

UNIVERSIDAD DE GUAYAQUIL

Facultad de Ciencias Matemáticas y Física

**Carrera de Ingeniería en Sistemas
Computacionales**

“Portal de Ventas Corporativas de Vehículos por Internet”

PROYECTO DE GRADO

Previo a la Obtención del Título de:

INGENIERO EN SISTEMAS COMPUTACIONALES

Autores:

Betty Diana Ronquillo Rodríguez

María Pamela Pin

Ruth Catalina Moreira Carvajal

GUAYAQUIL – ECUADOR

Año: 2007

AGRADECIMIENTO

Es prácticamente imposible dar gracias a todas aquellas personas que de una u otra manera nos han apoyado para que este proyecto se haya podido culminar con éxito, pero cabe recalcar que Dios fue nuestro pilar fundamental ya que él nos concedió la vida y nos dio fortaleza en los momentos más difíciles de nuestra carrera.

A nuestros padres quien con su gran preocupación, consejos, apoyo y paciencia nos formaron para ser mujeres de bien y emprendedoras.

Gracias a los amigos y familiares quienes con sus consejos y confianza en nosotras, nos motivaron a seguir adelante y a no detenernos en medio del camino.

DEDICATORIA

María Pamela Pin: A Dios por darme la fortaleza para culminar este proyecto. A mi madre por su apoyo incondicional a lo largo de esta etapa de mi vida.

Diana Ronquillo Rodríguez: A mis padres quienes me apoyaron sin dudarlos, ellos que siempre confiaron en mi capacidad e inteligencia, creyeron en mí y nunca me dejaron sola ante mis problemas.

Ruth Moreira Carvajal: A mis padres por haberme apoyado en mis estudios cuando los necesite, a Dios por darme la vida y a todas las personas que me ayudaron.

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”.

(Reglamento de Graduación de la Carrera de Ingeniería en sistemas Computacionales, Art. 26)

Betty Diana Ronquillo Rodríguez

María Pamela Pin

Ruth Catalina Moreira Carvajal

RESUMEN

El presente proyecto está orientado a la creación de un portal web el cual estará encargado de promocionar de manera centralizada la venta de vehículos, como también repuestos para vehículos, ya sea a nivel corporativo o por una persona natural, proporcionando a los usuarios obtener información de una manera fácil, y segura de aquellos productos (vehículos y repuestos) que se exhibirán en este portal. Hacemos énfasis en la calidad y en la seguridad de la información que se ofrece al usuario, ya que nuestro objetivo primordial es lograr que se vendan aquellos productos exhibidos en este sitio web. El motivo fundamental que nos lleva a realizar este proyecto, es contribuir con el mercado de vehículos a través del internet, por lo que hoy en día es una herramienta muy utilizada y de fácil acceso para obtener cualquier tipo de información.

ÍNDICE GENERAL

	Pág.
CAPITULO 1	1
PORTAL DE VENTAS CORPORATIVAS DE VEHICULOS	1
1.1 Antecedentes	1
1.2 Problemática	
1.3 Solución a la Problemática	2
1.4 Misión	4
1.5 Visión	4
1.6 Objetivos	4
1.6.1 Objetivos Generales	4
1.6.2 Objetivos Específicos	5
1.7 Alcances	6
1.8 Recursos	8
1.9 Arquitectura	10
1.10 Cronograma	12
1.11 Modelo	13
1.12 Metodología	14
 CAPITULO 2	 17
ANALISIS	17
2.1 Levantamiento de Información	18
2.1.1 Investigación de la competencia	18
2.2 Análisis de los Requerimientos	28
2.2.1 Análisis de los Recursos Humanos	28
2.3 Estudio de la Factibilidad	28

2.3.1	Análisis Costo Beneficio	28
2.3.1.1	Factibilidad Técnica	29
2.3.1.2	Factibilidad Económica	29
2.3.1.3	Factibilidad Operacional	29
2.4	Análisis Orientado a Objetos	30
2.4.1	Definición de Usuarios del Sistema	30
2.4.2	Especificaciones de los Requisitos del Sistema	30
2.4.3	Diagrama de clases	33
2.4.3.1	Clases y Atributos O.O	34
2.4.4	Diagrama de Caso de Usos	37
CAPITULO 3		39
DISEÑO		39
3.1	Escenarios	40
3.2	Diagramas de Secuencias	44
3.3	Diagrama Entidad Relación	49
3.4	Diccionario de Datos	51
3.5	Diseño del Portal Web	56
3.5.1	Diseño del Home	56
3.5.2	Diseño de Interfaces de Compradores	57
3.5.3	Diseño de Interfaces de Vendedores	63
CAPITULO 4		68
IMPLEMENTACION Y DESARROLLO DEL SISTEMA		68
4.1	Base de Datos	69
4.2	Clases y Métodos	75
4.3	Implantación del Portal	79

CAPITULO 5**PRUEBAS 80**

5.1 Tipo de Pruebas 81

5.2 Desarrollo de las pruebas 81

CAPITULO 6**RECOMENDACIONES Y CONCLUSIONES 87**

6.1.1 Recomendaciones 87

6.1.2 Hardware 88

6.1.3 Software 89

6.1.4 Puesta en Marcha 89

6.2 Conclusiones 91

GLOSARIO 92**BIBLIOGRAFIA 106**

ÍNDICE DE GRÁFICOS

GRÁFICO #1	11
Arquitectura del Sistema	11
GRÁFICO #2	13
Modelo de Analisis	13
GRÁFICO #3	20
Evaluador de Precios - Patiotuerca.....	20
GRÁFICO #4	21
Suscripción a Boletines - Patio Tuerca.....	21
GRÁFICO #5	22
Seccion Informativa-Patituerca.....	22
GRÁFICO #6	23
Busqueda de Autos - Patiotuerca.....	23
GRÁFICO #7	24
Busqueda de Modelos por Marcas.....	24
GRÁFICO #8	25
Busqueda Autos - Autosecuador.....	25
GRÁFICO #9	26
Lista de Precios - Autosecuador.....	26
GRÁFICO #10	26
Financiamiento para Vehiculos	26
GRÁFICO #11	33
Diagrama de Clases.....	33
GRÁFICO #12	38
Diagrama de Casos se Uso.....	38

GRÁFICO #13	44
Diagrama de Secuencia – Busca Producto	44
GRÁFICO #14	45
Diagrama de Secuencia – Registra Cliente Comprador	45
GRÁFICO #15	46
Diagrama de Secuencia – Registra Cliente Vendedor	46
GRÁFICO #16	46
Diagrama de Secuencia – Autenticar	46
GRÁFICO #17	47
Diagrama de Secuencia – Cotizar	47
GRÁFICO #18	48
Diagrama de Secuencia – Imprimir Proforma.....	48
GRÁFICO #19	48
Diagrama de Secuencia – Imprimir Orden de Pago	48
GRÁFICO #20	49
Diagrama de Secuencia – Inscribir Producto	49
GRÁFICO #21	
Diagrama Entidad Relacion.....	50
GRÁFICO #22	56
Diseno del Home.....	56
GRÁFICO #23	57
Interfaz de Compradores – Categoria Productos	57
GRÁFICO #24	58
Interfaz Compradores – Tipo Productos.....	58
GRÁFICO #25	59
Interfaz Compradores - Productos	59

GRÁFICO #26	60
Interfaz Compradores – Busqueda Productos.....	60
GRÁFICO #27	61
Interfaz Compradores – Registrar Cliente	61
GRÁFICO #28	62
Interfaz Compradores - Cotizacion.....	62
GRÁFICO #29	63
Interfaz Vendedores – Tipo Producto.....	63
GRÁFICO #30	63
Interfaz Vendedores – Consulta de Paquetes	63
GRÁFICO #31	64
Interfaz Vendedores – Registrar Cliente	64
GRÁFICO #32	65
Interfaz Vendedores – Cotizacion	65
GRÁFICO #33	66
Interfaz Vendedores – Orden de Pago	66
GRÁFICO #34	67
Interfaz Compradores – Imprimir Proforma	67
GRÁFICO #35	67
Interfaz Vendedores – Imprimir Orden de Pago	67

ÍNDICE DE TABLAS

TABLA # 1.....	12
Cronograma	12
TABLA # 2.....	21
Mensaje de Suscripcion a Boletines.....	21
TABLA # 3.....	24
Caracteristicas de Vehiculos - Patiotuercas	24
TABLA # 4.....	34
Clase Persona.....	34
TABLA # 5.....	34
Clase Vendedor.....	34
TABLA # 6.....	35
Clase Comprador	35
TABLA # 7.....	35
Clase Producto.....	35
TABLA # 8.....	35
Clase Vehiculo	35
TABLA # 9.....	36
Clase Repuesto.....	36
TABLA # 10.....	36
Clase Proforma	36
TABLA # 11.....	37
Clase Item	37

TABLA # 12.....	40
Escenario	40
TABLA # 13.....	51
Diccionario de Datos	51
TABLA # 14.....	69
Tabla Categoria.....	69
TABLA # 15.....	69
Tabla Tipo	69
TABLA # 16.....	69
Tabla Marca	69
TABLA # 17.....	70
Tabla Modelo.....	70
TABLA # 18.....	70
Tabla Tipo Identificacion.....	70
TABLA # 19.....	70
Tabla Identificacion	70
TABLA # 20.....	70
Tabla Identificación.....	70
TABLA # 21.....	70
Tabla Directorio	70
TABLA # 22.....	71
Tabla Persona	71
TABLA # 23.....	71
Tabla Productos	71
TABLA # 24.....	72
Tabla Proforma.....	72

TABLA # 25.....	72
Tabla Proforma Item.....	72
TABLA # 26.....	72
Tabla Orden	72
TABLA # 27.....	72
Tabla Ordem Item	72
TABLA # 28.....	73
Tabla Pago.....	73
TABLA # 29.....	73
Tabla Vigencia.....	73
TABLA # 30.....	74
Funcion Producto Buscar	74
TABLA # 31.....	75
Metodo Be	75
TABLA # 32.....	76
Metodo Bo.....	76
TABLA # 33.....	77
Metodo Dao.....	77
TABLA # 34.....	78
Conexion	78
TABLA # 35.....	81
Control de Calidad.....	81
TABLA # 36.....	85
Pruebas de Integridad	85

CAPITULO 1

PORTAL DE VENTAS CORPORATIVAS DE VEHICULOS POR INTERNET

1.1 Antecedentes

Por medio de la tecnología hoy en día hay muchas empresas que se dedican a la venta de productos y estas lo exhiben por medio de Internet, la cual ha tenido mucho éxito ya que estos son reconocidos local y mundialmente.

En la actualidad las personas podemos conocer los diferentes modelos de vehículos y al mismo tiempo, precio y características que ofrecen las empresas sin necesidad de acercarse a la concesionaria.

Existen muchos portales que promocionan vehículos de diferentes personas ya sea natural o jurídica pero estas carecen de información y no son conocidas en el mercado.

1.2 Problemática

Para obtener un vehículo los usuarios deben interactuar con algunas páginas Web para ver las diferentes marcas de los vehículos o en otras circunstancias las personas se acercan a la concesionaria para conocer los diferentes tipos de vehículos, costos u otra información de estos, motivo por el cual causa desinterés por parte del usuario.

Otro problema es la poca información del servicio y paginas con una interfaz poco amigable para el usuario.

1.3 Solución a la problemática:

Una solución a estas problemáticas que se presentan hoy en día, sería crear un sitio Web donde el usuario pueda ver los vehículos, de los

diferentes tipos de marcas, modelos y costos de diferentes concesionarias en un mismo portal. Además con este servicio que le ofrecemos al mercado automotriz se obtendrán ventajas competitivas.

Crear un portal Web que muestre productos con su información actualizada y además información referente a todas los propietarios que han contratado nuestro servicio, esto evitara que las personas vayan a la concesionaria o patios de ventas de vehículos de segunda mano, solo a ver catálogos o modelos de vehículos respectivamente.

Otra solución primordial es motivar al usuario de tal manera que adquiera nuestro servicio dándole a conocer cuál es nuestra diferencia frente a la competencia. Tener una mejor publicidad a nuestro servicio con el fin de ganar el mercado.

Un portal donde los usuarios puedan interactuar y expresar sus dudas o recomendaciones en cuanto al servicio que se ofrece y como también expresar que necesidades no están siendo satisfechas.

1.4 Misión

La misión de nuestro proyecto es darles a los usuarios y/o cliente la mejor información en cuanto a los diferentes vehículos que venden las diferentes concesionarias, así como también los datos referentes a estas, este portal estará orientado a un alto rendimiento, confort y diseño atractivo, cumpliendo con las expectativas y dudas de nuestros clientes.

1.5 Visión

Este portal brindará una mejor calidad en la atención de los clientes para así llegar a ser una de las mejores empresas que presten buen servicio de ventas por medio de la Web y como consecuencia obtener favorables ganancias. Será un sitio Web dinámico, innovador y con seguridad confiable en cuanto a la información que este contenga.

1.6 Objetivos

1.6.1 Objetivos Generales

Automatizar el servicio de ventas automotriz, para así dar facilidad de que cualquier usuario o cliente pueda conocer los diferentes vehículos que existen en el mercado, de una forma cómoda y ágil sin necesidad de asistir a un punto de venta.

1.6.2 Objetivos Específicos

- Obtener una Interfaz amigable e información necesaria acerca de los vendedores y facilidad de búsqueda de los productos a promocionar.
- Mantener actualizada la información de los productos.
- Brindar búsquedas específicas.
- Obtener una interfaz dinámica donde el vendedor pueda ingresar toda la información necesaria que le permita promocionar de la mejor manera sus productos.
- Ofrecer mejores precios que las ya existentes en este mercado.
- Promocionar repuestos.
- Receptar datos de los usuarios que desean solicitar un vehículo o repuesto.
- Permitir cotizar e imprimir proformas.
- Ofrecer mecanismos de pagos que den seguridad y confianza.
- Lograr posesionarnos en el mercado automotriz.

1.7 Alcances

- Obtendremos un sitio en la Web en la que los usuarios puedan obtener información oportuna y gratuita de una manera rápida y segura de los vehículos y datos necesarios de las los propietarios que promocionen los productos, por lo que nuestro sistema contará con una interfaz amigable para que el cliente pueda entender y manipular correctamente la información brindada.
- Tendremos una base de datos amplia y actualizada, referente a modelos, marcas y repuestos de vehículos que serán proporcionados por los vendedores, con sus respectivos costos.
- Actualizar periódicamente la información referente a los productos que se promocionan y de los vendedores.
- Se le permitirá al usuario hacer búsquedas más profundas, de acuerdo a los criterios que ellos deseen, como pueden ser por concesionaria, tipo, marca, modelo, precio y año.
- Se proporcionara un formulario fácil de manipular para cualquier usuario especificando los pasos a seguir para que así publique su vehículo sin ningún inconveniente.

- El gran beneficio que tiene nuestro sitio Web, es proporcionarle al mercado automotriz mejores ofertas creando así mas competitividad. Por parte de los usuarios que deseen comprar vehículos, se les ofrece el beneficio de que se eviten visitar varias páginas para consultar diferentes tipos de marcas, ya que en este portal encontrará varias marcas y modelos.
- Se ofrecerá varios paquetes donde el vendedor pueda escoger el que mejor le convenga, se contarán con dos opciones:
 - Para ingresar un solo vehículo tendrá un costo fijo.
 - Si es más de dos vehículos se contarán con paquetes que son:
 - Estático: rango de vehículos limitado, rango de tiempo, no podrá alterar la publicidad una vez levantada.
 - Dinámico: rango de vehículos limitado, rango de tiempo, podrá alterar la publicidad una vez levantada.
 - Ilimitado: Sin rango de vehículos, rango de tiempo, podrá alterar la publicidad una vez levantada.
- El portal Web se diferenciará con los ya existentes, en cuanto a que promocionaremos repuestos para vehículos como lo hemos mencionado anteriormente.

- Se permitirá realizar cotizaciones por parte de los usuarios tanto comprador como vendedor, cada usuario selecciona los productos o paquetes a su elección y esta automáticamente creara una proforma con los datos que el usuario ha ingresado al momento de registrarse y por ende podrá imprimir el documento.
- El cliente vendedor deberá imprimir su orden de pago, la que será cancelada en nuestras oficinas o a través de una entidad Bancaria que ofrezca el servicio OCP.
- Realizaremos ofertas por temporada, posesionarnos entre los primeros en los buscadores más reconocidos. Realizar publicidad en internet, radio, televisión o revistas.

1.8 Recursos

Detallaremos a continuación los recursos de hardware. Software, humano que se requiere para la implementación de nuestro proyecto:

- Investigación Internet y/o Tutoría proporcionada por la Carrera.
- Equipos, dispositivos:
 - Laptop IBM, Intel R Pentium ® Procesador 1600 MHZ. 512 MB de Memoria RAM. (CLIENTE).

- PC Pentium IV Intel Core2 duo, 2 GB Memoria Ram.(SERVIDOR).

- Procesador:
- RAM:
- Disco Duro:
- Tarjeta de Red

- 2 Impresoras Samsung ML-1610, Lexmark multifunción.
- Laboratorio de Graduación de CISC
- 3 Pen drivers

- Personal del Proyecto:

- Ruth Moreira Carvajal
Conocimientos en Java y Diseño Web.
- María Pamela Pin
Conocimientos en Java y Base de Datos Oracle.
- Diana Ronquillo Rodríguez
Conocimientos de Base Datos Oracle y Diseño Web.

