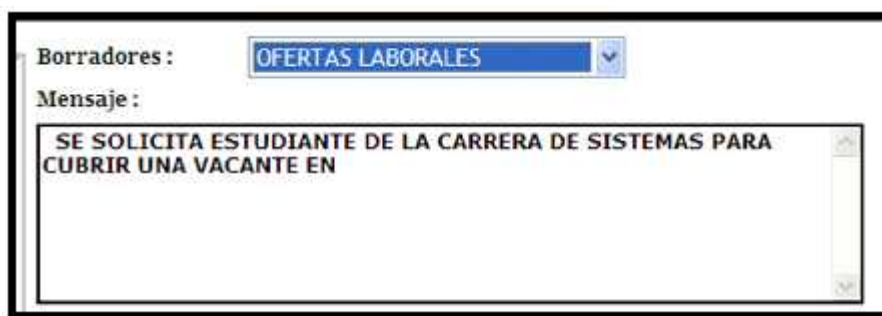
En esta página ingresaremos los siguientes datos:

- **Borradores:** Es una lista desplegable en la cual usted podrá seleccionar el borrador para que le facilite la escritura de un mensaje de texto.



En el caso que usted seleccione un borrador éste le aparecerá en la caja de texto del mensaje.

Por ejemplo: Si usted selección OFERTAR LABORALES le aparecerá en la caja de texto de mensaje lo siguiente:



Borradores : OFERTAS LABORALES

Mensaje :

SE SOLICITA ESTUDIANTE DE LA CARRERA DE SISTEMAS PARA CUBRIR UNA VACANTE EN

Figura 5.3 Selección de un borrador

- **Mensaje:** Es una caja de texto donde usted podrá ingresar el mensaje que desea enviar a los dispositivos.

Por ejemplo, como seleccionamos anteriormente un borrador, completamos la información como se muestra:



Mensaje :

SE SOLICITA ESTUDIANTE DE LA CARRERA DE SISTEMAS PARA CUBRIR UNA VACANTE EN LA EMPRESA OPEVIAL S.A. LOS INTERESADOS COMUNICARSE AL 2564253 - ING. JOSE PINEDA

Si usted no ingresa ningún mensaje y al hacer click en enviar el sistema le mostrará el siguiente mensaje:



- **Consola:** Es una caja de texto donde aparecerá el detalle de los envíos de los mensajes.

Por ejemplo:

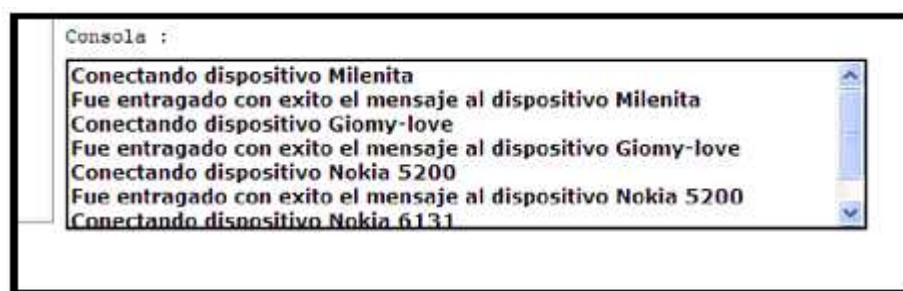


Figura 5.4 Presentación de la consola

Además los siguientes botones:

- **Botón Enviar:**



En primera instancia cuando se ingresa a la página de envío de mensajes, éste botón aparece desactivado.

Si usted hace click en éste botón sin haber seleccionado el o los dispositivos, el sistema le enviará el siguiente mensaje:



Luego de que usted seleccione el o los dispositivos a quienes va a enviar el mensaje, haga click en este botón para proceder a su envío. El sistema le mostrará el siguiente mensaje:



Al aceptar empieza el envío de los mensajes a los dispositivos. Una vez que termine el envío de los mensajes, aparece en la consola el detalle si fue entregado o hubo algún error.

- **Botón Borrar:**



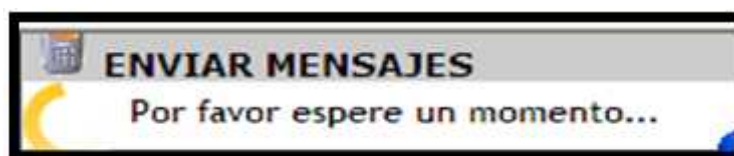
Este botón en primera instancia al igual que el botón enviar aparecen deshabilitados. Le permite borrar el mensaje que usted haya digitado.

- **Botón Actualizar:**



Cuando se ingresa a la página de envío de mensajes, éste es el único botón que se encuentra habilitado y le permite actualizar el listado de los dispositivos que el sistema ha detectado en ese momento.

En la parte del nombre de la página, aparecerá éste mensaje:



Cuando termine de detectar los dispositivos aparecerá un listado como el que se presenta a continuación y en donde usted podrá elegir el o los dispositivos:



Podrá seleccionar haciendo click con el mouse o si desea puede activar la casilla TODOS y elegirá todos los dispositivos:



5.2. Publicidad:

Haciendo click en Publicidad se despliega la página de consulta de envío de publicidad.



Figura 5.5 Página consulta de envío de publicidad

En esta página podemos observar los siguientes botones:

- **Nueva:**



Al hacer click en éste botón, se despliega una nueva página en donde usted podrá ingresar una nueva publicidad:

Bienvenido : JULIO CESAR CHÓEZ CATUTO

Principal Usuarios Privilegios Envíos Reportes 12 Peticiónes

INGRESO DE PUBLICIDAD

Nombre:

Imagen: Examinar...

Días:

- ☐ Lunes
- ☐ Martes
- ☐ Miércoles
- ☐ Jueves
- ☐ Viernes
- ☐ Sábado
- ☐ Domingo

Estado: SELECCIONAR EL ESTADO

Figura 5.6 Ingreso de publicidad

En ésta página tenemos la siguiente información:

- **Nombre:** Es un campo alfanumérico en donde usted podrá ingresar el nombre de la publicidad.

Por ejemplo:

Nombre

Si usted no ingresa ningún dato en éste campo y hace click en guardar el sistema le enviará el siguiente mensaje:



- **Imagen:** Es un campo alfanumérico en donde se almacenará la ruta o la ubicación de la imagen.

Si usted no selecciona la imagen y hace click en guardar el sistema le enviará el siguiente mensaje:



- **Días:** Son opciones en los que usted podrá definir en qué días desea que se envíe la publicidad.

Días :

Por ejemplo:

-
- ☒ Lunes
 - ☐ Martes
 - ☐ Miercoles
 - ☐ Jueves
 - ☐ Viernes
 - ☐ Sabado
 - ☐ Domingo

- **Estado:** Es una lista desplegable donde usted podrá seleccionar el estado de la publicidad es decir activo o inactivo.



Si usted no seleccionar ningún estado y hace click en el botón guardar el sistema le enviará el siguiente mensaje:



Y además tenemos los siguientes botones:

- **Botón Examinar:** 

Al hacer click en éste botón usted podrá buscar dentro de sus archivos la imagen de la publicidad.

Si usted intenta seleccionar una imagen que no sea con extensión gif o jpg y hace click en el botón guardar el sistema le enviará el siguiente mensaje:



- **Botón Limpiar:**



Al hacer click en este botón, podrá borrar todo lo que usted ha ingresado para volver a digitar o seleccionar.

- **Botón Guardar:**



Al hacer click en éste botón si usted no ha ingresado ninguna información, el sistema le pedirá que ingrese el nombre de la publicidad como primer elemento necesario para su registro y continuar con el proceso de ingreso de información en ésta página.

Una vez ingresada toda la información, al hacer click en guardar, el sistema le presentará el siguiente mensaje:



Al aceptar, se despliega la página de consulta en donde usted podrá confirmar el ingreso de la publicidad.

- **Botón Eliminar:**



Antes de hacer click en éste botón usted debe seleccionar la publicidad que desea eliminar caso contrario el sistema le enviará el siguiente mensaje:



Una vez seleccionada la publicidad, al hacer click en éste botón el sistema le enviará el siguiente mensaje de confirmación:



Al hacer click en aceptar, el registro de la publicidad que usted selecciono se eliminará de la base de datos. Y el sistema le enviará el siguiente mensaje:



- **Botón Reporte:**



Antes de hacer click en este botón usted debe seleccionar la publicidad de la cuál desea ver el reporte caso contrario el sistema le presentará el siguiente mensaje:



Una vez seleccionada la publicidad al hacer click en este botón podrá visualizar de manera gráfica global a cuántos dispositivos fue enviada la publicidad y cuántos dispositivos la ignoraron.

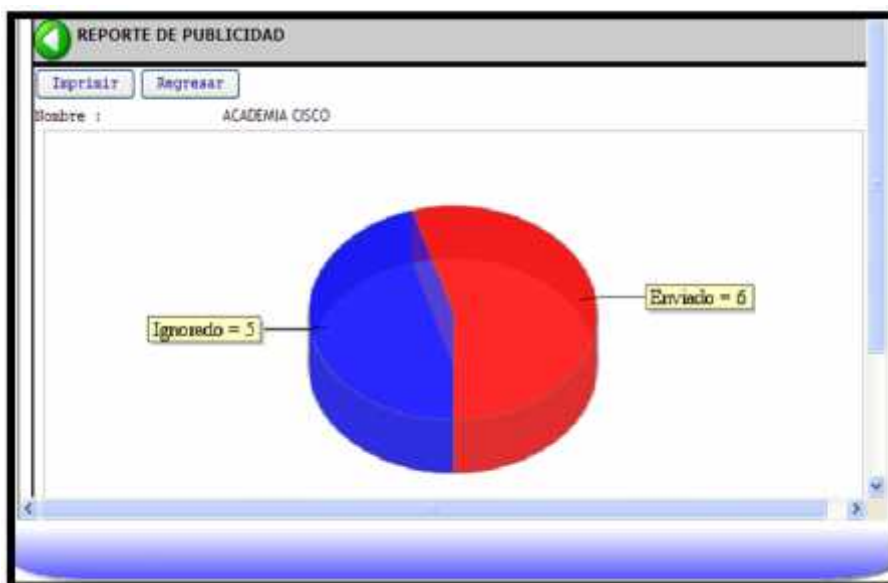


Figura 5.7 Reporte de una publicidad

En donde podemos observar estos botones:

- **Botón Imprimir:**



Al hacer click en éste botón podrá imprimir el gráfico. El sistema le presentará la ventana donde podrá configurar la impresora como se indicó en la figura 3.6.

Al hacer click en imprimir se despliega una nueva ventana donde usted podrá visualizar la información y proceder a imprimir.

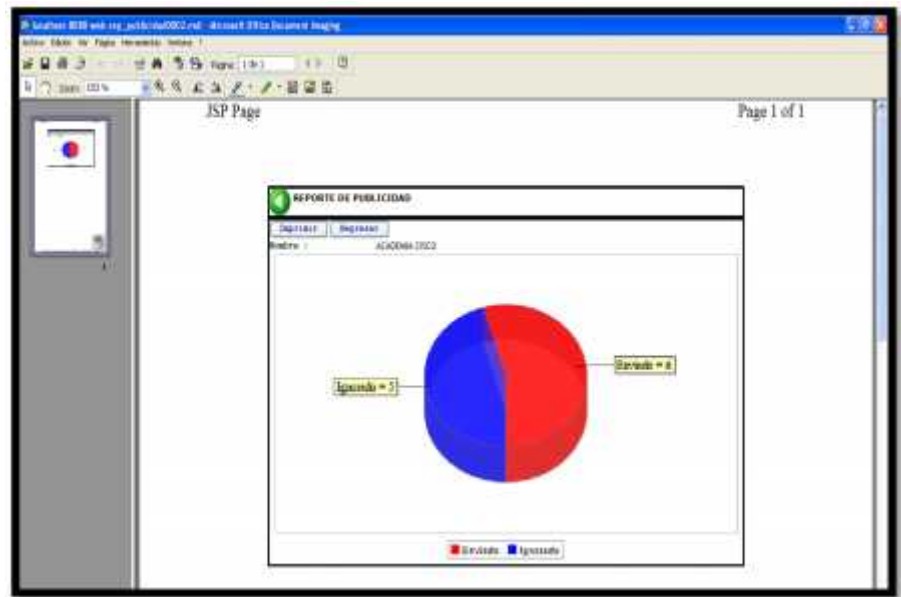


Figura 5.8 Impresión del reporte de publicidad

- **Botón Regresar:**



Al hacer click en éste botón, retornará a la página de consulta de envío de publicidad.

- **Botón Modificar:**



Antes de hacer click en éste botón, debe primero seleccionar la publicidad que se desea modificar, caso contrario el sistema le presentará el siguiente mensaje:



Una vez seleccionada la publicidad, al hacer click en éste botón el sistema le presentará el siguiente mensaje:



Al aceptar, se despliega la página de modificación

A screenshot of a web form titled 'MODIFICA DE PUBLICIDAD'. The form has a light gray header bar with a green circular icon and the title. Below the header, there are three main sections: 'Nombre' with a text input field containing 'ACADEMIA CISCO'; 'Días' with a list of days and checkboxes: 'Lunes' (checked), 'Martes', 'Miércoles', 'Jueves', 'Viernes', 'Sábado', and 'Domingo'; and 'Estado' with a dropdown menu showing 'ACTIVA'. At the bottom right, there are two buttons: 'Modificar' with a green circular icon and 'Regresar' with a green circular icon.

Figura 5.9 Modificación de una publicidad

En ésta página podemos observar los siguientes botones:

- **Botón modificar:**  El botón 'Modificar' es un rectángulo con un borde negro, contiene un icono de una llave inglesa a la izquierda y el texto 'Modificar' a la derecha.

Al hacer click en éste botón, el sistema le enviará el siguiente mensaje:



- **Botón regresar:**  El botón 'Regresar' es un rectángulo con un borde negro, contiene un icono de una flecha verde a la izquierda y el texto 'Regresar' a la derecha.

Al hacer click en éste botón, retornará a la página de consulta de envío de publicidad.

6. INGRESO A REPORTE

Haciendo click en Reporte se despliega los submenús que facilitará al administrador visualizar los diferentes reportes, acceder al buzón de mensajes y ver el top ten de los dispositivos que utilizan más el sistema.



Figura 6.1 Menú Reportes

En el menú principal de reportes aparecen los sub menú los cuales son:

6.1. Dispositivos Encontrados:

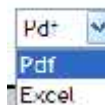
Haciendo click en Dispositivos Encontrados se despliega la página de reporte de los dispositivos que el Servidor Bluetooth ha detectado.

Nombre	Dirección	Fecha de Detección	Fecha de Último Acceso
Nokia 6131	00180FB4627B	2009-06-05 21:41:54	2009-06-22 00:00:00.0
Julian	001D6E8EA947	2009-06-05 22:32:10	2009-06-22 00:00:00.0
Julioch	001BAF623C9D	2009-06-05 22:32:54	2009-06-22 00:00:00.0
Nokia 5200	001E3BAE3C07	2009-06-05 21:41:56	2009-06-22 00:00:00.0
Glomy-love	00219E74F015	2009-06-05 21:41:56	2009-06-20 00:00:00.0
Adlenita	001BAFC75A5E	2009-06-05 22:00:40	2009-06-23 00:00:00.0
SAUL	001ADC975B1A	2009-06-09 21:35:16	2009-06-22 00:00:00.0
JULI	00247E243319	2009-06-16 15:55:38	2009-06-22 00:00:00.0
Federico	001ADC8FC033	2009-06-05 21:42:43	2009-06-21 00:00:00.0

Figura 6.2 Página envío de mensajes

En esta página ingresaremos los siguientes datos:

- **Reporte en:** Donde podemos seleccionar si queremos exportar en formato PDF o EXCEL.



Además podemos apreciar los siguientes botones:

- **Botón exportar:**



Al hacer clic en éste botón podemos exportar el documento en dos tipos de formato:

Si usted selecciona el formato PDF la información se le presentará de la siguiente manera:

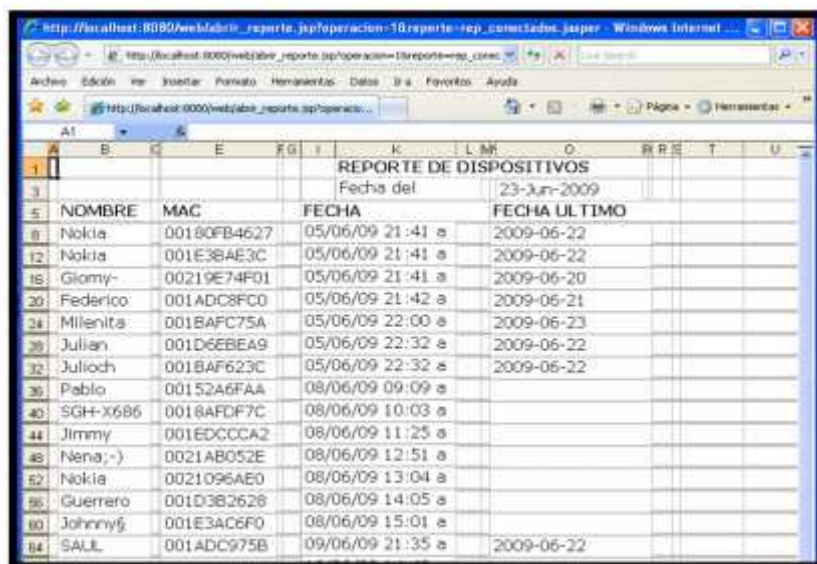
REPORTE DE DISPOSITIVOS
Fecha del Reporte 23-jun-2009

NOMBRES	MAC	FECHA DETECCION	FECHA ULTIMO ACCESO
Nokia 6131	00180FB46278	05/06/09 09:41 PM	2009-06-22 00:00:00
Nokia 5200	001E3BAE3C07	05/06/09 09:41 PM	2009-06-22 00:00:00
Giomy-love	00219E74FB15	05/06/09 09:41 PM	2009-06-20 00:00:00
Federico	001ADC8FC033	05/06/09 09:42 PM	2009-06-21 00:00:00
Milenita	001BAFC75A5E	05/06/09 10:00 PM	2009-06-23 00:00:00
Julian	001D6E8EA947	05/06/09 10:32 PM	2009-06-22 00:00:00
Julioch	001BAF623C5D	05/06/09 10:32 PM	2009-06-22 00:00:00
Pablo	00152A6FAA05	08/06/09 09:09 AM	
SGH-X688	0018AFDF7C0C	08/06/09 10:03 AM	
Jimmy	001EDCCCA2F8	08/06/09 11:25 AM	
Nena, J	0021AB052E23	08/06/09 12:51 PM	
Nokia 2630	0021096AE0C	08/06/09 01:04 PM	
Guernero de	001D3B262812	08/06/09 02:05 PM	
JohnnyS	001E3AC8F0A3	08/06/09 03:01 PM	
SALU	001ADC975B1A	09/06/09 09:35 PM	2009-06-22 00:00:00
*- TBE	001E3B80D338	16/06/09 02:49 PM	
KRta...l	002298B3A51C	16/06/09 03:02 PM	
JULI	00247E243319	16/06/09 03:55 PM	2009-06-22 00:00:00

Fecha Zona desconocida

Figura 6.3 Exportación Reporte de Dispositivos a PDF

Si usted selecciona el formato EXCEL la información se le presentará de la siguiente manera:



REPORTE DE DISPOSITIVOS			
Fecha del			23-Jun-2009
NOMBRE	MAC	FECHA	FECHA ULTIMO
Nokia	00180FB4627	05/06/09 21:41 a	2009-06-22
Nokia	001E38AE3C	05/06/09 21:41 a	2009-06-22
Glomy-	00219E74F01	05/06/09 21:41 a	2009-06-20
Federico	001ADC8FC0	05/06/09 21:42 a	2009-06-21
Milenita	001BAFC75A	05/06/09 22:00 a	2009-06-23
Julian	001D6E8EA9	05/06/09 22:32 a	2009-06-22
Juliodh	001BAF623C	05/06/09 22:32 a	2009-06-22
Pablo	00152A6FAA	08/06/09 09:09 a	
SGH-X686	0018AFDF7C	08/06/09 10:03 a	
Jimmy	001EDCCA2	08/06/09 11:25 a	
Nena;-)	0021AB052E	08/06/09 12:51 a	
Nokia	0021096AE0	08/06/09 13:04 a	
Guerrero	001D3B2628	08/06/09 14:05 a	
Johnnyfj	001E3AC6F0	08/06/09 15:01 a	
SAUL	001ADC975B	09/06/09 21:35 a	2009-06-22

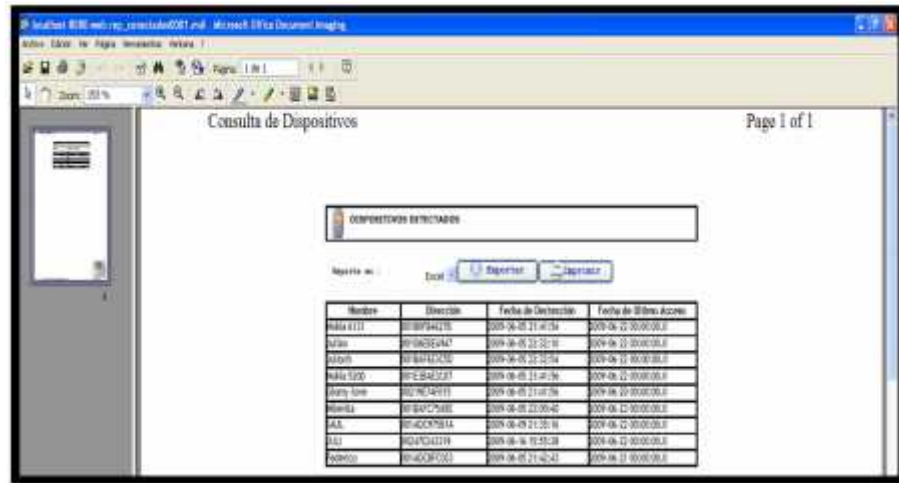
Figura 6.4 Exportación Reporte de Dispositivos a EXCEL

- **Botón imprimir:**



Al hacer click en éste botón usted podrá imprimir un detalle de los dispositivos detectados por el sistema, presentándole la ventana donde usted podrá indicar la impresora que va a utilizar como se mostro en la figura 3.6.

Al hacer click en imprimir se despliega una nueva ventana donde usted podrá visualizar la información y proceder a imprimir.



Consulta de Dispositivos

Page 1 of 1

DISPOSITIVOS DETECTADOS

Exportar

Nombre	Dirección	Fecha de Detección	Fecha de Último Acceso
Moto x111	88:9F:94:27:55	2019-06-05 21:48:54	2019-06-12 00:00:00.0
AJAX	88:9F:94:27:55	2019-06-05 22:22:19	2019-06-12 00:00:00.0
ADP01	88:9F:94:27:55	2019-06-05 22:22:19	2019-06-12 00:00:00.0
Moto X120	88:9F:94:27:55	2019-06-05 21:48:54	2019-06-12 00:00:00.0
Smart phone	88:9F:94:27:55	2019-06-05 21:48:54	2019-06-12 00:00:00.0
Moto x111	88:9F:94:27:55	2019-06-05 22:00:42	2019-06-12 00:00:00.0
AJAX	88:9F:94:27:55	2019-06-05 21:20:16	2019-06-12 00:00:00.0
AJAX	88:9F:94:27:55	2019-06-05 21:20:16	2019-06-12 00:00:00.0
AJAX	88:9F:94:27:55	2019-06-05 21:20:16	2019-06-12 00:00:00.0
AJAX	88:9F:94:27:55	2019-06-05 21:20:16	2019-06-12 00:00:00.0

Figura 6.5 Impresión de Dispositivos detectados

6.2. Buzones:

En este submenú tenemos a su vez 5 submenús que nos permitirán ver un reporte de los mensajes enviados, recibidos, los borradores, y el contador de mensajes:



Figura 6.6 Submenú Buzones

En el submenú Buzones aparecen los submenús:

6.2.1. Entrada:

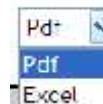
Al hacer click en éste submenú se despliega la página de Buzón de entrada.



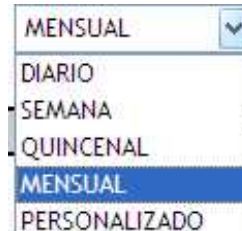
Figura 6.7 Buzón de entrada

En donde tenemos lo siguiente:

- **Reporte en:** Es una lista desplegable en donde podrá seleccionar el formato en que usted puede exportar la información en PDF o EXCEL.



- **Visualización:** Es una lista desplegable en donde podrá seleccionar la forma en que desea visualizar la información.



Además tenemos los botones:

- **Botón Exportar:**



Si usted selecciona el formato PDF la información se le presentará de la siguiente manera:

DISPOSITIVOS	NOMBRE	MENSAJES	FECHA
001BAFC75A5E	Milenita	0010508342.netas	05/06/09 10:40 PM
001BAFC75A5E	Milenita	00102403.nótielas primero	00/06/09 09:48 AM
001BAFC75A5E	Milenita	00102403.ayudomita	00/06/09 05:04 PM
001BAFC75A5E	Milenita	00102403.ayudantia	00/06/09 05:05 PM
001BAFC75A5E	Milenita	00102403.hocano	00/06/09 05:14 PM
001BAFC75A5E	Milenita	00102403.laboratorio	00/06/09 05:18 PM
001BAFC75A5E	Milenita	00102403.laboratorio	00/06/09 05:19 PM
001BAFC75A5E	Milenita	00102403.novedades	00/06/09 05:21 PM
001BAFC75A5E	Milenita	00102403.novedades	00/06/09 05:22 PM
001BAFC75A5E	Milenita	00102403.novedades	00/06/09 05:22 PM
001BAFC75A5E	Milenita	00102403.novedades.seminarios	00/06/09 05:37 PM
001BAFC75A5E	Milenita	00102403.novedades.seminarios	00/06/09 05:37 PM
001BAFC75A5E	Milenita	00102403.novedades.seminarios	00/06/09 05:38 PM
001BAFC75A5E	Milenita	00102403.examenos	00/06/09 05:42 PM
001BAFC75A5E	Milenita	00102403.examenos	00/06/09 05:51 PM
001BAFC75A5E	Milenita	00102403.examenos	00/06/09 05:52 PM

Figura 6.8 Exportación Buzón de entrada a PDF

Si usted selecciona el formato EXCEL la información se le presentará de la siguiente manera:

REPORTE DE BUZÓN DE ENTRADA			
Fecha del Reporte 21-Jun-2009			
DISPOSITIVOS	NOMBRE	MENSAJES	FECHA
001BAFC75ASE	Mianita	0919240342.notas	05/06/09 22:40 a
001BAFC75ASE	Mianita	090192403.noticias.primer	05/06/09 09:48 a
001BAFC75ASE	Mianita	090192403.ayudantia	09/06/09 17:04 a
001BAFC75ASE	Mianita	090192403.ayudantia	09/06/09 17:05 a
001BAFC75ASE	Mianita	090192403.horario	09/06/09 17:14 a
001BAFC75ASE	Mianita	090192403.laboratorio	09/06/09 17:18 a
001BAFC75ASE	Mianita	090192403.laboratorio	09/06/09 17:19 a
001BAFC75ASE	Mianita	090192403.novedades	09/06/09 17:21 a
001BAFC75ASE	Mianita	090192403.novedades	09/06/09 17:22 a
001BAFC75ASE	Mianita	090192403.novedades	09/06/09 17:22 a
001BAFC75ASE	Mianita	090192403.novedades.seminar	09/06/09 17:37 a
001BAFC75ASE	Mianita	090192403.novedades.seminar	09/06/09 17:37 a
001BAFC75ASE	Mianita	090192403.novedades.seminar	09/06/09 17:38 a
001BAFC75ASE	Mianita	090192403.examenes	09/06/09 17:42 a
001BAFC75ASE	Mianita	090192403.examenes	09/06/09 17:51 a
001BAFC75ASE	Mianita	090192403.examenes	09/06/09 17:52 a
001BAFC75ASE	Mianita	090192403.ayudantia.redes	09/06/09 18:08 a
001BAFC75ASE	Mianita	090192403.ayudantia.redes	09/06/09 18:09 a
001BAFC75ASE	Mianita	090192403.ayudantia.redes	14/06/09 21:47 a

Figura 6.9 Exportación Buzón de entrada a EXCEL

Si usted hace click en este botón y no ha ninguna búsqueda, el sistema le enviará el siguiente mensaje:



Si usted hace click en éste botón y no ha realizado ninguna búsqueda el sistema le presentará el siguiente mensaje:



Botón Buscar:



Al hacer click en este botón podrá visualizar la información de acuerdo a lo que usted seleccionó en la lista desplegable de visualización.



Figura 6.11 Búsqueda semanal buzón de entrada

6.2.2. Salida:

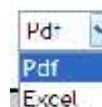
Al hacer click en éste submenú se despliega la página de Buzón de salida.



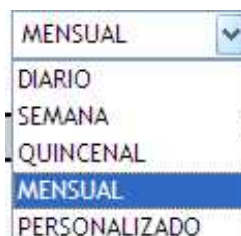
Figura 6.12 Buzón de salida

En donde tenemos lo siguiente:

- **Reporte en:** Es una lista desplegable en donde podrá seleccionar el formato en que usted puede exportar la información en PDF o EXCEL.

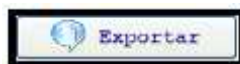


- **Visualización:** Es una lista desplegable en donde podrá seleccionar la forma en que desea visualizar la información.



Además tenemos los botones:

- Botón Exportar:

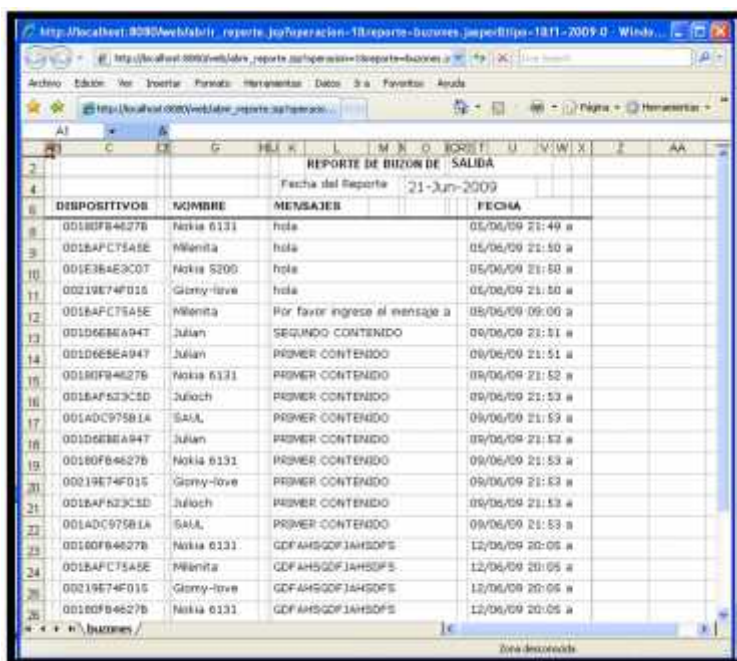


Si usted selecciona el formato PDF la información se le presentará de la siguiente manera:

DISPOSITIVOS	NOMBRE	MENSAJES	FECHA
0018FB4627B	Nokia 6131	hola	05/06/09 09:49 PM
0018AF623C5D	Milenita	hola	05/06/09 09:50 PM
001E3BAE3C07	Nokia 6200	hola	05/06/09 09:50 PM
00219E74FD15	Glamy love	hola	05/06/09 09:50 PM
0018AF623C5D	Milenita	Por favor ingrese el mensaje a	06/06/09 09:00 AM
001D8EBA947	Julian	SEGUNDO CONTENIDO	06/06/09 09:51 PM
001D8EBA947	Julian	PRIMER CONTENIDO	06/06/09 09:51 PM
0018FB4627B	Nokia 6131	PRIMER CONTENIDO	06/06/09 09:52 PM
0018AF623C5D	Julioch	PRIMER CONTENIDO	06/06/09 09:53 PM
001ADC975B1A	SAUL	PRIMER CONTENIDO	06/06/09 09:53 PM
001D8EBA947	Julian	PRIMER CONTENIDO	06/06/09 09:53 PM
0018FB4627B	Nokia 6131	PRIMER CONTENIDO	06/06/09 09:53 PM
00219E74FD15	Glamy love	PRIMER CONTENIDO	06/06/09 09:53 PM
0018AF623C5D	Julioch	PRIMER CONTENIDO	06/06/09 09:53 PM
001ADC975B1A	SAUL	PRIMER CONTENIDO	06/06/09 09:53 PM
0018FB4627B	Nokia 6131	GDFAH8GDFJAH8DFS	12/06/09 08:05 PM

Figura 6.13 Exportación Buzón de salida a PDF

Si usted selecciona el formato EXCEL la información se le presentará de la siguiente manera:



DISPOSITIVOS	NOMBRE	MENSAJES	FECHA
00180FB4627B	Nokia 6131	hola	05/06/09 21:49 a
0018AFC75A5E	Milenita	hola	05/06/09 21:50 a
001E3B4E9C07	Nokia S200	hola	05/06/09 21:50 a
00219E74F016	Gleny-love	hola	05/06/09 21:50 a
0018AFC75A5E	Milenita	Por favor ingrese el mensaje a	05/06/09 09:00 a
001D6E8EA947	Julian	SEGUNDO CONTENIDO	09/06/09 21:51 a
001D6E8EA947	Julian	PRIMER CONTENIDO	09/06/09 21:51 a
00180FB4627B	Nokia 6131	PRIMER CONTENIDO	09/06/09 21:52 a
0018AFC75A5E	Julian	PRIMER CONTENIDO	09/06/09 21:53 a
001AD0C97581A	SALA	PRIMER CONTENIDO	09/06/09 21:53 a
001D6E8EA947	Julian	PRIMER CONTENIDO	09/06/09 21:53 a
00180FB4627B	Nokia 6131	PRIMER CONTENIDO	09/06/09 21:53 a
00219E74F016	Gleny-love	PRIMER CONTENIDO	09/06/09 21:53 a
0018AFC75A5E	Julian	PRIMER CONTENIDO	09/06/09 21:53 a
001AD0C97581A	SALA	PRIMER CONTENIDO	09/06/09 21:53 a
00180FB4627B	Nokia 6131	GDFAHSGDFJAHSGPS	12/06/09 20:05 a
0018AFC75A5E	Milenita	GDFAHSGDFJAHSGPS	12/06/09 20:05 a
00219E74F016	Gleny-love	GDFAHSGDFJAHSGPS	12/06/09 20:05 a
00180FB4627B	Nokia 6131	GDFAHSGDFJAHSGPS	12/06/09 20:05 a

Figura 6.14 Exportación Buzón de salida a EXCEL

Si usted hace click en este botón y no ha ninguna búsqueda, el sistema le enviará el siguiente mensaje:



- **Botón Imprimir:**



Al hacer click en este botón le presentará una ventana en donde podrá configurar la impresora como se mostro en la figura 3.6.

Al hacer click en imprimir se despliega una nueva ventana donde usted podrá visualizar la información y proceder a imprimir.

Page 1 of 1

WATER DE NAIJES

Reporte en:

Visualizar en:

Dispositivo	Medida	Medida	Fecha
01-BF-640115	NADA 41 10	WGA	2009-06-09 11:05:00.0
01-BF-640120	NADA 41	WGA	2009-06-09 11:06:14.0
01-BF-640121	NADA 41 10	WGA	2009-06-09 11:06:31.0
01-BF-640111	Empty line	WGA	2009-06-09 11:06:38.0
01-BF-640120	NADA 41	For this report is necessary a service...	2009-06-09 11:06:41.0
01-BF-640147	Empty	PRIMER CONTINIO	2009-06-09 11:01:41.0
01-BF-640147	Empty	PRIMER CONTINIO	2009-06-09 11:01:58.0
01-BF-640115	NADA 41 10	PRIMER CONTINIO	2009-06-09 11:02:14.0
01-BF-640120	NADA 41	PRIMER CONTINIO	2009-06-09 11:02:00.0
01-AC-SF-0114	NADA	PRIMER CONTINIO	2009-06-09 11:03:08.0
01-BF-640147	Empty	PRIMER CONTINIO	2009-06-09 11:03:38.0
01-BF-640115	NADA 41 10	PRIMER CONTINIO	2009-06-09 11:03:38.0
01-BF-640111	Empty line	PRIMER CONTINIO	2009-06-09 11:03:41.0
01-BF-640120	NADA 41	PRIMER CONTINIO	2009-06-09 11:03:45.0
01-AC-SF-0114	NADA	PRIMER CONTINIO	2009-06-09 11:03:51.0
01-BF-640115	NADA 41 10	PRIMER CONTINIO	2009-06-09 11:03:58.0
01-BF-640120	NADA 41	PRIMER CONTINIO	2009-06-09 11:03:58.0
01-BF-640111	Empty line	PRIMER CONTINIO	2009-06-09 11:03:58.0
01-BF-640115	NADA 41 10	PRIMER CONTINIO	2009-06-09 11:03:58.0
01-BF-640121	NADA 41 10	PRIMER CONTINIO	2009-06-09 11:03:58.0

Si usted hace click en éste botón y no ha realizado ninguna búsqueda el sistema le presentará el siguiente mensaje:



- **Botón Buscar:**



Al hacer click en este botón podrá visualizar la información de acuerdo a lo que usted seleccionó en la lista desplegable de visualización.

Dispositivo	Nombre	Mensaje	Fecha
00180FB4627B	Nokia 6131	hola	2009-06-05 21:49:59.0
001BAFC75A5E	Wilkenita	hola	2009-06-05 21:50:14.0
001E3BAE3C07	Nokia 5200	hola	2009-06-05 21:50:23.0
0021NE74F015	Glomylove	hola	2009-06-05 21:50:38.0
001BAFC75A5E	Wilkenita	Por favor ingrese al mensaje a enviar...	2009-06-08 09:00:43.0
001D4EBEA947	Julian	SEGUNDO CONTENIDO	2009-06-09 21:51:41.0
001D4EBEA947	Julian	PRIMER CONTENIDO	2009-06-09 21:51:56.0
00180FB4627B	Nokia 6131	PRIMER CONTENIDO	2009-06-09 21:52:14.0
001BAF623C5D	Julioch	PRIMER CONTENIDO	2009-06-09 21:53:00.0

Figura 6.16 Búsqueda semanal buzón de salida

6.2.3. No enviados:

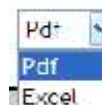
Al hacer click en éste submenú se despliega la página de no enviados.



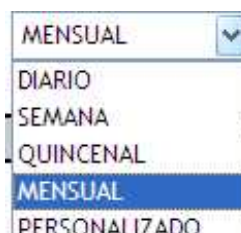
Figura 6.17 Buzón de No enviados

En donde tenemos lo siguiente:

- **Reporte en:** Es una lista desplegable en donde podrá seleccionar el formato en que usted puede exportar la información en PDF o EXCEL.



- **Visualización:** Es una lista desplegable en donde podrá seleccionar la forma en que desea visualizar la información.



Además tenemos los botones:

- **Botón Exportar:**

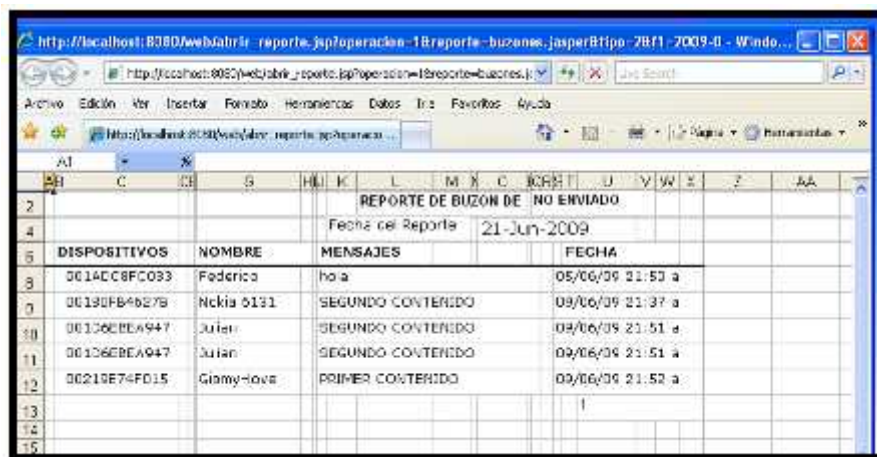


Si usted selecciona el formato PDF la información se le presentará de la siguiente manera:

DISPOSITIVOS	NOMBRE	MENSAJES	FECHA
001ADC8FC033	Federico	hola	09/09/09 09:50 PM
00180FB4627B	Nokia 6131	SEGUNDO CONTENIDO	09/09/09 09:37 PM
001D6EBA947	Julian	SEGUNDO CONTENIDO	09/09/09 09:51 PM
001D6EBA947	Julian	SEGUNDO CONTENIDO	09/09/09 09:51 PM
00219E74F015	Glomy-love	PRIMER CONTENIDO	09/09/09 09:52 PM

Figura 6.18 Exportación Buzón de no enviados a PDF

Si usted selecciona el formato EXCEL la información se le presentará de la siguiente manera:



REPORTE DE BUZÓN DE NO ENVIADO			
Fecha del Reporte: 21-Jun-2009			
DISPOSITIVOS	NOMBRE	MENSAJES	FECHA
00140C8FC033	Federico	hola	05/06/09 21:53 a
00130FB46278	Nckia 0131	SEGUNDO CONTENIDO	09/06/09 21:37 a
001360EE6947	Julian	SEGUNDO CONTENIDO	09/06/09 21:51 a
001360EE6947	Julian	SEGUNDO CONTENIDO	09/06/09 21:51 a
00210E74F015	Glamy-love	PRIMER CONTENIDO	09/06/09 21:52 a

Figura 6.19 Exportación Buzón de no enviados a EXCEL

Si usted hace click en este botón y no ha ninguna búsqueda, el sistema le enviará el siguiente mensaje:



- **Botón Imprimir:**



Al hacer click en este botón le presentará una ventana en donde podrá configurar la impresora como se mostro en la figura 3.6.

Al hacer click en imprimir se despliega una nueva ventana donde usted podrá visualizar la información y proceder a imprimir.

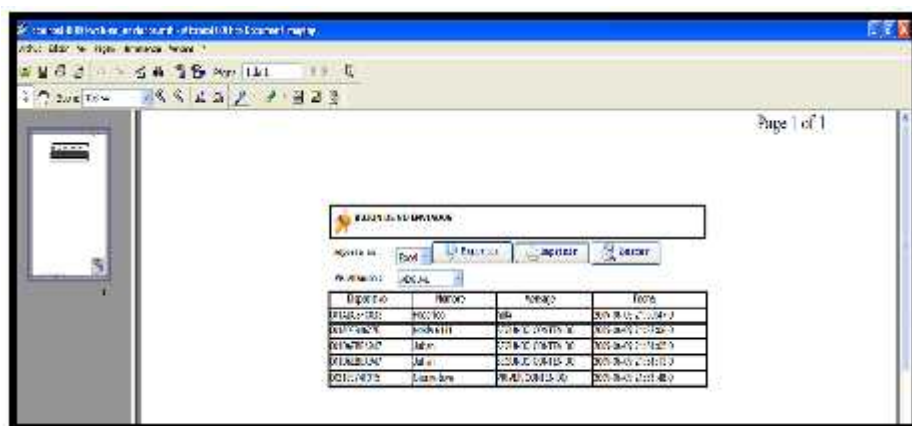


Figura 6.20 Impresión de buzón de no enviados

Si usted hace click en éste botón y no ha realizado ninguna búsqueda el sistema le presentará el siguiente mensaje:



- **Botón Buscar:**



Al hacer click en este botón podrá visualizar la información de acuerdo a lo que usted seleccionó en la lista desplegable de visualización.



Figura 6.21 Búsqueda semanal no enviados

6.2.4. Borradores:

Al hacer click en éste submenú se despliega la página de Borradores.



Figura 6.22 Borradores

En donde tenemos los siguientes botones:

- **Botón Ingresar:**



Al hacer click en éste botón se despliega la página de ingreso de borrador.

 A screenshot of a web application interface. At the top, there's a header with a Bluetooth logo and the text 'SERVIDOR BLUETOOTH'. Below the header, a navigation bar contains links: 'Principal', 'Usuarios', 'Privilegios', 'Envíos', 'Reportes', and '12 Peticiones'. The main content area is titled 'INGRESO DE BORRADORES'. It contains two input fields: 'Titulo:' and 'Contenido:'. Below these fields are three buttons: 'Guardar', 'Consultar', and 'Limpiar'.

Figura 6.23 Página de ingreso de borrador

En donde tenemos lo siguiente:

- **Título:** Es un campo de alfanumérico en donde se ingresará el título del borrador.

Por ejemplo:

 A form element showing the label 'Título:' followed by a text input field. The input field contains the word 'URGENTE' in red capital letters.

En el caso que usted no ingrese ningún dato en este campo y hace click en el botón guardar el sistema le enviará el siguiente mensaje:



- **Contenido:** En un campo alfanumérico donde se ingresará la plantilla del mensaje.

Por ejemplo:

Contenido :

Se cita con caracter de
urgente a

En el caso que usted no ingrese ningún dato en este campo y hace click en el botón guardar el sistema le enviará el siguiente mensaje:



- **Botón Guardar:**



Al hacer clic en este botón, si usted no ha ingresado ningún valor el sistema le pedirá que ingrese el título del borrador como primera información necesaria para su registro y luego continuar con el proceso de ingreso de un borrador.

Una vez que ha digitado los datos, al hacer click en este botón el sistema le enviará el siguiente mensaje de confirmación:



Al aceptar, el sistema le enviará el siguiente mensaje en el que indica que el borrador ha sido guardado.



- **Botón Consultar:**



Al hacer click en este botón, se despliega la página de consulta de borradores.

- **Botón Limpiar:**



Al hacer click en este botón, podrá borrar todo lo que usted ha ingresado para volver a digitar datos.

• **Botón Modificar:**



Si usted no ha seleccionado ningún borrador, el sistema le enviará el siguiente mensaje:



Una vez seleccionado el borrador a modificar, el sistema le envía el siguiente mensaje de confirmación:



Al aceptar, se despliega la página de modificación de borrador:



Figura 6.24 Modificación de borradores

En donde tenemos los siguientes botones:

- **Botón Guardar:**



Al hacer click en éste botón el sistema le enviará el siguiente mensaje de confirmación si usted desea hacer el cambio.



Al aceptar el sistema le enviará el siguiente mensaje de que se ha hecho la modificación.



- **Botón Consultar:**



Al hacer click en este botón, se despliega la página de consulta de borradores.

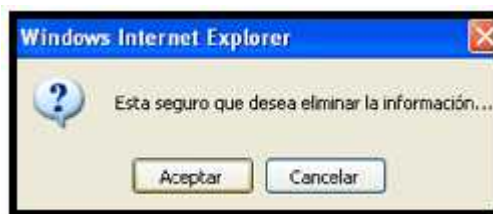
• **Botón Eliminar:**



Si usted no ha seleccionado ningún borrador, el sistema le enviará el siguiente mensaje:



Una vez seleccionado el borrador a eliminar, el sistema le envía el siguiente mensaje de confirmación:



Al aceptar el sistema le enviará el siguiente mensaje indicando que la eliminación del registro se ha realizado satisfactoriamente.

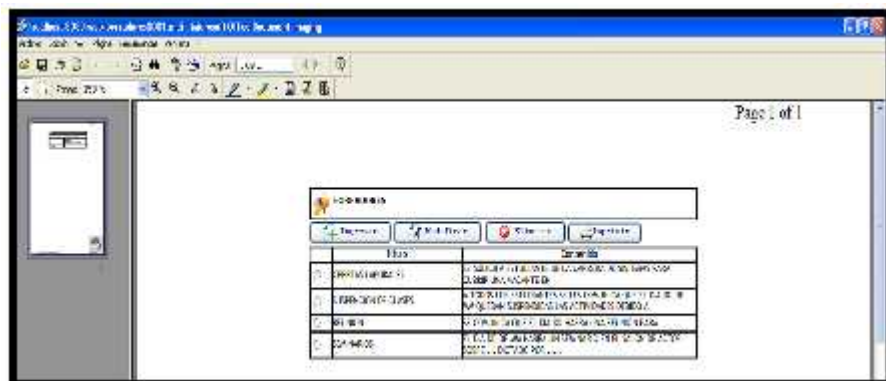


- **Botón Imprimir:**



Al hacer click en este botón le presentará una ventana en donde podrá configurar la impresora como se mostro en la figura 3.6.

Al hacer click en imprimir se despliega una nueva ventana donde usted podrá visualizar la información y proceder a imprimir.



Page 1 of 1

Id	Descripción	Fecha	Usuario
1	borrador de mensaje	2010-01-01 10:00:00	admin
2	borrador de mensaje	2010-01-01 10:00:00	admin
3	borrador de mensaje	2010-01-01 10:00:00	admin
4	borrador de mensaje	2010-01-01 10:00:00	admin
5	borrador de mensaje	2010-01-01 10:00:00	admin

Figura 6.25 Impresión de borradores

6.2.5. Contador:

Al hacer click en éste submenú se despliega la página de Contador de mensajes.

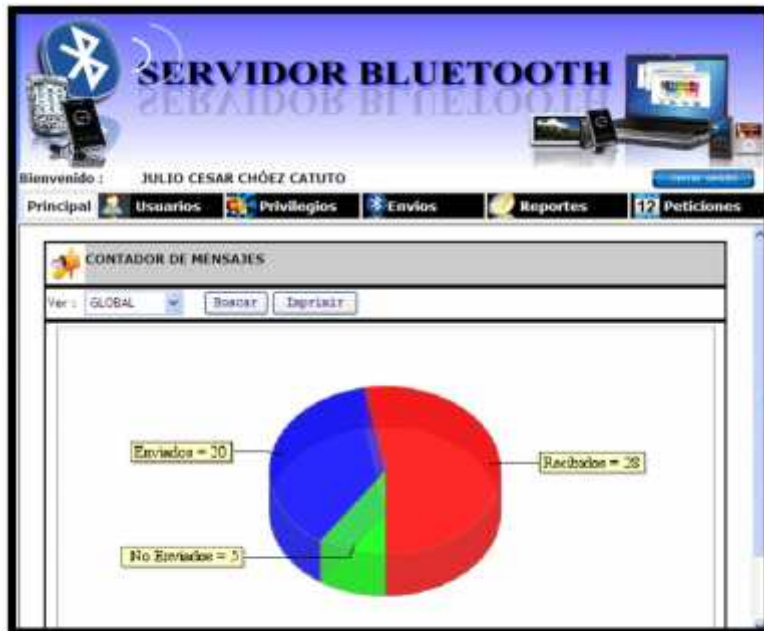
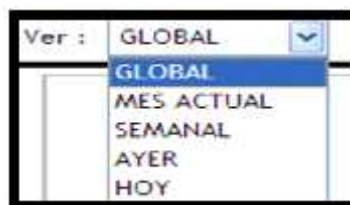


Figura 6.26 Contador de Mensajes

En donde tenemos lo siguiente:

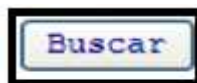
- **Global:** Es una lista desplegable en donde podemos escoger cómo deseamos ver la información.

Por ejemplo:



Y además los siguientes botones:

- **Botón Buscar:**



Al hacer click en éste botón se actualizará el gráfico dependiendo de la opción en Ver que usted haya elegido.

- **Botón Imprimir:**



Al hacer click en este botón le presentará una ventana en donde podrá configurar la impresora como se mostro en la figura 3.6.

Al hacer click en imprimir se despliega una nueva ventana donde usted podrá visualizar la información y proceder a imprimir.

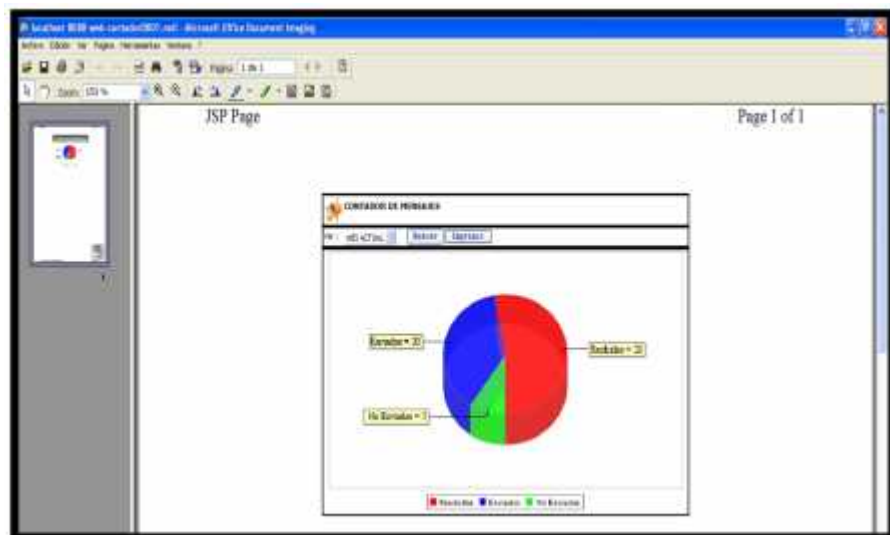


Figura 6.27 Impresión de contador de mensajes

6.3. Auditoria :

Al hacer click en éste submenú se despliega la página de Reporte de Auditoría.

Figure 6.28 shows the 'AUDITORIA DE USUARIOS' report. The report displays a table with columns: Nombres, Apellidos, Usuario, Fecha, and Acción. The table lists 10 entries for 'JULIO CESAR CHÓEZ CATUTO' with the username 'admin', showing various system actions like 'INGRESO AL SISTEMA' and 'SALIO DEL SISTEMA' with timestamps.

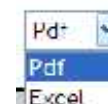
Nombres	Apellidos	Usuario	Fecha	Acción
JULIO CESAR	CHÓEZ CATUTO	admin	2009-06-05 21:03:29	INGRESO AL SISTEMA
JULIO CESAR	CHÓEZ CATUTO	admin	2009-06-05 21:48:39	INGRESO AL SISTEMA
JULIO CESAR	CHÓEZ CATUTO	admin	2009-06-05 21:52:53	SALIO DEL SISTEMA
JULIO CESAR	CHÓEZ CATUTO	admin	2009-06-05 21:53:00	INGRESO AL SISTEMA
JULIO CESAR	CHÓEZ CATUTO	admin	2009-06-05 22:33:18	SALIO DEL SISTEMA
JULIO CESAR	CHÓEZ CATUTO	admin	2009-06-05 22:41:50	INGRESO AL SISTEMA
JULIO CESAR	CHÓEZ CATUTO	admin	2009-06-06 18:24:24	INGRESO AL SISTEMA
JULIO CESAR	CHÓEZ CATUTO	admin	2009-06-06 18:26:00	SALIO DEL SISTEMA
JULIO CESAR	CHÓEZ CATUTO	admin	2009-06-06 18:26:20	INGRESO AL SISTEMA
JULIO CESAR	CHÓEZ CATUTO	admin	2009-06-06 18:28:45	SALIO DEL SISTEMA
JULIO CESAR	CHÓEZ CATUTO	admin	2009-06-06 18:28:59	INGRESO AL SISTEMA

Figura 6.28 Página de Reporte de Auditoría

En donde tenemos lo siguiente:

- **Reporte en:**

Es una lista desplegable en donde podrá seleccionar el formato en que usted puede exportar la información en PDF o EXCEL.



Y además los siguientes botones:

- **Botón Exportar:**



Si usted selecciona el formato PDF y hace click en éste botón la información se le presentará de la siguiente manera:

USUARIO	NOMBRES	APELLIDOS	FECHA	ACCION
admin	JULIO CESAR	CHÓEZ CATUTO	05/06/09 09:03	INGRESO AL SISTEMA
admin	JULIO CESAR	CHÓEZ CATUTO	05/06/09 09:48	INGRESO AL SISTEMA
admin	JULIO CESAR	CHÓEZ CATUTO	05/06/09 09:52	SALIO DEL SISTEMA
admin	JULIO CESAR	CHÓEZ CATUTO	05/06/09 09:53	INGRESO AL SISTEMA
admin	JULIO CESAR	CHÓEZ CATUTO	05/06/09 10:33	SALIO DEL SISTEMA
admin	JULIO CESAR	CHÓEZ CATUTO	05/06/09 10:41	INGRESO AL SISTEMA
admin	JULIO CESAR	CHÓEZ CATUTO	06/06/09 08:24	INGRESO AL SISTEMA
admin	JULIO CESAR	CHÓEZ CATUTO	06/06/09 08:26	SALIO DEL SISTEMA
admin	JULIO CESAR	CHÓEZ CATUTO	06/06/09 08:26	INGRESO AL SISTEMA
admin	JULIO CESAR	CHÓEZ CATUTO	06/06/09 08:28	SALIO DEL SISTEMA
admin	JULIO CESAR	CHÓEZ CATUTO	06/06/09 08:28	INGRESO AL SISTEMA
admin	JULIO CESAR	CHÓEZ CATUTO	06/06/09 08:33	SALIO DEL SISTEMA
admin	JULIO CESAR	CHÓEZ CATUTO	06/06/09 08:34	INGRESO AL SISTEMA
admin	JULIO CESAR	CHÓEZ CATUTO	06/06/09 08:36	SALIO DEL SISTEMA
admin	JULIO CESAR	CHÓEZ CATUTO	06/06/09 08:36	INGRESO AL SISTEMA
admin	JULIO CESAR	CHÓEZ CATUTO	06/06/09 07:05	INGRESO AL SISTEMA
admin	JULIO CESAR	CHÓEZ CATUTO	06/06/09 08:59	INGRESO AL SISTEMA
admin	JULIO CESAR	CHÓEZ CATUTO	06/06/09 03:43	INGRESO AL SISTEMA
admin	JULIO CESAR	CHÓEZ CATUTO	06/06/09 04:39	INGRESO AL SISTEMA

Figura 6.29 Exportación Reporte de Auditoría a PDF

Si usted selecciona el formato EXCEL la información se le presentará de la siguiente manera:

	C	E	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC
2																									
7	admin	JULIO CESAR	CHÓEZ CATUTO	05/06/09																					
12	admin	JULIO CESAR	CHÓEZ CATUTO	05/06/09																					
17	admin	JULIO CESAR	CHÓEZ CATUTO	05/06/09																					
23	admin	JULIO CESAR	CHÓEZ CATUTO	05/06/09																					
27	admin	JULIO CESAR	CHÓEZ CATUTO	05/06/09																					
33	admin	JULIO CESAR	CHÓEZ CATUTO	05/06/09																					
37	admin	JULIO CESAR	CHÓEZ CATUTO	06/06/09																					
42	admin	JULIO CESAR	CHÓEZ CATUTO	06/06/09																					
47	admin	JULIO CESAR	CHÓEZ CATUTO	06/06/09																					
52	admin	JULIO CESAR	CHÓEZ CATUTO	06/06/09																					
57	admin	JULIO CESAR	CHÓEZ CATUTO	06/06/09																					
62	admin	JULIO CESAR	CHÓEZ CATUTO	06/06/09																					
67	admin	JULIO CESAR	CHÓEZ CATUTO	06/06/09																					
72	admin	JULIO CESAR	CHÓEZ CATUTO	06/06/09																					
77	admin	JULIO CESAR	CHÓEZ CATUTO	06/06/09																					
83	admin	JULIO CESAR	CHÓEZ CATUTO	06/06/09																					
87	admin	JULIO CESAR	CHÓEZ CATUTO	06/06/09																					
92	admin	JULIO CESAR	CHÓEZ CATUTO	06/06/09																					
97	admin	JULIO CESAR	CHÓEZ CATUTO	06/06/09																					
102	admin	JULIO CESAR	CHÓEZ CATUTO	06/06/09																					
107	admin	JULIO CESAR	CHÓEZ CATUTO	06/06/09																					
112	admin	JULIO CESAR	CHÓEZ CATUTO	06/06/09																					

Figura 6.30 Exportación de Reporte de Auditoría a EXCEL



- **Botón Imprimir:**








Al hacer click en este botón le presentará una ventana en donde podrá configurar la impresora como se mostro en la figura 3.6.

Al hacer click en imprimir se despliega una nueva ventana donde usted podrá visualizar la información y proceder a imprimir.

6.4. Más usos:

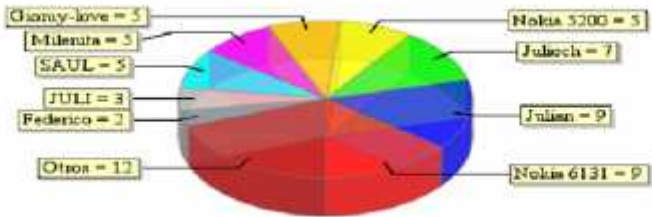



Bienvenido : **JULIO CESAR CHÓEZ CATUTO**

Principal  Usuarios  Privilegios  Envíos  Reportes  12 Peticiones

REPORTE DE LOS DISPOSITIVOS MÁS USADOS

[Imprimir](#)



Dispositivo	Usos
Nokia 5200	5
Jubech	7
Julian	9
Nokia 6131	9
Otros	12
Federico	2
JULI	3
SAUL	5
Milenra	5
Clony-love	5

Figura 6.32 Página de los dispositivos más usados

En donde tenemos el siguiente botón:

- **Botón Imprimir:**

Al hacer click en este botón le presentará una ventana en donde podrá configurar la impresora como se mostro en la figura 3.6.

Al hacer click en imprimir se despliega una nueva ventana donde usted podrá visualizar la información y proceder a imprimir.

