

CAPITULO 1

INTRODUCCION

1.1 Introducción

Los equipos informáticos que ofrecen servicios a múltiples usuarios son utilizados en todas las empresas que manejan información, por lo cual, existe diverso personal que tiene que acceder a dichos equipos para administrarlo o para realizar su trabajo cotidiano.

Se ha realizado una evaluación de varias empresas que posee personal en diversos sitios alejados geográficamente y surge la necesidad de ofrecerle el acceso a estos equipos de una manera confiable por lo que, nuestro grupo ofrece al mercado un sistema capaz de realizar una conexión a un servidor desde cualquier lugar del mundo, manteniendo estándares de seguridad y disponibilidad, por medio de uno de los recursos mas utilizados en la actualidad como lo es el Internet o red de redes.

1.1. Problemática a Resolver

Algunas veces es necesario administrar de forma remota un servidor; y para ello, debemos establecer una comunicación segura entre el servidor y el host desde el cual establecemos la conexión. Las sesiones de comunicación telnet no ofrecen mucha seguridad, ya que los datos viajan a través de la red sin ningún tipo de encriptamiento y son potencialmente susceptibles de ser interceptadas y aprovechadas por un atacante.

Cuando se realiza una conexión a un servidor remoto usando, por ejemplo, el comando telnet o ftp, el login(usuario) y password(contraseña) son transmitidos en la red de forma clara, lo cual representa un gran riesgo si llega a existir sobre la red un programa que capture la información, basándose en el modo promiscuo de las redes ethernet (comúnmente llamado sniffer), ocasionando obtener tanto el login como el password y pudiendo posteriormente irrumpir en el servidor con esta información.

1.2. Solución

La implementación del proyecto CLISSH, ofrece el servicio de conexión a un servidor remoto usando una interfaz Web, mejorando la comunicación entre el cliente y el servidor a través del protocolos seguros que ofrecen una garantía para todos los usuarios que acceden al sistema, de que los datos que intercambien con el servidor no van a ser interceptados o manipulados por terceros.

Secure Shell (SSH) es una solución basada en software que mantiene seguros los datos de la sesión en la red. SSH cifra todo el tráfico (incluidas las contraseñas) para eliminar de un modo efectivo las "escuchas", los secuestros de las conexiones y otros ataques a nivel de red. Además, SSH ofrece amplias posibilidades para la creación de túneles seguros, aparte de una variedad de métodos de autenticación.

CLISSH permitirá a los usuarios enviar comandos propios de Linux, los cuales se ejecutaran en el servidor, recibiendo la respuesta

correspondiente. Esta aplicación también ofrece la transferencia de archivos y la presentación de reportes sobre sesiones anteriores.

1.3. Misión

Llegar a constituirse en un sistema totalmente seguro, que pueda cumplir con los requerimientos principales del usuario en la administración remota.

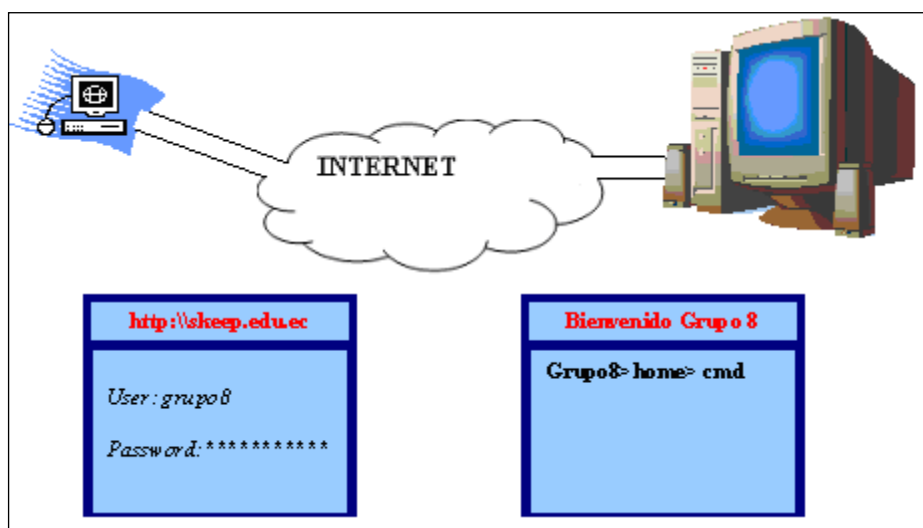
1.4. Visión

Haber consolidado los principios de seguridad, funcionalidad y disponibilidad del producto

1.5. Objetivo General

Crear un software de Terminal de conexión a host, que permita abrir una consola remota del servidor Linux por medio del protocolo SSH.

El usuario debe acceder a una página Web en la cual deberá digitar nombre de cuenta y su contraseña correspondiente, la cual deberá estar creada en los registros del servidor, automáticamente en su PC se muestra una consola en la cual podrá escribir los comandos de Linux a los cuales tenga privilegios.



1.6. Objetivo Especifico:

- ✓ Permitir que la maquina cliente inicie una conexión segura con la maquina de servidor.
- ✓ Establecer mecanismos y métodos de seguridad eficaces.
- ✓ Proporcionar Técnicas de protección.
- ✓ Desarrollar interfaces amigables para la conexión y el ingreso de comandos al servidor.
- ✓ Ofrecer reportes consistentes de acuerdo a los requerimientos de los usuarios.

CAPITULO 2

ANÁLISIS

DEL

SISTEMA

2.1. Análisis

2.1. Planificación y Desarrollo

2.1.1. Iteración

Esta primera iteración tendrá los siguientes objetivos:

1. Establecer el prototipo inicial del sitio Web Clissh
2. Establecer la arquitectura del sistema en un 100% con los requerimientos más importantes.
3. Establecer flujos de información y estandarización del sitio.
4. Establecer parámetros y protocolos de seguridad a implementarse en el sistema.

2.1.2. Definición de Roles y Tareas

Rol	Responsable
Administrador de Proyecto	Jonathan García
Analistas	Jonathan García, Patricia Zumba, Héctor Lara
Diseñadores	Patricia Zumba
Implementadores	Jonathan García, Patricia Zumba, Héctor Lara.
Ejecutores de Prueba	Héctor Lara, Patricia Zumba, Jonathan García.

El desarrollo del Sitio Web constará de las siguientes etapas:

2.1.3. Análisis – Requerimientos

- Esta etapa tendrá una duración de cuatro semanas.
- Los responsables de esta disciplina serán todos los miembros del equipo.

Actividades:

2.1.4. Casos de Uso Requerimientos

- Tendrá una duración de dos semanas.
- Participarán todos los miembros del equipo.
- Esta etapa tiene el objetivo de definir todos los requerimientos del usuario y definir las funcionalidades de Todo el Sistema.

2.1.5. Análisis

- Esta etapa tiene una duración de dos semanas
- Participarán todos los miembros del equipo
- El objetivo es desarrollar todos los documentos donde se detallen las necesidades del usuario y las soluciones a cada una de ellas.

2.1.6. Diseño

- Esta etapa tiene una planificación inicial de cuatro semanas.
- Participarán todos los miembros del equipo.
- El objetivo de esta etapa es definir arquitectura del sitio, así como la estrategia de marketing a seguir.

Actividades:

2.1.6.1. Diseño de Objetos

- Esta etapa tiene una planificación inicial de una semana.
- Las actividades a realizarse serán:
 - Desarrollo de diagramas de Secuencia
 - Desarrollo de diagramas de clases

2.1.6.2. Diseño de Bases de Datos

- Esta etapa tiene una planificación inicial de una semana.
- Desarrollo de Diagramas de Bases de Datos
- Configuraciones necesarias en el entorno de base de datos.
- Diseño de las clases a utilizar para la manipulación de datos.

2.1.6.3. Diseño de Páginas

- Definición de estándares de objetos
- Diseño inicial de las páginas.
- Diseño de navegación del sitio

2.1.7. Implementación

- Se desarrollarán todas las páginas Web diseñadas, así como todos los objetos necesarios para las funcionalidades establecidas
- Se realizarán dos fases de implementación.
- La primera fase corresponde a la implementación inicial de todas las funcionalidades.
- La segunda fase corresponde a la corrección de errores resultado de las primeras pruebas.

- Participarán todos los miembros del equipo.

2.1.7.1. Implementación Inicial

- Esta etapa tiene una planificación inicial de diecisiete semanas.
- Participarán todos los miembros del equipo.
- Se crearán todas las funcionalidades definidas en el diseño y los casos de uso.
- Creación de las bases de datos y procedimientos necesarios para la interacción entre la base de datos y la aplicación.
- Creación de las clases requeridas para el proyecto.
- Creación de las páginas .jsp establecidas en el diseño.
- Creación de los certificados necesarios para el sitio.
- Integración de los componentes en un ente único y operativo.

2.1.7.2. Implementación Final

- Esta etapa tiene una planificación inicial de dos semanas.
- Participarán todos los miembros del equipo.
- Se corregirá el 100% de defectos rebote de Control de Calidad.
- Esta etapa se considera la mas riesgosa del proyecto, puesto que puede estar expuesta a mayores problemas y a posibles holguras.

2.1.8. Pruebas

- Esta etapa tiene una planificación inicial de cuatro semanas. Entre los dos ciclos de ejecución.
- Los responsables de esta etapa serán todos los miembros del equipo
- Se realizarán pruebas de todas las transacciones del Sistema.

2.1.9. Certificación

- Este ciclo tendrá una duración de una semana.
- El objetivo es certificar los defectos que se reportaron en las diferentes pruebas.
- Se espera un porcentaje de errores del 5%.
- Todos los defectos que se presenten en esta etapa serán tomados como críticos por desarrollo

2.2 Descripción Funcional

El objetivo principal de este sistema “CLISSH” es mantener una comunicación lo más segura posible mediante el uso del protocolo SSH para que todas las personas que van a usar nuestro sitio Web puedan tener la tranquilidad que los datos intercambiados son absolutamente confiables y no caerán en poder de usuarios maliciosos que puedan afectar al desarrollo de la empresa.

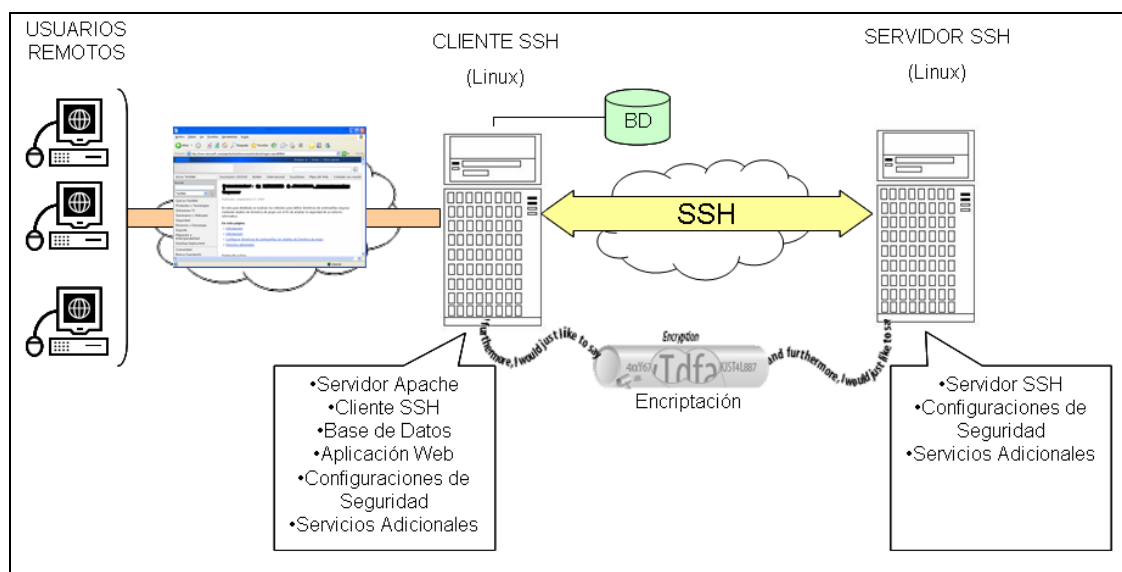
CLISSH se puede convertir en la herramienta básica de trabajo de cualquier empresa que tengan personal geográficamente distante y que tienen, o quieren tener, un control centralizado en un servidor trabajando en una plataforma Linux, los cuales realizarán sus

actividades desde algún lugar del mundo de una manera amigable y segura.

El usuario debe acceder a una página Web en la cual deberá digitar nombre de cuenta y su contraseña correspondiente, la cual deberá estar creada en los registros del servidor, automáticamente en su PC se muestra una consola en la cual podrá escribir los comandos de Linux a los cuales tenga privilegios.

La conexión al servidor Linux se lo realizará mediante un Servidor Web el cual canalizará las peticiones de los clientes hacia el servidor Linux, al mismo tiempo responderá a dichos clientes.

Este servidor debe tener la capacidad de satisfacer las necesidades de la empresa y contribuir al normal desenvolvimiento de las actividades requeridas.



Tipos de usuarios

Los usuarios tienen diferentes jerarquías y por consiguiente diferentes permisos para poder realizar ciertas operaciones en el servidor, a saber, existen los siguientes usuarios:

- ✓ **Súper usuario:** El usuario administrador (root) tendrá todos los permisos dentro del sistema, podrá consultar datos del sistema, del servidor, revisar la bitácora (reporte) de accesos al sistema, crear usuarios para nuevos clientes, eliminar usuarios existentes, darle privilegios a ciertos usuarios.

- ✓ Usuario especializado: Es aquel usuario que está dedicado a tareas especiales dentro de la empresa, es decir, tiene a su cargo actividades propias de cada departamento y estos podrán acceder a un espacio especial del servidor para realizar sus labores.

Una vez enviada la información correspondiente a la petición de realizar la conexión SSH con el servidor, el sistema debe validar dichos datos y, de ser correctos, debe crearle un espacio de sesión para que el usuario trabaje con los recursos del servidor. De ser incorrectos los datos enviados, el sistema deberá informar sobre los intentos fallidos de inicio de sesión y almacenar dichos intentos en un log los datos erróneos para mantener un control especial y de esta manera prevenir futuros ataques.

Procesos Principales

✓ Proceso de verificación de Usuario

Este es el proceso que se encargará de verificar si el usuario que esta realizando la petición de conexión esta habilitado para ingresar al sistema comparando los datos de usuario y contraseña receptados por el servidor Web.

✓ Recepción de peticiones.

Las peticiones realizadas por nuestros clientes deberán ser canalizadas por este proceso hacia el servidor el cual debe darle las respuestas correspondientes.

✓ Guardar datos

Todas las acciones que realicen los usuarios deberán estar registradas para saber exactamente las actividades que desempeñan dentro del sistema, por lo que este proceso guardara los comandos utilizados por el usuario con sus respectivas respuestas en el lugar que la administración del sistema crea conveniente para dichos archivos.

✓ **Encriptar- Desencriptar**

Este proceso es el que se va a encargar de mantener la seguridad del sistema, de acuerdo con los algoritmos utilizados por el protocolo SSH.

Este proceso encriptará los datos que serán enviados al servidor y desencriptará los datos que el servidor nos envíe.

Los siguientes procesos solo podrán ser ejecutados por el súper usuario.

✓ **Crear Cuentas**

Creara la cuenta del usuario en el servidor y guardara los datos necesarios del cliente en una BD para que después pueda iniciar una sesión.

✓ **Modificar Cuentas**

Puede modificar ciertos datos de la cuenta del usuario, estos cambios solo afectaran a la base de datos, mas no a la cuenta que se encuentra en el servidor.

2.3 Diagrama de Casos de Uso

2.3.1. Elementos:

Actores.- Es una identidad externa al sistema que realiza algún tipo de interacción en el mismo.

Casos de Uso.- Un caso de uso es una descripción de la secuencia de interacciones que se producen entre un actor y el sistema.

Relaciones entre casos de uso.- Entre dos casos de uso puede haber las siguientes relaciones:

Extiende: Cuando un caso de uso especializa a otro extendiendo su funcionalidad.

Usa: Cuando un caso de uso utiliza a otro.

2.3.1.1 Actores

Los actores que intervienen en nuestro Sistema serán los siguientes:

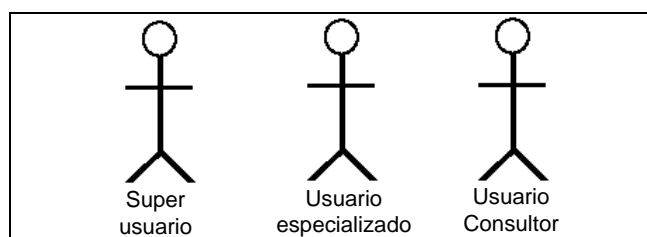


Figura 2.2. *Usuarios del Sistema*

El súper usuario como el usuario especializado serán los únicos que interactúe con el sistema de forma real, solo ellos poseen una cuenta para poder ingresar al sistema, mientras que el usuario consultor solo podrá navegar en las paginas de información general, más no en los principales servicios del sistema ó de aplicación.