- Herramienta para el desarrollo de la aplicación:

- Editor de Java Eclipse.
- J2EE/ PHP.
- Base de datos PostgreSQL.
- Servidor Web (Apache)

- FrontPage
 - Editor de fotos ADOBE
 - Animación .GIF
 - Herramienta de Reporte iReport
 - Herramienta de Análisis y Diseño (NetObjeting)
- Sistema Operativo:
 - Windows XP Profesional,
 - Windows XP Professional service pack 2

1.9 Arquitectura

En aplicaciones web suelen encontrarse con importantes problemas de escalabilidad, disponibilidad, seguridad, integración. Para evitar estos problemas se requiere la división de las aplicaciones en capas que en este caso utilizaremos la arquitectura de tres capas:

Una capa para la **interfaz gráfica** que facilite al usuario el uso del sistema (IU), que es el BrowseHTML, quien envía la información al servidor web y también muestra las páginas HTML enviadas por el servidor web, en nuestro caso se podrá utilizar Internet Explorer, Netscape y mozilla.

Una segunda capa, para centralizar la **lógica de negocio** (modelo o aplicación) es decir los procesos Java que son los encargados de manejar esta lógica, en este caso utilizaremos J2EE y PHP; y, el **servidor web** es el que procesa, coordina y ensambla la información enviada a través de las páginas web, aquí utilizaremos Apache. Por último capa que servirá para guardar los **datos** (base de datos), en donde almacenaremos datos referentes a los vehículos, repuestos, clientes (comprador-vendedor) u otra información que nos serán útil. en nuestro caso utilizaremos la Base de Datos PostgreSQL

Ésta es la distribución en capas más común para las aplicaciones web.

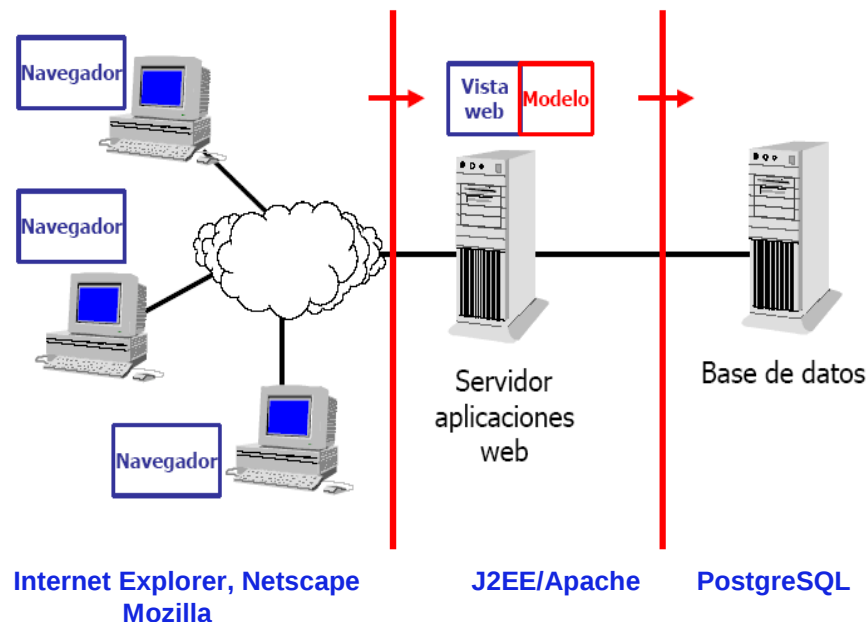


Gráfico 1
Arquitectura del Sistema

1.10 Cronograma

Con esta programación de tareas, lograremos cumplir con nuestros objetivos y alcances.

Tareas	Mayo				Junio				Julio				Agosto				Sept				Oct				Nov				Dic			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Introducción																																
Análisis																																
Especificación de los Requisitos del Software																																
Casos de Usos																																
Diagramas de Clases																																
Diseño																																
Escenario																																
Diagramas de Secuencia																																
Diseño de la Interfaz de Usuario (Prototipo)																																
Implementación y Desarrollo																																
Creación de la Base de Datos																																
Creación de las Entidades																																
Conexión a la Base de Datos																																
Lógica del Negocio																																
Diseño del Portal Web																																
Seguridad																																
Pruebas / Depuración																																
Implantación																																
Elementos físicos																																
Suscripción de Dominio																																
Hosting																																
Suscripción a Buscadores																																
Control / Evolución																																

Tabla 1
Cronograma

1.11 Modelo

Se ha escogido el MODELO ESPIRAL para el desarrollo del proyecto debido a que este posee una combinación del Modelo Lineal y el de Construcción de Prototipos, proporcionando el potencial desarrollo rápido de versiones del software.

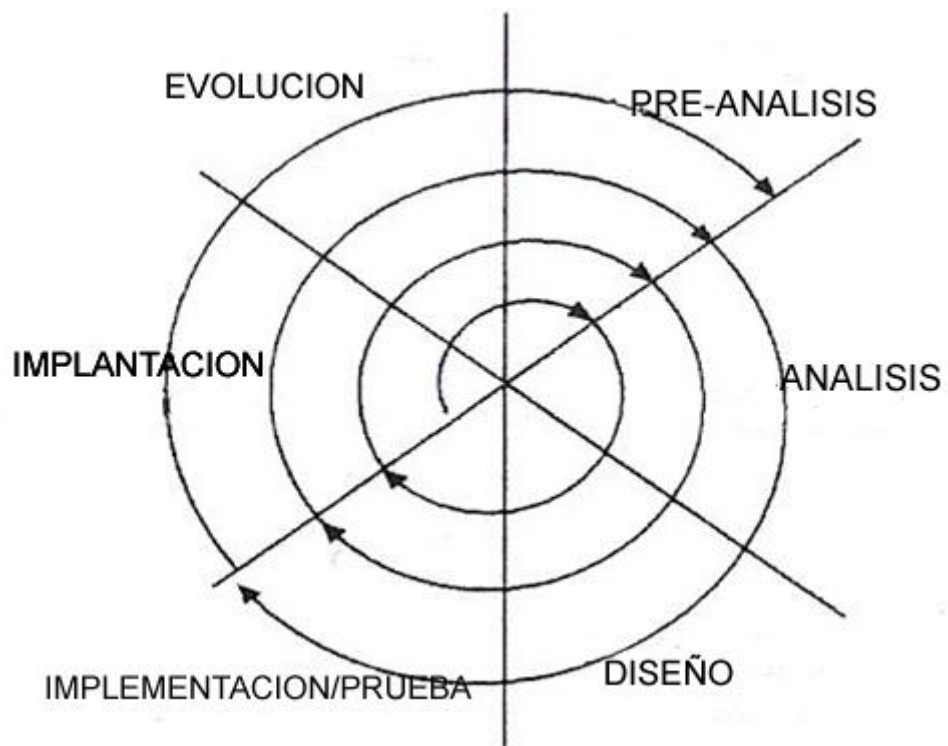


Gráfico 2
Modelo de Análisis

Este modelo está compuesto de Análisis, Diseño, Implementación, Implantación y Evolución.

1.12 Metodología

Para el proyecto “Portal de ventas Corporativas de Vehículos por Internet” se ha escogido la tecnología orientada a objetos ya que esta tiene una mayor disciplina que el de las técnicas de estructura convencional. Esto nos lleva a un mundo de clases reutilizables, donde la mayor parte del proceso de construcción de software consistirá en el ensamblaje de clases ya existentes y probadas.

Las actividades que van a efectuarse para alcanzar la culminación del proyecto son:

➤ Investigación Preliminar.

- ✓ Consultas Bibliográficas.
- ✓ Consultas por Internet.
- ✓ Consultas a Profesionales.
- ✓ Ámbito de estudio.

➤ Análisis

- ✓ Levantamiento de la información
- ✓ Análisis de los Requerimientos
 - Análisis de recursos humanos
 - Análisis de los Recursos Hardware y Software

- ✓ Estudio de factibilidad
 - Análisis costo beneficio
 - Factibilidad técnica
 - Factibilidad económica
 - Factibilidad operacional..
- ✓ Análisis Orientados a Objetos
 - Especificación de los Requisitos del Software (SRS).
 - Casos de Usos
 - Diagramas de Clases

➤ **Diseño**

- ✓ Escenario
- ✓ Diagramas de Secuencia
- ✓ Diseño de la Interfaz de Usuario (Prototipo)

➤ **Implementación y Desarrollo**

- ✓ Creación de la Base de Datos
- ✓ Creación de las Entidades
- ✓ Conexión a la Base de Datos
- ✓ Lógica del Negocio
- ✓ Diseño del Portal Web (Interfaz Gráfica del Usuario)
- ✓ Seguridad
- ✓ Pruebas / Depuración

➤ **Implantación**

- ✓ Elementos físicos
- ✓ Suscripción de Dominio
- ✓ Hosting
- ✓ Suscripción a Buscadores
- ✓ Control / Evolución

➤ **Conclusión y Recomendación.**

CAPITULO 2

ANALISIS

En este capítulo llevaremos a cabo la fase de análisis de todo este sistema **PORTAL WEB**, comenzando por las partes más generales del mismo, como puede ser la especificación de requerimientos, para poco a poco ir desarrollando en forma gradual los componentes de nuestro sistema.

2.1 Levantamiento De Información

La recopilación de información debe estar enfocada en los hechos que permitan conocer y analizar información específica y sobre todo útil para el proyecto y su desarrollo, caso contrario obtendremos interpretaciones erróneas, con lo que nos produciría retraso y desperdicio de recursos.

Para este proceso se utilizó dos formas de levantamiento de información: las entrevistas y la revisión de documentos. En relación a las entrevistas, se las realizó a las personas que utilizaran el portal, tales como compradores y vendedores de vehículos; y, en cuanto a revisión de documentos, se hizo la selección y análisis de aquellos escritos que contienen datos de interés relacionados con el proyecto, por medio de internet y de consultas a diversos proyectos en la biblioteca de la Carrera, también se analizó a la competencia visitando sus portales web.

2.1.1 Investigación de la Competencia

En el Ecuador existen dos competidores fuertes ya establecidos en lo que a venta de carros se refiere. El uno es Autosecuador y el otro es patiotuerca.com, de Guayaquil y Quito respectivamente, el primero vende una revista y como adicional

publica en la web, el otro tiene su negocio orientado a ser 100% web.

Ambos salen primeros en las búsquedas de Google cuando de buscar autos en el ecuador se refiere, e incluso los de patio tuerca van más allá puesto que si uno busca, Lada Ecuador o Skoda Ecuador, por ejemplo, aparecen primeros.

Patituerca tiene banners de Chevrolet, Audi, Mazda y otros auspiciantes de renombre.

Patio tuerca tiene su fuerza establecida en más de 30 patios a nivel de la sierra, y casi ninguno de la costa, en la página principal da acceso a publicaciones de autos de los patios registrados.

Características

Las características que vamos a tomar en gran parte son de patio tuerca debido a que ellos disponen de una interfaz muy bien elaborada.

- ✓ **Evaluador de precios.** Es una utilidad que pide tres parámetros marca, modelo y precio; y, devuelve el valor del mercado. Este valor lo sacan de un historial de las publicaciones de ese vehículo (ver Gráfico # 3).

http://www.patiotuerca.com - Mozilla Fir...

Evaluador de Precios

¿Sabe Cuanto Cuesta Su Vehículo En El Mercado?

Gracias a nuestro sistema evaluador de precios usted podrá saber por donde va el mercado.

Ingrese los criterios que más se aproximen a su vehículo:

Marca:
Mazda

Modelo:
323

Año:
1992

Cantidad: 1 Encontrados
Máximo: \$3600
Mínimo: \$3600
Promedio: \$3600

Precio Recomendado: \$3600

¿Como funciona?

Este cálculo se basa en la información de los vehículos publicados y dados de baja de los últimos meses.

VENDA SU VEHÍCULO YA!!!

Listo

Gráfico 3
Evaluador de Precios - Patiotuerca

- ✓ Permite suscribirse a un boletín que envían de vez en cuando para anunciar noticias referentes al sitio.

http://www.patiotuerca.com - Mozilla Firef...

Suscríbase **GRATIS**
al **Boletín Tuerca**

para obtener: noticias, novedades,
promociones y lo mejor del mundo tuerca a
su email.

Usted podrá darse de baja en cualquier
momento y su información
permanecerá 100% privada en nuestro
servidor. Registrarse es rápido y fácil.

**Ingrese su nombre y su email, luego
haga clic en "Enviar":**

Nombre:

Email:

Listo

Gráfico 4
Suscripción a Boletines - Patiotuerca

Luego haber enviado su nombre y dirección de correo, como muestra el **gráfico# 4**, llega un mensaje similar a este:

Hola Diana Ronquillo

Para activar su cuenta del Boletín Tuerca requerimos confirmar que esta es su dirección de e-mail y que nos concede el permiso para enviarle información. Por favor haga clic en el link a continuación para activar su cuenta. Si ya es suscriptor haga clic en el link de todas formas, estamos en proceso de depuración de la base de datos.

CONFIRM BY VISITING THE LINK BELOW:
<http://www.aweber.com/z/c/?znyc3cxb2di6mzg29s50xva2db3c7q2105>

Si no desea suscribirse, simplemente ignore este mensaje.

Gracias,

Patio Tuerca
 Av. La Prensa 2017 y Nicolás López esquina

Tabla 2
Mensaje Suscripción a Boletines - Patiotuerca

- ✓ **Tienda Tuerca.** Es un sitio de comercio electrónico, aquí se expende productos orientados al Tuning Car, ellos no han implementado pagos en línea, sólo órdenes de pedidos.
- ✓ **Secciones Informativas.** Tiene secciones informativas tales como: eventos, historia, calendario de eventos, directorio, entretenimiento.

Bikers Weekend 2006 (01/09/2006)

4ta Trepada de Montaña (03/09/2006)

Enduro Rally por equipos (16/09/2006)

Panamerican 20 000, el desafío de Ferrari (20/09/2006)

+ eventos

Tienda Tuerca *

Alarmas

Amplificadores

Luces

Parlantes

Radios

+ productos

Directorio *

Accesorios

Car Audio

Lavadoras

Mecánicas

Pintura

Motos y Accesorios

+ empresas tuerca

Entretenimiento *

Wallpapers

Chat

+ Juegos

Crea tu vehículo

Bienvenido a Patio Tuerca, el Portal Tuerca más visitado del Ecuador.

Evento

Panamerican 20 000, el desafío de Ferrari (09/20/2006): En los próximos días, Ecuador podrá mirar en vivo y en directo a los emblemáticos Ferrari 599 GTBs Fiorano, Autodelta, distribuidora de los vehículos ...



Historia

Citroen AX: El citroen AX inició su producción desde 1986 hasta 1998, este auto porto 11motores diferentes desde 1000 cc hasta 1500cc, con una potencia que varia... (08/30/2006)



Más de 1000 vehículos usados a la Venta

Acura (1)	Alfa Romeo (7)	Audi (12)
Bajaj (1)	BMW (60)	Changhe (4)
Chevrolet (309)	Chrysler (7)	Citroen (4)
Daewoo (18)	Daihatsu (11)	Datsun (1)
Dodge (5)	Ducati (2)	Fiat (9)
Ford (85)	Freightliner (1)	GMC (2)
Harley Davidson (1)	Hino (7)	Honda (17)
Honda - Motos (6)	Humber (1)	Hummer (1)
Hyundai (36)	International (5)	Isuzu (2)
JAC (3)	Jeep (39)	Kawasaki (5)
Kia (52)	KTM (2)	Lada (5)
Lancia (1)	Land Rover (12)	Lexus (1)
Mack (1)	Mazda (36)	Mercedes Benz (83)
Mini (5)	Mitsubishi (40)	Motor Uno (3)
Nissan (47)	Oldsmobile (1)	Otros (5)
Peugeot (27)	Porsche (3)	Renault (27)
Skoda (14)	Subaru (3)	Suzuki (11)
Tank (1)	Toyota (56)	Volkswagen (115)
Volvo (28)	Yamaha (3)	

Lista de Datos Afiliados

Suscribase GRATIS al BOLETÍN TUERCA

Es la mejor forma de mantenerse conectado con lo mejor de Patio Tuerca. [Registro Gratis.](#)

Consulte GRATIS nuestro Evaluador de Precios



Venda su Auto, Moto, Camión, etc.

Buscar Vehículos Usados

Listas de precios Autos Nuevos

Productos y Servicios Tuerca

Cupones Tuerca (Beneficios)

Patios Tuerca Afiliados

Evaluador de Precios de Vehículos

Consejos para comprar o vender



Lo Último

F1 2006 - Turquía, Felipe Massa se ha alzado esta tarde con la primera victoria de su carrera en la F1 tras dominar el GP de Turquía y cruzar la línea de meta con cinco segundos de ventaja sobre su más inmediato rival, Más..

4ta Ncional de Rally, Parece que la mala suerte persigue a esta provincia

Gráfico 5
Sección Informativa - Patiotuerca

- ✓ **Vehículos Top.** Un listado de vehículos aparecen en la página principal de manera privilegiada, Obviamente para que el vehículos este en esta sección hay que pagar un coste adicional.
- ✓ **Búsqueda de autos.** Se lo hace de 2 maneras, por marcas o por texto de búsquedas. En la primera opción uno hace clic a la marca en la página principal y le dirige a una página donde aparecen los autos de esa marca.