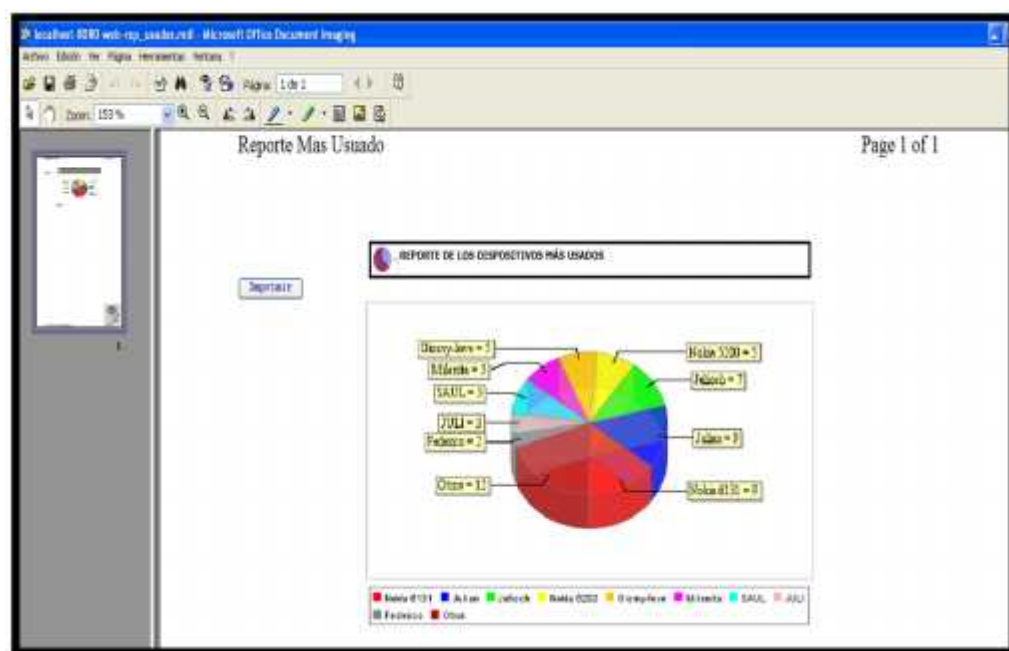


Figura 6.33 Impresión de los dispositivos más usados

7. INGRESO A PETICIONES

Haciendo click en Peticiones se despliega los submenús que facilitará al administrador ingresar y consultar las peticiones que podrán hacer los estudiantes desde su celular.



Figura 7.1 Menú peticiones

En el menú principal de Peticiones aparecen los sub menú los cuales son:

7.1. Ingresar:

Haciendo click en Ingresar se despliega la página de ingreso de peticiones.

Figura 7.2 Página de ingreso de peticiones

En esta página ingresaremos los siguientes datos:

- **Petición:** Es un campo alfanumérico, donde se ingresará el nombre de la petición.

Por ejemplo:

Petición:

Si usted no ingresa ningún valor en éste campo y hace click en el botón guardar el sistema le enviará el siguiente mensaje:



En el caso que usted ingrese dos o más palabras y hace click en el botón guardar el sistema le enviará el siguiente mensaje:



- **Argumento:**

Es un campo alfanumérico, donde se ingresará el argumento de la petición.

Por ejemplo:

Argumento:

Si usted no ingresa ningún valor en éste campo y hace click en el botón guardar el sistema le enviará el siguiente mensaje:



En el caso que usted ingrese dos o más palabras y hace click en el botón guardar el sistema le enviará el siguiente mensaje:



- **Respuestas:** Es un campo alfanumérico, donde se ingresará la respuesta de la petición.

Por ejemplo:

Respuestas:

Si usted no ingresa ningún valor en éste campo y hace click en el botón guardar el sistema le enviará el siguiente mensaje:



Además observamos los siguientes botones:

- **Botón Guardar:**



Al hacer click en este botón, y si usted no ha digitado ninguna información el sistema le enviará un mensaje requiriendo el ingreso de la petición como primer dato importante a ingresar para que luego continúe con el proceso de ingreso de una petición.

Una vez que haya llenado todos los campos, al hacer click en éste botón el sistema le envía el siguiente mensaje de confirmación:



Al aceptar, el sistema le envía un mensaje en el que le notifica el ingreso de la petición. Y se despliega la página de consulta para que usted confirme que se ha realizado el ingreso de la petición.



- **Botón Limpiar:**



Al hacer click en este botón, podrá borrar todo lo que usted ha ingresado para volver a digitar datos.

7.2. Consultar:

Haciendo click en Consultar se despliega la página de Consulta de peticiones.



Figura 7.3 Página de consulta de peticiones

En donde tenemos los siguientes botones:

- **Botón modificar:**



Antes de hacer click en este botón debe primero seleccionar la petición a modificar caso contrario le aparecerá el siguiente mensaje:



Una vez seleccionado haga click en este botón el cual le permite modificar los datos del usuario, el sistema le presentará un mensaje para verificar si usted en realidad desea modificar los datos:



Al aceptar se despliega la página en donde usted podrá modificar la petición:



Figura 7.4 Modificación de una petición

Como podemos observar en esta página usted podrá modificar cualquier dato del usuario del sistema, cada campo será validado como se explicó en el ingreso de un usuario.

Tenemos dos botones que son:

- **Modificar:**



Al hacer click en este botón le aparecerá el siguiente mensaje.



Al aceptar el sistema le enviará un mensaje indicando que la modificación se ha realizado con éxito.



- **Consultar:**



Al hacer click en éste botón, irá a la página de consulta de las peticiones.

- **Botón eliminar:**



Antes de hacer click en este botón debe primero seleccionar la petición a eliminar caso contrario el sistema le presentará el siguiente mensaje:



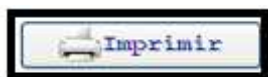
Una vez seleccionada la petición, al hacer click en el botón eliminar le aparecerá el siguiente mensaje para confirmar que desea realizar la acción:



Al aceptar, el sistema le presente el mensaje de confirmación de que la eliminación de la petición ha sido realizada:



- **Botón Imprimir:**



Al hacer click en este botón le presentará una ventana en donde podrá configurar la impresora como se mostro en la figura 3.6.

Al hacer click en imprimir se despliega una nueva ventana donde usted podrá visualizar la información y proceder a imprimir.

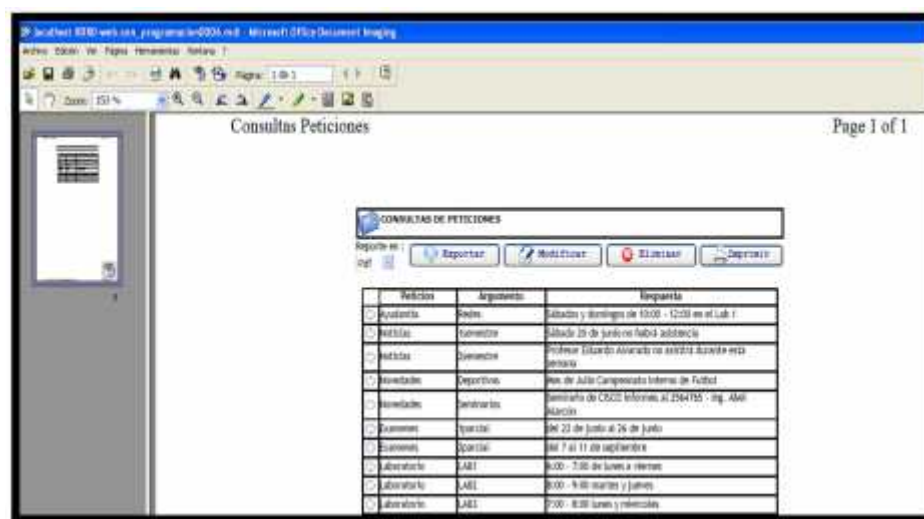
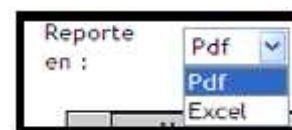


Figura 7.5 Impresión de peticiones

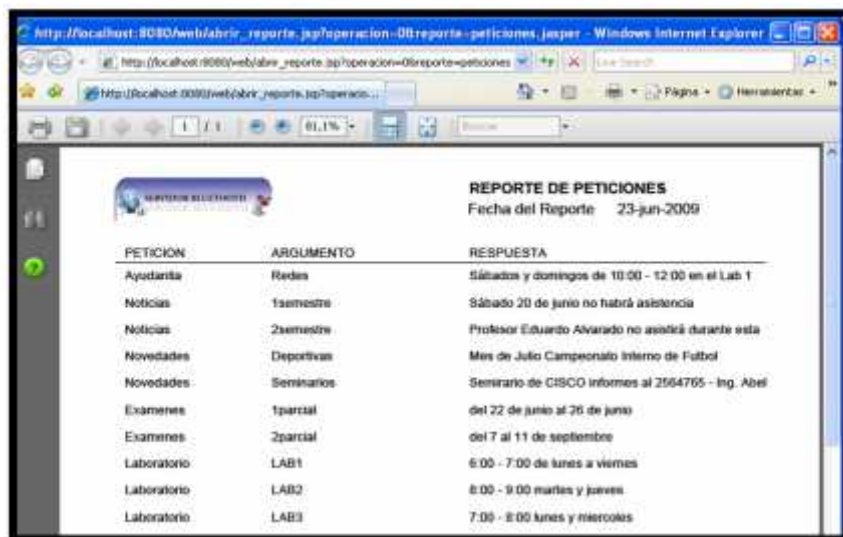
- **Botón Exportar:**



Este botón le permite exportar la información en las siguientes opciones:



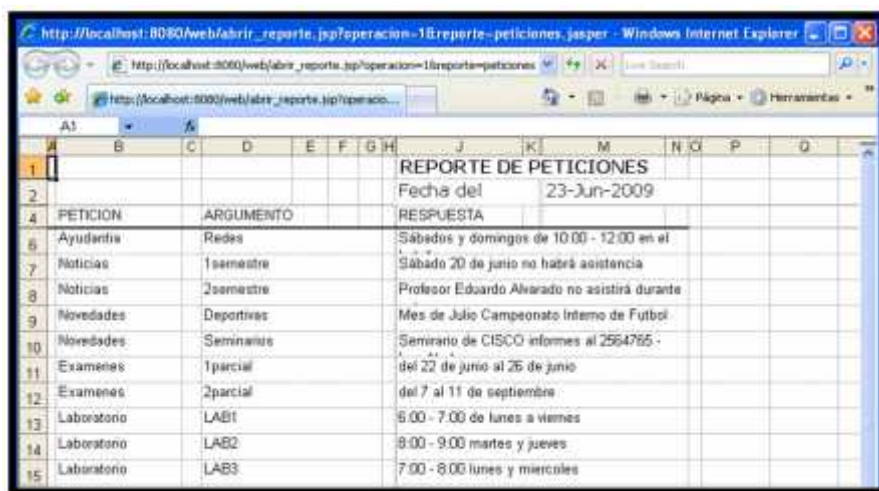
Si usted selecciona el formato PDF la información se le presentará de la siguiente manera:



PETICIÓN	ARGUMENTO	RESPUESTA
Ayudantía	Redes	Sábados y domingos de 10:00 - 12:00 en el Lab 1
Noticias	1 semestre	Sábado 20 de junio no habrá asistencia
Noticias	2 semestre	Profesor Eduardo Alvarado no asistirá durante esta
Novedades	Deportivas	Mes de Julio Campeonato Interno de Fútbol
Novedades	Seminarios	Seminario de CISCO informes al 2564765 - Ing. Abel
Exámenes	1 parcial	del 22 de junio al 26 de junio
Exámenes	2 parcial	del 7 al 11 de septiembre
Laboratorio	LAB1	6:00 - 7:00 de lunes a viernes
Laboratorio	LAB2	8:00 - 9:00 martes y jueves
Laboratorio	LAB3	7:00 - 8:00 lunes y miércoles

Figura 7.6 Exportación de Reporte de peticiones a PDF

Si usted selecciona el formato EXCEL la información se le presentará de la siguiente manera:



PETICIÓN	ARGUMENTO	RESPUESTA
Ayudantía	Redes	Sábados y domingos de 10:00 - 12:00 en el
Noticias	1 semestre	Sábado 20 de junio no habrá asistencia
Noticias	2 semestre	Profesor Eduardo Alvarado no asistirá durante
Novedades	Deportivos	Mes de Julio Campeonato Interno de Fútbol
Novedades	Seminarios	Seminario de CISCO informes al 2564765 -
Exámenes	1 parcial	del 22 de junio al 26 de junio
Exámenes	2 parcial	del 7 al 11 de septiembre
Laboratorio	LAB1	6:00 - 7:00 de lunes a viernes
Laboratorio	LAB2	8:00 - 9:00 martes y jueves
Laboratorio	LAB3	7:00 - 8:00 lunes y miércoles

Figura 7.7 Exportación de Reporte de peticiones a EXCEL

8. INSTRUCTIVO SERVIDOR BLUETOOTH

Para explicar en detalle el funcionamiento de las respuestas del servidor y cómo hacer las consultas, utilizaremos como ejemplo un celular modelo Nokia E51.

8.1. ENVÍO DE NOTAS, MATERIAS

El formato para que el estudiante pueda enviar una petición de consulta de notas es la siguiente:

- **[numero_telefono].notas:** Si desea saber todas las notas de manera general que usted está cursando.
- **[numero_telefono].notas.[codigo_materia]:** Si desea saber una nota de una materia específica.
- **[numero_telefono].materias:** Para conocer cuáles son las materias en las que usted se encuentra inscrito seguido de su código y el curso al que pertenece.

8.2. ENVÍO DE ACTUALIZACIÓN DE NUMERO CELULAR

Para realizar ésta petición usted puede digitar lo siguiente:

Numero_cedula_estudiante.[nuevo_numero_celular]

8.3. ENVÍO DE PETICIONES PROGRAMADAS:

Dependiendo de la base de datos usted podrá enviar una nota con el siguiente formato:

Numero_celular.peticion.argumento

8.4. RECEPCIÓN DE RESPUESTAS

Una vez que usted haya enviado las peticiones el sistema primero le enviará una confirmación si usted desea recibir un mensaje desde el servidor como por ejemplo:

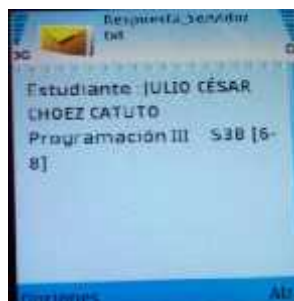


Dependiendo de la petición, usted recibirá:

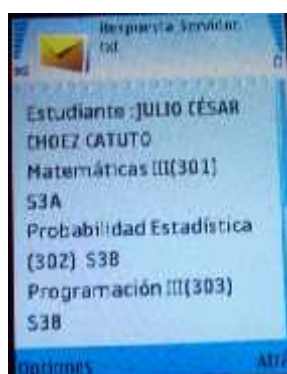
En el caso de la petición de notas de manera general usted recibirá lo siguiente:



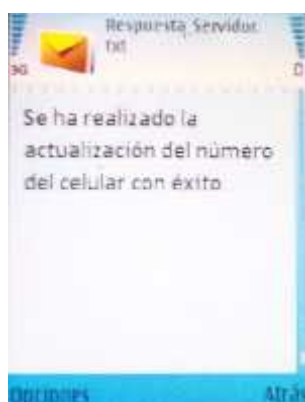
En el caso de la petición de una nota específica usted recibirá lo siguiente:



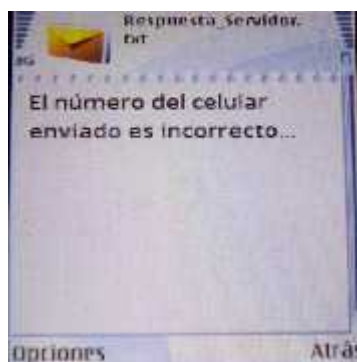
En el caso de la petición de materia para conocer el código y el curso en la que se encuentra registrado usted recibirá lo siguiente:



En el caso de actualizar su número celular usted recibirá lo siguiente:



En el caso de enviar su número incompleto el sistema le enviará el siguiente mensaje:



En el caso de solicitar una petición que se encuentren en la base de datos usted recibirá lo siguiente:

Por ejemplo: 097344226.ayudantia.redes nos retornará:





CAPÍTULO 2

MANUAL TÉCNICO

SERVIDOR BLUETOOTH

El manual técnico es el código fuente y documentado del sistema, además de las especificaciones del lenguaje y librerías usadas para su creación, además de los requerimientos necesarios para su correcta ejecución.

El sistema SERVIDOR BLUETOOTH se divide en dos aplicaciones para el cumplimiento de sus funcionalidades:

- BLUETOOTH APLICACIÓN.
- WEB BLUETOOTH.

BLUETOOTH APLICACIÓN

Esta aplicación correrá internamente con o sin la interfaz web bluetooth y tendrá un icono en la barra de estado y al dar click derecho saldrá un menú que contará con opciones adicionales que serán por el usuario; las opciones que maneja el icono “try icon” son las siguientes:

- Start servicio de envío.
- Stop servicio de envío.
- Stop servicio detección de dispositivos.
- Start servicio detección de dispositivos.
- Activar envío publicidad.
- Desactivar envío publicidad.
- Exit.

Para conseguir que la aplicación haga varias cosas a la vez o que el programa no se quede parado mientras realiza una tarea compleja, se utilizan lo que se llaman **hilos** para realice simultáneamente los procesos

APLICACIÓN BLUETOOTH maneja tres hilos; el primero permite el envío de mensajes al dispositivo con tecnología bluetooth (celular, laptop, palm); el segundo hilo permite detectar los dispositivos que están al alcance y el tercer hilo permite enviar la publicidad. Los hilos no consumen muchos recursos de la maquina debido al funcionamiento que otorga los hilos.

WEB BLUETOOTH

Esta interfaz presenta el sistema en si con el ingreso del usuario y contraseña y dependiendo de los permisos que el usuario administrador le otorgue el usuario final visualizará las siguientes opciones:

- Ingreso y consulta de usuarios del sistema.
- Ingreso y consulta de usuarios externos.
- Ingreso y consulta de privilegios.
- Envio de mensajes y publicidad.
- Reportes de dispositivos encontrados.
- Reportes de buzones de entrada, salida, no recibidos, borradores y contador.
- Reporte de auditoria.
- Ingreso y consulta de peticiones.

Para el correcto funcionamiento de aplicación bluetooth y web bluetooth detallamos las características de cada uno de los aplicativos que se necesita para levantar la aplicación, su instalación y configuración:

- Sistema operativo bajo ambiente Windows
- Java 6.0 maquina virtual
- Netbeant version 6.5
- Base de Datos MYSQL 5.0 – Administrador de la Base de Datos
- Ireport
- Librerías independientes como:
 - Jfreechart
 - Bluecove
 - Jdom
 - Itext
 - Jsonrpc
 - Mysqlconnector
 - Cos
 - Jcoomm
 - Poi

Procesos críticos

Una de las funciones principales y por ende el proceso más crítico es la detección de dispositivos bluetooth sin este proceso no funcionaria el sistema, ya que es primordial para el uso de la aplicación.

El segundo proceso más crítico es el envío y recepción de peticiones esta parte interactúa con los dispositivos previamente detectados y hace la comunicación entre el sistema y los dispositivos con tecnología bluetooth. Recibe las peticiones las verifica si proviene de un usuario externo con el dispositivo registrado y este recibirá la respuesta que el servidor devuelve como resultado.

Un tercer proceso crítico es en envío de la publicidad, ya que esta parte estará constantemente solicitando la búsqueda de dispositivos que están en su alcance y enviará la publicidad conforme vaya detectando dispositivos bluetooth.

PSEUDOCODIGO

El código a continuación es un modelo inicial de lo que es realmente el contenido de la aplicación, esta codificación se encuentra sujeto a cambios en caso de que en el sistema requiera otras funciones, alternativas o posibilidades que mejoren el proceso de implementación, así mejorar con la aplicación y lo que el sistema pueda realizar.

Detallaremos a continuación las clases que conforman las aplicaciones realizadas para el desarrollo de este proyecto.

1. CLASES

Esta clase se utiliza para realizar la conexión y desconexión con la base de datos, así como también la búsqueda y almacenamiento de dispositivos bluetooth.

1.1 CLASE “Conexión.java”

Es la encargada de realizar la conexión con la base de datos para su respectivo mantenimiento. Con funciones y procedimientos que ayudan a las demás clases. Es una de las clases más importantes del sistema.

```

package Clases;
/*
 * Base.java
 *
 * Created on 6 de junio de 2008, 20:20
 */
/**
 *
 * author grupo 13a
 */
import java.sql.*;
import javax.naming.Context;
import javax.naming.InitialContext;
import javax.sql.DataSource;
import javax.naming.NamingException;
import javax.naming.NoInitialContextException;
public class Conexion {
    String error;
    Connection con;
    public Conexion(){}
    public void conectar() throws
ClassNotFoundException,SQLException,Exception{
        try{
            Class.forName("org.gjt.mm.mysql.Driver").newInstance();
con=DriverManager.getConnection("jdbc:mysql://localhost/sms?user=root&p
assword=");}catch(ClassNotFoundException uno ){
    error="No puede localizar la base de datos";
    throw new ClassNotFoundException(error);
}catch(SQLException dos ){
    error="No puede realizar la conexion d la base de datos";
    throw new SQLException(error+dos.toString());
}catch(Exception tres ){
    error="Error desconocido";
    throw new Exception(error+tres.toString()); }
}
public Connection getConexion(){
    return con;}
public void desconectar () throws SQLException{
    try{    if(con!=null){
        if(!con.isClosed()){
            con.close(); }
        }
    }catch(SQLException dos ){
        error="Imposible desconectar la base de datos";
        throw new SQLException(error); }
}
public ResultSet Consultas(String sql) throws SQLException,Exception{
    ResultSet rs=null;
    try{

```

```

Statement stmt=con.createStatement();
rs=stmt.executeQuery(sql);
}catch(SQLException dos ){
    error="No puede ejecutar la consulta";
    throw new SQLException(error);
}catch(Exception tres ){
    error="Generacion de execepcion en la consulat";
    throw new Exception(error);
} return rs;}
public String Consultas_objecto(String sql) throws
SQLException,Exception{
    ResultSet rs=null; String retorna="";
    try{ Statement stmt=con.createStatement();
    rs=stmt.executeQuery(sql);
    while(rs.next()){
        retorna=rs.getString("Objecto");}
    }catch(SQLException dos ){
        error="No puede ejecutar la consulta";
        throw new SQLException(error);
    }catch(Exception tres ){
        error="Generacion de execepcion en la consulat";
        throw new Exception(error); } return retorna; }
public Boolean verificaconsultas(String sql) throws
SQLException,Exception{
    boolean ban=false; ResultSet rs=null;
    try{ Statement stmt=con.createStatement();
    rs=stmt.executeQuery(sql);
    if(rs.next()){ ban=true;}
    }catch(SQLException dos ){
        error="No puede ejecutar la consulta";
        throw new SQLException(error);
    }catch(Exception tres ){
        error="Generacion de execepcion en la consulat";
        throw new Exception(error);}
    return (new Boolean(ban)); }
public void mantenimiento(String sql) throws SQLException,Exception{
    if(con!=null){ try{
        PreparedStatement sentencia;
        sentencia=con.prepareStatement(sql);
        sentencia.execute();}catch(SQLException dos ){
            error="No puede ejecutar la mantenimiento";
            throw new SQLException(error);
        }catch(Exception tres ){
            error="Generacion de execepcion en la mantenimiento";
            throw new Exception(error); }
    }
}
public static void main(String[] args)throws SQLException,Exception {
    Conexion p= new Conexion();
    p.conectar();}

```

Clases

Class Conexion

java.lang.Object

└ **Clases.Conexion**

Subclases directas:

Dispositivos

public class **Conexion**
extends java.lang.Object

Resumen constructor

Conexion()

Resumen de métodos

void	conectar()
java.lang.String	Consultas_objeto (java.lang.String sql)
java.sql.ResultSet	Consultas (java.lang.String sql)
void	desconectar()
static void	main (java.lang.String[] args)
void	mantenimiento (java.lang.String sql)
java.lang.Boolean	verificaconsultas (java.lang.String sql)

Métodos heredados de la clase java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detalle

Conexion

public **Conexion**()

Método Detalle

conectar

public void **conectar**()
throws java.lang.ClassNotFoundException,
java.sql.SQLException,
java.lang.Exception

Throws:

java.lang.ClassNotFoundException
java.sql.SQLException
java.lang.Exception

desconectar

public void **desconectar**()
throws java.sql.SQLException

Throws:

java.sql.SQLException

Consultas

public java.sql.ResultSet **Consultas**(java.lang.String sql)
throws java.sql.SQLException,
java.lang.Exception

Throws:

java.sql.SQLException
java.lang.Exception

Consultas_objecto

public java.lang.String **Consultas_objecto**(java.lang.String sql)
throws java.sql.SQLException,
java.lang.Exception

Throws:

java.sql.SQLException

java.lang.Exception

verificaconsultas

public java.lang.Boolean **verificaconsultas**(java.lang.String sql)

throws java.sql.SQLException,

java.lang.Exception

Throws:

java.sql.SQLException

java.lang.Exception

mantenimiento

public void **mantenimiento**(java.lang.String sql)

throws java.sql.SQLException,

java.lang.Exception

Throws:

java.sql.SQLException

java.lang.Exception

main

public static void **main**(java.lang.String[] args)

throws java.sql.SQLException,

java.lang.Exception

Throws:

java.sql.SQLException

java.lang.Exception

1.2 CLASE “dispositivos.java”

Tiene como fin llevar el mantenimiento de las direcciones de los dispositivos.

Esta clase se encarga de la detectar, ingresar, modificar y consultar Los dispositivos bluetooth que son detectados por el sistema.

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package Clases;
import java.sql.*;
import java.util.Vector;
/**
 *
 * author grupo 13a
 */
public class Dispositivos extends Conexion {
    CallableStatement proceso = null;
    public String devuelve_ruta(String dispositivo){
        String ruta="";
        try{ conectar();
            ruta=Consultas_objecto("select ruta as Objecto from
dispositivos where direccion='"+dispositivo+"'");
            desconectar();
        }catch(java.sql.SQLException dos){
            dos.printStackTrace();
            System.out.println(dos.getMessage());
        }catch(Exception tres ){
            tres.printStackTrace();
            System.out.println(tres.getMessage());}
        return ruta;    }
    public Boolean ingresar_dispositivos(String nombre,String direccion,int
mayor,int menor){ boolean ban=false;
        try{ conectar();
            if(verificaconsultas("select * from dispositivos where
direccion='"+direccion+"'").booleanValue()==false){
                proceso=con.prepareCall("{ Call
MANTENIMIENTO_DISPOSITIVOS(?,?,?)}");
                proceso.setString(1,nombre.toUpperCase());
                proceso.setString(2,direccion);
                proceso.setInt(3,1);
                proceso.execute();
                proceso.close();
            }
        }
    }
}

```

```

        mantenimiento("update dispositivos set
mayor_class="+String.valueOf(mayor)+" ,minor_class="+String.valueOf(menor)
r)+" where direccion='"+direccion+"'");
        desconectar(); ban=true;
    } catch (java.sql.SQLException dos) {
        dos.printStackTrace();
        System.out.println(dos.getMessage());
    } catch (Exception tres) {
        tres.printStackTrace();
        System.out.println(tres.getMessage());
    }
    return (new Boolean(ban));
}

public void uso_dispositivos(String nombre,String direccion){
    try{ conectar();
        direccion=Consultas_objeto("select direccion as objeto from
dispositivos where ruta='"+direccion+"'");
        proceso=con.prepareStatement(" { Call MANTENIMIENTO_DISPOSITIVOS(?,?,?) }");
        proceso.setString(1,"");
        proceso.setString(2,direccion);
        proceso.setInt(3,2);
        proceso.execute();
        proceso.close();
        desconectar();
    } catch (java.sql.SQLException dos) {
        dos.printStackTrace();
        System.out.println(dos.getMessage());
    } catch (Exception tres) { tres.printStackTrace();
        System.out.println(tres.getMessage());
    }
}

public void registra_mensaje(String direccion,String mensaje, String
tipo){
    try{ conectar();
        direccion=Consultas_objeto("select direccion as objeto from
dispositivos where ruta='"+direccion+"'");
        mantenimiento("insert into mensajes
(mensaje,dispositivos,fecha,tipo)values('"+mensaje.toUpperCase()+"','"+
direccion+"',now(),'"+tipo+"')");
        mantenimiento("update dispositivos set numeros=numeros+1
where direccion='"+direccion+"'");
        desconectar();
    } catch (java.sql.SQLException dos) {
        dos.printStackTrace();
        System.out.println(dos.getMessage());
    } catch (Exception tres) {
        tres.printStackTrace();
        System.out.println(tres.getMessage());
    }
}

public String formatea(String direccion){
    String dir="";
    String h="";

```

```

        for(int i=0;i<direccion.length();i++){
            if((i%2)==0 && i>0){
                dir=dir.concat(":");
                dir=dir.concat(String.valueOf(direccion.charAt(i)));
            }else{ dir=dir.concat(String.valueOf(direccion.charAt(i))); }
        } return dir; }
public Vector Consulta_dispositivos(){
    Vector arreglo =new Vector();
    try{ conectar();
        proceso=con.prepareCall("{ Call VISTAS(?,?)}");
        proceso.setInt(1,15);
        proceso.setString(2,"");
        proceso.execute();
        while(proceso.getResultSet().next()){
            Vector objeto=new Vector();
            objeto.addElement(proceso.getResultSet().getString("nombre"));
            objeto.addElement(formatea(proceso.getResultSet().getString("direccion"
            )));
            objeto.addElement(proceso.getResultSet().getString("fecha_dectecion"));
            objeto.addElement(proceso.getResultSet().getString("fecha_acceso"));
            objeto.addElement(proceso.getResultSet().getString("numeros"));
            objeto.addElement(proceso.getResultSet().getString("mayor_class"));
            objeto.addElement(proceso.getResultSet().getString("minor_class"));
            objeto.addElement(proceso.getResultSet().getString("ruta"));
            arreglo.addElement(objeto);        }
        proceso.close();
        desconectar();
    }catch(java.sql.SQLException dos){
        dos.printStackTrace();
        System.out.println(dos.getMessage());
    }catch(Exception tres ){
        tres.printStackTrace();
        System.out.println(tres.getMessage());    }
    return arreglo; }

public String Consulta_Otros(String sql){
    String retorna="0";
    String valor="";
    try{
        conectar();
        valor=Consultas_objeto(sql);
        if(valor.length()>0){
            retorna=valor;        }
        desconectar();
    }catch(java.sql.SQLException dos){
        dos.printStackTrace();
        System.out.println(dos.getMessage());
    }catch(Exception tres ){
        tres.printStackTrace();
    }
}

```

```

        System.out.println(tres.getMessage());
    }
    return retorna;
}
public Vector Consulta_acceso(){
    Vector arreglo =new Vector();
    try{
        conectar();
        proceso=con.prepareCall("{ Call VISTAS(?,?)}");
        proceso.setInt(1,14);
        proceso.setString(2,"");
        proceso.execute();
        while(proceso.getResultSet().next()){
            Vector objeto=new Vector();
            objeto.addElement(proceso.getResultSet().getString("usuario"));
            objeto.addElement(formatea(proceso.getResultSet().getString("fecha")));
            objeto.addElement(proceso.getResultSet().getString("tipo"));
            objeto.addElement(proceso.getResultSet().getString("nombres"));
            objeto.addElement(proceso.getResultSet().getString("apellidos"));
            arreglo.addElement(objeto);
        }
        proceso.close();
        desconectar();
    }catch(java.sql.SQLException dos){
        dos.printStackTrace();
        System.out.println(dos.getMessage());
    }catch(Exception tres ){
        tres.printStackTrace();
        System.out.println(tres.getMessage());
    }
    return arreglo;
}
public Vector Consulta_mensajes(){
    Vector arreglo =new Vector();
    int i=0;
    String tipo="";
    try{
        conectar();
        for(i=0;i<=2;i++){
            proceso=con.prepareCall("{ Call VISTAS(?,?)}");
            proceso.setInt(1,12);
            proceso.setString(2,String.valueOf(i));
            proceso.execute();
            while(proceso.getResultSet().next()){
                Vector objeto=new Vector();
                objeto.addElement(proceso.getResultSet().getString("numero"));
                if(i==0){
                    tipo="Recibidos";
                }
                if(i==1){ tipo="Enviados";
                }
                if(i==2){tipo=" No Enviados";
                }
                objeto.addElement(tipo);
                arreglo.addElement(objeto);
            }
            proceso.close();
        }
        desconectar();
    }catch(java.sql.SQLException dos){

```

```

        dos.printStackTrace();
        System.out.println(dos.getMessage());
    }catch(Exception tres ){
        tres.printStackTrace();
        System.out.println(tres.getMessage());}
    return arreglo; }
public Vector Consulta_mensajes2(String f1,String f2){
    Vector arreglo =new Vector();
    int i=0;
    String tipo="";
    try{
        conectar();
        for(i=0;i<=2;i++){
            proceso=con.prepareStatement("{ Call CONTADOR_MENSAJES(?,?,?)}");
            proceso.setString(1,String.valueOf(i));
            proceso.setString(2,f1+" 00:00:01");
            proceso.setString(3,f2+" 23:59:59");
            proceso.execute();
            while(proceso.getResultSet().next()){
                Vector objeto=new Vector();
                objeto.addElement(proceso.getResultSet().getString("numero"));
                if(i==0){
                    tipo="Recibidos";}
                if(i==1){
                    tipo="Enviados";}
                if(i==2){
                    tipo=" No Enviados"; }
                objeto.addElement(tipo);
                arreglo.addElement(objeto);}
            proceso.close(); }
        desconectar();
    }catch(java.sql.SQLException dos){
        dos.printStackTrace();
        System.out.println(dos.getMessage());
    }catch(Exception tres ){
        tres.printStackTrace();
        System.out.println(tres.getMessage());}
    return arreglo; }
public Vector Consulta_mensaje(String tipo){
    Vector arreglo =new Vector();
    try{
        conectar();
        proceso=con.prepareStatement("{ Call VISTAS(?,?)}");
        proceso.setInt(1,5);
        proceso.setString(2,tipo);
        proceso.execute();
        while(proceso.getResultSet().next()){
            Vector objeto=new Vector();
            objeto.addElement(proceso.getResultSet().getString("mensaje"));

```

```

objeto.addElement(proceso.getResultSet().getString("dispositivos"));
objeto.addElement(proceso.getResultSet().getString("fecha"));
objeto.addElement(proceso.getResultSet().getString("nombre"));
        arreglo.addElement(objeto);}
    proceso.close();
    desconectar();
} catch (java.sql.SQLException dos) {
    dos.printStackTrace();
    System.out.println(dos.getMessage());
} catch (Exception tres) {
    tres.printStackTrace();
    System.out.println(tres.getMessage());}
return arreglo;}

public Vector Consulta_borradores(){
    Vector arreglo =new Vector();
    try{ conectar();
        proceso=con.prepareStatement("{ Call VISTAS(?,?)}");
        proceso.setInt(1,6);
        proceso.setString(2,"");
        proceso.execute();
        while(proceso.getResultSet().next()){
            Vector objeto=new Vector();
objeto.addElement(proceso.getResultSet().getString("id"));
objeto.addElement(proceso.getResultSet().getString("mensajes"));
            arreglo.addElement(objeto); }
        proceso.close();
        desconectar();
    } catch (java.sql.SQLException dos) {
        dos.printStackTrace();
        System.out.println(dos.getMessage());
    } catch (Exception tres) {
        tres.printStackTrace();
        System.out.println(tres.getMessage());}
    return arreglo; }

public void graba_obex(String direccion,String URL){
    try{ conectar();
        mantenimiento("update dispositivos set ruta='"+URL+"' where
direccion='"+direccion+"'");
        mantenimiento(" insert into nuevos
(rutas)values('"+URL+"')");
        desconectar();
    } catch (java.sql.SQLException dos) {
        dos.printStackTrace();
        System.out.println(dos.getMessage());
    } catch (Exception tres) {
        tres.printStackTrace();
        System.out.println(tres.getMessage()); }
    }
}

```


Clases

Class Dispositivos

java.lang.Object

└ Clases.Conexion

└ Clases.Dispositivos

Subclases directas:

Buscar_Servicios, RemoteDeviceDiscovery_1

public class Dispositivos

extends Conexion

Resumen constructor

Dispositivos()

Resumen métodos

void	borrar_nuevo(java.lang.String URL)
java.util.Vector	Consulta_dispositivos()
java.util.Vector	Consulta_nuevo_dispositivos()
java.util.Vector	Consulta_publicidad()
java.lang.String	formatea(java.lang.String direccion)
void	graba_mensaje(java.lang.String dispositivo, java.lang.String mensaje)
void	graba_obex(java.lang.String direccion,java.lang.String URL)
void	ignorada_publicidad(int codigo)
java.lang.Boolean	ingresar_dispositivos(java.lang.String nombre, java.lang.String direccion, int mayor, int menor, java.awt.TrayIcon trayIcon1)
java.util.Vector	Nueva_publicidad()
void	recibida_publicidad(int codigo)
void	uso_dispositivos(java.lang.String nombre, java.lang.String direccion)

Métodos heredados de la clase Clases.Conexion

conectar, Consultas_objeto, Consultas, desconectar, main, mantenimiento, verificaconsultas

Métodos heredados de la clase java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detalle

Dispositivos
public Dispositivos()

Método Detalle

ingresar_dispositivos
public java.lang.Boolean ingresar_dispositivos(java.lang.String nombre,
 java.lang.String direccion,
 int mayor,
 int menor,
 java.awt.TrayIcon trayIcon1)

graba_obex
public void graba_obex(java.lang.String direccion,
 java.lang.String URL)

graba_mensaje
public void graba_mensaje(java.lang.String dispositivo,
 java.lang.String mensaje)

borrar_nuevo
public void borrar_nuevo(java.lang.String URL)

uso_dispositivos
public void uso_dispositivos(java.lang.String nombre,
 java.lang.String direccion)

formatea
public java.lang.String formatea(java.lang.String direccion)

```
Consulta_dispositivos  
public java.util.Vector Consulta_dispositivos()
```

```
Consulta_nuevo_dispositivos  
public java.util.Vector Consulta_nuevo_dispositivos()
```

```
recibida_publicidad  
public void recibida_publicidad(int codigo)
```

```
ignorada_publicidad  
public void ignorada_publicidad(int codigo)
```

```
Consulta_publicidad  
public java.util.Vector Consulta_publicidad()
```

```
Nueva_publicidad  
public java.util.Vector Nueva_publicidad()
```

2. APPLICATION_BLUETOOTH

Esta clase se encargan de levantar el icono en la barra de estado que servira para iniciar el para el servicio de la busqueda de dispositivos y podran almacenarlos en la base de datos; asi como tambien activar o desactivar el envio de publicidad.

2.1 CLASE “Buscar_Archivos.java”

Esta clase ya que en ella permite interactuar el sistema con los usuarios por medio de peticiones. Estas otorgaran una respuesta procesadas por el sistema.

```

package application_bluetooth;
import java.io.*;
import java.util.logging.Level;
import java.util.logging.Logger;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
public class Buscar_Archivos extends Thread {
    TrayIcon trayIcon=null;
    Clases.Conexion base=null;
    public Buscar_Archivos(TrayIcon trayIcon1){
        System.out.println("Esperando peticiones.....");
        trayIcon=trayIcon1;
        base=new Clases.Conexion();
        try {    base.conectar();
        } catch (ClassNotFoundException ex) {
            Logger.getLogger(Buscar_Archivos.class.getName()).log(Level.SEVERE,
            null, ex); } catch (SQLException ex) {
            Logger.getLogger(Buscar_Archivos.class.getName()).log(Level.SEVERE,
            null, ex); } catch (Exception ex) {
            Logger.getLogger(Buscar_Archivos.class.getName()).log(Level.SEVERE,
            null, ex); }
    }
    private String devuelve_cadena(String cadena){
        String retorna="";
        int caracter=0;
        cadena=cadena.toLowerCase();
        for(int i=0;i<cadena.length();i++){
            caracter=(int)cadena.charAt(i);
            if((caracter>=97 && caracter<=122) ||
            (caracter>=48 && caracter<=59) || caracter==64 || caracter==46){
                retorna=retorna.concat(String.valueOf(cadena.charAt(i))); }
        } return retorna; }
    public String Devuelve_peticion(String archivo){
        String retorna="";
        File f = new File(archivo);
        BufferedReader entrada;
        try {
            String linea;
            entrada = new BufferedReader( new FileReader( f ) );
            while(entrada.ready()){
                linea = entrada.readLine();
                retorna=linea;
                break; }
            entrada.close();
        }catch (IOException e) {
            e.printStackTrace();
        } return retorna; }

```

```

public String Devuelve_peticion2(String archivo){
    String retorna="";
    File f = new File(archivo);
    BufferedReader entrada;
    try { String linea;
    entrada = new BufferedReader( new FileReader( f ) );
        while(entrada.ready()){
            linea = entrada.readLine();
            if(linea.indexOf("BODY:")==0){
                retorna=linea.substring(5);
                break;
            }
        } entrada.close();
    }catch (IOException e) {
        e.printStackTrace();
    }
    return retorna;
}

public void run(){ buscar();}
public void buscar() {
    String direccion="C:\\Users\\Pavilion\\Documents\\Directorio de
intercambio Bluetooth";
    String celular="";
    String peticion="";
    String argumento="";
    String cadena="";
    String mensaje_envia="";
    String dirs="";
    try{ while(true){
        File dir = new File(direccion);
        String[] ficheros = dir.list();
        if (ficheros!=null){
            for (int x=0;x<ficheros.length;x++){
                if(ficheros[x].indexOf(".txt")>=0 || ficheros[x].indexOf(".vnt")>=0 ){
                    mensaje_envia="";
                }
                if(ficheros[x].indexOf(".txt")>=0){
                    cadena=Devuelve_peticion(direccion+"\\ "+ficheros[x]);
                }else{
                    cadena=Devuelve_peticion2(direccion+"\\ "+ficheros[x]);
                }
                cadena=cadena.trim();
                cadena=devuelve_cadena(cadena);
                int i=cadena.indexOf(".");
                if(i>0){
                    celular=cadena.substring(0, i);
                    if(celular.length()==9 && isNumeric(celular)==true){
                        if(base.verificaconsultas("select * from usuarios_externos where
celular='"+celular+"'").booleanValue()==true){
                            int j=cadena.indexOf(".",i+1);
                            if(j>0){
                                peticion=cadena.substring(i+1, j);
                                if(j<cadena.length()){
                                    argumento=cadena.substring(j+1, cadena.length());
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}

```

```

argumento=argumento.trim();
String dispositivo=base.Consultas_objecto("select dispositivo as
Objecto from usuarios_externos where celular='"+celular+"'");
ResultSet consulta=base.Consultas("select * from peticiones where
peticion='"+peticion+"' and argumento='"+argumento+"'");
while(consulta.next()){
mensaje_envia=mensaje_envia.concat(consulta.getString("argumento"))+"\n
";
mensaje_envia=mensaje_envia.concat(consulta.getString("respuesta"))+"\n
"; }
if(mensaje_envia.length()>0){
Envia_Mensaje objeto=new Envia_Mensaje();
dirs=base.Consultas_objecto("select distinct ruta as Objecto from
dispositivos where direccion='"+dispositivo+"'");
if(dirs.trim().length()>0){
if(objeto.envia(mensaje_envia,dirs).booleanValue()==false){
trayIcon.displayMessage("Server Bluetooth", "No se realizo el envio al
dispositivo "+dispositivo, TrayIcon.MessageType.INFO);
}else{(new Clases.Dispositivos()).graba_mensaje(dispositivo, cadena); }
}
}
(new File(direccion+"\\ "+ficheros[x])).delete();}
}else{
peticion=cadena.substring(i+1,cadena.length());
if(peticion.compareTo("materias")==0){
if(base.verificaconsultas("select * from usuarios_externos a, alumnos b
where a.cedula=b.cedula and
a.celular='"+celular+"'").booleanValue()==true){
String cel=base.Consultas_objecto("select cedula as Objecto from
usuarios_externos where celular='"+celular+"'");
ResultSet consulta=base.Consultas("select funciones(5,materia,'') as
semetres,curso,funciones(4,materia,'') as nombre from matriculados
where alumno='"+cel+"'");
while(consulta.next()){
mensaje_envia=mensaje_envia.concat(consulta.getString("nombre"))+"\tS"+c
onsulta.getString("semetres")+consulta.getString("curso"))+"\n";}
String estudiante=base.Consultas_objecto("select concat(nombres,'
',apellidos) as Objecto from usuarios_externos where
cedula='"+cel+"'");
String dispositivo=base.Consultas_objecto("select dispositivo as
Objecto from usuarios_externos where cedula='"+cel+"'");
dirs=base.Consultas_objecto("select distinct ruta as Objecto from
dispositivos where direccion='"+dispositivo+"'");
Envia_Mensaje objeto=new Envia_Mensaje();
objeto.envia("Estudiante : "+estudiante+"\n"+mensaje_envia,dirs);}
}else{
if(peticion.compareTo("notas")==0){
int repetir_parcial=0;

```

```

if(base.verificaconsultas("select * from usuarios_externos a, alumnos b
where a.cedula=b.cedula and
a.celular='"+celular+"'").booleanValue()==true){
String ced=base.Consultas_objecto("select cedula as Objecto from
usuarios_externos where celular='"+celular+"'");
String no="";
ResultSet consulta=base.Consultas("select funciones(5,materia,'') as
semetres,curso,funciones(4,materia,'') as nombre,materia from
matriculados where alumno='"+ced+"' order by curso,materia asc");
while(consulta.next()){
ResultSet consultal=base.Consultas("select valor from notas where
alumno='"+ced+"' and materia='"+consulta.getString("materia")+"' order
by parcial asc");
while(consultal.next()){
no=no.concat(consultal.getString("valor")+"-");
repetir_parcial++; }
if(repetir_parcial==0){ no="[F-F]"; }else{
if(repetir_parcial==1){ no="["+no+"F]"; }else{
no=no.substring(0, no.length()-1); no="["+no+"]"; }
} repetir_parcial=0;
mensaje_envia=mensaje_envia.concat(consulta.getString("nombre")+"\tS"+c
onsulta.getString("semetres")+consulta.getString("curso")+
"+no+"\n";no=""); }
String estudiante=base.Consultas_objecto("select concat(nombres,
',apellidos) as Objecto from usuarios_externos where
cedula='"+ced+"'");
String dispositivo=base.Consultas_objecto("select dispositivo as
Objecto from usuarios_externos where cedula='"+ced+"'");
dirs=base.Consultas_objecto("select distinct ruta as Objecto from
dispositivos where direccion='"+dispositivo+"'");
Envia_Mensaje objeto=new Envia_Mensaje();
objeto.envia("Estudiante :"+estudiante+"\n"+mensaje_envia,dirs);}
}else{ if(base.verificaconsultas("select * from peticiones where
peticion='"+peticion+"'").booleanValue()==true){
String dispositivo=base.Consultas_objecto("select dispositivo as
Objecto from usuarios_externos where celular='"+celular+"'");
ResultSet consulta=base.Consultas("select * from peticiones where
peticion='"+peticion+"'");
while(consulta.next()){
mensaje_envia=mensaje_envia.concat(consulta.getString("argumento"))+"\n
";
mensaje_envia=mensaje_envia.concat(consulta.getString("respuesta"))+"\n
";}
if(mensaje_envia.length()>0){
Envia_Mensaje objeto=new Envia_Mensaje();
dirs=base.Consultas_objecto("select distinct ruta as Objecto from
dispositivos where direccion='"+dispositivo+"'");
if(dirs.trim().length()>0){
if(objeto.envia(mensaje_envia,dirs).booleanValue()==false){

```

```

trayIcon.displayMessage("Server Bluetooth", "No se realizo el envio al
dispositivo "+dispositivo, TrayIcon.MessageType.INFO);
}else{(new Clases.Dispositivos()).graba_mensaje(dispositivo, cadena);}
} } new File(direccion+"\\ "+ficheros[x])).delete(); } } }
if(celular.length()==10 && isNumeric(celular)==true){
if(base.verificaconsultas("select * from usuarios_externos where
cedula='"+celular+"'").booleanValue()==true){
String nuevo_celular=cadena.substring(i+1);
dirs=base.Consultas_objeto("select distinct ruta as Objecto from
dispositivos where direccion=(select dispositivo from usuarios_externos
where cedula='"+celular+"'");
String dispositivos=base.Consultas_objeto("select dispositivo as
Objecto from usuarios_externos where cedula='"+celular+"'");
if(nuevo_celular.length()==9 && isNumeric(nuevo_celular)==true ){
if(base.verificaconsultas("select * from usuarios_externos where
celular='"+nuevo_celular+"'").booleanValue()==true){
sleep(2000);
(new Envia_Mensaje()).envia("El numero del celular enviado ya se
encuentra registrado en la base de datos...", dirs);
(new File(direccion+"\\ "+ficheros[x])).delete();
(new Clases.Dispositivos()).graba_mensaje(dispositivos, cadena);
}else{
base.mantenimiento("update usuarios_externos set
celular='"+nuevo_celular+"' where cedula='"+celular+"'");
sleep(2000);
(new Envia_Mensaje()).envia("Se ha realizado la actualización del
número del celular con exito...", dirs);
(new File(direccion+"\\ "+ficheros[x])).delete();
(new Clases.Dispositivos()).graba_mensaje(dispositivos, cadena);}
}else{ sleep(2000);
(new Envia_Mensaje()).envia("El número del celular enviado es
incorrecto...", dirs);
(new File(direccion+"\\ "+ficheros[x])).delete();
(new Clases.Dispositivos()).graba_mensaje(dispositivos, cadena); }
} }else{
if((new File(direccion+"\\ "+ficheros[x])).exists()){
(new File(direccion+"\\ "+ficheros[x])).delete(); }
} } } } try { sleep(2000);
} catch (InterruptedException ex) {
Logger.getLogger(Buscar_Archivos.class.getName()).log(Level.SEVERE,
null, ex);} } } catch (SQLException ex) {
Logger.getLogger(Buscar_Archivos.class.getName()).log(Level.SEVERE,
null, ex); } catch (Exception ex) { } }
private static boolean isNumeric(String cadena){
try { Integer.parseInt(cadena);
return true;
} catch (NumberFormatException nfe){ return false; }
}}

```


application_bluetooth

Class Buscar_Archivos

java.lang.Object

└ java.lang.Thread

└ **application_bluetooth.Buscar_Archivos**

Todas las interfaces aplicadas: java.lang.Runnable

```
public class Buscar_Archivos
extends java.lang.Thread
```

Resumen de la clase anidada

Anidadas clases / interfaces heredado para la clase java.lang.Thread

java.lang.Thread.State, java.lang.Thread.UncaughtExceptionHandler

Resumen de Campo

Campos heredado de la clase java.lang.Thread

MAX_PRIORITY, MIN_PRIORITY, NORM_PRIORITY

Resumen constructor

Buscar_Archivos(java.awt.TrayIcon trayIcon1)

Resumen de método	
void	buscar()
java.lang.String	Devuelve_peticion (java.lang.String archivo)
java.lang.String	Devuelve_peticion2 (java.lang.String archivo)
static void	main (java.lang.String[] args)
void	run()

Métodos heredados de la clase java.lang.Thread
activeCount, checkAccess, countStackFrames, currentThread, destroy, dumpStack, enumerate, getAllStackTraces, getContextClassLoader, getDefaultUncaughtExceptionHandler, getId, getName, getPriority, getStackTrace, getState, getThreadGroup, getUncaughtExceptionHandler, holdsLock, interrupt, interrupted, isAlive, isDaemon, isInterrupted, join, join, join, resume, setContextClassLoader, setDaemon, setDefaultUncaughtExceptionHandler, setName, setPriority, setUncaughtExceptionHandler, sleep, sleep, start, stop, stop, suspend, toString, yield

Métodos heredados de la clase java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detalle
Buscar_Archivos public Buscar_Archivos (java.awt.TrayIcon trayIcon1)

Método Detalle
Devuelve_peticion public java.lang.String Devuelve_peticion (java.lang.String archivo)

Devuelve_peticion2

public java.lang.String **Devuelve_peticion2**(java.lang.String archivo)

run

public void **run**()

Especificada por:

ejecuta en la interfaz java.lang.Runnable

Overrides:

correr en la clase java.lang.Thread

buscar

public void **buscar**()

main

public static void **main**(java.lang.String[] args)

2.2 CLASE “Buscar_Servicios.java”

Esta clase busca los servicios de dispositivos que se han conectado al sistema.

```
package application_bluetooth;
import java.io.IOException;
import java.sql.SQLException;
import java.util.Enumeration;
import java.util.Vector;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.bluetooth.*;
public class Buscar_Servicios extends Clases.Dispositivos{
    static final UUID OBEX_FILE_TRANSFER = new UUID(0x1106);
    static final UUID OBEX_OBJECT_PUSH = new UUID(0x1105);
    static String envia="";
    public static final Vector/**/ serviceFound = new Vector();
    public void iniciar(Vector datos){
        UUID serviceUUID = OBEX_OBJECT_PUSH;
        serviceFound.clear();
        final Object serviceSearchCompletedEvent = new Object();
```

```

        DiscoveryListener listener = new DiscoveryListener() {
public void deviceDiscovered(RemoteDevice btDevice, DeviceClass cod) {
    }
public void inquiryCompleted(int discType) {
    }
public void servicesDiscovered(int transID, ServiceRecord[] servRecord)
{
    for (int i = 0; i < servRecord.length; i++) {
        String url
servRecord[i].getConnectionURL(ServiceRecord.NOAUTHENTICATE_NOENCRYPT,
false);

        if (url == null) { continue;
        }else{
            envia=url;
            int j=url.indexOf("//");
            int jj=url.indexOf(":",j);
            String dispositivos=url.substring(j+2, jj);
            graba_obex(dispositivos,url);
            serviceFound.add(url);
        }
    }
}

public void serviceSearchCompleted(int transID, int respCode) {
    System.out.println("service search completed!");
    synchronized(serviceSearchCompletedEvent){
        serviceSearchCompletedEvent.notifyAll();
    } };

UUID[] searchUuidSet = new UUID[] { serviceUUID };
int[] attrIDs = new int[] { 0x0100 // Service name
};
for(Enumeration en = datos.elements(); en.hasMoreElements(); ) {
    RemoteDevice btDevice = (RemoteDevice)en.nextElement();
    synchronized(serviceSearchCompletedEvent) {
        try {
LocalDevice.getLocalDevice().getDiscoveryAgent().searchServices(attrIDs
, searchUuidSet, btDevice, listener);
        } catch (BluetoothStateException ex) {
Logger.getLogger(Buscar_Servicios.class.getName()).log(Level.SEVERE,
null, ex);
        }
        try { serviceSearchCompletedEvent.wait();
        } catch (InterruptedException ex) {
Logger.getLogger(Buscar_Servicios.class.getName()).log(Level.SEVERE,
null, ex);
        }
    }
}
}

```

ApplicationBluetooth

Clase BuscarServicios

java.lang.Object

└ Clases.Conexion

└ Clases.Dispositivos

└ applicationBluetooth.BuscarServicios

```
public class BuscarServicios
extends Dispositivos
```

Resumen de Campo

static java.util.Vector	serviceFound
-------------------------	--------------

Resumen constructor

BuscarServicios()	
-------------------	--

Resumen de método

void	iniciar(java.util.Vector datos)
------	---------------------------------

Métodos heredados de la clase Clases.Dispositivos

borrar_nuevo, Consulta_dispositivos, Consulta_nuevo_dispositivos, Consulta_publicidad, formatea, graba_mensaje, graba_obex, ignorada_publicidad, ingresar_dispositivos, Nueva_publicidad, recibida_publicidad, uso_dispositivos

Métodos heredados de la clase Clases.Conexion

conectar, Consultas_objeto, Consultas, desconectar, main, mantenimiento, verificaconsultas

Métodos heredados de la clase java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Detalle campo

serviceFound
 public static final java.util.Vector serviceFound

Constructor Detalle

Buscar_Servicios
 public Buscar_Servicios()

Método Detalle

iniciar
 public void iniciar(java.util.Vector datos)

2.3 CLASE “Detectar_Dispositivo.java”

Esta clase trabaja como un hilo del sistema y tiene la finalidad de realizar una búsqueda detectar los dispositivo bluetooth que estan dentro del alcance, esto conlleva a buscar el tipo de servicio que utiliza el dispositivo para ser almacenado por el sistema.