2.3.1.2 Casos de Uso

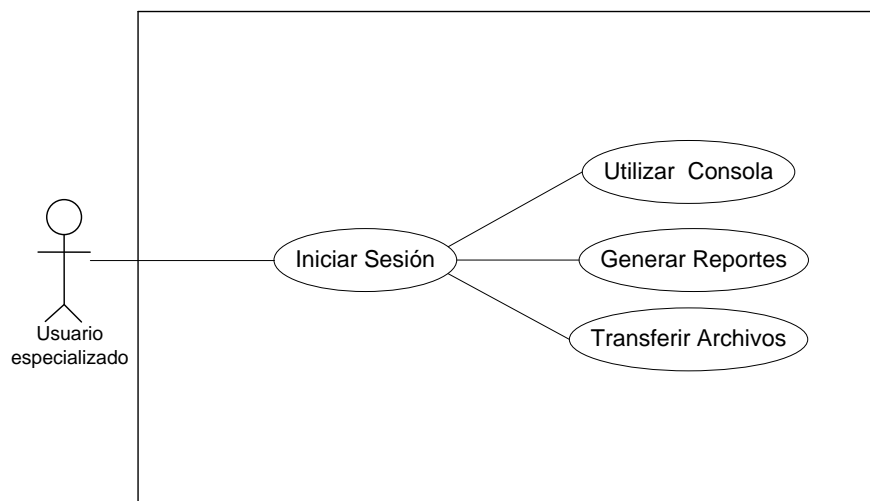


Figura 2.3. Diagrama de Casos de Uso – Usuario Especializado

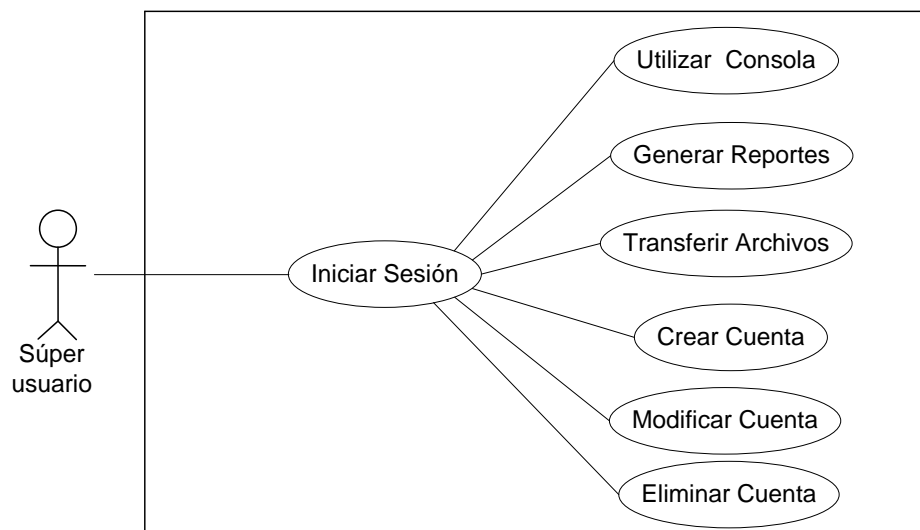


Figura 2.4. Diagrama de Casos de Uso – Súper Usuario

Iniciar Sesión

Inicio

Este caso de uso inicia cuando el usuario especializado o el súper usuario ingresa a la WebApp y digita su nombre de usuario y contraseña en la página de inicio de sesión

Fin del caso de uso

Utilizar Consola

Inicio

Este caso de uso inicia después de establecer una sesión con el servidor y se selecciona la opción de consola para enviar comandos propios de Linux al servidor y recibir su respuesta.

Fin del caso de uso

Generar Reportes

Inicio

Este caso de uso comienza después de haber establecido la sesión, en el cual podrá el usuario obtener un reporte de sus sesiones anteriores. Se puede escoger distintos criterios de selección para el reporte, los cuales son:

- ✓ Por rango fechas
- ✓ Por comandos utilizados

Fin del caso de uso

Transferir Archivos

Inicio

Este caso de uso comienza después de haber establecido la sesión, el usuario deberá establecer el origen y destino del archivo a transferir.

Fin del caso de uso

Crear Cuenta

Inicio

Este caso de uso comienza después de haber establecido la sesión, el usuario para crear una cuenta deberá llenar los datos necesarios del cliente y asignarle un nombre de usuario y su respectiva contraseña.

Fin del caso de uso

Modificar Cuenta

Inicio

Este caso de uso comienza después de haber establecido la sesión, el usuario deberá seleccionar la cuenta del usuario y modificar los datos que solo le sean permitidos.

Fin del caso de uso

Todos los casos de uso finalizan cuando el usuario cierra la sesión.