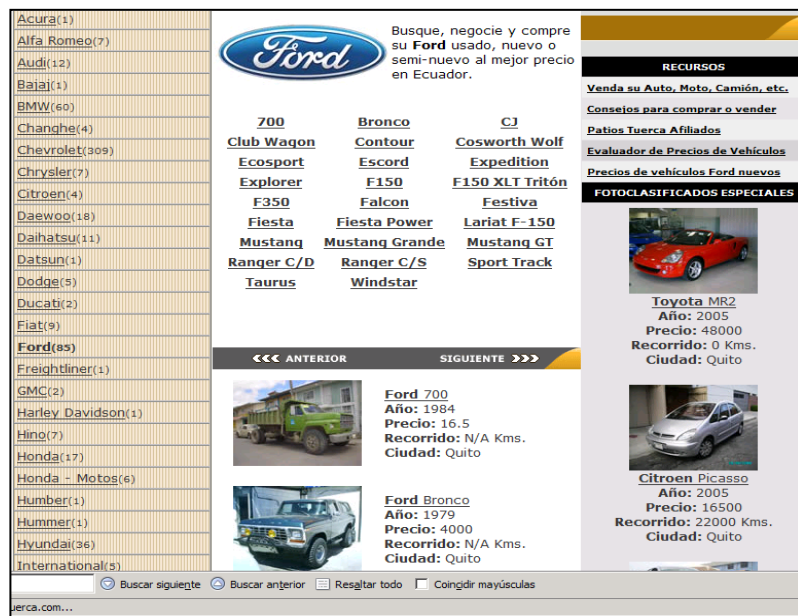


Gráfico 6
Búsqueda Autos - Patiotuerca

- ✓ **Costo de Publicación:** Tienen 3 niveles de precios:
 1. por 30 días con foto 6.99
 2. Por 90 días con publicación en home 13.99
 3. Hasta que se venda con inclusión en boletines 22.99

- ✓ Muestra información de modelos de las diferentes marcas de vehículos.

Patios Tuerca Afiliados
Evaluador de Precios de Vehículos
[Acura \(1\)](#)
[Alfa Romeo \(7\)](#)
[Audi \(12\)](#)
[Bajaj \(1\)](#)
[BMW \(60\)](#)
[Changhe \(4\)](#)
[Chevrolet \(309\)](#)
[Chrysler \(7\)](#)
[Citroen \(4\)](#)
[Daewoo \(18\)](#)
[Daihatsu \(11\)](#)
[Datsun \(1\)](#)
[Dodge \(5\)](#)
[Ducati \(2\)](#)
[Fiat \(9\)](#)
[Ford \(85\)](#)
[Freightliner \(1\)](#)
[GMC \(2\)](#)
[Harley Davidson \(1\)](#)
[Hino \(7\)](#)
[Honda \(17\)](#)
[Honda - Motos \(6\)](#)
[Humber \(1\)](#)
[Hummer \(1\)](#)
[Hyundai \(36\)](#)
[International \(5\)](#)



Explorer

Ver solo de: [Todos \(23\)](#) | [Quito \(22\)](#) | [Guayaquil \(1\)](#)
Ordenar por: [Fecha de publicación](#) | [Año](#) | [Precio](#)

Nueva funcionalidad: Ahora con la ayuda de nuestros filtros es más fácil encontrar el vehículo de sus sueños. Haga clic en las ciudades o en los criterios de ordenamiento: por Año, Precio o fecha de publicación.



Ford Explorer
Año: 1997
Precio: 10000
Recorrido: 110000 Kms.
Ciudad: Quito



Ford Explorer
Año: 1998
Precio: 11500
Recorrido: 78.000 Kms.
Ciudad: Quito



Ford Explorer
Año: 2004
Precio: 26000
Recorrido: 38000 Kms.
Ciudad: Quito



Ford Explorer
Año: 2003
Precio: 19300
Recorrido: 48000 Kms.
Ciudad: Quito



Ford Explorer
Año: 1992
Precio: 7800
Recorrido: 110000 Millas
Ciudad: Quito



Ford Explorer
Año: 2002
Precio: 22000
Recorrido: 150000 Kms.
Ciudad: Guayaquil



Ford Explorer



Ford Explorer



Ford Explorer

Gráfico 7
Búsqueda Modelos por Marcas - Patiotuerca

- ✓ Por cada vehículo muestra las siguientes características:

Tipo*:	Jeep	Marca*:	Ford
Modelo*:	Explorer	Precio*:	\$ 12000
Año:	1998	Recorrido*:	135000Kms.
Color:	Verde	Motor (Cilindraje):	4000cc
Transmisión:	Manual	Aire Acondicionado:	Sí
Vidrios:	Eléctricos	Tapizado:	Tela
Dirección:	Hidráulica	Tracción:	4 x 4
Placa:	Pichincha	Combustible:	Gasolina
Ensamblado en:	N/A		
Extras:  Airbag  Alarma  Bloqueo Central			

Tabla 3
Características de Vehículos - Patiotuerca

En cuanto al portal de autosecuador.com, como lo comentamos anteriormente no tiene una interfaz tan bien elaborada como la tiene Patiotuerca.com, por esta razón no profundizamos la investigación que hicimos sobre austosecuador.

A continuación mostramos unas cuantas pantallas de este portal:



Gráfico 8
Búsqueda Autos - Autosecuador

- Como podemos apreciar ofrece poca información de los vehículos.
- Este portal posee muy pocos banners, en comparación a patiotuercas.
- Formas de Pago, se lo puede hacer por 2 formas: directamente en las oficinas o vía on line (tarjeta de crédito).

- El listado de precios de los autos, se muestra de una manera un poco desorganizada.

Home Búsqueda Avanzada Contacto Inscribir Vehículos Financiamiento Autos Nuevos							
LISTADO DE PRECIOS DE AUTOS NUEVOS							
AUDI		Corsa 4P Wind 1.6 l	\$11,460	Ranger DC 4x4	\$ 23,750	Montero 3P 3.0l Col	\$ 23,990
A4 1.8 T Multitronic	\$47,000	Corsa 4P STD 1.7 l Diesel	\$12,680	Ranger RC 4x4	\$ 20,100	Montero 3P 3.8 TA	\$ 52,400
A4 1.8 Multitronic Mec.	\$49,800	Camiones		Sport Track XLT 4x4 (USA)	\$ 38,990	Montero 5P 3.0l Col	\$ 28,990
A4 3.0 Multitronic premium TA 220 HP /04	\$49,900	NHR Chasis Cabinado	\$18,390	HONDA		Montero 5P 2.8.L Diés. T/A	\$ 49,990
A6 3.0 Multitronic 220HP sin WCU package	\$68,000	NPR 71 l Chasis Cab.	\$26,990	Accord EX T/A (EE.UU.)	\$ 39,500	Montero 5P 3.5 MT	\$ 47,500
A6 2.4 Multitronic 178HP con WCU package	\$69,900	NKR Chasis Cabinado	\$22,590	Accord Ex T/M (EE.UU.)	\$ 37,500	Montero 1.0 5P M/T	\$ 24,990
A6 2.4 Multitronic 178HP sin WCU package	\$65,500	NKR Chasis Torpedo	\$20,500	Civic (Brasil)	\$ 27,790	Montero Sport 4x4 A/T	\$ 49,900
Q7 3.6	\$85,000	NPR 71P Chasis Torpedo	\$26,130	CR-V EX SE 4x4 (Japón)	\$ 35,890	Montero Sport 4x4 M/T	\$ 41,989
Q7 4.2	\$106,000	CHRYSLER		CR-V LX 4X2 (Japón)	\$ 29,690	Montero 4wd Semi full T/M	\$ 37,509
						Fortuner 4x4 T/A	

Gráfico 9
Lista de Precios - Autosecuador

- Después de haber registrado el o los vehículos, se genera una orden de pago.
- Autosecuador brinda información acerca de los financiamientos que ofrecen ciertos bancos.

Financiamiento para Vehículos Nuevos o Usados								
Banco	Teléfono	Requisitos	Financ. Hasta	Años Plazo Hasta	Interés desde	Seguro Desgrav. Gratuito	Beneficios	Restricciones
Global	(04) 2207371	Copia de cédula de identidad y certificado de votación. Edad: 22 a 65 años certificado de trabajo e ingresos. Estabilidad laboral	70%	4	12.17%	si	Financiamiento del Seguro del vehículo. Trámites legales incluidos. Se financian vehículos nuevos y usados. 2 primeros meses de gracia.	Los vehículos usados desde el año 2000.
Produbanco	(04) 2534170	Copia de la C.I. y papeleta de votación. Estado de situación personal. Copia de respaldos patrimoniales.	70%	4	12.10%		Premiará a la puntualidad en los pagos, sin cobrar los intereses de las cuotas 12,24,36,48. Financiamos vehículos nuevos y usados.	Los vehículos usados deben ser del 2002 en adelante.
							Vehículo nuevo sin límite	

Gráfico 10
Financiamiento para Vehículos - Autosecuador

Ventajas del sitio patiotuerca.com

- Tiene mucho auspicio de patios y concesionarias de la sierra.
- Tiene respaldo de marcas de vehículos nuevos.
- Tiene mucho contenido adicional, como artículos de vehículos y de automovilismo.
- Tiene una cantidad considerable de autos publicados.
- Están bien posicionados en google, y es de fácil acceso en el web.

Desventajas del sitio Patiotuerca.com

- Solo está en Ambato y Quito, esta focalizado a servir donde tiene oficinas
- No hay un servicio dinámico con respecto al vendedor, no le da información relevante que le permita vender mejor su vehículo
- El sistema de búsqueda es muy escueto y los resultados no permiten una visión ágil.
- Los Patios deberían tener un servicio dinámico, y no lo hay.
- Los precios son muy elevados.
- No hay comentarios ni sugerencias del vehículo en la web.

2.2 Análisis De Los Requerimientos

2.2.1 Análisis De Recursos Humanos

Los usuarios que van a tener acceso a este portal, serán las personas que navegan en la web y que están interesadas en obtener algún vehículo; y, también las empresas y/o concesionarias que desean vender sus vehículos a través de nuestro portal web.

Para estos usuarios no será necesario que tengan conocimientos avanzados de internet ni de administración de sistemas.

2.3 Estudio de la Factibilidad

El estudio de factibilidad utilizado para efectos de nuestro proyecto se basa en: Factibilidad Técnica, Económica y Operacional. A continuación evaluaremos cada uno de estos aspectos en el análisis de Beneficio Costo, estimando el impacto financiero acumulado de lo que queremos lograr en este proyecto.

2.3.1 Análisis Costo Beneficio

El análisis de costo-beneficio de este proyecto es muy sencillo, ya que para el desarrollo de este proyecto se va a utilizar en su

mayoría herramientas OpenSource, aunque para la implantación si incurriremos en gastos.

2.3.1.1 Factibilidad Técnica

Este Proyecto es factible técnicamente, ya que contamos con la tecnología que nos facilita la carrera CISC, con los equipos que obtuvimos para el desarrollo de este portal y sobre todo con lo que hoy en día nos proporciona el mercado web.

2.3.1.2 Factibilidad Económica

Tiene como objetivo central determinar las características económicas del proyecto, para ello es necesario identificar las necesidades de inversión, ingresos, los costos, gastos u otros; así determinar si al vender el productos nos genere un margen de utilidad.

2.3.1.3 Factibilidad Operacional

Esta dada por los recursos humanos disponibles para el proyecto, operación del sistema y el uso que se le dé una vez que sea instalado. El portal web que

desarrollaremos está calculado para proporcionar una Interfaz eficiente y accesible para que pueda ser utilizado por cualquier usuario sin limitación alguna a través de la web.

2.4 Análisis Orientados a Objetos

2.4.1 Definición de Usuarios del Sistema

- ✓ Cliente Comprador (usuario interesado en comprar vehículo).
- ✓ Cliente Vendedor (empresas/concesionarias o persona natural que desean vender sus vehículos y promocionan estos productos en este portal).

2.4.2 Especificaciones de los Requisitos del Sistema

- Los usuarios web podrán visitar la página principal (Home).
- Los usuarios web no registrados pueden acceder al sitio web para consultar información general y detallada de vehículos y repuestos y paquetes promocionales, pero no podrán cotizar.
- La consulta de vehículos y repuestos utilizará como criterios de búsqueda: vendedor, categoría, marca, modelo, precio y año.

- Se ofrecen los siguientes paquetes publicitarios a los clientes que desean publicar 2 o más productos:
 - Estático: rango de vehículos limitado, rango de tiempo, no podrá alterar la publicidad una vez levantada.
 - Dinámico: rango de vehículos limitado, rango de tiempo, podrá alterar la publicidad una vez levantada.
 - Ilimitado: Sin rango de vehículos, rango de tiempo, podrá alterar la publicidad una vez levantada.
- Los afiliados se dividen en: persona natural y persona jurídica.
- Los clientes vendedores y compradores no registrados que deseen cotizar deberán registrarse por medio de un formulario en el que se solicitará los datos básicos: nombre, dirección, teléfono, correo electrónico, etc.
- Se enviará una notificación por correo a los clientes recientemente registrados confirmando su ingreso.
- Los clientes registrados deberán autenticarse utilizando un nombre de usuario y contraseña.
- Los afiliados pueden cotizar los productos ofrecidos en la página o adquirir paquetes publicitarios para inscribir vehículos y repuestos.

- Los clientes vendedores registrados podrán cotizar paquetes publicitarios.
- Los afiliados podrán descargar y/o imprimir las proformas.
- Los afiliados podrán elaborar ordenes de pedido de los servicios publicitarios para publicar vehículos y repuestos.
- El pago se realizará a través de entidades bancarias vinculadas a nuestra empresa, que ofrezcan el servicio de recaudación, presentando la orden de pedido impresa.
- La activación de los servicios publicitarios se realizará en cuanto la entidad bancaria notifique la cancelación (Estado de cuenta OnLine).
- El afiliado podrá incluir su listado de vehículos y repuestos a ofrecer cuando haya generado su orden de pago y una vez cancelado se publicará el productos.
- Se dispondrá de las siguientes alternativas para el registro de productos:
 - Formulario simple (uno a uno).
 - Formulario tabular.
 - Levantar archivo con formato XML (opcional).

2.4.3 Diagrama de clases

Los diagramas de clases representan la estructura de objetos del sistema y las relaciones entre ellos, su objetivo es representar aspectos estáticos del sistema.

Aquí definiremos las características de cada una de las clases, interfaces, colaboraciones y relaciones de dependencia y generalización. Es decir, es donde daremos rienda suelta a nuestros conocimientos de diseño orientado a objetos, definiendo las clases e implementando las ya típicas relaciones de herencia y agregación.

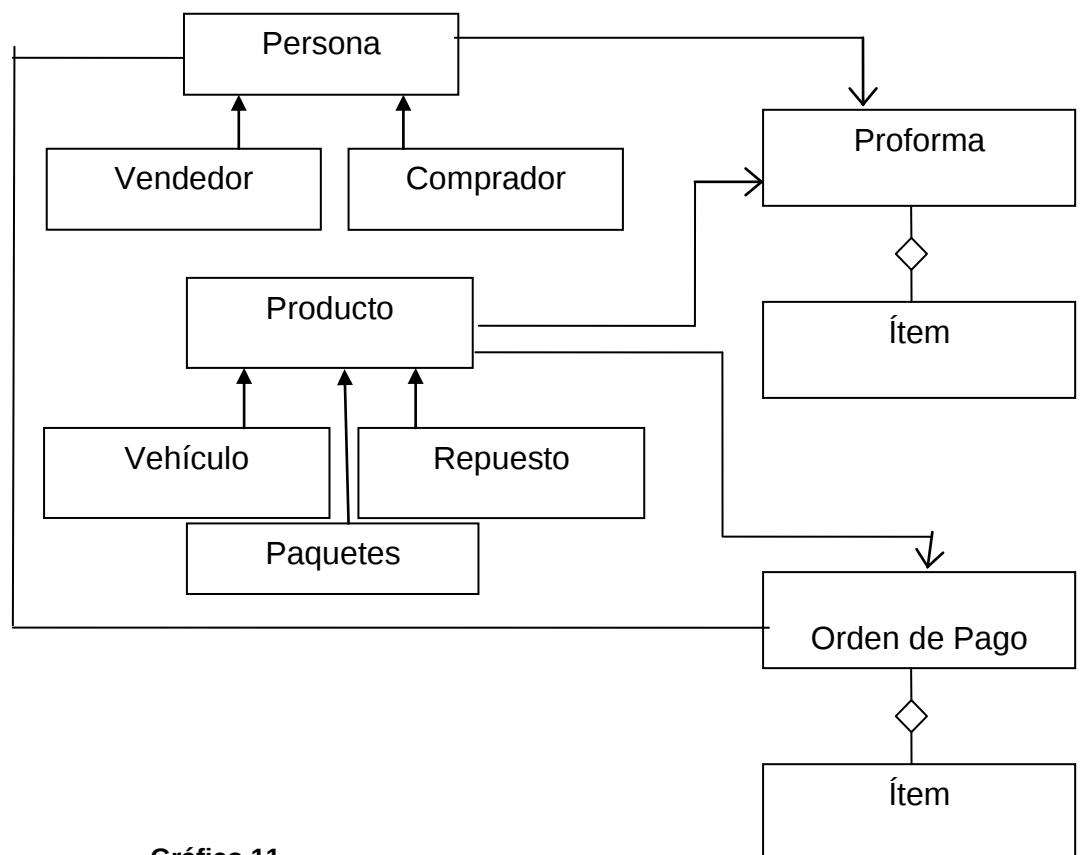


Gráfico 11
Diagrama de Clases

2.4.3.1 Clases y Atributos O.O.

Aquí describiremos las diferentes clases con sus propiedades y comportamientos en este sistema.