```

package application_bluetooth;

import java.io.IOException;
import java.util.logging.Level;
import java.util.logging.Logger;
import java.awt.*;
import java.awt.event.*;
/**
 *
 * author grupo 13a
 */
public class Detectar_Dispositivo extends Thread {
    TrayIcon trayIcon=null;
    RemoteDeviceDiscovery_1 objecto=null;
    public Detectar_Dispositivo(TrayIcon trayIcon1){
        objecto = new RemoteDeviceDiscovery_1();
        trayIcon=trayIcon1;}
    public void run(){ try { while(true){
        System.out.println("Servicion para Detectar dispositivos... OK");
        java.util.Vector dis=objecto.busca_dispositivos(trayIcon);
        if (dis.size()>0){ (new Buscar_Servicios()).iniciar(dis); }
            try { sleep(9000);} catch (InterruptedException ex) {
                }}
        } catch (IOException ex) {
        } catch (InterruptedException ex) {
        } } }

```

application_bluetooth

Clase Detectar_Dispositivo

java.lang.Object

└ java.lang.Thread

└ application_bluetooth.Detectar_Dispositivo

Todas las interfaces aplicadas: java.lang.Runnable

```

public class Detectar_Dispositivo
extends java.lang.Thread

```

Resumen de la clase anidada
Anidadas clases / interfaces heredado para la clase java.lang.Thread
java.lang.Thread.State, java.lang.Thread.UncaughtExceptionHandler

Resumen de Campo
Campos heredado de la clase java.lang.Thread
MAX_PRIORITY, MIN_PRIORITY, NORM_PRIORITY

Resumen constructor
Detectar_Dispositivo(java.awt.TrayIcon trayIcon1)

Resumen de método
void run()

Métodos heredados de de la clase java.lang.Thread
activeCount, checkAccess, countStackFrames, currentThread, destroy, dumpStack, enumerate, getAllStackTraces, getContextClassLoader, getDefaultUncaughtExceptionHandler, getId, getName, getPriority, getStackTrace, getState, getThreadGroup, getUncaughtExceptionHandler, holdsLock, interrupt, interrupted, isAlive, isDaemon, isInterrupted, join, join, join, resume, setContextClassLoader, setDaemon, setDefaultUncaughtExceptionHandler, setName, setPriority, setUncaughtExceptionHandler, sleep, sleep, start, stop, stop, suspend, toString, yield

Métodos heredados de la clase java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detalle

Detectar_Dispositivo

public Detectar_Dispositivo(java.awt.TrayIcon trayIcon1)

Método Detalle

run

public void run()

Especificada por :

ejecuta en la interfaz java.lang.Runnable

Overrides:

run in class java.lang.Thread

2.4 CLASE “Envia_Mensaje.java”

Esta clase permitita enviar las respuestas procesadas por el sistema a los usuarios por medio de los dispositivos bluetooth.

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package application_bluetooth;
import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import javax.microedition.io.Connector;
import javax.obex.*;
import java.lang.*;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;
/**
 *
 * author grupo 13a
 */
public class Envia_Mensaje {
    public Boolean envia(String mensaje,String dispositivo){
        boolean respuesta=true;
        String direccion="";
    }

```

```

        try{ClientSession clientSession = (ClientSession)
Connector.open(dispositivo);
        HeaderSet hsConnectReply = clientSession.connect(null);
        if (hsConnectReply.getResponseCode() !=
ResponseCodes.OBEX_HTTP_OK) {
            System.out.println("Failed to connect");
            respuesta = false;
            return (new Boolean(respuesta)); }

        HeaderSet hsOperation = clientSession.createHeaderSet();
        hsOperation.setHeader(HeaderSet.NAME, "Respuesta_Servidor.txt");
        hsOperation.setHeader(HeaderSet.TYPE, "text");
        //Create PUT Operation
        Operation putOperation = clientSession.put(hsOperation);
        // Send some text to server
        byte data[] = mensaje.getBytes("iso-8859-1");
        OutputStream os = putOperation.openOutputStream();
        os.write(data);
        for(int i=0;i<5000;i++){
        } os.close();
        putOperation.close();
        clientSession.disconnect(null);
        clientSession.close();
        direccion=dispositivo.substring(9,dispositivo.indexOf(":",8));
        try { Clases.Dispositivos objeto= new Clases.Dispositivos();
            objeto.conectar();
            objeto.mantenimiento("update dispositivos set
fecha_acceso=curdate() where ruta='" + dispositivo + "'");
            objeto.desconectar();
        } catch (SQLException ex) {
Logger.getLogger(Envia_Mensaje.class.getName()).log(Level.SEVERE, null,
ex);        } catch (Exception ex) {
Logger.getLogger(Envia_Mensaje.class.getName()).log(Level.SEVERE, null,
ex);        }    }
        catch(IOException e){
            respuesta=false;    }
        return (new Boolean(respuesta));    }
public Boolean envia_imagen(String dir,String dispositivo,int
publicidad){
    boolean respuesta=true;
    String direccion="";
    try{ ClientSession clientSession = (ClientSession)
Connector.open(dispositivo);
        HeaderSet hsConnectReply = clientSession.connect(null);
        if (hsConnectReply.getResponseCode() !=
ResponseCodes.OBEX_HTTP_OK) {
            System.out.println("Failed to connect");
            respuesta = false;
            (new Clases.Dispositivos()).ignorada_publicidad(publicidad);

```

```

        return (new Boolean(respuesta));    }
    InputStream is = new FileInputStream((new File(dir)));
    byte filebytes[] = new byte[is.available()];
    is.read(filebytes);
    is.close();
    HeaderSet hsOperation = clientSession.createHeaderSet();
    hsOperation.setHeader(HeaderSet.NAME, (new
File(dir)).getName());
    hsOperation.setHeader(HeaderSet.TYPE, "image/jpg");
    hsOperation.setHeader(HeaderSet.LENGTH, new
Long(filebytes.length));
    //Create PUT Operation
    Operation putOperation = clientSession.put(hsOperation);
    // Send some text to server
    OutputStream os = putOperation.openOutputStream();
    os.write(filebytes);
    for(int i=0;i<5000;i++){
    }
    os.close();
    putOperation.close();
    clientSession.disconnect(null);
    clientSession.close();
    direccion=dispositivo.substring(9,dispositivo.indexOf(":",8));
    (new Clases.Dispositivos()).recibida_publicidad(publicidad);
        try {
            Clases.Dispositivos objeto= new Clases.Dispositivos();
            objeto.conectar();
            objeto.mantenimiento("update dispositivos set
fecha_acceso=curdate() where ruta='" + dispositivo + "'");
            objeto.desconectar();
        } catch (SQLException ex) {
Logger.getLogger(Envia_Mensaje.class.getName()).log(Level.SEVERE, null,
ex);
        } catch (Exception ex) {
Logger.getLogger(Envia_Mensaje.class.getName()).log(Level.SEVERE, null,
ex);
        }
    } catch (IOException e){
        respuesta=false; (new
Clases.Dispositivos()).ignorada_publicidad(publicidad);    }
    return (new Boolean(respuesta));    }
    public static void main(String[] args) throws IOException,
InterruptedException {
        Envia_Mensaje objeto=new Envia_Mensaje();
        objeto.envia_imagen("C:\\publicidad\\ADMINISTRa.jpg", "btgoep://0017E5AB
7895:8;authenticate=false;encrypt=false;master=false", 1);
    }
}

```

ApplicationBluetooth

Class Envia_Mensaje

java.lang.Object

↳ applicationBluetooth.Envia_Mensaje

```
public class Envia_Mensaje
extends java.lang.Object
```

Resumen constructor

Envia_Mensaje()

Resumen de método

java.lang.Boolean	envia_imagen(java.lang.String dir, java.lang.String dispositivo, int publicidad)
java.lang.Boolean	envia(java.lang.String mensaje, java.lang.String dispositivo)
static void	main(java.lang.String[] args)

Métodos heredados de de la clase java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detalle

```
Envia_Mensaje
public Envia_Mensaje()
```

Método Detalle

envia

```
public java.lang.Boolean envia(java.lang.String mensaje,
                               java.lang.String dispositivo)
```

envia_imagen

```
public java.lang.Boolean envia_imagen(java.lang.String dir,
                                       java.lang.String dispositivo,
                                       int publicidad)
```

main

```
public static void main(java.lang.String[] args)
    throws java.io.IOException,
           java.lang.InterruptedException
```

Throws:

java.io.IOException

java.lang.InterruptedException

2.5 CLASE “Envio_publicidad.java”

Esta clase permite enviar una imagen a los dispositivos que se encuentran dentro del area de cobertura.

```
package application_bluetooth;
import java.io.*;
import java.awt.*;
import java.sql.*;
import java.util.GregorianCalendar;
import java.util.logging.Level;
import java.util.logging.Logger;

public class Envio_publicidad extends Thread {
    public void run(){
        while(true){
            try {
                java.util.Vector rutas = (new
                Clases.Dispositivos()).Consulta_nuevo_dispositivos();
```

```

        if (rutas.size() > 0) {
            java.util.Vector envia = (new
Clases.Dispositivos()).Consulta_publicidad();
            for (int j = 0; j < rutas.size(); j++) {
                for (int i = 0; i < envia.size(); i++) {
                    java.util.Vector is = (java.util.Vector) envia.get(0);
                    (new Envia_Mensaje()).envia_imagen(is.get(1).toString(),
rutas.get(j).toString(), Integer.parseInt(is.get(0).toString()));}
                    (new Clases.Dispositivos()).borrar_nuevo(rutas.get(j).toString()); }
                }else{
                    java.util.Vector envia = (new
Clases.Dispositivos()).Nueva_publicidad();
                    if(envia.size()>0){ llena_rutas2();
                        rutas = (new
Clases.Dispositivos()).Consulta_nuevo_dispositivos();
                        for (int j = 0; j < rutas.size(); j++) {
                            for (int i = 0; i < envia.size(); i++) {
                                java.util.Vector is = (java.util.Vector) envia.get(0);
                                (new Envia_Mensaje()).envia_imagen(is.get(1).toString(),
rutas.get(j).toString(), Integer.parseInt(is.get(0).toString()));}
                                (new Clases.Dispositivos()).borrar_nuevo(rutas.get(j).toString()); }
                                    cambiar_estado_publicidad();}
                            } sleep(7000);
                            llena_rutas();
                        } catch (InterruptedException ex) {
Logger.getLogger(Envio_publicidad.class.getName()).log(Level.SEVERE,
null, ex); }
                    } }
public void cambiar_estado_publicidad(){
    Clases.Dispositivos dis=new Clases.Dispositivos();
    try { dis.conectar();
        dis.mantenimiento("update publicidad set nueva=1");
        dis.desconectar();
    } catch (ClassNotFoundException ex) {
Logger.getLogger(Envio_publicidad.class.getName()).log(Level.SEVERE,
null, ex); } catch (SQLException ex) {
Logger.getLogger(Envio_publicidad.class.getName()).log(Level.SEVERE,
null, ex); } catch (Exception ex) {
Logger.getLogger(Envio_publicidad.class.getName()).log(Level.SEVERE,
null, ex); } }
public void llena_rutas(){
    boolean ban=false;
    GregorianCalendar calendario = new GregorianCalendar();
    int i=calendario.get(GregorianCalendar.DAY_OF_WEEK);
    if(i>=1 && i<=7){
        ban=true;
        Clases.Dispositivos dis=new Clases.Dispositivos();
        try {

```

```

        dis.conectar();
        if(dis.verificaconsultas("select * from configuracion where
dia<>'"+String.valueOf(i)+"'").booleanValue()==true){
            dis.mantenimiento("insert into nuevos (rutas) select ruta
from dispositivos where ruta is not NULL");
            dis.mantenimiento("update configuracion set
dia='"+String.valueOf(i)+"'");
            dis.desconectar();
        } catch (ClassNotFoundException ex) {
Logger.getLogger(Envio_publicidad.class.getName()).log(Level.SEVERE,
null, ex);
        } catch (SQLException ex) {
Logger.getLogger(Envio_publicidad.class.getName()).log(Level.SEVERE,
null, ex);
        } catch (Exception ex) {
Logger.getLogger(Envio_publicidad.class.getName()).log(Level.SEVERE,
null, ex);
        } } }
public void llena_rutas2(){
    Clases.Dispositivos dis=new Clases.Dispositivos();
        try {
            dis.conectar();
            dis.mantenimiento("insert into nuevos
(rutas) select ruta from dispositivos where ruta is not NULL");
            dis.desconectar();
        } catch (ClassNotFoundException ex) {
Logger.getLogger(Envio_publicidad.class.getName()).log(Level.SEVERE,
null, ex);
        } catch (SQLException ex) {
Logger.getLogger(Envio_publicidad.class.getName()).log(Level.SEVERE,
null, ex);
        } catch (Exception ex) {
Logger.getLogger(Envio_publicidad.class.getName()).log(Level.SEVERE,
null, ex);
        } } }

```

Application_bluetooth
Class Envio_publicidad
java.lang.Object

└ java.lang.Thread

└ application_bluetooth.Envio_publicidad

Todas las interfaces implementadas: java.lang.Runnable

```

public class Envio_publicidad
extends java.lang.Thread

```

Resumen de la clase anidada
Anidadas clases / interfaces heredado de la clase java.lang.Thread
java.lang.Thread.State, java.lang.Thread.UncaughtExceptionHandler

Resumen de Campo
Campos heredado de la clase java.lang.Thread
MAX_PRIORITY, MIN_PRIORITY, NORM_PRIORITY

Resumen constructor
<u>Envio_publicidad()</u>

Resumen de método
void <u>cambiar_estado_publicidad()</u>
void <u>llena_rutas</u>
void <u>llena_rutas2()</u>
void <u>run()</u>

Métodos heredados de la clase java.lang.Thread
activeCount, checkAccess, countStackFrames, currentThread, destroy, dumpStack, enumerate, getAllStackTraces, getContextClassLoader, getDefaultUncaughtExceptionHandler, getId, getName, getPriority, getStackTrace, getState, getThreadGroup, getUncaughtExceptionHandler, holdsLock, interrupt, interrupted, isAlive, isDaemon, isInterrupted, join, join, join, resume, setContextClassLoader, setDaemon, setDefaultUncaughtExceptionHandler, setName, setPriority, setUncaughtExceptionHandler, sleep, sleep, start, stop, stop, suspend, toString, yield

Métodos heredados de la clase java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detalle

Envio_publicidad
public Envio_publicidad()

Método Detalle

run
public void run()
Especificada por :
ejecuta en la interfaz java.lang.Runnable
Overrides:
run in class java.lang.Thread

cambiar_estado_publicidad
public void cambiar_estado_publicidad()

llena_rutas
public void llena_rutas()

2.6 CLASE “Main.java”

Es la clase que inicializa los procesos del sistema. Desde esta clase interactua los usuarios con la maquina.

```
package application_bluetooth;

import java.awt.*;
import java.awt.event.*;
import java.util.logging.Level;
import java.util.logging.Logger;
public class Main{
MenuItem defaultItem3;
MenuItem defaultItem2;
MenuItem defaultItem4;
MenuItem defaultItem5;
```

```

MenuItem defaultItem6;
MenuItem defaultItem7;
Buscar_Archivos objeto;
Detectar_Dispositivo objeto_dectata;
Envio_publicidad objeto_envia;
public Main() throws InterruptedException
{
    final TrayIcon trayIcon;
    if (SystemTray.isSupported()) {
        SystemTray tray = SystemTray.getSystemTray();
        Image image =
Toolkit.getDefaultToolkit().getImage("bluetooth.gif");
        MouseListener mouseListener = new MouseListener() {
            public void mouseClicked(MouseEvent e) {
            }
            public void mouseEntered(MouseEvent e) {
            }
            public void mouseExited(MouseEvent e) {
            }
            public void mousePressed(MouseEvent e) {
            }
            public void mouseReleased(MouseEvent e) {
            }
        };
        PopupMenu popup = new PopupMenu();
        MenuItem defaultItem = new MenuItem("Exit");
        defaultItem2 = new MenuItem("Stop servicio de envio");
        defaultItem3 = new MenuItem("Start servicio de envio");
        defaultItem4 = new MenuItem("Stop servicio Detección de
Dispositivos");
        defaultItem5 = new MenuItem("Start servicio Detección de
Dispositivos");
        defaultItem6 = new MenuItem("Activar envio de Publicidad");
        defaultItem7 = new MenuItem("Desactivar envio de Publicidad");
        trayIcon = new TrayIcon(image, "Server Bluetooth", popup);
        objeto=new Buscar_Archivos(trayIcon);
        objeto.start();
        objeto_dectata=new Detectar_Dispositivo(trayIcon);
        objeto_envia=new Envio_publicidad();
        objeto_envia.start();
        objeto_envia.suspend();
        ActionListener exitListener = new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                System.out.println("Exiting...");
                objeto.stop();
                objeto_dectata.stop();
                System.exit(0);
            }
        };
        ActionListener stopListener = new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                objeto.suspend();
            }
        };
    }
}

```

```

        defaultItem3.setEnabled(true);
        defaultItem2.setEnabled(false);
        trayIcon.displayMessage("Server Bluetooth",
"Servicio de envio se encuentra suspendido...",
TrayIcon.MessageType.INFO); }
    };
    ActionListener startListener = new ActionListener() {
        public void actionPerformed(ActionEvent e) {
defaultItem3.setEnabled(false);
            defaultItem2.setEnabled(true);
            trayIcon.displayMessage("Server Bluetooth",
"Servicio de envio se encuentra activo...",
TrayIcon.MessageType.INFO); }
    };
    ActionListener startde = new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            if(objecto_dectata.isInterrupted()){
                objeto_dectata.resume();}else{
                objeto_dectata.start();}
            defaultItem4.setEnabled(true);
            defaultItem5.setEnabled(false);
            trayIcon.displayMessage("Server Bluetooth",
"Servicio de envio detección se encuentra activo...",
TrayIcon.MessageType.INFO); }
    };
    ActionListener startdeno = new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            objeto.suspend();
            defaultItem4.setEnabled(false);
            defaultItem5.setEnabled(true);
            trayIcon.displayMessage("Server Bluetooth",
"Servicio de envio detección se encuentra desactivado...",
TrayIcon.MessageType.INFO); }
    };
    ActionListener startenvio = new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            objeto_envia.resume();
            defaultItem6.setEnabled(false);
            defaultItem7.setEnabled(true);
            trayIcon.displayMessage("Server Bluetooth", "Se
ha Activado el envio de la publicidad con exito...",
TrayIcon.MessageType.INFO); }
    };
    ActionListener stopenvio = new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            objeto_envia.suspend();
            defaultItem6.setEnabled(true);
            defaultItem7.setEnabled(false);
            trayIcon.displayMessage("Server Bluetooth", "Se
ha Desctivado el envio de la publicidad con exito...",
TrayIcon.MessageType.INFO); }
    };
    defaultItem.addActionListener(exitListener);
    defaultItem2.addActionListener(stopListener);

```

```

defaultItem3.addActionListener(startListener);
defaultItem4.addActionListener(startdeno);
defaultItem5.addActionListener(startde);
defaultItem6.addActionListener(startenvio);
defaultItem7.addActionListener(stopenvio);
popup.add(defaultItem3);
popup.add(defaultItem2);
popup.add(defaultItem4);
popup.add(defaultItem5);
popup.add(defaultItem6);
popup.add(defaultItem7);
popup.add(defaultItem);
defaultItem3.setEnabled(false);
defaultItem4.setEnabled(false);
defaultItem7.setEnabled(false);
ActionListener actionListener = new ActionListener() {
    public void actionPerformed(ActionEvent e) {
    }
};
trayIcon.setImageAutoSize(true);
trayIcon.addActionListener(actionListener);
trayIcon.addMouseListener(mouseListener);
try {
    tray.add(trayIcon);
} catch (AWTException e) {
}
} else {
    System.exit(0);
}
}
}

public static void main(String[] args) throws InterruptedException
{
    Main main = new Main();
}

```

ApplicationBluetooth

Class Main

java.lang.Object

└─applicationBluetooth.Main

public class **Main**

extends java.lang.Object

Resumen constructor

Main()

Resumen de método

static void **main**(java.lang.String[] args)

Métodos heredados de de la clase java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detalle**Main**

public **Main**()
throws java.lang.InterruptedException

Throws:

java.lang.InterruptedException

Método Detalle**main**

public static void **main**(java.lang.String[] args)
throws java.lang.InterruptedException

Throws:

java.lang.InterruptedException

2.7 CLASE “RemoteDeviceDiscovery.java”

Esta clase permite detectar los dispositivos que estan en el rango de cobertura.

```
package application_bluetooth;
import java.io.IOException;
import java.util.Vector;
import javax.bluetooth.*;
public class RemoteDeviceDiscovery {
    public static final Vector/**/ devicesDiscovered = new Vector();
    public static void main(String[] args) throws IOException,
    InterruptedException {
        final Object inquiryCompletedEvent = new Object();
        devicesDiscovered.clear();
```

```

        DiscoveryListener listener = new DiscoveryListener() {
            public void deviceDiscovered(RemoteDevice btDevice,
DeviceClass cod) {
                System.out.println("Device " +
btDevice.getBluetoothAddress() + " found");
                devicesDiscovered.addElement(btDevice);
                try {
                    System.out.println("    name " +
btDevice.getFriendlyName(false));
                } catch (IOException cantGetDeviceName) {
                }
                public void inquiryCompleted(int discType) {
                    System.out.println("Device Inquiry completed!");
                    synchronized(inquiryCompletedEvent){
                        inquiryCompletedEvent.notifyAll();
                    }
                }
                public void serviceSearchCompleted(int transID, int
respCode) {
                }
                public void servicesDiscovered(int transID, ServiceRecord[]
servRecord) {
                }
            };
            synchronized(inquiryCompletedEvent) {
                boolean started =
LocalDevice.getLocalDevice().getDiscoveryAgent().startInquiry(Discovery
Agent.GIAC, listener);
                if (started) {
                    System.out.println("wait for device inquiry to complete...");
                    inquiryCompletedEvent.wait();
                    System.out.println(devicesDiscovered.size() + " device(s) found");
                }
            }
        }
    }
}

```

application_bluetooth

Class RemoteDeviceDiscovery

java.lang.Object

└ application_bluetooth.RemoteDeviceDiscovery

```

public class RemoteDeviceDiscovery
extends java.lang.Object

```

Resumen de Campo

static java.util.Vector **devicesDiscovered**

Resumen constructor

RemoteDeviceDiscovery()

Resumen de método

static void **main**(java.lang.String[] args)

Métodos heredados de de la clase java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Detalle campo

devicesDiscovered

public static final java.util.Vector **devicesDiscovered**

Constructor Detalle

RemoteDeviceDiscovery

public **RemoteDeviceDiscovery()**

Método Detalle

main

public static void **main**(java.lang.String[] args)
 throws java.io.IOException,
 java.lang.InterruptedException

Throws:

java.io.IOException
 java.lang.InterruptedException

2.8 CLASE “RemoteDeviceDiscovery_1.java”

```

/**
 * BlueCove - Java library for Bluetooth
 */
package application_bluetooth;
import java.io.IOException;
import java.util.Vector;
import java.awt.*;
import java.awt.event.*;
import javax.bluetooth.*;
/**
 * author grupo 13a
 *
 * Minimal DeviceDiscovery example for javadoc.
 */
public class RemoteDeviceDiscovery_1 extends Clases.Dispositivos {
    public static final Vector/*<RemoteDevice>*/ devicesDiscovered =
new Vector();
    public static final Vector tipo=new Vector();
    TrayIcon trayIcon=null;
    public Vector busca_dispositivos(TrayIcon trayIcon1)throws
IOException, InterruptedException {
        final Object inquiryCompletedEvent = new Object();
        trayIcon=trayIcon1;
        devicesDiscovered.clear();
        tipo.clear();
        DiscoveryListener listener = new DiscoveryListener() {
            public void deviceDiscovered(RemoteDevice btDevice, DeviceClass cod) {
                try {
                    if(ingresar_dispositivos(btDevice.getFriendlyName(false).toString(),btD
evice.getBluetoothAddress().toString(),cod.getMajorDeviceClass(),cod.ge
tMinorDeviceClass(),trayIcon).booleanValue()){
                        tipo.addElement(btDevice);
                    }
                } catch (IOException cantGetDeviceName) {
                }
            }
        };
        inquiryCompletedEvent.notifyAll();
    }
}

```



```

    }
    public void serviceSearchCompleted(int transID, int respCode) {
        System.out.println("Termino"); }

    public void servicesDiscovered(int transID, ServiceRecord[] servRecord)
    {
        }; synchronized(inquiryCompletedEvent) {
            boolean started =
LocalDevice.getLocalDevice().getDiscoveryAgent().startInquiry(Discovery
Agent.GIAC, listener);
            if (started) { inquiryCompletedEvent.wait(); }
        } return tipo; }}

```

Application_bluetooth

Class RemoteDeviceDiscovery_1

java.lang.Object

└ Clases.Conexion

└ Clases.Dispositivos

└ application_bluetooth.RemoteDeviceDiscovery_1

```

public class RemoteDeviceDiscovery_1
extends Dispositivos

```

Resumen de Campo

static java.util.Vector	<u>devicesDiscovered</u>
static java.util.Vector	<u>tipo</u>

Resumen constructor

<u>RemoteDeviceDiscovery_1()</u>

Resumen de método

java.util.Vector **busca_dispositivos**(java.awt.TrayIcon trayIcon1)

Métodos heredados de la clase Clases.Dispositivos

borrar_nuevo, Consulta_dispositivos, Consulta_nuevo_dispositivos, Consulta_publicidad, formatea, graba_mensaje, graba_obex, ignorada_publicidad, ingresar_dispositivos, Nueva_publicidad, recibida_publicidad, uso_dispositivos

Métodos heredados de la clase Clases.Conexion

conectar, Consultas_objeto, Consultas, desconectar, main, mantenimiento, verificaconsultas

Métodos heredados de la clase java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Detalle campo

devicesDiscovered

public static final java.util.Vector **devicesDiscovered**

tipo

public static final java.util.Vector **tipo**

Constructor Detalle

RemoteDeviceDiscovery_1

public **RemoteDeviceDiscovery_1**()

Método Detalle

busca_dispositivos

public java.util.Vector **busca_dispositivos**(java.awt.TrayIcon trayIcon1)
 throws java.io.IOException,
 java.lang.InterruptedException

Throws:

java.io.IOException

java.lang.InterruptedException

2.9 CLASE “ServicesSearch.java”

Esta clase permite buscar los servicios de los dispositivos detectados.

```
package application_bluetooth;

import java.io.IOException;
import java.sql.SQLException;
import java.util.Enumeration;
import java.util.Vector;
import javax.bluetooth.*;

public class ServicesSearch {
    static final UUID OBEX_FILE_TRANSFER = new UUID(0x1106);
    static final UUID OBEX_OBJECT_PUSH = new UUID(0x1105);
    public static final Vector/**/ serviceFound = new Vector();
    public static void main(String[] args) throws IOException,
    InterruptedException, SQLException, Exception {
        // First run RemoteDeviceDiscovery and use discovered device
        RemoteDeviceDiscovery.main(null);
        serviceFound.clear();
        UUID serviceUUID = OBEX_OBJECT_PUSH;
        if ((args != null) && (args.length > 0)) {
            serviceUUID = new UUID(args[0], false);
        }
        final Object serviceSearchCompletedEvent = new Object();
        DiscoveryListener listener = new DiscoveryListener() {
        public void deviceDiscovered(RemoteDevice btDevice, DeviceClass cod) {
        }
        public void inquiryCompleted(int discType) {
        }
        public void servicesDiscovered(int transID, ServiceRecord[] servRecord)
        {
            for (int i = 0; i < servRecord.length; i++) {
```


Resumen de Campo

static java.util.Vector **serviceFound**

Resumen constructor

ServicesSearch()

Resumen de método

static void **main**(java.lang.String[] args)

Métodos heredados de la clase java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Detalle campo

serviceFound

public static final java.util.Vector **serviceFound**

Constructor Detalle

ServicesSearch

public **ServicesSearch()**

Método Detalle

main

public static void **main**(java.lang.String[] args)

throws java.io.IOException,
java.lang.InterruptedException,
java.sql.SQLException,
java.lang.Exception

Throws:

java.io.IOException
 java.lang.InterruptedExceptio
 java.sql.SQLException
 java.lang.Exception

3. WEB_BLUETOOTH CLASE

3.1 CLASE “Borradores.java”

Permite realizar manteminiento, consulta, y eliminacion de datos de la tabla borrados.

```

package Clases;
import java.sql.*;
public class Borradores extends Conexion{
    CallableStatement proceso = null;
    public void ingresar_borradores(String titulo,String contenido){
        try{
            conectar();
            proceso=con.prepareCall("{ Call
MANTENIMIENTO_BORRADORES(?,?,?,?)}");
            proceso.setInt(1,1);
            proceso.setString(2,titulo.toUpperCase());
            proceso.setString(3,contenido.toUpperCase());
            proceso.setInt(4,0);
            proceso.execute();
            proceso.close();
            desconectar();
        }catch(java.sql.SQLException dos){
            dos.printStackTrace();
            System.out.println(dos.getMessage());
        }catch(Exception tres ){
            tres.printStackTrace();
            System.out.println(tres.getMessage()); }
    }
    public void modificar_borradores(String titulo,String contenido,String
codigo){
        try{
            conectar();
            proceso=con.prepareCall("{ Call
MANTENIMIENTO_BORRADORES(?,?,?,?)}");
            proceso.setInt(1,2);
            proceso.setString(2,titulo.toUpperCase());

```

```

        proceso.setString(3, contenido.toUpperCase());
        proceso.setInt(4, Integer.parseInt(codigo));
        proceso.execute();
        proceso.close();
        desconectar(); } catch (java.sql.SQLException dos) {
            dos.printStackTrace();
            System.out.println(dos.getMessage());
        } catch (Exception tres) {
            tres.printStackTrace();
            System.out.println(tres.getMessage()); } }
    public void elimina_borradores(String codigo) {
        try {
            conectar();
            proceso=con.prepareStatement(" { Call
MANTENIMIENTO_BORRADORES(?,?,?,?) } ");
            proceso.setInt(1,3);
            proceso.setString(2,"");
            proceso.setString(3,"");
            proceso.setInt(4,Integer.parseInt(codigo));
            proceso.execute();
            proceso.close();
            desconectar();
        } catch (java.sql.SQLException dos) {
            dos.printStackTrace();
            System.out.println(dos.getMessage());
        } catch (Exception tres) {
            tres.printStackTrace();
            System.out.println(tres.getMessage());
        }
    }
}

```

Clases

3.1.1 Class Borradores

java.lang.Object

└ Clases.Conexion

└ **Clases.Borradores**

```

public class Borradores
extends Conexion

```

Resumen constructor	
<u>Borradores()</u>	

Resumen de método	
void	<u>elimina_borradores</u> (java.lang.String codigo)
void	<u>ingresar_borradores</u> (java.lang.String titulo, java.lang.String contenido)
void	<u>modificar_borradores</u> (java.lang.String titulo, java.lang.String contenido, java.lang.String codigo)

Métodos heredados de la clase Clases. <u>Conexion</u>
<u>conectar</u> , <u>Consultas_objecto</u> , <u>Consultas</u> , <u>desconectar</u> , <u>getConexion</u> , <u>main</u> , <u>mantenimiento</u> , <u>verificaconsultas</u>

Métodos heredados de la clase java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
Constructor Detalle

Borradores
public **Borradores()**

Método Detalle

ingresar_borradores

public void **ingresar_borradores**(java.lang.String titulo,
java.lang.String contenido)

modificar_borradores

public void **modificar_borradores**(java.lang.String titulo,
java.lang.String contenido,
java.lang.String codigo)

elimina_borradores

public void **elimina_borradores**(java.lang.String codigo)

3.2 CLASE “Conexion.java”

Permite realizar mantenimiento, consulta, y eliminacion de datos de la tabla conexión.

```
package Clases;
/**
 *
 * author grupo 13a
 */
import java.sql.*;
import javax.naming.Context;
import javax.naming.InitialContext;
import javax.sql.DataSource;
import javax.naming.NamingException;
import javax.naming.NoInitialContextException;
public class Conexion {
    String error;
    Connection con;
    public Conexion(){}
    public void conectar() throws
ClassNotFoundException, SQLException, Exception{
        try{
            Class.forName("org.gjt.mm.mysql.Driver").newInstance();
con=DriverManager.getConnection("jdbc:mysql://localhost/sms?user=root&p
assword=");
        }catch(ClassNotFoundException uno ){
            error="No puede localizar la base de datos";
            throw new ClassNotFoundException(error);
        }catch(SQLException dos ){
            error="No puede realizar la conexion d la base de datos";
            throw new SQLException(error+dos.toString());
        }catch(Exception tres ){
            error="Error desconocido";
            throw new Exception(error+tres.toString());
        }
    }
    public Connection getConexion(){
        return con;
    }
    public void desconectar () throws SQLException{
        try{
            if(con!=null){
                if(!con.isClosed()){ con.close(); }
            }
        }catch(SQLException dos ){
            error="Imposible desconectar la base de datos";
            throw new SQLException(error); }
    }
    public ResultSet Consultas(String sql) throws SQLException,Exception{
        ResultSet rs=null;
        try{
```

```

Statement stmt=con.createStatement();
rs=stmt.executeQuery(sql);
}catch(SQLException dos ){
    error="No puede ejecutar la consulta";
    throw new SQLException(error);
}catch(Exception tres ){
    error="Generacion de execepcion en la consulat";
    throw new Exception(error);    }
return rs;    }
public String Consultas_objecto(String sql) throws
SQLException,Exception{
    ResultSet rs=null;    String retorna="";
    try{    Statement stmt=con.createStatement();
rs=stmt.executeQuery(sql);
while(rs.next()){    retorna=rs.getString("Objecto");    }
}catch(SQLException dos ){
    error="No puede ejecutar la consulta";
    throw new SQLException(error);
}catch(Exception tres ){
    error="Generacion de execepcion en la consulat";
    throw new Exception(error);    }
return retorna;    }
public Boolean verificaconsultas(String sql) throws
SQLException,Exception{    boolean ban=false;
ResultSet rs=null;
try{    Statement stmt=con.createStatement();
rs=stmt.executeQuery(sql);
if(rs.next()){    ban=true;    }
}catch(SQLException dos ){
    error="No puede ejecutar la consulta";
    throw new SQLException(error);
}catch(Exception tres ){
    error="Generacion de execepcion en la consulat";
    throw new Exception(error);    }
return (new Boolean(ban));    }
public void mantenimiento(String sql)    throws
SQLException,Exception{    if(con!=null){        try{
        PreparedStatement sentencia;
        sentencia=con.prepareStatement(sql);
        sentencia.execute();
        }catch(SQLException dos ){
            error="No puede ejecutar la mantenimiento";
            throw new SQLException(error);
        }catch(Exception tres ){
            error="Generacion de execepcion en la mantenimiento";
            throw new Exception(error);        }        }
public static void main(String[] args)throws SQLException,Exception
{
    Conexion p= new Conexion();
    p.conectar();    }}

```

Clases

Class Conexion

java.lang.Object

└ **Clases.Conexion Conocido Subclases directas:**

Borradores, Dispositivos, Externos, Mensajes, Permisos, Peticiones, Publicidad, Usuarios

```
public class Conexion
extends java.lang.Object
```

Resumen constructor

Conexion()

Resumen de método

void	conectar()
java.lang.String	Consultas_objeto (java.lang.String sql)
java.sql.ResultSet	Consultas (java.lang.String sql)
void	desconectar()
java.sql.Connection	getConexion()
static void	main (java.lang.String[] args)
void	mantenimiento (java.lang.String sql)
java.lang.Boolean	verificaconsultas (java.lang.String sql)

Métodos heredados de la clase java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detalle

Conexion

public **Conexion**()

Método Detalle

conectar

public void **conectar**()
throws java.lang.ClassNotFoundException,
java.sql.SQLException,
java.lang.Exception

Throws:

java.lang.ClassNotFoundException
java.sql.SQLException
java.lang.Exception

getConexion

public java.sql.Connection **getConexion**()

desconectar

public void **desconectar**()
throws java.sql.SQLException

Throws:

java.sql.SQLException

Consultas

public java.sql.ResultSet **Consultas**(java.lang.String sql)
throws java.sql.SQLException,
java.lang.Exception

Throws:

java.sql.SQLException
java.lang.Exception

Consultas_objecto

```
public java.lang.String Consultas_objecto(java.lang.String sql)
                                throws java.sql.SQLException,
                                java.lang.Exception
```

Throws:

```
java.sql.SQLException
java.lang.Exception
```

3.3 CLASE “Dispositivos.java”

Ingreso de dispositivos a la base de datos, asignar un dispositivo a un usuario, consulta de dispositivos detectados, rutas de envios de los dispositivos, modificar y consulta rutas de envio.

```
package Clases;
import java.sql.*;
import java.util.Vector;
/**
 *
 * author grupo 13a
 */
public class Dispositivos extends Conexion {
    CallableStatement proceso = null;
    public String devuelve_ruta(String dispositivo){
        String ruta="";
        try{
            conectar();
            ruta=Consultas_objecto("select ruta as Objecto from
dispositivos where direccion='"+dispositivo+"'");
            desconectar(); }catch(java.sql.SQLException dos){
                dos.printStackTrace();
                System.out.println(dos.getMessage());
            }catch(Exception tres ){
                tres.printStackTrace();
                System.out.println(tres.getMessage()); }
        return ruta; }
    public Boolean ingresar_dispositivos(String nombre,String
direccion,int mayor,int menor){
        boolean ban=false;
        try{
            conectar();
            if(verificaconsultas("select * from dispositivos where
direccion='"+direccion+"'").booleanValue()==false){
                proceso=con.prepareStatement(" { Call
MANTENIMIENTO_DISPOSITIVOS(?, ?, ?) }");
```

```

        proceso.setString(1,nombre.toUpperCase());
        proceso.setString(2,direccion);
        proceso.setInt(3,1);
        proceso.execute();
        proceso.close();
        mantenimiento("update dispositivos set
mayor_class="+String.valueOf(mayor)+",minor_class="+String.valueOf(menor)+
r)+" where direccion='"+direccion+"'");
        desconectar();
        ban=true;    }
    }catch(java.sql.SQLException dos){
        dos.printStackTrace();
        System.out.println(dos.getMessage());
    }catch(Exception tres ){
        tres.printStackTrace();
        System.out.println(tres.getMessage());
    }
    return (new Boolean(ban)); }
    public void uso_dispositivos(String nombre,String direccion){
        try{
            conectar();
            direccion=Consultas_objecto("select direccion as objeto from
dispositivos where ruta='"+direccion+"'");
            proceso=con.prepareStatement("{ Call
MANTENIMIENTO_DISPOSITIVOS(?,?,?)}");
            proceso.setString(1,"");
            proceso.setString(2,direccion);
            proceso.setInt(3,2);
            proceso.execute();
            proceso.close();
            desconectar();
        }catch(java.sql.SQLException dos){
            dos.printStackTrace();
            System.out.println(dos.getMessage());
        }catch(Exception tres ){
            tres.printStackTrace();
            System.out.println(tres.getMessage()); } }
    public void registra_mensaje(String direccion,String mensaje,
String tipo){    try{
        conectar();
        direccion=Consultas_objecto("select direccion as objeto from
dispositivos where ruta='"+direccion+"'");
        mantenimiento("insert into mensajes
(mensaje,dispositivos,fecha,tipo)values('"+mensaje.toUpperCase()+"','"+
direccion+"',now(),'+tipo+')");
        mantenimiento("update dispositivos set numeros=numeros+1
where direccion='"+direccion+"'");
        desconectar();
    }catch(java.sql.SQLException dos){
        dos.printStackTrace();
        System.out.println(dos.getMessage());
    }
}

```

```

        }catch(Exception tres ){
            tres.printStackTrace();
            System.out.println(tres.getMessage()); } }
public String formatea(String direccion){
    String dir="";
    String h="";
    for(int i=0;i<direccion.length();i++){
        if((i%2)==0 && i>0){
            dir=dir.concat(":");
            dir=dir.concat(String.valueOf(direccion.charAt(i)));
        }else{dir=dir.concat(String.valueOf(direccion.charAt(i))); }
    } return dir; }
public Vector Consulta_dispositivos(){
    Vector arreglo =new Vector();
    try{
        conectar();
        proceso=con.prepareCall("{ Call VISTAS(?,?)}");
        proceso.setInt(1,15);
        proceso.setString(2,"");
        proceso.execute();
        while(proceso.getResultSet().next()){
            Vector objeto=new Vector();
            objeto.addElement(proceso.getResultSet().getString("nombre"));
            objeto.addElement(formatea(proceso.getResultSet().getString("direccion"
            )));
            objeto.addElement(proceso.getResultSet().getString("fecha_dectecion"));
            objeto.addElement(proceso.getResultSet().getString("fecha_acceso"));
            objeto.addElement(proceso.getResultSet().getString("numeros"));
            objeto.addElement(proceso.getResultSet().getString("mayor_class"));
            objeto.addElement(proceso.getResultSet().getString("minor_class"));
            objeto.addElement(proceso.getResultSet().getString("ruta"));
            arreglo.addElement(objeto);
        }
        proceso.close();
        desconectar();
    }catch(java.sql.SQLException dos){
        dos.printStackTrace();
        System.out.println(dos.getMessage());
    }catch(Exception tres ){
        tres.printStackTrace();
        System.out.println(tres.getMessage()); }
    return arreglo; }
public String Consulta_Otros(String sql){
    String retorna="0";
    String valor="";
    try{
        conectar();
        valor=Consultas_objeto(sql);
        if(valor.length()>0){
            retorna=valor;
        }
    }

```

```

        desconectar();
    }catch(java.sql.SQLException dos){
        dos.printStackTrace();
        System.out.println(dos.getMessage());
    }catch(Exception tres ){
        tres.printStackTrace();
        System.out.println(tres.getMessage());
    }
return retorna; }
public Vector Consulta_acceso(){
    Vector arreglo =new Vector();
    try{
        conectar();
        proceso=con.prepareStatement("{ Call VISTAS(?,?)}");
        proceso.setInt(1,14);
        proceso.setString(2,"");
        proceso.execute();
        while(proceso.getResultSet().next()){
            Vector objeto=new Vector();
            objeto.addElement(proceso.getResultSet().getString("usuario"));
            objeto.addElement(formatea(proceso.getResultSet().getString("fecha")));
            objeto.addElement(proceso.getResultSet().getString("tipo"));
            objeto.addElement(proceso.getResultSet().getString("nombres"));
            objeto.addElement(proceso.getResultSet().getString("apellidos"));
            arreglo.addElement(objeto);
        }
        proceso.close();
        desconectar();
    }catch(java.sql.SQLException dos){
        dos.printStackTrace();
        System.out.println(dos.getMessage());
    }catch(Exception tres ){
        tres.printStackTrace();
        System.out.println(tres.getMessage());
    }
return arreglo; }
public Vector Consulta_mensajes(){
    Vector arreglo =new Vector();
    int i=0;
    String tipo="";
    try{
        conectar();
        for(i=0;i<=2;i++){
            proceso=con.prepareStatement("{ Call VISTAS(?,?)}");
            proceso.setInt(1,12);
            proceso.setString(2,String.valueOf(i));
            proceso.execute();
            while(proceso.getResultSet().next()){
                Vector objeto=new Vector();
                objeto.addElement(proceso.getResultSet().getString("numero"));
                if(i==0){ tipo="Recibidos";
                } if(i==1){
                    tipo="Enviados";
                }
            }
        }
    }
}

```



```

        } if(i==2){
            tipo=" No Enviados";
        }
        objeto.addElement(tipo);
        arreglo.addElement(objeto);
    } proceso.close();
}
desconectar();
}catch(java.sql.SQLException dos){
    dos.printStackTrace();
    System.out.println(dos.getMessage());
}catch(Exception tres ){
    tres.printStackTrace();
    System.out.println(tres.getMessage());
}
return arreglo;
}
public Vector Consulta_mensajes2(String f1,String f2){
    Vector arreglo =new Vector();
    int i=0;
    String tipo="";
    try{
        conectar();
        for(i=0;i<=2;i++){
            proceso=con.prepareStatement("Call CONTADOR_MENSAJES(?,?,?)");
            proceso.setString(1,String.valueOf(i));
            proceso.setString(2,f1+" 00:00:01");
            proceso.setString(3,f2+" 23:59:59");
            proceso.execute();
            while(proceso.getResultSet().next()){
                Vector objeto=new Vector();
                objeto.addElement(proceso.getResultSet().getString("numero"));
                if(i==0){
                    tipo="Recibidos";
                }
                if(i==1){
                    tipo="Enviados";
                }
                if(i==2){
                    tipo=" No Enviados";
                }
                objeto.addElement(tipo);
                arreglo.addElement(objeto);
            }
            proceso.close();
        }
        desconectar();
    }catch(java.sql.SQLException dos){
        dos.printStackTrace();
        System.out.println(dos.getMessage());
    }catch(Exception tres ){
        tres.printStackTrace();
        System.out.println(tres.getMessage());
    }
    return arreglo;
}
public Vector Consulta_mensaje(String tipo){
    Vector arreglo =new Vector();
    try{
        conectar();
    }
}

```

```

        proceso=con.prepareCall("{ Call VISTAS(?,?) }");
        proceso.setInt(1,5);
        proceso.setString(2, tipo);
        proceso.execute();
        while(proceso.getResultSet().next()){
            Vector objeto=new Vector();
            objeto.addElement(proceso.getResultSet().getString("mensaje"));
            objeto.addElement(proceso.getResultSet().getString("dispositivos"));
            objeto.addElement(proceso.getResultSet().getString("fecha"));
            objeto.addElement(proceso.getResultSet().getString("nombre"));
            arreglo.addElement(objeto);
        }
        proceso.close();
        desconectar();
    }catch(java.sql.SQLException dos){
        dos.printStackTrace();
        System.out.println(dos.getMessage());
    }catch(Exception tres ){
        tres.printStackTrace();
        System.out.println(tres.getMessage());
    }
    return arreglo;
}

public Vector Consulta_borradores(){
    Vector arreglo =new Vector();
    try{
        conectar();
        proceso=con.prepareCall("{ Call VISTAS(?,?) }");
        proceso.setInt(1,6);
        proceso.setString(2, "");
        proceso.execute();
        while(proceso.getResultSet().next()){
            Vector objeto=new Vector();
            objeto.addElement(proceso.getResultSet().getString("id"));
            objeto.addElement(proceso.getResultSet().getString("titulo"));
            objeto.addElement(proceso.getResultSet().getString("contenido"));
            arreglo.addElement(objeto);
        }
        proceso.close();
        desconectar();
    }catch(java.sql.SQLException dos){
        dos.printStackTrace();
        System.out.println(dos.getMessage());
    }catch(Exception tres ){
        tres.printStackTrace();
        System.out.println(tres.getMessage());
    }
    return arreglo;
}

public void graba_obex(String direccion,String URL){
    try{
        conectar();
        mantenimiento("update dispositivos set ruta='"+URL+"' where
direccion='"+direccion+"'");
    }
}

```

```

        mantenimiento(" insert into nuevos
(rutas)values('"+URL+"'");
        desconectar();
    }catch(java.sql.SQLException dos){
        dos.printStackTrace();
        System.out.println(dos.getMessage());
    }catch(Exception tres ){
        tres.printStackTrace();
        System.out.println(tres.getMessage());    }
    }}

```

Clases

Class Dispositivos

java.lang.Object

└ Clases.Conexion

└ **Clases.Dispositivos**

Conocido Subclases directas:

Buscar_Servicios, Envia_Mensaje, RemoteDeviceDiscovery

```

public class Dispositivos
extends Conexion

```

Resumen constructor

<u>Dispositivos()</u>	
------------------------------	--

Resumen de método

java.util.Vector	Consulta_acceso()
java.util.Vector	Consulta_borradores()

java.util.Vector	Consulta_dispositivos()
java.util.Vector	Consulta_mensaje (java.lang.String tipo)
java.util.Vector	Consulta_mensajes()
java.util.Vector	Consulta_mensajes2 (java.lang.String f1, java.lang.String f2)
java.lang.String	Consulta_Otros (java.lang.String sql)
java.lang.String	devuelve_ruta (java.lang.String dispositivo)
java.lang.String	formatea (java.lang.String direccion)
void	graba_obex (java.lang.String direccion, java.lang.String URL)
java.lang.Boolean	ingresar_dispositivos (java.lang.String nombre, java.lang.String direccion, int mayor, int menor)
void	registra_mensaje (java.lang.String direccion, java.lang.String mensaje, java.lang.String tipo)
void	uso_dispositivos (java.lang.String nombre, java.lang.String direccion)

Métodos heredados de la clase Clases.Conexion

conectar, Consultas_objeto, Consultas, desconectar, getConexion, main, mantenimiento, verificaconsultas

Métodos heredados de la clase java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detalle

Dispositivos

public **Dispositivos()**

Método Detalle**devuelve_ruta**

```
public java.lang.String devuelve_ruta(java.lang.String dispositivo)
```

ingresar_dispositivos

```
public java.lang.Boolean ingresar_dispositivos(java.lang.String nombre,  
                                                java.lang.String direccion,  
                                                int mayor,  
                                                int menor)
```

uso_dispositivos

```
public void uso_dispositivos(java.lang.String nombre,  
                             java.lang.String direccion)
```

registra_mensaje

```
public void registra_mensaje(java.lang.String direccion,  
                             java.lang.String mensaje,  
                             java.lang.String tipo)
```

formatea

```
public java.lang.String formatea(java.lang.String direccion)
```

Consulta_dispositivos

```
public java.util.Vector Consulta_dispositivos()
```

Consulta_Otros

```
public java.lang.String Consulta_Otros(java.lang.String sql)
```

Consulta_acceso

```
public java.util.Vector Consulta_acceso()
```

Consulta_mensajes

```
public java.util.Vector Consulta_mensajes()
```

Consulta_mensajes2

```
public java.util.Vector Consulta_mensajes2(java.lang.String f1,
                                             java.lang.String f2)
```

Consulta_mensaje

```
public java.util.Vector Consulta_mensaje(java.lang.String tipo)
```

Consulta_borradores

```
public java.util.Vector Consulta_borradores()
```

graba_obex

```
public void graba_obex(java.lang.String direccion,
                       java.lang.String URL)
```

3.4 CLASE “externos.java”

Esta clase permite ingresar modificar eliminar usuarios externos y Consulta de exportar datos en pdf y excel.

```
package Clases;
import java.sql.*;
/**
 *
 * author grupo 13a
 */
public class Externos extends Conexion {
    CallableStatement proceso = null;
    public Boolean verifica_cedula(String cedula){
        Boolean objeto=null;
        try{
            conectar();
            objeto= verificaconsultas("select * from usuarios_externos
where cedula='"+cedula+"'");
            desconectar();
        }catch (java.sql.SQLException dos){
            System.out.println(dos.getMessage());
        }catch (Exception tres ){
            System.out.println(tres.getMessage());
        }
        return objeto;
    }
    public Boolean verifica_cedula1(String cedula,String codigo){
```

```

        Boolean objeto=null;
        try{
            conectar();
            objeto= verificaconsultas("select * from usuarios_externos
where cedula='"+cedula+"' and id='"+codigo+"'");
            desconectar();
        }catch(java.sql.SQLException dos){
            System.out.println(dos.getMessage());
        }catch(Exception tres ){
            System.out.println(tres.getMessage()); }
        return objeto; }
        public Boolean verifica_celular1(String celular,String codigo){
            Boolean objeto=null;
            try{
                conectar();
                objeto= verificaconsultas("select * from usuarios_externos
where celular='"+celuar+"' and id='"+codigo+"'");
                desconectar();
            }catch(java.sql.SQLException dos){
                System.out.println(dos.getMessage());
            }catch(Exception tres ){
                System.out.println(tres.getMessage()); }
            return objeto; }
        public Boolean verifica_celular(String celular){
            Boolean objeto=null;
            try{
                conectar();
                objeto= verificaconsultas("select * from usuarios_externos
where celular='"+celuar+"'");
                desconectar();
            }catch(java.sql.SQLException dos){
                System.out.println(dos.getMessage());
            }catch(Exception tres ){
                System.out.println(tres.getMessage()); }
            return objeto; }
        public void ingresar_externo(String nombre,String
apellido,String celular,String dispositivo,String cedula){
            try{
                conectar();
                dispositivo=dispositivo.replaceAll(":", "");
                proceso=con.prepareCall("{ Call
MANTENIMIENTO_EXTERNOS(?,?,?,?,?,?,?)}");
                proceso.setInt(1,1);
                proceso.setString(2,nombre.toUpperCase());
                proceso.setString(3,apellido.toUpperCase());
                proceso.setString(4,celular);
                proceso.setString(5,dispositivo);
                proceso.setString(6,cedula);
                proceso.setInt(7,1);
                proceso.execute();
                proceso.close();
                desconectar();
            }

```

```

    }catch(java.sql.SQLException dos){
        dos.printStackTrace();
        System.out.println(dos.getMessage());
    }catch(Exception tres ){
        tres.printStackTrace();
        System.out.println(tres.getMessage());} }
    public void actualizar_externo(String nombre,String
apellido,String celular,String dispositivo,String cedula,String id){
    try{        conectar();
        dispositivo=dispositivo.replaceAll(":", "");
        proceso=con.prepareStatement(" { Call
MANTENIMIENTO_EXTERNOS(?,?,?,?,?,?,?)");
        proceso.setInt(1,2);
        proceso.setString(2,nombre.toUpperCase());
        proceso.setString(3,apellido.toUpperCase());
        proceso.setString(4,celular);
        proceso.setString(5,dispositivo);
        proceso.setString(6,cedula);
        proceso.setInt(7,Integer.parseInt(id));
        proceso.execute();
        proceso.close();
        desconectar();
    }catch(java.sql.SQLException dos){
        dos.printStackTrace();
        System.out.println(dos.getMessage());
    }catch(Exception tres ){
        tres.printStackTrace();
        System.out.println(tres.getMessage()); }
} }

```

Clases

Class Externos

java.lang.Object

└ Clases.Conexion

└ **Clases.Externos**

```

public class Externos
extends Conexion

```

Resumen constructor	
Externos()	

Resumen de método	
void	actualizar_externo (java.lang.String nombre, java.lang.String apellido, java.lang.String celular, java.lang.String dispositivo, java.lang.String cedula, java.lang.String id)
void	ingresar_externo (java.lang.String nombre, java.lang.String apellido, java.lang.String celular, java.lang.String dispositivo, java.lang.String cedula)
java.lang.Boolean	verifica_cedula (java.lang.String cedula)
java.lang.Boolean	verifica_cedula1 (java.lang.String cedula, java.lang.String codigo)
java.lang.Boolean	verifica_celular (java.lang.String celular)
java.lang.Boolean	verifica_celular1 (java.lang.String celular, java.lang.String codigo)

Métodos heredados de la clase Clases.Conexion
conectar, Consultas_objeto, Consultas, desconectar, getConexion, main, mantenimiento, verificaconsultas

Métodos heredados de la clase java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detalle
Externos
public Externos ()

Método Detalle

verifica_cedula

```
public java.lang.Boolean verifica_cedula(java.lang.String cedula)
```

verifica_cedula1

```
public java.lang.Boolean verifica_cedula1(java.lang.String cedula,  
                                             java.lang.String codigo)
```

verifica_celular1

```
public java.lang.Boolean verifica_celular1(java.lang.String celular,  
                                             java.lang.String codigo)
```

verifica_celular

```
public java.lang.Boolean verifica_celular(java.lang.String celular)
```

ingresar_externo

```
public void ingresar_externo(java.lang.String nombre,  
                             java.lang.String apellido,  
                             java.lang.String celular,  
                             java.lang.String dispositivo,  
                             java.lang.String cedula)
```

actualizar_externo

```
public void actualizar_externo(java.lang.String nombre,  
                               java.lang.String apellido,  
                               java.lang.String celular,  
                               java.lang.String dispositivo,  
                               java.lang.String cedula,  
                               java.lang.String id)
```

3.5 CLASE “fecha.java”

Esta clase permite dar el formato a las fechas que utilizan los buzones de entrada, salida, no recibidos, borradores y contador.

```

package Clases;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.GregorianCalendar;
/**
 *
 * author grupo 13a
 */
public class Fechas {
    public String f1="";    public String f2="";
    public String f3="";    public String f4="";
    public String f5="";    public Fechas(){
        Calendar cal = new GregorianCalendar();
        java.util.Date fecha =cal.getTime();
        SimpleDateFormat formato = new SimpleDateFormat("yyyy-MM-dd");
        f1= formato.format(fecha);
        Calendar call = new GregorianCalendar();
        call.add(Calendar.DATE,-7 );
        java.util.Date fechal =call.getTime();
        f2= formato.format(fechal);
        Calendar call4 = new GregorianCalendar();
        call4.add(Calendar.DATE,-1 );
        java.util.Date fechal4 =call4.getTime();
        f5= formato.format(fechal4);
        Calendar cal2 = new GregorianCalendar();
        cal2.add(Calendar.DATE,-15 );
        java.util.Date fecha2 =cal2.getTime();
        f3= formato.format(fecha2);
        Calendar cal3 = new GregorianCalendar();
        cal3.add(Calendar.DATE,-30 );
        java.util.Date fecha3 =cal3.getTime();
        f4= formato.format(fecha3); } }

```

Clases

Class Fechas

java.lang.Object

└ Clases.Fechas

```

public class Fechas
extends java.lang.Object

```

Resumen de Campo	
java.lang.String	<u>f1</u>
java.lang.String	<u>f2</u>
java.lang.String	<u>f3</u>
java.lang.String	<u>f4</u>
java.lang.String	<u>f5</u>

Resumen constructor	
<u>Fechas()</u>	

Resumen de método	
Métodos heredados de de la clase java.lang.Object	
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait	

Detalle campo	
f1	public java.lang.String f1
f2	public java.lang.String f2
f3	public java.lang.String f3
f4	public java.lang.String f4
f5	public java.lang.String f5

Constructor Detalle

Fechas

public **Fechas()**

3.6 CLASE “funciones.java”

Permite validar a los usuarios que ingresan a la pagina, encriptar la clave que fue ingresada por el usuario por medio del algoritmo md5. Y tambien guarda registros para realizar las auditorias en el sistema.

```
package Clases;
import java.security.*;
/**
 *
 * author grupo 13a
 */
public class Funciones {
    // verifica el usuario y la clave del login de la pagina index.jsp
    public Boolean verifica_login(String usuario, String clave,String
aletorio){
        boolean bandera=true;
        String clave_usuario="";
        try{
            Conexion conectar =new Conexion();
            conectar.conectar();
            if(true==conectar.verificaconsultas("select * from usuarios
where usuario='"+usuario+"' ").booleanValue()){
                clave_usuario=conectar.Consultas_objecto("select clave as
objecto from usuarios where usuario='"+usuario+"'");
                if(clave.toUpperCase().compareTo(convertir_clave(clave_usuario,aletorio
))!=0){
                    bandera=false;
                }else{
                    conectar.mantenimiento("update usuarios set
sesion='"+aletorio+"' where usuario='"+usuario+"' ");
                }
            }else{
                bandera=false;
            }
            conectar.desconectar();
        }catch(java.sql.SQLException dos){
        }catch(ClassNotFoundException uno ){
        }catch(Exception tres ){
        }
        return (new Boolean(bandera));
    }
    public void graba_acceso(String usuario,String tipo){
        try{
            Conexion conectar =new Conexion();
```

```

        conectar.conectar();
        conectar.mantenimiento("insert into auditoria
(usuario,fecha,tipo)value('"+usuario+"',now(),'"+tipo+"')");
        conectar.desconectar();
    }catch(java.sql.SQLException dos){
    }catch(ClassNotFoundException uno ){
    }catch(Exception tres ){
    }
}
//convierte la clave encriptada MD5
private String convertir_clave(String clave, String numero){
    byte[] hash = null;
    byte[] uniqueKey =null;
        clave=clave.concat(numero);
        uniqueKey = clave.getBytes();
        try {
            hash =
MessageDigest.getInstance("MD5").digest(uniqueKey);
        }catch (NoSuchAlgorithmException e) {
            throw new Error("no MD5 support in this VM"); }
        StringBuffer hashString = new StringBuffer();
        for ( int i = 0; i < hash.length; ++i ) {
            String hex = Integer.toHexString(hash[i]);
            if ( hex.length() == 1 ) {
                hashString.append('0');
                hashString.append(hex.charAt(hex.length()-1));
            } else {
                hashString.append(hex.substring(hex.length()-2));
            } }
        return hashString.toString().toUpperCase(); }}

```

Clases

Class Funciones

java.lang.Object

└ Clases.Funciones

```

public class Funciones
extends java.lang.Object

```

Resumen constructor

Funciones()**Resumen de método**

void	graba_acceso (java.lang.String usuario, java.lang.String tipo)
java.lang.Boolean	verifica_login (java.lang.String usuario, java.lang.String clave, java.lang.String aleatorio)

Métodos heredados de la clase java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detalle**Funciones**

public **Funciones**()

Método Detalle**verifica_login**

```
public java.lang.Boolean verifica_login(java.lang.String usuario,
                                         java.lang.String clave,
                                         java.lang.String aleatorio)
```

graba_acceso

```
public void graba_acceso(java.lang.String usuario,
                         java.lang.String tipo)
```

3.7 CLASE “mensajes.java”

Esta clase me permite realizar una consulta entre fechas a la tabla mensajes en la cual me retornara un objeto tipo vector que va a ser utilizado en las consultas de los buzones.

```
package Clases;
import java.sql.*;
import java.util.Vector;
public class Mensajes extends Conexion{
    CallableStatement proceso = null;
    public Vector Consulta_mensaje(String tipo,String f1,String f2){
        Vector arreglo =new Vector();
        try{
            conectar();
            proceso=con.prepareCall("{ Call VISTA_MENSAJES(?,?,?)}");
            proceso.setString(1,tipo);
            proceso.setString(2,f1);    proceso.setString(3,f2);
            proceso.execute();
            while(proceso.getResultSet().next()){
                Vector objeto=new Vector();
                objeto.addElement(proceso.getResultSet().getString("mensaje"));
                objeto.addElement(proceso.getResultSet().getString("dispositivos"));
                objeto.addElement(proceso.getResultSet().getString("fecha"));
                objeto.addElement(proceso.getResultSet().getString("nombre"));
                arreglo.addElement(objeto);    }
            proceso.close();    desconectar();
        }catch(java.sql.SQLException dos){
            dos.printStackTrace();
            System.out.println(dos.getMessage());
        }catch(Exception tres ){
            tres.printStackTrace();
            System.out.println(tres.getMessage());    }
        return arreglo;    }}

```

Clases

Class Mensajes

java.lang.Object

└ Clases.Conexion

└ **Clases.Mensajes**


```
public class Mensajes
extends Conexion
```

Resumen constructor

Mensajes()

Resumen de método

```
java.util.Vector Consulta mensaje(java.lang.String tipo,  
java.lang.String f1, java.lang.String f2)
```

3.8 CLASE “permisos.java”

Esta clase permite realizar un mantenimiento de la tabla permisos en la base de datos, la consulta, ingresos, modificación y eliminación de permisos.

```
package Clases;
import java.sql.*;
/**
 *
 * author grupo 13a
 */
public class Permisos extends Conexion {
    CallableStatement proceso = null;
    public Boolean verifica_permisos(String nombre){
        Boolean objeto=null;
        try{
            conectar();
            objeto= verificaconsultas("select * from permisos where
nombre='"+nombre+"'");
            desconectar();
        }catch(java.sql.SQLException dos){
            System.out.println(dos.getMessage());
        }catch(Exception tres ){
            System.out.println(tres.getMessage());
        } return objeto;    }
    public Boolean permisos_usuarios(String codigo){
        Boolean objeto=null;
        try{ conectar();
            objeto= verificaconsultas("select * from usuarios where
permiso="+codigo);
            desconectar();
        }catch(java.sql.SQLException dos){
            System.out.println(dos.getMessage());
        }catch(Exception tres ){
            System.out.println(tres.getMessage());
        } return objeto;    }
    public Boolean verificar_permisos(String nombre,String codigo){
        Boolean objeto=null;
        try{
            conectar();
            objeto= verificaconsultas("select * from permisos where
nombre='"+nombre+"' and not(codigo="+codigo+" )");
            desconectar();
        }catch(java.sql.SQLException dos){
            System.out.println(dos.getMessage());
        }catch(Exception tres ){
            System.out.println(tres.getMessage());
        } return objeto;    }
```

```

    public void ingresar_permisos(String nombre,String op1,String
op2,String op3,String op4,String op5,String op6,String op7,String
op8,String op9,String op10,String op11,String op12,String op13,String
op14,String op15,String op16,String op17,String op18){
        try{    conectar();
            proceso=con.prepareStatement(" { Call
MANTENIMIENTO_PERMISOS(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?) }");
            proceso.setInt(1,(new Integer(op1)).intValue());
            proceso.setInt(2,(new Integer(op2)).intValue());
            proceso.setInt(3,(new Integer(op3)).intValue());
            proceso.setInt(4,(new Integer(op4)).intValue());
            proceso.setInt(5,(new Integer(op5)).intValue());
            proceso.setInt(6, (new Integer(op6)).intValue());
            proceso.setInt(7, (new Integer(op7)).intValue());
            proceso.setInt(8, (new Integer(op8)).intValue());
            proceso.setInt(9, (new Integer(op9)).intValue());
            proceso.setInt(10, (new Integer(op10)).intValue());
            proceso.setInt(11, (new Integer(op11)).intValue());
            proceso.setInt(12, (new Integer(op12)).intValue());
            proceso.setInt(13, (new Integer(op13)).intValue());
            proceso.setInt(14, (new Integer(op14)).intValue());
            proceso.setInt(15, (new Integer(op15)).intValue());
            proceso.setInt(16, (new Integer(op16)).intValue());
            proceso.setInt(17, (new Integer(op17)).intValue());
            proceso.setInt(18, (new Integer(op18)).intValue());
            proceso.setString(19, nombre.toUpperCase());
            proceso.setInt(20,0);
            proceso.setInt(21,1);
            proceso.execute();
            proceso.close();
        }catch(java.sql.SQLException dos){
            dos.printStackTrace();
            System.out.println(dos.getMessage());
        }catch(Exception tres ){
            tres.printStackTrace();
            System.out.println(tres.getMessage()); } }

    public void eliminar_permisos(String codigo){
    // public void eliminar_usuarios(){
        try{
            conectar();
            proceso=con.prepareStatement(" { Call
MANTENIMIENTO_PERMISOS(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?) }");
            proceso.setInt(1,0);
            proceso.setInt(2,0);
            proceso.setInt(4,0);
            proceso.setInt(3,0);
            proceso.setInt(5,0);
            proceso.setInt(6,0);
            proceso.setInt(7,0);

```

```

        proceso.setInt(8,0);
        proceso.setInt(9,0);
        proceso.setInt(10,0);
        proceso.setInt(11,0);
        proceso.setInt(12,0);
        proceso.setInt(13,0);
        proceso.setInt(14,0);
        proceso.setInt(15,0);
        proceso.setInt(16,0);
        proceso.setInt(17,0);
        proceso.setInt(18,0);
        proceso.setString(19, "");
        proceso.setInt(20,Integer.parseInt(codigo));
        proceso.setInt(21,3);
        proceso.execute();
        proceso.close();
    }catch(java.sql.SQLException dos){
        System.out.println(dos.getMessage());
    }catch(Exception tres ){
        System.out.println(tres.getMessage());} }

    public void modificar_permisos(String nombre,String op1,String
op2,String op3,String op4,String op5,String op6,String op7,String
op8,String op9,String op10,String op11,String op12,String op13,String
op14,String op15,String op16,String op17,String op18,String codigo){
    try{
        conectar();
        proceso=con.prepareStatement(" { Call
MANTENIMIENTO_PERMISOS(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)");
        proceso.setInt(1,(new Integer(op1)).intValue());
        proceso.setInt(2,(new Integer(op2)).intValue());
        proceso.setInt(3,(new Integer(op3)).intValue());
        proceso.setInt(4,(new Integer(op4)).intValue());
        proceso.setInt(5,(new Integer(op5)).intValue());
        proceso.setInt(6, (new Integer(op6)).intValue());
        proceso.setInt(7, (new Integer(op7)).intValue());
        proceso.setInt(8, (new Integer(op8)).intValue());
        proceso.setInt(9, (new Integer(op9)).intValue());
        proceso.setInt(10, (new Integer(op10)).intValue());
        proceso.setInt(11, (new Integer(op11)).intValue());
        proceso.setInt(12, (new Integer(op12)).intValue());
        proceso.setInt(13, (new Integer(op13)).intValue());
        proceso.setInt(14, (new Integer(op14)).intValue());
        proceso.setInt(15, (new Integer(op15)).intValue());
        proceso.setInt(16, (new Integer(op16)).intValue());
        proceso.setInt(17, (new Integer(op17)).intValue());
        proceso.setInt(18, (new Integer(op18)).intValue());
        proceso.setString(19, nombre.toUpperCase());
        proceso.setInt(20,(new Integer(codigo).intValue() ));
        proceso.setInt(21,2);