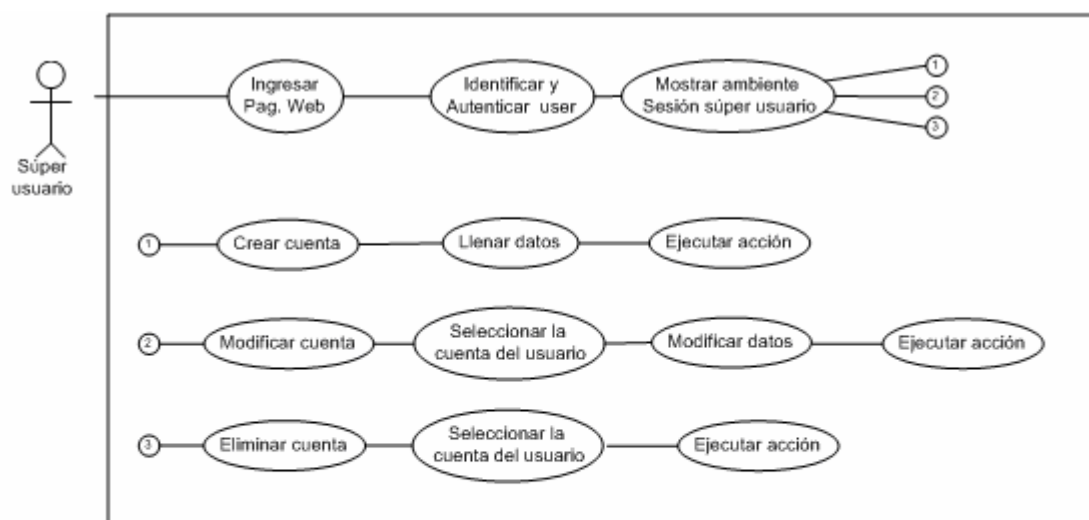


Figura 2.5. Diagrama de Casos de Uso General - Súper Usuario

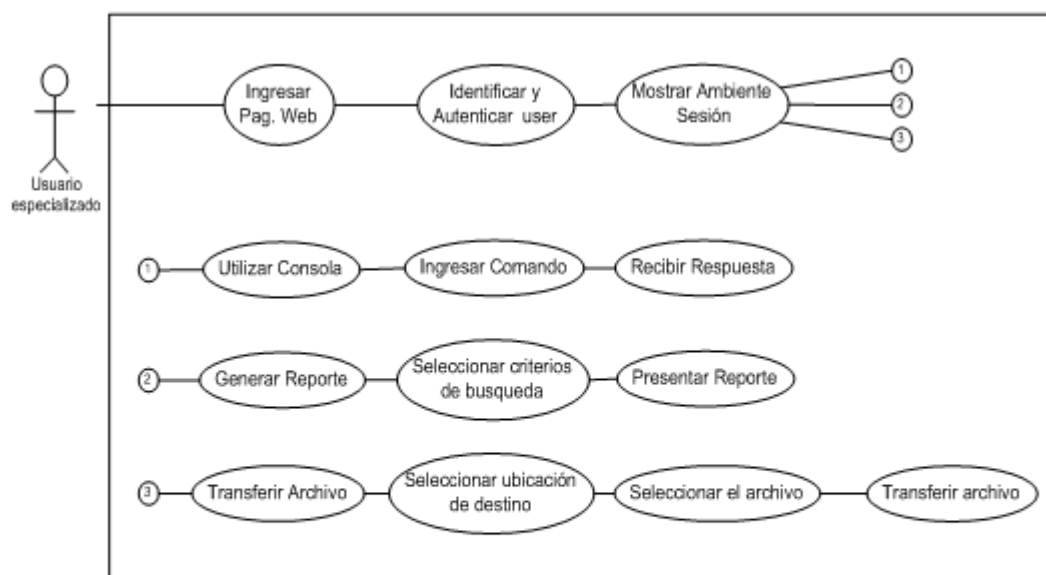


Figura 2.6. Diagrama de Casos de Uso General – Usuario Especializado

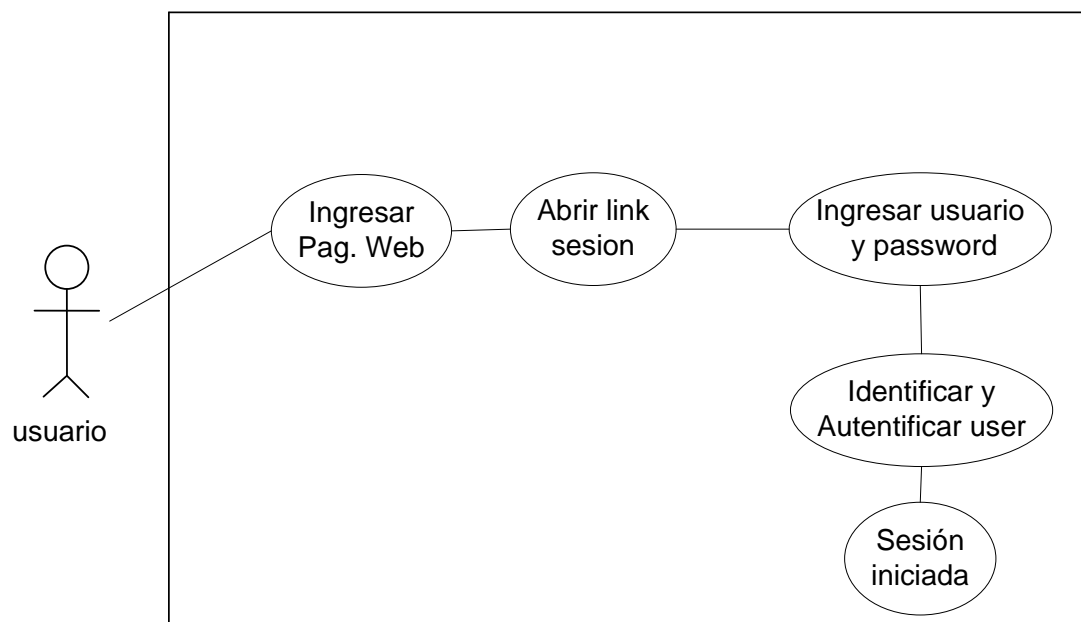


Figura 2.7. Diagrama de Casos de Uso – Iniciar Sesión



Figura 2.8. Diagrama de Casos de Uso – Consola

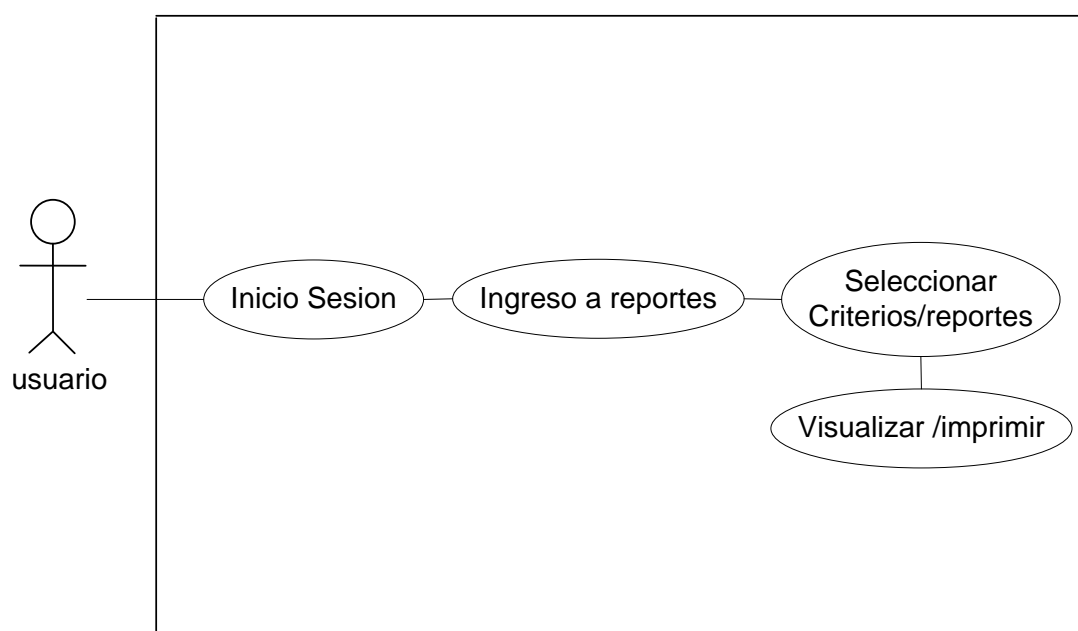


Figura 2.8. Diagrama de Casos de Uso – Reportes

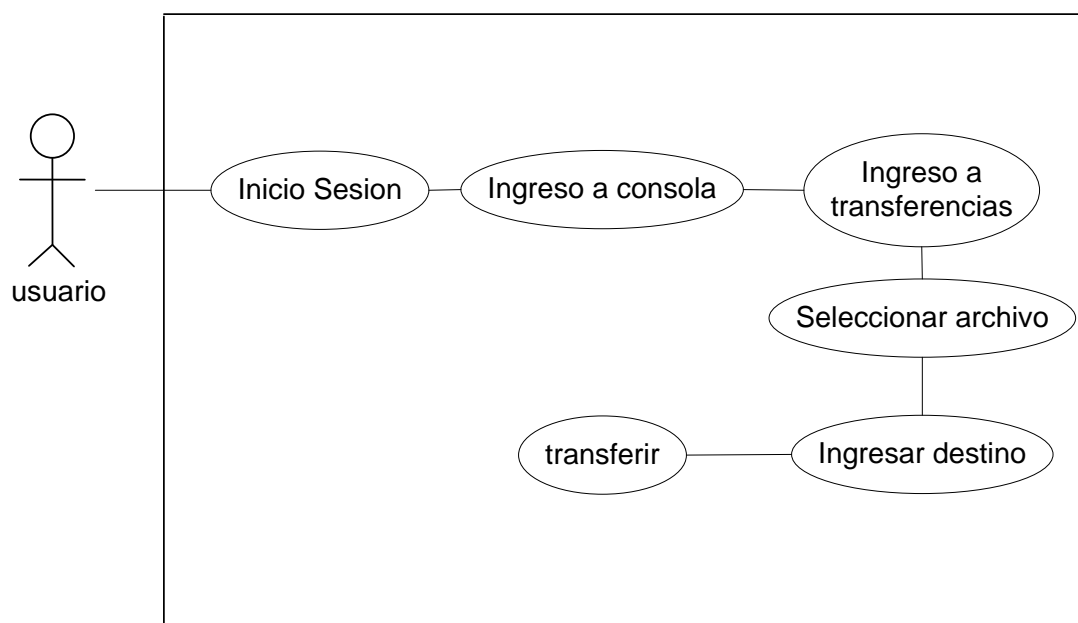


Figura 2.9. Diagrama de Casos de Uso – Transferencia de archivos

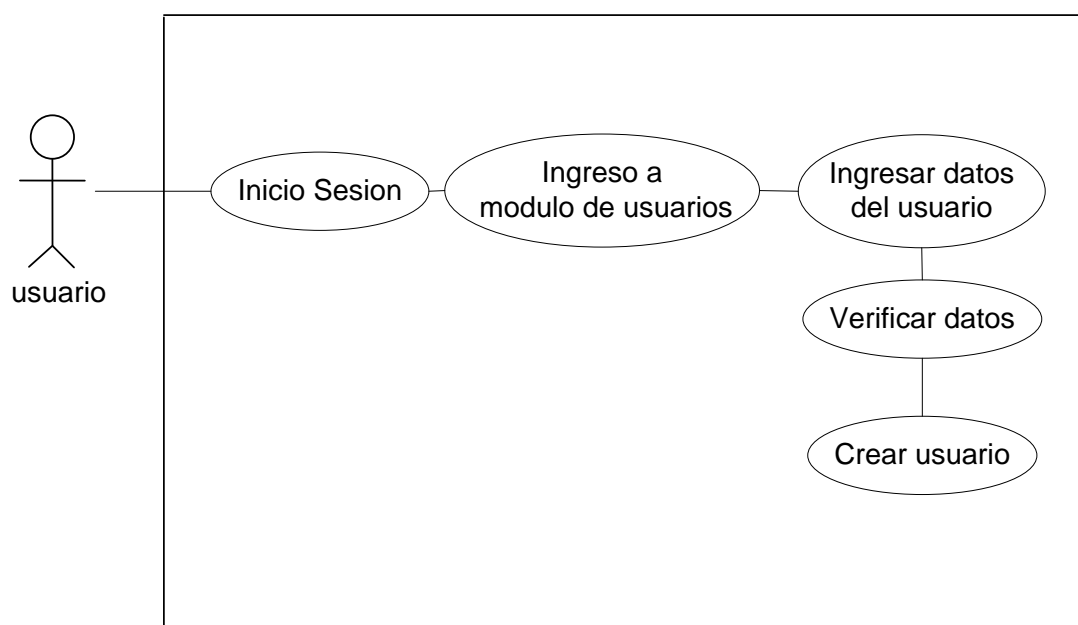


Figura 2.10. Diagrama de Casos de Uso – Crear cuentas de usuarios

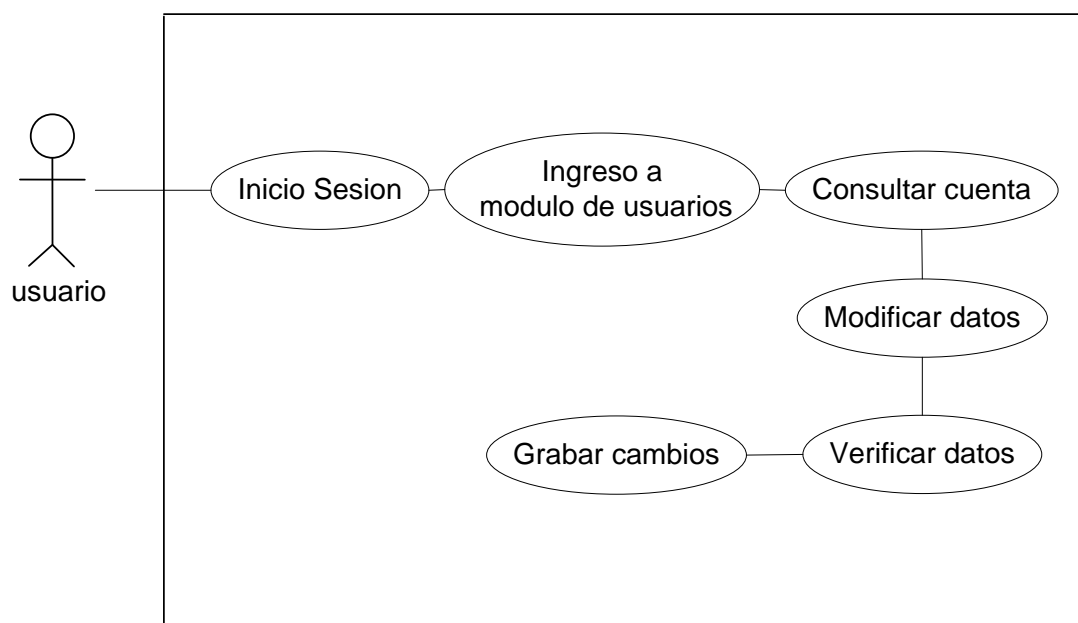


Figura 2.11. Diagrama de Casos de Uso – Modificar Cuentas de Usuario

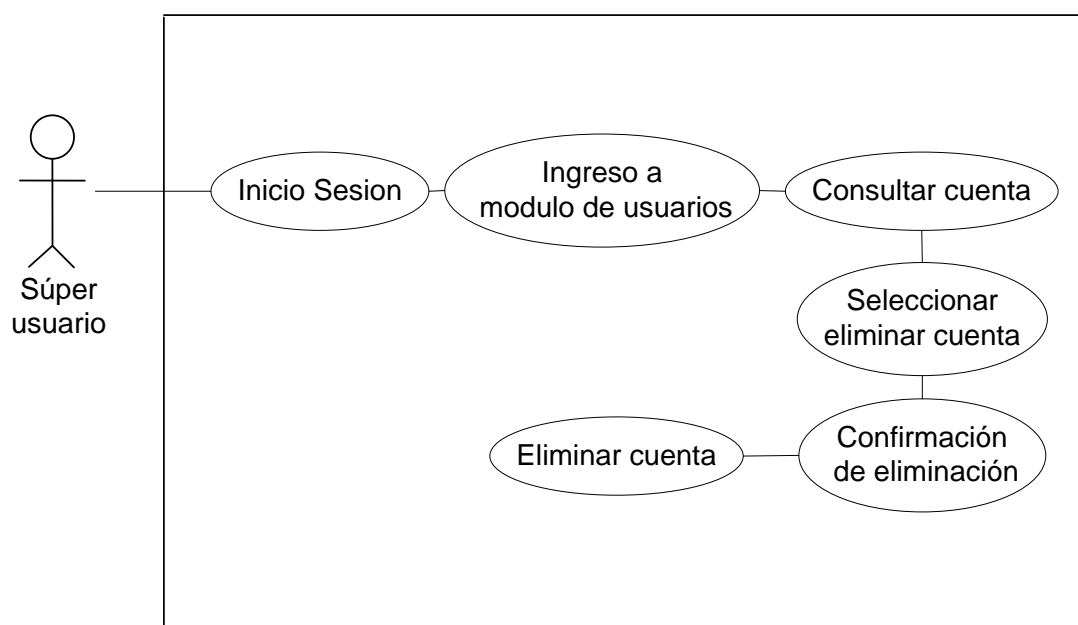


Figura 2.12. Diagrama de Casos de Uso – Eliminar Cuentas de Usuario

2.4 Diagrama de Componentes

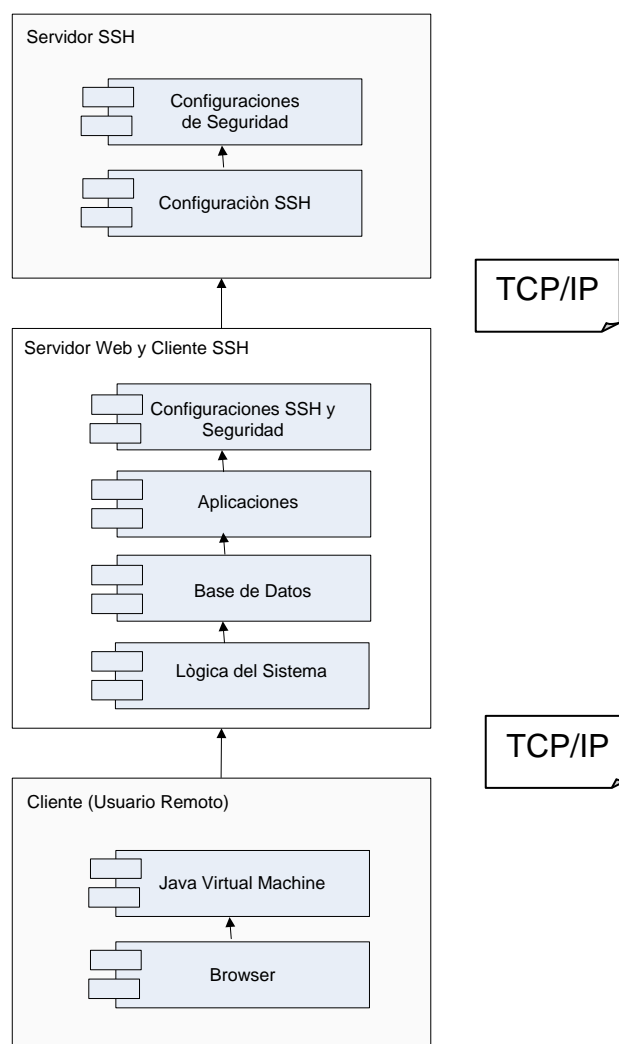


Figura 2.13. *Diagrama de Componentes*

Servidor SSH.

Es el equipo al cual se van a conectar los usuarios del sistema y en el que se van a ejecutar todos los comandos que se envíe a través de la consola del software, va a tener instalado un Sistema Operativo Multiusuario como lo es Linux con su distribución Fedora Core 4.

- **Capa Servidor SSH**

En esta capa correrá las configuraciones de Servidor del protocolo SSH, la cual contendrá los parámetros de puertos, ficheros, librerías, entre otros que varios a permitir levantar la comunicación con el cliente SSH y los usuarios remotos.

- **Capa de Configuraciones de Seguridades**

En esta capa se efectuará todas las configuraciones de seguridad necesarias para garantizar el correcto funcionamiento del sistema como son:

- ✓ Comunicación con el cliente SSH en el cual se va a seguir minuciosamente todos los requerimientos que exige el protocolo como son las seguridades de autenticación, los algoritmos de encriptación, la seguridad de los puertos de comunicación.

- ✓ La información de los usuarios y archivos deben estar protegidos en este equipo de manera que no sea vulnerada por accesos indebidos.

Servidor Web y Cliente SSH

- **Servidor Web**

La capa de servidor Web es el encargado de gestionar todos los requerimientos del sitio Web de parte de los usuarios.

Este servidor Web será el encargado de direccionar correctamente las peticiones y las respuestas generadas.

- **Cliente SSH**

La capa de Cliente SSH es el encargado de realizar la comunicación con el servidor SSH, aplicando los mecanismos que detalla el protocolo y será el responsable de mantener los datos enviados y recibidos de manera segura mediante la encriptación de datos, comunicación confiable, etc.

- **Configuraciones SSH y Seguridad.**

En esta capa se detallan todos los aspectos concernientes a las configuraciones propias del protocolo como son los puertos a utilizar, los algoritmos que se van a implementar,

los estándares de comunicación y las rutas permitidas para los accesos.

- **Aplicaciones**

Esta capa está compuesta por todos los elementos del aplicativo: paginas Web (HTLM), clases, librerías, configuraciones de plataformas y protocolos, entre otros.

- **Base de Datos.**

La base de datos estará alojada en esta capa y será la encargada de almacenar toda la información intercambiada entre los sistemas, para lo cual se define que el software de base de datos relacional que se va a utilizar será PostgreSQL 8.2.

Este motor de base de datos será el encargado de gestionar todos los requerimientos que se presenten a los datos almacenados mediante consultas que se requieran para realizar los reportes que se necesiten.

- **Lógica del sistema**

El sistema de manera física va a ir implementado en esta capa y allí se va a gestionar todas las opciones que el software va a permitir realizar a los usuarios, cabe recalcar

que esta lógica del sistema deberá estar protegida para que no se pueda alterar el funcionamiento del sistema.

La aplicación va a estar desarrollada en el lenguaje de programación Java, haciendo uso de la librería applets la cual nos va a ayudar a descargar en el cliente una instancia de los objetos utilizados en el sistema y permite una conexión rápida y confiable.

Usuario Remoto

Esta es la capa que van a utilizar directamente los usuarios clientes de la aplicación y en donde estos podrán navegar en todas las opciones a las que puedan acceder según los privilegios que tengan dentro del sistema.

En esta capa los equipos que acceden al sistema deben de tener ciertos elementos necesarios para realizar las actividades previstas y estos son:

- **Java Virtual Machine**

La maquina virtual de java es un componente necesario para la interpretación de de las respuestas enviadas al equipo ya que estas vendrán al cliente como código html y/o código Java.

- **Browser.**

El navegador o browser es el componente requerido y necesario para la comunicación a través de Internet, pero no hay ninguna restricción del software a utilizar.