Persona.- Clase que encapsula los datos de las personas o usuarios que se registran en nuestra página.

PERSONA
Id
CiRuc
Nombre
Apellido
Ciudad
Direccion
Telefono
Email
Usuario

Tabla 4
Clase Persona

Vendedor.- Clase que encapsula información relevante sobre las personas jurídicas que se afilian a nuestra página.

VENDEDOR
IdPersona
IdIndustria
Sector
PoBox

Tabla 5
Clase Vendedor

Comprador.- Clase que encapsula información relevante sobre las personas naturales que se afilian a nuestra página.

COMPRADOR
IdPersona

Tabla 6
Clase Comprador

Producto.- Clase que encapsula los datos básicos de los productos que se promocionan en esta página.

PRODUCTO
Id
IdCategoria
IdProveedor
IdMarca
Descripción
Valor

Tabla 7
Clase Producto

Vehículo.- Clase que encapsula información relevante sobre los vehículos.

VEHICULO
IdProducto
Modelo
Tipo
Color
Placa
Año

Tabla 8
Clase Vehículo

Pieza.- Clase que encapsula información relevante sobre los repuestos.

REPUESTO
IdProducto
Tamaño
Marca
Modelo
Nombre
Año

Tabla 9
Clase Repuesto

Proforma.- Clase que encapsula información de la cabecera de la proforma.

PROFORMA
Id
IdEmpresa
IdConcesionaria
IdComprador
Tipo
Secuencia
Fecha

Tabla 10
Clase Proforma

Item.- Clase que encapsula información del detalle de la proforma que incluye vehículos y repuestos.

ITEM
Id
IdProducto
Cantidad

Tabla 11
Clase Ítem

2.4.4 Diagrama de Caso de Usos

Un Diagrama de Casos de Uso muestra la relación entre los actores y los casos de uso del sistema, representaremos la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa. Aquí definiremos como los usuarios interactúan con el sistema es decir que uso pueden darle a este sistema.

Actores:

- Clientes Comprador
- Cliente Vendedor

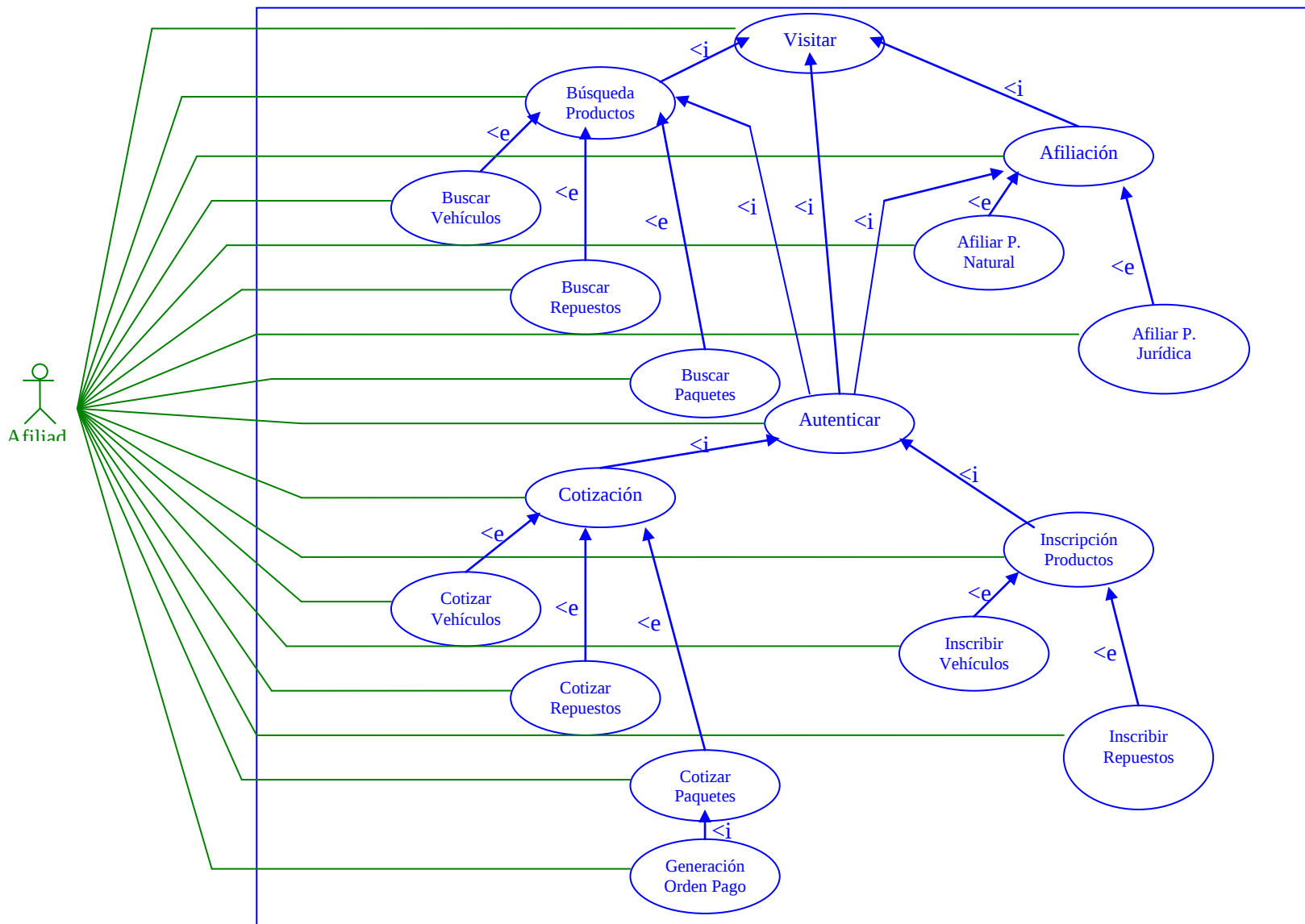


Gráfico 12
Diagrama de Casos de Uso

CAPITULO 3

DISEÑO

En esta etapa trataremos todo a lo referente al diseño del “PORTAL DE VENTAS CORPORATIVAS DE VEHICULOS”, desde el diseño de las interfaces hasta la arquitectura del mismo. Detallaremos los diferentes procesos que podrá realizar los usuarios, tanto clientes compradores como clientes corporativos.

3.1 ESCENARIOS

Visitar	Descripción:	El usuario consulta la página principal.	
	Condiciones Previas:	Ninguna	
	Condiciones Posteriores:	El usuario tiene la posibilidad de realizar la búsqueda de productos, afiliarse o autenticarse.	
Búsqueda de Productos	Condiciones Previas:	El usuario ha consultado la página principal.	
	Descripción:	El usuario escoge si quiere publicar o cotizar vehículos y repuestos para presentar la búsqueda correspondiente.	
	Condiciones Posteriores:	El usuario tiene la posibilidad de realizar la búsqueda de productos de acuerdo a la opción escogida	
Buscar Vehículos	Condiciones Previas:	El usuario escoge la opción de cotizar vehículos y se presenta la búsqueda correspondiente.	
	Descripción:	El usuario introduce la información de búsqueda de vehículos. Los criterios de búsqueda son: tipo, vendedor, año, marca, precio, modelo. Tras enviar la petición de búsqueda, se presenta al usuario una lista de los vehículos que coinciden con el criterio especificado.	
	Condiciones Posteriores:	El usuario tiene la posibilidad de registrarse o autenticarse para acceder a la opción de cotizar.	
Buscar Repuestos	Condiciones Previas:	El usuario escoge la opción de cotizar repuestos y se presenta la búsqueda correspondiente.	
	Descripción:	El usuario introduce la información de búsqueda de repuestos. Los criterios de búsqueda son: tipo, vendedor, año, marca, precio. Tras enviar la petición de búsqueda, se presenta al usuario una lista de los repuestos que coinciden con el criterio especificado.	
	Condiciones Posteriores:	El usuario tiene la posibilidad de registrarse o autenticarse para acceder a la opción de cotizar.	

Buscar Paquetes	Condiciones Previas:	El usuario escoge la opción de cotizar paquetes y se presenta la búsqueda correspondiente.
	Descripción:	El usuario introduce la información de búsqueda de paquetes. Los criterios de búsqueda son: tipo, número de productos, tiempo, precio. Tras enviar la petición de búsqueda, se presenta al usuario una lista de los paquetes que coinciden con el criterio especificado.
	Condiciones Posteriores:	El usuario tiene la posibilidad de registrarse o autenticarse para acceder a la opción de cotizar.
Afiliación	Condiciones Previas:	El usuario ha consultado la página principal y/o realizado la búsqueda de productos y desea acceder a la opción de cotización.
	Descripción:	El usuario tiene que ingresar información para afiliarse a la página. El usuario debe escoger el registro para persona natural o persona jurídica.
	Condiciones Posteriores:	El usuario ingresa la información requerida.
Afiliar P. Natural	Condiciones Previas:	El usuario escoge la opción de registro de persona natural y se presenta el formulario correspondiente.
	Descripción:	El usuario ingresa información para afiliarse. El registro se compone de: nombres, apellidos, tipo de identificación, número de identificación, ciudad, dirección, teléfono, correo electrónico, usuario y contraseña.
	Condiciones Posteriores:	El afiliado recibe un mensaje de confirmación en su correo electrónico con el usuario y contraseña para que se autentique.
Afiliar P. Jurídica	Condiciones Previas:	El usuario escoge la opción de registro de persona jurídica y se presenta el formulario correspondiente.
	Descripción:	El usuario ingresa información para afiliarse. El registro se compone de: razón social, nombre comercial, tipo de identificación, número de identificación, representante legal, actividad, ciudad, dirección, teléfono, correo electrónico, nombre

		del contacto, cargo del contacto, correo electrónico del contacto, usuario y contraseña.
	Condiciones Posteriores:	El afiliado recibe un mensaje de confirmación en su correo electrónico con el usuario y contraseña para que se autentique.
Autenticación	Condiciones Previas:	El afiliado ha consultado la página principal, realizado búsquedas y/o afiliado a nuestra página.
	Descripción:	El afiliado introduce su nombre se usuario y contraseña para autenticarse. Tras enviar la petición de autenticación, el usuario será reconocido por la página.
	Condiciones Posteriores:	El afiliado puede realizar cotizaciones e inscripciones de productos.
Cotización	Condiciones Previas:	El afiliado se ha autenticado.
	Descripción:	El afiliado escoge los productos que desea cotizar de la lista de productos que ha buscado.
	Condiciones Posteriores:	El afiliado tiene la posibilidad de imprimir el documento de cotización.
Cotizar Vehículos	Condiciones Previas:	El afiliado ha escogido la búsqueda de vehículos y se ha autenticado.
	Descripción:	El afiliado selecciona los vehículos que desea cotizar.
	Condiciones Posteriores:	El afiliado tiene la posibilidad de imprimir el documento de cotización de vehículos.
Cotizar Repuestos	Condiciones Previas:	El afiliado ha escogido la búsqueda de repuestos y se ha autenticado.
	Descripción:	El afiliado selecciona los repuestos que desea cotizar.
	Condiciones Posteriores:	El afiliado tiene la posibilidad de imprimir el documento de cotización de repuestos.
Cotizar Paquetes	Condiciones Previas:	El afiliado ha escogido la búsqueda de paquetes y se ha autenticado.
	Descripción:	El afiliado selecciona los paquetes que desea cotizar.
	Condiciones Posteriores:	El afiliado tiene la posibilidad de imprimir el documento de cotización de paquetes y/o generar la orden de pago para su

		adquisición.
Generación de Orden de Pago	Condiciones Previas:	El afiliado ha realizado la cotización de los paquetes que desea adquirir.
	Descripción:	El afiliado ejecuta la realización de la orden de pago. La orden de pago tendrá un número secuencial, datos de la empresa, datos del afiliado, datos de la cotización realizada y valor total.
	Condiciones Posteriores:	El afiliado tiene la posibilidad de imprimir la orden de pago. El afiliado podrá realizar el pago a través de entidades bancarias vinculadas a nuestra empresa.
Inscripción de Producto	Condiciones Previas:	El afiliado se ha autenticado. La activación de los servicios publicitarios se realizará en cuanto la entidad bancaria notifique la cancelación.
	Descripción:	El afiliado escoge si quiere publicar vehículos o repuestos para presentar el registro correspondiente
	Condiciones Posteriores:	El afiliado tiene la posibilidad de ingresar la información requerida.
Inscribir Vehículos	Condiciones Previas:	El afiliado escoge la opción de publicar vehículos
	Descripción:	El afiliado ingresa información de vehículos. El registro se compone de: tipo, modelo, marca, año, descripción, precio. El producto debe asociarse a un paquete activo.
	Condiciones Posteriores:	Los productos son publicados en nuestra página.
Inscribir Repuestos	Condiciones Previas:	El afiliado escoge la opción de publicar repuestos.
	Descripción:	El afiliado ingresa información de repuestos. El registro se compone de: tipo, modelo, marca, descripción, precio. El producto debe asociarse a un paquete activo.
	Condiciones Posteriores:	Los productos son publicados en nuestra página.

Tabla 12
Escenario

3.2 DIAGRAMAS DE SECUENCIA

Con la ayuda de los diagramas de secuencia se realizarán las diferentes transacciones que se van a realizar en el sitio web.

Buscar producto

Los clientes (vendedores o compradores) podrán consultar los diferentes tipos de productos a través de los criterios de búsquedas y como resultado se generará una lista de productos.

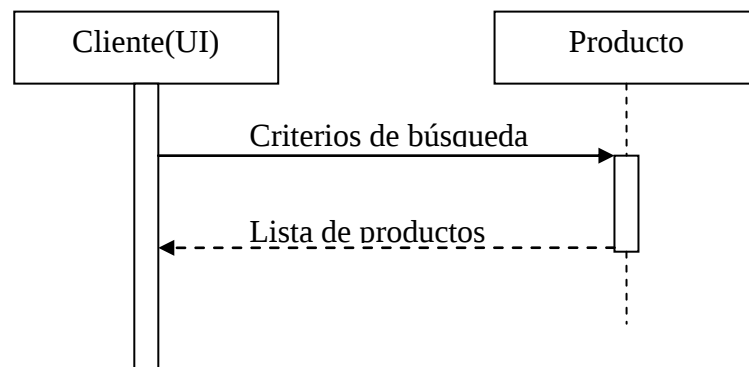


Gráfico 13
Diagrama de Secuencia
Busca Producto

Registrar cliente:

Los Clientes pueden registrarse si así lo desean, estos clientes pueden ser vendedores o compradores según sea el caso.

Registrar cliente comprador

Los compradores se registrarán, ingresando sus datos personales a través de una interfaz, cuyos datos se almacenaran en nuestra base de datos. Si la acción fue satisfactoria se enviará una confirmación.

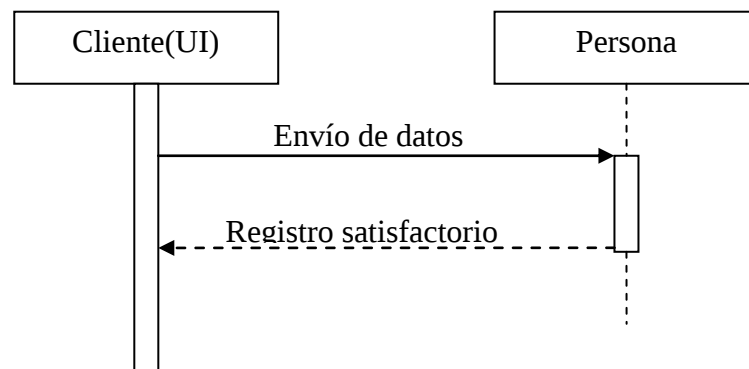


Gráfico 14
Diagrama de Secuencia
Registra Cliente Comprador

Registrar cliente vendedor

Los vendedores se registrarán, ingresando sus datos personales a través de una interfaz, cuyos datos se almacenaran en nuestra base de datos. Si la acción fue satisfactoria se enviará una confirmación.

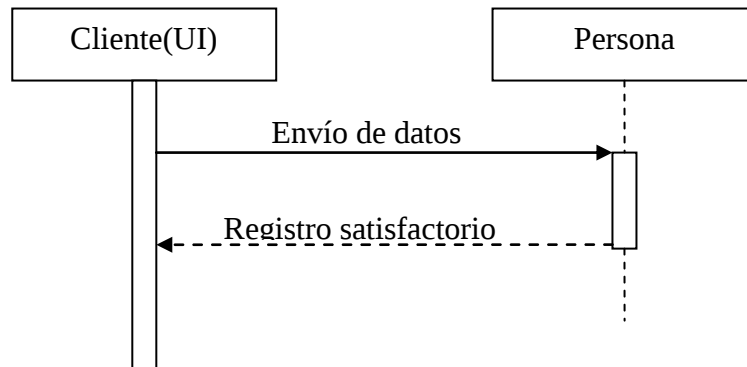


Gráfico 15
Diagrama de Secuencia
Registra Cliente Vendedor

Autenticar

Para que un usuario pueda realizar ciertos procesos, este deberá autenticarse, ingresando el usuario y la contraseña.