```

```

        proceso.execute();
        proceso.close();
    }catch(java.sql.SQLException dos){
        System.out.println(dos.getMessage());
    }catch(Exception tres ){
        System.out.println(tres.getMessage()); }
    }
}

```

Clases

Class Permisos

java.lang.Object

└ Clases.Conexion

└ **Clases.Permisos**

```

public class Permisos
extends Conexion

```

Resumen constructor

Permisos()

Resumen de método

void	eliminar_permisos (java.lang.String codigo)
void	ingresar_permisos (java.lang.String nombre, java.lang.String op1, java.lang.String op2, java.lang.String op3, java.lang.String op4, java.lang.String op5, java.lang.String op6, java.lang.String op7, java.lang.String op8,

	java.lang.String op9, java.lang.String op10, java.lang.String op11, java.lang.String op12, java.lang.String op13, java.lang.String op14, java.lang.String op15, java.lang.String op16, java.lang.String op17, java.lang.String op18)
void	modificar_permisos (java.lang.String nombre, java.lang.String op1, java.lang.String op2, java.lang.String op3, java.lang.String op4, java.lang.String op5, java.lang.String op6, java.lang.String op7, java.lang.String op8, java.lang.String op9, java.lang.String op10, java.lang.String op11, java.lang.String op12, java.lang.String op13, java.lang.String op14, java.lang.String op15, java.lang.String op16, java.lang.String op17, java.lang.String op18, java.lang.String codigo)
java.lang.Boolean	permisos_usuarios (java.lang.String codigo)
java.lang.Boolean	verifica_permisos (java.lang.String nombre)
java.lang.Boolean	verificar_permisos (java.lang.String nombre, java.lang.String codigo)

Métodos heredados de la clase Clases.Conexion

conectar, Consultas_objeto, Consultas, desconectar, getConexion, main, mantenimiento, verificaconsultas

Métodos heredados de la clase java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detalle

Permisos

```
public Permisos()
```

Método Detalle

verifica_permisos

```
public java.lang.Boolean verifica_permisos(java.lang.String nombre)
```

permisos_usuarios

```
public java.lang.Boolean permisos_usuarios(java.lang.String codigo)
```

verificar_permisos

[illegible]

ingresar_permisos

```
public void ingresar_permisos(java.lang.String nombre,  
    java.lang.String op1,  
    java.lang.String op2,  
    java.lang.String op3,  
    java.lang.String op4,  
    java.lang.String op5,  
    java.lang.String op6,  
    java.lang.String op7,  
    java.lang.String op8,  
    java.lang.String op9,  
    java.lang.String op10,  
    java.lang.String op11,  
    java.lang.String op12,  
    java.lang.String op13,  
    java.lang.String op14,  
    java.lang.String op15,  
    java.lang.String op16,  
    java.lang.String op17,  
    java.lang.String op18)
```

eliminar_permisos

```
public void eliminar_permisos(java.lang.String codigo)
```

modificar_permisos

```
public void modificar_permisos(java.lang.String nombre,  
    java.lang.String op1,  
    java.lang.String op2,  
    java.lang.String op3,  
    java.lang.String op4,  
    java.lang.String op5,  
    java.lang.String op6,  
    java.lang.String op7,  
    java.lang.String op8,  
    java.lang.String op9,  
    java.lang.String op10,  
    java.lang.String op11,  
    java.lang.String op12,  
    java.lang.String op13,  
    java.lang.String op14,  
    java.lang.String op15,  
    java.lang.String op16,  
    java.lang.String op17,  
    java.lang.String op18,  
    java.lang.String codigo)
```

3.9 CLASE “peticiones.java”

Esta clase permite realizar un mantenimiento de la tabla `peticion` en la base de datos, la consulta, ingresos, modificación y eliminación de peticiones.

```
En la tabla peticion  
package Clases;  
import java.sql.*;  
/**  
 *  
 * author grupo 13a  
 */  
public class Peticiones extends Conexion{  
    CallableStatement proceso = null;  
    public void ingresar_peticion(String peticion,String argumento,String  
    respuesta){
```



```

        try{
            conectar(); proceso=con.prepareStatement("{ Call
MANTENIMIENTO_PETICIONES(?,?,?,?,?)}");
            proceso.setString(1,peticion.toUpperCase());
            proceso.setString(2,argumento.toUpperCase());
            proceso.setString(3,respuesta.toUpperCase());
            proceso.setInt(4,0); proceso.setInt(5,1);
            proceso.execute(); proceso.close();
            desconectar();
        }catch(java.sql.SQLException dos){
            dos.printStackTrace();
            System.out.println(dos.getMessage());
        }catch(Exception tres ){
            tres.printStackTrace();
            System.out.println(tres.getMessage()); } }
    public void eliminar_peticion(String codigo){
    // public void eliminar_usuarios(){
        try{ conectar();
            proceso=con.prepareStatement("{ Call
MANTENIMIENTO_PETICIONES(?,?,?,?,?)}");
            proceso.setString(1,"");
            proceso.setString(2,"");
            proceso.setString(3,"");
            proceso.setInt(4, Integer.parseInt(codigo));
            proceso.setInt(5, 3);
            proceso.execute(); proceso.close();
            desconectar();
        }catch(java.sql.SQLException dos){
            System.out.println(dos.getMessage());
        }catch(Exception tres ){
            System.out.println(tres.getMessage()); }
    } }

```

Clases

Class Peticiones

java.lang.Object

└ Clases.Conexion

└ **Clases.Peticiones**

public class **Peticiones**

extends Conexion

Resumen constructor

<u>Peticiones()</u>	
----------------------------	--

Resumen de método	
void	<u>eliminar_peticion</u> (java.lang.String codigo)
void	<u>ingresar_peticion</u> (java.lang.String peticion, java.lang.String argumento, java.lang.String respuesta)

Métodos heredados de la clase Clases. <u>Conexion</u>
conectar, Consultas_objeto, Consultas, desconectar, getConexion, main, mantenimiento, verificaconsultas

Métodos heredados de la clase java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detalle

Peticiones

public **Peticiones**()

Método Detalle

ingresar_peticion

```
public void ingresar_peticion(java.lang.String peticion,
                             java.lang.String argumento,
                             java.lang.String respuesta)
```

eliminar_peticion

```
public void eliminar_peticion(java.lang.String codigo)
```

3.10 CLASE “publicidad.java”

Esta clase permite realizar un mantenimiento de la tabla publicidad en la base de datos, la consulta, ingresos, modificación y eliminación de publicidades.

```
package Clases;
import java.sql.*;
import java.util.Vector;
public class Publicidad extends Conexion{
    CallableStatement proceso = null;
    public void ingresar_publicidad(String nombre,String direccion,int
d1,int d2,int d3,int d4,int d5,int d6,int d7,int estado){
```

```

try{
    conectar();
    proceso=con.prepareStatement(" { Call
MANTENIMIENTO_PUBLICIDAD(?,?,?,?,?,?,?,?,?,?,?) }");
    proceso.setInt(1,1);
    proceso.setString(2,nombre.toUpperCase());
    proceso.setString(3,direccion);
    proceso.setInt(4,d1);
    proceso.setInt(5,d2);
    proceso.setInt(6,d3);
    proceso.setInt(7,d4);
    proceso.setInt(8,d5);
    proceso.setInt(9,d6);
    proceso.setInt(10,d7);
    proceso.setInt(11,estado);
    proceso.setInt(12,0);
    proceso.execute();
    proceso.close();
    desconectar();
}catch(java.sql.SQLException dos){
    dos.printStackTrace();
    System.out.println(dos.getMessage());
}catch(Exception tres ){
    tres.printStackTrace();
    System.out.println(tres.getMessage()); } }

public void modificar_publicidad(String nombre,String
direccion,int d1,int d2,int d3,int d4,int d5,int d6,int d7,int
estado,int codigo){
    try{ conectar();
        proceso=con.prepareStatement(" { Call
MANTENIMIENTO_PUBLICIDAD(?,?,?,?,?,?,?,?,?,?,?) }");
        proceso.setInt(1,2);
        proceso.setString(2,nombre.toUpperCase());
        proceso.setString(3,"");
        proceso.setInt(4,d1);
        proceso.setInt(5,d2);
        proceso.setInt(6,d3);
        proceso.setInt(7,d4);
        proceso.setInt(8,d5);
        proceso.setInt(9,d6);
        proceso.setInt(10,d7);
        proceso.setInt(11,estado);
        proceso.setInt(12,codigo);
        proceso.execute();
        proceso.close();
        desconectar();
    }catch(java.sql.SQLException dos){
        dos.printStackTrace();
        System.out.println(dos.getMessage());
    }
}

```

```

        }catch(Exception tres ){
            tres.printStackTrace();
            System.out.println(tres.getMessage()); } }
    public void eliminar_publicidad(int codigo){
        try{ conectar();
            proceso=con.prepareCall("{ Call
MANTENIMIENTO_PUBLICIDAD(?,?,?,?,?,?,?,?,?,?,?,?,?)}");
            proceso.setInt(1,3);
            proceso.setString(2,"");
            proceso.setString(3,"");
            proceso.setInt(4,0);
            proceso.setInt(5,0);
            proceso.setInt(6,0);
            proceso.setInt(7,0);
            proceso.setInt(8,0);
            proceso.setInt(9,0);
            proceso.setInt(10,0);
            proceso.setInt(11,0);
            proceso.setInt(12,codigo);
            proceso.execute();
            proceso.close();
            desconectar();
        }catch(java.sql.SQLException dos){
            dos.printStackTrace();
            System.out.println(dos.getMessage());
        }catch(Exception tres ){
            tres.printStackTrace();
            System.out.println(tres.getMessage()); } }
    public void recibida_publicidad(int codigo){
        try{ conectar();
            proceso=con.prepareCall("{ Call
MANTENIMIENTO_PUBLICIDAD(?,?,?,?,?,?,?,?,?,?,?,?,?)}");
            proceso.setInt(1,4);
            proceso.setString(2,"");
            proceso.setString(3,"");
            proceso.setInt(4,0);
            proceso.setInt(5,0);
            proceso.setInt(6,0);
            proceso.setInt(7,0);
            proceso.setInt(8,0);
            proceso.setInt(9,0);
            proceso.setInt(10,0);
            proceso.setInt(11,0);
            proceso.setInt(12,codigo);
            proceso.execute();
            proceso.close();
            desconectar();
        }catch(java.sql.SQLException dos){
            dos.printStackTrace();

```

```

        System.out.println(dos.getMessage());
    }catch(Exception tres ){
        tres.printStackTrace();
        System.out.println(tres.getMessage()); } }
public void ignorada_publicidad(int codigo){
    try{ conectar();
        proceso=con.prepareCall("{ Call
MANTENIMIENTO_PUBLICIDAD(?,?,?,?,?,?,?,?,?,?,?,?,?)}");
        proceso.setInt(1,5);
        proceso.setString(2,"");
        proceso.setString(3,"");
        proceso.setInt(4,0);
        proceso.setInt(5,0);
        proceso.setInt(6,0);
        proceso.setInt(7,0);
        proceso.setInt(8,0);
        proceso.setInt(9,0);
        proceso.setInt(10,0);
        proceso.setInt(11,0);
        proceso.setInt(12,codigo);
        proceso.execute();
        proceso.close();
        desconectar();
    }catch(java.sql.SQLException dos){
        dos.printStackTrace();
        System.out.println(dos.getMessage());
    }catch(Exception tres ){
        tres.printStackTrace();
        System.out.println(tres.getMessage()); } }
public Vector Consulta_publicidad(){
    Vector arreglo =new Vector();
    try{
        conectar();
        proceso=con.prepareCall("{ Call VISTAS(?,?)}");
        proceso.setInt(1,7);
        proceso.setString(2,"");
        proceso.execute();
        while(proceso.getResultSet().next()){
            Vector objeto=new Vector();
            objeto.addElement(proceso.getResultSet().getString("id"));
            objeto.addElement(proceso.getResultSet().getString("nombre"));
            objeto.addElement(proceso.getResultSet().getString("estado"));
            arreglo.addElement(objeto);
        }
        proceso.close(); desconectar();
    }catch(java.sql.SQLException dos){
        dos.printStackTrace();
        System.out.println(dos.getMessage());
    }catch(Exception tres ){
        tres.printStackTrace();
    }
}

```

```

        System.out.println(tres.getMessage()); }
    return arreglo;    }
    public Vector Consultas_publicidad(int numero){
        Vector arreglo =new Vector();
        try{    conectar();
        proceso=con.prepareCall("{ Call VISTAS(?,?)}");
        proceso.setInt(1,10);
        proceso.setInt(2,numero);
        proceso.execute();
        while(proceso.getResultSet().next()){
            Vector objeto=new Vector();
            objeto.addElement(proceso.getResultSet().getString("nombre"));
            objeto.addElement(proceso.getResultSet().getString("d1"));
            objeto.addElement(proceso.getResultSet().getString("d2"));
            objeto.addElement(proceso.getResultSet().getString("d3"));
            objeto.addElement(proceso.getResultSet().getString("d4"));
            objeto.addElement(proceso.getResultSet().getString("d5"));
            objeto.addElement(proceso.getResultSet().getString("d6"));
            objeto.addElement(proceso.getResultSet().getString("d7"));
            objeto.addElement(proceso.getResultSet().getString("estado"));
            arreglo.addElement(objeto); }
        proceso.close();    desconectar();
    }catch(java.sql.SQLException dos){
        dos.printStackTrace();
        System.out.println(dos.getMessage());
    }catch(Exception tres ){
        tres.printStackTrace();
        System.out.println(tres.getMessage()); }
    return arreglo; }
    public String publicidad(int consulta,int numero){
        String arreglo ="";
        try{    conectar();
        proceso=con.prepareCall("{ Call VISTAS(?,?)}");
        proceso.setInt(1,consulta);
        proceso.setInt(2,numero);
        proceso.execute();
        while(proceso.getResultSet().next()){
            arreglo=proceso.getResultSet().getString("numero");    }
        proceso.close();
        desconectar();
    }catch(java.sql.SQLException dos){
        dos.printStackTrace();
        System.out.println(dos.getMessage());
    }catch(Exception tres ){
        tres.printStackTrace();
        System.out.println(tres.getMessage())}
    return arreglo;    }    }

```

Clases

Class Publicidad

```
java.lang.Object
├─ Clases.Conexion
└─ Clases.Publicidad
```

```
public class Publicidad
extends Conexion
```

Resumen constructor	
<u>Publicidad()</u>	

Resumen de método	
java.util.Vector	<u>Consulta_publicidad()</u>
java.util.Vector	<u>Consultas_publicidad</u> (int numero)
void	<u>eliminar_publicidad</u> (int codigo)
void	<u>ignorada_publicidad</u> (int codigo)
void	<u>ingresar_publicidad</u> (java.lang.String nombre, java.lang.String direccion, int d1, int d2, int d3, int d4, int d5, int d6, int d7, int estado)
void	<u>modificar_publicidad</u> (java.lang.String nombre, java.lang.String direccion, int d1, int d2, int d3, int d4, int d5, int d6, int d7, int estado, int codigo)
java.lang.String	<u>publicidad</u> (int consulta, int numero)
void	<u>recibida_publicidad</u> (int codigo)
Métodos heredados de la clase Clases. <u>Conexion</u>	
conectar, Consultas_objeto, Consultas, desconectar, getConexion, main, mantenimiento, verificaconsultas	
Métodos heredados de la clase java.lang.Object	
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait	

Constructor Detalle**Publicidad**

```
public Publicidad()
```

Método Detalle**ingresar_publicidad**

```
public void ingresar_publicidad(java.lang.String nombre,  
                                java.lang.String direccion,  
                                int d1,  
                                int d2,  
                                int d3,  
                                int d4,  
                                int d5,  
                                int d6,  
                                int d7,  
                                int estado)
```

modificar_publicidad

```
public void modificar_publicidad(java.lang.String nombre,  
                                java.lang.String direccion,  
                                int d1,  
                                int d2,  
                                int d3,  
                                int d4,  
                                int d5,  
                                int d6,  
                                int d7,  
                                int estado,  
                                int codigo)
```

eliminar_publicidad

```
public void eliminar_publicidad(int codigo)
```

recibida_publicidad

```
public void recibida_publicidad(int codigo)
```

ignorada_publicidad

```
public void ignorada_publicidad(int codigo)
```

Consulta_publicidad**public java.util.Vector Consulta_publicidad()**

Consultas_publicidad**public java.util.Vector Consultas_publicidad(int numero)**

publicidad**public java.lang.String publicidad(int consulta, int numero)**

3.11 CLASE “usuarios.java”

Esta clase permite realizar un mantenimiento de la tabla usuarios en la base de datos, la consulta, ingresos, modificación y eliminación de usuarios.

```

package Clases;
import java.sql.*;
import java.util.Vector;
/**
 *
 * author grupo 13a
 */
public class Usuarios extends Conexion {
    CallableStatement proceso = null;
    public Boolean verifica_celular(String celular){
        Boolean objeto=null;
        try{ conectar();
            objeto= verificaconsultas("select * from usuarios where
celular='"+celular+"'");
            desconectar();
        }catch(java.sql.SQLException dos){
            System.out.println(dos.getMessage());
        }catch(Exception tres ){
            System.out.println(tres.getMessage());
        }
        return objeto; }
    public Boolean verifica_usuarios(String usuario){
        Boolean objeto=null;
        try{conectar();
            objeto= verificaconsultas("select * from usuarios where
usuario='"+usuario+"'");
            desconectar();
        }catch(java.sql.SQLException dos){
            System.out.println(dos.getMessage());

```

```

        }catch(Exception tres ){
            System.out.println(tres.getMessage());}
        return objeto; }
    public Boolean verificar_usuarios(String usuario,String codigo){
        Boolean objeto=null;
        try{ conectar();
            objeto= verificaconsultas("select * from usuarios where
usuario='"+usuario+"' and not(codigo='"+codigo+"' )");
            desconectar();
        }catch(java.sql.SQLException dos){
            System.out.println(dos.getMessage());
        }catch(Exception tres ){
            System.out.println(tres.getMessage()); }
        return objeto;}
    public void ingresar_usuarios(String nombre,String apellido,String
celular,String usuario, String clave,String permiso,String correo){
        try{ conectar();
            proceso=con.prepareCall("{ Call
MANTENIMIENTO_USUARIOS(?,?,?,?,?,?,?,?,?,?)}");
            proceso.setString(1,nombre.toUpperCase());
            proceso.setString(2,apellido.toUpperCase());
            proceso.setString(4,celular);
            proceso.setString(3,usuario);
            proceso.setString(5,clave);
            proceso.setInt(6, (new Integer(permiso).intValue()));
            proceso.setInt(7, 0);
            proceso.setInt(8, 1);
            proceso.setString(9,correo);
            proceso.execute();
            proceso.close();
            desconectar();
        }catch(java.sql.SQLException dos){
            dos.printStackTrace();
            System.out.println(dos.getMessage());
        }catch(Exception tres ){
            tres.printStackTrace();
            System.out.println(tres.getMessage()); }
    }
    public void eliminar_usuarios(String codigo){
// public void eliminar_usuarios(){
        try{ conectar();
            proceso=con.prepareCall("{ Call
MANTENIMIENTO_USUARIOS(?,?,?,?,?,?,?,?,?,?)}");
            proceso.setString(1,"");
            proceso.setString(2,"");
            proceso.setString(3,"");
            proceso.setString(4,"");
            proceso.setString(5,"");
            proceso.setInt(6, 0);

```

```

        proceso.setInt(7, Integer.parseInt(codigo));
        proceso.setInt(8, 3);
        proceso.setString(9, "");
        proceso.execute();
        proceso.close();
        desconectar();
    } catch (java.sql.SQLException dos) {
        System.out.println(dos.getMessage());
    } catch (Exception tres) {
        System.out.println(tres.getMessage()); } }
public void modificar_usuarios(String nombre,String apellido,String
celular,String usuario, String clave,String codigo,String
permiso,String correo){
    try{ conectar();
        proceso=con.prepareStatement("{ Call
MANTENIMIENTO_USUARIOS(?,?,?,?,?,?,?,?,?)}");
        proceso.setString(1,nombre.toUpperCase());
        proceso.setString(2,apellido.toUpperCase());
        proceso.setString(3,usuario);
        proceso.setString(5,clave);
        proceso.setString(4,celular);
        proceso.setInt(6, (new Integer(permiso).intValue()));
        proceso.setInt(7, Integer.parseInt(codigo));
        proceso.setInt(8, 2);
        proceso.setString(9,correo);
        proceso.execute();
        proceso.close();
    } catch (java.sql.SQLException dos) {
        System.out.println(dos.getMessage());
    } catch (Exception tres) {
        System.out.println(tres.getMessage()); } }
public String datos_personas(String codigo){
    String nombres="";
    try{ conectar();
        proceso=con.prepareStatement("{ Call VISTAS(?,?)}");
        proceso.setInt(1,1);
        proceso.setString(2,codigo);
        proceso.execute();
        while(proceso.getResultSet().next()){
            nombres=proceso.getResultSet().getString("nombres")+
"+proceso.getResultSet().getString("apellidos");}
    } catch (java.sql.SQLException dos) {
        System.out.println(dos.getMessage());
    } catch (Exception tres) {
        System.out.println(tres.getMessage()); }
    return nombres;}
public Vector permisos_usuario(String usuario){
    Vector permiso=new Vector(10);
    try{ conectar();

```

```

        proceso=con.prepareCall("{ Call VISTAS(?,?)}");
        proceso.setInt(1,2);
        proceso.setString(2,usuario);
        proceso.execute();
        while(proceso.getResultSet().next()){
permiso.addElement(proceso.getResultSet().getString("usua_ing"));
permiso.addElement(proceso.getResultSet().getString("usua_con"));
permiso.addElement(proceso.getResultSet().getString("usua_inge"));
permiso.addElement(proceso.getResultSet().getString("usua_cone"));
permiso.addElement(proceso.getResultSet().getString("pri_ing"));
permiso.addElement(proceso.getResultSet().getString("pri_con"));
permiso.addElement(proceso.getResultSet().getString("dis_con"));
permiso.addElement(proceso.getResultSet().getString("dis_env"));
permiso.addElement(proceso.getResultSet().getString("rep_1"));
permiso.addElement(proceso.getResultSet().getString("rep_2"));
permiso.addElement(proceso.getResultSet().getString("rep_3"));
permiso.addElement(proceso.getResultSet().getString("rep_4"));
permiso.addElement(proceso.getResultSet().getString("rep_5"));
permiso.addElement(proceso.getResultSet().getString("rep_6"));
permiso.addElement(proceso.getResultSet().getString("rep_7"));
permiso.addElement(proceso.getResultSet().getString("rep_8"));
permiso.addElement(proceso.getResultSet().getString("pro_ing"));
permiso.addElement(proceso.getResultSet().getString("pro_con"));
        }
        desconectar();
    }catch(java.sql.SQLException dos){
        System.out.println(dos.getMessage());
    }catch(Exception tres ){
        System.out.println(tres.getMessage());
    }
    return permiso; }

```

Clases

Class Usuarios

java.lang.Object

└ Clases.Conexion

└ Clases.Usuarios

public class **Usuarios**

extends Conexion

Resumen constructor

Usuarios()

Resumen de método	
java.lang.String	<u>datos_personas</u> (java.lang.String codigo)
void	<u>eliminar_usuarios</u> (java.lang.String codigo)
void	<u>ingresar_usuarios</u> (java.lang.String nombre, java.lang.String apellido, java.lang.String celular, java.lang.String usuario, java.lang.String clave, java.lang.String permiso, java.lang.String correo)
void	<u>modificar_usuarios</u> (java.lang.String nombre, java.lang.String apellido, java.lang.String celular, java.lang.String usuario, java.lang.String clave, java.lang.String codigo, java.lang.String permiso, java.lang.String correo)
java.util.Vector	<u>permisos_usuario</u> (java.lang.String usuario)
java.lang.Boolean	<u>verifica_celular</u> (java.lang.String celular)
java.lang.Boolean	<u>verifica_usuarios</u> (java.lang.String usuario)
java.lang.Boolean	<u>verificar_usuarios</u> (java.lang.String usuario, java.lang.String codigo)

Métodos heredados de la clase Clases.Conexion

conectar, Consultas_objeto, Consultas, desconectar, getConexion, main, mantenimiento, verificaconsultas

Métodos heredados de la clase java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detalle

Usuarios

public **Usuarios**()

Método Detalle

verifica_celular

public java.lang.Boolean **verifica_celular**(java.lang.String celular)

verifica_usuarios

public java.lang.Boolean **verifica_usuarios**(java.lang.String usuario)

verificar_usuarios

```
public java.lang.Boolean verificar_usuarios(java.lang.String usuario,
                                             java.lang.String codigo)
```

ingresar_usuarios

```
public void ingresar_usuarios(java.lang.String nombre,
                               java.lang.String apellido,
                               java.lang.String celular,
                               java.lang.String usuario,
                               java.lang.String clave,
                               java.lang.String permiso,
                               java.lang.String correo)
```

eliminar_usuarios

```
public void eliminar_usuarios(java.lang.String codigo)
```

modificar_usuarios

```
public void modificar_usuarios(java.lang.String nombre,
                               java.lang.String apellido,
                               java.lang.String celular,
                               java.lang.String usuario,
                               java.lang.String clave,
                               java.lang.String codigo,
                               java.lang.String permiso,
                               java.lang.String correo)
```

datos_personas

```
public java.lang.String datos_personas(java.lang.String codigo)
```

permisos_usuario

```
public java.util.Vector permisos_usuario(java.lang.String usuario)
```

4 . DISPOSITIVOS

4.1 CLASE “Busca_Servicio.java”

Esta clase permite obtener la ruta de envio de los dispositivos detectados por la aplicación.

```

package Dispositivos;
import java.io.IOException;
import java.util.Enumeration;
import java.util.Vector;
import javax.bluetooth.*;
/**
 *
 * author grupo 13a
 */
public class Busca_Servicio {
    static final UUID OBEX_OBJECT_PUSH = new UUID(0x1105);
    String direccion="";
    public static void main(String[] args) throws IOException,
    InterruptedException {
        Busca_Servicio objeto=new Busca_Servicio();
        objeto.buscar("001ADC8DB2A1");}
    public String buscar(String dis) throws IOException,
    InterruptedException{    UUID serviceUUID = OBEX_OBJECT_PUSH;
        if ((dis != null) && (dis.length() > 0)) {
            serviceUUID = new UUID(dis, false);    }
        final Object serviceSearchCompletedEvent = new Object();
        DiscoveryListener listener = new DiscoveryListener() {
            public void deviceDiscovered(RemoteDevice btDevice,
DeviceClass cod) {
                }public void inquiryCompleted(int discType) {    }
        public void servicesDiscovered(int transID, ServiceRecord[] servRecord)
        {    for (int i = 0; i < servRecord.length; i++) {    String url =
servRecord[i].getConnectionURL(ServiceRecord.NOAUTHENTICATE_NOENCRYPT,
false);    if (url == null) {
                }else{ direccion= url;}}}    }
        public void serviceSearchCompleted(int transID, int respCode) {
            System.out.println("service search completed!");
            synchronized(serviceSearchCompletedEvent){
                serviceSearchCompletedEvent.notifyAll();
            }
        };
        UUID[] searchUuidSet = new UUID[] { serviceUUID };
        int[] attrIDs = new int[] {
            0x0100 // Service name};
        for(Enumeration en =RemoteDeviceDiscovery.devicesDiscovered.elements();
en.hasMoreElements(); ) {
            RemoteDevice btDevice = (RemoteDevice)en.nextElement();
            synchronized(serviceSearchCompletedEvent) {
                System.out.println("search services on " +
btDevice.getBluetoothAddress() + " " +
btDevice.getFriendlyName(false));
                LocalDevice.getLocalDevice().getDiscoveryAgent().searchServices(attrIDs
, searchUuidSet, btDevice, listener);
                serviceSearchCompletedEvent.wait();
            }
        }
    }
}

```

```
    } return direccion; } }
```

Dispositivos

Class Busca_Servicio

java.lang.Object

└─Dispositivos.Busca_Servicio

```
public class Busca_Servicio
extends java.lang.Object
```

Resumen constructor

Busca_Servicio()

Resumen de método

java.lang.String **buscar**(java.lang.String dis)

static void **main**(java.lang.String[] args)

Métodos heredados de la clase java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detalle

Busca_Servicio

public **Busca_Servicio()**

Método Detalle

main

```
public static void main(java.lang.String[] args)
    throws java.io.IOException,
           java.lang.InterruptedException
```

Throws:

java.io.IOException

java.lang.InterruptedException

buscar

public java.lang.String **buscar**(java.lang.String dis)
 throws java.io.IOException,
 java.lang.InterruptedException

Throws:

java.io.IOException
 java.lang.InterruptedException

4.2 CLASE “Busca_Servicios.java”

Permite buscar la ruta de envío de un conjunto de dispositivos detectados por la aplicación.

```
package Dispositivos;
/**
 *
 * author grupo 13a
 */
import java.io.*;
public class Busca_Servicios {
public String ver_servicios(String dispositivos){
    Process proceso=null;
    String error="";
    String aux="";
    String s="";
    String comando="";
    try {
        comando="sdptool browse "+dispositivos;
        proceso=Runtime.getRuntime().exec(comando);
        proceso.waitFor();
        InputStream is = proceso.getInputStream();
        BufferedReader br9 = new BufferedReader(new InputStreamReader(is));
        aux = br9.readLine();
        while (aux!=null){
            s=s+aux;
            aux=br9.readLine();}
    }catch (IOException e) {
        error="Exception el comando";
    } catch (InterruptedException e) {
        error="Interrupcion del comando";
    } return s;    } }
```

Dispositivos

Class Busca_Servicios

java.lang.Object

└ **Dispositivos.Busca_Servicios**

public class **Busca_Servicios**
extends java.lang.Object

Resumen constructor

Busca_Servicios()

Resumen de método

java.lang.String **ver_servicios**(java.lang.String dispositivos)

Métodos heredados de la clase java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detalle

Busca_Servicios

public **Busca_Servicios()**

Método Detalle

ver_servicios

public java.lang.String **ver_servicios**(java.lang.String dispositivos)

4.3 CLASE “Buscar_Servicios.java”

Aparte de que llama el servicio también hace una retroalimentación de la misma clase utilizando clases abstractas.

```

package Dispositivos;
import java.io.IOException;
import java.sql.SQLException;
import java.util.Enumeration;
import java.util.Vector;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.bluetooth.*;

public class Buscar_Servicios extends Clases.Dispositivos{
    static final UUID OBEX_FILE_TRANSFER = new UUID(0x1106);
    static final UUID OBEX_OBJECT_PUSH = new UUID(0x1105);
    static String envia="";
    public static final Vector/**/ serviceFound = new Vector();
    public Vector iniciar(Vector datos){
        UUID serviceUUID = OBEX_OBJECT_PUSH;
        serviceFound.clear();
        final Object serviceSearchCompletedEvent = new Object();
        DiscoveryListener listener = new DiscoveryListener() {
        public void deviceDiscovered(RemoteDevice btDevice, DeviceClass cod) {
            }
        public void inquiryCompleted(int discType) {
            }
        public void servicesDiscovered(int transID, ServiceRecord[] servRecord)
        {
            for (int i = 0; i < servRecord.length; i++) {
                String url =
servRecord[i].getConnectionURL(ServiceRecord.NOAUTHENTICATE_NOENCRYPT,
false);

                if (url == null) {
                    continue;
                }else{
                    envia=url;
                    int j=url.indexOf("//");
                    int jj=url.indexOf(":",j);
                    String dispositivos=url.substring(j+2, jj);
                    graba_obex(dispositivos,url);
                    serviceFound.add(url); } } }
        public void serviceSearchCompleted(int transID, int respCode) {
            System.out.println("service search completed!");
            synchronized(serviceSearchCompletedEvent){
                serviceSearchCompletedEvent.notifyAll();}
        }
    };
    UUID[] searchUuidSet = new UUID[] { serviceUUID };
    int[] attrIDs = new int[] {
        0x0100 // Service name };
    for(Enumeration en = datos.elements(); en.hasMoreElements(); ) {
        RemoteDevice btDevice = (RemoteDevice)en.nextElement();
        synchronized(serviceSearchCompletedEvent) {
            try {
LocalDevice.getLocalDevice().getDiscoveryAgent().searchServices(attrIDs
, searchUuidSet, btDevice, listener);
            } catch (BluetoothStateException ex) {

```

```

Logger.getLogger(Buscar_Servicios.class.getName()).log(Level.SEVERE,
null, ex); } try {
    serviceSearchCompletedEvent.wait();
} catch (InterruptedException ex) {
Logger.getLogger(Buscar_Servicios.class.getName()).log(Level.SEVERE,
null, ex); } } return serviceFound; } }

```

Dispositivos

Class Buscar_Servicios

```

java.lang.Object
├─ Clases.Conexion
├─ Clases.Dispositivos
└─ Dispositivos.Buscar_Servicios

```

```

public class Buscar_Servicios
extends Dispositivos

```

Resumen de Campo

static java.util.Vector	<u>serviceFound</u>
-------------------------	----------------------------

Resumen constructor

<u>Buscar_Servicios()</u>	
----------------------------------	--

Resumen de método

java.util.Vector	<u>iniciar</u> (java.util.Vector datos)
------------------	--

Métodos heredados de la clase Clases.Dispositivos

Consulta_acceso, Consulta_borradores, Consulta_dispositivos, Consulta_mensaje, Consulta_mensajes, Consulta_mensajes2, Consulta_Otros, devuelve_ruta, formatea, graba_obex, ingresar_dispositivos, registra_mensaje, uso_dispositivos

Métodos heredados de la clase Clases.Conexion

conectar, Consultas_objeto, Consultas, desconectar, getConexion, main, mantenimiento, verificaconsultas

Métodos heredados de la clase java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Detalle campo

serviceFound

public static final java.util.Vector **serviceFound**

Constructor Detalle

Buscar_Servicios

public **Buscar_Servicios**()

Método Detalle

iniciar

public java.util.Vector **iniciar**(java.util.Vector datos)

4.4 CLASE “Envia_Mensaje.java”

Esta clase permite enviar los mensajes a los dispositivos bluetooth detectados.

```
Envia los mensajes
package Dispositivos;
import java.io.IOException;
import java.io.OutputStream;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.microedition.io.Connector;
import javax.obex.*;
import java.lang.*;
import java.sql.SQLException;
/**
 *
 * author grupo 13a
 */
public class Envia_Mensaje extends Clases.Dispositivos {
    public Boolean envia(String mensaje,String dispositivo){
        boolean respuesta=true;
        String direccion="";
```

```

try{
    ClientSession clientSession = (ClientSession)
Connector.open(dispositivo);
    HeaderSet hsConnectReply = clientSession.connect(null);
    if (hsConnectReply.getResponseCode() !=
ResponseCodes.OBEX_HTTP_OK) {
        System.out.println("Failed to connect");
        respuesta = false;
        registra_mensaje(dispositivo,mensaje,"2");
        return (new Boolean(respuesta));    }
    HeaderSet hsOperation = clientSession.createHeaderSet();
    hsOperation.setHeader(HeaderSet.NAME, "Envio_Servidor.txt");
    hsOperation.setHeader(HeaderSet.TYPE, "text");
    hsOperation.setHeader(HeaderSet.LENGTH, new
Long(mensaje.length()));
    //Create PUT Operation
    Operation putOperation = clientSession.put(hsOperation);
    // Send some text to server
    byte data[] = mensaje.getBytes();
    OutputStream os = putOperation.openOutputStream();
    os.write(data);
    for(int i=0;i<5000;i++){        }
    os.close();
    putOperation.close();
    clientSession.disconnect(null);
    clientSession.close();
    direccion=dispositivo.substring(9,dispositivo.indexOf(":",8));
    uso_dispositivos("",dispositivo);
    registra_mensaje(dispositivo,mensaje,"1");
    try {
        Clases.Dispositivos objeto= new Clases.Dispositivos();
        objeto.conectar();
        objeto.mantenimiento("update dispositivos set
fecha_acceso=curdate() where ruta='" + dispositivo + "'");
        objeto.desconectar();
    } catch (SQLException ex) {
        Logger.getLogger(Envia_Mensaje.class.getName()).log(Level.SEVERE, null,
ex);
    } catch (Exception ex) {
        Logger.getLogger(Envia_Mensaje.class.getName()).log(Level.SEVERE, null,
ex);
    }
    } catch(IOException e){
        e.printStackTrace();
        respuesta=false;
        registra_mensaje(dispositivo,mensaje,"2");    }
    return (new Boolean(respuesta));    }
    public static void main(String[] args) throws IOException,
InterruptedException {
        Envia_Mensaje objeto=new Envia_Mensaje();
        objeto.envia("hola a todos","001ADC8DB2A1");    } }

```

Dispositivos

Class Envia_Mensaje

java.lang.Object

└ Clases.Conexion

└ Clases.Dispositivos

└ Dispositivos.Envia_Mensaje

public class **Envia_Mensaje**

extends Dispositivos

Resumen constructor

Envia_Mensaje()

Resumen de método

java.lang.Boolean	<u>envia</u> (java.lang.String mensaje, ava.lang.String dispositivo)
-------------------	---

static void	<u>main</u> (java.lang.String[] args)
-------------	---------------------------------------

Métodos heredados de la clase Clases.Dispositivos

Consulta_acceso, Consulta_borradores, Consulta_dispositivos,
Consulta_mensaje, Consulta_mensajes, Consulta_mensajes2,
Consulta_Otros, devuelve_ruta, formatea, graba_obex,
ingresar_dispositivos, registra_mensaje, uso_dispositivos

Métodos heredados de la clase Clases.Conexion

conectar, Consultas_objeto, Consultas, desconectar, getConexion,
mantenimiento, verificaconsultas

Métodos heredados de la clase java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait,
wait, wait

Constructor Detalle**Envia_Mensaje**

```
public Envia_Mensaje()
```

Método Detalle**envia**

```
public java.lang.Boolean envia(java.lang.String mensaje,  
                                java.lang.String dispositivo)
```

main

```
public static void main(java.lang.String[] args)  
    throws java.io.IOException,  
           java.lang.InterruptedException
```

Throws:

```
java.io.IOException
```

```
java.lang.InterruptedException
```

4.5 CLASE “RemoteDeviceDiscovery.java”

Esta clase permite enviar los mensajes a los dispositivos bluetooth detectados desde la pagina.

```
/**
 * BlueCove - Java library for Bluetooth
 */
package Dispositivos;
import java.io.IOException;
import java.util.Vector;
import javax.bluetooth.*;
/**
 * author grupo 13a
 *
 * Minimal DeviceDiscovery example for javadoc.
 */
public class RemoteDeviceDiscovery extends Clases.Dispositivos {
    public static final Vector/*<RemoteDevice>*/ devicesDiscovered =
new Vector();
    public Vector nuevo=new Vector();
    public Vector busca_dispositivos(Vector nuevos)throws IOException,
InterruptedException {
```



```

        final Object inquiryCompletedEvent = new Object();
        nuevo=nuevos;
        nuevo.clear();
        devicesDiscovered.clear();
        DiscoveryListener listener = new DiscoveryListener() {
            public void deviceDiscovered(RemoteDevice btDevice, DeviceClass cod) {
                System.out.println("Device " + btDevice.getBluetoothAddress() + "
found");
            }
        };
        devicesDiscovered.addElement(btDevice);
        try {
            if(ingresar_dispositivos(btDevice.getFriendlyName(false).toString(),btD
evice.getBluetoothAddress().toString(),cod.getMajorDeviceClass(),cod.ge
tMinorDeviceClass()).booleanValue()==true){
                nuevo.addElement(btDevice); }
        } catch (IOException cantGetDeviceName) {
        }
    }
    public void inquiryCompleted(int discType) {
        System.out.println("Device Inquiry completed!");
        synchronized(inquiryCompletedEvent){
            inquiryCompletedEvent.notifyAll(); }
        public void serviceSearchCompleted(int transID, int respCode) { }
        public void servicesDiscovered(int transID, ServiceRecord[] servRecord)
        {
            };
        synchronized(inquiryCompletedEvent) {
            boolean started =
LocalDevice.getLocalDevice().getDiscoveryAgent().startInquiry(Discovery
Agent.GIAC, listener);
            if (started) {
                System.out.println("wait for device inquiry to complete...");
                inquiryCompletedEvent.wait();
                System.out.println(devicesDiscovered.size() + " device(s) found");}}
            return devicesDiscovered; }
        public static void main(String[] args) throws IOException,
InterruptedException {
            RemoteDeviceDiscovery objecto = new RemoteDeviceDiscovery();
            Vector uno =new Vector(); } }

```

Dispositivos

4.5.1 Class RemoteDeviceDiscovery

java.lang.Object

└ Clases.Conexion

└ Clases.Dispositivos

└ Dispositivos.RemoteDeviceDiscovery

```
public class RemoteDeviceDiscovery
extends Dispositivos
```

Resumen de Campo

static java.util.Vector	<u>devicesDiscovered</u>
java.util.Vector	<u>nuevo</u>

Resumen constructor

RemoteDeviceDiscovery()

Resumen de método

java.util.Vector	<u>busca_dispositivos</u> (java.util.Vector nuevos)
static void	<u>main</u> (java.lang.String[] args)

Métodos heredados de la clase Clases.Dispositivos

Consulta_acceso, Consulta_borradores, Consulta_dispositivos, Consulta_mensaje, Consulta_mensajes, Consulta_mensajes2, Consulta_Otros, devuelve_ruta, formatea, graba_obex, ingresar_dispositivos, registra_mensaje, uso_dispositivos

Métodos heredados de la clase Clases.Conexion

conectar, Consultas_objeto, Consultas, desconectar, getConexion, mantenimiento, verificaconsultas

Métodos heredados de la clase java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Detalle campo

devicesDiscovered

public static final java.util.Vector **devicesDiscovered**

nuevo

```
public java.util.Vector nuevo
```

Constructor Detalle

RemoteDeviceDiscovery

```
public RemoteDeviceDiscovery()
```

Método Detalle

busca_dispositivos

```
public java.util.Vector busca_dispositivos(java.util.Vector nuevos)
    throws java.io.IOException,
        java.lang.InterruptedIOException
```

Throws:

java.io.IOException

```
java.lang.InterruptedException
```

main

```
public static void main(java.lang.String[] args)
    throws java.io.IOException,
        java.lang.InterruptedException
```

Throws:

java.io.IOException

```
java.lang.InterruptedException
```

4.6 CLASE “ServicesSearch.java”

Esta clase permite buscar los servicios de los dispositivos que estan dentro del alcance de la aplicación.

```
/**
 * BlueCove - Java library for Bluetooth
 * Copyright (C) 2006-2007 Vlad Skarzhevskyy
 *
 */

package Dispositivos;
import java.io.IOException;
import java.util.Enumeration;
import java.util.Vector;
import javax.bluetooth.*;
```

```

/**
 * author grupo 13a
 *
 * Minimal ServicesSearch example for javadoc.
 */
public class ServicesSearch {
    static final UUID OBEX_OBJECT_PUSH = new UUID(0x1105);
    public static final Vector/*<String>*/ serviceFound = new Vector();
    public static void main(String[] args) throws IOException,
    InterruptedException {
        // First run RemoteDeviceDiscovery and use discovered device
        RemoteDeviceDiscovery.main(null);
        serviceFound.clear();
        UUID serviceUUID = OBEX_OBJECT_PUSH;
        if ((args != null) && (args.length > 0)) {
            serviceUUID = new UUID(args[0], false);
        }
        final Object serviceSearchCompletedEvent = new Object();
        DiscoveryListener listener = new DiscoveryListener() {
            public void deviceDiscovered(RemoteDevice btDevice, DeviceClass cod){}
            public void inquiryCompleted(int discType) {}
            public void servicesDiscovered(int transID, ServiceRecord[] servRecord)
            {
                for (int i = 0; i < servRecord.length; i++) {
                    String url =
servRecord[i].getConnectionURL(ServiceRecord.NOAUTHENTICATE_NOENCRYPT,
false);

                    if (url == null) { continue; }
                    serviceFound.add(url);

                    DataElement serviceName = servRecord[i].getAttributeValue(0x0100);
                    if (serviceName != null) {
System.out.println("service " + serviceName.getValue() + " found " +
url);
                    } else {
                        System.out.println("service found " + url);
                    }
                }
            }
            public void serviceSearchCompleted(int transID, int respCode) {
                System.out.println("service search completed!");
                synchronized(serviceSearchCompletedEvent){
                    serviceSearchCompletedEvent.notifyAll();
                }
            }
        };
        UUID[] searchUuidSet = new UUID[] { serviceUUID };
        int[] attrIDs = new int[] {
            0x0100 // Service name
        };
        for(Enumeration en =
RemoteDeviceDiscovery.devicesDiscovered.elements();
en.hasMoreElements(); ) {
            RemoteDevice btDevice = (RemoteDevice)en.nextElement();

            synchronized(serviceSearchCompletedEvent) {
                System.out.println("search services on " +
btDevice.getBluetoothAddress() + " " +
btDevice.getFriendlyName(false));
            }
        }
    }
}

```

```
LocalDevice.getLocalDevice().getDiscoveryAgent().searchServices(attrIDs
, searchUuidSet, btDevice, listener);
    serviceSearchCompletedEvent.wait(); } } }}
```

Dispositivos

4.6.1 Class ServicesSearch

java.lang.Object

└─ **Dispositivos.ServicesSearch**

```
public class ServicesSearch
extends java.lang.Object
```

Resumen de Campo

static java.util.Vector **serviceFound**

Resumen constructor

ServicesSearch()

Resumen de método

static void **main**(java.lang.String[] args)

Métodos heredados de la clase java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Detalle campo

serviceFound

public static final java.util.Vector **serviceFound**

Constructor Detalle

ServicesSearch

public **ServicesSearch()**

Método Detalle**main**

```
public static void main(java.lang.String[] args)
    throws java.io.IOException,
           java.lang.InterruptedException
```

Throws:

```
java.io.IOException
java.lang.InterruptedException
```

5. CLASE COM.METAPARADIGM.JSONRPC

Esta clase es es utilizado en toda la pagina web y tiene como finalidad realizar un ajax asincronico desde la pagina hacia las clases java ya conpiladas directamente.

5.1 CLASE “JSONRPCCallback.java”

```
package com.metaparadigm.jsonrpc;

import javax.servlet.http.HttpSession;
import java.lang.reflect.Method;
/**
 * Interfaz para ser aplicado por los objetos registrados para llamadas
 * con la invocación JSONRPCBridge.
 */
public interface JSONRPCCallback {
    public void preInvoke(Object context, Object instance,
        Method m, Object arguments[])
        throws Exception;
    public void postInvoke(Object context, Object instance,
        Method m, Object result)
        throws Exception; }
```

com.metaparadigm.jsonrpc
Interface JSONRPCCallback

```
public interface JSONRPCCallback
Interface to be implemented by objects registered for invocation callbacks with
the JSONRPCBridge.
```

Resumen de método

void	postInvoke (java.lang.Object context, java.lang.Object instance, java.lang.reflect.Method m, java.lang.Object result)
void	preInvoke (java.lang.Object context, java.lang.Object instance, java.lang.reflect.Method m, java.lang.Object[] arguments)

Método Detalle

preInvoke

void **preInvoke**(java.lang.Object context,
 java.lang.Object instance,
 java.lang.reflect.Method m,
 java.lang.Object[] arguments)
 throws java.lang.Exception

Throws:

java.lang.Exception

postInvoke

void **postInvoke**(java.lang.Object context,
 java.lang.Object instance,
 java.lang.reflect.Method m,
 java.lang.Object result)
 throws java.lang.Exception

Throws:

java.lang.Exception

5.2 CLASE “JSONRPCServlet.java”

```
package com.metaparadigm.jsonrpc;
import java.io.IOException;
import java.io.OutputStream;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.io.BufferedReader;
import java.io.PrintWriter;
import java.io.CharArrayWriter;
import java.util.NoSuchElementException;
```

```

import javax.servlet.http.HttpSession;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.text.ParseException;
import org.json.JSONObject;
import org.json.JSONArray;
/**
 * Este servlet se encarga de las solicitudes JSON-RPC sobre HTTP y las
manos a
 * JSONRPCBridge un ejemplo registrado en el HttpSession.
 * </p>
 * Un ejemplo de la JSONRPCBridge objeto colocan automáticamente en la
 * HttpSession objeto registrado bajo el atributo "JSONRPCBridge" por
 * el JSONRPCServlet.
 * <p />
 * Los siguientes se pueden agregar a su web.xml para exportar el
servlet
 * Bajo la URI "<code> / JSON-RPC </ code>"
 * <p />
 * <code>
 * <!--servlet-->
 * <!--servlet-
name-->com.metaparadigm.jsonrpc.JSONRPCServlet</servlet-name-->
 * <!--servlet-
class-->com.metaparadigm.jsonrpc.JSONRPCServlet</servlet-class-->
 * </servlet-->
 * <!--servlet-mapping-->
 * <!--servlet-
name-->com.metaparadigm.jsonrpc.JSONRPCServlet</servlet-name-->
 * <!--url-pattern-->/JSON-RPC</url-pattern-->
 * </servlet-mapping-->
 * </code>
 */
public class JSONRPCServlet extends HttpServlet
{
    private final static int buf_size = 4096;

    public void service(HttpServletRequest request,
                        HttpServletResponse response)
        throws IOException, ClassCastException
    {
        // Encuentre el JSONRPCBridge para este período de sesiones o
//crear uno si no existe
        HttpSession session = request.getSession();
        JSONRPCBridge json_bridge = null;
        json_bridge = (JSONRPCBridge)
            session.getAttribute("JSONRPCBridge");
        if(json_bridge == null) {

```



```

        json_bridge = new JSONRPCBridge();
        session.setAttribute("JSONRPCBridge", json_bridge);
    }
    // Codifica con UTF-8, aunque en realidad ASCII Estamos limpios
    // Todos los datos Unicode se escapó JSON utilizando la barra u. Esto
    // es
    // Menos de datos eficiente para conjuntos de caracteres extranjeros,
    // pero es
    // Necesario para apoyar malo navegadores como Konqueror y Safari
    // Que no cumplen con la serie de caracteres en la respuesta
    response.setContentType("text/plain;charset=utf-8");
    OutputStream out = response.getOutputStream();
    // Descifrar utilizando el juego de caracteres en la solicitud si es
    // que existe otra cosa
    // Uso de UTF-8, ya que es lo que todas las implementaciones de uso
    // del navegador.
    // La JSON-RPC-Java JavaScript cliente es ASCII limpio por lo que
    // Aunque aquí podemos manejar correctamente los datos de otros
    // clientes
    // Que no escapan a los datos no ASCII String charset =
    request.setCharacterEncoding();
    if(charset == null) charset = "UTF-8";
    BufferedReader in = new BufferedReader
        (new InputStreamReader(request.getInputStream(), charset));
    // Leer la petición
    CharArrayWriter data = new CharArrayWriter();
    char buf[] = new char[buf_size];
    int ret;
    while((ret = in.read(buf, 0, buf_size)) != -1) {
        data.write(buf, 0, ret);
    }
    if(json_bridge.isDebugEnabled())
        System.out.println("JSONRPCServlet.service recv: " +
            data.toString());
    // Procesar la solicitud
    JSONObject json_req = null;
    Object json_res = null;
    try {
        json_req = new JSONObject(data.toString());

        // Obtener nombre de método y argumentos
        String methodName = null;
        JSONArray arguments = null;
        try { methodName = json_req.getString("method");
        } catch (NoSuchElementException ne) {}
        // Volver compatibilidad de <= 0,7 clientes
        if (methodName != null) {
            arguments = json_req.getJSONArray("params");
        } else {
            methodName = json_req.getString("methodName");

```

```

        arguments = json_req.getJSONArray("arguments");
        System.err.println("JSONRPCServlet.service: " +
            "methodName in request deprecated, " +
            "please update your JSON-RPC client.");
    }
    // ¿Se trata de un CallableReference tendrá un valor distinto
    de cero ObjectID
    int object_id = json_req.optInt("objectID");
    if(json_bridge.isDebugEnabled())
        if(object_id != 0)
            System.out.println("JSONRPCServlet.service call " +
                "objectID=" + object_id + " " +
                methodName + "(" + arguments + ")");
        else
            System.out.println("JSONRPCServlet.service call " +
                methodName + "(" + arguments + ")");
    json_res = json_bridge.call(session,
        object_id, methodName, arguments);
} catch (ParseException e) {
    System.err.println
        ("JSONRPCServlet.service can't parse call: " + data);
    json_res = JSONRPCResult.ERR_PARSE;
} catch (NoSuchElementException e) {
    System.err.println
        ("JSONRPCServlet.service no method in request");
    json_res = JSONRPCResult.ERR_NOMETHOD; }
// Escriba la respuesta
if(json_bridge.isDebugEnabled())
    System.out.println("JSONRPCServlet.service send: " +
        json_res.toString());
byte[] bout = json_res.toString().getBytes("UTF-8");
response.setIntHeader("Content-Length", bout.length);
response.setHeader("Connection", "keep-alive");
out.write(bout);
out.flush();
out.close();
}
}

```

5.2 com.metaparadigm.jsonrpc

Class JSONRPCServlet

java.lang.Object

└ javax.servlet.GenericServlet

└ javax.servlet.http.HttpServlet

└ **com.metaparadigm.jsonrpc.JSONRPCServlet**

Todas las interfaces implementadas:

java.io.Serializable, javax.servlet.Servlet, javax.servlet.ServletConfig

public class **JSONRPCServlet** extends javax.servlet.http.HttpServlet

This servlet handles JSON-RPC requests over HTTP and hands them to a JSONRPCBridge instance registered in the HttpSession.

An instance of the JSONRPCBridge object is automatically placed in the HttpSession object registered under the attribute "JSONRPCBridge" by the JSONRPCServlet.

The following can be added to your web.xml to export the servlet under the URI "/JSON-RPC"

```
<servlet> <servlet-name>com.metaparadigm.jsonrpc.JSONRPCServlet</servlet-
name> <servlet-class>com.metaparadigm.jsonrpc.JSONRPCServlet</servlet-
class> </servlet> <servlet-mapping>
<servlet-name>com.metaparadigm.jsonrpc.JSONRPCServlet</servlet-name>
<url-pattern>/JSON-RPC</url-pattern> </servlet-mapping>
```

See Also:

Serialized Form

Resumen constructor

JSONRPCServlet()

Resumen de método

void **service**(javax.servlet.http.HttpServletRequest request,
javax.servlet.http.HttpServletResponse response)

Métodos heredados de la clase javax.servlet.http.HttpServlet

doDelete, doGet, doHead, doOptions, doPost, doPut, doTrace,
getLastModified, service

Métodos heredados de la clase javax.servlet.GenericServlet

destroy, getInitParameter, getInitParameterNames, getServletConfig,
getServletContext, getServletInfo, getServletName, init, init, log, log

Métodos heredados de la clase java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait,
wait, wait

Constructor Detalle**JSONRPCServlet**

```
public JSONRPCServlet()
```

Método Detalle**service**

```
public void service(javax.servlet.http.HttpServletRequest request,
                    javax.servlet.http.HttpServletResponse response)
    throws java.io.IOException,
           java.lang.ClassCastException
```

Overrides:

service in class javax.servlet.http.HttpServlet

Throws:

java.io.IOException

java.lang.ClassCastException

5.3 CLASE ORG.JFREE.CHART.DEMO.SERVLET

Esta clase permite reializar los gráficos que son presentados en la pagina.