2.5 Diagramas de Interacción

2.5.1 Diagramas de Secuencia

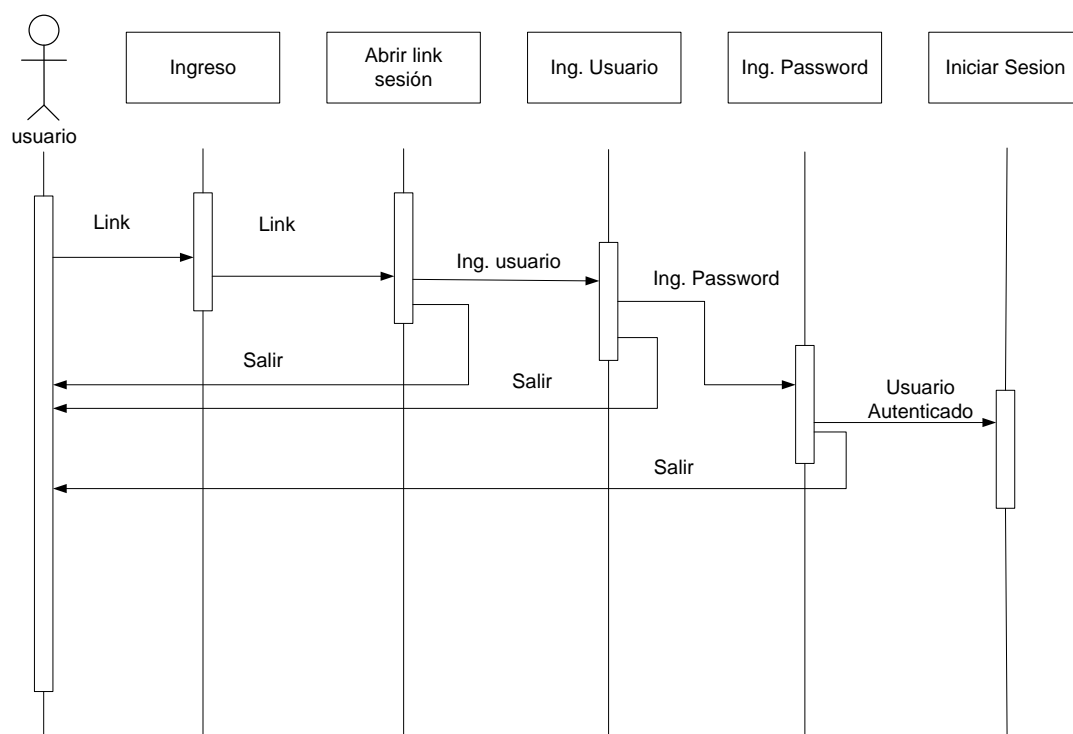


Figura 2.14. Diagrama de Secuencia – Iniciar Sesión

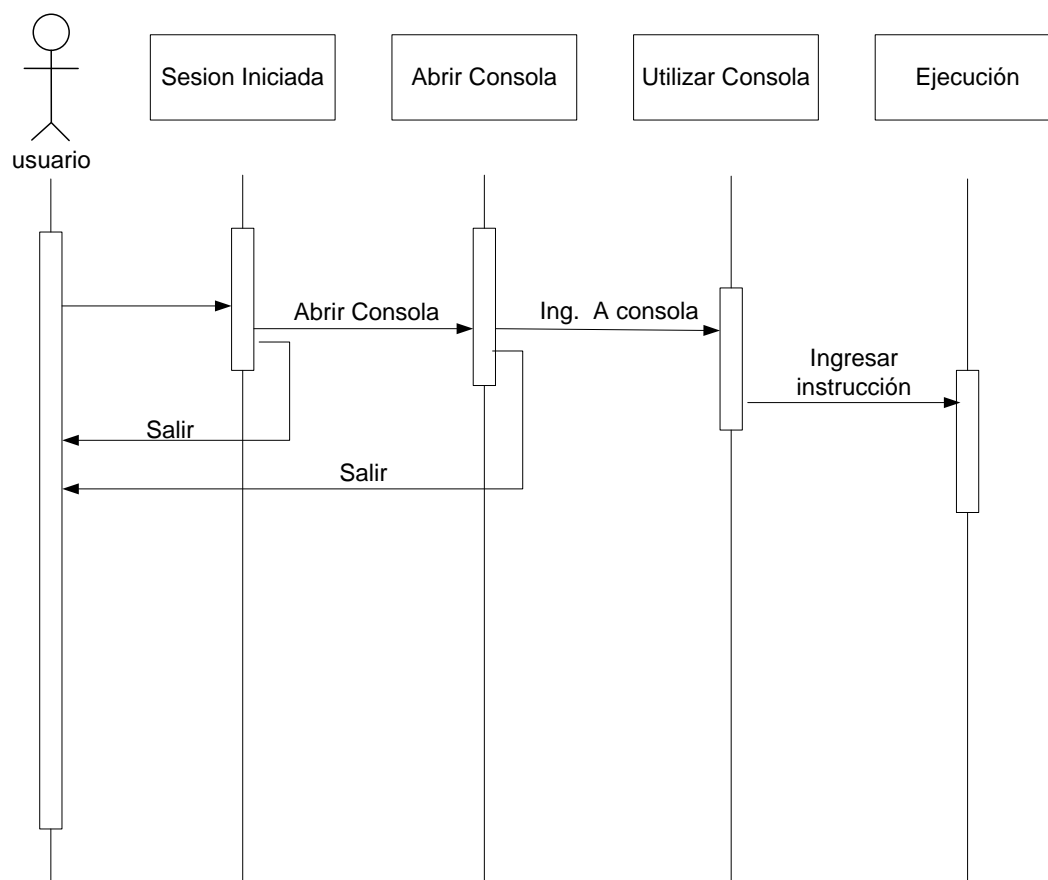


Figura 2.15. Diagrama de Secuencia – Utilizar Consola

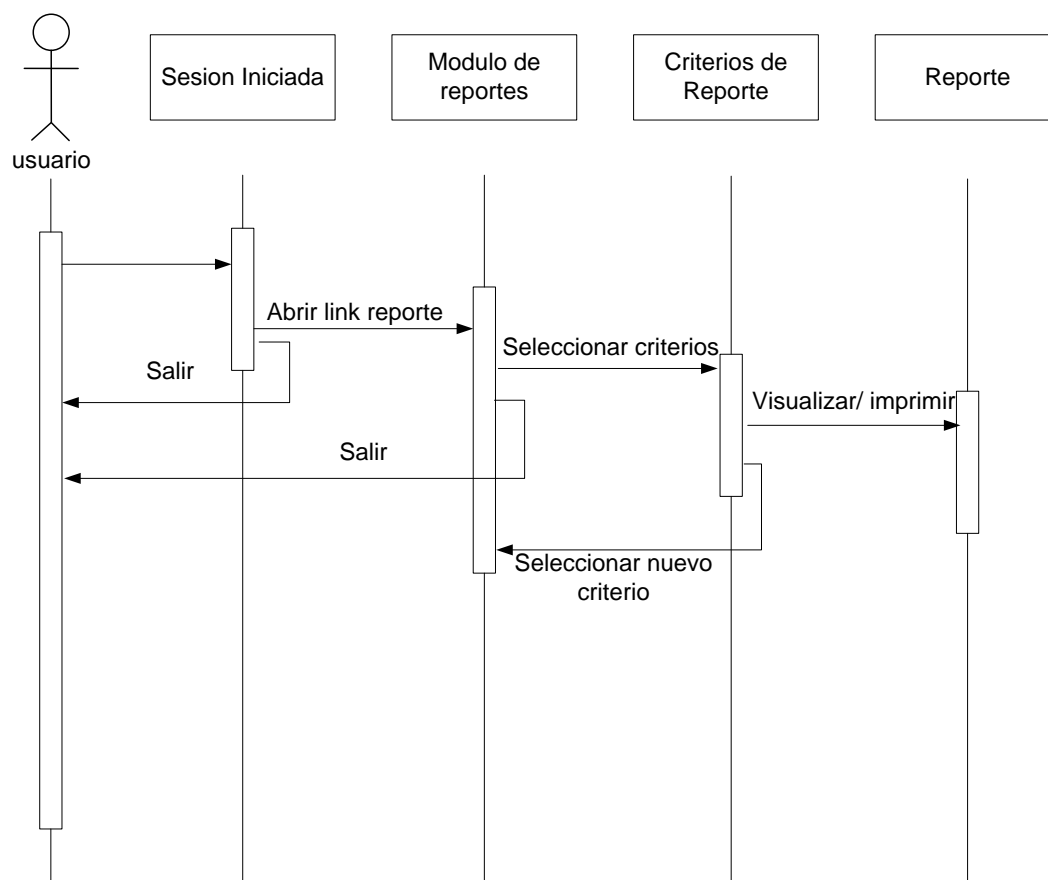


Figura 2.16. Diagrama de Secuencia – Reportes

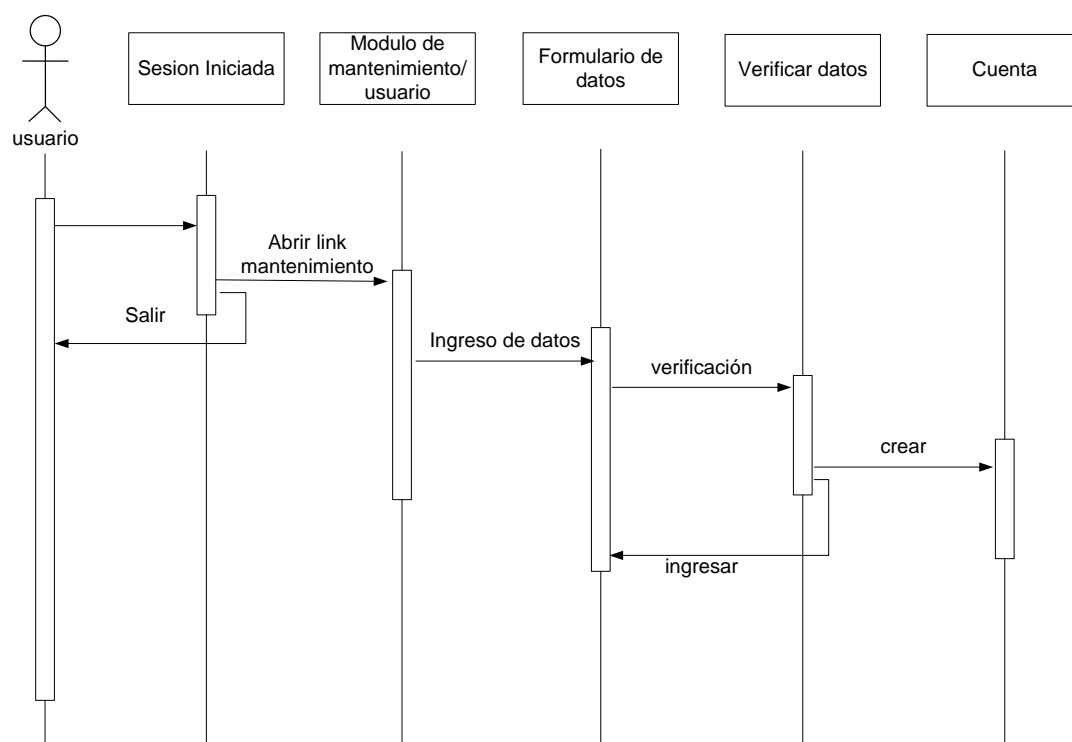


Figura 2.17. Diagrama de Secuencia – Crear cuentas de usuario

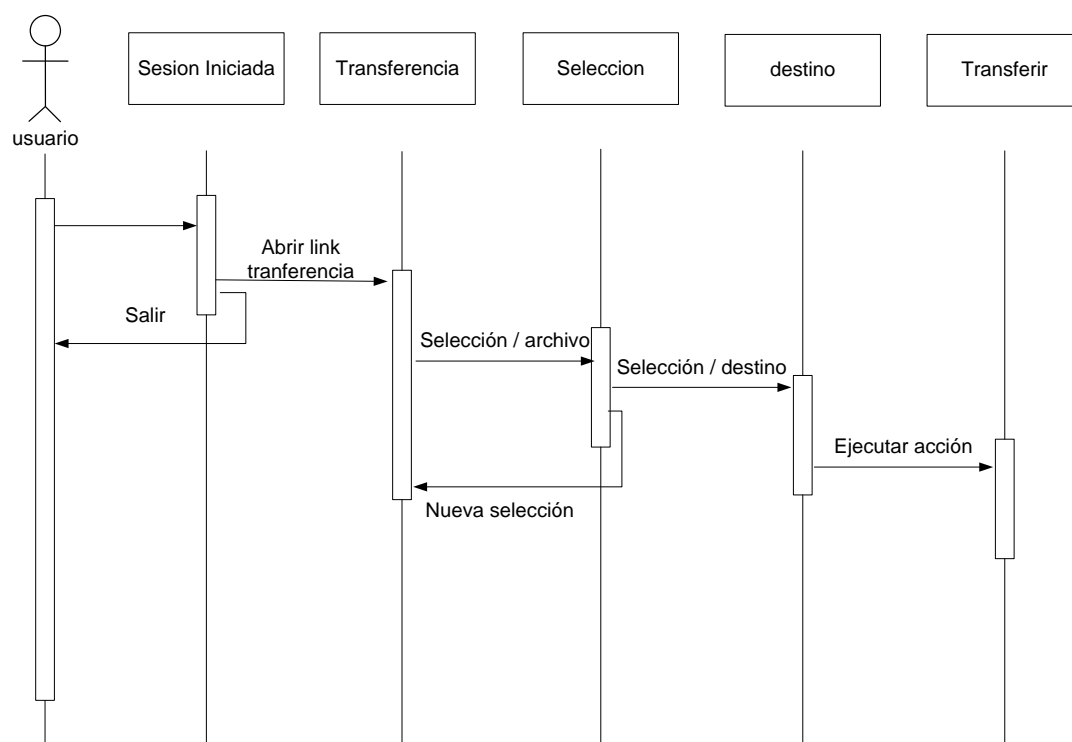


Figura 2.18. Diagrama de Secuencia – Transferencia de Archivos

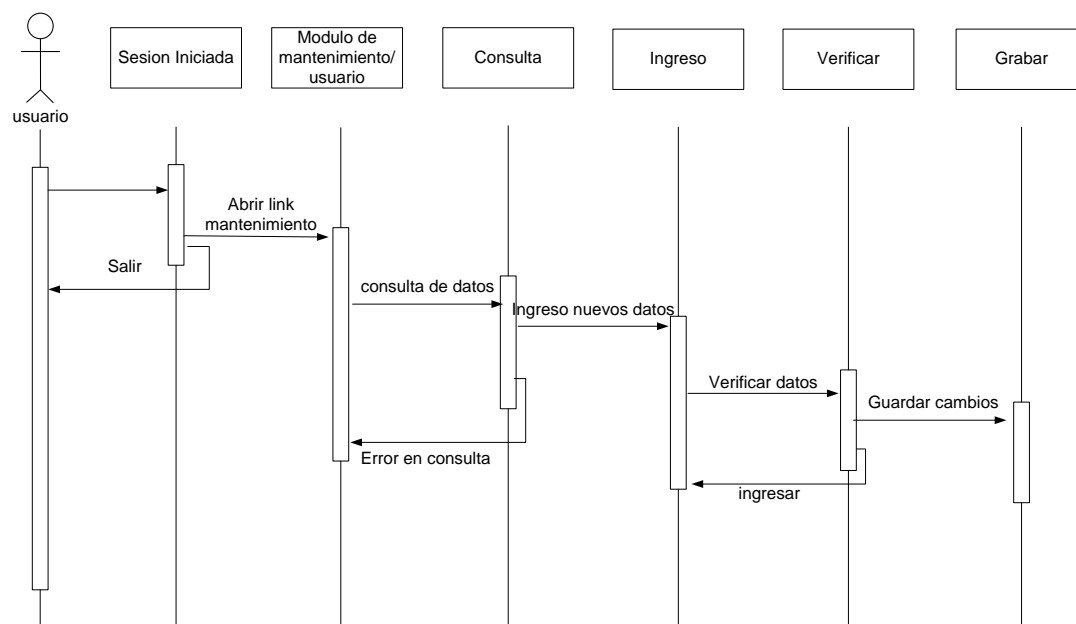


Figura 2.19. Diagrama de Secuencia – Modificación cuentas de usuario

2.5.2 Diagramas de Estado

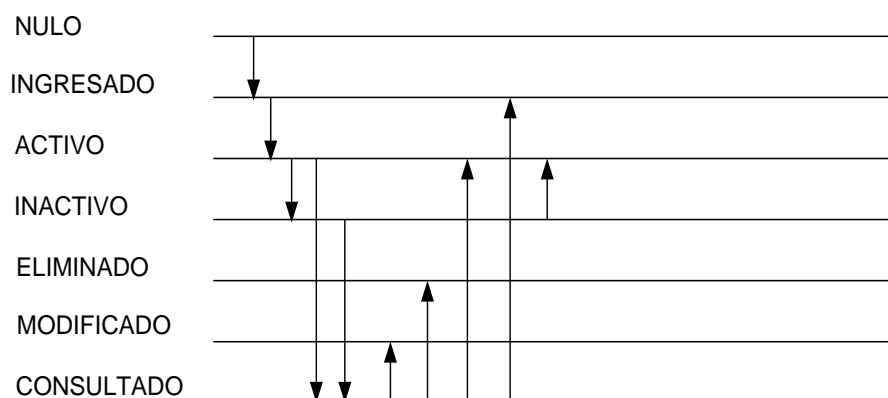


Figura 2.21. *Diagrama de Estado - Usuario*

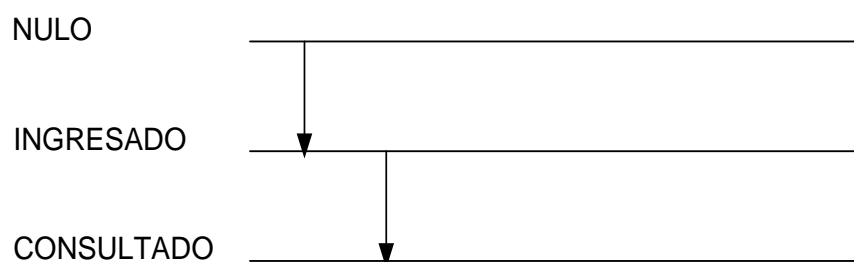


Figura 2.22. *Diagrama de Estado - Sesión*

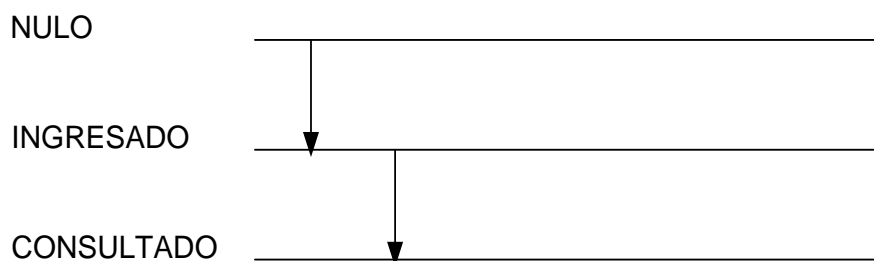


Figura 2.23. *Diagrama de Estado - Detalle de Sesión*

2.5.3 Diagrama de Flujo de Objetos

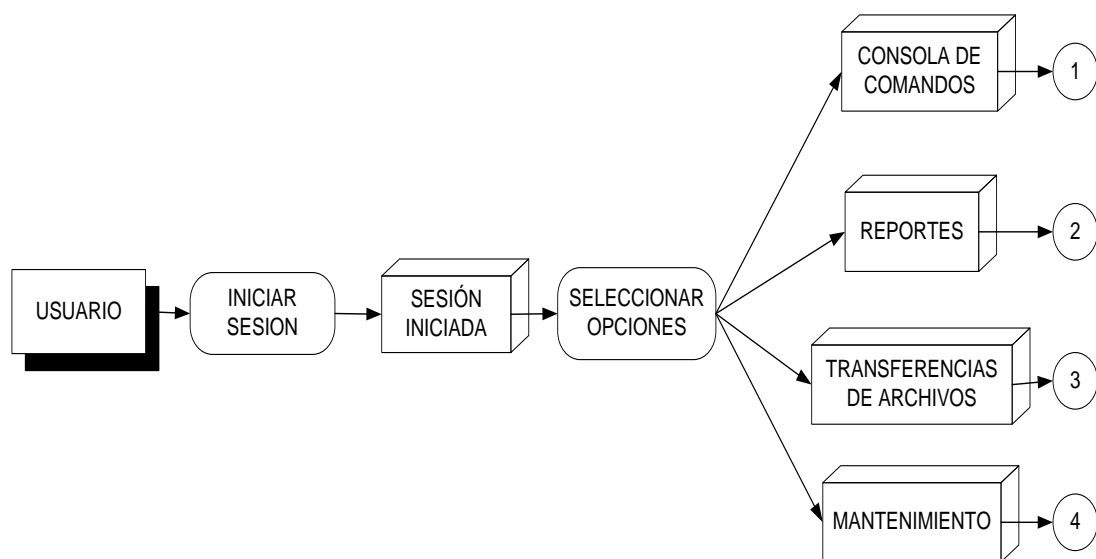


Figura 2.24. Diagrama Flujo de Objetos (1ª parte)

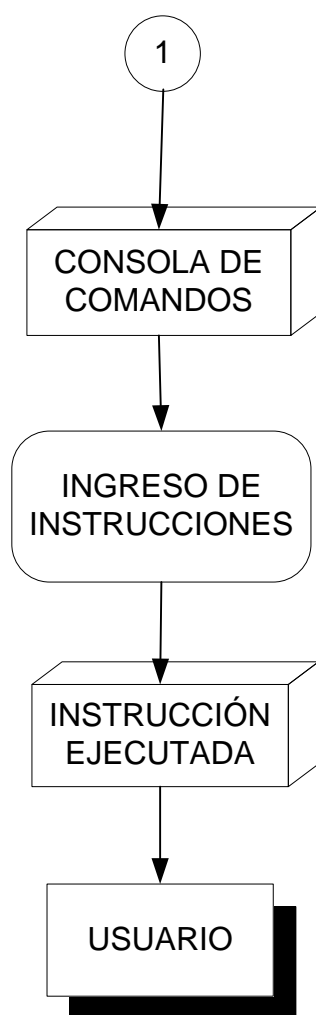


Figura 2.25. *Diagrama Flujo de Objetos (Consola)*

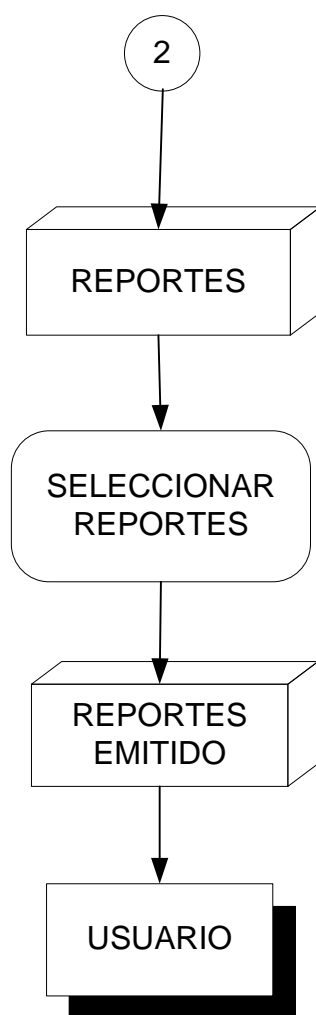


Figura 2.26. *Diagrama Flujo de Objetos (Reportes)*



Figura 2.27. *Diagrama Flujo de Objetos (Transferencia de Archivos)*

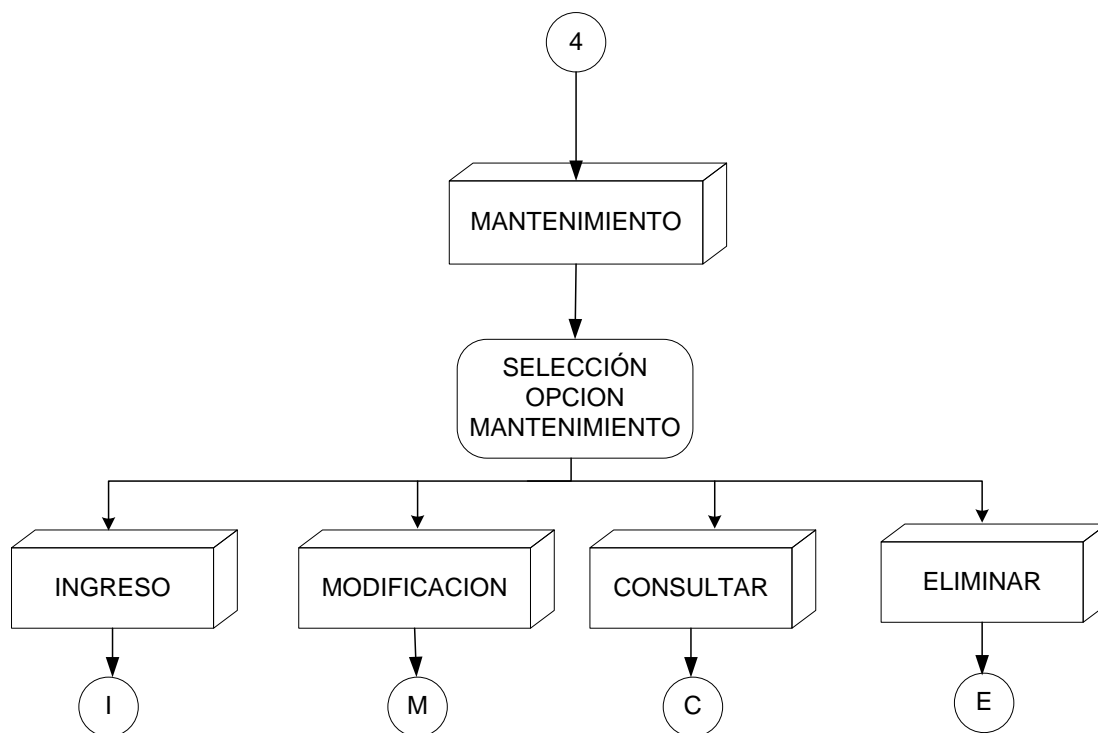


Figura 2.28. *Diagrama Flujo de Objetos (Mantenimiento de usuarios)*

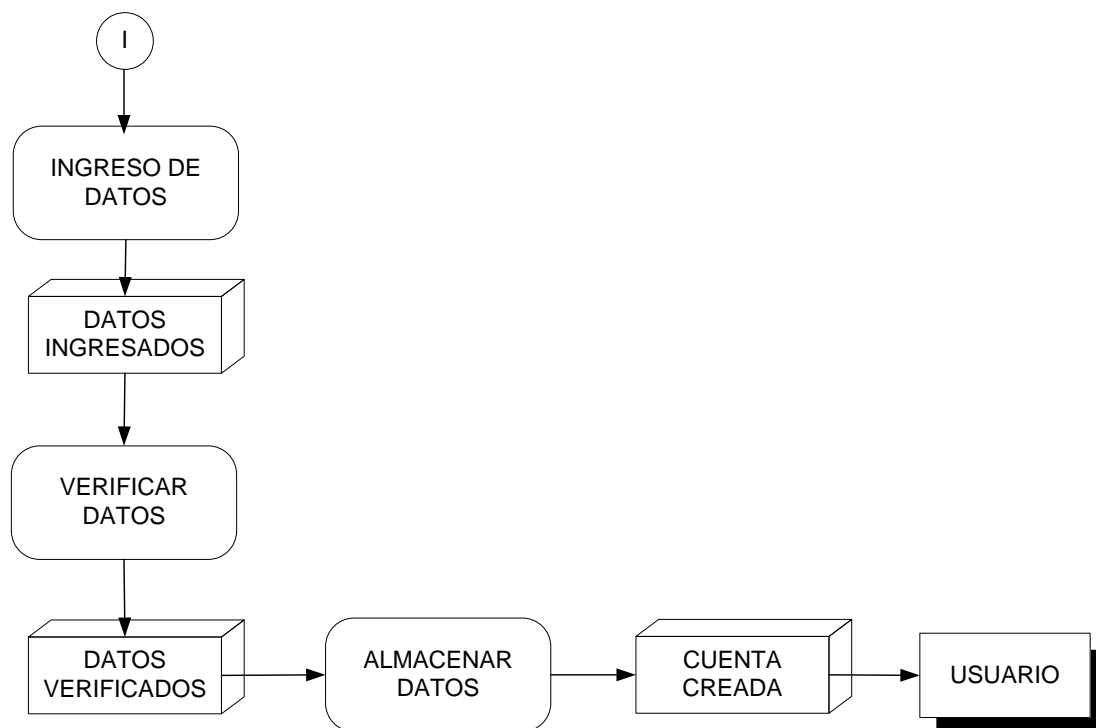


Figura 2.29 Diagrama Flujo de Objetos (Crear cuenta de usuario)