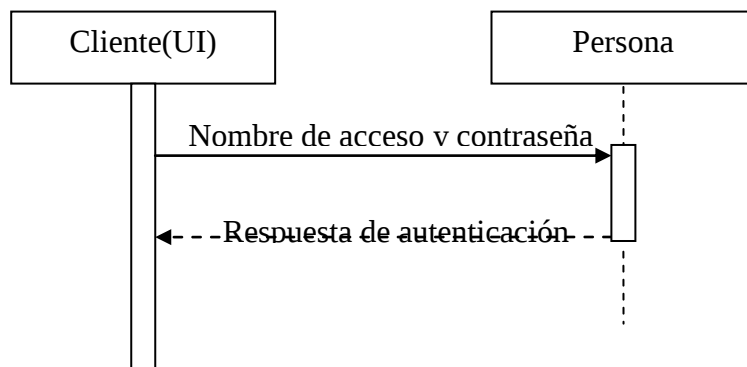


Gráfico 16
Diagrama de Secuencia
Autenticar

Cotizar

Para que un usuario pueda cotizar deberá haberse autenticado, es decir tiene que ser un cliente registrado. Luego de identificar el producto deseado, el usuario tendrá que agregar dicho producto a la proforma, la misma que se creará con su respectivo detalle.

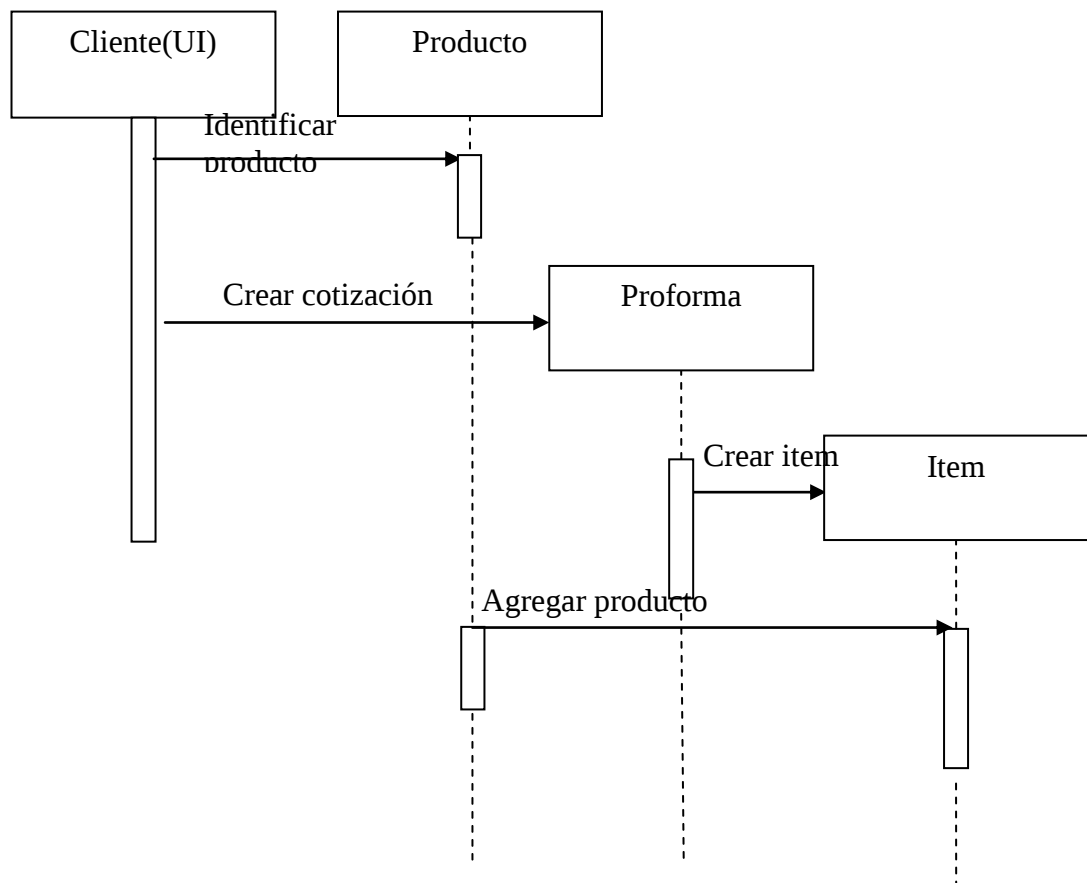


Gráfico 17
Diagrama de Secuencia
Cotizar

Imprimir Proforma

Una vez generada la Proforma, el usuario podrá imprimirla e incluso descargar como un archivo pdf.

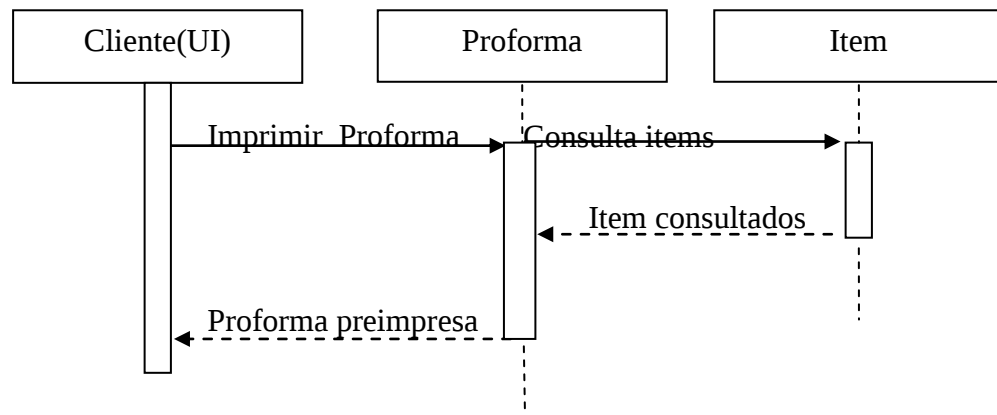


Gráfico 18
Diagrama de Secuencia
Imprimir Proforma

Imprimir Orden de Pago

Luego de haberse creado la proforma, el usuario podrá generar una orden de pago la cual podrá ser impresa si este lo desea.

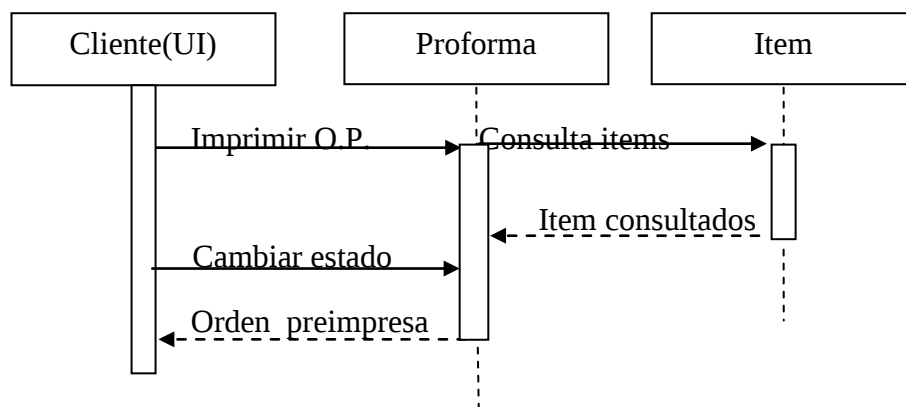


Gráfico 19
Diagrama de Secuencia
Imprimir Orden de Pago

Inscribir producto

El cliente vendedor es el que inscribirá un producto ya sea vehículo o repuesto ingresando la información de dicho producto y este quedará registrado.

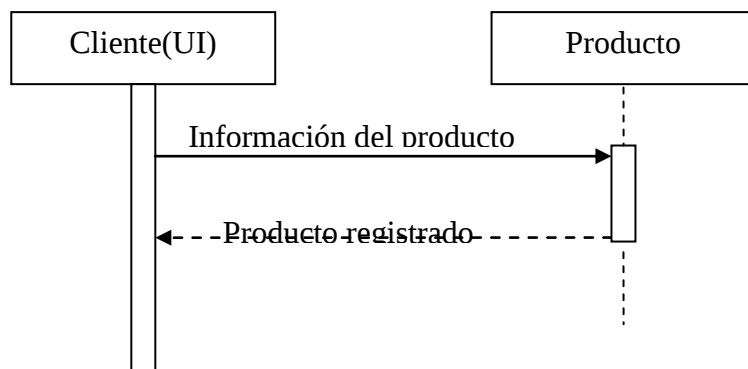


Gráfico 20
Diagrama de Secuencia
Inscribir Producto

3.3 DIAGRAMAS ENTIDAD RELACIÓN

En este diagrama representamos todas las relaciones existentes entre las tablas que hemos utilizados para la base de datos Publicidad. Para esta base de datos hemos creado 17 tablas cada una con sus respectivos campos como se visualiza a continuación:

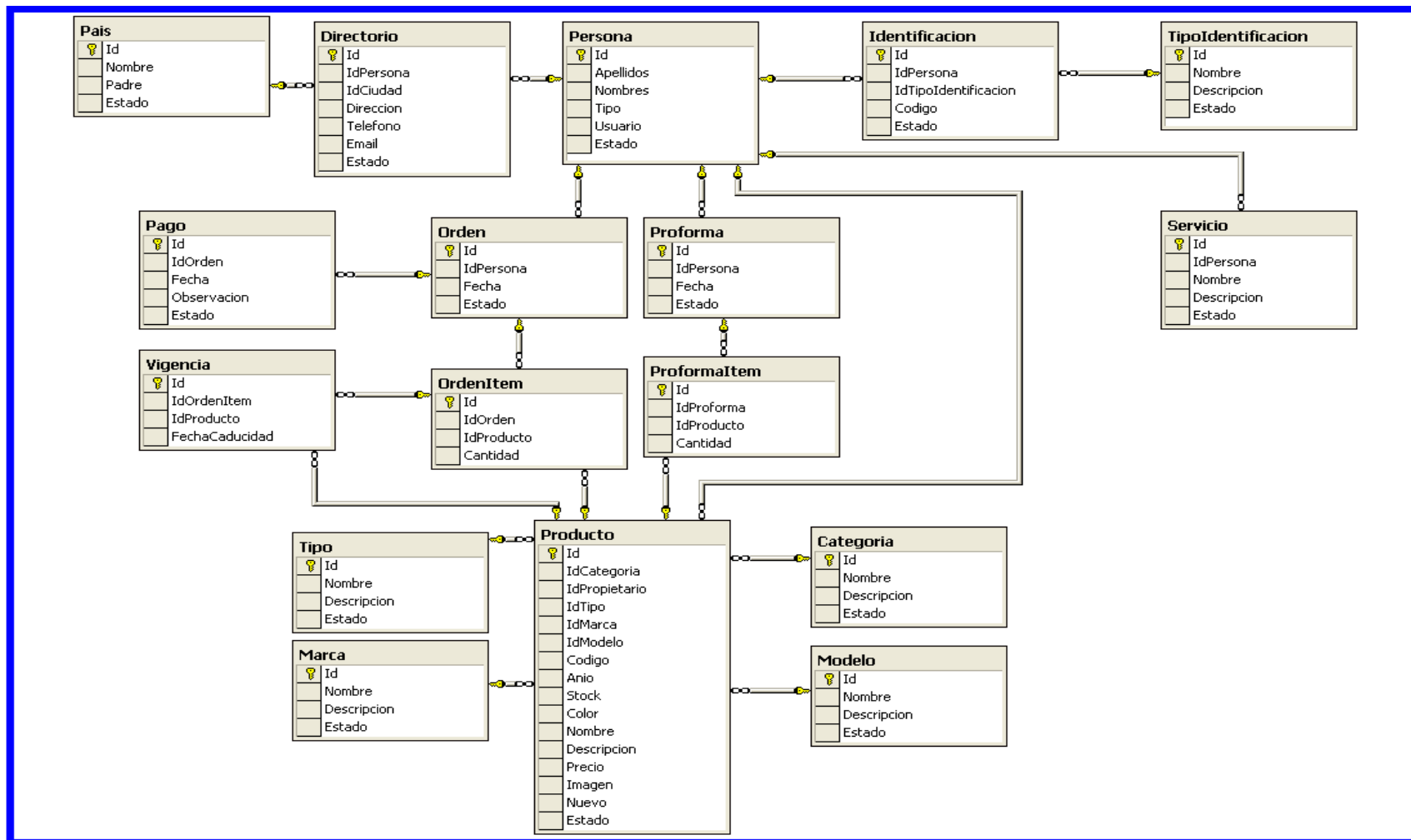


Gráfico 21
Diagrama Entidad Relación

3.4 Diccionario de Datos

	Nombre	Descripción	Observación
1	Categoria	Clases de productos semejantes	
	Id	Identificador único del registro de la tabla	
	Nombre	Nombre	Vehículo, repuesto, paquete, medicina.
	Descripcion	Descripción	
	Estado	1 = activo, 0 = inactivo	Inactivo = anulado
2	Directorio	Datos de ubicación de la persona	
	Id	Identificador único del registro de la tabla	
	IdPersona	Clave foranea de la tabla Persona	
	IdCiudad	Clave foranea de la tabla Pais	
	Direccion	Dirección	
	Telefono	Teléfono	
	Email	Dirección electrónica	
	Estado	1 = activo, 0 = inactivo	Inactivo = anulado
3	Identificacion	Datos de identificación de la persona	
	Id	Identificador único del registro de la tabla	
	IdPersona	Clave foranea de la tabla Persona	
	IdTipoidentificacio	Clave foranea de la tabla Tipoidentificacion	
	Codigo	Número de identificacion	
	Estado	1 = activo, 0 = inactivo	Inactivo = anulado
4	Marca	Distintivo que el fabricante pone a los productos de su industria	
	Id	Identificador único del registro de la tabla	
	Nombre	Nombre	Toyota, Chevrolet, Ford
	Descripcion	Descripción	
	Estado	1 = activo, 0 = inactivo	Inactivo = anulado

5	Modelo	Subdivisiones de un tipo	
	Id	Identificador único del registro de la tabla	
	Nombre	Nombre	Luv, Stoud
	Descripcion	Descripción	
	Estado	1 = activo, 0 = inactivo	Inactivo = anulado
6	Orden	Cabecera de la orden	
	Id	Identificador único del registro de la tabla	
	IdPersona	Clave foranea de la tabla Persona	
	Fecha	Fecha de emisión	
	Estado	1 = activo, 0 = inactivo	Inactivo = anulado
7	OrdenItem	Detalles de la orden	
	Id	Identificador único del registro de la tabla	
	IdOrden	Clave foranea de la tabla Orden	
	IdProducto	Clave foranea de la tabla Producto	
	Cantidad	Cantidad	
8	Pago	Registro de las ordenes canceladas	
	Id	Identificador único del registro de la tabla	
	IdOrden	Clave foranea de la tabla Orden	
	Fecha	Fecha de cancelación de la orden.	Considerada fecha activación de paquetes
	Observacion	Comentario	
	Estado	1 = activo, 0 = inactivo	Inactivo = anulado

9	Pais	Países, Provincias y Ciudades asociadas recursivamente	
	Id	Identificador único del registro de la tabla	
	Nombre	Nombre	
	Padre	A quien pertenece	
	Estado	1 = activo, 0 = inactivo	Inactivo = anulado
10	Persona	Datos de personas naturales o jurídicas	
	Id	Identificador único del registro de la tabla	
	Apellidos	Apellidos	Clase de empresa: S.A., Cia Ltda.
	Nombres	Nombres / Razón Social	
	Tipo	1 = Jurídica, 0 = Natural	
	Sesion	Nombre de usuario	
	Estado	1 = activo, 0 = inactivo	Inactivo = anulado
11	Producto	Datos de productos a publicar	
	Id	Identificador único del registro de la tabla	
	IdCategoria	Clave foranea de la tabla Categoria	
	IdPropietario	Clave foranea de la tabla Persona	
	IdTipo	Clave foranea de la tabla Tipo	
	IdMarca	Clave foranea de la tabla Marca	
	IdModelo	Clave foranea de la tabla Modelo	
	Codigo	Placa asignada a un vehículo nuevo o usado / Serie asignada a un repuesto o paquete	
	Anio	Año de producción	Duración de paquete en días.
	Stock	Cantidad de existencia	Cantidad de productos a publicar según paquete

	Color	Color	
	Nombre	Nombre	
	Descripcion	Descripción o características adicionales del producto	AA, full equipo
	Precio	Precio	
	Imagen	Ruta/directorio/*,jpg de foto del producto	
	Nuevo	1 = nuevo, 0 = usado	
	Estado	1 = activo, 0 = inactivo	Inactivo = anulado
12	Proforma	Cabecera de la proforma	
	Id	Identificador único del registro de la tabla	
	IdPersona	Clave foranea de la tabla Persona	
	Fecha	Fecha de emisión	
	Estado	1 = activo, 0 = inactivo	Inactivo = anulado
13	Proformaltem	Detalles de la proforma	
	Id	Identificador único del registro de la tabla	
	IdProforma	Clave foranea de la tabla Proforma	
	IdProducto	Clave foranea de la tabla Producto	
	Cantidad	Cantidad	
14	Tipo	Divisiones de las categorías	
	Id	Identificador único del registro de la tabla	
	Nombre	Nombre	Camioneta, auto
	Descripcion	Descripción	
	Estado	1 = activo, 0 = inactivo	Inactivo = anulado

15	Tipoidentificacion	Tipo de identificaciones	
	Id	Identificador único del registro de la tabla	
	Nombre	Nombre	Cedula, ruc, pasaporte
	Descripcion	Descripción	
	Estado	1 = activo, 0 = inactivo	Inactivo = anulado
16	Vigencia	Vigencia de la publicidad para cada producto según paquete activado	
	Id	Identificador único del registro de la tabla	
	IdOrdenItem	Clave foranea de la tabla OrdenItem	
	IdProducto	Clave foranea de la tabla Producto	
	FechaCaducidad	Fecha de caducidad de la publicidad	

Tabla 13
Diccionario de Datos

3.5 Diseño del Portal Web

3.5.1 Diseño del Home

Página principal donde se presenta las dos formas que un cliente puede hacer uso del portal web, ya sea como cliente comprador o cliente vendedor.