CLASE “NoDataException.java”

```
package org.jfree.chart.demo.servlet;
public class NoDataException extends Exception {
    public NoDataException() {
        super(); }
}
```

org.jfree.chart.demo.servlet**Class NoDataException**

java.lang.Object

└ java.lang.Throwable

└ java.lang.Exception

└ **org.jfree.chart.demo.servlet.NoDataException**

Todas las interfaces implementadas:

java.io.Serializable

```
public class NoDataException
```

```
extends java.lang.Exception
```

See Also:

[Serialized Form](#)

Resumen constructor**NoDataException()****Resumen de método****Métodos heredados de la clase java.lang.Throwable**

fillInStackTrace, getCause, getLocalizedMessage, getMessage,
 getStackTrace, initCause, printStackTrace, printStackTrace,
 printStackTrace, setStackTrace, toString

Métodos heredados de de la clase java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detalle**NoDataException**public **NoDataException()****5.4 CLASE “WebHit.java”**

```

package org.jfree.chart.demo.servlet;
import java.util.Date;
public class WebHit {
    protected Date hitDate = null;
    protected String section = null;
    protected long hitCount = 0;
    public WebHit(Date dHitDate, String sSection, long lHitCount) {
        this.hitDate = dHitDate;
        this.section = sSection;
        this.hitCount = lHitCount;    }
    public WebHit( String sSection, long lHitCount) {
        this.section = sSection;
        this.hitCount = lHitCount;    }
    public WebHit( Date dHitDate, long lHitCount) {
        this.hitDate = dHitDate;
        this.hitCount = lHitCount;    }
    public Date getHitDate() {return this.hitDate; }
    public String getSection() {
        return this.section;    }
}

```

```

public long getHitCount() {
    return this.hitCount;
}
public void setHitDate(Date dHitDate) {
    this.hitDate = dHitDate;
}
public void setSection(String sSection) {
    this.section = sSection;
}
public void setHitCount(long lHitCount) {
    this.hitCount = lHitCount;
}
}

```

org.jfree.chart.demo.servlet

Class WebHit

java.lang.Object

└ **org.jfree.chart.demo.servlet.WebHit**

public class **WebHit**
 extends java.lang.Object

Resumen de Campo

protected long	<u>hitCount</u>
protected java.util.Date	<u>hitDate</u>
protected java.lang.String	<u>section</u>

Resumen constructor

<u>WebHit</u> (java.util.Date dHitDate, long lHitCount)	
<u>WebHit</u> (java.util.Date dHitDate, java.lang.String sSection, long lHitCount)	
<u>WebHit</u> (java.lang.String sSection, long lHitCount)	

Resumen de método

long	<u>getHitCount</u> ()
java.util.Date	<u>getHitDate</u> ()
java.lang.String	<u>getSection</u> ()
void	<u>setHitCount</u> (long lHitCount)
void	<u>setHitDate</u> (java.util.Date dHitDate)
void	<u>setSection</u> (java.lang.String sSection)

Métodos heredados de la clase java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Detalle campo

hitDate

protected java.util.Date **hitDate**

section

protected java.lang.String **section**

hitCount

protected long **hitCount**

Constructor Detalle

WebHit

```
public WebHit(java.util.Date dHitDate,
               java.lang.String sSection,
               long lHitCount)
```

WebHit

```
public WebHit(java.lang.String sSection,
               long lHitCount)
```

WebHit

```
public WebHit(java.util.Date dHitDate,
               long lHitCount)
```

Método Detalle

getHitDate

```
public java.util.Date getHitDate()
```

getSection

```
public java.lang.String getSection()
```

getHitCount

```
public long getHitCount()
```

setHitDate

```
public void setHitDate(java.util.Date dHitDate)
```

setSection

```
public void setSection(java.lang.String sSection)
```

setHitCount

```
public void setHitCount(long lHitCount)
```

5.5 CLASE “WebHitChart.java”

```
package org.jfree.chart.demo.servlet;
import org.jdom.*;
import org.jdom.input.*;
import java.awt.*;
import java.awt.geom.Ellipse2D;
import java.io.PrintWriter;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.Locale;
import java.text.NumberFormat;
import javax.servlet.http.HttpSession;
import org.jfree.data.*;
import org.jfree.chart.*;
import org.jfree.chart.axis.*;
import org.jfree.chart.renderer.BarRenderer;
import org.jfree.chart.renderer.CategoryItemRenderer;
import org.jfree.chart.renderer.StandardXYItemRenderer;
import org.jfree.chart.renderer.StackedXYAreaRenderer;
import org.jfree.chart.renderer.XYAreaRenderer;
import org.jfree.chart.plot.*;
import org.jfree.chart.entity.*;
import org.jfree.chart.labels.*;
import org.jfree.chart.urls.*;
import org.jfree.chart.servlet.*;
import org.jfree.chart.labels.ItemLabelAnchor;
import org.jfree.chart.labels.ItemLabelPosition;
import org.jfree.ui.TextAnchor;
import java.util.*;
import java.io.*;
import org.jfree.util.Rotation;
public class WebHitChart {
    Document doc;
    public static String generateBarChart(String xml, HttpSession session,
    PrintWriter pw) {
        String filename = null;
        String fecha;
        String consumo;
        int inicio,fin;
```



```

        ArrayList data = new ArrayList();
    try {
        // Retrieve list of WebHits
        //WebHitDataSet whDataSet = new WebHitDataSet();
        //ArrayList list =
        whDataSet.getDataBySection(hitDate);
        System.out.println(xml.length());
        for(int h=0;h<xml.length();h++){
            if(xml.indexOf("<hora>",h)>0){
                inicio=xml.indexOf("<hora>",h)+6;
                // System.out.println("inicio fecha "+inicio);
                fin=xml.indexOf("</hora>",inicio);
                // System.out.println("fin fecha "+fin);
                fecha=xml.substring(inicio,fin);
                System.out.println(fecha);
                inicio=fin;
                inicio=xml.indexOf("<conexion>",inicio)+10;
                // System.out.println("inicio consumo "+inicio);
                fin=xml.indexOf("</conexion>",inicio);
                //System.out.println("fin consumo "+fin);
                consumo=xml.substring(inicio,fin);
                h=fin;
                System.out.println(consumo);
                data.add(new
WebHit(fecha,Long.parseLong(consumo)));
            }else{
                h=xml.length()+1;
            }
        }
        // Lanzar una costumbre NoDataException si no hay datos
        if (data.size() == 0) {
            System.out.println("No data has been found");
            throw new NoDataException();
        }
        // Crear y llenar un CategoryDataset
        Iterator iter = data.listIterator();
        DefaultCategoryDataset dataset = new
DefaultCategoryDataset();
        while (iter.hasNext()) {
            WebHit wh = (WebHit)iter.next();
            dataset.addValue(new
Long(wh.getHitCount()), "Reporte grafico por Hora", wh.getSection());
        }
        // Crear la tabla de objetos
        CategoryAxis categoryAxis = new CategoryAxis("");
        ValueAxis valueAxis = new NumberAxis("");
        BarRenderer renderer = new BarRenderer();
        renderer.setMaxBarWidth(1.05);
        // renderer.setItemURLGenerator(new
StandardCategoryURLGenerator("xy_chart.jsp","series","section"));
    }
}

```

```

        renderer.setToolTipGenerator(new
StandardCategoryToolTipGenerator());
        renderer.setPaint(java.awt.Color.blue);
        renderer.setBasePositiveItemLabelPosition(new
ItemLabelPosition(ItemLabelAnchor.OUTSIDE1, TextAnchor.TOP_CENTER ));
        Plot plot = new CategoryPlot(dataset,
categoryAxis, valueAxis, renderer);
        plot.setOutlinePaint(java.awt.Color.blue);
        ///JFreeChart chart = new JFreeChart("Consumo",
JFreeChart.DEFAULT_TITLE_FONT, plot, false);
        JFreeChart chart =
ChartFactory.createBarChart3D("", "Horas", "Conexiones", dataset,
PlotOrientation.VERTICAL, true, true, false);
        final CategoryPlot plot2=chart.getCategoryPlot();
        final CategoryItemRenderer
rend=plot2.getRenderer();
        final BarRenderer r=(BarRenderer)rend;
        r.setPaint(java.awt.Color.blue);
        chart.setBackgroundPaint(java.awt.Color.white);
        // Escriba la imagen gráfica para el directorio temporal
        ChartRenderingInfo info = new ChartRenderingInfo(new
StandardEntityCollection());
        filename = ServletUtilities.saveChartAsPNG(chart, 700, 300, info,
session);

        // Escriba la imagen de mapa para el PrintWriter
        ChartUtilities.writeImageMap(pw, filename, info);
        pw.flush();
    } catch (NoDataException e) {
        System.out.println(e.toString());
        filename = "public_nodata_500x300.png";
    } catch (Exception e) {
        System.out.println("Exception - " + e.toString());
        e.printStackTrace(System.out);
        filename = "public_error_500x300.png";
    }
    return filename; }

public static String generateBarChart_dias(String xml, HttpSession
session, PrintWriter pw) {
    String filename = null;
    String fecha;
    String consumo;
    int inicio,fin;
    ArrayList data = new ArrayList();
    try {
        // Retrieve list of WebHits
        //WebHitDataSet whDataSet = new WebHitDataSet();
        //ArrayList list =
whDataSet.getDataBySection(hitDate);
        System.out.println(xml.length());
        for(int h=0;h<xml.length();h++){

```

```

        if(xml.indexOf("<dispositivo>",h)>0){
            inicio=xml.indexOf("<dispositivo>",h)+13;
            // System.out.println("inicio fecha "+inicio);
            fin=xml.indexOf("</dispositivo>",inicio);
            // System.out.println("fin fecha "+fin);
            fecha=xml.substring(inicio,fin);
            System.out.println(fecha);
            inicio=fin;
            inicio=xml.indexOf("<numero>",inicio)+8;
            // System.out.println("inicio consumo "+inicio);
            fin=xml.indexOf("</numero>",inicio);
            //System.out.println("fin consumo "+fin);
            consumo=xml.substring(inicio,fin);
            h=fin;
            System.out.println(consumo);
            data.add(new
WebHit(fecha,Long.parseLong(consumo)));
        }else{
            h=xml.length()+1;
        }
        // Lanzar una costumbre NoDataException si no hay datos
        if (data.size() == 0) {
            System.out.println("No data has been found");
            throw new NoDataException();
        }
        // Crear y llenar un CategoryDataset
        Iterator iter = data.listIterator();
        DefaultCategoryDataset dataset = new
DefaultCategoryDataset();
        while (iter.hasNext()) {WebHit wh = (WebHit)iter.next();
            dataset.addValue(new
Long(wh.getHitCount()), "Dispositivos mas Usado ", wh.getSection());
        }
        // Crear la tabla de objetos
        CategoryAxis categoryAxis = new CategoryAxis("");
        ValueAxis valueAxis = new NumberAxis("");
        BarRenderer renderer = new BarRenderer();
        renderer.setMaxBarWidth(1.05);
        // renderer.setItemURLGenerator(new
StandardCategoryURLGenerator("xy_chart.jsp","series","section"));
        renderer.setToolTipGenerator(new StandardCategoryToolTipGenerator());
        renderer.setPaint(java.awt.Color.blue);
        renderer.setBasePositiveItemLabelPosition(new
ItemLabelPosition(ItemLabelAnchor.OUTSIDE1, TextAnchor.TOP_CENTER ));
        Plot plot = new CategoryPlot(dataset, categoryAxis, valueAxis,
renderer);
        plot.setOutlinePaint(java.awt.Color.blue);
        ///JFreeChart chart = new JFreeChart("Consumo",
JFreeChart.DEFAULT_TITLE_FONT, plot, false);

```

```

        JFreeChart chart =
ChartFactory.createBarChart3D("", "Horas", "Conexiones", dataset,
PlotOrientation.VERTICAL, true, true, false);
        final CategoryPlot plot2=chart.getCategoryPlot();
        final CategoryItemRenderer rend=plot2.getRenderer();
            final BarRenderer r=(BarRenderer)rend;
            r.setPaint(java.awt.Color.blue);
            chart.setBackgroundPaint(java.awt.Color.white);
        // Escriba la imagen gráfica para el directorio temporal
ChartRenderingInfo info = new ChartRenderingInfo(new
StandardEntityCollection());
filename = ServletUtilities.saveChartAsPNG(chart, 700, 300, info,
session);

        // Escriba la imagen de mapa para el PrintWriter
ChartUtilities.writeImageMap(pw, filename, info);
        pw.flush();
    } catch (NoDataException e) {
        System.out.println(e.toString());
        filename = "public_nodata_500x300.png";
    } catch (Exception e) {
        System.out.println("Exception - " + e.toString());
        e.printStackTrace(System.out);
        filename = "public_error_500x300.png";
    }
    return filename; }

public static String generatePieChart(String xml, HttpSession session,
PrintWriter pw) {
    String filename = null;
    String fecha;
    String consumo;
    int inicio,fin;
    try {
        // Recuperar la lista de WebHits
WebHitDataSet whDataSet = new WebHitDataSet();
ArrayList list= new ArrayList();
        for(int h=0;h<xml.length();h++){
            if(xml.indexOf("<dispositivo>",h)>0){
                inicio=xml.indexOf("<dispositivo>",h)+13;
                // System.out.println("inicio fecha "+inicio);
                fin=xml.indexOf("</dispositivo>",inicio);
                // System.out.println("fin fecha "+fin);
                fecha=xml.substring(inicio,fin);
                System.out.println(fecha);
                inicio=fin;
                inicio=xml.indexOf("<numero>",inicio)+8;
                // System.out.println("inicio consumo "+inicio);
                fin=xml.indexOf("</numero>",inicio);
                //System.out.println("fin consumo "+fin);
                consumo=xml.substring(inicio,fin);
                h=fin;
            }
        }
    }
}

```

```

        System.out.println(consumo);
        list.add(new
WebHit(fecha,Long.parseLong(consumo)));
    }else{
        h=xml.length()+1;    }
    // Lanzar una costumbre NoDataException si no hay datos
    if (list.size() == 0) {
        System.out.println("No data has been found");
        throw new NoDataException(); }
    // Crear y llenar un PieDataSet
    DefaultPieDataset data = new DefaultPieDataset();
    Iterator iter = list.listIterator();
    while (iter.hasNext()) {
        WebHit wh = (WebHit)iter.next();
data.setValue(wh.getSection(), new Long(wh.getHitCount()));}
    // Crear la tabla de objetos
    PiePlot plot = new PiePlot(data);
    plot.setInsets(new Insets(0, 5, 5, 5));
//plot.setURLGenerator(new
StandardPieURLGenerator("xy_chart.jsp","section"));
    plot.setToolTipGenerator(new
StandardPieItemLabelGenerator());
        JFreeChart chart = new JFreeChart("",
JFreeChart.DEFAULT_TITLE_FONT, plot, true);
        chart.setBackgroundPaint(java.awt.Color.white);
    // Escriba la imagen gráfica para el directorio temporal
    ChartRenderingInfo info = new ChartRenderingInfo(new
StandardEntityCollection());
    filename = ServletUtilities.saveChartAsPNG(chart, 500, 300, info,
session);
    // Escriba la imagen de mapa para el PrintWriter
    ChartUtilities.writeImageMap(pw, filename, info);
        pw.flush();
    } catch (NoDataException e) {
        System.out.println(e.toString());
        filename = "public_nodata_500x300.png";
    } catch (Exception e) {
        System.out.println("Exception - " + e.toString());
        e.printStackTrace(System.out);
        filename = "public_error_500x300.png";}
    return filename; }

public static String generate3DPieChart(String xml, HttpSession
session, PrintWriter pw) {
    String filename = null;
    String fecha;
    String consumo;
    int inicio,fin;
    try {

```

```

        // Recuperar la lista de WebHits
WebHitDataSet whDataSet = new WebHitDataSet();
        ArrayList list= new ArrayList();
        for(int h=0;h<xml.length();h++){
            if(xml.indexOf("<dispositivo>",h)>0){
                inicio=xml.indexOf("<dispositivo>",h)+13;
                // System.out.println("inicio fecha "+inicio);
                fin=xml.indexOf("</dispositivo>",inicio);
                // System.out.println("fin fecha "+fin);
                fecha=xml.substring(inicio,fin);
                System.out.println(fecha);
                inicio=fin;
                inicio=xml.indexOf("<numero>",inicio)+8;
                // System.out.println("inicio consumo "+inicio);
                fin=xml.indexOf("</numero>",inicio);
                //System.out.println("fin consumo "+fin);
                consumo=xml.substring(inicio,fin);
                h=fin;
                System.out.println(consumo);
                list.add(new
WebHit(fecha,Long.parseLong(consumo)));
            }else{
                h=xml.length()+1;    }    }
        // Lanzar una costumbre NoDataException si no hay datos
        if (list.size() == 0) {
            System.out.println("No data has been found");
            throw new NoDataException();}
        // Crear y llenar un PieDataSet
DefaultPieDataset data = new DefaultPieDataset();
        Iterator iter = list.listIterator();
        while (iter.hasNext()) {
            WebHit wh = (WebHit)iter.next();
data.setValue(wh.getSection(), new Long(wh.getHitCount()));}
        final JFreeChart chart = ChartFactory.createPieChart3D("",
data, true,true,false);
            chart.setBackgroundPaint(Color.white);
            final PiePlot3D plot = (PiePlot3D) chart.getPlot();
plot.setLabelFont(new Font("Times New Roman", Font.PLAIN, 12));
            plot.setStartAngle(270);
            plot.setDirection(Rotation.ANTICLOCKWISE);
            plot.setForegroundAlpha(0.60f);
            plot.setInteriorGap(0.33);
        //Color[] colors = { Color.yellow,Color.blue, Color.orange};
        // PieRenderer renderer = new PieRenderer(colors);
        // renderer.setColor(plot, data);
        // final CategoryItemRenderer renderer = new
CustomRenderer( new Paint[] {Color.red, Color.blue, Color.green,
Color.yellow, Color.orange, Color.cyan,Color.magenta, Color.blue});
        // Create the chart object

```

```

        /*      PiePlot plot = new PiePlot(data);
        plot.setInsets(new Insets(0, 5, 5, 5));
        //plot.setURLGenerator(new
StandardPieURLGenerator("xy_chart.jsp","section"));
        plot.setToolTipGenerator(new
StandardPieItemLabelGenerator());
        //JFreeChart chart = new JFreeChart("",
JFreeChart.DEFAULT_TITLE_FONT, plot, true);
        chart.setBackgroundPaint(java.awt.Color.white);*/
        //  Escriba la imagen gráfica para el directorio temporal
        ChartRenderingInfo info = new ChartRenderingInfo(new
StandardEntityCollection());
filename = ServletUtilities.saveChartAsPNG(chart, 500, 300, info,
session);

        //  Escriba la imagen de mapa para el PrintWriter
ChartUtilities.writeImageMap(pw, filename, info);
        pw.flush();
    } catch (NoDataException e) {
        System.out.println(e.toString());
        filename = "public_nodata_500x300.png";
    } catch (Exception e) {
        System.out.println("Exception - " + e.toString());
        e.printStackTrace(System.out);
        filename = "public_error_500x300.png";
    }
    return filename; }

public static String generateXYChart(String xml, HttpSession session,
PrintWriter pw) {
    String filename = null;
    String fecha;
    String consumo;
    int inicio,fin;
    try {
        //  Recuperar la lista de WebHits
SimpleDateFormat sdf1 = new SimpleDateFormat("yyyy-MM-
dd",Locale.UK);

        ArrayList list= new ArrayList();
        for(int h=0;h<xml.length();h++){
            if(xml.indexOf("<dia>",h)>0){
                inicio=xml.indexOf("<dia>",h)+5;
                // System.out.println("inicio fecha "+inicio);
                fin=xml.indexOf("</dia>",inicio);
                // System.out.println("fin fecha "+fin);
                fecha=xml.substring(inicio,fin);
                System.out.println(fecha);
                inicio=fin;
                inicio=xml.indexOf("<conexion>",inicio)+10;
                // System.out.println("inicio consumo "+inicio);
                fin=xml.indexOf("</conexion>",inicio);
                //System.out.println("fin consumo "+fin);
            }
        }
    }
}

```

```

        consumo=xml.substring(inicio,fin);
        h=fin;
        System.out.println(consumo);
list.add(new WebHit(sdf1.parse(fecha),Long.parseLong(consumo)));
    }else{ h=xml.length()+1;    }    }
// Lanzar una costumbre NoDataException si no hay datos
if (list.size() == 0) {
    System.out.println("No data has been found");
    throw new NoDataException(); }
// Crear y llenar una colección XYSeries
XYSeries dataSet = new XYSeries("Hits");
Iterator iter = list.listIterator();
while (iter.hasNext()) {
    WebHit wh = (WebHit)iter.next();
    dataSet.add(wh.getHitDate().getTime(),wh.getHitCount());}
XYSeriesCollection xyDataset = new XYSeriesCollection(dataSet);
// Crear descripción y URL generadores
SimpleDateFormat sdf = new SimpleDateFormat("dd-MMM-yyyy", Locale.UK);
StandardXYToolTipGenerator ttg = new
StandardXYToolTipGenerator(
    StandardXYToolTipGenerator.DEFAULT_TOOL_TIP_FORMAT,
    sdf, NumberFormat.getInstance());
TimeSeriesURLGenerator urlg = new TimeSeriesURLGenerator(
    sdf, "pie_chart.jsp", "series", "hitDate");
// Crear la tabla de objetos
ValueAxis timeAxis = new DateAxis("");
NumberAxis valueAxis = new NumberAxis("");
valueAxis.setAutoRangeIncludesZero(false); // override default
StandardXYItemRenderer renderer = new StandardXYItemRenderer(
StandardXYItemRenderer.LINES + StandardXYItemRenderer.SHAPES,
    ttg, urlg);
renderer.setShapesFilled(true);
XYPlot plot = new XYPlot(xyDataset, timeAxis, valueAxis, renderer);
JFreeChart chart = new JFreeChart("", JFreeChart.DEFAULT_TITLE_FONT,
plot, false);
chart.setBackgroundPaint(java.awt.Color.white);
// Escriba la imagen gráfica para el directorio temporal
ChartRenderingInfo info = new ChartRenderingInfo(new
StandardEntityCollection());
filename = ServletUtilities.saveChartAsPNG(chart, 500, 300, info,
session);

// Escriba la imagen de mapa para el PrintWriter
ChartUtilities.writeImageMap(pw, filename, info);
pw.flush();
} catch (NoDataException e) {
    System.out.println(e.toString());
    filename = "public_nodata_500x300.png";
} catch (Exception e) {
    System.out.println("Exception - " + e.toString());
}

```



```

        e.printStackTrace(System.out);
        filename = "public_error_500x300.png";
    }return filename; }
public static String generateXYAreaChart(HttpSession session,
    PrintWriter pw) {
    String filename = null;
    try {
        // Recuperar la lista de WebHits para cada sección y
        llenar un TableXYDataset
        WebHitDataSet whDataSet = new WebHitDataSet();
        ArrayList sections = whDataSet.getSections();
        Iterator sectionIter = sections.iterator();
        DefaultTableXYDataset dataset = new
        DefaultTableXYDataset();
        while (sectionIter.hasNext()) {
            String section = (String)sectionIter.next();
            ArrayList list = whDataSet.getDataByHitDate(section);
            XYSeries dataSeries = new XYSeries(section, true, false);
            Iterator webHitIter = list.iterator();
            while (webHitIter.hasNext()) {
                WebHit webHit = (WebHit)webHitIter.next();
                dataSeries.add(webHit.getHitDate().getTime(), webHit.getHitCount()); }
            dataset.addSeries(dataSeries);
        }
        // Lanzar una costumbre NoDataException si no hay datos
        if (dataset.getItemCount() == 0) {
            System.out.println("No data has been found");
            throw new NoDataException();
        }
        // Crear descripción y URL generadores
        SimpleDateFormat sdf = new SimpleDateFormat("dd-MMM-yyyy", Locale.UK);
        StandardXYToolTipGenerator ttg = new StandardXYToolTipGenerator(
            StandardXYToolTipGenerator.DEFAULT_TOOL_TIP_FORMAT,
            sdf, NumberFormat.getInstance());
        TimeSeriesURLGenerator urlg = new TimeSeriesURLGenerator(
            sdf, "bar_chart.jsp", "series", "hitDate");
        // Crear el eje X
        DateAxis xAxis = new DateAxis(null);
        xAxis.setLowerMargin(0.0);
        xAxis.setUpperMargin(0.0);
        // Create the X-Axis
        NumberAxis yAxis = new NumberAxis(null);
        yAxis.setAutoRangeIncludesZero(true);
        // Crear el render
        StackedXYAreaRenderer renderer =
        new StackedXYAreaRenderer(XYAreaRenderer.AREA_AND_SHAPES, ttg, urlg);
        renderer.setSeriesPaint(0, new Color(255, 255, 180));
        renderer.setSeriesPaint(1, new Color(206, 230, 255));
        renderer.setSeriesPaint(2, new Color(255, 230, 230));
        renderer.setSeriesPaint(3, new Color(206, 255, 206));
        renderer.setShapePaint(Color.gray);
    }
}

```

```

        renderer.setShapeStroke(new BasicStroke(0.5f));
        renderer.setShape(new Ellipse2D.Double(-3, -3, 6, 6));
        renderer.setOutline(true);
        // Crear la parcela
        XYPlot plot = new XYPlot(dataset, xAxis, yAxis, renderer);
        plot.setForegroundAlpha(0.65f);
        // Volver a eje por lo que la gama de auto-conocimiento
que los datos se apila
        yAxis.configure();
        // Crear la tabla
        JFreeChart chart = new JFreeChart(null,
JFreeChart.DEFAULT_TITLE_FONT, plot, true);
        chart.setBackgroundPaint(java.awt.Color.white);
        // Escriba la imagen gráfica para el directorio temporal
        ChartRenderingInfo info = new ChartRenderingInfo(new
StandardEntityCollection());
        filename = ServletUtilities.saveChartAsPNG(chart, 500, 500, info,
session);

        // Escriba la imagen de mapa para el PrintWriter
        ChartUtilities.writeImageMap(pw, filename, info);
        pw.flush();

    } catch (NoDataException e) {
        System.out.println(e.toString());
        filename = "public_nodata_500x300.png";
    } catch (Exception e) {
        System.out.println("Exception - " + e.toString());
        e.printStackTrace(System.out);
        filename = "public_error_500x300.png";
    } return filename;
}

public static void main(java.lang.String[] args) {
    try {
        SimpleDateFormat sdf = new SimpleDateFormat("dd-MMM-yyyy", Locale.UK);
        PrintWriter pw = new PrintWriter(System.out);
        String filename = WebHitChart.generateXYChart("<?xml
version='1.0' encoding='ISO-8859-1'?><dia>2008-07-
19</dia><conexion>0</conexion><dia>2008-07-
21</dia><conexion>7</conexion><dia>2008-07-
24</dia><conexion>25</conexion><dia>2008-07-
28</dia><conexion>36</conexion>", null, pw);
        // String filename =
WebHitChart.generatePieChart(sdf.parse("01-Aug-2002"), null, pw);
        // String filename =
WebHitChart.generateXYChart("service", null, pw);
        System.out.println("filename - " + filename);
    } catch (Exception e) {
        System.out.println("Exception - " + e.toString());
    }
}

```

```

        e.printStackTrace();}
        return;}
    public static class PieRenderer    {
        private Color[] color;
        public PieRenderer(Color[] color) {
            this.color = color;        }
        public void setColor(PiePlot plot, DefaultPieDataset dataset) {
            java.util.List <Comparable> keys = dataset.getKeys();
            int aInt;
            for (int i = 0; i < keys.size(); i++) {
                aInt = i % this.color.length;
                plot.setSectionPaint(this.color[aInt]);    }    } }}

```

5.6 CLASE ORG.JSON

Esta clase permite trabajar en conjunto para realizar la interacción del cliente hacia un servidor por medio de un objeto ajax.

5.6.1 CLASE “JSONArray.java” ARREGLO

```

package org.json;
import java.text.ParseException;
import java.util.ArrayList;
import java.util.Collection;
import java.util.NoSuchElementException;
/**
 * Un JSONArray es una secuencia ordenada de valores. Su forma externa
 * es una cadena
 *   * Rodeada por corchetes cuadrados con comas entre los valores. El
 *     interior de forma
 *   * Es un objeto que tengan get () y elegir () los métodos para acceder
 *     a los valores por
 *   * Índice, y poner () métodos para añadir o sustituir los valores. Los
 *     valores pueden ser
 *   * Cualquiera de estos tipos: Boolean, JSONArray, JSONObject, Número,
 *     de cuerda, o la
 *   * JSONObject.NULL objeto.
 *   * <p>
 *   * El constructor puede convertir una cadena JSON forma externa en un
 *   * Java forma interna objeto. El método toString () crea una externa
 *   * Forma de cadena.
 *   * <p>
 *   * Un get () devuelve un método de valor si se puede encontrar uno, y
 *     lanza una excepción
 *   * Si uno no se puede encontrar. Una opción () método devuelve un
 *     valor por defecto en lugar de

```

```

* Lanzar una excepción, por lo que es útil para la obtención de
  valores opcionales.
* <p>
*El genérico get () y elegir () métodos de retorno de un objeto que se
  puede emitir o
* Tipo de consulta. Hay también escribió get () y elegir () métodos
  que hacer escribir
* Tipo de control y coercion para usted.
* <p>
* Los textos producidos por el toString () son métodos muy estrictos.
* Los constructores son más clementes en los textos van a aceptar.
* <ul>
* Un extra <li> coma puede aparecer justo antes del cierre de
  paréntesis. </ Li>
* Cadenas <li> puede ser citado con comillas simples. </ Li>
* Cadenas <li> no necesitan ser citado a todos si no contienen líder
  O los espacios, y si no contienen cualquiera de estos caracteres:
* () [] / \:, </ Li>
* Números <li> pueden tener de 0 - (octal) o 0x-(hex) prefijo. </ Li>
* </ul>
* <p>
* Public Domain 2002 JSON.org
* @author JSON.org
* @version 0.1
*/
public class JSONArray {
    /**
     * El getArrayList donde el JSONArray las propiedades se mantienen.
     */
    private ArrayList myArrayList;
    /**
     * Construir un vacío JSONArray.
     */
    public JSONArray() {
        myArrayList = new ArrayList();
    }
    /**
     * Construir un JSONArray de un JSNTokener.
     * @ ParseException A excepción JSONArray debe comenzar con '['
     * @ ParseException previsto una excepción ',' o ']'
     * @ Param x A JSNTokener
     */
    public JSONArray(JSNTokener x) throws ParseException {
        this();
        if (x.nextClean() != '[') {
            throw x.syntaxError("A JSONArray must start with '['");
        }
        if (x.nextClean() == ']') {
            return;
        }
    }
}

```

```

        x.back();    while (true) {
            myArrayList.add(x.nextValue());
            switch (x.nextClean()) {
                case ',':
                    if (x.nextClean() == ']') {
                        return;    }
                    x.back();
                    break;
                case ']':
                    return;
                default:
                    throw x.syntaxError("Expected a ',' or ']'");    }
            }    }
/**
 * Construir un JSONArray de una cadena de origen.
 * @ Excepción ParseException La cadena debe ajustarse a la
 * sintaxis de JSON.
 * @ Param cadena Una cadena que comienza con '[' y termina con
 * ']'.
 */

public JSONArray(String string) throws ParseException {
    this(new JSNTokener(string));    }
/**
 * Construir un JSONArray de una colección.
 * @ Param colección Colección.
 */
public JSONArray(Collection collection) {
    myArrayList = new ArrayList(collection);    }
/**
 * Obtener el valor de objeto asociado a un índice.
 * @ Excepción NoSuchElementException
 * @ Param índice subíndice
 * El índice debe estar entre 0 y la longitud () - 1.
 * @ Return Un objeto de valor.
 */
public Object get(int index) throws NoSuchElementException {
    Object o = opt(index);
    if (o == null) {
        throw new NoSuchElementException("JSONArray[" + index +
            "] not found.");    }
    return o;    }
/**
 * Obtener el ArrayList que es la celebración de los elementos de la
 * JSONArray.
 * @ Return El ArrayList.
 */
ArrayList getArrayList() {
    return myArrayList;    }

```

```

/**
 * Obtener el valor booleano asociado a un índice.
 * La cadena de valores "verdadero" y "falso" se convierten a
 * boolean.
 *
 * @ NoSuchElementException excepción si el índice no se encuentra
 * @ Excepción ClassCastException
 * @ Param índice subíndice
 * @ Return La verdad.
 */
public boolean getBoolean(int index)
    throws ClassCastException, NoSuchElementException {
    Object o = get(index);
    if (o == Boolean.FALSE || o.equals("false")) {
        return false;
    } else if (o == Boolean.TRUE || o.equals("true")) {
        return true;
    }
    throw new ClassCastException("JSONArray[" + index +
        "] not a Boolean."); }

/**
 * Obtener el doble valor asociado a un índice.
 *
 * @ NoSuchElementException excepción si la clave no se encuentra
 * @ Excepción NumberFormatException
 * Si el valor no se pueden convertir a un número.
 *
 * @ Param índice subíndice
 * @ Return El valor.
 */
public double getDouble(int index)
    throws NoSuchElementException, NumberFormatException {
    Object o = get(index);
    if (o instanceof Number) {
        return ((Number) o).doubleValue();
    }
    if (o instanceof String) {
        return new Double((String)o).doubleValue();
    }
    throw new NumberFormatException("JSONObject[" +
        index + "] is not a number."); }

/**
Int * Obtener el valor asociado a un índice.
 *
 * @ NoSuchElementException excepción si la clave no se encuentra
 * @ Excepción NumberFormatException
 * Si el valor no se pueden convertir a un número.
 *
 * @ Param índice subíndice
 * @ Return El valor.
 */
public int getInt(int index)
    throws NoSuchElementException, NumberFormatException {
    Object o = get(index);

```

```

        if (o instanceof Number) {
            return ((Number)o).intValue();    }
        return (int)getDouble(index);    }
    /**
    * Obtener el JSONArray asociados con un índice.
    * @ NoSuchElementException excepción si el índice no se encuentra
    * o si la
    * No es un valor JSONArray
    * @ Param índice subíndice
    * @ Return Un JSONArray valor.
    */
    public JSONArray getJSONArray(int index) throws
        NoSuchElementException {
        Object o = get(index);
        if (o instanceof JSONArray) {
            return (JSONArray)o;    }
        throw new NoSuchElementException("JSONArray[" + index +
            "] is not a JSONArray.");    }
    /**
    * Obtener el JSONArray asociados con un índice.
    * @ NoSuchElementException excepción si el índice no se encuentra
    * o si la
    * No es un valor JSONArray
    * @ Param índice subíndice
    * @ Return Un JSONArray valor.
    */
    public JSONObject getJSONObject(int index) throws
        NoSuchElementException {
        Object o = get(index);
        if (o instanceof JSONObject) {
            return (JSONObject)o;    }
        throw new NoSuchElementException("JSONArray[" + index +
            "] is not a JSONObject.");    }
    /**
    * Obtener la cadena asociada a un índice.
    * @ Excepción NoSuchElementException
    * @ Param índice subíndice
    * @ Return un valor de cadena.
    */
    public String getString(int index) throws NoSuchElementException {
        return get(index).toString();
    }
    /**
    * Determinar si el valor es nulo.
    * @ Param índice subíndice
    * @ Return true si el valor en el índice es nulo, o si no hay
    ningún valor.
    */
    public boolean isNull(int index) {

```

```

        Object o = opt(index);
        return o == null || o.equals(null);    }
    /**
    * Hacer una cadena de los contenidos de esta JSONArray. El separador de
    cadena
    * Se inserta entre cada elemento.
    * Advertencia: Este método supone que la estructura de datos es
    acyclical.
    * @ Param separador de una cadena que se insertará entre los
    elementos.
    * @ Return una cadena.
    */
    public String join(String separator) {
        int i;
        Object o;
        StringBuffer sb = new StringBuffer();
        for (i = 0; i < myArrayList.size(); i += 1) {
            if (i > 0) {
                sb.append(separator);
            }
            o = myArrayList.get(i);
            if (o == null) {
                sb.append("");
            } else if (o instanceof String) {
                sb.append(JSONObject.quote((String)o));
            } else if (o instanceof Number) {
                sb.append(JSONObject.numberToString((Number)o));
            } else {
                sb.append(o.toString());
            }
        }
        return sb.toString();
    }
    /**
    * Obtener la longitud de la JSONArray.
    *
    * @ Return La duración (o tamaño).
    */
    public int length() {
        return myArrayList.size();
    }
    /**
    * Obtener el facultativo objeto valor asociado a un índice.
    * @ Param índice subíndice
    * @ Return Un objeto de valor, o null si no hay
    * Objeto en ese índice.
    */
    public Object opt(int index) {
        if (index < 0 || index >= length()) {
            return null;
        } else {
            return myArrayList.get(index);
        }
    }

```



```

/**
 * Obtener el valor booleano opcional asociada a un índice.
 * Devuelve falso si no hay ningún valor en ese índice,
 * O, si el valor no es Boolean.TRUE o la cadena "verdadero".
 *
 * @ Param índice subíndice
 * @ Return La verdad.
 */
public boolean optBoolean(int index) {
    return optBoolean(index, false);
}

/**
 * Obtener el valor booleano opcional asociada a un índice.
 * Devuelve la defaultValue si no hay ningún valor en ese índice o
 * si no se
 * Un booleano o la cadena "verdadero" o "falso".
 *
 * @ Param índice subíndice
 * @ Param boolean defaultValue un defecto.
 * @ Return La verdad.
 */
public boolean optBoolean(int index, boolean defaultValue) {
    Object o = opt(index);
    if (o != null) {
        if (o == Boolean.FALSE || o.equals("false")) {
            return false;
        } else if (o == Boolean.TRUE || o.equals("true")) {
            return true;
        }
    }
    return defaultValue;
}

/**
 * Obtener el facultativo doble valor asociado a un índice.
 * NaN es devuelto si el índice no se encuentra,
 * O, si el valor no es un número y no se pueden convertir a un
 * número.
 *
 * @ Param índice subíndice
 * @ Return El valor.
 */
public double optDouble(int index) {
    return optDouble(index, Double.NaN);
}

/**
 * Obtener el facultativo doble valor asociado a un índice.
 * El defaultValue se devuelve si el índice no se encuentra,
 * O, si el valor no es un número y no se pueden convertir a un
 * número.
 *
 * @ Param índice subíndice
 * @ Param defaultValue El valor por defecto.
 * @ Return El valor.

```

```

    */
    public double optDouble(int index, double defaultValue) {
        Object o = opt(index);
        if (o != null) {
            if (o instanceof Number) {
                return ((Number) o).doubleValue();
            } try {
                return new Double((String)o).doubleValue();
            } catch (Exception e) { }
        } return defaultValue; }
    /**
    * Obtener el facultativo int valor asociado a un índice.
    * Cero es devuelto si el índice no se encuentra,
    * O, si el valor no es un número y no se pueden convertir a un
    * número.
    *
    * @ Param índice subíndice
    * @ Return El valor.
    */

    public int optInt(int index) { return optInt(index, 0); }
    /**
    * Obtener el facultativo int valor asociado a un índice.
    * El defaultValue se devuelve si el índice no se encuentra,
    * O, si el valor no es un número y no se pueden convertir a un
    * número.
    * @ Param índice subíndice
    * @ Param defaultValue El valor por defecto.
    * @ Return El valor.
    */
    public int optInt(int index, int defaultValue) {
        Object o = opt(index);
        if (o != null) {
            if (o instanceof Number) {
                return ((Number)o).intValue(); }
            try { return Integer.parseInt((String)o); }
            catch (Exception e) { }
        } return defaultValue; }
    /**
    * Obtener el facultativo JSONArray asocia con un índice.
    * @ Param índice subíndice
    * @ Return un valor JSONArray, o nulo, si el índice no tiene
    * ningún valor,
    * O, si el valor no es un JSONArray.
    */
    public JSONArray optJSONArray(int index) {
        Object o = opt(index);
        if (o instanceof JSONArray) {
            return (JSONArray)o;

```

```

        } return null; }
    /**
    * Obtener el facultativo JSONObject asocia con un índice.
    * Nulo si se devuelve la clave no se encuentra, o null si el
    * índice ha
    * No tiene valor, o si el valor no es un JSONObject.
    *
    * @ Param índice subíndice
    * @ Return Un JSONObject valor.
    */
    public JSONObject optJSONObject(int index) {
        Object o = opt(index);
        if (o instanceof JSONObject) {
            return (JSONObject)o;
        } return null; }
    /**
    * Obtener el valor de cadena opcional asociada con un índice. Devuelve
    * una
    * Cadena vacía si no hay ningún valor en ese índice. Si el valor
    * No es una cadena y no es nulo, entonces se convierte en una
    * cadena.
    *
    * @ Param índice subíndice
    * @ Return un valor de cadena.
    */
    public String optString(int index){
        return optString(index, ""); }
    /**
    * Obtener el facultativo cadena asociada con un índice.
    * El defaultValue se devuelve si la clave no se encuentra.
    *
    * @ Param índice subíndice
    * @ Param defaultValue El valor por defecto.
    * @ Return un valor de cadena.
    */
    public String optString(int index, String defaultValue){
        Object o = opt(index);
        if (o != null) {
            return o.toString(); }
        return defaultValue; }
    /**
    * Agrega un valor booleano.
    *
    * @ Param valor Un valor booleano.
    * @ Return presente.
    */
    public JSONArray put(boolean value) {
        put(new Boolean(value));
        return this; }

```

```

    /**
    * Anexar un doble valor.
    *
    * @ Param valor de un doble valor.
    * @ Return presente.
    */
    public JSONArray put(double value) {
        put(new Double(value));
        return this; }
    /**
    * Agrega un valor int.
    *
    * @ Param valor un valor int.
    * @ Return presente.
    */
    public JSONArray put(int value) {
        put(new Integer(value));
        return this; }

    /**
    * Anexar un objeto de valor.
    * @ Param valor Un objeto de valor. El valor debe ser un
    * Boolean, Doble, Entero, JSONArray, JSONObject, o de cuerda, o la
    * JSONObject.NULL objeto.
    * @ Return presente.
    */
    public JSONArray put(Object value) {
        myArrayList.add(value);
        return this; }
    /**
    * Ponga o sustituir un valor booleano en la JSONArray.
    * @ Excepción NoSuchElementException El índice no debe ser
    negativo.
    * @ Param El índice subíndice subíndice. Si el índice es mayor
    que la longitud de
    * El JSONArray, entonces nula elementos según sea necesario, se
    añadirá a la almohadilla
    * A cabo.
    * @ Param valor Un valor booleano.
    * @ Return presente.
    */
    public JSONArray put(int index, boolean value) {
        put(index, new Boolean(value));
        return this; }
    /**
    * Ponga o sustituir un doble valor.
    * @ Excepción NoSuchElementException El índice no debe ser
    negativo.

```

```

    * @ Param El índice subíndice subíndice. Si el índice es mayor
    que la longitud de
    * El JSONArray, entonces nula elementos según sea necesario, se
    añadirá a la almohadilla
    * A cabo.
    * @ Param valor de un doble valor.
    * Devolver este.
    */
    public JSONArray put(int index, double value) {
        put(index, new Double(value));
        return this; }
    /**
    * Ponga o sustituir un valor int.
    * @ Excepción NoSuchElementException El índice no debe ser
    negativo.
    * @ Param El índice subíndice subíndice. Si el índice es mayor
    que la longitud de
    * El JSONArray, entonces nula elementos según sea necesario, se
    añadirá a la almohadilla
    * A cabo.
    * @ Param valor un valor int.
    * @ Return presente.
    */
    public JSONArray put(int index, int value) {
        put(index, new Integer(value));
        return this; }
    /**
    * Ponga o sustituir un objeto de valor en el JSONArray.
    * @ Excepción NoSuchElementException El índice no debe ser
    negativo.
    * @ Param El índice subíndice. Si el índice es mayor que la
    longitud de
    * El JSONArray, entonces nula elementos según sea necesario, se
    añadirá a la almohadilla
    * A cabo.
    * @ Param valor Un objeto de valor.
    * Devolver este.
    */
    public JSONArray put(int index, Object value)
        throws NoSuchElementException, NullPointerException {
        if (index < 0) {
            throw new NoSuchElementException("JSONArray[" + index +
                "] not found.");
        } else if (value == null) {
            throw new NullPointerException();
        } else if (index < length()) {
            myArrayList.set(index, value);
        } else { while (index != length()) {
            put(null); }
        }
    }

```

```

        put(value);    }
        return this;   }
    /**
    * Producir un JSONObject por la combinación de un JSONArray de los
    nombres con los valores
    * De este JSONArray.
    * @ Param A JSONArray nombres que contiene una lista de las
    principales cadenas. Estos serán
    * Junto con los valores.
    * @ Return Un JSONObject, o null si no hay nombres, o si esta
    JSONArray
    * No tiene valores.
    */
    public JSONObject toJSONObject(JSONArray names) {
        if (names == null || names.length() == 0 || length() == 0) {
            return null;
        }
        JSONObject jo = new JSONObject();
        for (int i = 0; i < names.length(); i += 1) {
            jo.put(names.getString(i), this.opt(i));
        }
        return jo;
    }

    /**
    * Hacer una cadena JSON forma externa de esta JSONArray. Por
    compacidad, no
    * Se añade en blanco innecesarios.
    * Advertencia: Este método supone que la estructura de datos es
    acyclical.
    *
    * @ Return una impresora, mostrar, transmisibles
    * Representación de la matriz.
    */
    public String toString() {
        return '[' + join(",") + ']';
    }

    /**
    * Hacer un prettyprinted JSON JSONArray de esta cadena.
    * Advertencia: Este método supone que la estructura de datos no
    es cíclico.
    * @ Param indentFactor El número de espacios para añadir a cada
    nivel de
    * Indentación.
    * @ Return una impresora, mostrar, transmisibles
    * La representación del objeto, que comienza con '[' y acabando
    con ']'.
    */
    public String toString(int indentFactor) {
        return toString(indentFactor, 0);
    }
}
/**

```

```

* Hacer un prettyprinted cadena de este JSONArray.
  * Advertencia: Este método supone que la estructura de datos no
    es cíclico.
  * @ Param indentFactor El número de espacios para añadir a cada
    nivel de
  * Indentación.
  * @ Param indentación El guión de la de nivel superior.
  * @ Return una impresora, mostrar, transmisibles
  * Representación de la matriz.
*/
String toString(int indentFactor, int indent) {
    int i;
    Object o;
    String pad = "";
    StringBuffer sb = new StringBuffer();
    indent += indentFactor;
    for (i = 0; i < indent; i += 1) {
        pad += ' ';
    }
    sb.append("[\n");
    for (i = 0; i < myArrayList.size(); i += 1) {
        if (i > 0) {
            sb.append(",\n");
        }
        sb.append(pad);
        o = myArrayList.get(i);
        if (o == null) {
            sb.append("null");
        } else if (o instanceof String) {
            sb.append(JSONObject.quote((String) o));
        } else if (o instanceof Number) {
            sb.append(JSONObject.numberToString((Number) o));
        } else if (o instanceof JSONObject) {
            sb.append(((JSONObject)o).toString(indentFactor, indent));
        } else if (o instanceof JSONArray) {
            sb.append(((JSONArray)o).toString(indentFactor, indent));
        } else {
            sb.append(o.toString());
        }
        sb.append(' ');
    }
    return sb.toString();
}

```

5.7 CLASE “JSONObject.java” STRING O UN OBJETO

```

package org.json;
import java.util.HashMap;
import java.util.Iterator;
import java.util.Map;
import java.util.NoSuchElementException;
import java.text.ParseException;
/**

```

```

* Un JSONObject es una desordenada colección de pares nombre / valor.
  Su
* Forma externa está envuelto en una cadena con llaves de colones
  entre el
* Los nombres y valores, y comas entre los valores y los nombres. El
  interior de forma
* Es un objeto que tengan get () y elegir () los métodos para acceder
  a los valores por su nombre,
Y poner * () métodos para añadir o sustituir los valores por su
  nombre. Los valores pueden ser
* Cualquiera de estos tipos: Boolean, JSONArray, JSONObject, Número,
  de cuerda, o la
* JSONObject.NULL objeto.
* <p>
* El constructor puede convertir una cadena de forma externa en una
  forma interna
* Java objeto. El método toString () crea una cadena de forma
  externa.
* <p>
* Un get () devuelve un método de valor si se puede encontrar uno, y
  lanza una excepción
* Si uno no se puede encontrar. Una opción () método devuelve un
  valor por defecto en lugar de
* Lanzar una excepción, por lo que es útil para la obtención de
  valores opcionales.
* <p>
* El genérico get () y elegir () métodos de retorno de un objeto, que
  puede emitir o
* Tipo de consulta. Hay también escribió get () y elegir () métodos
  que hacer escribir
* Tipo de control y coercion para usted.
* <p>
* Los textos producidos por el toString () son métodos muy estrictos.
* Los constructores son más clementes en los textos van a aceptar.
* <ul>
* Un extra <li> coma puede aparecer justo antes de la llave de
  cierre. </ Li>
* Cadenas <li> puede ser citado con comillas simples. </ Li>
* Cadenas <li> no necesitan ser citado a todos si no contienen líder
  O los espacios, y si no contienen cualquiera de estos caracteres:
* () [] / \:, </ Li>
* Números <li> pueden tener de 0 - (octal) o 0x-(hex) prefijo. </ Li>
* </ Ul>
* <p>
* Dominio Público JSON.org 2002
* @ Autor JSON.org
* @ Versión 0.1
*/
public class JSONObject {

```



```

/**
 * JSONObject.NULL es equivalente al valor que pide tener Javascript
 * nulo,
 *   * Java es nula, mientras que es equivalente al valor que pide
 *   JavaScript
 *   * Indefinido.
 */
private static final class Null {
    /**
     * Make a Null object.
     */
    private Null() {
    }
    /**
     * No está destinada únicamente a ser una única instancia de
     la NULL objeto,
     * Por lo que el método devuelve propio clon.
     * @ Return NULL.
     * /
     */
    protected final Object clone() {
        return this;
    }
    /**
     * Un objeto nulo es igual al valor nulo y para sí mismo.
     * @ Param objeto Un objeto de prueba para nullness.
     * @ Return true si el objeto parámetro es el objeto
     JSONObject.NULL
     * O nula.
     */
    public boolean equals(Object object) {
        return object == null || object == this;
    }
    /**
     * Obtener el "nulo" valor de cadena.
     * @ Return La cadena "null".
     */
    public String toString() {
        return "null";    }    }
/**
 * El hash JSONObject el mapa donde las propiedades se mantienen.
 */
private HashMap myHashMap;
/**
 * A veces es más conveniente y menos ambiguo para tener un NULL
 *   * Objeto de usar Java de valor nulo.
 *   * JSONObject.NULL.equals (null) devuelve true.
 *   * JSONObject.NULL.toString () devuelve "null".
 */

```

```

    public static final Object NULL = new Null();
    /**
 * Construir un vacío JSONObject.
 */
    public JSONObject() {
        myHashMap = new HashMap();
    }
    /**
 * Construir un JSONObject de un JSONTokener.
 * @ Lanza ParseException si hay un error de sintaxis en la cadena
    origen.
 * @ Param x JSONTokener un objeto que contenga la cadena de
    origen.
 */
    public JSONObject(JSONTokener x) throws ParseException {
        this();
        char c;
        String key;
        if (x.next() == '%') {
            x.unescape();
        }
        x.back();
        if (x.nextClean() != '{') {
            throw x.syntaxError("A JSONObject must begin with '{'");
        }
        while (true) {
            c = x.nextClean();
            switch (c) {
            case 0:
                throw x.syntaxError("A JSONObject must end with '}'");
            case '}':
                return;
            default:
                x.back();
                key = x.nextValue().toString();
                if (x.nextClean() != ':') {
                    throw x.syntaxError("Expected a ':' after a key");
                }
                myHashMap.put(key, x.nextValue());
                switch (x.nextClean()) {
                case ',':
                    if (x.nextClean() == '}') {
                        return;
                    }
                    x.back();
                    break;
                case '}':
                    return;
                default:
                    throw x.syntaxError("Expected a ',' or '}'");
                }
            }
        }
    }
    /**
 * Construir un JSONObject de una cadena.
 * @ ParseException La excepción debe ser el formato adecuado.

```

```

    * @ Param cadena Una cadena que comienza con '(' y termina con
    * ')'.
    */
public JSONObject(String string) throws ParseException {
    this(new JSNTokener(string)); }
/**
    * Construir un JSONObject de un mapa.
    * @ Param mapa un mapa de objetos que se pueden utilizar para
    * inicializar el contenido de
    * El JSONObject.
    */
public JSONObject(Map map) {
    myHashMap = new HashMap(map);}
/**
    * Acumula valores en virtud de una tecla. Es similar al método
    * que, excepto
    * Que si ya hay un objeto almacenado en la clave entonces una
    * JSONArray se almacena en la clave para mantener el acumulado de
    * todos los valores.
    * Si ya hay una JSONArray, entonces el nuevo valor se añade a la
    * misma.
    * En cambio, el método que reemplaza el valor anterior.
    * @ Lanza NullPointerException si la clave es nula
    * @ Param clave Una de las principales cadenas.
    * @ Param valor Un objeto que se acumula bajo la clave.
    * @ Return presente.
    */
public JSONObject accumulate(String key, Object value)
    throws NullPointerException {
    JSONArray a;
    Object o = opt(key);
    if (o == null) {
        put(key, value);
    } else if (o instanceof JSONArray) {
        a = (JSONArray)o;
        a.put(value);
    } else {
        a = new JSONArray();
        a.put(o);
        a.put(value);
        put(key, a);
    } return this; }
/**
* Obtener el valor asociado a un objeto clave.
    * @ NoSuchElementException excepción si la clave no se encuentra.
    *
    * @ Param clave Una de las principales cadenas.
    * @ Return El objeto asociado con la tecla.
    */

```

```

public Object get(String key) throws NoSuchElementException {
    Object o = opt(key);
    if (o == null) {
        throw new NoSuchElementException("JSONObject[" +
            quote(key) + "] not found.");
    }
    return o;
}
/**
 * Obtener el valor booleano asociado a una tecla.
 * @ NoSuchElementException excepción si la clave no se encuentra.
 * @ Excepción ClassCastException
 * Si el valor no es un booleano o la cadena "verdadero" o
 "falso".
 *
 * @ Param clave Una de las principales cadenas.
 * @ Return La verdad.
 */
public boolean getBoolean(String key)
    throws ClassCastException, NoSuchElementException {
    Object o = get(key);
    if (o == Boolean.FALSE || o.equals("false")) {
        return false;
    } else if (o == Boolean.TRUE || o.equals("true")) {
        return true;
    }
    throw new ClassCastException("JSONObject[" +
        quote(key) + "] is not a Boolean.");
}
/**
 * Obtener el doble valor asociado a una tecla.
 * @ NoSuchElementException excepción si la clave no se encuentra
 o
 * Si el valor es un número de objetos.
 * @ NumberFormatException excepción si el valor no puede ser
 convertida a una
 * Número.
 * @ Param clave Una de las principales cadenas.
 * @ Return El valor numérico.
 */
public double getDouble(String key)
    throws NoSuchElementException, NumberFormatException {
    Object o = get(key);
    if (o instanceof Number) {
        return ((Number)o).doubleValue();
    }
    if (o instanceof String) {
        return new Double((String)o).doubleValue();
    }
    throw new NumberFormatException("JSONObject[" +
        quote(key) + "] is not a number.");
}
/**
 * Obtener el HashMap sostiene que el contenido de la JSONObject.
 * @ Return El getHashMap.
 */

```

```

    HashMap getHashMap() {
        return myHashMap; }
/**
Int * Obtener el valor asociado a una tecla.
    * @ NoSuchElementException excepción si la clave no se encuentra
    * @ Excepción NumberFormatException
    * Si el valor no se pueden convertir a un número.
    *
    * @ Param clave Una de las principales cadenas.
    * @ Return El valor entero.
    */
public int getInt(String key)
    throws NoSuchElementException, NumberFormatException {
    Object o = get(key);
    if (o instanceof Number) {
        return ((Number)o).intValue(); }
    return (int)getDouble(key); }
/**
    * @ NoSuchElementException excepción si la clave no se encuentra o
    * Si el valor no es un JSONArray.
    *
    * @ Param clave Una de las principales cadenas.
    * @ Return Un JSONArray, que es el valor.
    */
public JSONArray getJSONArray(String key) throws
    NoSuchElementException {
    Object o = get(key);
    if (o instanceof JSONArray) {
        return (JSONArray)o; }
    throw new NoSuchElementException("JSONObject[" +
        quote(key) + "] is not a JSONArray."); }
/**
* Obtener el JSONObject valor asociado a una tecla.
    * @ NoSuchElementException excepción si la clave no se encuentra
    o
    * Si el valor no es un JSONObject.
    *
    * @ Param clave Una de las principales cadenas.
    * @ Return Un JSONObject, que es el valor.
    */
public JSONObject getJSONObject(String key) throws
    NoSuchElementException {
    Object o = get(key);
    if (o instanceof JSONObject) {
        return (JSONObject)o; }
    throw new NoSuchElementException("JSONObject[" +
        quote(key) + "] is not a JSONObject."); }
/**
* Obtener la cadena asociada a una tecla.

```

```

    * @ NoSuchElementException excepción si la clave no se encuentra.
    *
    * @ Param clave Una de las principales cadenas.
    * @ Return una cadena que es el valor.
    */
    public String getString(String key) throws NoSuchElementException {
        return get(key).toString(); }
    /**
    * Determine si el JSONObject contiene una clave específica.
    * @ Param clave Una de las principales cadenas.
    * @ Return true si la clave existe en el JSONObject.
    */
    public boolean has(String key) {
        return myHashMap.containsKey(key); }
    /**
    * Determine if the value associated with the key is null or if
    * there is
    * no value.
    * @param key    A key string.
    * @return      true if there is no value associated with the key
    * or if
    * the value is the JSONObject.NULL object.
    */
    public boolean isNull(String key) {
        return JSONObject.NULL.equals(opt(key)); }
    /**
    * Obtener una enumeración de las claves de la JSONObject.
    *
    * @ Return Un iterador de las claves.
    */
    public Iterator keys() {
        return myHashMap.keySet().iterator(); }
    /**
    * Obtener el número de claves almacenadas en la JSONObject.
    *
    * @ Return El número de llaves en la JSONObject.
    */
    public int length() {
        return myHashMap.size(); }
    /**
    * Producir un JSONArray con los nombres de los elementos de este
    * JSONObject.
    * @ Return Un JSONArray que contiene la clave de las cadenas, o
    * nulo si la JSONObject
    * Está vacía.
    */
    public JSONArray names() {
        JSONArray ja = new JSONArray();
        Iterator keys = keys();

```

```

        while (keys.hasNext()) {
            ja.put(keys.next());    }
        if (ja.length() == 0) {
            return null;
        }
        return ja; }
    /**
 * Producir una cadena de un número.
 * @ ArithmeticException JSON excepción sólo puede serializar
 * números finitos.
 * @ Param n Un Número
 * @ Return una cadena.
 */
static public String numberToString(Number n) throws
    ArithmeticException {
    if (
        (n instanceof Float &&
            (((Float)n).isInfinite() || ((Float)n).isNaN())) ||
        (n instanceof Double &&
            (((Double)n).isInfinite() || ((Double)n).isNaN())))
    {
        throw new ArithmeticException(
            "JSON can only serialize finite numbers.");    }
    // ceros fuera de punto decimal, si es posible.
    String s = n.toString().toLowerCase();
    if (s.indexOf('e') < 0 && s.indexOf('.') > 0) {
        while (s.endsWith("0")) {
            s = s.substring(0, s.length() - 1);
        } if (s.endsWith(".")) {
            s = s.substring(0, s.length() - 1);    }
    } return s;    }
    /**
 * Obtener una opcional valor asociado a una tecla.
 * @ Excepción NullPointerException La clave no debe ser nulo.
 * @ Param clave Una de las principales cadenas.
 * @ Return Un objeto que es el valor, o null si no hay ningún
 * valor.
 */
public Object opt(String key) throws NullPointerException {
    if (key == null) {
        throw new NullPointerException("Null key");
    } return myHashMap.get(key);    }
    /**
 * Obtener un booleano opcional asociadas con una tecla.
 * Devuelve false si no existe dicha clave, o si el valor no es
 * Boolean.TRUE o la cadena "verdadero".
 *
 * @ Param clave Una de las principales cadenas.
 * @ Return La verdad.
 */

```

```

    public boolean optBoolean(String key) {
        return optBoolean(key, false); }
    /**
 * Obtener un booleano opcional asociadas con una tecla.
 * Devuelve la defaultValue si no existe dicha clave, o si no se
 * Un booleano o la cadena "verdadero" o "falso".
 *
 * @ Param clave Una de las principales cadenas.
 * @ Param defaultValue El valor por defecto.
 * @ Return La verdad.
 */
    public boolean optBoolean(String key, boolean defaultValue) {
        Object o = opt(key);
        if (o != null) {
            if (o == Boolean.FALSE || o.equals("false")) {
                return false;
            } else if (o == Boolean.TRUE || o.equals("true")) {
                return true; }
        } return defaultValue; }
    /**
 * Obtener una doble opcional asociada con una clave,
 * NaN o si no existe dicha clave, o si su valor no es un número.
 * Si el valor es una cadena, se hará un intento para evaluar como
 * Un número.
 *
 * @ Param clave Una cadena que es la clave.
 * @ Return Un objeto que es el valor.
 */
    public double optDouble(String key) {
        return optDouble(key, Double.NaN); }
    /**
 * Obtener una doble opcional asociada con una clave, o la
 * DefaultValue si no existe dicha clave, o si su valor no es un
 * número.
 * Si el valor es una cadena, se hará un intento para evaluar como
 * Un número.
 *
 * @ Param clave Una de las principales cadenas.
 * @ Param defaultValue El valor por defecto.
 * @ Return Un objeto que es el valor.
 */
    public double optDouble(String key, double defaultValue) {
        Object o = opt(key);
        if (o != null) {
            if (o instanceof Number) {
                return ((Number)o).doubleValue();
            } try {
                return new Double((String)o).doubleValue(); }
            catch (Exception e) {

```



```

        }
    }return defaultValue;
}
/**
 * Obtener un facultativo int valor asociado a una tecla,
 * Cero o, si no existe tal tecla o si el valor no es un número.
 * Si el valor es una cadena, se hará un intento para evaluar como
 * Un número.
 *
 * @ Param clave Una de las principales cadenas.
 * @ Return Un objeto que es el valor.
 */
public int optInt(String key) {
    return optInt(key, 0); }
/**
 * Obtener un facultativo int valor asociado a una tecla,
 * O el valor por defecto si no existe esa clave o si el valor no
 * es un número.
 * Si el valor es una cadena, se hará un intento para evaluar como
 * Un número.
 *
 * @ Param clave Una de las principales cadenas.
 * @ Param defaultValue El valor por defecto.
 * @ Return Un objeto que es el valor.
 */
public int optInt(String key, int defaultValue) {
    Object o = opt(key);
    if (o != null) {
        if (o instanceof Number) {
            return ((Number)o).intValue();
        }
        try {
            return Integer.parseInt((String)o);
        } catch (Exception e) { }
    } return defaultValue; }
/**
 * Obtener una JSONArray opcional asociada con una tecla.
 * Vuelve nula, si no existe tal tecla, o si su valor no es un
 * JSONArray.
 *
 * @ Param clave Una de las principales cadenas.
 * @ Return Un JSONArray, que es el valor.
 */
public JSONArray optJSONArray(String key) {
    Object o = opt(key);
    if (o instanceof JSONArray) {
        return (JSONArray) o;
    } return null; }
/**

```

```

    * Obtener una JSONObject opcional asociada con una tecla.
    * Vuelve nula, si no existe tal tecla, o si su valor no es un
    * JSONObject.
    *
    * @ Param clave Una de las principales cadenas.
    * @ Return Un JSONObject, que es el valor.
    */
    public JSONObject optJSONObject(String key) {
        Object o = opt(key);
        if (o instanceof JSONObject) {
            return (JSONObject)o;
        } return null; }
    /**
    * Obtener una cadena opcional asociada con una tecla.
    * Devuelve una cadena vacía si no existe tal tecla. Si el valor
    no es
    * Una cadena y no es nulo, entonces se convierte en una cadena.
    *
    * @ Param clave Una de las principales cadenas.
    * @ Return una cadena que es el valor.
    */
    public String optString(String key) {
        return optString(key, ""); }
    /**
    * Obtener una cadena opcional asociada con una tecla.
    * Devuelve la defaultValue si no existe tal tecla.
    *
    * @ Param clave Una de las principales cadenas.
    * @ Param defaultValue El valor por defecto.
    * @ Return una cadena que es el valor.
    */
    public String optString(String key, String defaultValue) {
        Object o = opt(key);
        if (o != null) {
            return o.toString(); }
        return defaultValue; }
    /**
    * Ponga una clave / boolean par en el JSONObject.
    *
    * @ Param clave Una de las principales cadenas.
    * @ Param Un valor booleano, que es el valor.
    * @ Return presente.
    */
    public JSONObject put(String key, boolean value) {
        put(key, new Boolean(value));
        return this;}
    /**
    * Ponga una clave o doble par en el JSONObject.
    *

```

```

    * @ Param clave Una de las principales cadenas.
    * @ Param un doble valor, que es el valor.
    * @ Return presente.
    */
    public JSONObject put(String key, double value) {
        put(key, new Double(value));
        return this;    }
    /**
    * Ponga una clave / int par en el JSONObject.
    *
    * @ Param clave Una de las principales cadenas.
    * @ Param int un valor que es el valor.
    * @ Return este.
    */
    public JSONObject put(String key, int value) {
        put(key, new Integer(value));
        return this;    }
    /**
    * Ponga una clave / valor par en el JSONObject. Si el valor es nulo,
    * Entonces la clave será removido de la JSONObject si está
    * presente.
    * @ Excepción NullPointerException La clave debe ser no nulo.
    * @ Param clave Una de las principales cadenas.
    * @ Param valor Un objeto que es el valor. Debe ser de una de
    * estas
    * Tipos: Boolean, Doble, Entero, JSONArray, JSONObject, de
    * cuerda, o la
    * JSONObject.NULL objeto.
    * @ Return presente.
    */
    public JSONObject put(String key, Object value) throws
        NullPointerException {
        if (key == null) {
            throw new NullPointerException("Null key.");    }
        if (value != null) {
            myHashMap.put(key, value);} else {
            remove(key);    }
        return this;    }
    /**
    * Ponga una clave / valor par en el JSONObject, pero sólo si la
    * El valor es no nulo.
    * @ Excepción NullPointerException La clave debe ser no nulo.
    * @ Param clave Una de las principales cadenas.
    * @ Param valor Un objeto que es el valor. Debe ser de una de
    * estas
    * Tipos: Boolean, Doble, Entero, JSONArray, JSONObject, de
    * cuerda, o la
    * JSONObject.NULL objeto.
    * @ Return presente.