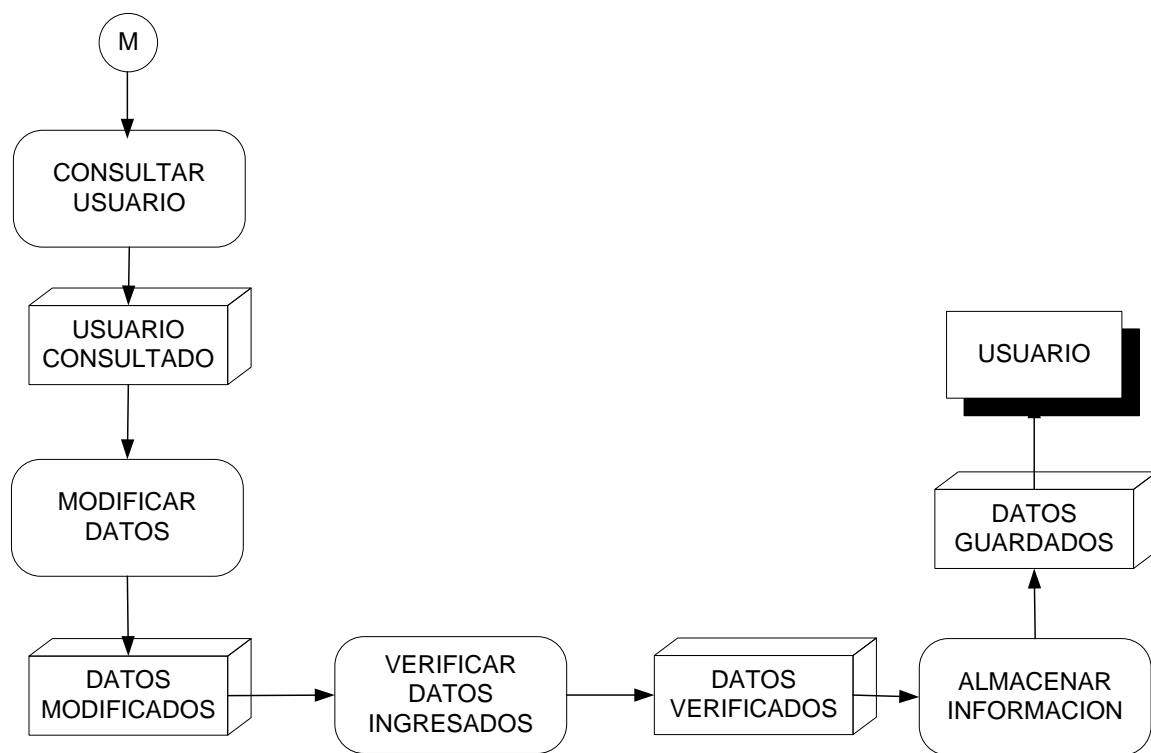


Figura 2.30. *Diagrama Flujo de Objetos (Modificar Usuario)*

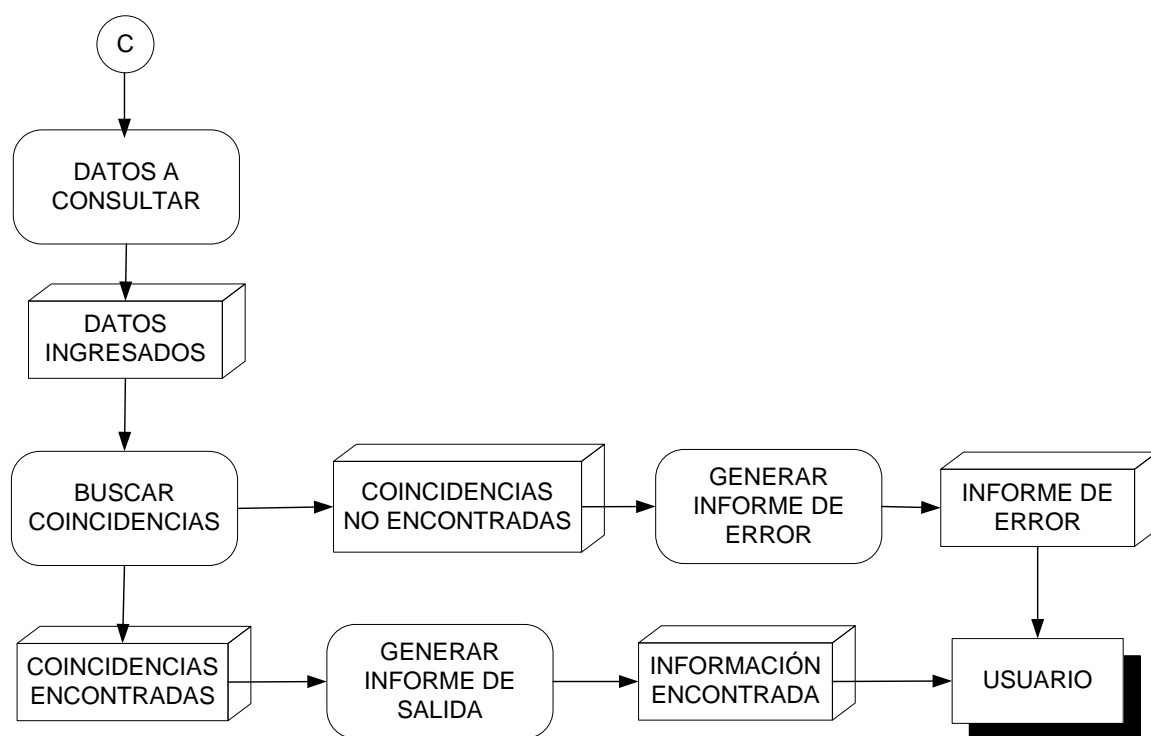


Figura 2.31. Diagrama Flujo de Objetos (Consultar Usuario)

1.7. Metodología y Modelo de Desarrollo de Proyecto

A medida que la evolución de las aplicaciones Web pasa de utilizar recursos estáticos de información controlada por el contenido a utilizar entornos de aplicaciones dinámicas controladas por el usuario, cada vez es más importante la necesidad de aplicar una gestión sólida y unos principios de ingeniería. Para conseguir esto es necesario desarrollar un marco de trabajo que acompañe a un modelo de proceso eficaz popularizado por las actividades del marco de trabajo y por las tareas de ingeniería.

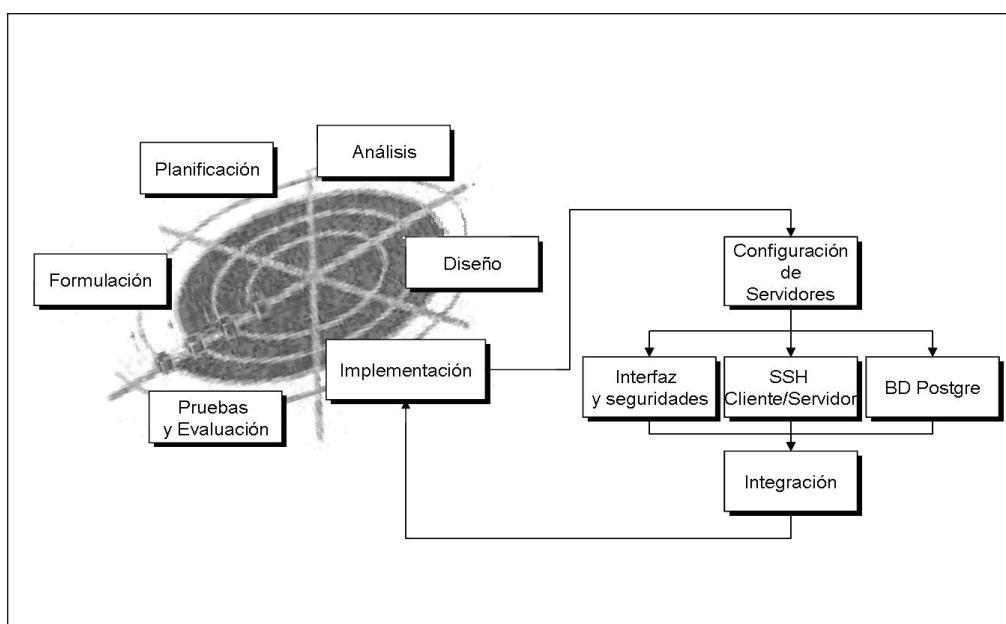


Figura 2.32. *Modelo de Desarrollo*

2.6 Recursos:

2.6.1 Recursos Humanos:

La cantidad de personas requeridas para el desarrollo del proyecto de software es de tres personas mes, a continuación listamos el recurso humano:

- ✓ Jonathan García
- ✓ Héctor Lara
- ✓ Johanna Zumba

2.6.2 Recursos o Componentes de Software:

Software:

- ✓ Postgresql.
- ✓ Java: scripts, applets, jsp
- ✓ OpenSSH

Componentes ya desarrollados: Utilizaremos las librerías existentes de OpenSSH para lograr la comunicación encriptada con el cliente.

Algoritmos de encriptación/desencriptación.

Componentes nuevos: Crearemos nuestros propios componentes.

2.6.3 Recursos del Entorno:

Necesitamos dos computadores para que funcionen como servidor y cliente respectivamente, el servidor deberá tener implementado los servicios httpd (Servicio Web Apache) y sshd (Secure Shell), mientras que el cliente se desarrollara con tecnología JAVA.

A continuación detallamos los requerimientos que necesitamos en el servidor:

- ✓ Sistema Operativo Linux (Fedora Core 4)
- ✓ Configuración del protocolo SSH
- ✓ Configuración de la Conexión Telnet
- ✓ Configuraciones de Servicio Web Apache
- ✓ Directorios de Fuentes y ficheros del aplicativo Web
- ✓ Configuración DNS y usuarios
- ✓ Configuración de Servicios adicionales necesarios

El Cliente (o estación remota):

Sistema Operativo: Win / Linux en el número de unidades que se consideren necesarios.

Características Técnicas del computador:

Elemento	Cliente	Servidor
Procesador	Pentium 4	Pentium 4
CPU	3 GHz	3 GHz
Memoria Ram	512 GB	1GB
Disco Duro	80 GB	80 GB

Figura 2.33. *Tabla de características técnicas*

2.7 Cronograma de Trabajo:

A continuación presentamos el cronograma de actividades en la primera fase del proyecto:

Etapas	Inicio	Fin	Objetivo
1	16/09/06	24/09/06	Obtener conocimientos necesarios sobre el protocolo Secure Shell (SSH) y otros protocolos de seguridad.
2	30/09/06	15/10/06	Determinar Objetivos del proyecto y los recursos necesarios
3	21/10/06	05/11/06	Establecer los diferentes casos de uso de nuestra aplicación
4	11/11/06	26/11/06	Diseño de todos los componentes del proyecto, datos, clases, objetos, interfaz, etc.

Figura 2.34. *Cronograma de la Primera Fase*

Etapas	Descripción
1	INVESTIGACIÓN
2	INTRODUCCIÓN
3	ANÁLISIS
4	DISEÑO

Figura 2.35. *Etapas de la Primera Fase*

CAPITULO 3

DISEÑO DEL SISTEMA

3. DISEÑO

3.1. Diseño de Clases y Objetos

3.2.1. Diagramas de Clases

A continuación se detallan las clases y los métodos principales involucradas dentro del diseño del sistema:

usuario	sesion	detalle
id_user no_user ds_user st_user	id_sesn id_user fe_sesn hr_sesn ip_user	id_detl id_sesn fe_detl hr_detl ds_detl
get_id_user() get_no_user() get_ds_user() get_dt_user() set_id_user() set_no_user() set_ds_user() set_dt_user()	get_id_sesn() get_id_user() get_fe_sesn() get_hr_sesn() get_ip_user() set_id_sesn() set_id_user() set_fe_sesn() set_hr_sesn() set_ip_user()	get_id_detl() get_id_sesn() get_fe_detl() get_hr_detl() get_ds_detl() set_id_detl() set_id_sesn() set_fe_detl() set_hr_detl() set_ds_detl()

Figura 3.1 *Diagrama de Clases*



Figura 3.2 *Diagrama de Objetos*

3.2. Asociaciones

Las asociaciones entre las clases se representan mediante una línea que las une, la línea puede tener una serie de elementos gráficos que expresan características particulares de la asociación.

3.2.1. Nombre de la Asociación y Dirección

En el gráfico se puede leer la asociación como “Un usuario inicia una sesión y una sesión contiene varios detalles”



Figura 3.3 *Diagrama de Asociación*

3.2.3. Multiplicidad (Cardinalidad)

La multiplicidad es una restricción que se pone a una asociación, que limita el número de instancias de una clase que pueden tener esa asociación con una instancia de otra clase.

El gráfico indica que un usuario puede tener muchas sesiones y una sesión puede tener muchos detalles que contienen información sobre instrucciones ejecutadas por el usuario.



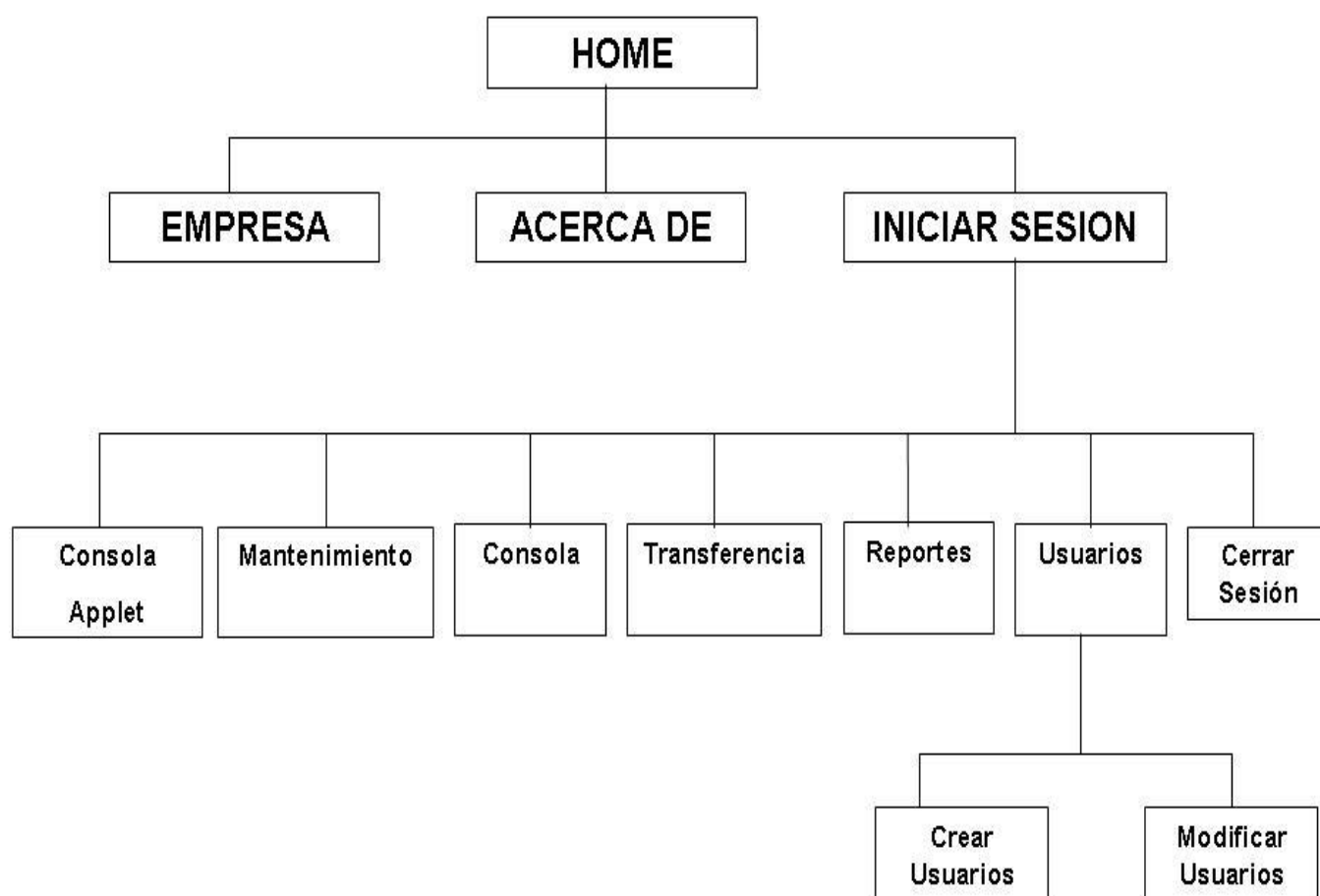
Figura 3.4 *Diagrama de Multiplicidad*

3.3. Fase de Diseño de Contenido e Interfaz

El contenido que se presentara en nuestra Aplicación Web, será la misión y visión de nuestro proyecto, además de una introducción acerca de SSH y una pequeña publicidad de nuestra aplicación.