Gráfico 22
Diseño del Home

3.5.2 Diseño de Interfaces de Compradores.

Luego que seleccionamos la opción compradores, se mostrara los diferentes tipos de vehículos y repuesto que se promocionan en este website, otra opción que tendrá la pagina será de que el cliente desee registrarse, ya que para poder cotizar y realizar proforma tiene que estar registrado.

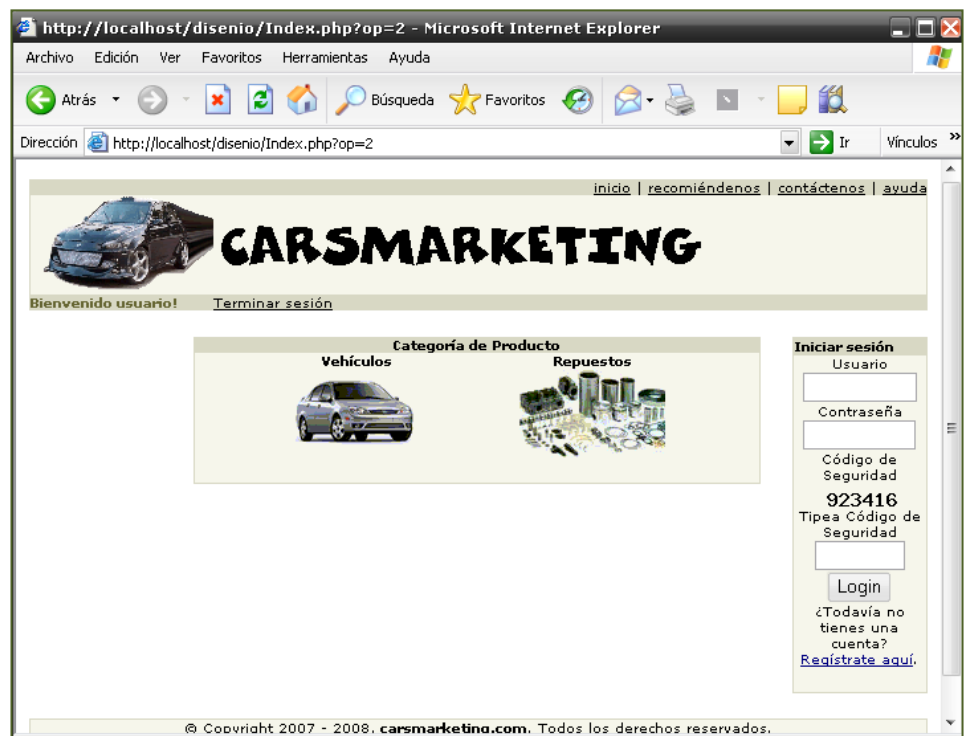


Gráfico 23
Interfaz Compradores - Categoría Productos

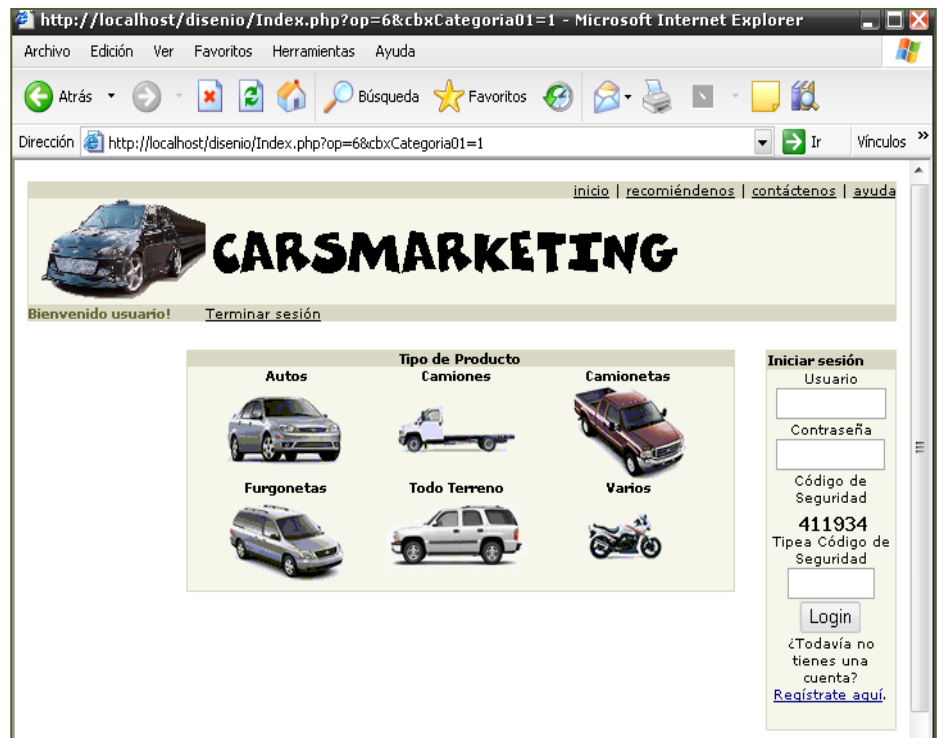


Gráfico 24
Interfaz Compradores – Tipo Producto

Después de haber seleccionado cualquiera de la categoría de vehículos que se mostro en la página anterior, se desplegara esta página con información de todos los vehículos del tipo seleccionado y del propietario de los productos.

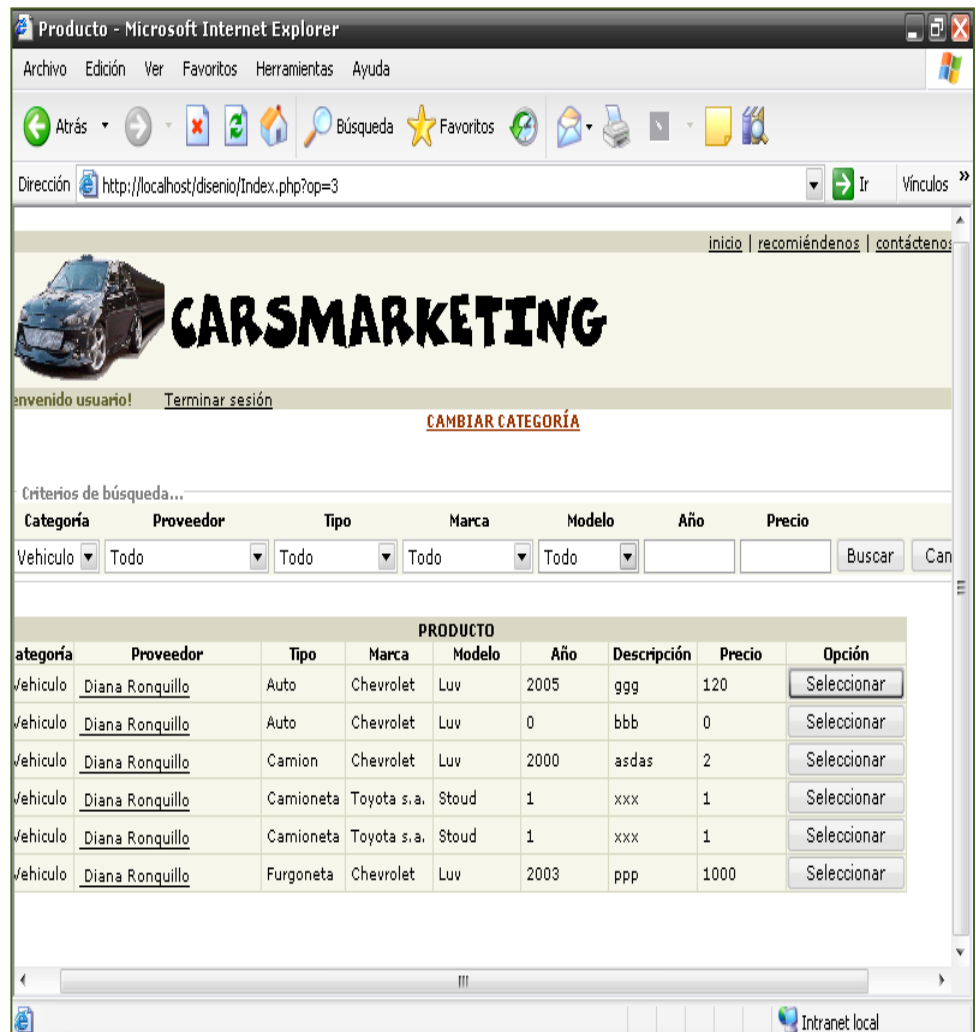


Gráfico 25
Interfaz Compradores – Productos

Habrà otra opción de que el comprador pueda profundizar en su búsqueda ya sea por categoría, concesionaria, tipo, marca, precio o año, habrá una caja de texto donde se pondrá especificar lo que desea consultar, también se

contara con un checkbox que tiene como opción profundizar la búsqueda y listo le saldrá solo la información que ha elegido.

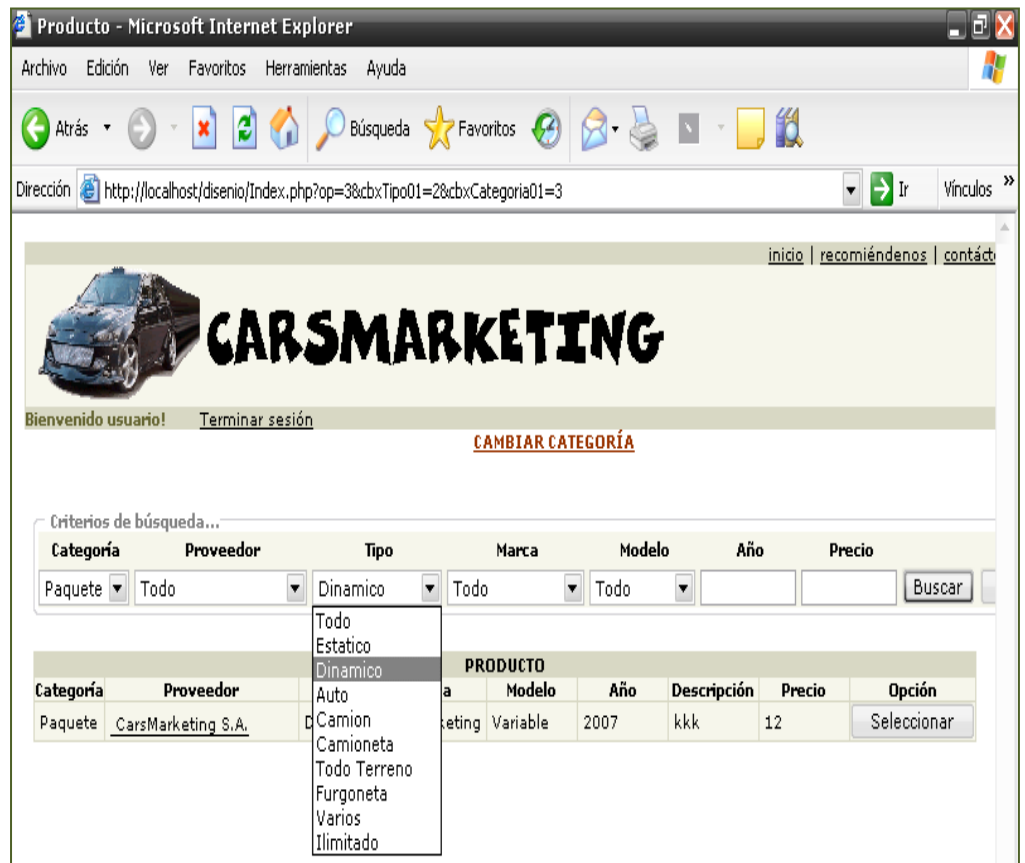


Gráfico 26
Interfaz Compradores – Búsqueda Productos

Ejemplo: si el comprador elije la categoría marca y escribe en la caja de texto el nombre de la marca con la opción profundizar, se desplegará las diferentes marcas del tipo de vehículo que eligió.

Esta página también tendrá la opción de registrarse Si el cliente no se ha registrado anteriormente no podrá realizar la opción de cotizar, a continuación se despliega otra página con el formulario que tendrá que llenar con sus datos personales:

← Atrás → Búsqueda Favoritos Ir Vínculos

dirección <http://localhost/disenio/Index.php?op=4>

[inicio](#) | [recomiéndenos](#) | [contáctenos](#) | [ayuda](#)

 **CARSMARKETING**

Bienvenido usuario! [Terminar sesión](#)

Persona

Id:

Nombres:

Apellidos:

Persona Jurídica: ☒

Usuario:

Contraseña:

Confirmar contraseña:

Estado: ☒

Identificación

Tipo de Identificación:

Código:

Estado: ☒

Directorio

País:

Provincia:

Ciudad:

Dirección:

Teléfono:

Email:

Estado: ☒

Gráfico 27
Interfaz Compradores – Registrar Cliente

diciembre 06, 2007

inicio | servicios | contáctenos | recomien

CARS MARKETING

Bienvenido Pamela Pin! Terminar sesión

Inicio
La empresa
Quiénes somos?
Misión
Visión
Objetivos Generales

Opciones

Datos Personales

Identificaciones

Direcciones

Inf. Financiamiento

Para Vehículos Nuevos y Usados

click here

PROFORMA

Fecha: 2007-12-06
Cliente: Pamela Pin CI/RUC: 0910590934

Cantidad	Propietario	Producto	Precio	Total	Opciones
	Automotores CorpGuesa S.A.	Auto Chevrolet Aveo, Año 2005	3500	3500	
	Autolasas S.A.	Auto Chevrolet Optra, Año 2007	6500	6500	
			10000		

Seguir Buscando

Imprimir Proforma

Ver Orden

La información de la proforma sólo puede ser descargada una vez al pulsar botón "Imprimir proforma".
Si desea realizar modificaciones en la proforma, antes de descargarla, pulse el botón "Seguir Buscando"

Cotiza tu
vehículo
nuevo o usado

COMONUEVOS
Autolasas - Chevrolet

Gráfico 28
Interfaz Compradores – Cotización

3.5.3 Diseño de Interfaces de Vendedores

Una vez que se ha seleccionado en el home la opción de VENDEDORES, se desplegará opciones de los servicios que se ofrecen a los vendedores como son: ingreso de uno o más de dos productos, para los que deseen la segunda opción se le ofrecerá tres tipos de paquetes.

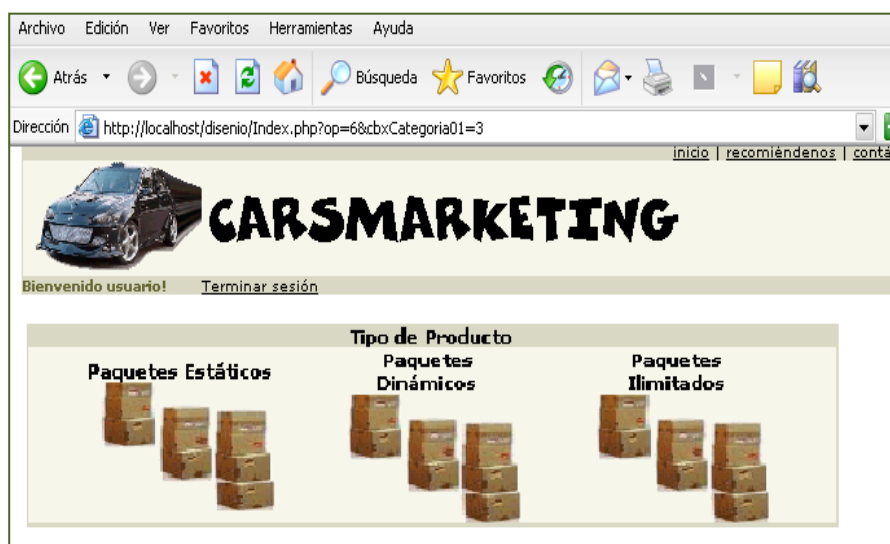


Gráfico 29
Interfaz Vendedores – Tipo Producto



Gráfico 30
Interfaz Vendedores – Consulta de Paquetes

Si el vendedor desea cotizar de igual manera tendrá primero que registrarse.