```

```

    */
    public JSONObject putOpt(String key, Object value) throws
        NullPointerException {
        if (value != null) {
            put(key, value);
        } return this; }
    /**
    Producir una cadena entre comillas dobles con la barra en todas las
    secuencias
    * Derecho lugares.
    * @ Param cadena de cadenas
    * @ Retorno de cadenas el formato correcto para su inserción en
    un mensaje JSON.
    */
    public static String quote(String string) {
        if (string == null || string.length() == 0) {
            return "\"\"";
        }
        char        c;
        int          i;
        int          len = string.length();
        StringBuffer sb = new StringBuffer(len + 4);
        String       t;
        sb.append('');
        for (i = 0; i < len; i += 1) {
            c = string.charAt(i);
            switch (c) {
                case '\\':
                case '"':
                case '/':
                    sb.append('\\');
                    sb.append(c);
                    break;
                case '\b':
                    sb.append("\\b");
                    break;
                case '\t':
                    sb.append("\\t");
                    break;
                case '\n':
                    sb.append("\\n");
                    break;
                case '\f':
                    sb.append("\\f");
                    break;
                case '\r':
                    sb.append("\\r");
                    break;
                default:
                    if (c < ' ' || c >= 128) {
                        t = "000" + Integer.toHexString(c);
                        sb.append("\\u" + t.substring(t.length() - 4));
                    } else {
                        sb.append(c);
                    }
            }
        }
    }

```

```

        } sb.append('');
        return sb.toString();}
    /**
    * Borra el nombre y su valor, si está presente.
    * @ Param el nombre clave que deben ser extraídos.
    * @ Return El valor que se asocia con el nombre,
    * O null si no hay valor.
    */
    public Object remove(String key) {
        return myHashMap.remove(key);
    }
    /**
    * Producir un JSONArray que contengan los valores de los miembros de
    este
    * JSONObject.
    * @ Param A JSONArray nombres que contiene una lista de las
    principales cadenas. Esto
    * Determina la secuencia de los valores en el resultado.
    * @ Return Un JSONArray de valores.
    */
    public JSONArray toJSONArray(JSONArray names) {
        if (names == null || names.length() == 0) {
            return null;
        }
        JSONArray ja = new JSONArray();
        for (int i = 0; i < names.length(); i += 1) {
            ja.put(this.opt(names.getString(i)));
        } return ja; }
    /**
    * Hacer una cadena JSON forma externa de esta JSONObject. Por
    compacidad, no
    * Se añade en blanco innecesarios.
    * <p>
    * Advertencia: Este método supone que la estructura de datos es
    acyclical.
    *
    * @ Return una impresora, mostrar, portátiles, transmisibles
    * La representación del objeto, que comienza con '(' y termina
    con ')'.
    */
    public String toString() {
        Iterator keys = keys();
        Object o = null;
        String s;
        StringBuffer sb = new StringBuffer();
        sb.append('{');
        while (keys.hasNext()) {
            if (o != null) {
                sb.append(',');
            }

```

```

        }    s = keys.next().toString();
        o = myHashMap.get(s);
        if (o != null) {
            sb.append(quote(s));
            sb.append(':');
            if (o instanceof String) {
                sb.append(quote((String)o));
            } else if (o instanceof Number) {
                sb.append(numberToString((Number)o));
            } else {
                sb.append(o.toString());
            }
        }
        sb.append('}');
        return sb.toString();    }

/**
 * Hacer un prettyprinted JSON forma externa de esta cadena JSONObject.
 * <p>
 * Advertencia: Este método supone que la estructura de datos es
 * acyclical.
 * @ Param indentFactor El número de espacios para añadir a cada
 * nivel de
 * Indentación.
 * @ Return una impresora, mostrar, portátiles, transmisibles
 * La representación del objeto, que comienza con '(' y termina
 * con ')'. .
 */
public String toString(int indentFactor) {
    return toString(indentFactor, 0);    }

/**
 * Hacer un prettyprinted JSON JSONObject de esta cadena.
 * <p>
 * Advertencia: Este método supone que la estructura de datos es
 * acyclical.
 * @ Param indentFactor El número de espacios para añadir a cada
 * nivel de
 * Indentación.
 * @ Param guión El sangrado de las de nivel superior.
 * @ Return una impresora, mostrar, transmisibles
 * La representación del objeto, que comienza con '(' y termina
 * con ')'. .
 */
String toString(int indentFactor, int indent) {
    int    i;
    Iterator    keys = keys();
    String    pad = "";
    StringBuffer sb = new StringBuffer();
    indent += indentFactor;
    for (i = 0; i < indent; i += 1) {

```

```

        pad += ' ';
    }
    sb.append("{\n");
    while (keys.hasNext()) {
        String s = keys.next().toString();
        Object o = myHashMap.get(s);
        if (o != null) {
            if (sb.length() > 2) {
                sb.append(",\n");
            }
            sb.append(pad);
            sb.append(quote(s));
            sb.append(": ");
            if (o instanceof String) {
                sb.append(quote((String)o));
            } else if (o instanceof Number) {
                sb.append(numberToString((Number) o));
            } else if (o instanceof JSONObject) {
                sb.append(((JSONObject)o).toString(indentFactor, indent));
            } else if (o instanceof JSONArray) {
                sb.append(((JSONArray)o).toString(indentFactor, indent));
            } else {
                sb.append(o.toString());
            }
        }
    }
    sb.append('}');
    return sb.toString();
}

```

5.8 CLASE “JSONTokener.java” ESPECIE MATRIZ

```

package org.json;
import java.text.ParseException;
/**
 * Un JSONTokener toma una cadena de origen y sus extractos y fichas de
 * personajes
 * Ella. Es utilizado por el JSONObject y constructores para analizar
 * JSONArray
 * JSON fuente cuerdas.
 * <p>
 * Dominio Público JSON.org 2002
 * @ Autor JSON.org
 * @ Versión 0.1
 */public class JSONTokener {
    /**
 * El índice de la siguiente carácter.
 */private int myIndex;
    /**
 * La cadena de origen que se tokenized.
 */private String mySource;
}

```

```

* Construir un JSONTokener de una cadena.
*
* @ Param s Una cadena de origen.
*/
public JSONTokener(String s) {
    myIndex = 0;
    mySource = s;
}
/**
* Copia de seguridad de un carácter. Esto proporciona una especie de
lookahead capacidad,
* De modo que usted puede probar un dígito o una carta antes de
intentar analizar
* El próximo número o identificador.
*/
public void back() {
    if (myIndex > 0) {
        myIndex -= 1; } }
/**
* Obtener el valor hexadecimal de un carácter (base16).
* @ Param c Un carácter entre '0' 'y' '9' o entre 'A' y 'F' o
* Entre 'a' y 'f'.
* @ Return Un int entre 0 y 15, o -1 si c no es un dígito
hexadecimal.
*/
public static int dehexchar(char c) {
    if (c >= '0' && c <= '9') {
        return c - '0'; }
    if (c >= 'A' && c <= 'F') {
        return c + 10 - 'A'; }
    if (c >= 'a' && c <= 'f') {
        return c + 10 - 'a'; }
    return -1; }
/**
* Determine si la cadena origen todavía contiene caracteres que
el próximo ()
* Puede consumir.
* @ Return true si no al final de la fuente.
*/
public boolean more() {
    return myIndex < mySource.length(); }
/**
* Obtener el siguiente carácter en la cadena de origen.
*
* @ Return El siguiente carácter, o 0 si pasado el final de la
cadena de origen.
*/
public char next() {
    char c = more() ? mySource.charAt(myIndex) : 0;

```



```

        myIndex += 1;
        return c; }
/**
 * Consumir el siguiente carácter, y comprobar que coincide con un
 * Carácter.
 * @ Lanza ParseException si el carácter no coincide.
 * @ Param c El carácter de partido.
 * @ Return El carácter.
 */
public char next(char c) throws ParseException {
    char n = next();
    if (n != c) {
throw syntaxError("Expected '" + c + "' and instead saw '" +
                    n + "'.");
    }    return n; }
/**
 * Obtener el próximo n caracteres.
 * @ Excepción ParseException
 * Subcadena límites de error si no hay
 * N caracteres restantes en la cadena de origen.
 *
 * @ Param n El número de caracteres a tomar.
 * @ Return una cadena de n caracteres.
 */
public String next(int n) throws ParseException {
    int i = myIndex;
    int j = i + n;
    if (j >= mySource.length()) {
        throw syntaxError("Substring bounds error");
    }    myIndex += n;
    return mySource.substring(i, j); }
/**
 * Obtener el siguiente carácter en la cadena, dejando en blanco
 * Y comentarios (slashslash y slashstar).
 * @ Lanza ParseException
 * @ Return Un personaje, o 0 si no hay más caracteres.
 */
public char nextClean() throws java.text.ParseException {
    while (true) {
        char c = next();
        if (c == '/') {
            switch (next()) { case '/':
                do { c = next();
                } while (c != '\n' && c != '\r' && c != 0);
                break; case '*':
                while (true) {
                    c = next();
                    if (c == 0) {
                        throw syntaxError("Unclosed comment.");
                    }
                }
            }
        }
    }
}

```



```

        sb.append(c);  }
        break;
    default:
        if (c == quote) {
            return sb.toString();  }
        sb.append(c);  }
    }
}
/**
 * Obtener el texto, pero sin incluir el carácter o el
especificado
 * Final de la línea, lo que suceda primero.
 * @ Param d un carácter delimitador.
 * @ Return una cadena.
 */
public String nextTo(char d) {
    StringBuffer sb = new StringBuffer();
    while (true) {
        char c = next();
        if (c == d || c == 0 || c == '\n' || c == '\r') {
            if (c != 0) {
                back(); }
            return sb.toString().trim();
        }  sb.append(c);  }
    }
}
/**
 * Obtener el texto, pero no incluye una de las delimitador
especificado
 * Caracteres o el final de la línea, lo que suceda primero.
 * @ Param delimitadores delimitador de un conjunto de caracteres.
 * @ Return una cadena, recortada.
 */
public String nextTo(String delimiters) {
    char c;
    StringBuffer sb = new StringBuffer();
    while (true) {
        c = next();
        if (delimiters.indexOf(c) >= 0 || c == 0 ||
            c == '\n' || c == '\r') {
            if (c != 0) {  back(); }
            return sb.toString().trim(); }
        sb.append(c);  }
    }
}
/**
 * Obtener el siguiente valor. El valor puede ser un booleano,
Doble, Entero,
 * JSONArray, JSONObject, o de cuerda, o la JSONObject.NULL
objeto.

```

* @ Excepción ParseException La fuente se ajustan a la sintaxis de JSON.

*

* @ Return Un objeto.

*/

```
public Object nextValue() throws ParseException {
    char c = nextClean();
    String s;

    if (c == '"' || c == '\\') {
        return nextString(c);
    }
    if (c == '{') { back();
        return new JSONObject(this); }
    if (c == '[') { back();
        return new JSONArray(this); }
    StringBuffer sb = new StringBuffer();
    char b = c;
    while (c >= ' ' && c != ':' && c != ',' && c != ']' && c != '}' &&
           c != '/') {
        sb.append(c); c = next();
    } back();
    s = sb.toString().trim();
    if (s.equals("true")) { return Boolean.TRUE; }
    if (s.equals("false")) { return Boolean.FALSE; }
    if (s.equals("null")) {
        return JSONObject.NULL; }
    if ((b >= '0' && b <= '9') || b == '.' || b == '-' || b == '+') {
        try {
            return new Integer(s);
        } catch (Exception e) {
        } try {
            return new Double(s); } catch (Exception e) {
        }
    } if (s.equals("")) {throw syntaxError("Missing value.");
    } return s; }
}
```

/**

* Pasar caracteres hasta el siguiente carácter es el carácter solicitado.

* Si el carácter solicitado no se encuentra, no se saltan los personajes.

* @ Param a un personaje para saltar a la pista.

* @ Return El pedido carácter, o cero si el carácter

* No se encuentra.

*/

```
public char skipTo(char to) {
    char c;
    int index = myIndex;
    do { c = next();
```

```

        if (c == 0) {
            myIndex = index; return c;
        } while (c != to);
        back(); return c;    }
    /**
    * Pasar caracteres hasta pasado el pedido de cadena.
    * Si no lo encuentra, nos encontramos a la izquierda al final de
    la fuente.
    * @ Param a una cuerda para saltar pasado.
    */
    public void skipPast(String to) {
        myIndex = mySource.indexOf(to, myIndex);
        if (myIndex < 0) {
            myIndex = mySource.length();
        } else { myIndex += to.length(); }
    }
    /**
    * Haga una señal ParseException a un error de sintaxis.
    *
    * @ Param mensaje El mensaje de error.
    * @ Return un objeto ParseException, apta para lanzar
    */
    public ParseException syntaxError(String message) {
        return new ParseException(message + toString(), myIndex);
    }
    /**
    * Hacer una impresora de esta cadena JSONTokener.
    *
    * @ Return "en el carácter [MYIndex] de [mySource]"
    */
    public String toString() {
        return " at character " + myIndex + " of " + mySource;
    }
    /**
    * Unescape el texto de origen. Hh% convertir secuencias de
    caracteres,
    * Y así convertir al espacio. Hay web de sistemas de transporte
    que insisten en
    * Haciendo innecesaria la codificación URL. Esto proporciona una
    manera de deshacerlo.
    */
    public void unescape() {
        mySource = unescape(mySource);
    }
    /**
    * Convertir% hh secuencias solo a los personajes, y además de
    convertir al espacio.
    * @ Param s Una cadena que puede contener más y secuencias% hh.
    * @ Return El unescaped cadena.
    */
    public static String unescape(String s) {
        int len = s.length();

```

```

StringBuffer b = new StringBuffer();
for (int i = 0; i < len; ++i) {
    char c = s.charAt(i);
    if (c == '+') { c = ' ';
    } else if (c == '%' && i + 2 < len) {
        int d = dehexchar(s.charAt(i + 1));
        int e = dehexchar(s.charAt(i + 2));
        if (d >= 0 && e >= 0) {
            c = (char)(d * 16 + e);    i += 2; }
        } b.append(c);    }return b.toString(); }
}

```

6. PÁGINAS

Son las pantallas que permiten al usuario interactuar con la aplicación.

6.1 PÁGINA "INDEX.JSP"

La primera pantalla de la herramienta, esta es la que permite dar o no el acceso al usuario.



Figura 6.1 Acceso al sistema

```

<%@ page contentType="text/html; charset=utf-8" language="java"
import="java.sql.*" errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Bienvenidos</title>
<link href="css/estilos.css" rel="stylesheet" type="text/css" />
<script src="scripts/md5.js"
type="text/javascript"></script>
<script src="scripts/index.js"
type="text/javascript"></script>
<style type="text/css">
<!--
body {
    background-color: #7793FF;
    background-image: url(IMAGENES/fondo.png);
    background-repeat: repeat-x;}
-->
</style>
<script src="scripts/AC_RunActiveContent.js"
type="text/javascript"></script>
</head>
<body>
<table width="696" height="255" border="0" align="center"
cellpadding="0" cellspacing="0">
<tr>
<td width="599" height="171">&nbsp;</td>
<td width="599"><script type="text/javascript">
AC_FL_RunContent(
'codebase','http://download.macromedia.com/pub/shockwave/cabs/flash/swf
lash.cab#version=9,0,28,0','width','599','height','169','src','proyecto
','wmode','transparent','quality','high','pluginspage','http://www.adob
e.com/shockwave/download/download.cgi?P1_Prod_Version=ShockwaveFlash','
movie','proyecto' ); //end AC code
</script><noscript><object classid="clsid:D27CDB6E-AE6D-11cf-96B8-
444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swfla
sh.cab#version=9,0,28,0" width="599" height="169">
<param name="movie" value="proyecto.swf" />
<param name="quality" value="high" />
<param name="wmode" value="transparent">
<embed src="proyecto.swf" wmode=transparent quality="high"
pluginspage="http://www.adobe.com/shockwave/download/download.cgi?P1_Pr
od_Version=ShockwaveFlash" type="application/x-shockwave-flash"
width="599" height="169"></embed>
</object>
</noscript></td>

```

```

        <td width="599">&nbsp;</td>
    </tr>
    <tr> <td>&nbsp;</td>
        <td bgcolor="#FFFFFF"><form id="form1" name="form1" method="post"
action="acceder.jsp">
        <p>&nbsp;</p>
        <table width="367" border="0" align="center">
            <tr> <td colspan="3"><div align="center"
class="estilos">Identificación de Usuario</div></td>
            </tr> <tr> <td width="81" class="estilos">Usuario :</td>
                <td width="180"><input name="txt_usuario" type="text"
id="txt_usuario" size="30" maxlength="10" /></td>
                <td width="92" rowspan="3"></td>
            </tr> <tr> <td class="estilos">Clave :</td>
                <td><input name="txt_clave" type="password" id="txt_clave"
size="30" maxlength="10" /></td>
            </tr> <tr> <td height="21">&nbsp;</td>
                <td><button value="Ingresar" class="botones"
onclick="validar_usuarios()" type="button">  Ingresar</button>
                <input type="hidden" name="pass" id="pass" />
                <input type="hidden" name="numero" id="numero" /></td>
            </tr>
            <tr> <td>&nbsp;</td> <td>&nbsp;</td> <td>&nbsp;</td>
            </tr> <tr> <td>&nbsp;</td> <td>&nbsp;</td> <td>&nbsp;</td>
            </tr> <tr> <td>&nbsp;</td> <td>&nbsp;</td> <td>&nbsp;</td>
            </tr> <tr> <td>&nbsp;</td> <td>&nbsp;</td> <td>&nbsp;</td>
            </tr> <tr> <td>&nbsp;</td> <td>&nbsp;</td> <td>&nbsp;</td>
            </tr>
        </table>
    </form> </td>
    <td>&nbsp;</td>
    </tr> <tr> <td>&nbsp;</td> <td><div align="center">
        <script type="text/javascript">
AC_FL_RunContent(
'codebase','http://download.macromedia.com/pub/shockwave/cabs/flash/swf
lash.cab#version=9,0,28,0','width','600','height','81','src','pie','qua
lity','high','pluginspage','http://www.adobe.com/shockwave/download/dow
nload.cgi?Pl_Prod_Version=ShockwaveFlash','scale','exactfit','movie','p
ie' ); //end AC code
        </script>
        <noscript>
            <object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swfla
sh.cab#version=9,0,28,0" width="600" height="81">
                <param name="movie" value="pie.swf" />
                <param name="quality" value="high" />

```



```

        <param name="SCALE" value="exactfit" />
        <embed src="pie.swf" width="600" height="81" quality="high"
pluginpage="http://www.adobe.com/shockwave/download/download.cgi?Pl_Pr
od_Version=ShockwaveFlash" type="application/x-shockwave-flash"
scale="exactfit"></embed>
    </object>
</noscript>
</div></td>
<td>&nbsp;</td>
</tr>
</table>
<script language="JavaScript">
    document.forms["form1"].txt_usuario.focus();
</script>
</body>
</html>

```

6.2 PÁGINA "PRINCIPAL.JSP"

Esta ventana permite una vez validado el usuario ingresar al sistema y por los privilegios otorgador podra realizar los accesos a cada menu y podra interactuar con el sistema.



Figura 6.2 Página principal

```

<jsp:include page="seguridad.jsp" flush="true" ></jsp:include>
<%@ page contentType="text/html; charset=utf-8" language="java"
import="java.sql.*" errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Bienvenidos</title>
<link href="css/estilos.css" rel="stylesheet" type="text/css" />
<style type="text/css">
<!--
body {
    background-color: #7793FF;
    background-image: url(IMAGENES/fondo.png);
    background-repeat: repeat-x;}
-->
</style>
</head>
<body>
<jsp:include page="cabecera.jsp" flush="true"/>
<table width="200" border="0" align="center" cellpadding="1"
cellspacing="0" bgcolor="#FFFFFF">
    <tr>
        <td><iframe src="centro.jsp" width="730" height="400"
name="trabajo" frameborder="1" ></iframe></td> </tr>
</table>
<jsp:include page="abajo.jsp" flush="true"/>
</body>
</html>

```

6.3 PÁGINA "ABRIR_REPORTES.JSP"

Esta ventana una vez elegido la opción exportar y ya sea las opciones de visualización que elija de los buzones de entrada, salida, no enviados, borradores y ya sea en los formatos pdf o en archivos excel levantará una pagina con la información solicitada.

 <div> REPORTE DE BUZÓN DE ENTRADA Entrada: Psp-411 20-JUL-2009 </div>			
DISPOSITIVOS	NOMBRE	MENSAJES	FECHA
001B4F32E05D	Jul Loh	093506050.ayudante redes	20-03-09 04:22 PM
001B4F32E05D	Jul Loh	093506050.ayudante redes	20-03-09 04:24 PM
001B4F32E05D	Jul Loh	093506050.ayudante redes	20-03-09 04:25 PM
001D4F32E05D	Jul Loh	093506050.nc.vedades.seminarios	20-03-09 04:40 PM
001B4F32E05D	Jul Loh	093506050.nc.vedades.seminarios	20-03-09 04:47 PM
001B4F32E05D	Jul Loh	093506050.nc.vedades.seminarios	20-03-09 04:48 PM

Figura 6.3 Reporte buzón de entrada

```

<%@ page import="net.sf.jasperreports.engine.*" %>
<%@ page import="net.sf.jasperreports.engine.design.JasperDesign"%>
<%@ page import="net.sf.jasperreports.engine.xml.JRXmlLoader" %>
<%@ page import="net.sf.jasperreports.engine.export.*" %>
<%@ page import="net.sf.jasperreports.engine.export.JExcelApiExporter"
%>
<%@ page import="java.util.*" %>
<%@ page import="java.io.*" %>
<%
Clases.Conexion conne=new Clases.Conexion();
conne.conectar();String reporte=request.getParameter("reporte");
String dir_imagen="/IMAGENES/frontal.png";
File ruta_imagen= new File(application.getRealPath(dir_imagen));
File reportFile = new File(application.getRealPath("/"+reporte));
Map parameters = new HashMap(); parameters.put("ruta",
ruta_imagen.getPath());
if(reporte.compareTo("buzones.jasper")==0){
    parameters.put("estado",request.getParameter("tipo"));
    parameters.put("reporte",request.getParameter("repo"));
    parameters.put("f1",request.getParameter("f1"));
    parameters.put("f2",request.getParameter("f2"));
}
int operacion=(new
Integer(request.getParameter("operacion"))).intValue();
if(operacion==0){byte[] bytes
=JasperRunManager.runReportToPdf(reportFile.getPath(),parameters,conne.
getConexion()); response.setContentType("application/pdf");
    response.setContentLength(bytes.length);
ServletOutputStream ouputStream = response.getOutputStream();
    ouputStream.write(bytes, 0, bytes.length);
    ouputStream.flush();ouputStream.close();}
if(operacion==1){

```

```

        JasperPrint jasperPrint=JasperFillManager.fillReport
(reportFile.getPath(), parameters, conne.getConexion());
        String xlsFileName = "reporte.xls"; JRXlsExporter exporter
= new JRXlsExporter ();
exporter.setParameter(JRXlsExporterParameter.JASPER_PRINT,
jasperPrint);
exporter.setParameter(JRXlsExporterParameter.IS_DETECT_CELL_TYPE,
Boolean.TRUE);
exporter.setParameter(JRXlsExporterParameter.IS_WHITE_PAGE_BACKGROUND,
Boolean.FALSE);
exporter.setParameter(JRXlsExporterParameter.IS_REMOVE_EMPTY_SPACE_BETW
EEN_ROWS, Boolean.TRUE);
exporter.setParameter(JRXlsExporterParameter.OUTPUT_STREAM,
response.getOutputStream()); exporter.exportReport ();File f = new File
( xlsFileName); response.setContentType ("application/vnd.ms-excel");
//Tipo de fichero.
response.setHeader ("Content-Disposition", "attachment;filename=\"\" +
xlsFileName + "\""); //Configurar cabecera http
        InputStream in = new FileInputStream (f);
        ServletOutputStream out1 = response.getOutputStream();
        int bit = 256;
        while ((bit)>= 0){    bit = in.read ();    out1.write (bit);
        }out1.flush ();out1.close (); in.close ();
    }conne.desconectar();
%>

```

6.4 PÁGINA "ACCEDER.JSP"

Esta ventana saldrá cuando en la validación del usuario no exista el usuario o la contraseña sea mal ingresada. Con esta ventada esta bloqueando el acceso a usuarios no autorizados.

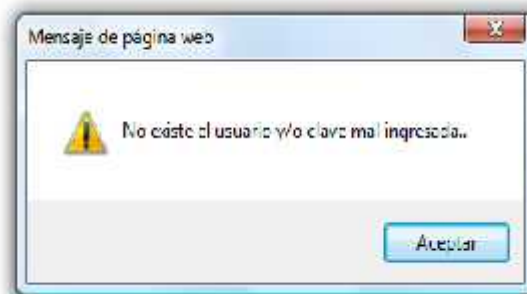


Figura 6.4 Ventana de error de validación de usuario

```

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%
Clases.Funciones objeto = new Clases.Funciones();
String usuario=request.getParameter("txt_usuario");
String clave=request.getParameter("pass");
String numero=request.getParameter("numero");
if(true==objeto.verifica_login(usuario, clave, numero).booleanValue()){
    session.setAttribute("usuario", usuario);
    objeto.graba_acceso(usuario,"0");
    out.println("<meta http-equiv=Refresh content=\ '0 ;
url=principal.jsp\ '>"); }else{out.println("<script>");
    out.println("alert('No existe el usuario y/o clave mal
ingresada..')"); out.println("</script>");
    out.println("<meta http-equiv=Refresh content=\ '0 ;
url=index.jsp\ '>"); }
%>

```

6.5 PÁGINA "BANDEJAS.JSP"

Esta ventana mostrará las opciones de tipo de reporte para exportar la información, imprimir, y buscar por opciones de visualización.

Del buzón que se seleccione: los buzones de entrada, salida y no enviados al presionar el boton buscar aparecerá la información solicitada.

Dispositivo	Nombre	Mensaje	Fecha
-------------	--------	---------	-------

Figura 6.5 Bandeja de entrada

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html lang="en">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <meta http-equiv="Content-Style-Type" content="text/css">
    <meta http-equiv="Content-Script-Type" content="text/javascript">
    <script type="text/javascript" src="scripts/prototype.js"
language="javascript"></script>
    <script type="text/javascript" src="scripts/prototype.js"
language="javascript"></script>
    <script type="text/javascript" src="popcalendar.js"
language="javascript"></script>
    <script src="jquery-1.1.3.1.pack.js" type="text/javascript"></script>
    <script src="jquery.history_remote.pack.js"
type="text/javascript"></script>
    <script src="jquery.tabs.pack.js" type="text/javascript"></script>
    <script src="scripts/jsonrpc.js" type="text/javascript"></script>
    <script src="scripts/buzones.js" type="text/javascript"></script>
    <script type="text/javascript" src="scripts/dispositivos.js"
language="javascript"></script>
    <script type="text/javascript">
      $(function() {
        $('#container-1').tabs();
        $('#container-2').tabs(2);
        $('#container-3').tabs({ fxSlide: true });
        $('#container-4').tabs({ fxFade: true, fxSpeed: 'fast'
});
        $('#container-5').tabs({ fxSlide: true, fxFade: true, fxSpeed: 'normal'
});
        $('#container-6').tabs({
          fxFade: true,
          fxSpeed: 'fast',
          onClick: function() { alert('onClick'); },
          onHide: function() { alert('onHide'); },
          onShow: function() { alert('onShow'); }
});
        $('#container-7').tabs({ fxAutoHeight: true });
        $('#container-8').tabs({ fxShow: { height: 'show',
opacity: 'show' }, fxSpeed: 'normal' });
        $('#container-9').tabs({ remote: true });
        $('#container-10').tabs();
        $('#container-11').tabs({ disabled: [3] });
        $('<p><a href="#">Disable third
tab</a></p>').prependTo('#fragment-28').find('a').click(function() {
          $(this).parents('div').eq(1).disableTab(3);
          return false;
        });
        $('<p><a href="#">Activate third
tab</a></p>').prependTo('#fragment-28').find('a').click(function() {

```

```

        $(this).parents('div').eq(1).triggerTab(3);
        return false;    });
        $('<p><a href="#">Enable third
tab</a></p>').prependTo('#fragment-28').find('a').click(function() {
        $(this).parents('div').eq(1).enableTab(3);
        return false;    });
    });
</script>
<link href="css/estilos.css" rel="stylesheet" type="text/css" />
<link rel="stylesheet" href="jquery.tabs.css" type="text/css" >
<!-- Additional IE/Win specific style sheet
(Conditional Comments) -->
<!--[if lte IE 7]>
<link rel="stylesheet" href="jquery.tabs-ie.css"
type="text/css" media="projection, screen">
<![endif]-->
<style type="text/css" media="screen, projection">
/* Not required for Tabs, just to make this demo look better... */
    body {    font-size: 16px; /* @ EOMB */
    }    * html body {    font-size: 100%; /* @ IE */
    } body * {    font-size: 87.5%;
    font-family: "Trebuchet MS", Trebuchet, Verdana, Helvetica, Arial,
    sans-serif; }
    body * * {    font-size: 100%;    }
    h1 {    margin: 1em 0 1.5em;
        font-size: 18px;    }
    h2 {    margin: 2em 0 1.5em;
        font-size: 16px;    }
    p {    margin: 0;    }
    pre, pre+p, p+p {    margin: 1em 0 0;    }
    code { font-family: "Courier New", Courier, monospace; }
</style>
<title>Bandejas</title>
</head>
<body onload="cargando()">
<h1> Buzones
Bluetooth</h1>
<div id="container-9">
<ul>
<li><a href="entrada.jsp"><span>Entrada</span></a></li>
<li><a href="salida.jsp"><span>Salida</span></a></li>
<li><a href="no_enviados.jsp"><span>No Enviados</span></a></li>
<li><a href="borradores.jsp"><span>Borradores</span></a></li>
<li><a href="contador.jsp"><span>Contador</span></a></li>
</ul>
</div>
</body>
</html>

```



```

        <form id="form1" name="form1" method="post" action="">
        <table width="700" border="0" align="center" cellpadding="1"
        cellspacing="0" bordercolor="#CCCCCC" bgcolor="#CCCCCC">
        <tr><td class="mensajes"> BORRADORES</td> </tr>
        </table> <table width="704" border="0" align="center">
        <tr> <td width="128"><button value="Ingresar" class="botones"
        onclick="window.location='ing_borradores.jsp'" type="button">  Ingresar</button></td>
        <td width="134"><button value="Ingresar" class="botones"
        onclick="mod_borrador()" type="button">  Modificar</button></td>
        <td width="130"><button value="Ingresar" class="botones"
        onclick="borrar_borrador()" type="button">  Eliminar</button></td>
        <td width="142"><button value="Ingresar" class="botones"
        onclick="imprimir()" type="button">  Imprimir</button></td>
        <td width="148"><input type="hidden" name="usuario"
        id="usuario"></td>
        </tr> </table>
        <table width="700" border="1" align="center" cellpadding="1"
        cellspacing="0" bordercolor="#000000">
        <tr> <td width="35" bgcolor="#CCCCCC"></td>
        <td width="232" bgcolor="#CCCCCC"><div
        align="center"><strong>Titulo</strong></div></td>
        <td width="419" bgcolor="#CCCCCC"><div
        align="center"><strong>Contenido</strong></div></td>
        </tr>
        <%
        Vector consul=consulta.Consulta_borradores();
        for(int i=0;i<consul.size();i++){
            Vector info=(Vector)consul.elementAt(i);
            out.println("<tr>");
            out.println("<td><input type='radio' name='radio' id='radio'
            onclick='captura(\""+info.elementAt(0).toString()+"\")' /></td>");
            out.println("<td>"+info.elementAt(1).toString()+"</td>");
            out.println("<td>"+info.elementAt(2).toString()+"</td>");
            out.println("</tr>"); }
        %>
        </table>
    </form>
</body>

</html>

```

6.7 PÁGINA "CABECERA.JSP"

Esta parte que presenta la pagina es un gif animado de la aplicación.



Figura 6.7 Cabecera del sistema

```
<%@ page contentType="text/html; charset=utf-8" language="java"
import="java.sql.*" errorPage="" %>
<style type="text/css">
<!--
.style1 {
    font-family: Verdana, Arial, Helvetica, sans-serif;
    font-weight: bold;
    font-size: 12px;}
-->
</style>
<script src="scripts/cerrar_sesion.js"
type="text/javascript"></script>
<script src="scripts/AC_RunActiveContent.js"
type="text/javascript"></script>
<table width="884" height="198" border="0" align="center"
cellpadding="0" cellspacing="0">
    <tr>
        <td width="78" height="171">&nbsp;</td>
        <td width="746"><script type="text/javascript">
AC_FL_RunContent(
'codebase', 'http://download.macromedia.com/pub/shockwave/cabs/flash/swf
lash.cab#version=9,0,28,0', 'width', '728', 'height', '173', 'src', 'proyecto
', 'wmode', 'transparent', 'quality', 'high', 'pluginspage', 'http://www.adob
e.com/shockwave/download/download.cgi?Pl_Prod_Version=ShockwaveFlash', '
scale', 'exactfit', 'movie', 'proyecto' ); //end AC code
</script> <noscript>
    <object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swfla
sh.cab#version=9,0,28,0" width="728" height="173">
        <param name="movie" value="proyecto.swf" />
        <param name="wmode" value="transparent">
        <param name="quality" value="high" />
        <param name="SCALE" value="exactfit" />
        <embed src="proyecto.swf" wmode="transparent" width="728"
height="173" quality="high">
    </object>
</noscript>
</td>
    </tr>
</table>
```

```

pluginspage="http://www.adobe.com/shockwave/download/download.cgi?P1_Pro
od_Version=ShockwaveFlash" type="application/x-shockwave-flash"
scale="exactfit"></embed>
</object>
</noscript>
</td> <td width="60">&nbsp;</td> </tr>
<tr> <td height="21">&nbsp;</td>
<td><table width="727" border="0" cellpadding="0" cellspacing="0"
bgcolor="#FFFFFF">
<tr> <td width="112"><span class="style1">Bienvenido :
</span></td> <td width="471"><span class="style1">
<%
        Clases.Usuarios objeto=new Clases.Usuarios();
        String codigo=(String)session.getAttribute("usuario");
        out.println(objeto.datos_personas(codigo));
        %>
</span> </td>
<td width="110"><a href="#" onclick="terminar()"
onmouseover="document.cuatro.src='IMAGENES/cerrarover.png';"
onmouseout="document.cuatro.src='IMAGENES/cerrar.png';"></a></td>
</tr>
</table></td>
<td>&nbsp;</td>
</tr> <tr> <td height="21" >&nbsp;</td>
<td ><table width="98%" border="0" cellpadding="1" cellspacing="0"
bgcolor="#FFFFFF">
<tr><td><%@ include file="menu.jsp" %></td>
</tr>
</table></td>
<td>&nbsp;</td>
</tr>
</table>

```

6.8 PÁGINA "CENTRO.JSP"

Una vez ingresado al menu principal cuando se valido el ingreso del usuario

mostrara en la parte central de la pagina una imagen de bienvenida presentando

el titulo de la aplicación.



Figura 6.8 Centro del sistema

```
<style type="text/css">
<!--
.style1 {
    font-size: 19px;
    font-weight: bold;
    color: #0000FF;}
.style3 {font-size: 25px}
.style4 {
    font-size: 24px}
-->
</style>
<p>&nbsp;&nbsp;&nbsp;</p>
<p>&nbsp;&nbsp;&nbsp;</p>
<table width="352" height="238" border="0" cellpadding="0"
cellspacing="0" bgcolor="#FFFFFF" align="center">
    <tr>  <td width="710"> <div align="center" class="style1">Bienvenidos
: </div></td>
    </tr>  <tr>
        <td><div align="center"><span class="style3">Servidor
Bluetooth</span></div></td>
    </tr>  <tr>
        <td><div align="center" class="style4">Universidad de
Guayaquil</div></td>
    </tr>  <tr>
        <td><div align="center"></div></td>
    </tr>  <tr>
        <td><div align="center"><span class="style4">Facultad Fisicas y
Matematicas </span></div></td>
    </tr></table>
```

6.9 PÁGINA "CON_PRIVILEGIOS.JSP"

Esta ventana que esta en el menú privilegios muestra los tipos de privilegios creados en el sistema, estos tambien pueden ser modificados o eliminados. Tambien esta la opción para imprimir.

CONSULTA DE PRIVILEGIOS DE USUARIOS			
 Modificar  Eliminar  Imprimir			
	Codigo	Nombre	Fecha de Creación
	1	ADMINISTRADOR	2008-01-09
	2	OPERADOR	2009-01-09
	3	CONSULTOR	2009-02-02
	4	AUDITOR	2009-06-09
	5	USUARIO 2	2009-06-18

Figura 6.9 Privilegios

```
<%@ page contentType="text/html; charset=utf-8" language="java"
import="java.sql.*" errorPage="" %>
<jsp:useBean id="JSONRPCBridge" scope="session"
class="com.metaparadigm.jsonrpc.JSONRPCBridge" />
<jsp:useBean id="permisos" scope="session" class="Clases.Permisos" />
<% JSONRPCBridge.registerObject("permisos", permisos); %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Consultas De Privilegios</title>
    <link href="css/estilos.css" rel="stylesheet" type="text/css" />
    <script src="scripts/permisos.js"></script>
    <script src="scripts/jsonrpc.js" type="text/javascript"></script>
  </head>
  <body onLoad="cargando()">
    <form name="form1" method="post" action="mod_privilegios.jsp">
      <table width="641" border="0" align="center" cellpadding="1"
cellspacing="0" bordercolor="#CCCCCC" bgcolor="#CCCCCC">
        <tr>
          <td class="mensajes"> CONSULTA DE PRIVILEGIOS DE USUARIOS</td>
        </tr>
      </table>
    </form>
  </body>
</html>
```

```

        <table width="649" border="0" align="center">
        <tr>
            <td width="142">
                <button value="Ingresar" class="botones"
                onclick="permiso_modifica()" type="button">  Modificar</button>
            </td>
            <td width="126">
                <button value="Ingresar" class="botones"
                onclick="elimina_permisos()" type="button">  Eliminar</button>
            </td>
            <td width="367">
                <button value="Ingresar" class="botones"
                onclick="imprimir()" type="button">  Imprimir</button>
                <input type="hidden" name="permiso" id="permiso"></td>
        </tr>
        </table> </form>
        <table width="641" border="1" align="center"
        cellpadding="1" cellspacing="0" bordercolor="#000000">
        <tr>
            <td width="32" bgcolor="#CCCCCC"><div align="center"></div></td>
            <td width="66" bgcolor="#CCCCCC"><div
            align="center"><strong>Codigo</strong></div></td>
            <td width="390" bgcolor="#CCCCCC"><div
            align="center"><strong>Nombre </strong></div></td>
            <td width="135" bgcolor="#CCCCCC"><div align="center"><strong>Fecha
            de Creación</strong></div></td>
        </tr>
        <%
        Clases.Conexion cone= new Clases.Conexion();
        cone.conectar(); int e=0;
        java.sql.ResultSet listado=cone.Consultas("select * from permisos
        order by codigo asc");
        while(listado.next()){
            out.println("<tr id='fila"+String.valueOf(e)+"'
            onMouseOver='enciende("+String.valueOf(e)+" ,1)'
            onMouseOut='apaga("+String.valueOf(e)+" ,1)'>");
            out.println("<td> <input type='radio' name='codigo'
            value='"+listado.getString("codigo")+"' id='RadioGroup1_0'
            onClick='elementos("+listado.getString("codigo")+")' /></td>");
            out.println(" <td>"+listado.getString("codigo")+"</td>");
            out.println(" <td>"+listado.getString("nombre")+"</td>");
            out.println(" <td>"+listado.getString("fecha")+"</td>");
            //out.println(" <td>"+listado.getString("privilegio")+"</td>");
            out.println("</tr>"); e++; } cone.desconectar();
        %>
        </table>
        <p>
        </p>
        </body>
    </html>

```

6.10 PÁGINA "CON_PROGRAMACION.JSP"

Esta pagina permite consultar las peticiones que podrán ser enviadas desde un dispositivos bluetooth, con la peticion y argumento.



Codigo	Peticion	Argumento	Respuesta
1	Ayudantia	Reclas	Sabados y domingos de 10:00 a 12:00 en el Lab 1
2	Noticias	1er semestre	Sabados 20 de junio en la sala de actividades
3	Noticias	2do semestre	Profesor Eduardo Alvarado no asistirá durante esta semana
4	Noticias	Deportivas	Mes de Julio Campeonato Interno de Fútbol
5	Noticias	Seminarios	Seminario de CISO Informes al 2564765 - Ing. Abel Alarcón
6	Exámenes	1 parcial	del 22 de junio al 26 de junio
7	Exámenes	2 parcial	del 7 al 11 de septiembre
8	Laboratorio	A111	6:00 - 7:00 de lunes a viernes
9	Laboratorio	LAB2	8:00 - 9:00 martes y jueves

Figura 6.10 Peticiones

```
<%@ page contentType="text/html; charset=utf-8" language="java"
import="java.sql.*" errorPage="" %>
<jsp:useBean id="JSONRPCBridge" scope="session"
class="com.metaparadigm.jsonrpc.JSONRPCBridge" />
<jsp:useBean id="peticion" scope="session" class="Clases.Peticiones"
/>
<% JSONRPCBridge.registerObject("peticion", peticion); %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Consultas Peticiones</title>
<link href="css/estilos.css" rel="stylesheet" type="text/css" />
<script src="scripts/jsonrpc.js" type="text/javascript"></script>
<script src="scripts/peticiones.js"
type="text/javascript"></script>
</head>
<body onload="cargando()">
<form name="form1" method="post" action="mod_usuarios.jsp">
<table width="633" border="0" align="center" cellpadding="1"
cellspacing="0" bordercolor="#CCCCCC" bgcolor="#CCCCCC">
```

```

        <tr> <td class="mensajes"> CONSULTAS DE PETICIONES</td>
        </tr> </table> <table width="649" border="0" align="center">
        <tr> <td width="120">
            <button value="Ingresar" class="botones"
onclick="elimina_petcciones()" type="button">  Eliminar</button>
        </td> <td width="156">
            <button value="Ingresar" class="botones" onclick="imprimir()"
type="button"> Imprimir</button>
        </td> <td width="359"><input type="hidden" name="usuario"
id="usuario"></td>
        </tr></table>
</form> <table width="641" border="1" align="center" cellpadding="1"
cellspacing="0" bordercolor="#000000">
<tr> <td width="20" bgcolor="#CCCCCC"><div align="center"></div></td>
        <td width="118" bgcolor="#CCCCCC"><div
align="center"><strong>Peticion</strong></div></td>
        <td width="130" bgcolor="#CCCCCC"><div
align="center"><strong>Argumento</strong></div></td>
        <td width="300" bgcolor="#CCCCCC"><div
align="center"><strong>Respuesta</strong></div></td>
        </tr> <%
        Clases.Conexion cone= new Clases.Conexion();
        cone.conectar();
        java.sql.ResultSet listado=cone.Consultas("select * from peticiones
order by id asc");
        int e=0; while(listado.next()){
            out.println("<tr id='fila'+String.valueOf(e)+' '
onMouseOver='enciende('"+String.valueOf(e)+"',1)'
onMouseOut='apaga('"+String.valueOf(e)+"',1)'+>");
            out.println("<td> <input type='radio' name='codigo'
value='"+listado.getString("id")+"' id='RadioGroup1_0'
onClick='elementos('"+listado.getString("id")+"' )' /></td>");
            out.println(" <td>"+listado.getString("peticion")+"</td>");
            out.println(" <td>"+listado.getString("argumento")+"</td>");
            out.println(" <td>"+listado.getString("respuesta")+"</td>");
            out.println("</tr>");e++; }
        cone.desconectar();
        %> </table>
    </body>
</html>

```


6.11 PÁGINA "CON_USUARIOS.JSP"

Esta ventana permitira la consulta de los usuarios del sistema, en el cual se podrá exportar el documento por medio de formatos pdf y en archivo excel. Se podrá modificar y eliminar los usuarios; también tiene la opción imprimir.

CONSULTA DE USUARIOS DEL SISTEMA					
 Modificar  Eliminar  Imprimir Exportar: Pdf  Exportar					
Usuario	Nombres	Apellidos	Celular	Privilegio	
admin	JULIO CESAR	CHOLZ CATUTO	097344225	ADMINISTRADOR	
gehaez	GIOMAR PAOLA	CHÓEZ CATUTO	090192403	OPERADOR	
szambrano	WASHINGTON SAUL	ZAMBRANO RIVAS	099148655	CONSULTOR	
nicun	MARCELO JAVIER	LEÓN GONZALEZ	095685645	AUDITOR	
juanc	JUAN CARLOS	CARDENAS VALDEZ	094921718	CONSULTOR	

Figura 6.11 Consulta de usuarios del sistema

```

<%@page contentType="text/html" pageEncoding="UTF-8"
import=" java.sql.*"%>
<jsp:useBean id="JSONRPCBridge" scope="session"
class="com.metaparadigm.jsonrpc.JSONRPCBridge" />
<jsp:useBean id="usuarios" scope="session" class="Clases.Usuarios" />
<% JSONRPCBridge.registerObject("usuarios", usuarios); %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <link href="css/estilos.css" rel="stylesheet" type="text/css" />
    <script src="scripts/jsonrpc.js" type="text/javascript"></script>
    <script language="javascript" src="scripts/usuarios.js"></script>
    <title>Consultas de Usuarios</title>
  </head>
  <body onLoad="cargando()">
    <table width="641" border="0" align="center" cellpadding="1"
cellspacing="0" bordercolor="#CCCCCC" bgcolor="#CCCCCC">
      <tr> <td class="mensajes"> CONSULTA DE USUARIOS DEL
SISTEMA</td> </tr> </table>
      <form name="form1" method="post" action="mod_usuarios.jsp">
        <table width="649" border="0" align="center">

```

```
|  |  |  |
| --- | --- | --- |
| <button value="Ingresar" class="botones" onclick="usuarios_modifica()" type="button">  Modificar</button> | <button value="Ingresar" class="botones" onclick="elimina_usuarios()" type="button">  Eliminar</button> | <button value="Ingresar" class="botones" onclick="imprimir()" type="button">  Imprimir</button> <input type="hidden" name="usuario" id="usuario"></td> |

</tr></table></form>





<% Clases.Conexion cone= new Clases.Conexion();
cone.conectar(); java.sql.ResultSet listado=cone.Consultas("select
codigo,usuario,nombres,apellidos,celular,funciones(8,permiso,0) as
privilegio from usuarios order by codigo asc");
int e=0; while(listado.next()){
out.println("<tr id='fila"+String.valueOf(e)+"'
onMouseOver='enciende(\""+String.valueOf(e)+"\",1)'
onMouseOut='apaga(\""+String.valueOf(e)+"\",1)'>");
out.println("<td> <input type='radio' name='codigo'
value='\""+listado.getString(\"codigo\")+\"' id='RadioGroup1_0'
onClick='elementos(\""+listado.getString(\"codigo\")+\"') /></td>");
out.println("<td>"+listado.getString(\"usuario\")+"</td>");
out.println("<td>"+listado.getString(\"nombres\")+"</td>");
out.println("<td>"+listado.getString(\"apellidos\")+"</td>");
out.println("<td>"+listado.getString(\"celular\")+"</td>");
out.println("<td>"+listado.getString(\"privilegio\")+"</td>");
out.println("</tr>");e++; } cone.desconectar();
%> </table>
</body>
</html>

```

6.12 PÁGINA "CONSULTA_MENSAJES.JSP"

esta ventana mostrará el contenido de la búsqueda de los mensajes según la visualización escogida.

Visualización :

Dispositivo	Nombre	Mensaje	Fecha
001BAFC75A5E	Milenita	0919598342.notas	2009-06-05 22:40:22.0
001BAFC75A5F	Milenita	090192403.novidades.primary	2009-06-08 09:48:17.0
001BAFC75A5F	Milenita	090192403.ayudencia	2009-06-09 17:04:25.0
001BAFC75A5E	Milenita	090192403.ayudencia	2009-06-09 17:05:02.0
001BAFC75A5E	Milenita	090192403.horario	2009-06-09 17:14:42.0
001BAFC75A5E	Milenita	090192403.laboratorio	2009-06-09 17:18:44.0
001BAFC75A5F	Milenita	090192403.laboratorio	2009-06-09 17:19:15.0
001BAFC75A5F	Milenita	090192403.novidades	2009-06-09 17:21:07.0
001BAFC75A5E	Milenita	090192403.novidades	2009-06-09 17:22:43.0
001BAFC75A5E	Milenita	090192403.novidades	2009-06-09 17:22:51.0
001BAFC75A5E	Milenita	090192403.novidades.seminarios	2009-06-09 17:37:01.0
001BAFC75A5F	Milenita	090192403.novidades.seminarios	2009-06-09 17:37:46.0

Figura 6.12 Consulta de mensajes

```
<table width="663" border="1" align="center" cellpadding="1"
cellspacing="0" bordercolor="#000000">
  <tr>
    <td width="129" bgcolor="#CCCCCC"><div
align="center"><strong>Dispositivo</strong></div></td>
    <td width="129" bgcolor="#CCCCCC"><div
align="center"><strong>Nombre</strong></div></td>
    <td width="150" bgcolor="#CCCCCC"><div
align="center"><strong>Mensaje</strong></div></td>
    <td width="177" bgcolor="#CCCCCC"><div align="center"><strong>Fecha
</strong></div></td>
  </tr>
  <%
    Clases.Mensajes consulta= new Clases.Mensajes();
    java.util.Vector
    consul=consulta.Consulta_mensaje(request.getParameter("estado"),
    request.getParameter("f1"),request.getParameter("f2"));
    for(int i=0;i<consul.size();i++){
      java.util.Vector info=(java.util.Vector)consul.elementAt(i);
      out.println("<tr>");out.println("<td>"+info.elementAt(1).toString()+"</
td>"); out.println(" <td>"+info.elementAt(3).toString()+"</td>");
      out.println("<td> "+info.elementAt(0).toString()+"</td>");
      out.println(" <td>"+info.elementAt(2).toString()+"</td>");
      out.println("</tr>"); }
  %>
</table>
```

6.13 PÁGINA "CONTADOR.JSP"

Esta pagina permitirá mostrar un contador gráfico de mensajes enviados recibidos y no enviados. Tambien se podra imprimir según la opción de la visualización.

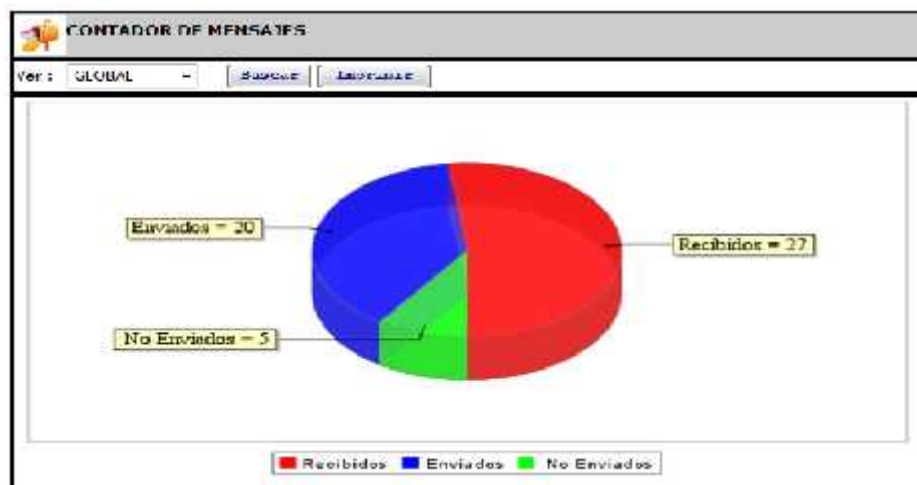


Figura 6.13 Contador de mensajes

```
<%@ page contentType="text/html; charset=utf-8" language="java"
import="java.util.*, java.sql.*, java.text.*, java.io.*" errorPage="" %>
<jsp:useBean id="consulta" scope="session" class="Clases.Dispositivos"
/>
<jsp:useBean id="fecha" scope="session" class="Clases.Fechas" />
<jsp:useBean id="grafico"
class="org.jfree.chart.demo.servlet.WebHitChart"/>
<%
    Vector consul=consulta.Consulta_mensajes();
    String contenido="";
    contenido="<?xml version='1.0' encoding='ISO-8859-1'?'>";
        for(int i=0;i<consul.size();i++){
            Vector info=(Vector)consul.elementAt(i);
            contenido=contenido+"<dispositivo>"+info.elementAt(1).toString()+"</dis
positivo><numero>"+info.elementAt(0).toString()+"</numero>";
        }
    String filename = grafico.generate3DPieChart(contenido, session, new
    PrintWriter(out));
    String graphURL = request.getContextPath() +
    "/servlet/DisplayChart?filename=" + filename;
```

```

String imagen="<img src='"+graphURL+"' width=650 height=350 border=0
usemap='#"+filename+"'>";
%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
    <link href="css/estilos.css" rel="stylesheet" type="text/css"
/>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
        <title>JSP Page</title>
        <style>
            .L {
                position:absolute;
                left:277px; top:134px;
                width:257px; height:26px;
                z-index:1; overflow: hidden;
                visibility: hidden;}
            .L1 {
                position:inherit;
                left:435px; top:313px;
                width:250px; height:26px;
                z-index:1; overflow: visible;
                visibility: visible;background-color: #FFFFFF;}
        </style>
    </head>
    <script type="text/javascript" src="scripts/prototype.js"
language="javascript"></script>
    <script language="javascript">
        function ver_grafica(){
            if(document.form1.ver.value==0){
                ver_grafico('2008-01-01','2010-12-30'); }
            if(document.form1.ver.value==1){
                ver_grafico('2009-06-01','<%=fecha.f1%>'); }
            if(document.form1.ver.value==2){
                ver_grafico('<%=fecha.f5%>','<%=fecha.f5%>'); }
            if(document.form1.ver.value==3){
                ver_grafico('<%=fecha.f1%>','<%=fecha.f1%>'); }
            if(document.form1.ver.value==4){
                ver_grafico('<%=fecha.f2%>','<%=fecha.f1%>'); }
        }
        function ver_grafico(f1,f2){
            var aleatorio = Math.random();
            var url = 'funciones.jsp';CambiarEstilo()
var pars = 'f1='+f1+'&f2='+f2+'&aleatorio='+aleatorio+'&funcion=2';
            var myAjax = new Ajax.Request( url,{ method: 'post',
parameters: pars,onComplete: Respuesta12 });
        }
    </script>

```

```

function Respuesta12 (oRequest){
    CambiarEstilo1()
    $('capaexpansion').innerHTML = oRequest.responseText;    }
    function CambiarEstilo() {
        var elemento = document.getElementById("L");
        elemento.className = "L1";}
    function CambiarEstilo1() {
        var elemento = document.getElementById("L");
        elemento.className = "L";}
    </script>
<body><form name="form1" method="post" action="">
    <table width="649" border="0" align="center" cellpadding="1"
cellspacing="0" bordercolor="#CCCCCC" >
        <tr>        <td class="mensajes" bgcolor="#CCCCCC">CONTADOR DE MENSAJES</td>
        </tr>    <tr>
        <td class="mensajes"><table width="643" border="0"
cellspacing="0" cellpadding="1">
            <tr>    <td width="43">Ver :</td>
            <td width="132"><select name="ver" id="ver">
                <option value="0">GLOBAL</option>
                <option value="1">MES ACTUAL</option>
                <option value="4">SEMANAL</option>
                <option value="2">AYER</option>
                <option value="3">HOY</option>
            </select>        </td>
            <td width="65"><input name="button2" type="button"
class="botones" id="button" value="Buscar"
onClick="ver_grafica()"></td>
            <td width="395"><input name="button" type="button"
class="botones" id="button2" value="Imprimir" onClick="window.print()"
/>    </td>    <td width="395"> <div class="L" id=L >
<strong>Por favor espere un momento... </strong></div> </td>
            </tr>    </table></td>
        </tr> <tr> <td class="mensajes"><table width="200" border="0"
align="center">    <tr>    <td><div id=capaexpansion>
                <%=imagen%>
            </div>    </td>    </tr>
        </table></td>    </tr> </table>
</form></body>
</html>

```

6.14 PÁGINA "ENTRADA.JSP"

Esta ventana permite realizar la búsqueda de los mensajes de entrada según las opciones de visualización se podrá exportar e imprimir la información seleccionada.



Figura 6.14 Buzón de entrada

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<%@page contentType="text/html" pageEncoding="UTF-8"
import="java.util.*" %>
<jsp:useBean id="consulta" scope="session" class="Clases.Dispositivos"
/><jsp:useBean id="fecha" scope="session" class="Clases.Fechas" />
<script src="scripts/formexp.js" type="text/javascript"></script>
<script type="text/javascript" src="scripts/prototype.js"
language="javascript"></script>
<script type="text/javascript" src="popcalendar.js"
language="javascript"></script>
<script src="scripts/buzones.js" type="text/javascript"></script>
<script type="text/javascript" src="scripts/dispositivos.js"
language="javascript"></script>
<link href="css/estilos.css" rel="stylesheet" type="text/css" />
```

```

    <body> <form name="form1" method="post" action="salida.jsp">
    <table width="658" border="0" align="center" cellpadding="1"
cellspacing="0" bordercolor="#CCCCCC" bgcolor="#CCCCCC">
        <tr> <td class="mensajes"> BUZON DE ENTRADA</td>
        </tr> </table>
        <table width="643" border="0" align="center">
            <tr> <td width="107" class="estilos">Reporte en : </td>
            <td width="537"> <select name="operacion" class="caja_texto"
id="operacion"> <option value="0">Pdf</option>
                <option value="1">Excel</option>
            </select> <button value="Ingresar" class="botones"
onclick="exportar_reporte()" type="button">  Exportar</button>
            <button value="Ingresar" class="botones" onclick="imprimir()"
type="button"> Imprimir</button>
            <button value="Ingresar" class="botones" onclick="buscar()"
type="button">  Buscar</button>
            </td> </tr> <tr>
                <td colspan="2" class="estilos">
                <table width="644" border="0">
                    <tr> <td width="105">Visualizaci&oacute;n : </td>
                    <td width="529"><select name="reportes" id="reportes"
onchange="ver_reporte()">
                        <option value="1" selected>DIARIO</option>
                        <option value="2">SEMANA</option>
                        <option value="3">QUINCENAL</option>
                        <option value="4">MENSUAL</option>
                        <option value="5">PERSONALIZADO</option> </select>
                        <input type="hidden" name="reporte" value="buzones.jasper" />
                        <input type="hidden" name="tipo" value="0" />
                        <input type="hidden" name="repo" value="ENTRADA" />
                        <input name="fecha_hoy" type="hidden" id="fecha_hoy"
value="<%=fecha.f1%>">
                        <input name="fecha_semana" type="hidden" id="fecha_semana"
value="<%=fecha.f2%>">
                        <input name="fecha_quincena" type="hidden" id="fecha_quincena"
value="<%=fecha.f3%>">
                        <input name="fecha_mes" type="hidden" id="fecha_mes"
value="<%=fecha.f4%>"></td>
                    </tr></table><div id="capaexpansion">
                        <table width="648" border="0" cellspacing="0" cellpadding="1">
                            <tr> <td width="106">Fecha desde :</td>
                            <td width="184"><input name="f1" type="text" id="f1"
readonly="true" value="">

```



```

</td>
<td width="103">Fecha hasta :</td>
<td width="170"><input name="f2" type="text" id="f2" readonly="true"
value="">
</td>
<td width="64">&nbsp;</td></tr>
</table></div> </td> </tr> </table>
<div id="consulta_divd">
<table width="663" border="1" align="center" cellpadding="1"
cellspacing="0" bordercolor="#000000">
<tr> <td width="129" bgcolor="#CCCCCC"><div
align="center"><strong>Dispositivo</strong></div></td>
<td width="129" bgcolor="#CCCCCC"><div
align="center"><strong>Nombre</strong></div></td>
<td width="150" bgcolor="#CCCCCC"><div
align="center"><strong>Mensaje</strong></div></td>
<td width="177" bgcolor="#CCCCCC"><div
align="center"><strong>Fecha </strong></div></td>
</tr>
<%
/*      Vector consul=consulta.Consulta_mensaje("0");
for(int i=0;i<consul.size();i++){
    Vector info=(Vector)consul.elementAt(i);
    out.println("<tr>");
    out.println("<td>"+info.elementAt(1).toString()+"</td>");
    out.println(" <td>"+info.elementAt(3).toString()+"</td>");
    out.println("<td> "+info.elementAt(0).toString()+"</td>");
    out.println(" <td>"+info.elementAt(2).toString()+"</td>");
    out.println("</tr>");
}*/      %>      </table>
<script>      oculta_div();
</script>      </div>
</form>      </body>

```

6.15 PÁGINA "ENV_DISPOSITIVOS.JSP"

Esta pagina permitirá ver en pantalla los dispositivos detectados dentro del rango, a estos se les podra enviar mensajes ya sea de los borradores o escrito en la caja de texto. Aquí se podra visualizar si el mensaje fue enviado o hubo un error en el envio ya sea por la no aceptación por parte del dueño del dispositivo o el dispositivo ya no esta en el alcance.



Figura 6.15 Envío de mensajes

```
<%@ page contentType="text/html; charset=utf-8" language="java"
import="javax.bluetooth.*" errorPage="" %>
<jsp:useBean id="JSONRPCBridge" scope="session"
class="com.metaparadigm.jsonrpc.JSONRPCBridge" />
<jsp:useBean id="bluetooth" scope="session"
class="Dispositivos.Envia_Mensaje" />
<jsp:useBean id="consulta" scope="session" class="Clases.Dispositivos"
/>
<% JSONRPCBridge.registerObject("bluetooth", bluetooth); %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>    <head>
        <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">        <title>Enviar Mensajes</title>
        <link href="css/estilos.css" rel="stylesheet" type="text/css" />
<script type="text/javascript" src="scripts/stlib.js"
language="JavaScript"></script>
        <script type="text/javascript" src="scripts/prototype.js"
language="javascript"></script>
        <script type="text/javascript" src="scripts/dispositivos.js"
language="javascript"></script>
        <script src="scripts/jsonrpc.js" type="text/javascript"></script>
        <style type="text/css">
<!--
.style1 {    font-family: Verdana, Arial, Helvetica, sans-serif;
            font-size: 12px;}
-->
    </style></head>
        <script src="scripts/dispositivos.js"
language="javascript"></script>
```

```

<body onload="cargando()">
    <form name="form1" method="post" action="">
        <table width="707" border="0" align="center">
            <tr> <td width="701"><table width="701" border="1" cellpadding="1"
cellspacing="0" bordercolor="#000000" bgcolor="#CCCCCC">
            <tr> <td class="estilos"> <div align="left" class="style1">ENVIAR MENSAJES
</div> <div class="L" id=L >
                <strong>Por favor espere un momento... </strong></div></td>
            </tr> </table> </td> </tr>
        </table> <table width="692" height="261" border="0" align="center">
            <tr> <td width="180"> <div id="dis">
                <select name="dispositivos" size="18" multiple id="dispositivos"
class="lista"> <% %> </select>
            </div> <input name="lista" type="checkbox" id="lista" value="1"
onClick="seleccionar_items()">
                <strong>Todos </strong>
            </td> <td width="504"><table width="457" border="0">
                <tr> <td width="104" class="letras">Borradores :</td>
                    <td width="372"><select name="borradores"
id="borradores" onChange="usar_borrador()">
                        <%out.println("<option value=' ' >Seleccione el borrador a
enviar</option>");
                            java.util.Vector consul=consulta.Consulta_borradores();
                            for(int i=0;i<consul.size();i++){
                                java.util.Vector info=(java.util.Vector)consul.elementAt(i);
                                out.println("<option value='"+info.elementAt(2).toString()+''
>"+info.elementAt(1).toString()+"</option>"); }
                                %> </select></td>
                            </tr> <tr> <td width="104" class="letras">Mensaje :</td>
                                <td width="372">&nbsp;</td> </tr>
                            <tr> <td colspan="2"><textarea name="mensaje" class="mensajes2"
id="mensaje"></textarea></td>
                            </tr> <tr> <td colspan="2">
                                <button value="Ingresar" class="botones"
onClick="enviar_mensaje()" type="button" name="button1" >  Enviar </button>
                                <button value="Ingresar" class="botones" type="reset"
name="button2">  Borrar </button>
                                <button value="Ingresar" class="botones" onClick="actualiza()"
type="button" name="button3"> Actualizar</button> <div align="right"></div></td>
                            </tr> <tr> <td class="estilos">Consola :</td>
                                <td>&nbsp;</td> </tr>

```

```

<td colspan="2"><textarea name="consola" cols="50" rows=" "
readonly="readonly" class="mensajes2" id="consola"></textarea></td>
</tr> <tr> <td colspan="2">&nbsp;</td> </tr>
<tr> <td colspan="2"> </td> </tr>
</table></td> </tr> </table> </form>
</body></html>

```

6.16 PÁGINA "ENV_PUBLICIDAD.JSP"

Esta pagina permitira mostrar las publicidades creadas por el usuario y el estado que tiene la publicidad. Se puede crear nuevas modificar y eliminar publicidades.