La aplicación Web contendrá enlaces a otras páginas o sitios Web, sobre las herramientas, protocolos, servicios que componen el sistema, estos enlaces pueden ser útiles para futuras consultas de los usuarios.

3.3.1 Estructura del sitio



3.3.2 Descripción del sitio

Menú principal:

- ✓ **Iniciar Sesión**: esta opción me permite abrir la página para iniciar sesión, aquí deberá ingresar los datos de usuario y password.

- ✓ **Acerca de**: Muestra información básica de los componentes del sistema, así como links a los sitios de las deferentes páginas relacionadas a los protocolos o servicios utilizados.

- ✓ **Empresa**: Contiene información de la misión, visión, etc. del proyecto.

- ✓ **Home**: Link para regresar a la pagina principal.

Menú de sesión (usuario especializado):

- ✓ **Consola**: permitirá cargar el applet para utilizar la consola de comandos, donde el usuario podrá ejecutar la instrucción hacia el servidor SSH.

- ✓ **Reportes**: Abrirá la página donde nos mostrara los diferentes tipos de reportes en la cual el usuario podrá elegir de acuerdo a las criterios de búsqueda.

- ✓ **Transferencias**: Abrirá un applet donde se permite seleccionar un archivo determinado para transferirlo hacia el servidor.

- ✓ **Cerrar sesión**: Opción para terminar la sesión del usuario.

Menú de sesión (súper usuario):

- ✓ **Consola**: permitirá cargar en la página un cuadro y una area de texto donde se ingresará el comando a ejecutar y se mostrará la respuesta del servidor respectivamente, mediante la comunicación SSH, con ciertas limitaciones en cuanto a comandos a utilizar.
- ✓ **Consola Applet**: permitirá levantar un Applet en la aplicación el cual va a estar fuera de la página y a diferencia de la consola sencilla este applet podrá ejecutar casi cualquier comando Linux que se ingrese.
- ✓ **Reportes**: Abrirá la página donde nos mostrara los diferentes tipos de reportes en la cual el usuario podrá elegir de acuerdo a las criterios de búsqueda.
- ✓ **Transferencias**: Abrirá un applet donde se permite seleccionar un archivo determinado para transferirlo hacia el servidor.
- ✓ **Cuentas de Usuario**: Permitirá realizar las siguientes opciones en las cuentas de usuario.
 - Crear
 - Modificar
- ✓ **Cerrar sesión**: Opción para terminar la sesión del usuario.

3.4. Diseño de Navegación

Para definir las rutas de navegación que permiten al usuario acceder al contenido o a los servicios de la página Web, hemos establecido los siguientes puntos:

- ✓ **Semántica de la navegación**

Usuario Especializado (cliente registrado en el servidor)

Podrá acceder a los servicios de la aplicación Web después de iniciar su respectiva sesión.

- ✓ **Mecánica (sintaxis) para lograr la navegación**

Mostramos los diferentes caminos que tiene el usuario para recorrer las paginas Web.

Para el súper usuario:

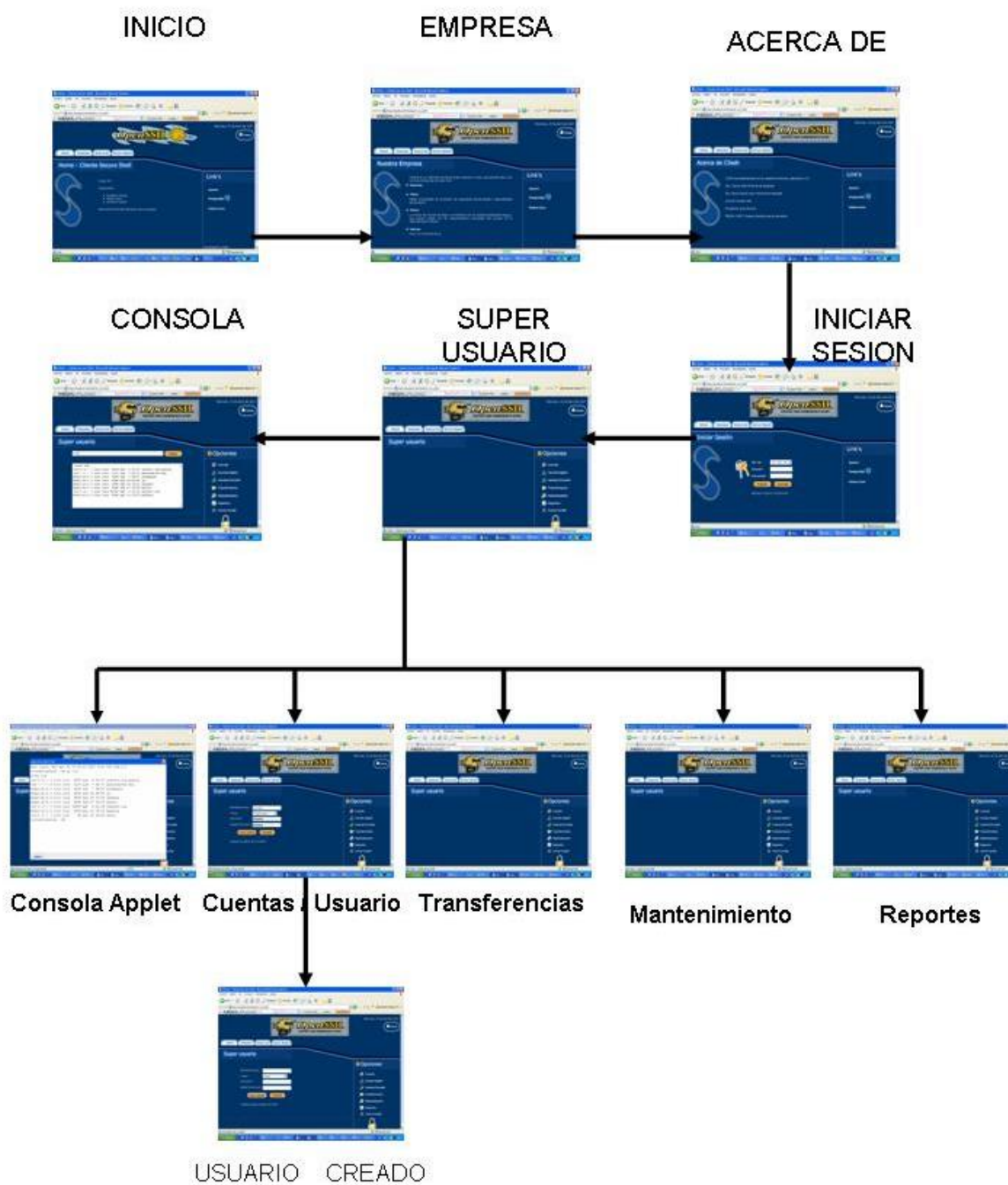


Figura 3.5 Diagrama de Navegación – Súper Usuario

Para el usuario especializado:

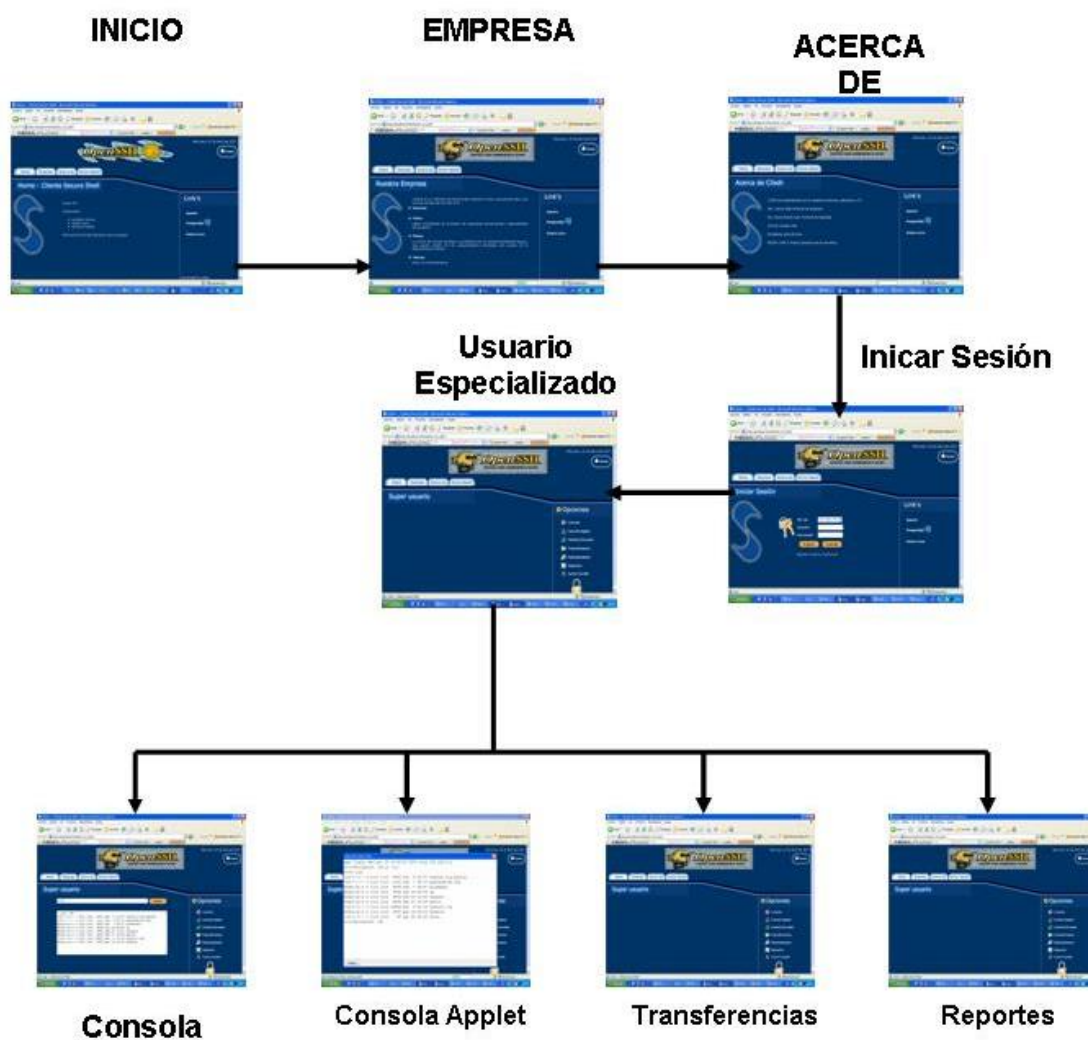


Figura 3.6 Diagrama de Navegación – Usuario Especializado

3.5. Diseño de la Interfaz



Figura 3.8. *Página Principal*



Figura 3.9. Nuestra Empresa

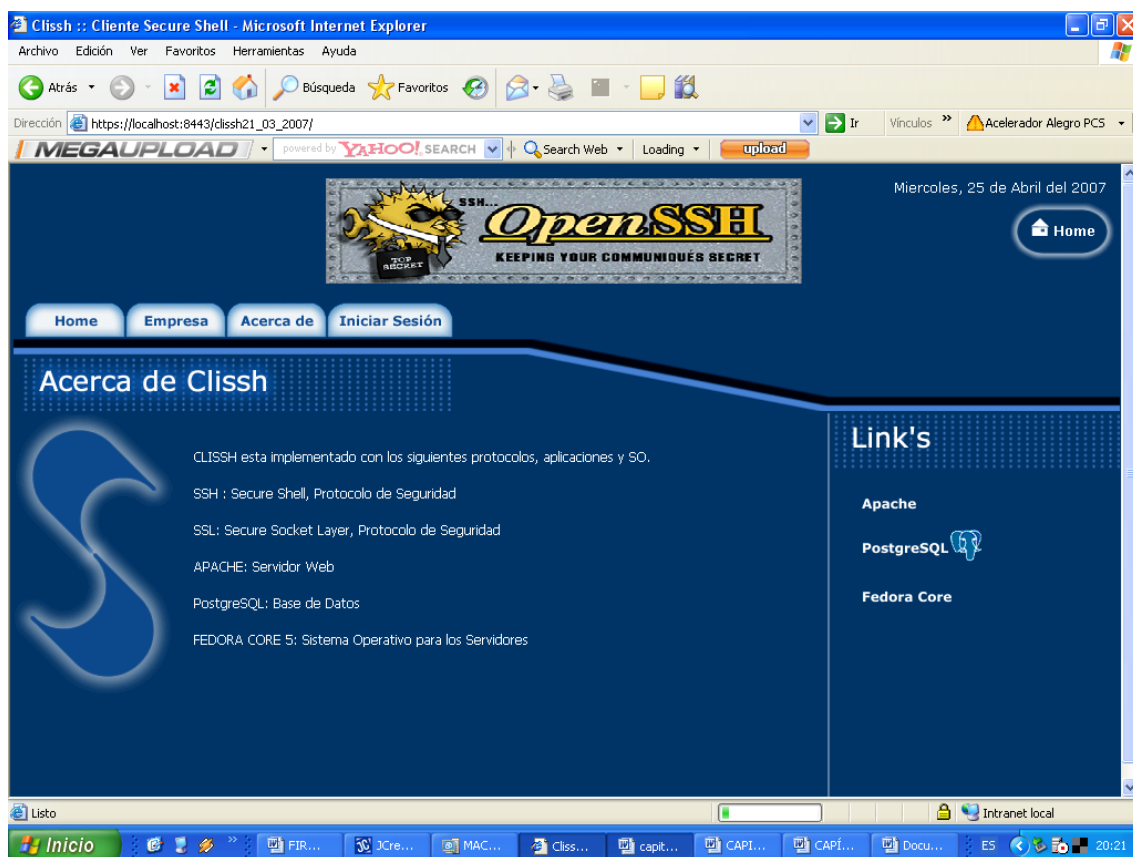


Figura 3.10. Página Acerca de



Figura 3.10. Página Iniciar Sesión

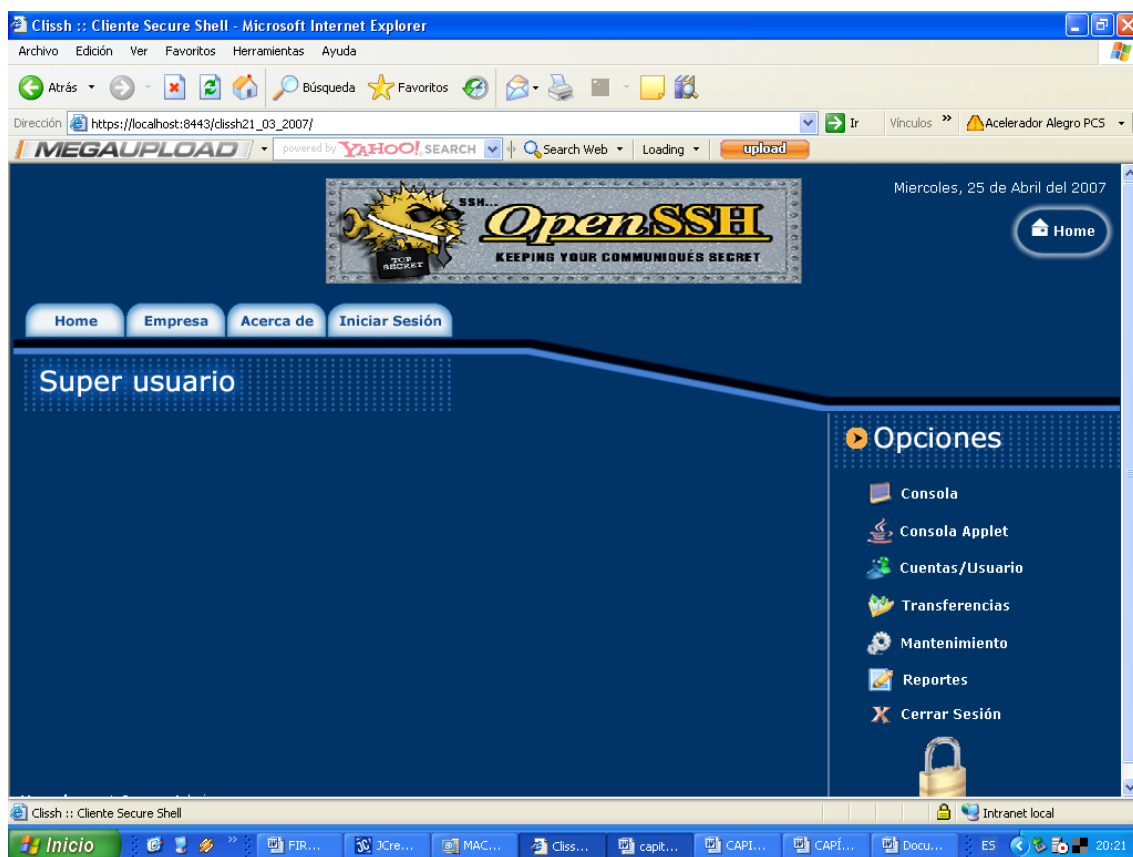


Figura 3.11. Menú de Opciones

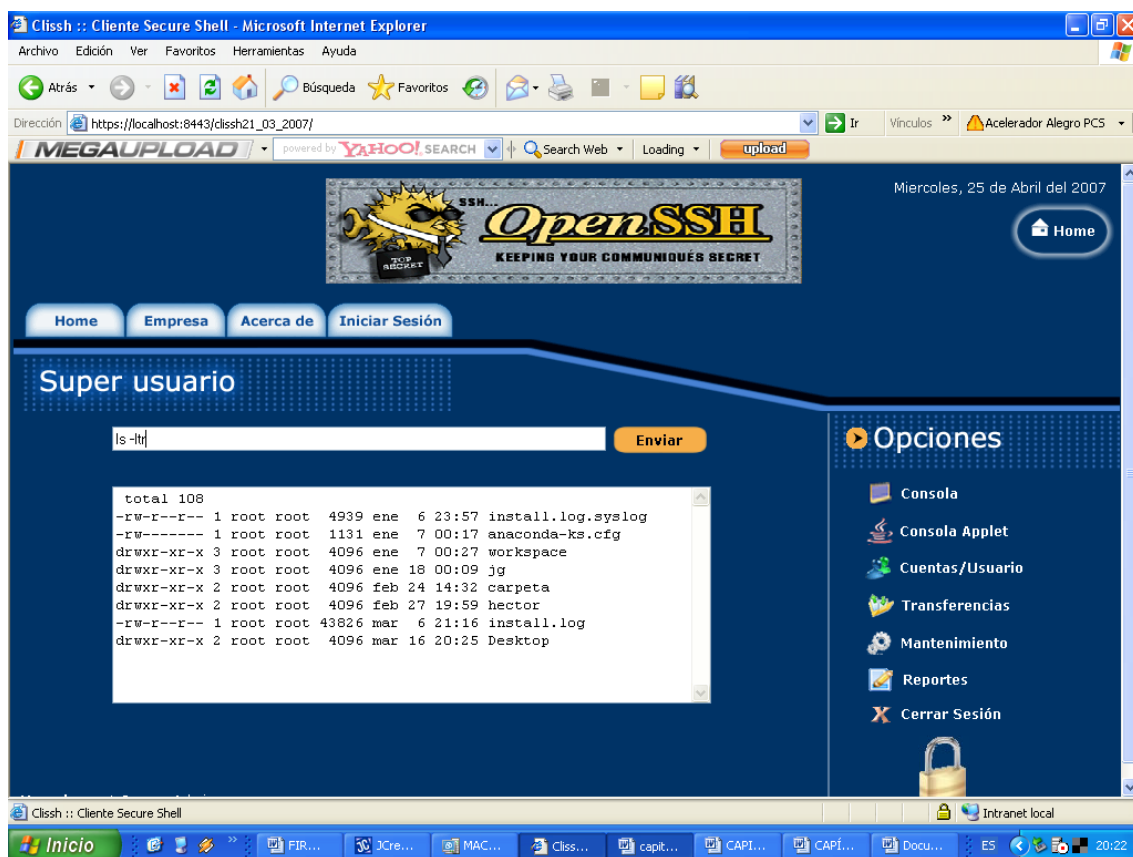


Figura 3.12. Consola

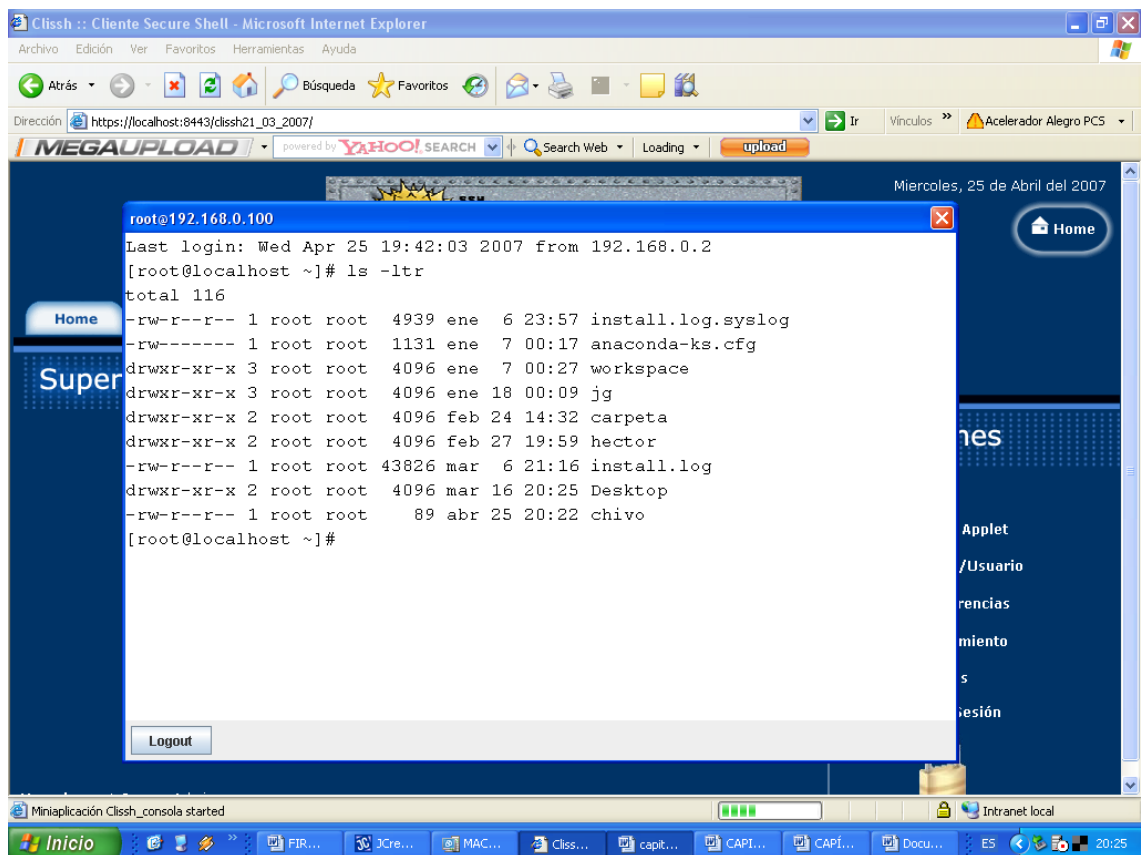


Figura 3.13. Consola Applet

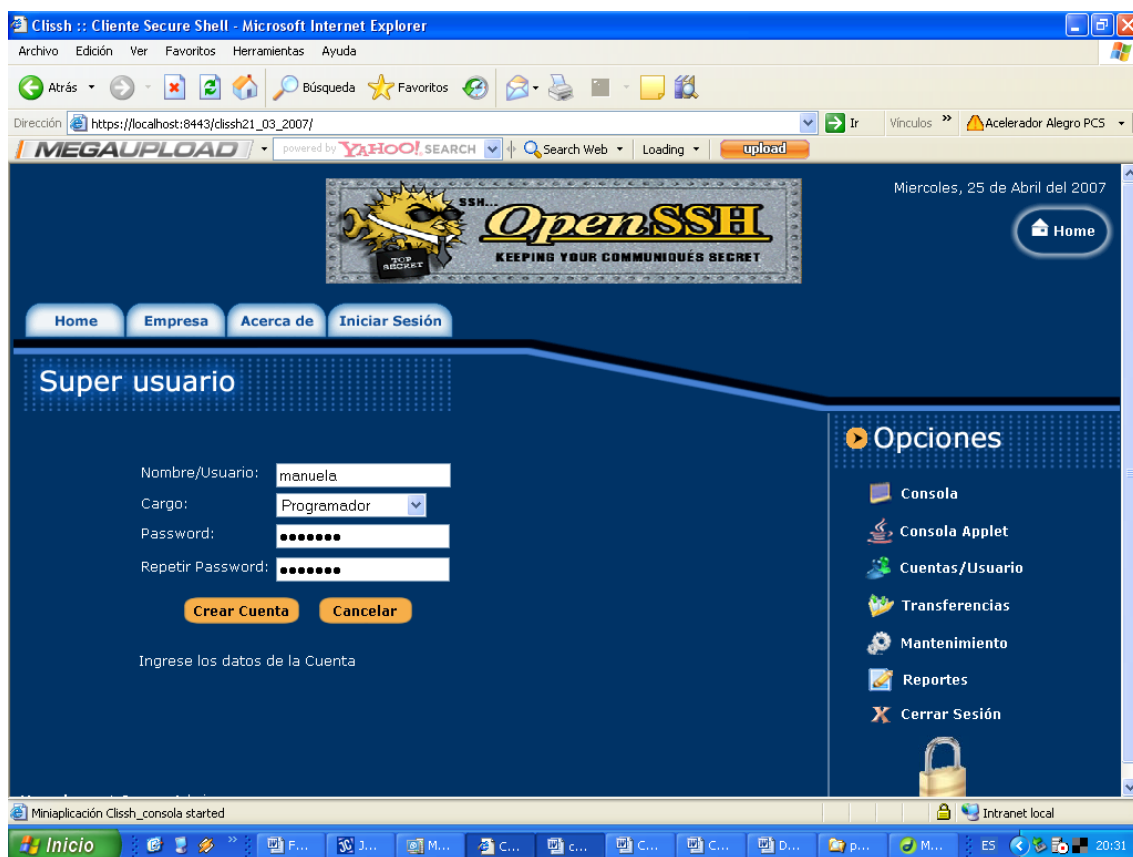


Figura 3.14. Administración de Usuarios

3.6. Diseño de Datos

Para la realización de nuestro proyecto de acuerdo al análisis y diseño de clases y objetos, la base de datos se implementara en Postgresql 8.2

Se ha definido como herramienta de almacenamiento de datos al motor de base Postgresql8.2 por reunir las características básicas de seguridad, capacidad de almacenamiento, disponibilidad del servicio y el buen desempeño demostrado en múltiples plataformas por lo que es considerada la mejor base de datos Open Source en a actualidad.

Para iniciar con el diseño de datos vamos a definir los diferentes diagramas que componen nuestro proyecto los cuáles los detallamos a continuación:

3.6.1 Diagrama Entidad Relación

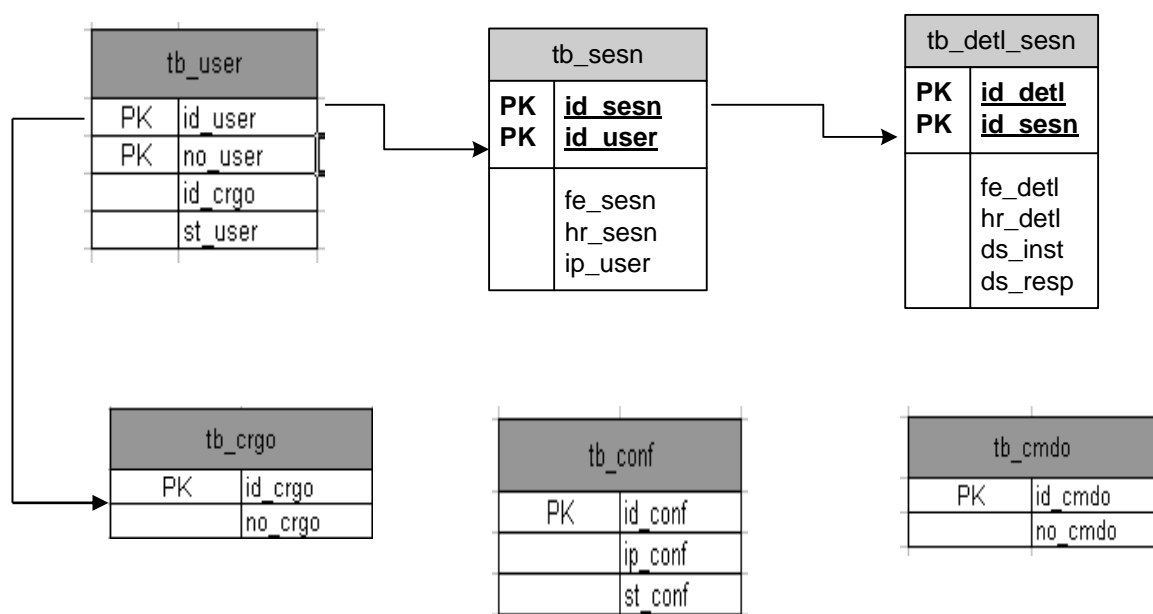


Figura 3.12. Diagrama Entidad - Relación

3.6.2 Estándar de los Datos

A continuación se detalla los estándares para la denominación de tablas y campos aplicados a la base de datos del sistema.

Tipo del campo	Descripción
tb	Tabla
id	identificador
no	nombre
ds	descripción
st	status
fe	fecha
hr	hora
ip	dirección ip

Figura 3.13. *Tabla de Estándares*

Para la denominación de los componentes de la base de datos se establecerá la siguiente nomenclatura:

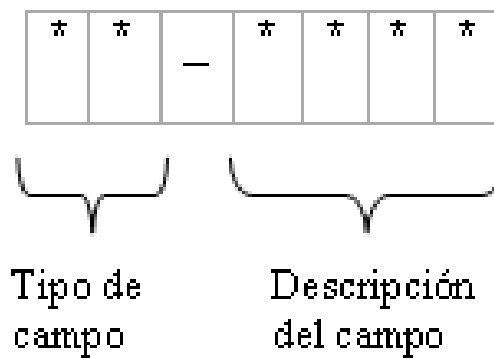


Figura 3.14. *Estructura de nomenclatura*

Los dos primeros 2 caracteres para el tipo de objeto: si es tabla o campo; en el caso de ser campo dependerá del dato que se valla a almacenar en el mismo.

Los 4 caracteres restantes corresponderán a la descripción del componente de base de datos.

tb_user	
Descripción	Campo
Identificador de Usuario	id_user
Nombre del Usuario	no_user
Cargo del Usuario	id_crgo
Estado	st_user

Figura 3.15. *Tabla de Usuario*

tb_sesn	
Descripción	Campo
Identificador de Sesión	id_sesn
Identificador de Usuario	id_user
Fecha de la Sesión	fc_sesn
Hora de la Sesión	hr_sesn
Dirección IP del user	ip_user

Figura 3.16. *Tabla de Sesión*

tb_detl_sesn	
Descripción	Campo
Identificador de detalle	id_detl
Identificador de la Sesión	Id_sesn
Fecha de la Sesión	fc_detl
Hora de la Sesión	hr_detl
Línea del comando utilizado	ds_inst
Respuesta del Comando	ds_resp

Figura 3.17. *Tabla del Detalle de Sesión*

tb_conf	
Identificador de configuración	id_conf
Dirección ip del servidor	ip_conf
Status de Registro	st_conf

Figura 3.18. *Tabla de Configuración*

tb_cmndo	
Identificador de comando	id_cmndo
Descripción de comando	no_cmndo

Figura 3.19. *Tabla de Comandos*

tb_cmndo	
Identificador de cargo	id_crgo
Descripción de commando	no_crgo

Figura 3.20. *Tabla de Cargos*

3.6.3 Definición de la estructura de datos

tb_user			
Contiene información de la cuenta del usuario.			
Descripción	Campo	Tipo	Null
Identificador de Usuario	id_user	Integer	not null
Nombre del Usuario	no_user	varchar(8)	not null
Cargo del Usuario	Id_crgo	Integer	not null
Estado	st_user	char	

Figura 3.19. Estructura de la Tabla Usuario

tb_sesn			
Contiene información de la sesión que ha establecido el usuario.			
Descripción	Campo	Tipo	Null
Identificador de Sesión	id_sesn	Integer	
Identificador de Usuario	id_user	Integer	not null
Fecha de la Sesión	fc_sesn	Date	not null
Hora de la Sesión	hr_sesn	Date	not null
Dirección IP del user	lp_user	varchar(15)	not null

Figura 3.20. *Estructura de la Tabla Sesión*

tb_detl_sesn			
Contiene información del detalle de la sesión del usuario.			
Descripción	Campo	Tipo	Null
Identificador de detalle	id_detl	integer	not null
Identificador de la Sesión	id_sesn	integer	not null
Fecha de la Sesión	fe_detl	date	not null
Hora de la Sesión	hr_detl	date	not null
Línea del comando utilizado	ds_inst	varchar(50)	not_null
Respuesta del Comando	ds_resp	varchar(500)	

Figura 3.21. Estructura de la Tabla Detalle Sesión

tb_conf			
Tabla que contiene las configuraciones del servidor SSH			
Descripción	Campo	Tipo	Null
Identificador de configuración	id_conf	integer	not null
Dirección ip del servidor	lp_conf	integer	not null
Status de Registro	st_conf	integer	not null

Figura 3.22. Estructura de la Tabla Configuración

tb_cmdo			
Tabla que contiene los comandos no permitidos			
Descripción	Campo	Tipo	Null
id_cmdo	id_cmdo	Integer	not null
no_cmdo	no_cmdo	Varchar(20)	not null

Figura 3.22. Estructura de la Tabla comandos

tb_crgo			
Tabla que contiene los cargos de los usuarios			
Descripción	Campo	Tipo	Null
id_crgo	Id_crgo	Integer	not null
no_crgo	No_crgo	Varchar(20	not null

Figura 3.22. Estructura de la Tabla Cargos

3.7. Diseño de Comunicaciones

Como ya estamos acostumbrados en las aplicaciones de red y en especial dentro de Internet, ssh/sshd actúan basándose en la arquitectura cliente/servidor, en este caso concreto sshd se ejecuta en el servidor en un puerto (el defecto es el 22) a la espera de que alguien utilizando un cliente ssh se conecte para ofrecerle una sesión segura encriptándola de extremo a extremo.