The screenshot shows a web browser window with the address bar displaying `http://localhost/disenio/Index.php?op=4`. The page header includes navigation links: [inicio](#), [recomiéndenos](#), [contáctenos](#), and [ayuda](#). Below the header is a banner featuring a car image and the text "CARSMARKETING". A message "Bienvenido usuario!" is followed by a [Terminar sesión](#) link.

The registration form is divided into three sections:

- Persona**: Contains fields for "Id:" (value 0), "Nombres:", "Apellidos:", "Persona Jurídica:" (checked), "Usuario:", "Contraseña:", "Confirmar contraseña:", and "Estado:" (checked).
- Identificación**: Contains a dropdown for "Tipo de Identificación:" (value C.I.), a "Código:" field, and an "Estado:" (checked) checkbox.
- Directorio**: Contains dropdowns for "País:" (value Ecuador), "Provincia:" (value Azuay), and "Ciudad:" (value Cuenca), followed by fields for "Dirección:", "Teléfono:", "Email:", and an "Estado:" (checked) checkbox.

At the bottom of the form are two buttons: "Guardar" and "Cancelar".

Gráfico 31
Interfaz Vendedores – Registrar Cliente

[illegible]

Gráfico 32
Interfaz Vendedores – Cotización

Si el vendedor escogió la opción de Realizar Orden se generara una orden de pago con su respectivo número y datos de la concesionaria, donde la tendrá que imprimir para posteriormente acercarse a cancelar el pedido.

diciembre 06, 2007

[Inicio](#) | [servicios](#) | [contáctenos](#) | [recom](#)


CARSMARKETING

Bienvenido Pamela Pin!
 [Terminar sesión](#)

[Inicio](#)
[La empresa](#)
[Quiénes somos?](#)
[Misión](#)
[Visión](#)
[Objetivos Generales](#)

[Opciones](#)
[Datos Personales](#)
[Identificaciones](#)
[Direcciones](#)
[Producto](#)
[Vigencia de Paquetes](#)
[Servicios](#)
[Reporte: Preferencia de Clientes](#)
[Reporte: Consulta de Clientes](#)

Para Vehículos Nuevos y Usados

click here

ORDEN

Fecha: 2007-12-06
 Cliente:

CI/RUC:

Cantidad	Propietario	Producto	Precio	Total	Opciones
1	CarsMarketing S.A.	Dinamico CarsMarketing Variable, vigencia en ds.: 30	3	3	 
			3		

Seguir Buscando

Imprimir Orden

La información de la orden sólo puede ser descargada una vez al pulsar botón "Imprimir orden".

Si desea realizar modificaciones en la orden, antes de descargarla, pulse el botón "Seguir Buscando"

Para cancelar la orden acérquese, con el documento impreso, a cualquier agencia de Banco del Pacífico o a nuestra oficina.

De tener ud. una cuenta en el Banco del Pacífico puede utilizar los servicios de: débitos automáticos, audiomático e intermático.

Si tiene alguna pregunta por favor escribanos a info@carsmarketing.com.

Con
 CarsMarketing



Gráfico 33
Interfaz Vendedores – Orden de Pago

Si el usuario desea imprimir la proforma o la orden se les presentara el en pdf, para que así puedan guardar el archivo de las cotizaciones en su maquina e imprimirlo cuando desee.

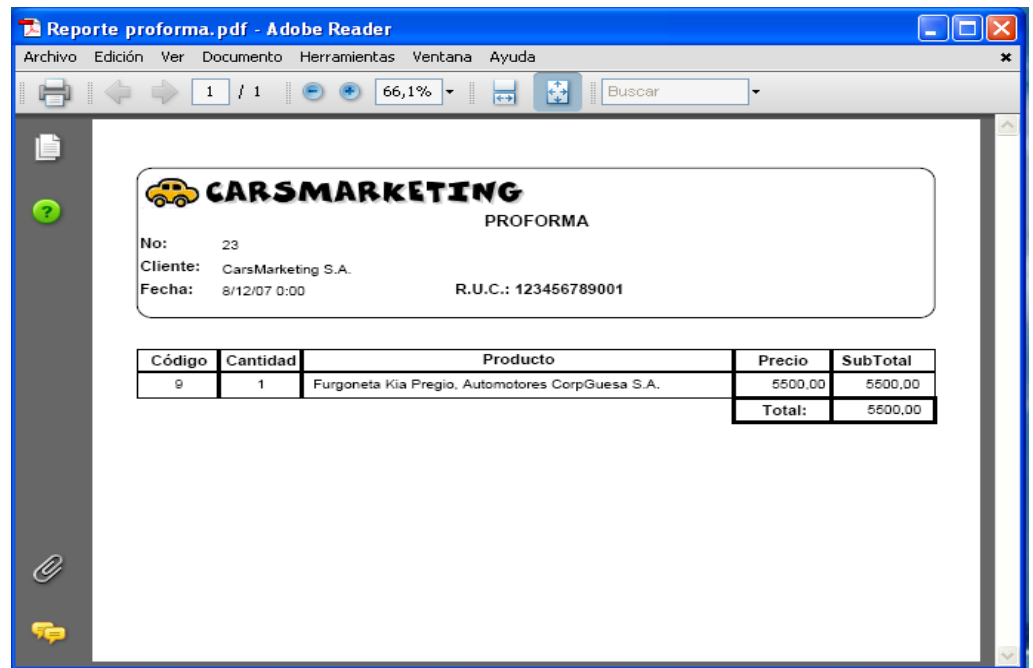


Gráfico 34
Interfaz Compradores – Imprimir Proforma

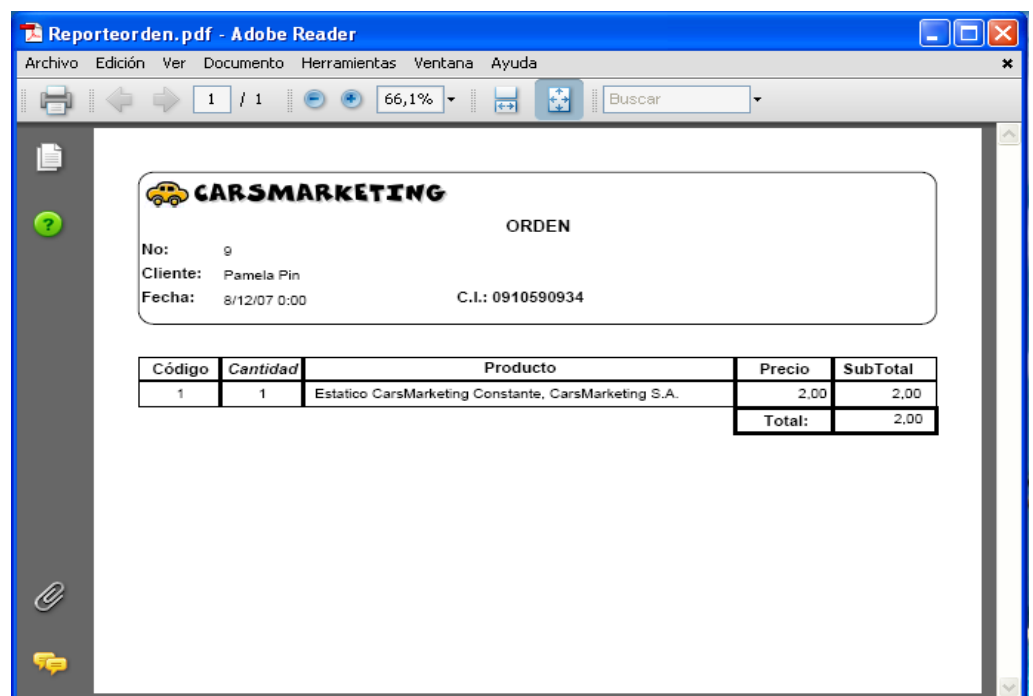


Gráfico 35
Interfaz Vendedores – Imprimir Orden de Pago

CAPITULO 4

IMPLEMENTACIÓN Y

DESARROLLO DEL SISTEMA

En este capítulo hablaremos todo concerniente al desarrollo del portal, explicaremos ciertos procesos como funciones, clases, procedimientos u otros, relevantes para la creación de nuestro sitio web. Para la codificación del proyecto se ha utilizado php y java como lenguaje de programación.

4.1 Base de Datos

Para el almacenamiento de la información, hemos creado la base de datos "Publicidad", la misma que contiene las diferentes tablas y sus relaciones, de acuerdo al DER (ver Gráfico # 21).

```
CREATE DATABASE "Publicidad"  
WITH OWNER = postgres  
ENCODING = 'UNICODE'  
TABLESPACE = pg_default;
```

Para esta base de datos (Publicidad) hemos creado de 16 tablas, cada una con sus respectivos campos, los mismos con su respectivo tipo datos:

<u>CATEGORIA</u>
"Id" int8 NOT NULL , "Nombre" varchar (50) NOT NULL , "Descripcion" varchar (100) NOT NULL , "Estado" bit(1) NOT NULL

Tabla 14
Tabla Categoria

<u>TIPO</u>
"Id" int8 NOT NULL , "Nombre" varchar (50) NOT NULL , "Descripcion" varchar (100) NOT NULL , "Estado" bit NOT NULL

Tabla 15
Tabla Tipo

<u>MARCA</u>
"Id" int8 NOT NULL , "Nombre" varchar (50)NOT NULL , "Descripcion" varchar (100)NOT NULL , "Estado" bit NOT NULL

Tabla 16
Tabla Marca

<u>MODELO</u>
"Id" int8 NOT NULL , "Nombre" varchar (50) NOT NULL , "Descripcion" varchar (100) NOT NULL , "Estado" bit NOT NULL

Tabla 17
Tabla Modelo

<u>TIPO IDENTIFICACION</u>
"Id" int8 NOT NULL , "Nombre" varchar (50) NOT NULL , "Descripcion" varchar (100) NOT NULL , "Estado" bit NOT NULL

Tabla 18
Tabla Tipo Identificación

<u>IDENTIFICACION</u>
"Id" int8 NOT NULL , "Nombre" varchar (50) NOT NULL , "Descripcion" varchar (100) NOT NULL , "Estado" bit NOT NULL

Tabla 19
Tabla Identificación

<u>PAIS</u>
"Id" int8 NOT NULL , "Nombre" varchar (50) NOT NULL , "Padre" int8 NOT NULL , "Estado" bit NOT NULL

Tabla 20
Tabla Pais

<u>DIRECTORIO</u>
"Id" int8 NOT NULL , "IdPersona" int8 NOT NULL , "IdCiudad" int8 NOT NULL , "Direccion" varchar (100) NOT NULL , "Telefono" varchar (9) NOT NULL , "Email" varchar (100) NOT NULL , "Estado" bit(1) NOT NULL

Tabla 21
Tabla Directorio

<u>PERSONA</u>
"Id" int8 NOT NULL , "Apellidos" varchar (50) NOT NULL , "Nombres" varchar (50) NOT NULL , "Tipo" bit NOT NULL , "Sesion" varchar (20) NOT NULL , "Estado" bit NOT NULL

Tabla 22
Tabla Persona

<u>PRODUCTOS</u>
"Id" int8 NOT NULL, "IdCategoria" int8 NOT NULL, "IdPropietario" int8 NOT NULL, "IdTipo" int8 NOT NULL, "IdMarca" int8 NOT NULL, "IdModelo" int8 NOT NULL, "Codigo" varchar(10) NOT NULL, "Anio" int8 NOT NULL, "Stock" int8 NOT NULL, "Color" varchar(50) NOT NULL, "Nombre" varchar(100) NOT NULL, "Descripcion" varchar(100) NOT NULL, "Precio" numeric(10,2) NOT NULL, "Imagen" varchar(100) NOT NULL, "Nuevo" bit(1) NOT NULL, "Estado" bit(1) NOT NULL

Tabla 23
Tabla Productos

<u>PROFORMA</u>
"Id" int8 NOT NULL , "IdPersona" int8 NOT NULL , "Fecha" date NOT NULL , "Estado" bit NOT NULL

Tabla 24
Tabla Proforma

<u>PROFORMA ITEM</u>
"Id" int8 NOT NULL , "IdProforma" int8 NOT NULL , "IdProducto" int8 NOT NULL , "Cantidad" int8 NOT NULL

Tabla 25
Tabla Proforma Item

<u>ORDEN</u>
"Id" int8 NOT NULL , "IdPersona" int8 NOT NULL , "Fecha" date NOT NULL , "Estado" bit NOT NULL

Tabla 26
Tabla Orden

<u>ORDEN ITEM</u>
"Id" int8 NOT NULL , "IdOrden" int8 NOT NULL , "IdProducto" int8 NOT NULL , "Cantidad" int8 NOT NULL

Tabla 27
Tabla Orden Item

<u>PAGO</u>
"Id" int8 NOT NULL, "IdOrden" int8 NOT NULL, "Fecha" date NOT NULL, "Observacion" varchar(100) NOT NULL, "Estado" bit(1) NOT NULL,

Tabla 28
Tabla Pago

<u>VIGENCIA</u>
"Id" int8 NOT NULL, "IdOrdenItem" int8 NOT NULL, "IdProducto" int8 NOT NULL, "FechaCaducidad" date NOT NULL,

Tabla 29
Tabla Vigencia

A nivel de base de datos, se crearon funciones tales como Seleccionar, Buscar, Insertar, Eliminar y Actualizar para cada una de las tablas que anteriormente mencionamos. A continuación se ilustra la estructura de una función como ejemplo:

```
CREATE OR REPLACE FUNCTION "ProductosBuscar"("varchar", "varchar", "varchar",
"varchar", "varchar", "varchar", "varchar", "varchar", "varchar", "varchar",
"varchar", "varchar", "varchar", "varchar")
RETURNS SETOF "Productos" AS
$BODY$
SELECT
    "Id",
    "IdCategoria",
    "IdPropietario",
    "IdTipo",
    "IdMarca",
    "IdModelo",
    "Codigo",
    "Anio",
    "Stock",
    "Color",
    "Nombre",
    "Descripcion",
    "Precio",
    "Imagen",
    "Nuevo",
    "Estado"
FROM
    "Productos"
WHERE
    "Id" LIKE $1      AND "IdCategoria" LIKE $2
    AND "IdPropietario" LIKE $3  AND "IdTipo" LIKE $4
    AND "IdMarca" LIKE $5  AND "IdModelo" LIKE $6
    AND "Codigo" LIKE $7  AND "Anio" LIKE $8
    AND "Stock" LIKE $9  AND "Color" LIKE $10
    AND UPPER("Nombre") LIKE '%' || UPPER($11) || '%'
    AND UPPER("Descripcion") LIKE '%' || UPPER($12) || '%'
    AND "Precio" <= CAST($13 AS NUMERIC(10,2)) --Hasta Precio Maximo
    AND UPPER("Imagen") LIKE '%' || UPPER($14) || '%'
    AND CAST("Nuevo" AS int8) LIKE $15
    AND CAST("Estado" AS int8) LIKE $16
ORDER BY "Id";
$BODY$
LANGUAGE 'sql' VOLATILE;
ALTER FUNCTION "ProductosBuscar"("varchar", "varchar", "varchar", "varchar", "varchar",
"varchar", "varchar", "varchar", "varchar", "varchar", "varchar", "varchar", "varchar",
"varchar", "varchar") OWNER TO grupo8;
```

Tabla 30
Función Producto Buscar

4.2 Clases y Métodos

Para mantener el enfoque orientado a objetos, se desarrollaron una colección de clases, para poder manipular la información contenida en cada una de las tablas de la base de datos, es decir que a aquí está centrada la lógica de negocio. Por cada tabla se crearon las clases: Be, Bo y Dao.

La **Clase Be** es aquella clase que está constituida por métodos que tienen la función de ingresar y obtener (Set / Get) datos de los objetos. En esta clase declaramos el tipo de dato de cada uno de los objetos.

Tabla 31
Método Be

```
private int Id;
private int IdCategoria;
private int IdProveedor;
private int IdTipo;
.
.
public Producto () {
    this.Id = 0;
    this.IdCategoria = 0;
    this.IdProveedor = 0;
    .
    .
}
public Producto (int id, int idCategoria ..... ) {
    this.Id = id;
    this.IdCategoria = idCategoria;
    this.IdProveedor = idProveedor;
    .
    .
}
public int getId() {
    return this.Id;
}
public void setId(int value) {
    this.Id = value;
}
```

La **Clase Bo** contiene todo aquello que deseamos realizar con los datos almacenados en la base de datos, es decir métodos tales como: guardar (insertar / actualizar), seleccionar, buscar y eliminar.

Tabla 32
Método Bo

```
import java.sql.Connection;
import java.sql.SQLException;
import java.util.ArrayList;

public class ProductoBo {
    public int guardar(Producto oProducto) throws SQLException, Exception {
        if(oProducto.getId() == 0) {
            return insertar(oProducto);
        } else {
            return actualizar(oProducto);
        }
    }

    private int insertar(Producto oProducto) throws SQLException, Exception {
        Connection con = null;
        int n = 0;
        try {
            con = Conexion.getConnection();
            ProductoDao oProductoDao = new ProductoDao(con);
            n = oProductoDao.insertar(oProducto);
            con.commit();
        } catch (Exception e) {
            if (con != null) {
                con.rollback();
                throw new Exception(e.getMessage());
            }
        } finally {
            if (con != null) {
                con.close();
            }
        }
        return n;
    }

    public Producto seleccionar(int id) throws SQLException, Exception {
        Producto oProducto = new Producto();
        Connection con = null;
        try {
            con = Conexion.getConnection();
            ProductoDao oProductoDao = new ProductoDao(con);
            oProducto = oProductoDao.seleccionar(id);
        } catch (Exception e) {
            throw new Exception(e.getMessage());
        } finally {
            if (con != null) {
                con.close();
            }
        }
        return oProducto;
    }
}
```

La **Clase Dao** es aquella que realiza la conexión a la base de datos, esta clase está directamente relacionada con las otras dos clases anteriormente mencionadas.