Figura 6.16 Consulta de envío de publicidad

```

<jsp:useBean id="JSONRPCBridge" scope="session"
class="com.metaparadigm.jsonrpc.JSONRPCBridge" />
<jsp:useBean id="publicidad" scope="session" class="Clases.Publicidad"
/>
<% JSONRPCBridge.registerObject("publicidad",publicidad); %>
<%@page contentType="text/html" pageEncoding="UTF-8"
import="java.util.*" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head> <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">

```

```

<title>JSP Page</title>
<link href="css/estilos.css" rel="stylesheet" type="text/css" />
<script src="scripts/jsonrpc.js" type="text/javascript"></script>
<script language="javascript" src="scripts/publicidad.js"></script>
</head>
<body onLoad="cargando()">
    <form name="form1" method="post" action="">
        <table width="641" border="0" align="center" cellpadding="1"
cellspacing="0" bordercolor="#CCCCCC" bgcolor="#CCCCCC">
            <tr>
                <td class="mensajes">CONSULTA DE ENVIO DE PUBLICIDAD
                <input type="hidden" name="usuario" id="usuario"/></td>
            </tr>
        </table>
        <table width="649" border="0" align="center">
            <tr>
                <td width="91"><button value="Ingresar"
class="botones" onclick="ingresar_publicidad()" type="button"> Nueva</button>
                </td>
                <td width="113"> <button value="Ingresar" class="botones"
onclick="eliminar_publicidad()" type="button"> Eliminar</button>
                </td><td width="108">
                    <button value="Ingresar" class="botones"
onclick="reporte_publicidad()" type="button"> Reporte</button>
                </td> <td width="319">
                    <button value="Ingresar" class="botones"
onclick="modificar_publicidad()" type="button"> Modificar</button>
                </td> </tr>
        </table>
        <table width="641" border="1" align="center" cellpadding="1"
cellspacing="0" bordercolor="#000000">
            <tr> <td width="32" bgcolor="#CCCCCC"><div
align="center"></div></td>
                <td width="81" bgcolor="#CCCCCC"><div
align="center"><strong>Codigo</strong></div></td>
                <td width="340" bgcolor="#CCCCCC"><div
align="center"><strong>Nombre de la Publicidad</strong></div></td>
                <td width="170" bgcolor="#CCCCCC"><div
align="center"><strong>Estado</strong></div></td> </tr>
            <%
                Vector consul=publicidad.Consulta_publicidad();
                for(int i=0;i<consul.size();i++){
                    Vector info=(Vector)consul.elementAt(i);

```

```

        out.println("<tr id='fila"+String.valueOf(i)+"'
onMouseOver='enciende("+String.valueOf(i)+",1)'
onMouseOut='apaga("+String.valueOf(i)+",1)'>");
        out.println("<td> <input type='radio' name='codigo'
value='"+info.elementAt(0).toString()+"' id='RadioGroup1_0'
onClick='elementos("+info.elementAt(0).toString()+"","\n"+info.elementAt(
1).toString()+"\n")' /></td>");
        out.println("<td> "+info.elementAt(0).toString()+"</td>");
        out.println("<td>"+info.elementAt(1).toString()+"</td>");
        if(info.elementAt(2).toString().compareTo("1")==0){
        out.println(" <td>INACTIVA</td>");
        }else{ out.println(" <td>ACTIVA</td>");}
out.println("</tr>");
    }
    %>    </table>
</form> </body> </html>

```

6.17 PÁGINA "ING_BORRADORES.JSP"

Esta ventana permite la creación de borradores que serán presentados como opciones de envío.

Figura 6.17 Ingreso de borradores

```

<%@page contentType="text/html" pageEncoding="UTF-8"
import="java.util.*" %>
<jsp:useBean id="consulta" scope="session" class="Clases.Dispositivos"
/>
<jsp:useBean id="JSONRPCBridge" scope="session"
class="com.metaparadigm.jsonrpc.JSONRPCBridge" />
<jsp:useBean id="mantenimiento" scope="session"
class="Clases.Borradores" />
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

```

```

<% JSONRPCBridge.registerObject("mantenimiento", mantenimiento); %>
<html>
    <head>    <script src="scripts/jsonrpc.js"
type="text/javascript"></script>
        <script type="text/javascript" src="scripts/prototype.js"
language="javascript"></script>
        <script src="scripts/buzones.js"
type="text/javascript"></script>
        <script src="scripts/borradores.js"
type="text/javascript"></script>
        <link href="css/estilos.css" rel="stylesheet" type="text/css"
/></head>    <body onload="cargando()">
        <form id="form1" name="form1" method="post" action="">
<table width="700" border="0" align="center" cellpadding="1"
cellspacing="0" bordercolor="#CCCCCC" bgcolor="#CCCCCC">
<tr>
    <td class="mensajes"> INGRESO DE BORRADORES</td>
</tr>
    </table>
    <table width="200" border="1" align="center" cellpadding="1"
cellspacing="0" bordercolor="#000000">
        <tr>
            <td><table width="691" border="0" align="center"
cellpadding="0" cellspacing="0">
                <tr> <td width="132">&nbsp;</td>
                <td width="118">&nbsp;</td>
                <td width="136">&nbsp;</td>
                <td width="164">&nbsp;</td>
                <td width="141">&nbsp;</td>
            </tr><tr> <td><strong>Titulo :</strong></td>
                <td colspan="2"><input name="titulo" type="text"
class="caja_texto" id="titulo" size="33"></td>
                <td>&nbsp;</td>
                <td>&nbsp;</td>
            </tr> <tr> <td><strong>Contenido :</strong></td>
                <td colspan="2"><textarea name="contenido" cols="30"
class="caja_texto" id="contenido"></textarea></td>
                <td>&nbsp;</td>
                <td>&nbsp;</td>
            </tr>
                <tr>
                    <td>&nbsp;</td> <td><button value="Ingresar"
class="botones" onclick="guarda_datos()" type="button">  Guardar</button></td>
                    <td><button value="Ingresar" class="botones"
onclick="window.location='borradores.jsp'" type="button">  Consultar</button></td>
                    <td><button value="Ingresar" class="botones"
onclick="limpiar()" type="button"> 
Limpiar</button></td>
    <td>&nbsp;</td>
</tr> <tr> <td>&nbsp;</td>
    <td>&nbsp;</td> <td>&nbsp;</td>
    <td>&nbsp;</td> </tr> <tr>
<td>&nbsp;</td> <td>&nbsp;</td>
<td>&nbsp;</td> <td>&nbsp;</td>
<td>&nbsp;</td> </tr> <tr> <td>&nbsp;</td>
<td>&nbsp;</td> <td>&nbsp;</td> <td>&nbsp;</td>
<td>&nbsp;</td> </tr>
    <tr> <td>&nbsp;</td> <td>&nbsp;</td>
<td>&nbsp;</td> <td>&nbsp;</td>
<td>&nbsp;</td> </tr>
    <tr> <td>&nbsp;</td> <td>&nbsp;</td> <td>&nbsp;</td>
<td>&nbsp;</td> <td>&nbsp;</td>
<td>&nbsp;</td> </tr>
    <tr> <td>&nbsp;</td> <td>&nbsp;</td> <td>&nbsp;</td>
<td>&nbsp;</td> <td>&nbsp;</td>
<td>&nbsp;</td> </tr>
    <tr> <td>&nbsp;</td> <td>&nbsp;</td> <td>&nbsp;</td>
<td>&nbsp;</td> <td>&nbsp;</td>
<td>&nbsp;</td> </tr>
    <tr> <td>&nbsp;</td> <td>&nbsp;</td> <td>&nbsp;</td>
<td>&nbsp;</td> <td>&nbsp;</td>
<td>&nbsp;</td> </tr>
</table></td>
</tr> </table>
</form> </body>
</html>

```

6.18 PÁGINA "ING_PRIVILEGIOS.JSP"

Esta ventana permie realizar el ingreso de privilegios si el usuario es tipo administrador, dependiendo de los diferentes menú podrá seleccionar las opciones diferentes que podra acceder el usuario que sera designado tal privilegio.

INGRESO DE PRIVILEGIOS DE USUARIOS

Nombre:

Menu:

PRIVILEGIO PARA ESE MODULO

- ☐ Ingresos Usuarios del Sistema
- ☐ Consultas Usuarios del Sistema
- ☐ Ingresos de Usuarios Externos
- ☐ Consultas de Usuarios Externos

Figura 6.18 Ingreso de privilegios

```
<%@ page contentType="text/html; charset=utf-8" language="java"
import="java.sql.*" errorPage="" %>
<jsp:useBean id="JSONRPCBridge" scope="session"
class="com.metaparadigm.jsonrpc.JSONRPCBridge" />
<jsp:useBean id="permisos" scope="session" class="Clases.Permisos" />
<% JSONRPCBridge.registerObject("permisos", permisos); %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
    <title>Ingreso de Privilegios</title>
    <link href="css/estilos.css" rel="stylesheet" type="text/css" />
    <!--[if lt IE 7]>
<script defer type="text/javascript" src="scripts/pngfix.js"></script>
<![endif]-->
    <script src="scripts/formexp.js"></script>
    <script src="scripts/permisos.js"></script>
    <script src="scripts/jsonrpc.js"
type="text/javascript"></script>
    <script>
      function desactiva(){
        xDisplay('capaexpansion2', 'none')
        xDisplay('capaexpansion3', 'none')
        xDisplay('capaexpansion4', 'none')
        xDisplay('capaexpansion5', 'none')
      }
      function ver_opcion(){
```

```

        if(document.form1.tipo.value==1){
            xDisplay('capaexpansion1', 'block')
            xDisplay('capaexpansion2', 'none')
            xDisplay('capaexpansion3', 'none')
            xDisplay('capaexpansion4', 'none')
            xDisplay('capaexpansion5', 'none')
        }
        if(document.form1.tipo.value==2){
            xDisplay('capaexpansion2', 'block')
            xDisplay('capaexpansion1', 'none')
            xDisplay('capaexpansion3', 'none')
            xDisplay('capaexpansion4', 'none')
            xDisplay('capaexpansion5', 'none')
        }
        if(document.form1.tipo.value==3){
            xDisplay('capaexpansion2', 'none')
            xDisplay('capaexpansion1', 'none')
            xDisplay('capaexpansion3', 'block')
            xDisplay('capaexpansion4', 'none')
            xDisplay('capaexpansion5', 'none')
        }
        if(document.form1.tipo.value==4){
            xDisplay('capaexpansion2', 'none')
            xDisplay('capaexpansion1', 'none')
            xDisplay('capaexpansion3', 'none')
            xDisplay('capaexpansion4', 'block')
            xDisplay('capaexpansion5', 'none')
        }
        if(document.form1.tipo.value==5){
            xDisplay('capaexpansion5', 'block')
            xDisplay('capaexpansion4', 'none')
            xDisplay('capaexpansion1', 'none')
            xDisplay('capaexpansion3', 'none')
            xDisplay('capaexpansion2', 'none')
        }
    }
</script>
</head>
<body onLoad="cargando()">
    <table width="523" height="391" border="0" align="center">
        <tr>
            <td><form name="form1" method="post"
action="ing_privilegios.jsp">
<table width="641" border="0" align="center" cellpadding="1"
cellspacing="0" bordercolor="#CCCCCC" bgcolor="#CCCCCC">
    <tr> <td class="mensajes"> INGRESO DE PRIVILEGIOS DE
USUARIOS</td> </tr></table><table width="624" border="0"
align="center">
    <tr> <td width="143" class="letras">Nombre :</td>

```

```

        <td width="165"><input name="txt_nombre" type="text"
class="caja_texto" id="txt_nombre" maxlength="20"></td>
        <td width="146">
            <button value="Ingresar" class="botones" onclick="verifica_items()"
type="button">  Guardar</button>
            </td> <td width="152">&nbsp;</td>
        </tr> <tr> <td class="letras">Menu :</td>
        <td><select name="tipo" class="caja_texto" id="tipo"
onchange="ver_opcion()">
            <option value="1" selected>USUARIOS</option>
            <option value="2">PRIVILEGIOS</option>
            <option value="3">ENVIOS</option>
            <option value="4">REPORTES</option>
            <option value="5">PETICIONES</option>
        </select> </td> <td>&nbsp;</td>
        <td>&nbsp;</td> </tr> </table>
        <table width="630" border="1" cellpadding="1"
cellspacing="0" bordercolor="#000000">
            <tr> <td><strong>PRIVILEGIO PARA ESE MODULO</strong></td>
            </tr> <tr> <td>
                <div id="capaexpansion1">
                    <table width="483" border="0" align="center"
cellpadding="1" cellspacing="0">
                        <tr> <td><input name="p1" type="checkbox" id="p1" value="1">
Ingresos Usuarios del Sistemas</td>
                        <td>&nbsp;</td> </tr> <tr>
                            <td><input name="p2" type="checkbox" id="p2" value="1">
Consultas Usuarios del Sistemas</td> <td>&nbsp;</td>
                        </tr> <tr> <td><input name="p3" type="checkbox" id="p3" value="1">
Ingresos de Usuarios Externos</td>
                        <td>&nbsp;</td>
                        </tr> <tr>
                            <td><input name="p4" type="checkbox" id="p4" value="1">
Consultas de Usuarios Externos</td>
                            <td>&nbsp;</td>
                        </tr> </table>
                    </div>
                    <table width="200" border="0" cellspacing="0" cellpadding="1">
                        <tr> <td>&nbsp;</td> </tr> <tr>
                            <td>&nbsp;</td>
                        </tr> <tr> <td>&nbsp;</td>
                        </tr> <tr> <td>&nbsp;</td>
                        </tr> <tr> <td>&nbsp;</td>
                        </tr> <tr> <td>&nbsp;</td>
                        </tr> </table>
                </div> <div id="capaexpansion2">
                    <table width="488" border="0" align="center" cellpadding="1"
cellspacing="0"> <tr>

```

```

        <td><input name="p5" type="checkbox" id="p5" value="1">
Ingresos Privilegios</td> <td>&nbsp;</td>
        </tr> <tr>
        <td><input name="p6" type="checkbox" id="p6" value="1">
Consultas Privilegios</td> <td>&nbsp;</td>
        </tr> </table>
<table width="200" border="0" cellspacing="0" cellpadding="1">
        <tr> <td>&nbsp;</td></tr>
        <tr><td>&nbsp;</td>
        </tr> <tr> <td>&nbsp;</td>
        </tr> <tr> <td>&nbsp;</td>
        </tr> <tr> <td>&nbsp;</td>
        </tr> <tr><td>&nbsp;</td>
        </tr></table> <p>&nbsp;</p>
</div> <div id=capaexpansion3>
<table width="491" border="0" align="center" cellpadding="1"
cellspacing="0"> <tr>
        <td><input name="p7" type="checkbox" id="p7" value="1">
        Envios de Mensajes </td>
        <td>&nbsp;</td>
        </tr> <tr>
        <td><input name="p8" type="checkbox" id="p8" value="1">
        Envios de Publicidad</td> <td>&nbsp;</td>
        </tr> </table>
<table width="200" border="0" cellspacing="0" cellpadding="1">
        <tr><td>&nbsp;</td></tr>
        <tr><td>&nbsp;</td>
        </tr> <tr> <td>&nbsp;</td>
        </tr> <tr> <td>&nbsp;</td>
        </tr><tr> <td>&nbsp;</td>
        </tr><tr><td>&nbsp;</td>
        </tr></table>
        <p>&nbsp;</p>
</div>
<div id=capaexpansion4>
<table width="491" border="0" align="center" cellpadding="1"
cellspacing="0"> <tr> <td><input name="p9" type="checkbox" id="p9"
value="1"> Dispositivos Encontrados</td> <td>&nbsp;</td>
        </tr> <tr>
        <td><input name="p10" type="checkbox" id="p10" value="1">
        Mas Utilizados</td><td>&nbsp;</td>
        </tr> <tr> <td><input name="p11" type="checkbox" id="p11" value="1">
        Auditoria</td><td>&nbsp;</td> </tr>
        <tr> <td><input name="p12" type="checkbox" id="p12" value="1">
        Buzon de Entrada</td> <td>&nbsp;</td> </tr> <tr>
        <td><input name="p13" type="checkbox" id="p13" value="1">
        Buzon de Salida</td><td>&nbsp;</td>

```


The screenshot shows a web form with a title bar that says 'INGRESO DE PETICIONES'. Below the title bar, there are three input fields. The first is labeled 'Petición:', the second 'Argumento:', and the third 'Respuestas:'. Below these fields are two buttons: 'Guardar' (Save) and 'Limpiar' (Clear).

Figura 6.19 Ingreso de peticiones

```
<%@ page contentType="text/html; charset=utf-8" language="java"
import="java.sql.*" errorPage="" %>
<jsp:useBean id="JSONRPCBridge" scope="session"
class="com.metaparadigm.jsonrpc.JSONRPCBridge" />
<jsp:useBean id="peticion" scope="session" class="Clases.Peticiones"
/>
<% JSONRPCBridge.registerObject("peticion", peticion); %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Ingreso Peticiones</title>
<link href="css/estilos.css" rel="stylesheet" type="text/css" />
<script src="scripts/jsonrpc.js" type="text/javascript"></script>
<script src="scripts/peticiones.js"
type="text/javascript"></script>
</head><body onload="cargando()">
<table width="633" border="0" align="center" cellpadding="1"
cellspacing="0" bordercolor="#CCCCCC" bgcolor="#CCCCCC">
<tr> <td class="mensajes"> INGRESO DE PETICIONES</td>
</tr></table>
<form id="form1" name="form1" method="post" action="">
<p>&nbsp;</p> <table width="455" border="0" align="center"> <tr> <td
width="103"><strong class="letras">Petición :</strong></td>
<td width="342"><input name="peticion" type="text"
class="caja_texto" id="peticion" /></td>
</tr> <tr> <td class="letras">Argumento :</td>
<td><input name="argumento" type="text" class="caja_texto"
id="argumento" /></td> </tr>
<tr> <td><span class="letras">Respuestas :</span></td>
<td><textarea name="respuesta" class="caja_texto"
id="respuesta"></textarea></td>
</tr> <tr> <td>&nbsp;</td> <td></td>
```

```

        <button value="Ingresar" class="botones" onclick="valida()"
type="button">  Guardar</button>
        <button value="Ingresar" class="botones" type="reset">  Limpiar</button>
    </td> </tr>
</table> <p>&nbsp;</p>
</form></body>
</html>

```

6.20 PÁGINA "ING_USUARIOS.JSP"

Esta pagina permite ingresar usuarios del sistema con todas las validaciones requeridas para una creación conforme a los requerimientos del sistema.



Figura 6.20 Ingreso de usuarios del sistema

```

<%@ page contentType="text/html; charset=utf-8" language="java"
import="java.sql.*" errorPage="" %>
<jsp:useBean id="JSONRPCBridge" scope="session"
class="com.metaparadigm.jsonrpc.JSONRPCBridge" />
<jsp:useBean id="usuarios" scope="session" class="Clases.Usuarios" />
<% JSONRPCBridge.registerObject("usuarios", usuarios); %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

```

```

<title>Ingresos de Usuarios</title>
    <link href="css/estilos.css" rel="stylesheet" type="text/css" />
    <script src="scripts/jsonrpc.js" type="text/javascript"></script>
    <script language="javascript" src="scripts/usuarios.js"></script>
</head>
<body onLoad="cargando()">
<table width="641" border="0" align="center" cellpadding="1"
cellspacing="0" bordercolor="#CCCCCC" bgcolor="#CCCCCC">
    <tr><td class="mensajes"> INGRESO DE USUARIOS DEL
SISTEMA</td>
    </tr></table>
<form id="form1" name="form1" method="post" action="usuarios.jsp">
    <table width="370" border="0" align="center">
        <tr><td width="120" class="letras">Nombres :</td>
        <td width="240"><input name="txt_nombres" type="text"
class="caja_texto" id="txt_nombres"maxlength="50" /></td>
        </tr> <tr> <td class="letras">Apellidos :</td>
        <td><input name="txt_apellidos" type="text" class="caja_texto"
id="txt_apellidos"maxlength="50" /></td>
        </tr> <tr> <td class="letras">Celular :</td>
        <td><input name="txt_celular" type="text" class="caja_texto"
id="txt_celular"maxlength="9" /></td>
        </tr> <tr> <td height="30" class="letras">Correo :</td>
        <td><input name="txt_correo" type="text" class="caja_texto"
id="txt_correo" maxlength="30" /></td>
        </tr> <tr>
        <td height="30" class="letras">Usuario :</td>
        <td><input name="txt_usuarios" type="text" class="caja_texto"
id="txt_usuario" maxlength="10" /></td>
        </tr> <tr>
        <td class="letras">Clave :</td>
        <td><input name="txt_clave" type="password" class="caja_texto"
id="txt_clave" maxlength="10" /></td>
        </tr> <tr> <td class="letras">Confirmar Clave :</td>
        <td><input name="txt_confirma" type="password" class="caja_texto"
id="txt_confirma" maxlength="10" /></td>
        </tr> <tr>
        <td class="letras">Privilegio :</td>
        <td>
            <select name="permiso" class="caja_texto" id="permiso">
                <%
                    Clases.Conexion cone= new Clases.Conexion();
                    cone.conectar();
                    java.sql.ResultSet listado=cone.Consultas("select * from permisos ");
                    while(listado.next()){
                        out.println("<option
value='"+listado.getString("codigo")+"'>"+listado.getString("nombre")+
"</option>");
                    }
                <%> </select> </td>
            </tr> <tr>
                <td> <button value="Ingresar"
class="botones" onclick="validar_usuarios()" type="button"> <img

```



```

src="IMAGENES/botones/Save As....png" width="23" height="22"
align="absmiddle" /> Guardar</button></td>
    <td> <button value="Ingresar" class="botones" type="reset">  Limpiar</button></td>
    </tr> </table>
    <p>&nbsp;</p></form>
</body>
</html>

```

6.21 PÁGINA "ING_USUARIOS_EXTERNOS.JSP"

Esta pagina permite ingresar usuarios externos con todas las validaciones requeridas para una creación conforme a los requerimientos del sistema; a este usuario se le asignará un dispositivo bluetooth el cual quedará registrado por el sistema.



Figura 6.21 Ingreso de usuarios externos

```

<%@ page contentType="text/html; charset=utf-8" language="java"
import="java.util.*" errorPage="" %>
<jsp:useBean id="JSONRPCBridge" scope="session"
class="com.metaparadigm.jsonrpc.JSONRPCBridge" />
<jsp:useBean id="externos" scope="session" class="Clases.Externos" />
<jsp:useBean id="consulta" scope="session" class="Clases.Dispositivos"
/>
<% JSONRPCBridge.registerObject("peticion", externos); %>

```

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Ingreso Usuarios Externos</title>
    <link href="css/estilos.css" rel="stylesheet" type="text/css" />
    <script src="scripts/jsonrpc.js" type="text/javascript"></script>
    <script src="scripts/externos.js" type="text/javascript"></script>
</head>
<body onload="cargando()">
<table width="641" border="0" align="center" cellpadding="1"
cellspacing="0" bordercolor="#CCCCCC" bgcolor="#CCCCCC">
    <tr> <td class="mensajes"> INGRESO DE USUARIOS EXTERNOS</td>
    </tr></table>
<form id="form1" name="form1" method="post" action="">
    <p>&nbsp;</p> <table width="455" border="0" align="center">
        <tr> <td width="171"><strong class="letras">Nombres :</strong></td>
            <td width="274"><input name="nombres" type="text"
class="caja_texto" id="nombres" maxlength="50" /></td>
        </tr> <tr> <td class="letras">Apellidos :</td>
            <td><input name="apellidos" type="text" class="caja_texto"
id="apellidos" maxlength="50" /></td>
        </tr> <tr> <td><span class="letras">Celular :</span></td>
            <td><input name="celular" type="text" class="caja_texto"
id="celular" maxlength="9" /></td>
        </tr> <tr> <td><span class="letras">Cedula :</span></td>
            <td><input name="cedula" type="text" class="caja_texto"
id="cedula" maxlength="10" /></td>
        </tr> <tr> <td><span class="letras">Dispositivo :</span></td>
            <td> <select name="dispositivos" class="caja_texto" id="dispositivos">
                <%
                    Vector consul=consulta.Consulta_dispositivos();
                    for(int i=0;i<consul.size();i++){
                        Vector info=(Vector)consul.elementAt(i);
                        out.println("<option
value='"+info.elementAt(1).toString()+"'>"+info.elementAt(0).toString()
+"</option>"); }
                <%>
            </select> </td> </tr>
        <tr> <td colspan="2"><div align="center">
            <button value="Ingresar" class="botones" onclick="valida()"
type="button">  Guardar</button>
            <button value="Ingresar" class="botones" type="reset">  Limpiar</button>

```

```

        <button value="Ingresar" class="botones" onclick="consulta()"
type="button"> Consultar</button>
    </div></td> </tr>
</table> <p>&nbsp;</p>
</form></body>
</html>

```

6.22 PÁGINA "MENU.JSP"

Esta ventana aparecerá siempre que hayan ingresado al sistema como un usuario validado y aparecerá el menú con todas las opciones posibles de acceso.



Figura 6.22 Menú

```

<%
    Clases.Usuarios objeto=new Clases.Usuarios();
    java.util.Vector
per=objeto.permisos_usuario((String)session.getAttribute("usuario"));
%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<script type="text/javascript" src="scripts/stmenu.js"></script>
<title>Untitled Document</title>
</head>
<body><script type="text/javascript">
<!--
stm_bm(["menu2fbb",900,"","blank.gif",0,"","",0,0,0,0,50,1,0,0,"","",0,
0,1,1,"default","hand","",1,25],this);
stm_bp("p0",[0,4,0,0,1,4,0,10,100,"",-2,"",-
2,90,0,0,"#FFCC00","transparent","",3,1,1,"#CCCCC"]);
stm_ai("p0i0",[0,"Principal","",",-1,-
1,0,"centro.jsp","trabajo","",",",0,0,0,"",0,0,0,0,1,"#FFFFFF",

```

```

0,"#FFFFFF",0,"",",",3,3,1,1,"#CCCCC", "#999999", "#000000", "#000000", "bo
ld 10pt Verdana", "bold 10pt Verdana", 0, 0));
stm_aix("p0i1", "p0i0", [0, "      Usuarios", "", "", -1, -
1, 0, "#", "_self", "", "", "", "", 0, 0, 0, "IMAGENES/arrow_r.gif", "IMAGENES/arro
w_r.gif", 10, 5, 0, 0, 1, "#333333", 0, "#666666", 0, "IMAGENES/usuario.gif", "IMA
GENES/usuario2.gif", 3, 3, 1, 1, "#CCCCC", "#999999", "#FFFFFF", "#FFFFFF", "bo
ld 10pt Verdana", "bold 10pt Verdana"], 120, 20);
stm_bp("p1", [1, 4, 0, 0, 1, 4, 0, 10, 100, "", -2, "", -
2, 58, 0, 0, "#FFCC00", "transparent", "", 3, 0, 0, "#000000"]);
stm_aix("pli0", "p0i0", [0, "Sistema", "", "", -1, -
1, 0, "#", "_self", "", "", "", "", 0, 0, 0, "060420arrow2.gif", "060420arrow2.gif",
10, 5, 0, 0, 1, "#666666", 0, "#000000", 0, "", "", 3, 3, 1, 1, "#CCCCC", "#CCCCC", "
#FFFFFF", "#FFFFFF", "10pt Verdana", "10pt Verdana"], 0, 20);
stm_bpx("p2", "p1", [1, 2, 0, 0, 1, 4, 0, 0]);
stm_aix("p2i0", "pli0", [0, "Ingresar", "", "", -1, -1, 0, "<%
if(per.get(0).toString().compareTo("1")==0){ %>ing_usuarios.jsp<% }
%>", "trabajo", "", "", "", "", 0, 0, 0, "", "", 0, 0, 0, 0, 1, "#333333", 0, "#666666", 0
, "", ""]);
stm_aix("p2i1", "p2i0", [0, "Consultar", "", "", -1, -1, 0, "<%
if(per.get(1).toString().compareTo("1")==0){ %>con_usuarios.jsp<% }
%>", "trabajo", "", "", "", "", 0, 0, 0, "", "", 0, 0, 0, 0, 1, "#333333", 0, "#666666", 0
, "", ""]);stm_ep();
stm_aix("pli1", "pli0", [0, "Externos"], 0, 20);stm_bpx("p3", "p2", []);
stm_aix("p3i0", "p2i0", [0, "Ingresar", "", "", -1, -1, 0, "<%
if(per.get(2).toString().compareTo("1")==0){
%>ing_usuarios_externos.jsp<% }
%>", "trabajo", "", "", "", "", 0, 0, 0, "", "", 0, 0, 0, 0, 1, "#333333", 0, "#666666", 0
, "", ""]);
stm_aix("p3i1", "p2i1", [0, "Consultar", "", "", -1, -1, 0, "<%
if(per.get(3).toString().compareTo("1")==0){ %>rep_externos.jsp<% }
%>", "trabajo", "", "", "", "", 0, 0, 0, "", "", 0, 0, 0, 0, 1, "#333333", 0, "#666666", 0
, "", ""]);stm_ep();stm_ep();
stm_aix("p0i2", "p0i1", [0, "      Privilegios", "", "", -1, -
1, 0, "#", "_self", "", "", "", "", 0, 0, 0, "IMAGENES/arrow_r.gif", "IMAGENES/arro
w_r.gif", 10, 5, 0, 0, 1, "#333333", 0, "#666666", 0, "IMAGENES/NUEVO1.gif", "IMAG
ENES/NUEVO1.2.gif", 3, 3, 1, 1, "#CCCCC", "#999999", "#FFFFFF", "#FFFFFF", "bol
d 10pt Verdana", "bold 10pt Verdana"], 120, 20);
stm_bpx("p4", "p2", [1, 4]);
stm_aix("p4i0", "pli0", [0, "Ingresar", "", "", -1, -1, 0, "<%
if(per.get(4).toString().compareTo("1")==0){ %>ing_privilegios.jsp<% }
%>", "trabajo", "", "", "", "", 0, 0, 0, "", "", 0, 0, 0, 0, 1, "#333333", 0, "#666666", 0
, "", ""]);
stm_aix("p4i1", "p4i0", [0, "Consultar", "", "", -1, -1, 0, "<%
if(per.get(5).toString().compareTo("1")==0){ %>con_privilegios.jsp<% }
%>", "trabajo", "", "", "", "", 0, 0, 0, "", "", 0, 0, 0, 0, 1, "#333333", 0, "#666666", 0
, "", ""]);stm_ep();
stm_aix("p0i3", "p0i1", [0, "      Envios", "", "", -1, -
1, 0, "#", "_self", "", "", "", "", 0, 0, 0, "IMAGENES/arrow_r.gif", "IMAGENES/arro
w_r.gif", 10, 5, 0, 0, 1, "#333333", 0, "#666666", 0, "IMAGENES/dispositivos.gif"

```

```
, "IMAGENES/dispositivos2.gif", 3, 3, 1, 1, "#CCCCC", "#999999", "#FFFFFF", "#F
FFFFF", "bold 10pt Verdana", "bold 10pt Verdana", 120, 20);
stm_bpx("p5", "p4", []);
stm_aix("p5i0", "p4i0", [0, "Mensajes", "", "", -1, -1, 0, "<%
if(per.get(6).toString().compareTo("1")==0){ %>env_dispositivos.jsp<%
}
%>", "trabajo", "", "", "", "", 0, 0, 0, "", "", 0, 0, 0, 0, 1, "#333333", 0, "#666666", 0
, "", ""]);
stm_aix("p5i1", "p5i0", [0, "Publicidad", "", "", -1, -1, 0, "<%
if(per.get(7).toString().compareTo("1")==0){ %>env_publicidad.jsp<% }
%>", "trabajo", "", "", "", "", 0, 0, 0, "", "", 0, 0, 0, 0, 1, "#333333", 0, "#666666", 0
, "", ""]); stm_ep();
stm_aix("p0i4", "p0i1", [0, "Reportes", "", "", -1, -
1, 0, "", "_self", "", "", "", "", 0, 0, 0, "IMAGENES/arrow_r.gif", "IMAGENES/arrow
_r.gif", 7, 7, 0, 0, 1, "#333333", 0, "#666666", 0, "IMAGENES/academico.gif", "IMA
GENES/academico2.gif", 0, 20); stm_bpx("p6", "p1", []);
stm_aix("p6i1", "p5i0", [0, "Dispositivos Encontrados", "", "", -1, -1, 0, "<%
if(per.get(8).toString().compareTo("1")==0){ %>rep_conectados.jsp<% }
%>", "trabajo", "", "", "", "", 0, 0, 0, "", "", 0, 0, 0, 0, 1, "#333333", 0, "#666666", 0
, "", ""]); stm_aix("p6i2", "p1i0", [0, "Buzones
", "", "", -1, -1, 0, 120, 0); stm_bpx("p7", "p2", []);
stm_aix("p7i0", "p2i0", [0, "Entrada", "", "", 0, -1, 0, "<%
if(per.get(11).toString().compareTo("1")==0){ %>entrada.jsp<% }
%>", "trabajo", "", "", "", "", 0, 0, 0, "", "", 0, 0, 0, 0, 1, "#333333", 0, "#666666", 0
, "", ""]);
stm_aix("p7i1", "p2i0", [0, "Salida", "", "", 0, -1, 0, "<%
if(per.get(12).toString().compareTo("1")==0){ %>salida.jsp<% }
%>", "trabajo", "", "", "", "", 0, 0, 0, "", "", 0, 0, 0, 0, 1, "#333333", 0, "#666666", 0
, "", ""]);
stm_aix("p7i2", "p2i0", [0, "No Recibidos", "", "", 0, -1, 0, "<%
if(per.get(13).toString().compareTo("1")==0){ %>no_enviados.jsp<% }
%>", "trabajo", "", "", "", "", 0, 0, 0, "", "", 0, 0, 0, 0, 1, "#333333", 0, "#666666", 0
, "", ""]);
stm_aix("p7i3", "p2i0", [0, "Borradores", "", "", 0, -1, 0, "<%
if(per.get(14).toString().compareTo("1")==0){ %>borradores.jsp<% }
%>", "trabajo", "", "", "", "", 0, 0, 0, "", "", 0, 0, 0, 0, 1, "#333333", 0, "#666666", 0
, "", ""]);
stm_aix("p7i4", "p2i0", [0, "Contador", "", "", 0, -1, 0, "<%
if(per.get(15).toString().compareTo("1")==0){ %>contador.jsp<% }
%>", "trabajo", "", "", "", "", 0, 0, 0, "", "", 0, 0, 0, 0, 1, "#333333", 0, "#666666", 0
, "", ""]);
stm_ep(); stm_aix("p6i3", "p5i0", [0, "Auditoria", "", "", -1, -1, 0, "<% if(per.get(9).toString().compareTo("1")==0){
%>rep_acceso.jsp<% }
%>", "trabajo", "", "", "", "", 0, 0, 0, "", "", 0, 0, 0, 0, 1, "#333333", 0, "#666666", 0
, "", ""]); stm_ep(); stm_aix("p0i5", "p0i4", [0, "Peticiones", "", "", -1, -
1, 0, "", "_self", "", "", "", "", 0, 0, 0, "IMAGENES/arrow_r.gif", "IMAGENES/arrow
```

```

_r.gif",7,7,0,0,1,"#333333",0,"#666666",0,"IMAGENES/eventos.gif","IMAGE
NES/eventos2.gif"],0,20);stm_bpx("p8","p4",[]);
stm_aix("p8i0","p5i0",[0,"Ingresar","",-1,-1,0,"<%
if(per.get(16).toString().compareTo("1")==0){ %>ing_programacion.jsp<%
}
%>","trabajo","","","","",0,0,0,"",-1,-1,0,1,"#333333",0,"#666666",0
,"",""]); stm_aix("p8i1","p5i0",[0,"Consultar","",-1,-1,0,"<%
if(per.get(17).toString().compareTo("1")==0){ %>con_programacion.jsp<%
}
%>","trabajo","","","","",0,0,0,"",-1,-1,0,1,"#333333",0,"#666666",0
,"",""]);stm_ep();stm_ep();stm_em();
//-->
</script>
</body>
</html>

```

6.23 PÁGINA "MOD_BORRADORES.JSP"

Esta ventana realizará las modificaciones de borradores creador en la aplicación.



Figura 6.23 Modificación de borradores

```

<%
String id=request.getParameter("id");
String titulo="";String contenido="";
Clases.Conexion cone= new Clases.Conexion();
cone.conectar(); java.sql.ResultSet listado=cone.Consultas("select *
from borrados where id="+id);
while(listado.next()){

```

```

        titulo=listado.getString("titulo");
        contenido=listado.getString("contenido");
    }   cone.desconectar();
%>
<%@page contentType="text/html" pageEncoding="UTF-8"
import="java.util.*" %>
<jsp:useBean id="consulta" scope="session" class="Clases.Dispositivos"
/><jsp:useBean id="JSONRPCBridge" scope="session"
class="com.metaparadigm.jsonrpc.JSONRPCBridge" />
<jsp:useBean id="mantenimiento" scope="session"
class="Clases.Borradores" />
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<% JSONRPCBridge.registerObject("mantenimiento", mantenimiento); %>
<html>
    <head>    <script src="scripts/jsonrpc.js"
type="text/javascript"></script>    <script type="text/javascript"
src="scripts/prototype.js" language="javascript"></script>
        <script src="scripts/buzones.js"
type="text/javascript"></script>    <script src="scripts/borradores.js"
type="text/javascript"></script>
        <link href="css/estilos.css" rel="stylesheet" type="text/css"
/></head> <body onload="cargando()">
        <form id="form1" name="form1" method="post" action="">
<table width="700" border="0" align="center" cellpadding="1"
cellspacing="0" bordercolor="#CCCCCC" bgcolor="#CCCCCC">
<tr> <td class="mensajes"> MODIFICACION DE
BORRADORES</td> </tr> </table>
        <table width="200" border="1" align="center" cellpadding="1"
cellspacing="0" bordercolor="#000000">
        <tr><td><table width="691" border="0" align="center"
cellpadding="0" cellspacing="0">
                <tr> <td width="132">&nbsp;</td>
                <td width="118">&nbsp;</td> <td width="136">&nbsp;</td>
                <td width="164">&nbsp;</td> <td width="141">&nbsp;</td>
            </tr> <tr> <td><strong>Titulo :</strong></td>
                <td colspan="2"><input name="titulo" type="text"
class="caja_texto" id="titulo" size="33" value="<%=titulo%>"></td>
                <td><input type="hidden" name="codigo" id="codigo"
value="<%=id%>"></td> <td>&nbsp;</td>
            </tr> <tr> <td><strong>Contenido :</strong></td>
                <td colspan="2"><textarea name="contenido" cols="30"
class="caja_texto" id="contenido"> <%=contenido%></textarea></td>
                <td>&nbsp;</td> <td>&nbsp;</td>
            </tr> <tr> <td>&nbsp;</td> <td>&nbsp;</td>
            </tr> <tr> <td><button value="Ingresar" class="botones"
onclick="modificar_datos()" type="button"> <img

```



```

<%@ page contentType="text/html; charset=utf-8" language="java"
import="java.util.*" errorPage="" %>
<jsp:useBean id="JSONRPCBridge" scope="session"
class="com.metaparadigm.jsonrpc.JSONRPCBridge" />
<jsp:useBean id="externos" scope="session" class="Clases.Externos" />
<jsp:useBean id="consulta" scope="session" class="Clases.Dispositivos"
/> <% JSONRPCBridge.registerObject("peticion", externos); %>
<%String id=request.getParameter("id");
String nombres="";String apellidos="";String cedula="";
String dispositivo="";String celular="";
Clases.Conexion cone= new Clases.Conexion();
    cone.conectar();
java.sql.ResultSet listado=cone.Consultas("select * from
usuarios_externos where id="+id);
    while(listado.next()){ nombres=listado.getString("nombres");
        apellidos=listado.getString("apellidos");
cedula=listado.getString("cedula");
        dispositivo=listado.getString("dispositivo");
        celular=listado.getString("celular"); }    cone.desconectar();%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head><meta http-equiv="Content-Type" content="text/html; charset=utf-
8" /> <title>Ingreso Usuarios Externos</title>
    <link href="css/estilos.css" rel="stylesheet" type="text/css" />
        <script src="scripts/jsonrpc.js" type="text/javascript"></script>
        <script src="scripts/externos.js" type="text/javascript"></script>
</head><body onload="cargando()">
<table width="641" border="0" align="center" cellpadding="1"
cellspacing="0" bordercolor="#CCCCCC" bgcolor="#CCCCCC">
    <tr>    <td class="mensajes"> MODIFICAR USUARIOS EXTERNOS</td>    </tr>
</table><form id="form1" name="form1" method="post" action="">
    <p>&nbsp;</p>    <table width="455" border="0" align="center">
        <tr>    <td width="103"><strong class="letras">Nombres
:</strong></td>    <td width="342"><input name="nombres" type="text"
class="caja_texto" id="nombres" value="<%=nombres%>" maxlength="50" />
        <input type="hidden" name="id" id="id" value="<%=id%>" /></td>
        </tr>    <tr>    <td class="letras">Apellidos :</td>
        <td><input name="apellidos" type="text" class="caja_texto"
id="apellidos" value="<%=apellidos%>" maxlength="50"/></td>
        </tr>    <tr>    <td><span class="letras">Celular :</span></td>
        <td><input name="celular" type="text" class="caja_texto"
id="celular" value="<%=celular%>" maxlength="9"/></td>
        </tr>    <tr>    <td><span class="letras">Cedula :</span></td>
        <td><input name="cedula" type="text" class="caja_texto"
id="cedula" value="<%=cedula%>" maxlength="10"/></td>

```

```

</tr> <tr> <td><span class="letras">Dispositivo :</span></td>
<td> <select name="dispositivos" class="caja_texto"
id="dispositivos"> <%Vector consul=consulta.Consulta_dispositivos();
for(int i=0;i<consul.size();i++){
Vector info=(Vector)consul.elementAt(i);
if(dispositivo.compareTo(info.elementAt(1).toString())==0){
out.println("<option value='"+info.elementAt(1).toString()+"'
selected='selected'>"+info.elementAt(0).toString()+"</option>");
}else{ out.println("<option
value='"+info.elementAt(1).toString()+"'>"+info.elementAt(0).toString()
+"</option>");}} %> </select> </td> </tr> <tr> <td>&nbsp;</td> <td>
<button value="Ingresar" class="botones"
onclick="actualiza_usuarios()" type="button"> Modificar</button> <button value="Ingresar"
class="botones" onclick="consulta()" type="button"> Regresar</button>
</td> </tr> </table> <p>&nbsp;</p> </form></body>
</html>

```

6.25 PÁGINA "MOD_PRIVILEGIOS.JSP"

Esta pagina permitirá modificar el privilegio a seleccionar o deseleccionar las opciones por cada menú que aparece.



Figura 6.25 Modificación de privilegios

```

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<jsp:useBean id="JSONRPCBridge" scope="session"
class="com.metaparadigm.jsonrpc.JSONRPCBridge" />
<jsp:useBean id="permisos" scope="session" class="Clases.Permisos" />
<% JSONRPCBridge.registerObject("permisos", permisos); %>
<%
String numero=request.getParameter("permiso");
String op1="";String op2="";String op3="";String op4="";
String op5="";String op6="";String op7="";String op8="";
String op9="";String op10="";String op11="";String op1
2="";String op13="";String op14="";String op15="";
String op16="";String op17="";String op18="";
String nombre="";
Clases.Conexion cone= new Clases.Conexion();
cone.conectar();
java.sql.ResultSet listado=cone.Consultas("select * from permisos where
codigo="+numero); while(listado.next()){
    nombre=listado.getString("nombre");
    if(1==listado.getInt("usua_ing")){ op1="checked='checked' "; }
    if(1==listado.getInt("usua_con")){ op2="checked='checked' "; }
    if(1==listado.getInt("usua_inge")){ op3="checked='checked' "; }
    if(1==listado.getInt("usua_cone")){ op4="checked='checked' "; }
    if(1==listado.getInt("pri_ing")){ op5="checked='checked' "; }
    if(1==listado.getInt("pri_con")){ op6="checked='checked' "; }
    if(1==listado.getInt("dis_con")){ op7="checked='checked' "; }
    if(1==listado.getInt("dis_env")){ op8="checked='checked' "; }
    if(1==listado.getInt("rep_1")){ op9="checked='checked' "; }
    if(1==listado.getInt("rep_2")){ op10="checked='checked' "; }
    if(1==listado.getInt("rep_3")){ op11="checked='checked' "; }
    if(1==listado.getInt("rep_4")){ op12="checked='checked' "; }
    if(1==listado.getInt("rep_5")){ op13="checked='checked' "; }
    if(1==listado.getInt("rep_6")){ op14="checked='checked' "; }
    if(1==listado.getInt("rep_7")){ op15="checked='checked' "; }
    if(1==listado.getInt("rep_8")){ op16="checked='checked' "; }
    if(1==listado.getInt("pro_ing")){ op17="checked='checked' "; }
    if(1==listado.getInt("pro_con")){ op18="checked='checked' "; }
} cone.desconectar(); %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html> <head> <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
    <title>Modificar Privilegios</title>
    <link href="css/estilos.css" rel="stylesheet" type="text/css"
/> <!--[if lt IE 7]>
<script defer type="text/javascript" src="scripts/pngfix.js"></script>
<![endif]-->
    <script src="scripts/formexp.js"></script>
    <script src="scripts/permisos.js"></script>

```

```

<script src="scripts/jsonrpc.js"
type="text/javascript"></script>
    <script> function desactiva(){
        xDisplay('capaexpansion2', 'none')
        xDisplay('capaexpansion3', 'none')
        xDisplay('capaexpansion4', 'none')
        xDisplay('capaexpansion5', 'none')
    } function ver_opcion(){
        if(document.form1.tipo.value==1){
            xDisplay('capaexpansion1', 'block')
            xDisplay('capaexpansion2', 'none')
            xDisplay('capaexpansion3', 'none')
            xDisplay('capaexpansion4', 'none')
            xDisplay('capaexpansion5', 'none')
        } if(document.form1.tipo.value==2){
            xDisplay('capaexpansion2', 'block')
            xDisplay('capaexpansion1', 'none')
            xDisplay('capaexpansion3', 'none')
            xDisplay('capaexpansion4', 'none')
            xDisplay('capaexpansion5', 'none')
        } if(document.form1.tipo.value==3){
            xDisplay('capaexpansion2', 'none')
            xDisplay('capaexpansion1', 'none')
            xDisplay('capaexpansion3', 'block')
            xDisplay('capaexpansion4', 'none')
            xDisplay('capaexpansion5', 'none')
        } if(document.form1.tipo.value==4){
            xDisplay('capaexpansion2', 'none')
            xDisplay('capaexpansion1', 'none')
            xDisplay('capaexpansion3', 'none')
            xDisplay('capaexpansion4', 'block')
            xDisplay('capaexpansion5', 'none')
        } if(document.form1.tipo.value==5){
            xDisplay('capaexpansion5', 'block')
            xDisplay('capaexpansion4', 'none')
            xDisplay('capaexpansion1', 'none')
            xDisplay('capaexpansion3', 'none')
            xDisplay('capaexpansion2', 'none')}
    }
</script>
</head>    <body onLoad="cargando()">
    <table width="523" height="391" border="0" align="center">
        <tr> <td><form name="form1" method="post"
action="ing_privilegios.jsp">
<table width="641" border="0" align="center" cellpadding="1"
cellspacing="0" bordercolor="#CCCCCC" bgcolor="#CCCCCC"> <tr>
    <td class="mensajes"> MODIFICAR DE PRIVILEGIOS DE
USUARIOS</td> </tr>
</table><table width="615" border="0" align="center">

```



```

        </tr><tr> <td>&nbsp;</td>
        </tr> </table><p>&nbsp;</p>
    </div> <div id=capaexpansion2>
        <table width="488" border="0" align="center" cellpadding="1"
cellspacing="0"> <tr> <td><input name="p5" type="checkbox" id="p5"
value="1" <% out.print(op5); %>>
Ingresos Privilegios</td>
        <td>&nbsp;</td> </tr>
        <tr> <td><input name="p6" type="checkbox" id="p6"
value="1" <% out.print(op6); %>>
Consultas Privilegios</td> <td>&nbsp;</td>
        </tr></table>
        <table width="200" border="0" cellspacing="0" cellpadding="1">
            <tr><td>&nbsp;</td>
            </tr> <tr> <td>&nbsp;</td>
            </tr><tr><td>&nbsp;</td>
            </tr> <tr><td>&nbsp;</td>
            </tr> <tr><td>&nbsp;</td>
            </tr><tr><td>&nbsp;</td>
            </tr> </table><p>&nbsp;</p>
    </div> <div id=capaexpansion3>
<table width="491" border="0" align="center" cellpadding="1"
cellspacing="0">
    <tr> <td><input name="p7" type="checkbox" id="p7" value="1" <%
out.print(op7); %>>
        Envios de Mensajes </td>
        <td>&nbsp;</td> </tr>
        <tr> <td><input name="p8" type="checkbox" id="p8"
value="1" <% out.print(op8); %>>
        Envios de Publicidad</td>
        <td>&nbsp;</td> </tr> </table> <table width="200" border="0"
cellspacing="0" cellpadding="1">
            <tr> <td>&nbsp;</td> </tr>
            <tr> <td>&nbsp;</td> </tr>
            <tr> <td>&nbsp;</td> </tr>
            <tr> <td>&nbsp;</td> </tr>
            <tr> <td>&nbsp;</td> </tr>
            <tr> <td>&nbsp;</td> </tr>
            </table> <p>&nbsp;</p></div>
    <div id=capaexpansion4>
        <table width="491" border="0" align="center" cellpadding="1"
cellspacing="0"><tr>
    <td><input name="p9" type="checkbox" id="p9" value="1" <%
out.print(op9); %>>Dispositivos Encontrados</td>
        <td>&nbsp;</td></tr>
        <tr> <td><input name="p10" type="checkbox" id="p10"
value="1" <% out.print(op10); %>>

```

```

        Auditoria</td> <td>&nbsp;</td>
    </tr> <tr> <td><input name="p12" type="checkbox"
id="p12" value="1" <% out.print(op12); %>>
        Buzon de Entrada</td> <td>&nbsp;</td>
    </tr> <tr>
        <td><input name="p13" type="checkbox" id="p13"
value="1" <% out.print(op13); %>>
        Buzon de Salida</td><td>&nbsp;</td>
    </tr> <tr>
        <td><input name="p14" type="checkbox" id="p14"
value="1" <% out.print(op14); %>>
        Buzon de No Recibidos</td> <td>&nbsp;</td>
    </tr> <tr> <td><input name="p15" type="checkbox"
id="p15" value="1" <% out.print(op15); %>>
        Borradores</td> <td>&nbsp;</td>
    </tr> <tr> <td><input name="p16" type="checkbox"
id="p16" value="1" <% out.print(op16); %>>
        Contador</td> <td>&nbsp;</td> </tr>
</table><table width="200" border="0" cellpadding="0"
cellpadding="1"> <tr> <td>&nbsp;</td>
    </tr> <tr> <td>&nbsp;</td>
    </tr> </table>
</div> <div id="capaexpansion5">
    <table width="486" border="0" align="center"
cellpadding="1" cellspacing="0">
        <tr><td><input name="p17" type="checkbox" id="p17"
value="1" <% out.print(op17); %>>
            Ingreso de Petici&oacute;n</td><td>&nbsp;</td> </tr>
        <tr> <td><input name="p18" type="checkbox" id="p18"
value="1" <% out.print(op18); %>>
            Consulta de Petici&oacute;n</td>
            <td>&nbsp;</td> </tr> </table>
    <table width="200" border="0" cellspacing="0" cellpadding="1">
        <tr> <td>&nbsp;</td> </tr> <tr> <td>&nbsp;</td> </tr>
        <tr> <td>&nbsp;</td> </tr><tr> <td>&nbsp;</td> </tr>
        <tr> <td>&nbsp;</td> </tr> </table>
        <p>&nbsp;</p> </div> </td> </tr> </table>
</form> </td> </tr>
</table><script>
    desactiva()
</script>
</body></html>

```

6.26 PÁGINA "MOD_PUBLICIDAD.JSP"

Esta pagina permitirá modificar los días que la publicidad se desee enviar. Y a la vez se podra poner en el estado si lo quiere tener activo o desea desactivar la publicidad.

Figura 6.26 Modificación de publicidad

```
<%@ page import="java.sql.*"%>
<%
String nombre = request.getParameter("nombre");
String codigo = request.getParameter("codigo");
String d1 = request.getParameter("D1");
if(d1==null){
d1="0"; }else{ d1="1"; }
String d2 = request.getParameter("D2");
if(d2==null){
d2="0"; }else{ d2="1"; }
String d3 = request.getParameter("D3");
if(d3==null){
d3="0"; }else{ d3="1"; }
String d4 = request.getParameter("D4");
if(d4==null){
d4="0"; }else{ d4="1"; }
String d5 = request.getParameter("D5");
if(d5==null){
d5="0"; }else{ d5="1"; }
String d6 = request.getParameter("D6");
```



```

if(d6==null){
d6="0"; }else{ d6="1"; }
String d7 = request.getParameter("D7");
if(d7==null){
d7="0"; }else{ d7="1"; }
String estado= request.getParameter("estado");
Clases.Publicidad objeto=new Clases.Publicidad();
objeto.modificar_publicidad(nombre,"", Integer.parseInt(d1),
Integer.parseInt(d2), Integer.parseInt(d3), Integer.parseInt(d4),
Integer.parseInt(d5), Integer.parseInt(d6), Integer.parseInt(d7),
Integer.parseInt(estado),Integer.parseInt(codigo));
out.println("<script>alert('La publicidad fue modifica con
exitito...')</script>");
out.println("<script>window.location='env_publicidad.jsp';</script>");
%>

```

6.27 PÁGINA "NO_ENVIADOS.JSP"

Esta ventada permitirá obtener una busqueda por opciones de visualización de los mensjaes que no fueron enviador al dispositivo bluetoth.



Dispositivo	Nombre	Mensaje	Fecha
001AD08FC033	Federico	hola	2009-06-05 21:50:47.0
00180F846778	Nelson 6131	SEGUNDO CONTENIDO	2009-06-05 21:57:06.0
001D6E3EA947	Julian	SEGUNDO CONTENIDO	2009-06-05 21:51:05.0
001D6E3EA947	Julian	SEGUNDO CONTENIDO	2009-06-05 21:51:13.0
00219E74F015	Gloria love	PRIMER CONTENIDO	2009-06-05 21:52:48.0

Figura 6.27 Buzón de mensajes no enviados

```

<%@page contentType="text/html" pageEncoding="UTF-8"
import="java.util.*" %>
<jsp:useBean id="consulta" scope="session" class="Clases.Dispositivos"
/><jsp:useBean id="fecha" scope="session" class="Clases.Fechas" />
<script src="scripts/formexp.js"
type="text/javascript"></script>

```

```

        <script type="text/javascript" src="scripts/prototype.js"
language="javascript"></script>
        <script type="text/javascript" src="popcalendar.js"
language="javascript"></script>
        <script src="scripts/buzones.js"
type="text/javascript"></script>
        <script type="text/javascript" src="scripts/dispositivos.js"
language="javascript"></script>
        <link href="css/estilos.css" rel="stylesheet" type="text/css"
/> <body>
        <form name="form1" method="post" action="salida.jsp">
            <table width="658" border="0" align="center" cellpadding="1"
cellspacing="0" bordercolor="#CCCCCC" bgcolor="#CCCCCC">
                <tr> <td class="mensajes"> BUZON DE NO ENVIADOS</td>
                </tr> </table>
                <table width="643" border="0" align="center">
                    <tr> <td width="107" class="estilos">Reporte en : </td> <td
width="537"><select name="operacion" class="caja_texto" id="operacion">
                        <option value="0">Pdf</option>
                        <option value="1">Excel</option>
                    </select> <button value="Ingresar" class="botones"
onclick="exportar_reporte()" type="button">  Exportar</button>
                        <button value="Ingresar" class="botones"
onclick="imprimir()" type="button">  Imprimir</button>
                        <button value="Ingresar" class="botones" onclick="buscar()"
type="button">  Buscar</button>
                        <input name="reporte" type="hidden" id="reporte"
value="buzones.jasper" />
                        <input name="tipo" type="hidden" id="tipo" value="2" />
                        <input name="repo" type="hidden" id="repo" value="NO ENVIADO" />
                        <input name="fecha_hoy" type="hidden" id="fecha_hoy"
value="<%=fecha.f1%>" />
                        <input name="fecha_semana" type="hidden" id="fecha_semana"
value="<%=fecha.f2%>" />
                        <input name="fecha_quincena" type="hidden"
id="fecha_quincena" value="<%=fecha.f3%>" />
                        <input name="fecha_mes" type="hidden" id="fecha_mes"
value="<%=fecha.f4%>" />
                    </tr> </table>
                <td colspan="2" class="estilos"> <table width="644" border="0">
                    <tr> <td width="105">Visualizaci&#243;n : </td>
                    <td width="529"><select name="reportes" id="reportes"
onChange="ver_reporte()" />

```

```

        <option value="1" selected>DIARIO</option>
        <option value="2">SEMANA</option>
        <option value="3">QUINCENAL</option>
        <option value="4">MENSUAL</option>
        <option value="5">PERSONALIZADO</option>
    </select>    </td>
</tr></table>    <div id=capaexpansion><table width="648" border="0"
cellspacing="0" cellpadding="1">
    <tr>    <td width="106">Fecha desde :</td>
    <td width="184"><input name="f1" type="text" id="f1"
readonly="true" value="">
        </td>
    <td width="103">Fecha hasta :</td>
    <td width="170"><input name="f2" type="text" id="f2"
readonly="true" value="">
        </td>
    <td width="64">&nbsp;</td> </tr>
</table> </div>
</td> </tr> </table>
<div id="consulta_divd">
    <table width="663" border="1" align="center" cellpadding="1"
cellspacing="0" bordercolor="#000000">
    <tr> <td width="129" bgcolor="#CCCCCC"><div
align="center"><strong>Dispositivo</strong></div></td>
    <td width="129" bgcolor="#CCCCCC"><div
align="center"><strong>Nombre</strong></div></td>
    <td width="150" bgcolor="#CCCCCC"><div
align="center"><strong>Mensaje</strong></div></td>
    <td width="177" bgcolor="#CCCCCC"><div
align="center"><strong>Fecha </strong></div></td>
    </tr> </table>
</div><script>oculta_div();
</script> </form>
</body>

```

6.28 PÁGINA "NUEVA_PUBLICIDAD.JSP"

Esta ventana realizará el ingreso de nuevas publicidades y escoger entre los días de la semana que se deseará seleccionar. También se eligirá en el estado que se desea crear.

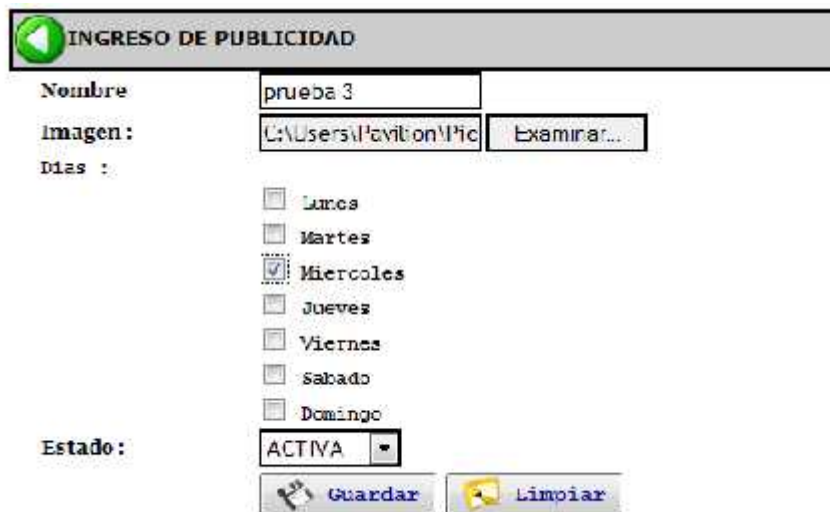


Figura 6.28 Ingreso de publicidad

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html> <head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <title>JSP Page</title>
  <link href="css/estilos.css" rel="stylesheet" type="text/css" />
  <script src="scripts/jsonrpc.js"
type="text/javascript"></script>
  <script language="javascript" src="scripts/publicidad.js"></script>
</head>
<body>
  <table width="506" border="0" align="center" cellpadding="1"
cellspacing="0" bordercolor="#CCCCCC" bgcolor="#CCCCCC">
    <tr> <td class="mensajes">INGRESO DE PUBLICIDAD </td>
    </tr> </table>
  <form action="guarda_publicidad.jsp" method="post"
enctype="multipart/form-data" name="form1">
    <table width="469" border="0" align="center" cellpadding="1"
cellspacing="0"> <tr>
      <td width="130" class="letras">Nombre </td>
      <td width="335"><input name="nombre" type="text"
class="caja_texto" id="nombre"></td>
    </tr> <tr> <td class="letras">Imagen :</td>
      <td><input name="archivo" type="file" class="caja_texto"
id="archivo"></td> </tr><tr> <td class="estilos">Dias :</td>
      <td class="estilos">&nbsp;</td> </tr>
      <tr> <td class="estilos">&nbsp;</td>
      <td class="estilos"><input type="checkbox" name="D1" id="D1">
```

```

        Lunes</td> </tr>
        <tr><td class="estilos">&nbsp;</td>
        <td class="estilos"><input type="checkbox" name="D2" id="D2">
        Martes</td></tr>
        <tr> <td class="estilos">&nbsp;</td>
        <td class="estilos"><input type="checkbox" name="D3" id="D3">
        Miercoles</td></tr>
        <tr> <td class="estilos">&nbsp;</td>
        <td class="estilos"><input type="checkbox" name="D4" id="D4">
        Jueves</td></tr>
        <tr> <td class="estilos">&nbsp;</td>
        <td class="estilos"><input type="checkbox" name="D5" id="D5">
        Viernes</td></tr>
        <tr><td class="estilos">&nbsp;</td>
        <td class="estilos"><input type="checkbox" name="D6" id="D6">
        Sabado</td></tr><tr>
        <td class="estilos">&nbsp;</td>
        <td class="estilos"><input type="checkbox" name="D7" id="D7">
        Domingo</td></tr>
        <tr> <td class="letras">Estado :</td>
        <td><select name="estado" class="caja_texto" id="estado">
        <option value="1">INACTIVA</option>
        <option value="2">ACTIVA</option>
        </select>
        </td></tr>
        <tr><td>&nbsp;</td><td>
        <button value="Ingresar" class="botones"
onclick="ingreso()" type="button">  Guardar</button>
        <button value="Ingresar" class="botones" type="reset">
 Limpiar</button></td>
        </tr> </table> </form></body>
</html>

```

6.29 PÁGINA "REP_ACCESO.JSP"

Esta ventana permite al usuario auditor ver los accesos cuando ingreso o salieron los usuarios del sistema.