Software requerido

- ✓ openssh-4.0p1-3.i386
- ✓ openssh-clients-4.0p1-3.i386
- ✓ openssh-server-4.0p1-3.i386

Ficheros de configuración y parámetros

Los archivos de configuración de OpenSSH se encuentran en /etc/ssh

✓ Archivo sshd_config

En este archivo se describe la configuración del servidor SSH. Analizaremos las opciones más relevantes:

Port: Esta opción permite especificar el puerto TCP que utilizará el servidor. El valor usual es 22.

Protocol: Versión del protocolo a utilizar. Usaremos solamente el valor 2.

HostKey: Clave privada de RSA o DSA del host. Normalmente las claves de host son generadas en el momento de la instalación de OpenSSH, y el valor de esta opción es /etc/ssh/ssh_host_rsa_key.

PubkeyAuthentication: Si el valor de esta opción es yes, entonces se permite la autenticación de usuarios mediante clave pública.

AuthorizedKeysFile: Mediante esta opción se indica al servidor en donde están almacenadas las claves públicas de los usuarios. El valor por defecto es

%h/.ssh/authorized_keys, e indica que deben buscarse en el archivo authorized_keys, del directorio .ssh dentro del directorio home de cada usuario.

PasswordAuthentication: Si el valor de esta opción es yes, se permite la autenticación de usuarios mediante contraseñas.

✓ Archivo ssh_config

En este archivo se describe la configuración del cliente SSH. Las opciones más importantes son:

Host: Esta opción actúa como un divisor de sección. Puede repetirse varias veces en el archivo de configuración. Su valor, que puede ser una lista de patrones, determina que las opciones subsiguientes sean aplicadas a las conexiones realizadas a los hosts en cuestión. El valor * significa “todos los hosts”.

Port: Es el puerto de TCP que utiliza el servidor (normalmente, 22).

Protocol: Es la versión del protocolo utilizada (normalmente 2,1, pero se recomienda solamente 2).

PubkeyAuthentication: Autenticación mediante clave pública (se recomienda yes).

IdentityFile: Archivo que contiene la clave pública (en caso de usar RSA, lo usual es ~/.ssh/id_rsa).

PasswordAuthentication: Autenticación mediante contraseñas (yes o no).

StrictHostKeyChecking: Define que hará el cliente al conectarse a un host del cual no se dispone de su clave pública. El valor no hace que sea rechazada la clave del servidor (y por lo tanto, se aborta la conexión), el valor yes hace que se acepte automáticamente la clave recibida, y el valor ask hace que se pida confirmación al usuario.

Ciphers: Algoritmos de cifrado simétrico soportados para su uso durante la sesión.

CAPITULO 4

DESARROLLO Y PRUEBAS DEL SISTEMA

4. DESARROLLO Y PRUEBA DEL SISTEMA

4.1. Creación de la Base de Datos

Las características de la base de datos son las siguientes:

1. Nuestra base de datos va a ser desarrollada en Postgresql 8.2
2. Reconocida mundialmente como la mejor base de datos de toda la gama de software existente actualmente en el medio tecnológico y lo mas importante y beneficioso para las personas que quieran usarla es que se distribuye bajo licencia Open Source.
3. La capacidad de almacenamiento y crecimiento es muy amplia por lo que podemos almacenar datos de manera muy flexible ya que posee una gran variedad de tipos de datos para acoplarse casi a cualquier necesidad que se presente.
4. La seguridad de los datos es un factor predominante en este motor de base de datos ya que posee procedimientos y funciones almacenados que ayudan a la protección de claves y contraseñas, como por ejemplo nos permite la encriptación de datos.

5. Nos ofrece un canal de comunicación seguro ya que soporta conexiones SSL.
6. Los objetos de conexión y manipulación de datos de la base están desarrollados en su totalidad en lenguaje java, un lenguaje que interactúa de una manera optima con el motor de base de datos definido.

4.2. Desarrollo de Aplicación

1. Para la realización de este proyecto se ha planificado desarrollarlo en dos fases muy bien definidas, a conocer:

Comunicación: En esta fase del proyecto se va a desarrollar todas las clases necesarias en lenguaje Java que van a permitir realizar la conexión del host cliente con el servidor Linux teniendo en consideración todas las actividades que se relacionan al realizarlo utilizando el protocolo SSH y la manera de autenticación del servidor.

Adicionalmente a las clases y objetos diseñados para la implementación del proyecto se deben alterar una serie de archivos de configuración tanto en el servidor como en la maquina

intermedia que va a actuar como cliente SSH para que la comunicación se desarrolle de manera eficiente

Presentación.- Para la parte visual del proyecto necesitamos mantener un equilibrio entre la estética de las páginas a desarrollar para que sean agradables al usuario y funcionalidad que debe tener para que interactue con las clases desarrolladas en Java en donde se va a realizar todas las transacciones que el cliente solicite, a excepción de los comandos de Linux que le envíe al servidor, por lo que las páginas serán desarrolladas en JSP(Java Server Pages).

Estas páginas se deben alojar en un servidor Web para que los usuarios puedan acceder a ellas y poder obtener los servicios planteados en este proyecto, por lo que se utiliza el servidor Apache Tomcat el cual va a ser el encargado de gestionar las peticiones de los clientes que se conecten al sistema.

El servidor Web debe mantener un nivel de seguridad muy alto para evitar la interceptación de información o cualquier tipo de ataque malicioso por parte de terceros a nuestro sistema por lo

que se añade un módulo SSL a dicho servidor para evitar los contratiempos mencionados.

Adicionalmente señalamos algunas fases por las que tiene que pasar en el tiempo el desarrollo del proyecto:

Configuración de Servidores: Fase de instalación y configuración de servidores SSH y Web (plataformas, ficheros de Servicios)

Interfaz y Seguridades: Fase de implementación de las pantallas, applets y configuraciones de seguridad entre ellos.

SSH Cliente Servidor: Fase de Configuración de Protocolos SSH y configuración de seguridades de comunicación entre cliente / servidor.

Fase de Integración de la aplicación: integración de los componentes de datos, interfaces, comunicación SSH y servidores. Esta fase incluye por completo el desarrollo del aplicativo a utilizar como son los applets, clases, paquetes, etc.

4.3. Prueba del Sistema

Revisión del modelo de contenido: Lo utilizaremos para descubrir errores tipográficos, gramaticales, inconsistencia de contenido, representaciones gráficas y de referencias cruzadas.

Revisión del modelo de diseño la webApp: Para descubrir errores en la navegación y en los enlaces.

Pruebas de integración: Se integran todas las páginas que conforma la webApp y se realizan las pruebas respectivas.

Pruebas de la webApp ensamblada: Se prueba para conseguir una funcionalidad global y un contenido, esto se lo realiza con la interacción del usuario en el sistema usando datos reales.

Pruebas en otros S.O, plataformas y hardware: La webApp se implementa en una variedad de configuraciones diferentes de entornos y comprobar así la compatibilidad con cada configuración

Pruebas de La WebApp con varios usuarios: Se comprueba con una población de usuarios finales controlada y monitorizada, este grupo de usuarios deben abarcar todos los roles posibles de usuarios.

4.3.1. Modelo de prueba

A continuación se destalla los modelos con sus escenarios y funcionalidades:

Navegación

#	Escenario	Descripción Escenario
1	E1	Disponibilidad de la página
2	E2	Prueba de contenido
3	E3	Enlaces

Tipo de Caso E: Entorno **T:** Trazabilidad **PD:** Prueba de Datos

Escenario: Disponibilidad

Test Plan: Paginas solicitadas

Tipo Caso	Test Input Folder	Test Case	Estado
E	Página principal	Visualiza la aplicación	Ok

Tipo de Caso E: Entorno **T:** Trazabilidad **PD:** Prueba de Datos

Escenario: Prueba de contenido

Test Plan: Redacción

Tipo Caso	Test Input Folder	Test Case	Estado
E	Página principal	Ortografía / Gramática	Ok

Tipo de Caso E: Entorno **T:** Trazabilidad **PD:** Prueba de Datos

Escenario: Enlaces

Test Plan: Links

Tipo Caso	Test Input Folder	Test Case	Estado
E	Página principal	Hipervínculos	Ok

Comunicación

#	Escenario	Descripción Escenario
1	E1	Inicio Sesión
2	E2	Base de datos
3	E3	Tiempo de respuesta

Tipo de Caso E: Entorno **T:** Trazabilidad **PD:** Prueba de Datos

Escenario: Inicio Sesión

Test Plan: Verificar usuario

Tipo Caso	Test Input Folder	Test Case	Estado
T	Inicio Sesión	Datos incompletos	Ok

Tipo de Caso E: Entorno **T:** Trazabilidad **PD:** Prueba de Datos

Escenario: Inicio Sesión

Test Plan: Verificar usuario

Tipo Caso	Test Input Folder	Test Case	Estado
T	Inicio Sesión	Usuario encontrado	Ok

Tipo de Caso E: Entorno **T:** Trazabilidad **PD:** Prueba de Datos

Escenario: Inicio Sesión

Test Plan: Verificar usuario

Tipo Caso	Test Input Folder	Test Case	Estado
T	Inicio Sesión		
		Usuario no valido	Ok

Tipo de Caso E: Entorno **T:** Trazabilidad **PD:** Prueba de Datos

Escenario: Inicio Sesión

Test Plan: Verificar usuario

Tipo Caso	Test Input Folder	Test Case	Estado
T	Inicio Sesión		
		Error de logon de Linux	Ok

Comunicación

#	Escenario	Descripción Escenario
1	E1	Consola
2	E2	Comunicación
3	E3	Tiempo de respuesta

Tipo de Caso E: Entorno **T:** Trazabilidad **PD:** Prueba de Datos

Escenario: Consola

Test Plan: Verificar
disponibilidad

Tipo Caso	Test Input Folder	Test Case	Estado
T	Consola	Visualización de consola	Ok

Tipo de Caso E: Entorno **T:** Trazabilidad **PD:** Prueba de Datos

Escenario: Consola

Test Plan: Interacción

Tipo Caso	Test Input Folder	Test Case	Estado
T	Consola	Ingreso de comandos	Ok

Tipo de Caso E: Entorno **T:** Trazabilidad **PD:** Prueba de Datos

Escenario: Consola

Test Plan: Obtener respuesta

Tipo Caso	Test Input Folder	Test Case	Estado
T	Consola	Visualización de respuesta	Ok

Tipo de Caso E: Entorno **T:** Trazabilidad **PD:** Prueba de Datos

Escenario: Consola

Test Plan: Obtener respuesta

Tipo Caso	Test Input Folder	Test Case	Estado
T	Consola	Error de comando	Ok

Tipo de Caso E: Entorno **T:** Trazabilidad **PD:** Prueba de Datos

Escenario: Consola

Test Plan: Obtener respuesta

Tipo Caso	Test Input Folder	Test Case	Estado
T	Consola	Comando no permitido	Ok

Tipo de Caso E: Entorno **T:** Trazabilidad **PD:** Prueba de Datos

Escenario: Comunicación

Test Plan: Conexión

Tipo Caso	Test Input Folder	Test Case	Estado
T	Conexión	Enlace a servidor Web	Ok

Tipo de Caso E: Entorno **T:** Trazabilidad **PD:** Prueba de Datos

Escenario: Comunicación

Test Plan: Conexión

Tipo Caso	Test Input Folder	Test Case	Estado
T	Conexión	Enlace a servidor Linux	Ok

Tipo de Caso E: Entorno **T:** Trazabilidad **PD:** Prueba de Datos

Escenario: Comunicación

Test Plan: Conexión

Tipo Caso	Test Input Folder	Test Case	Estado
T	Conexión		
		Error de página	Ok

Tipo de Caso E: Entorno **T:** Trazabilidad **PD:** Prueba de Datos

Escenario: Tiempo de respuesta

Test Plan: Intervalo de tiempo

Tipo Caso	Test Input Folder	Test Case	Estado
T	Consola		
		Intervalo de tiempo para responder aceptable	Ok

CAPITULO 5

IMPLEMENTACIÓN

DEL

SISTEMA

5. IMPLEMENTACIÓN DEL SISTEMA

5.1 Recursos:

5.1.1 Recursos Humanos:

La cantidad de personas requeridas para el desarrollo del proyecto de software es de tres personas mes, a continuación listamos el recurso humano:

- ✓ Jonathan García
- ✓ Héctor Lara
- ✓ Johanna Zumba

5.1.2 Recursos o Componentes de Software:

Software:

- ✓ Postgre.
- ✓ Java: scripts, applets
- ✓ OpenSSH

Componentes ya desarrollados: Utilizaremos las librerías existentes de OpenSSH para lograr la comunicación encriptada con el cliente.

Algoritmos de encriptación/desencriptación.

Componentes nuevos: Crearemos nuestros propios componentes.

5.1.3 Recursos del Entorno:

Necesitamos dos computadores para que funcionen como servidor y cliente respectivamente, el servidor deberá tener implementado los servicios httpd (Servicio Web Apache) y sshd (Secure Shell), mientras que el cliente se desarrollara con tecnología JAVA.

A continuación detallamos los requerimientos que necesitamos en el servidor:

- ✓ Sistema Operativo Linux (Fedora Core 4)
- ✓ Configuración del protocolo SSH
- ✓ Configuración de la Conexión Telnet
- ✓ Configuraciones de Servicio Web Apache
- ✓ Directorios de Fuentes y ficheros del aplicativo Web
- ✓ Configuración DNS y usuarios
- ✓ Configuración de Servicios adicionales necesarios

El Cliente (o estación remota):

Sistema Operativo: Win / Linux en el número de unidades que se consideren necesarios.

Características Técnicas del computador:

Elemento	Cliente	Servidor
Procesador	Pentium 4	Pentium 4
CPU	3 GHz	3 GHz
Memoria Ram	512 GB	1GB
Disco Duro	80 GB	80 GB

Figura 5.3. *Tabla de características técnicas*

5.2. Elementos Lógicos

Software

Se requirió de software adicional para desarrollo de las páginas, herramientas gráficas y de creación de animaciones, herramientas de programación, base de datos, etc..

Cantidad	Detalle
2	JCreatorPro
1	Dreamweaver
2	Visio 2000
1	Fedora 5
1	Apahe Tomcat
1	PgAdmin III

CAPITULO 6

RECOMENDACIONES

Y

CONCLUSIONES

En el transcurso de la implementación del proyecto se ha definido las siguientes observaciones y surgen como recomendaciones algunas pautas que nos permitirán obtener resultados favorables para la utilización del sistema establecido, las cuales pasamos a listar.

6.1. Software

En cuanto a al Sistema Operativo ponemos a consideración:

- Fedora Core 5

Por que es una distribución de Linux bastante confiable y ofrece una estabilidad en cuanto a los servicios que presta y las herramientas que utiliza para asegurar de una manera confiable el trabajo de los clientes.

Entre sus características tenemos:

- Multitarea
- Multiusuario
- Multiplataforma
- protección de memoria ante los procesos

- Copia en escritura aumentando así su velocidad y consumiendo menos memoria
- Memoria virtual usando paginación
- Volcados de estado
- Código fuente disponible
- Consolas virtuales múltiples
- Soporte para varios sistemas de archivos comunes
- Acceso transparente a particiones MS-DOS, TCP/IP incluyendo ftp, telnet, NFS, etc.

Algunas de sus fortalezas son:

- Código Abierto, todo en uno,
- Liberación rápida de distribuciones

En cuanto a la Base de Datos ponemos consideración:

- PostgreSQL 8.2,

Este motor de base de datos esta considerada como la mejor aplicación para este tipo de tareas en el mundo tecnologico por que permite una innumerable gama de herramientas y procesos que permiten al desarrollador de sistemas planificar casi cualquier actividad usando esta base de datos.

Mantiene los datos de las empresas seguros y las entidades tienen un mayor grado de confianza.

Para la elaboración de las Páginas:

- Recomendamos para actualizaciones, modificaciones o alcances el uso de páginas JSP las cuales permiten una interacción dinámica entre la aplicación transaccional y mantiene la estética visual para el mejor desempeño del sistema.

Para la verificación de los Datos ingresados en los Formularios.

- Javascript

Ya que al igual que jsp también permite la creación de páginas dinámicas y son scripts insertados en los documentos html e interpretados del lado del cliente lo que le proporciona mayor rapidez en la verificación de datos ingresados por el usuario en un formulario.

6.2. Hardware

- El equipo para el Servidor debe ser como mínimo un Pentium IV
- Mínimo 256 de memoria RAM
- Verificación de que los servicios del SSH estén ejecutándose del lado del servidorssh
- El servidor web debe ser como mínimo Pentium IV
- Verificación de que en equipo que hace de servidor de aplicaciones esten ejecutandose el cliente ssh.
- El servicio de postgresql debe estar levantado en el equipo que hace de servidor web.
- El servicio de manipulación de aplicaciones debe estar levantado en el servidor destinado a este propósito.

6.1. Conclusiones

- El Internet es una Herramienta que puede ser explotada de una manera muy variada en distintos campos de la ciencia y artes, en este caso hemos usado este concepto para poder realizar nuestro trabajo cotidiano desde sitios geográficamente alejados.
- Desarrollamos un proceso de Ingeniería de software debidamente organizado y estandarizado desde el análisis hasta la ejecución y prueba del mismo listo para ponerlo en marcha con lo cual aseguramos un nivel de flexibilidad y homogeneidad del producto que permitirá el mantenimiento y crecimiento del sistema en aras de mejorar los procesos que en el se llevan a cabo.
- “ **La seguridad no es un producto, es un proceso**”. Muchas veces, decimos que “el software X es seguro” o que “el protocolo Y es seguro”, siendo que lo correcto sería decir que “Este sistema ofrece ciertos mecanismos que permiten elevar la seguridad en tal o cual sentido”. Nuestro sistema esta implementado para proveer un nivel de seguridad apropiado a los usuarios para brindarles un nivel de confianza aceptable en sus transacciones.