Nombre	Apellidos	Usuario	Fecha	Acción
JULIO CESAR	CHÓEZ CATUTO	admin	201001-010-001	INGRESO AL SISTEMA
JULIO CESAR	CHÓEZ CATUTO	admin	201001-010-002	INGRESO AL SISTEMA
JULIO CESAR	CHÓEZ CATUTO	admin	201001-010-003	INGRESO AL SISTEMA
JULIO CESAR	CHÓEZ CATUTO	admin	201001-010-004	INGRESO AL SISTEMA
JULIO CESAR	CHÓEZ CATUTO	admin	201001-010-005	INGRESO AL SISTEMA
JULIO CESAR	CHÓEZ CATUTO	admin	201001-010-006	INGRESO AL SISTEMA
JULIO CESAR	CHÓEZ CATUTO	admin	201001-010-007	INGRESO AL SISTEMA
JULIO CESAR	CHÓEZ CATUTO	admin	201001-010-008	INGRESO AL SISTEMA
JULIO CESAR	CHÓEZ CATUTO	admin	201001-010-009	INGRESO AL SISTEMA
JULIO CESAR	CHÓEZ CATUTO	admin	201001-010-010	INGRESO AL SISTEMA
JULIO CESAR	CHÓEZ CATUTO	admin	201001-010-011	INGRESO AL SISTEMA
JULIO CESAR	CHÓEZ CATUTO	admin	201001-010-012	INGRESO AL SISTEMA
JULIO CESAR	CHÓEZ CATUTO	admin	201001-010-013	INGRESO AL SISTEMA
JULIO CESAR	CHÓEZ CATUTO	admin	201001-010-014	INGRESO AL SISTEMA
JULIO CESAR	CHÓEZ CATUTO	admin	201001-010-015	INGRESO AL SISTEMA
JULIO CESAR	CHÓEZ CATUTO	admin	201001-010-016	INGRESO AL SISTEMA
JULIO CESAR	CHÓEZ CATUTO	admin	201001-010-017	INGRESO AL SISTEMA
JULIO CESAR	CHÓEZ CATUTO	admin	201001-010-018	INGRESO AL SISTEMA
JULIO CESAR	CHÓEZ CATUTO	admin	201001-010-019	INGRESO AL SISTEMA
JULIO CESAR	CHÓEZ CATUTO	admin	201001-010-020	INGRESO AL SISTEMA

Figura 6.29 Auditoría

```

<%@page contentType="text/html" pageEncoding="UTF-8"
import="java.util.*" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head><meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
  <link href="css/estilos.css" rel="stylesheet" type="text/css" />
  <script language="javascript" src="scripts/usuarios.js"></script>
  <title>JSP Page</title>
</head>  <body>
  <form name="form1" method="post" action="">
    <table width="643" border="0" align="center">
      <tr><td width="123" class="estilos">Reporte en : </td>
        <td width="361"><select name="operacion" class="caja_texto"
id="operacion"> <option value="0">Pdf</option>
          <option value="1">Excel</option>
        </select>  <button value="Ingresar" class="botones"
onclick="exportar_reporte()" type="button">  Exportar</button>
        <button value="Ingresar" class="botones" onclick="imprimir()"
type="button">  Imprimir</button>
        <input type="hidden" name="reporte" value="acceso.jasper"
/></td>
      <td width="11">&nbsp;</td> </tr>
    </table>  </form>
    <table width="663" border="1" align="center" cellpadding="1"
cellspacing="0" bordercolor="#000000">
      <tr> <td width="129" bgcolor="#CCCCCC"><div
align="center"><strong>Usuario</strong></div></td>

```

```

        <td width="129" bgcolor="#CCCCCC"><div
align="center"><strong>Nombres</strong></div></td>
        <td width="129" bgcolor="#CCCCCC"><div
align="center"><strong>Apellidos</strong></div></td>
        <td width="150" bgcolor="#CCCCCC"><div
align="center"><strong>Fecha</strong></div></td>
        <td width="177" bgcolor="#CCCCCC"><div
align="center"><strong>Accion </strong></div></td>
    </tr> <% Clases.Dispositivos consulta =new Clases.Dispositivos();
        Vector consul=consulta.Consulta_acceso();
        for(int i=0;i<consul.size();i++){
            Vector info=(Vector)consul.elementAt(i);
            out.println("<tr>");
            out.println("<td> "+info.elementAt(3).toString()+"</td>");
            out.println("<td> "+info.elementAt(4).toString()+"</td>");
            out.println("<td> "+info.elementAt(0).toString()+"</td>");
            out.println("<td> "+info.elementAt(1).toString()+"</td>");
            out.println(" <td> "+info.elementAt(2).toString()+"</td>");
            out.println("</tr>");
        }
    %>
    </table>
</body></html>

```

6.30 PÁGINA "REP_CONECTADOS.JSP"

esta pagina permitirá ver los dispositivos que han sido detectados y almacenado por el sistema. Tambien podrá imprimir el documento.

Reporte en : Pdf Exportar Imprimir

Nombre	Direccion	Fecha de Detección	Fecha de Ultimo Acceso
Nokia 6131	00:18:0F:34:62:73	2009-06-05 21:41:54.0	2009-06-19 00:00:00.0
Julian	00:1D:5E:3E:A9:47	2009-06-05 22:32:10.0	2009-06-20 00:00:00.0
Glenn-Jove	00:21:9E:74:F0:15	2009-06-05 21:41:56.0	2009-06-19 00:00:00.0
Nokia 5200	00:1E:3B:AE:3C:07	2009-06-05 21:41:56.0	2009-06-16 00:00:00.0
Milenita	00:1B:AF:C7:5A:5E	2009-06-05 22:00:40.0	2009-06-17 00:00:00.0
Julioch	00:1B:AF:52:3C:5D	2009-06-05 22:32:54.0	2009-06-20 00:00:00.0
SAUL	00:1A:DC:97:5B:1A	2009-06-09 21:35:16.0	2009-06-19 00:00:00.0
Federico	00:1A:DC:8F:C0:33	2009-06-05 21:42:43.0	
Pablo	00:15:2A:6F:AA:95	2009-06-08 09:09:44.0	
SGH X685	00:18:AF:DF:7C:0C	2009-06-08 10:03:30.0	

Figura 6.30 Conectados

```

<%@page contentType="text/html" pageEncoding="UTF-8"
import="java.util.*" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html> <head> <meta http-equiv="Content-Type"
content="text/html; charset=UTF-8">
<title>Consulta de Dispositivos</title>
<link href="css/estilos.css" rel="stylesheet" type="text/css" />
<script language="javascript"
src="scripts/usuarios.js"></script>
<body> <form name="form1" method="post" action="">
<table width="643" border="0" align="center">
<tr> <td width="123" class="estilos">Reporte en : </td>
<td width="361"><select name="operacion" class="caja_texto"
id="operacion"><option value="0">Pdf</option>
<option value="1">Excel</option>
</select> <button value="Ingresar" class="botones"
onclick="exportar_reporte()" type="button">  Exportar</button>
<button value="Ingresar" class="botones" onclick="imprimir()"
type="button">  Imprimir</button> </td>
<td width="11"><input type="hidden" name="reporte"
value="rep_conectados.jasper" /></td> </tr> </table></form> <table
width="663" border="1" align="center" cellpadding="1" cellspacing="0"
bordercolor="#000000">
<tr> <td width="129" bgcolor="#CCCCCC"><div
align="center"><strong>Nombre</strong></div></td>
<td width="150" bgcolor="#CCCCCC"><div
align="center"><strong>Dirección</strong></div></td>
<td width="177" bgcolor="#CCCCCC"><div
align="center"><strong>Fecha de Detección</strong></div></td>
<td width="189" bgcolor="#CCCCCC"><div
align="center"><strong>Fecha de Ultimo Acceso</strong></div></td>
</tr> <%Clases.Dispositivos consulta =new Clases.Dispositivos();
Vector consul=consulta.Consulta_dispositivos();
for(int i=0;i<consul.size();i++){
Vector info=(Vector)consul.elementAt(i);
out.println("<tr>");
out.println("<td> "+info.elementAt(0).toString()+"</td>");
out.println("<td> "+info.elementAt(1).toString()+"</td>");
out.println(" <td> "+info.elementAt(2).toString()+"</td>");
if(info.elementAt(3)==null){
out.println("<td></td>"); }else{
out.println("<td> "+info.elementAt(3).toString()+"</td>");
} out.println("</tr>");
}
%> </table><p>
</p> </body> </html>

```


6.31 PÁGINA "REP_EXTERNOS.JSP"

Esta pagina permitirá realizar modificaciones a los usuarios externos registrados en la aplicación.

CONSULTA DE USUARIOS EXTERNOS					
Reporte en :	Pdf	Modificar	Eliminar	Exportar	Imprimir
	Nombres	Apellidos	Celular	Dispositivo	Nombre Dispositivo
	GIOMAR PAOLA	CHUEZ CATUTO	090192403	0015AFC72A0E	Milenia
	JULIO CÉSAR	CHOLZ CATUTO	097344226	001D6LLA947	Julian
	PAOLA CAROLINA	CATUTO CUELLO	090192402	00219L74015	Giomy-love
	TOMÁS LUIS	ERRARI MORA	088149508	00180104527D	Nokia 6131
	JUAN SEBASTIAN	CASTRO ROBLES	088568944	001LDBALJCD7	Nokia 5200
	ABRILIO ENRIQUE	MORLIRA JARA	097564854	001ADCB0033	Federico
	LOURDES ESTEFANIA	RUILO ARMAS	097356246	001LDBALJCD7	Nokia 5200
	MILNA PAOLA	FLORIS CORDERO	093505050	0013A1623C5D	Julioch
	JUAN FERNANDO	LOPEZ RAMIREZ	090392005	00180104527D	Nokia 6131

Figura 6.31 Reporte de usuarios externos

```
<%@page contentType="text/html" pageEncoding="UTF-8"
import="java.sql.*"%>
<jsp:useBean id="usuarios" scope="session" class="Clases.Usuarios" />
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<link href="css/estilos.css" rel="stylesheet" type="text/css" />
<script src="scripts/jsonrpc.js" type="text/javascript"></script>
<script language="javascript" src="scripts/usuarios.js"></script>
<title>Consultas de Usuarios</title>
</head>
<body>
<form name="form1" method="post" action="mod_usuarios.jsp">
<table width="641" border="0" align="center" cellpadding="1"
cellspacing="0" bordercolor="#CCCCCC" bgcolor="#CCCCCC">
<tr>
<td class="mensajes"> CONSULTA DE USUARIOS EXTERNOS
<input type="hidden" name="reporte" value="usuarios.jasper" /></td>
</tr>
</table>
<table width="643" border="0" align="center">
<tr>
<td width="95" class="estilos">Reporte en : </td>
```

```

        <td width="533"><select name="operacion" class="caja_texto"
id="operacion">      <option value="0">Pdf</option>
        <option value="1">Excel</option>
        </select>      <button value="Ingresar" class="botones"
onclick="externos_modifica()" type="button">  Modificar</button>
        <button value="Ingresar" class="botones"
onclick="exportar_reporte()" type="button">  Exportar</button>
        <button value="Ingresar" class="botones" onclick="imprimir()"
type="button"> Imprimir</button>
        </td><td width="10">&nbsp;</td></tr>
    </table> </form>
    <table width="641" border="1" align="center" cellpadding="1"
cellspacing="0" bordercolor="#000000">
<tr> <td width="20" bgcolor="#CCCCCC"><div align="center"></div></td>
        <td width="120" bgcolor="#CCCCCC"><div
align="center"><strong>Nombres</strong></div></td>
        <td width="120" bgcolor="#CCCCCC"><div
align="center"><strong>Apellidos</strong></div></td>
        <td width="52" bgcolor="#CCCCCC"><div
align="center"><strong>Celular</strong></div></td>
        <td width="100" bgcolor="#CCCCCC"><div
align="center"><strong>Dispositivo</strong></div></td>
        <td width="100" bgcolor="#CCCCCC"><div
align="center"><strong>Nombre Dispositivo</strong></div></td>
    </tr>    <% Clases.Conexion cone= new Clases.Conexion();
        cone.conectar();java.sql.ResultSet listado=cone.Consultas("select
id,nombres,apellidos,celular,dispositivo,funciones(3,dispositivo,') as
nombre_dispositivo from usuarios_externos where dispositivo is not null
order by id asc"); int e=0;
        while(listado.next()){
            out.println("<tr id='fila'+String.valueOf(e)+'"
onMouseOver='enciende('"+String.valueOf(e)+"',1)"
onMouseOut='apaga('"+String.valueOf(e)+"',1)'>");
            out.println("<td><input type='radio' name='codigo'
value='"+listado.getString("id")+"' id='RadioGroup1_0'
onClick='elementos('"+listado.getString("id")+"' /> </td>");
            out.println("<td>"+listado.getString("nombres").toUpperCase()+"</td>");
            out.println("<td>"+listado.getString("apellidos").toUpperCase()+"</td>");
            out.println("<td>"+listado.getString("celular")+"</td>");
            out.println("<td>"+listado.getString("dispositivo")+"</td>");
            out.println("<td>"+listado.getString("nombre_dispositivo")+"</td>");
            out.println("</tr>");
            e++; } cone.desconectar(); %>
    </table>    </body> </html>

```

6.32 PÁGINA "REP_PUBLICIDAD.JSP"

Esta pagina permite mediante graficos tar un reporte de enviados e ingorados.

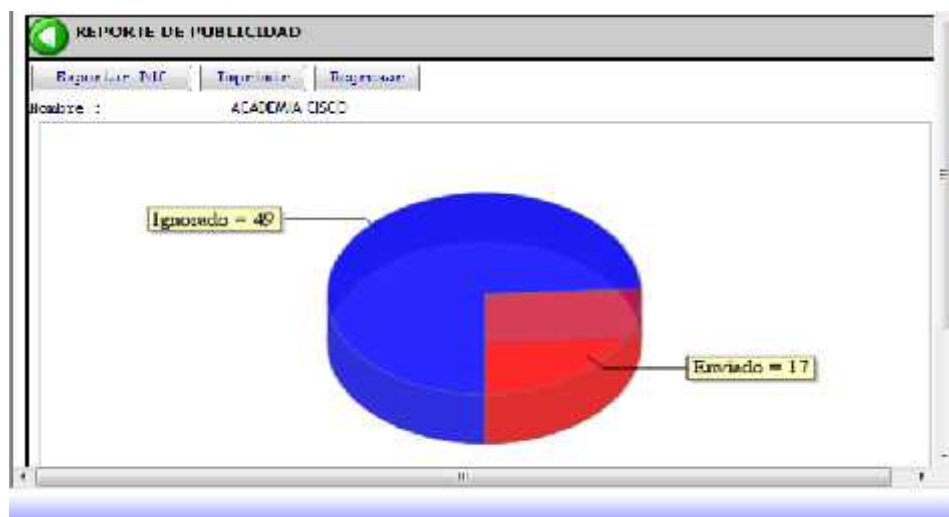


Figura 6.32 Reporte de publicidad

```
<%@ page contentType="text/html; charset=utf-8" language="java"
import="java.util.*,java.sql.*,java.text.*,java.io.*" errorPage="" %>
<jsp:useBean id="grafico"
class="org.jfree.chart.demo.servlet.WebHitChart"/>
<%
    String numero=request.getParameter("pub");
        String nombre=request.getParameter("nombre");
        Clases.Publicidad consulta=new Clases.Publicidad();
        String env=consulta.publicidad(8,Integer.parseInt(numero));
        String ign=consulta.publicidad(9,Integer.parseInt(numero));
        String contenido="";
        contenido="<?xml version='1.0' encoding='ISO-8859-1'?>";
contenido=contenido+"<dispositivo>Enviado</dispositivo><numero>" +env+"<
/numero><dispositivo>Ignorado</dispositivo><numero>" +ign+"</numero>";
String filename = grafico.generate3DPieChart(contenido, session, new
PrintWriter(out)); String graphURL = request.getContextPath() +
"/servlet/DisplayChart?filename=" + filename;
%><!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
        <title>JSP Page</title>
        <link href="css/estilos.css" rel="stylesheet" type="text/css" />
    </head>
    <body>
        <table width="641" border="0" align="center" cellpadding="0"
cellspacing="0" >
            <tr>
                <td class="mensajes" bgcolor="#CCCCCC">
REPORTE DE PUBLICIDAD </td></tr>    <tr>    <td class="mensajes"><table
width="629" border="0" cellspacing="0" cellpadding="1">
    <tr>    <td colspan="2">    <input name="button"
type="button" class="botones" id="button2" value="Exportar Pdf"
onClick="window.open('exporta_pdf.jsp')" />
    <input name="button" type="button" class="botones"
id="button2" value="Imprimir" onClick="window.print()" />
    <input name="Button" type="button" class="botones"
id="button" value="Regresar"
onClick="window.location='env_publicidad.jsp'" /></td>
    </tr>    <tr>    <td width="82" class="estilos">Nombre :</td>
    <td width="414"><%=nombre%></td>    </tr>    <tr>    <td colspan="2">
    <%=    String imagen="<img src='"+graphURL+"' width=700 height=350
border=0 usemap='#"+filename+"'>";
%>    <%=imagen%>    </td></tr>    </table></td>    </tr><tr>    <td
class="mensajes"><form name="form1" method="post" action="">    </form>
</td>    </tr>    </table>    <h1>&nbsp;</h1></body> </html>

```

6.33 PÁGINA "REP_USADOS.JSP"

Esta ventana permite saber que dispositivos han sido utilizados con mas frecuencia.

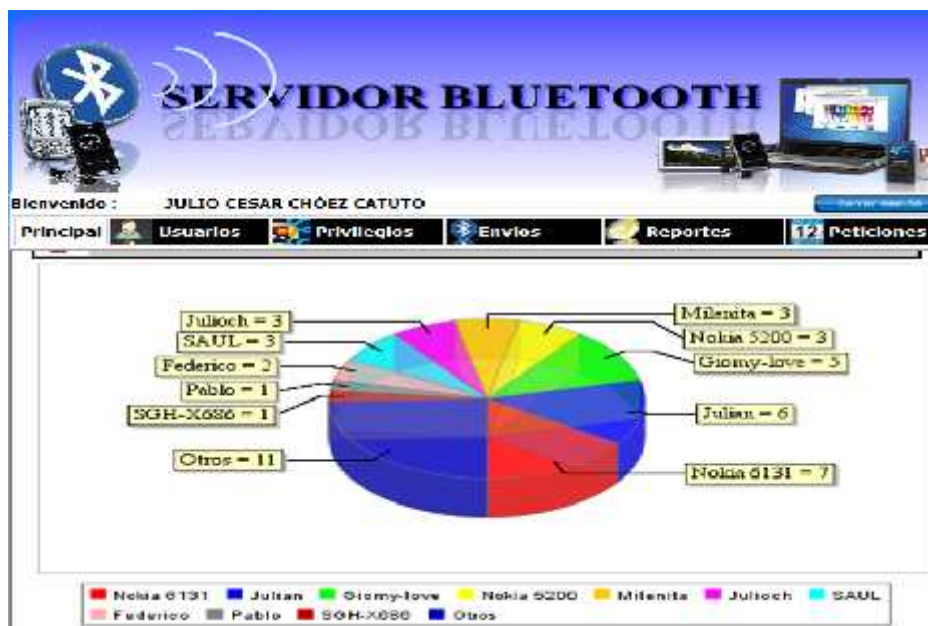


Figura 6.33 Reporte de los más usados

```

<%@ page contentType="text/html; charset=utf-8" language="java"
import="java.util.*,java.sql.*,java.text.*,java.io.*" errorPage="" %>
<jsp:useBean id="consulta" scope="session" class="Clases.Dispositivos"
/>
<jsp:useBean id="grafico"
class="org.jfree.chart.demo.servlet.WebHitChart"/>
<%
    Vector consul=consulta.Consulta_dispositivos();
    String contenido="";
    String str_where="";
    String dis="";
    contenido="<?xml version='1.0' encoding='ISO-8859-1'?>";
    for(int i=0;i<consul.size();i++){
        Vector info=(Vector)consul.elementAt(i);

contenido=contenido+"<dispositivo>"+info.elementAt(0).toString()+"</dis
positivo><numero>"+info.elementAt(4).toString()+"</numero>";
        dis=info.elementAt(1).toString();
        dis=dis.replace(":", " ");
        str_where=str_where.concat("direccion='"+dis+"'");
        if(i+1<consul.size()){
            str_where=str_where.concat(" or ");
        }
        str_where=" select count(numeros) as Objecto from
dispositivos where not("+str_where+")";
        String valor=consulta.Consulta_Otros(str_where);
        if(valor.compareTo("0")!=0){
contenido=contenido+"<dispositivo>Otros</dispositivo><numero>"+valor+"<
/numero>";
        }
String filename = grafico.generatePieChart(contenido, session, new
PrintWriter(out));
String graphURL = request.getContextPath() +
"/servlet/DisplayChart?filename="+ filename;
String imagen="<img src='"+graphURL+"' width=700 height=350 border=0
usemap='#"+filename+"'>";
%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html> <head>
    <meta http-equiv="Content-Type" content="text/html;
charset=windows-1252">
    <title>Reporte Mas Usuado</title> </head>
    <body> <table width="200" border="0" align="center">
        <tr> <td><div align="center"><%=imagen%></div></td>
        </tr></table>

</body>
</html>

```

6.34 PÁGINA "SALIDA.JSP"

Esta ventana permite ver mediante opciones de visualización los mensajes que fueron enviados desde el sistema.

Dispositivo	Nombre	Mensaje	Fecha
00180FH467/H	Nokia 6131	Nota	2009-06-09 21:49:59.0
0010A1C75A5L	Milenita	Nota	2009-06-09 21:50:14.0
001E3BAE3C07	Nokia 5200	Nota	2009-06-09 21:50:23.0
00219L74H015	Granny Love	Nota	2009-06-09 21:50:38.0
0010AFC75A5C	Milenita	Por favor Ingrese el mensaje a enviar...	2009-06-09 09:00:41.0
00106FH467/H	Julian	PRIMER CONTENIDO	2009-06-09 21:51:41.0
00106LULA917	Julian	PRIMER CONTENIDO	2009-06-09 21:51:56.0
00180FH4627B	Nokia 6131	PRIMER CONTENIDO	2009-06-09 21:52:14.0
0010AFC75A5D	Julian	PRIMER CONTENIDO	2009-06-09 21:53:00.0
0010AFC75A5A	SALL	PRIMER CONTENIDO	2009-06-09 21:53:08.0
00106E8EA947	Julian	PRIMER CONTENIDO	2009-06-09 21:53:26.0
00180FH467/H	Nokia 6131	PRIMER CONTENIDO	2009-06-09 21:53:36.0
00219L74H015	Granny Love	PRIMER CONTENIDO	2009-06-09 21:53:41.0
001EAF523C5D	Julian	PRIMER CONTENIDO	2009-06-09 21:53:45.0
0010AFC75A5A	SALL	PRIMER CONTENIDO	2009-06-09 21:53:53.0
00180FH4627U	Nokia 6131	GDFAHSGDFJAHSGF5	2009-06-12 20:05:06.0
001EAF523C5E	Milenita	GDFAHSGDFJAHSGF5	2009-06-12 20:05:30.0
00219L74H015	Granny Love	GDFAHSGDFJAHSGF5	2009-06-12 20:05:39.0
00180FH4627U	Nokia 6131	GDFAHSGDFJAHSGF5	2009-06-12 20:05:47.0
001E3BAE3C07	Nokia 5200	GDFAHSGDFJAHSGF5	2009-06-12 20:05:55.0

Figura 6.34 Buzón de salida

```
<%@page contentType="text/html" pageEncoding="UTF-8"
import="java.util.*" %>
<jsp:useBean id="consulta" scope="session" class="Clases.Dispositivos"
/><jsp:useBean id="fecha" scope="session" class="Clases.Fechas" />
<script src="scripts/formexp.js" type="text/javascript"></script>
<script type="text/javascript" src="scripts/prototype.js"
language="javascript"></script>
<script type="text/javascript" src="popcalendar.js"
language="javascript"></script>
<script src="scripts/buzones.js"
type="text/javascript"></script>
<script type="text/javascript" src="scripts/dispositivos.js"
language="javascript"></script>
<link href="css/estilos.css" rel="stylesheet" type="text/css"
/>
<%
Vector consul=null;
String f1=""; String f2="";
%>
```

```

<body> <form name="form1" method="post" action="salida.jsp">
  <table width="668" border="0" align="center" cellpadding="1"
cellspacing="0" bordercolor="#CCCCCC" bgcolor="#CCCCCC">
<tr> <td class="mensajes"> BUZON DE SALIDA</td>
  </tr> </table> <table width="666" border="0" align="center">
  <tr> <td width="107" class="estilos">Reporte en : </td>
    <td width="537"><select name="operacion" class="caja_texto"
id="operacion"><option value="0">Pdf</option>
      <option value="1">Excel</option> </select>
      <button value="Ingresar" class="botones"
onclick="exportar_reporte()" type="button">  Exportar</button>
      <button value="Ingresar" class="botones"
onclick="imprimir()" type="button">  Imprimir</button>
      <button value="Ingresar" class="botones" onclick="buscar()"
type="button">  Buscar</button>
      <input name="reporte" type="hidden" id="reporte"
value="buzones.jasper" />
      <input name="tipo" type="hidden" id="tipo" value="1" />
      <input name="repo" type="hidden" id="repo" value="SALIDA" />
      <input name="fecha_hoy" type="hidden" id="fecha_hoy"
value="<%=fecha.f1%>"> <input name="fecha_semana" type="hidden"
id="fecha_semana" value="<%=fecha.f2%>">
      <input name="fecha_quincena" type="hidden"
id="fecha_quincena" value="<%=fecha.f3%>">
      <input name="fecha_mes" type="hidden" id="fecha_mes"
value="<%=fecha.f4%>"> </td></tr>
  <tr> <td colspan="2" class="estilos">
    <table width="644" border="0">
      <tr><td width="105">Visualizaci&ocute;n : </td>
        <td width="529"><select name="reportes" id="reportes"
onchange="ver_reporte()">
          <option value="1" selected>DIARIO</option>
          <option value="2">SEMANA</option>
          <option value="3">QUINCENAL</option>
          <option value="4">MENSUAL</option>
          <option value="5">PERSONALIZADO</option>
        </select> </td> </tr>
    </table><div id="capaexpansion">
<table width="669" border="0" cellspacing="0" cellpadding="1"> <tr>
  <td width="107">Fecha desde :</td>
  <td width="187"><input name="f1" type="text"
class="caja_texto" id="f1" value="" readonly="true">

```

```

        </td>
        <td width="98">Fecha hasta :</td>
        <td width="182"><input name="f2" type="text"
class="caja_texto" id="f2" value="" readonly="true">
        </td>
        <td width="64">&nbsp;</td>
    </tr></table>
    </div> </td>
</tr></table> </p>
<div id="consulta_divd">
    <table width="663" border="1" align="center" cellpadding="1"
cellspacing="0" bordercolor="#000000">
        <tr> <td width="129" bgcolor="#CCCCCC"><div
align="center"><strong>Dispositivo</strong></div></td>
        <td width="129" bgcolor="#CCCCCC"><div
align="center"><strong>Nombre</strong></div></td>
        <td width="150" bgcolor="#CCCCCC"><div
align="center"><strong>Mensaje</strong></div></td>
        <td width="177" bgcolor="#CCCCCC"><div
align="center"><strong>Fecha </strong></div></td>
    </tr> </table>
</div> <script>        oculta_div();
    </script>        </form>
</body>

```

7 FUNCIONES

DELIMITER \$\$

```

/*!50003 SET @TEMP_SQL_MODE=@@SQL_MODE,
SQL_MODE='STRICT_TRANS_TABLES,NO_AUTO_CREATE_USER' */ $$

-- FUNCTION FUNCIONES
-- PARAMETROS
-- OPERACION INTEGER
-- VALOR1 VARCHAR(50)
-- VALOR2 VARCHAR(50)
-- RETORNA VARCHAR
-- ESTA FUNCION PERMITE REALIZAR CONSULTAS A LAS BASE DE DATOS
DEPENDIENDO LA OPERACION QUE EL USUARIO DESEA REALIZAR
-- SOLO RETORNA UN SOLO DATO DE TIPO VARCHAR
CREATE DEFINER=`root`@`localhost` FUNCTION `FUNCIONES`(OPERACION
INT,VALOR1 VARCHAR(50),VALOR2 VARCHAR(50)) RETURNS varchar(50) CHARSET
latin1
BEGIN
declare respuesta varchar(50);

```



```

if(operacion=1)then
-- CONSULTA A LA TABLA USUARIOS SEGUN EL USUARIO Y RETORNA EL NOMBRE
DEL USUARIO
    select nombres into respuesta from usuarios where usuario=VALOR1;
end if;
if(operacion=2)then
-- CONSULTA A LA TABLA USUARIOS SEGUN EL USUARIO Y RETORNA EL APELLIDO
DEL USUARIO
    select apellidos into respuesta from usuarios where usuario=VALOR1;
end if;
if(operacion=3)then
-- CONSULTA LOS DISPOSITIVOS SEGUN LA DIRECCION
    select nombre into respuesta from dispositivos where
direccion=VALOR1;
end if;
if(operacion=4)then
-- CONSULTA LA TABLA MATERIAS DEVUELVE EL NOMBRE
    select nombre into respuesta from materias where codigo=VALOR1;
end if;
if(operacion=5)then
-- CONSULTA LA TABLA MATERIAS DEVUELVE EL SEMESTRE
    select semestre into respuesta from materias where codigo=VALOR1;
end if;
if(operacion=6)then
-- CONSULTA A LA PUBLICIDAD SEGUN EL CODIGO
    select recibidos into respuesta from publicidad where id=VALOR1;
end if;
if(operacion=7)then
-- CONSULTA TABLA PUBLICIDAD SEGUN EL CODIGO
    select ignorados into respuesta from publicidad where id=VALOR1;
end if;
if(operacion=8)then
-- CONSULTA TABLA PERMISOS SEGUN EL CODIGO
    select nombre into respuesta from permisos where codigo=VALOR1;
end if;
set respuesta=IFNULL(respuesta,'');
-- RETORNA RESULTADO
return respuesta;
END $$
/*!50003 SET SESSION SQL_MODE=@TEMP_SQL_MODE */ $$
DELIMITER ;

```

8. PROCEDIMIENTOS ALMACENADOS MY_SQL

Los procedimientos son los encargados de realizar los mantenimiento dentro de la base datos con los parametros enviado desde las clases e identificado por un tipo que indica que opcion se va a realizar, como por ejemplo Consulta("C"), Inserccion("I"), Modificacion("U"), Eliminacion("D"), y en cierto procedimiento almacenado la "B" que es una consulta especificaz basado en un parametro especifico. Ademas tienen una variable de retorno que nos especifican, en los tipos de mantenimiento mas no de consulta, si la operación se realizo con éxito ("S") o si hubo algun erros ("N").

Los procedimientos almacenados llevan el mismo nombre de las tablas a las que afectan.

8.1 PROCEDIMIENTO ALMACENADO "CONSULTAS"

Este procedimiento se encarga de verficar los datos de la tabla levantamiento la cual posee las fechas de procesos de informacion. Adicional se encarga de dar mantenimiento a dicha tabla.

```
-- =====
-- Author:          GRUPO # 13 PARALELO A REDES
-- Create date: 28/11/2009
-- Description:     Tabla
-- =====
-- Definition of procedure `consultas`

-- PROCEDIMIENTO CONSULTA
```

```

-- PARAMETROS OPERACION PARAMETRO1,PARAMETRO2,PARAMATRO3,PARAMETRO4
-- PERMITE REALIZAR CONSULTA A LA TABLAS SEGUN LA OPERACION QUE EL
SISTEMA EJECUTE
-- RETORNA RESULTSET DE LA CONSULTA REALIZADA
DROP PROCEDURE IF EXISTS `consultas`;
DELIMITER $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `consultas`(OPERACION
INT,PARAMETRO1 VARCHAR(50),PARAMETRO2 VARCHAR(50),PARAMETRO3
VARCHAR(50),PARAMETRO4 VARCHAR(50))
BEGIN
IF(OPERACION=1)THEN
    SELECT idBitacora,funciones(3,a.usuario,',','(',') as
users,funciones(3,a.idPersona,',','(',') as
cliente,a.fecha,a.hora,a.descripcion as problema,b.descripcion as tipo
FROM bitacora a, tipo_bitacora b where a.tipo=b.IdTipo_Bitacora and
a.estado=2;
END IF;
IF(OPERACION=2)THEN
    select idPersonas,usuario from personas  where  idtipo_usuario=2;
end if;
IF(OPERACION=3)THEN
    select idEstados_Calificacion,descripcion from estados_calificacion;
end if;
if(OPERACION=4)THEN
    SELECT * FROM PERSONAS;
END IF;
IF(OPERACION=5)THEN
    SELECT idBitacora,funciones(3,a.usuario,',','(',') as
users,funciones(3,a.idPersona,',','(',') as
cliente,a.fecha,a.hora,a.descripcion as problema,b.descripcion as tipo
FROM bitacora a, tipo_bitacora b where a.tipo=b.IdTipo_Bitacora and
a.estado=1 order by a.idbitacora asc;
END IF;
IF(OPERACION=6)THEN
    if(STRCMP(PARAMETRO1,"0")=0)THEN
        SET PARAMETRO1 ="%";
    END IF;
    if(STRCMP(PARAMETRO2,"0")=0)THEN
        SET PARAMETRO2 ="%";
    END IF;
    if(STRCMP(PARAMETRO3,"0")=0)THEN
        SET PARAMETRO3 ="%";
    END IF;
    SELECT idBitacora,funciones(3,a.usuario,',','(',') as
users,funciones(3,a.idPersona,',','(',') as
cliente,a.fecha,a.hora,a.descripcion as problema,b.descripcion as tipo
FROM bitacora a, tipo_bitacora b where a.tipo=b.IdTipo_Bitacora and
a.usuario like PARAMETRO1 and a.estado like PARAMETRO2 and
a.calificacion like PARAMETRO3;

```

```

END IF;
IF(OPERACION=7)THEN
    SELECT * FROM TIPO_BITACORA;
END IF;
IF(OPERACION=8)THEN
    SELECT
idpersonas,USUARIO,NOMBRES,APELLIDOS,funciones(7,idpersonas','','','')
as soporte FROM PERSONAS WHERE IDTIPO_USUARIO=2;
END IF;
IF(OPERACION=9)THEN
    SELECT USUARIO,NOMBRES,APELLIDOS,funciones(10,idpersonas','','','')
as soporte FROM PERSONAS WHERE idpersonas=PARAMETRO1;
END IF;
IF(OPERACION=10)THEN
    SELECT idTipo_bitacora,descripcion FROM tipo_bitacora;
END IF;
IF(OPERACION=11)THEN
    SELECT descripcion FROM tipo_bitacora where
idTipo_bitacora=PARAMETRO1;
END IF;
IF(OPERACION=12)THEN
    SELECT a.idpersonas,a.cedula,a.apellidos,a.usuario,b.direccion_casa
FROM personas a,direccion b where a.id_direccion=b.iddireccion and
idtipo_usuario=3;
END IF;
IF(OPERACION=13)THEN
    SELECT * FROM personas a,direccion b where
a.id_direccion=b.iddireccion and a.idtipo_usuario=3 and
a.idPersonas=PARAMETRO1;
END IF;
IF(OPERACION=14)THEN
    SELECT a.usuario,a.cedula,a.apellidos,a.nombres,b.direccion_casa
FROM personas a,direccion b,organizacion_cliente c where
a.id_direccion=b.iddireccion and a.idtipo_usuario=4 and
a.idPersonas=c.cliente and c.organizacion=PARAMETRO1;
END IF;
IF(OPERACION=15)THEN
    SELECT idBitacora,funciones(3,a.usuario','','','') as
users,funciones(3,a.idPersona','','','') as
cliente,a.fecha,a.hora,a.descripcion as problema,b.descripcion as tipo
FROM bitacora a, tipo_bitacora b where a.tipo=b.IdTipo_Bitacora and
a.estado=1 and a.organizacion=PARAMETRO1;
END IF;
IF(OPERACION=16)THEN
    SELECT count(idBitacora) as numero,funciones(12,tipo','','','') as
nombre FROM bitacora where organizacion=PARAMETRO1 and tipo in (select
idtipo_bitacora from tipo_bitacora) group by tipo;
END IF;
IF(OPERACION=17)THEN

```

```

        SELECT codigo,nombre,cantidad,descripcion FROM productos where
organizacion=PARAMETRO1;
END IF;
IF(OPERACION=18)THEN
    SELECT
a.idpersonas,a.usuario,a.cedula,a.apellidos,a.nombres,b.direccion_casa
FROM personas a,direccion b,organizacion_cliente c where
a.id_direccion=b.iddireccion and a.idtipo_usuario=4 and
a.idPersonas=c.cliente and c.organizacion=PARAMETRO1;
END IF;
IF(OPERACION=19)THEN
    SELECT id,factura,funciones(11,cliente','','','') as users FROM
ventas where organizacion=PARAMETRO1;
END IF;
IF(OPERACION=20)THEN
    SELECT * FROM proveedor;
END IF;
IF(OPERACION=21)THEN
    SELECT * FROM proveedor where id=PARAMETRO1;
END IF;
IF(OPERACION=22)THEN
    SELECT * FROM paquetes;
END IF;
IF(OPERACION=23)THEN
    SELECT * FROM paquetes where id=PARAMETRO1;
END IF;
IF(OPERACION=24)THEN
    SELECT distinct
idpersonas,usuario,funciones(20,idpersonas','','','') as ordenes FROM
personas where idtipo_usuario=2;
END IF;
IF(OPERACION=25)THEN
    SELECT usuario,funciones(3,usuario','','','') as
usuario,idbitacora,funciones(12,tipo','','',''),funciones(11,idpersona,
','','') as cliente FROM bitacora where estado=2 and
usuario=PARAMETRO1;
END IF;
if(operacion=26)then
    SELECT idBitacora,funciones(3,a.usuario','','','') as
users,funciones(11,a.idPersona','','','') as
cliente,a.fecha,a.hora,a.descripcion as problema,b.descripcion as tipo
FROM bitacora a, tipo_bitacora b where a.tipo=b.IdTipo_Bitacora and
a.estado=2 and a.usuario=PARAMETRO1 order by idbitacora;
end if;
if(operacion=27)then
    SELECT idBitacora,funciones(3,a.usuario','','','') as
users,funciones(11,a.idPersona','','','') as
cliente,a.fecha,a.hora,a.descripcion as problema,b.descripcion as tipo

```

```

FROM bitacora a, tipo_bitacora b where a.tipo=b.IdTipo_Bitacora and
a.estado=1 and a.usuario=PARAMETRO1;
end if;
if(operacion=28)then
    SELECT idBitacora,funciones(11,idPersona,'','','') as
cliente,descripcion FROM bitacora where idbitacora=PARAMETRO1;
end if;
if(operacion=29)then
    SELECT idBitacora,funciones(3,a.usuario,'','','') as
users,funciones(11,a.idPersona,'','','') as
cliente,a.fecha,a.hora,a.descripcion as problema,b.descripcion as
tipo,a.observacion,a.idPersona FROM bitacora a, tipo_bitacora b where
a.tipo=b.IdTipo_Bitacora and a.idBitacora=PARAMETRO1 order by
idbitacora;
end if;
if(operacion=30)then
    SELECT idBitacora,funciones(3,a.usuario,'','','') as
users,funciones(11,a.idPersona,'','','') as
cliente,a.fecha,a.hora,a.descripcion as problema,b.descripcion as
tipo,a.observacion,a.idPersona FROM bitacora a, tipo_bitacora b where
a.tipo=b.IdTipo_Bitacora and a.idPersona=PARAMETRO1 and a.estado=1
order by idbitacora;
end if;

if(operacion=31)then
    SELECT idBitacora,funciones(3,a.usuario,'','','') as
users,funciones(11,a.idPersona,'','','') as
cliente,a.fecha,a.hora,a.descripcion as problema,b.descripcion as
tipo,a.observacion,a.idPersona FROM bitacora a, tipo_bitacora b where
a.tipo=b.IdTipo_Bitacora and a.idPersona=PARAMETRO1 and a.estado=2
order by idbitacora;
end if;
if(operacion=32)then
    SELECT * FROM personas where conectado=1 and idpersonas in (select
distinct idpersona from bitacora where usuario=PARAMETRO1);
end if;
if(operacion=33)then
    SELECT * from preguntas_frecuentes;
end if;

END $$
/*!50003 SET SESSION SQL_MODE=@TEMP_SQL_MODE */  $$
DELIMITER ;

```

8.2 PROCEDIMIENTO ALMACENADO “CONTADOR MENSAJES”

Se encarga de dar mantenimiento a la tabla “Producto”, al igual que los demas SP que llevan el mismo nombre de las tablas de la herramienta.

```
-- =====
-- Author:          GRUPO # 13 PARALELO A REDES
-- Create date: 28/11/2009
-- Description:     Tabla
-- =====
-- Definition of procedure `CONTADOR_MENSAJES`

DROP PROCEDURE IF EXISTS `CONTADOR_MENSAJES`;
DELIMITER $$
-- PROCEDIMIENTO CONTADOR_MENSAJES
-- PARAMETROS IESTADO,F1,F2
-- PERMITE REALIZAR CONSULTA DEPENDIENDO EL ESTADO LOS MENSAJES Y
FECHAS INGRESADAS.
-- ESTADO 1 ENTRADA
-- ESTADO 2 SALIDA
-- ESTADO 3 NO ENVIADO
-- RETORNA EL NUMERO DE REGISTROS DE LA CONSULTA TIPO VARCHAR
CREATE DEFINER=`root`@`localhost` PROCEDURE `CONTADOR_MENSAJES`(IESTADO
VARCHAR(1),F1 VARCHAR(30),F2 VARCHAR(30))
BEGIN
IF(IESTADO=0)THEN
    SELECT COUNT(*) AS numero FROM MENSAJES WHERE tipo=IESTADO AND
(FECHA>=F1 AND FECHA<=F2);
END IF;IF(IESTADO=1)THEN
    SELECT COUNT(*) AS numero FROM MENSAJES WHERE tipo=IESTADO AND
(FECHA>=F1 AND FECHA<=F2);
END IF;
IF(IESTADO=2)THEN
    SELECT COUNT(*) AS numero FROM MENSAJES WHERE tipo=IESTADO AND
(FECHA>=F1 AND FECHA<=F2);
END IF;END $$
/*!50003 SET SESSION SQL_MODE=@TEMP_SQL_MODE */  $$
DELIMITER ;
```

8.3 PROCEDIMIENTO ALMACENADO “MANTENIMIENTO_BORRADORES”

```
-- =====
-- Author:          GRUPO # 13 PARALELO A REDES
-- Create date: 28/11/2009
-- Description:     Tabla
-- =====
DELIMITER $$
-- PROCEDIMIENTOS MANTENIMIENTO_BORRADORES
-- PARAMETROS OPERACION ITITULO ICONTENIDO CODIGO
-- PERMITE REALIZAR INGRESO, MODIFICACIÓN Y ELIMINACIÓN DE LA TABLA
BORRADOS
-- OPERACION = 1 = INGRESO
-- OPERACION = 2 = MODIFICACIÓN
-- OPERACION = 3 = ELIMINACIÓN
CREATE DEFINER=`root`@`localhost` PROCEDURE
`MANTENIMIENTO_BORRADORES`(IN OPERACION INTEGER,IN ITITULO
VARCHAR(50),IN ICONTENIDO VARCHAR(255),IN CODIGO INTEGER)
BEGIN
    if(OPERACION=1)then
        insert into borrados
(titulo,contenido)VALUES(ITITULO,ICONTENIDO);
    end if;
    if(OPERACION=2)then
        update borrados set titulo=ITITULO,contenido=ICONTENIDO where
id=CODIGO;
    end if;
    if(OPERACION=3)then
        delete from borrados where id=CODIGO;
    end if;
END $$
/*!50003 SET SESSION SQL_MODE=@TEMP_SQL_MODE */  $$
DELIMITER ;
```

8.4 PROCEDIMIENTO ALMACENADO “MANTENIMIENTO_DISPOSITIVOS”

Es un procedimiento almacenado importante porque realiza el proceso de escoger los datos de las transacciones seleccionadas por el usuario como críticas.


```

-- =====
-- Author:          GRUPO # 13 PARALELO A REDES
-- Create date: 28/04/2009
-- Description:     Tabla
-- =====

-- Definition of procedure `MANTENIMIENTO_DISPOSITIVOS`
--

DROP PROCEDURE IF EXISTS `MANTENIMIENTO_DISPOSITIVOS`;

DELIMITER $$

-- PROCEDIMIENTOS MANTENIMIENTO_DISPOSITIVOS
-- PARAMETROS INOMBRE IDIRECCION , OPERACION
-- PERMITE REALIAR INGRESO,MODIFICACION Y ELIMINACION DE LA TABLA
DISPOSITIVOS
-- OPERACION = 1 = INGRESO
-- OPERACION = 2 = MODIFICACION
-- OPERACION = 3 = ELIMINACION

CREATE DEFINER=`root`@`localhost` PROCEDURE
`MANTENIMIENTO_DISPOSITIVOS`(IN INOMBRE VARCHAR(40),IN IDIRECCION
VARCHAR(50),IN OPERACION INTEGER)
BEGIN
DECLARE NUMERO INTEGER;
IF(OPERACION=1)THEN
    SELECT COUNT(*) INTO NUMERO FROM DISPOSITIVOS WHERE
DIRECCION=IDIRECCION;
    IF(NUMERO=1)THEN
        UPDATE DISPOSITIVOS SET FECHA_ACCESO=NOW();
    ELSE
        INSERT INTO DISPOSITIVOS
(NOMBRE,DIRECCION,FECHA_DECTECION)VALUES( INOMBRE ,IDIRECCION ,NOW( ) );
        SELECT 1;
    END IF;
END IF;
IF(OPERACION=2)THEN
    UPDATE DISPOSITIVOS SET NUMEROS=NUMEROS+1 WHERE DIRECCION=IDIRECCION;
END IF;
END $$
/*!50003 SET SESSION SQL_MODE=@TEMP_SQL_MODE */  $$

DELIMITER ;

```

8.5 PROCEDIMIENTO ALMACENADO “MANTENIMIENTO_EXTERNOS”

Este procedimiento almacenado se encarga de recoger la informacion de transacciones especificas del usuario seleccionado.

```
-- =====
-- Author:          GRUPO # 13 PARALELO A REDES
-- Create date: 28/11/2009
-- Description:     Tabla
-- =====

-- Definition of procedure `MANTENIMIENTO_EXTERNOS`
DROP PROCEDURE IF EXISTS `MANTENIMIENTO_EXTERNOS`;
DELIMITER $$
-- PROCEDIMIENTOS MANTENIMIENTO_EXTERNOS
-- PARAMETROS OPERACION INOMBRES IAPELLIDOS ICELULAR IDISPOSITIVO
ICEDULA ICODIGO
-- PERMITE REALIAR INGRESO,MODIFICACION Y ELIMINACION DE LA TABLA
USUARIOS_EXTERNOS
-- OPERACION = 1 = INGRESO
-- OPERACION = 2 = MODIFICACION
-- OPERACION = 3 = ELIMINACION
CREATE DEFINER=`root`@`localhost` PROCEDURE `MANTENIMIENTO_EXTERNOS`(IN
OPERACION INTEGER,IN INOMBRES VARCHAR(50),IN IAPELLIDOS VARCHAR(50),IN
ICELULAR VARCHAR(10),IN IDISPOSITIVO VARCHAR(40),IN ICEDULA
VARCHAR(10),IN ICODIGO INTEGER)
BEGIN
if(OPERACION=1)then
INSERT INTO USUARIOS_EXTERNOS
(NOMBRES,APELLIDOS,CELULAR,DISPOSITIVO,CEDULA)VALUES( INOMBRES ,IAPELLIDO
S, ICELULAR, IDISPOSITIVO, ICEDULA);
end if;
if(OPERACION=2)THEN
UPDATE USUARIOS_EXTERNOS SET
NOMBRES=INOMBRES,APELLIDOS=IAPELLIDOS,CELULAR=ICELULAR,DISPOSITIVO=IDIS
POSITIVO,CEDULA=ICEDULA WHERE ID=ICODIGO;
END IF;
if(OPERACION=3)THEN
DELETE FROM USUARIOS_EXTERNOS WHERE ID=ICODIGO;
END IF;
END $$
/*!50003 SET SESSION SQL_MODE=@TEMP_SQL_MODE */ $$
DELIMITER ;
```

8.6 PROCEDIMIENTO ALMACENADO “MANTENIMIENTO_PETICIONES”

Este procedimiento almacenado se encarga de mostrar todos los modulos a los que tiene acceso un usuario ademas de aquellos modulos a los que no tiene acceso.

```
-- =====
-- Author:          GRUPO # 13 PARALELO A REDES
-- Create date: 28/04/2009
-- Description:     Tabla
-- =====

-- Definition of procedure `MANTENIMIENTO_PETICIONES`
DROP PROCEDURE IF EXISTS `MANTENIMIENTO_PETICIONES`;
DELIMITER $$
-- PROCEDIMIENTOS MANTENIMIENTO_PETICIONES
-- PARAMETROS IPETICION IARGUMENTO IRESPUESTA ICODIGO OPERACION
-- PERMITE REALIAR INGRESO,MODIFICACION Y ELIMINACION DE LA TABLA
PETICIONES
-- OPERACION = 1 = INGRESO
-- OPERACION = 2 = MODIFICACION
-- OPERACION = 3 = ELIMINACION
CREATE DEFINER=`root`@`localhost` PROCEDURE
`MANTENIMIENTO_PETICIONES`(IN IPETICION VARCHAR(40),IN IARGUMENTO
VARCHAR(40),IN IRESPUESTA TEXT,IN ICODIGO INTEGER,IN OPERACION INTEGER)
BEGIN
IF (OPERACION=1)THEN
INSERT INTO sms.peticiones
(PETICION,ARGUMENTO,RESPUESTA)VALUES( IPETICION,IARGUMENTO,IRESPUESTA);
ELSEIF (OPERACION=2)THEN
UPDATE sms.peticiones SET
PETICION=IPETICION,ARGUMENTO=IARGUMENTO,RESPUESTA=IRESPUESTA WHERE
CODIGO=ICODIGO;
ELSEIF (OPERACION=3)THEN
DELETE FROM sms.peticiones WHERE ID=ICODIGO;
END IF;
END $$
/*!50003 SET SESSION SQL_MODE=@TEMP_SQL_MODE */  $$
DELIMITER ;
```

8.7 PROCEDIMIENTO ALMACENADO “MANTENIMIENTO_PUBLICIDAD”

Este procedimiento almacenado se encarga de mostrar todos los modulos a los que tiene acceso un usuario ademas de aquellos modulos a los que no tiene acceso.

```
-- =====
-- Author:          GRUPO # 13 PARALELO A REDES
-- Create date: 28/11/2009
-- Description:     Tabla
-- =====

-- Definition of procedure `MANTENIMIENTO_PUBLICIDAD`
DROP PROCEDURE IF EXISTS `MANTENIMIENTO_PUBLICIDAD`;
DELIMITER $$
-- PROCEDIMIENTO MANTENIMIENTO_PUBLICIDAD
-- PARAMETROS OPERACION INOMBRE IRUTA ID1 ID2 ID3 ID4 ID5 ID6 ID7
-- IESTADO ICODIGO
-- PERMITE REALIAR INGRESO,MODIFICACION Y ELIMINACION DE LA TABLA
-- PUBLICIDAD
-- OPERACION = 1 = INGRESO
-- OPERACION = 2 = MODIFICACION
-- OPERACION = 3 = ELIMINACION
CREATE DEFINER=`root`@`localhost` PROCEDURE
`MANTENIMIENTO_PUBLICIDAD`(OPERACION INTEGER,INOMBRE VARCHAR(50),IRUTA
VARCHAR(200),ID1 INTEGER,ID2 INTEGER,ID3 INTEGER,ID4 INTEGER,ID5
INTEGER,ID6 INTEGER,ID7 INTEGER,IESTADO INTEGER,ICODIGO INTEGER)
BEGIN
IF(OPERACION=1)THEN
    INSERT INTO PUBLICIDAD
    (NOMBRE,RUTA,D1,D2,D3,D4,D5,D6,D7,ESTADO)VALUES( INOMBRE,IRUTA,ID1,ID2,I
D3,ID4,ID5,ID6,ID7,IESTADO);
END IF;
IF(OPERACION=2)THEN
    UPDATE PUBLICIDAD SET
    NOMBRE=INOMBRE,D1=ID1,D2=ID2,D3=ID3,D4=ID4,D5=ID5,D6=ID6,D7=ID7,ESTADO=
IESTADO WHERE ID=ICODIGO;
END IF;
IF(OPERACION=3)THEN
    DELETE FROM PUBLICIDAD WHERE ID=ICODIGO;
END IF;
IF(OPERACION=4)THEN
    UPDATE PUBLICIDAD SET RECIBIDOS=RECIBIDOS+1 WHERE ID=ICODIGO;
```

```

END IF;
IF(OPERACION=5)THEN
    UPDATE PUBLICIDAD SET IGNORADOS=IGNORADOS+1 WHERE ID=ICODIGO;
END IF;
END $$
/*!50003 SET SESSION SQL_MODE=@TEMP_SQL_MODE */  $$
DELIMITER ;

```

8.8 PROCEDIMIENTO ALMACENADO “MANTENIMIENTO_USUARIOS”

Este procedimiento almacenado se encarga de mostrar todos los modulos a los que tiene acceso un usuario ademas de aquellos modulos a los que no tiene acceso.

```

-- =====
-- Author:          GRUPO # 13 PARALELO A REDES
-- Create date: 28/11/2009
-- Description:     Tabla
-- =====

-- Definition of procedure `MANTENIMIENTO_USUARIOS`

DROP PROCEDURE IF EXISTS `MANTENIMIENTO_USUARIOS`;

DELIMITER $$
-- PROCEDIMIENTO MANTENIMIENTO_USUARIOS
-- PARAMETROS INOMBRES IAPELLIDOS IUSUARIO ICELULAR ICLAVE IPERMISO
ICODIGO OPERACION ICORREO
-- PERMITE REALIAR INGRESO,MODIFICACION Y ELIMINACION DE LA TABLA
USUARIOS
-- OPERACION = 1 = INGRESO
-- OPERACION = 2 = MODIFICACION
-- OPERACION = 3 = ELIMINACION
CREATE DEFINER=`root`@`localhost` PROCEDURE `MANTENIMIENTO_USUARIOS`(IN
INOMBRES VARCHAR(40),IN IAPELLIDOS VARCHAR(40),IN IUSUARIO
VARCHAR(40),IN ICELULAR VARCHAR(40),IN ICLAVE VARCHAR(40),IN IPERMISO
INTEGER,IN ICODIGO INTEGER,IN OPERACION INTEGER,ICORREO VARCHAR(40))
BEGIN
IF (OPERACION=1)THEN
    INSERT INTO sms.usuarios
(NOMBRES,APELLIDOS,USUARIO,CELULAR,CLAVE,PERMISO,CORREO)VALUES( INOMBRES
, IAPELLIDOS, IUSUARIO, ICELULAR, ICLAVE, IPERMISO, ICORREO) ;

```

```

ELSEIF (OPERACION=2)THEN
    UPDATE sms.usuarios SET
NOMBRES=INOMBRES,APELLIDOS=IAPELLIDOS,USUARIO=IUSUARIO,CLAVE=ICLAVE,CEL
ULAR=ICELULAR,PERMISO=IPERMISO,CORREO=ICORREO WHERE CODIGO=ICODIGO;
    ELSEIF (OPERACION=3)THEN
        DELETE FROM sms.usuarios WHERE CODIGO=ICODIGO;
END IF;
END $$
/*!50003 SET SESSION SQL_MODE=@TEMP_SQL_MODE */  $$
DELIMITER ;

```

8.9 PROCEDIMIENTO ALMACENADO “VISTA_MENSAJES”

Este procedimiento almacenado se encarga de mostrar todos los modulos a los que tiene acceso un usuario ademas de aquellos modulos a los que no tiene acceso.

```

-- =====
-- Author:          GRUPO # 13 PARALELO A REDES
-- Create date: 28/04/2009
-- Description:     Tabla
-- =====

-- Definition of procedure `VISTA_MENSAJES`

DROP PROCEDURE IF EXISTS `VISTA_MENSAJES`;
DELIMITER $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `VISTA_MENSAJES`(IESTADO
VARCHAR(1),F1 VARCHAR(30),F2 VARCHAR(30))
BEGIN
SELECT mensaje,dispositivos,fecha,funciones(3,dispositivos,'') as
nombre FROM MENSAJES WHERE tipo=IESTADO AND (FECHA>=F1 AND FECHA<=F2);
END $$
/*!50003 SET SESSION SQL_MODE=@TEMP_SQL_MODE */  $$

DELIMITER ;

```

8.10 PROCEDIMIENTO ALMACENADO “VISTA”

Este procedimiento almacenado se encarga de mostrar todos los modulos a los que tiene acceso un usuario ademas de aquellos modulos a los que no tiene acceso.

```
-- =====
-- Author:          GRUPO # 13 PARALELO A REDES
-- Create date: 28/04/2009
-- Description:     Tabla
-- =====

DELIMITER ;

-- Definition of procedure `VISTAS`
DROP PROCEDURE IF EXISTS `VISTAS`;
DELIMITER $$
-- PROCEDIMIENTO VISTAS
-- PARAMETROS OPERACION IUSUARIO
-- CONSULTA GENERALES Y PARA GENERAR REPORTE

CREATE DEFINER=`root`@`localhost` PROCEDURE `VISTAS` (IN OPERACION
INTEGER, IN IUSUARIO VARCHAR(40))
BEGIN
declare ICODIGO INTEGER;
declare dia integer;
IF(OPERACION=1)THEN
    SELECT * FROM USUARIOS WHERE usuario=IUSUARIO;
ELSEIF(OPERACION=2)THEN
    SELECT PERMISO INTO ICODIGO FROM USUARIOS WHERE usuario=IUSUARIO;
    SELECT * FROM PERMISOS WHERE CODIGO=ICODIGO;
ELSEIF(OPERACION=3)THEN
    SELECT * FROM DISPOSITIVOS;
ELSEIF(OPERACION=4)THEN
    SELECT * FROM PETICIONES;
END IF;
if(operacion=5)then
    select mensaje,dispositivos,fecha,funciones(3,dispositivos,'') as
nombre from mensajes where tipo=IUSUARIO;
end if;
if(operacion=6)then
    select * from borrados;
end if;
```

```

if(operacion=7)then
    select id,nombre,estado from publicidad;
end if;
if(operacion=8)then
    select recibidos as numero from publicidad where id=IUSUARIO;
end if;
if(operacion=9)then
    select ignorados as numero from publicidad where id=IUSUARIO;
end if;
if(operacion=10)then
    select * from publicidad where id=IUSUARIO;
end if;
if(operacion=11)then
    select DAYOFWEEK(curdate()) into dia ;
    if(dia=1)then
        select * from publicidad where estado=2 and d7=1 and nueva=0;
    end if;
    if(dia=2)then
        select * from publicidad where estado=2 and d1=1 and nueva=0;
    end if;
    if(dia=3)then
        select * from publicidad where estado=2 and d2=1 and nueva=0;
    end if;
    if(dia=4)then
        select * from publicidad where estado=2 and d3=1 and nueva=0;
    end if;
    if(dia=5)then
        select * from publicidad where estado=2 and d4=1 and nueva=0;
    end if;
    if(dia=6)then
        select * from publicidad where estado=2 and d5=1 and nueva=0;
    end if;
    if(dia=7)then
        select * from publicidad where estado=2 and d6=1 and nueva=0;
    end if;
end if;
if(operacion=12)then
    select count(*) as numero from mensajes where tipo=IUSUARIO;
end if;
if(operacion=13)then
    select rutas from nuevos;
end if;
if(operacion=14)then
    SELECT usuario,fecha,if(tipo=0,"INGRESO AL SISTEMA"," SALIO DEL
    SISTEMA")as tipo,funciones(1,usuario,'') as
    nombres,funciones(2,usuario,'') as apellidos FROM AUDITORIA;
end if;
if(operacion=15)then
    SELECT * FROM dispositivos order by numeros desc limit 0,9;

```



```

end if;
if(operacion=16)then
  select DAYOFWEEK(curdate()) into dia ;
  if(dia=1)then
    select * from publicidad where estado=2 and d7=1 and nueva=0;
  end if;
  if(dia=2)then
    select * from publicidad where estado=2 and d1=1 and nueva=0;
  end if;
  if(dia=3)then
    select * from publicidad where estado=2 and d2=1 and nueva=0;
  end if;
  if(dia=4)then
    select * from publicidad where estado=2 and d3=1 and nueva=0;
  end if;
  if(dia=5)then
    select * from publicidad where estado=2 and d4=1 and nueva=0;
  end if;
  if(dia=6)then
    select * from publicidad where estado=2 and d5=1 and nueva=0;
  end if;
  if(dia=7)then
    select * from publicidad where estado=2 and d6=1 and nueva=0;
  end if;
end if;
if(operacion=17)then
  select * from publicidad where estado=2;
end if;
END $$

DELIMITER ;

```