Tabla 33
Método Dao

```
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;

public class ProductoDao {
    private Connection con;

    public ProductoDao (Connection con) {
        this.con = con;
    }

    public int insertar(Producto oProducto) throws SQLException, Exception {
        ResultSet rs;
        int n = 0;
        String oSQL = "SELECT \"ProductoInsertar\"(\"" +
            oProducto.getIdCategoria() + "\", \"" + oProducto.getIdProveedor() +
            "\", \"" + oProducto.getIdTipo() + "\", \"" + oProducto.getIdMarca() + "\",
            \"" + oProducto.getIdModelo() + "\", \"" + oProducto.getAnio() + "\", \"" +
            oProducto.getDescripcion() + "\", \"" + oProducto.getPrecio() + "\"),";
        Statement stmt = con.createStatement();
        rs = stmt.executeQuery(oSQL);
        while (rs.next()) {
            n = rs.getInt("ProductoInsertar");
        }
        rs.close();
        return n;
    }
}
```

Antes de crear las respectivas clases Be, Bo, Dao, construimos la clase denominada **“Conexión”**, aquella que nos permite conectarnos con la base de datos **“Publicidad”**, para así poder obtener los datos que solicitamos a través de los métodos.

```

public class Conexion {
    private static String Provider;
    private static String DataSource;
    private static String InitialCatalog;
    private static String User;
    private static String Password;
    private static Connection con;

    public Conexion(){
        super();
    }
    public static Connection getConnection() throws ClassNotFoundException,
SQLException{
        Provider = "postgresql";
        DataSource = "SERVIDOR";
        InitialCatalog = "Publicidad";
        User = "postgres";
        Password = "rroca";

        try {
            if ((con == null)||con.isClosed()) {
                Class.forName("org.postgresql.Driver");
                con = DriverManager.getConnection("jdbc:" + Provider + "://" +
DataSource + "/" + InitialCatalog, User, Password);
                con.setAutoCommit(true);
            }
        } catch (ClassNotFoundException e) {
            throw new ClassNotFoundException("ClassNotFoundException: Localición
fallida del driver de conexión.");
        } catch (SQLException e) {
            throw new SQLException(e.toString());
        }
        return con;
    }
    public static Connection getConnection(String user, String password) throws
ClassNotFoundException, SQLException {
        Provider = "postgresql";
        DataSource = "SERVIDOR";
        InitialCatalog = "Publicidad";
        User = user;
        Password = password;
        try{
            if((con == null)||con.isClosed()){
                Class.forName("org.postgresql.Driver");
                con = DriverManager.getConnection("jdbc:" + Provider +
"//" + DataSource + "/" + InitialCatalog, User, Password);
                con.setAutoCommit(true);
            }
        } catch (ClassNotFoundException e) {
            throw new ClassNotFoundException("ClassNotFoundException:
Localición fallida del driver de conexión.");
        } catch (SQLException e) {
            throw new SQLException(e.toString());
        }
        return con;
    }
}

```

Tabla 34
Conexión

CAPITULO 5

PRUEBAS

Las pruebas que se realizan en un ambiente web son mecanismos que sirven para verificar la correcta funcionalidad, tiempo de respuesta y controlar cualquier tipo de error que se presente durante la navegación. También aseguran que se cumplan con los requerimientos y es el responsable de evaluar el diseño. Estas pruebas deberán realizarse periódicamente al término de cada módulo.

5.1 Tipo de Pruebas

Pruebas en la Fase de Desarrollo

- Pruebas funcional
- Identificación de resultados

Pruebas Rigurosas

- Prueba de componentes
- Prueba de base de datos
- Prueba de Infraestructura
- Prueba de Seguridad
- Prueba de Integración
- Prueba de aceptación y usabilidad del usuario
- Prueba de estrés, capacidad y rendimiento

5.2 Desarrollo de las Pruebas

Pruebas en la Fase de Desarrollo

Realizamos una clase denominada Control de Calidad, en donde logramos comprobar si cada uno de los procedimientos obtiene el resultado esperado, con diferentes parámetros.

```

public class ControlCalidad {
    private Orden oOrden;
    private OrdenBo oOrdenBo;

    public ControlCalidad() {
        oOrden = new Orden();
        oOrdenBo = new OrdenBo();
    }

    public void insertar() throws SQLException, Exception {
        oOrden.setIdPersona(7);

        OrdenItem oOrdenItem = new OrdenItem();
        oOrdenItem.setIdProducto(1);
        oOrdenItem.setCantidad(5);

        OrdenItem oOrdenItem2 = new OrdenItem();
        oOrdenItem2.setIdProducto(2);
        oOrdenItem2.setCantidad(5);

        oOrden.agregarItem(oOrdenItem);
        oOrden.agregarItem(oOrdenItem2);
        System.out.println(oOrdenBo.guardar(oOrden));
    }

    public void seleccionar() throws SQLException, Exception {
        oOrden = oOrdenBo.seleccionar(5);
        System.out.println(oOrden.getId());
        System.out.println(oOrden.getIdPersona());
        System.out.println(oOrden.getFecha());
        System.out.println(oOrden.getEstado());
        ArrayList items = oOrden.getOrdenItems();

        for (int i = 0; i < items.size(); i++) {
            OrdenItem oOrdenItem = (OrdenItem)items.get(i);
            System.out.println(oOrdenItem.getId() + " " +
oOrdenItem.getIdOrden() + " " + oOrdenItem.getIdProducto() + " " + oOrdenItem.getCantidad());
        }
    }

    public void actualizar() throws SQLException, Exception {
        int id;
        Date oDate = new Date("2007/10/21");

        oOrden = oOrdenBo.seleccionar(5);
        oOrden.setIdPersona(8);
        oOrden.setFecha(oDate);
        oOrden.setEstado((byte)0);
        ArrayList items = oOrden.getOrdenItems();

        for (int i = 0; i < items.size(); i++) {
            OrdenItem oOrdenItem = (OrdenItem)items.get(i);
            oOrdenItem.setIdProducto(3);
            oOrdenItem.setCantidad(10);
        }
        id = oOrdenBo.guardar(oOrden);
        System.out.print(id);
    }

    public void eliminar() throws SQLException, Exception {
        System.out.print(oOrdenBo.eliminar(5));
    }

    public void buscar() throws SQLException, Exception {
        ArrayList Ordens;

        Ordens = oOrdenBo.buscar("%", "%", "2007-11-03", "2007-11-03", "%");
    }
}

```

```

        for (int i = 0; i < Ordens.size(); i++) {
            oOrden = (Orden)Ordens.get(i);

            System.out.println(oOrden.getId());
            System.out.println(oOrden.getIdPersona());
            System.out.println(oOrden.getFecha());
            System.out.println(oOrden.getEstado());
            ArrayList items = oOrden.getOrdenItems();

            for (int j = 0; j < items.size(); j++) {
                OrdenItem oOrdenItem = (OrdenItem)items.get(j);
                System.out.println(oOrdenItem.getId() + " " +
oOrdenItem.getIdOrden() + " " + oOrdenItem.getIdProducto() + " " + oOrdenItem.getCantidad());
            }
        }

    public static void main(String[] args) throws SQLException, Exception {
        ControlCalidad oControlCalidad;

        oControlCalidad = new ControlCalidad();
        /*oControlCalidad.insertar();*/
        /*oControlCalidad.seleccionar();*/
        /*oControlCalidad.actualizar();*/
        oControlCalidad.eliminar();
        /*oControlCalidad.buscar();*/
    }
}

```

Tabla 35
Control de Calidad

Para la identificación de los resultados, hemos creados los métodos de Insertar, Seleccionar, Actualizar, Eliminar y Buscar, los cuales nos generan los valores esperados.

Pruebas Rigurosas

En esta fase hemos realizado algunas de las siguientes pruebas que detallaremos a continuación:

A nivel de **Base de Datos**, ejecutamos sentencias SQL en el administrador de Postgre, de las diferentes funciones para cada una de las tablas, por ejemplo mencionaremos una de ellas:

Función Seleccionar

```
SELECT "PagoSeleccionar"(1);
```

Función Insertar

```
SELECT "PersonalInsertar"('Armijos', 'Queenie', CAST(0 AS BIT),  
'qarmijos', 'qarmijos');
```

Función Buscar

```
SELECT "TipoidentificacionBuscar"('%', '%', '%', '1');
```

Función Actualizar

```
SELECT "ProductosActualizar"(1, 1, 1, 1, 1, 1, '000000', '2007',  
12, 'estandar', 'S/N', 'prueba', '25.36', 'C:/images/img.jpg', CAST(0  
AS BIT), CAST(1 AS BIT));
```

Función Eliminar

```
SELECT "CategoriaEliminar"(5);
```

Ejecutando cada una de estas funciones verificamos que los valores retornados son correctos.

En cuanto a la **Seguridad** del Portal, hemos realizado pruebas, ya que en cualquier sistemas es un tema importante, aun mas si se trata de una aplicación Web, Como por ejemplo se pueden insertar sentencias SQL para poder iniciar sesión dentro de la aplicación con el primer usuario que encuentre en nuestra base de datos. Pero como esta aplicación trabaja con procedimientos almacenados, en la que el Login y las Claves son manejadas por parámetros y los mismos son afectadas por una sentencia.

Otra prueba a nivel de seguridad del portal que se ha realizado, fue ingresar alguna dirección URL donde se realizan procesos restringidos (dependiendo del usuario), pero en esta parte del sistema se verifica si se ha iniciado sesión, como esta sesión es nula se re direcciona a una página indicando que no se ha ingresado el nombre del usuario y password.

En cuanto a la **Integridad** del proyecto se ha realizado pruebas parciales al termino de cada modulo, como por ejemplo a nivel de la creación de la base de dato, construcción de clases y diseño de la GUI (interfaz).

A continuación mostraremos una pequeña tablas, en la que se resume aquellas pruebas mencionadas anteriormente.

No.	DESCRIP.	BASE DE DATOS							CLASES				GUI		
		CT	I	S	U	D	F	P	BE	DAO	BO	P	B/N	FINAL	PRUEB
1	Categoria	X	X	X	X	X	X	X	X	X	X	X	X	X	X
2	Directorio	X	X	X	X	X	X	X	X	X	X	X	X	X	X
3	Identificacion	X	X	X	X	X	X	X	X	X	X	X	X	X	X
4	Marca	X	X	X	X	X	X	X	X	X	X	X	X	X	X
5	Modelo	X	X	X	X	X	X	X	X	X	X	X	X	X	X
6	Orden	X	X	X	X	X	X	X	X	X	X	X	X	X	X
7	OrdenItem	X	X	X	X	X	X	X	X	X	X	X	X	X	X
8	Pago	X	X	X	X	X	X	X	X	X	X	X	X	X	X
9	Pais	X	X	X	X	X	X	X	X	X	X	X	X	X	X
10	Persona	X	X	X	X	X	X	X	X	X	X	X	X	X	X
11	Productos	X	X	X	X	X	X	X	X	X	X	X	X	X	X
12	Proforma	X	X	X	X	X	X	X	X	X	X	X	X	X	X
13	Proformaltem	X	X	X	X	X	X	X	X	X	X	X	X	X	X
14	Servicio	X	X	X	X	X	X	X	X	X	X	X	X	X	X
15	Tipo	X	X	X	X	X	X	X	X	X	X	X	X	X	X
16	Tipoidentific	X	X	X	X	X	X	X	X	X	X	X	X	X	X
17	Vigencia	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Tabla 36
Pruebas de Integridad

En lo que se refiere a las **Pruebas de Aceptación y Usabilidad del Usuario**, se ha desarrollado una prueba denominada Casa Adentro, es decir con opiniones de los integrantes del grupo de trabajo (desarrolladores). También hemos realizado pruebas como la denominada Beta que se trata de que personas ajenas al proyecto pero que conozcan el negocio naveguen por todo el portal para así poder filtrar mayor cantidad de errores y mejorar la calidad del sistema.

CAPÍTULO 6

RECOMENDACIONES Y CONCLUSIONES

6.1.1 Recomendaciones.

A continuación se menciona recomendaciones para el buen funcionamiento del sistema en cuanto a hardware y software.

6.1.2 Hardware

En cuanto al hardware recomendamos como mínimo los siguientes elementos:

Servidor

- Procesadores Pentium IV, Celeron, AMD.
- Memoria de 768 a 1G o más para el servidor.
- Disco duro de 10Gb mínimo para la instalación del sistema operativo.

Cliente

- No existe condición alguna ya que una maquina de escritorio sencilla es suficiente para poder navegar en internet y así visitar nuestro portal.
- Se recomienda tener una maquina con características sencillas conectadas a través de una red con el servidor para realizar las pruebas de navegación del portal.

6.1.3 Software

El software recomendado para el desarrollo del sistema es el siguiente:

- En lo que respecta al cliente y al servidor no existen características específicas en cuanto al sistema operativo ya que la aplicación es multiplataforma.
- En cuanto a Herramientas de programación, bases de datos y Servidor Web, es preferible utilizar versiones actualizadas, ya que pueden adaptarse a diferentes Hosting en el mercado.
- Se recomienda un análisis previo de las herramientas que se utilizaran, ya que en el momento de la implantación se presentan problemas con los Hosting existentes en el mercado, ya que muchos de estos no proveen ciertos servicios.

6.1.4 Puesta en marcha

Para la puesta en marcha de la aplicación se requiere de un dominio y de un Hosting con una capacidad mínima de 25 Mb. Este

servidor deberá contar con servicios tales como Base de Datos PostGres, Java, Php 5.0, JavaBridge.

Para poder obtener una excelente posición en el mercado, deberá este portal suscribirse a diferentes buscadores, preferentemente a los más visitados, como en la actualidad lo es GOOGLE.

Si por alguna razón no arranca, se deberá revisar las configuraciones, tomando en consideración lo que se menciona en el manual de usuario.

6.2 Conclusiones

El Portal de Ventas de Vehículos Corporativos por Internet, ha sido desarrollado con la finalidad de que las empresa o personas naturales puedan promocionar algún vehículo o repuestos destinado a la venta; y, al mismo tiempo obtener información de aquellos productos que desee adquirir.

Para la realización de este portal hemos implementado nuevas tecnologías ya sea para la investigación e implementación para el buen funcionamiento del sistema.

En el análisis del portal de Ventas de Vehículos hemos aplicado los conocimiento, experiencia en el medio real, esto se vive cuando uno egresa de la Ingeniería, por ello hemos considero que durante nuestra carrera aprendimos, aplicamos y comprobamos académica y laboralmente nuestros estudios.

Este portal se ha creado de una forma muy sencilla tratando de abarcar diferentes servicios, no solo con la promoción del vehículo si no también diferentes tipo de financiamiento a nivel bancario para satisfacer las necesidades de los usuarios.

GLOSARIO

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [J](#) [K](#) [L](#) [M](#) [N](#) [O](#) [P](#) [Q](#) [R](#) [S](#) [T](#) [U](#) [V](#) [W](#) [X](#)

[Y](#) [Z](#)

A

Ancho de Banda: Cantidad de información que se puede enviar a través de una conexión. Se mide normalmente en bits por segundos (bps).Una pagina completa de texto tiene aproximadamente 16.000 bits.

Autenticación.- Verificación de que el cliente sea quien dice ser para lograr acceso al sistema.

Applet: Pequeño programa escrito en Java que se ejecuta en Internet. Permite por ejemplo reproducir secuencias de animación o ejecutar funciones de hoja de cálculos.

Arpanet: Advanced Research Agency Network. Red Militar Norteamericana a través de líneas telefónicas de la que posteriormente derivó internet.

Administrador de la red: Persona responsable del funcionamiento, el mantenimiento y la gestión de una red.

Arquitectura cliente-servidor: Término utilizado para describir los sistemas de red de informática distribuida en los cuales las responsabilidades de la transacción se dividen en dos partes: cliente (frontal) y servidor (nodo). Ambos términos (cliente y servidor) pueden aplicarse a programas de software o a dispositivos reales de computación. También llamado informática distribuida

B

Browser (Buscador): También llamado navegador. Programa que se utiliza para ver paginas web. Netscape Navigator y Microsoft Explorer.

BACKUP : Copia de seguridad. Se hace para prevenir una posible pérdida de información.

C

Clase de Objetos (ObjectClass.-) Define la colección de atributos que pueden usarse para definir una entrada.

Cliente: usuario que interactúa con el portal Web.

Cookies: Pequeño trozo de datos que entrega el programa servidor de HTTP al navegador para que este lo guarde.

Concesionaria: Empresa que se dedica a la venta de vehículos.

Código Fuente.- Lenguaje mediante el cual fue programado el sistema.

Conexión.- Comunicación entre varias maquinas.

Cotización: Acción de consultar y comparar valores de diferentes productos que se promocionan.

.

D

Dirección IP: Todas las PC conectadas a Internet tienen una dirección numérica e irrepetible llamada dirección IP (132.248.54.10), la cual nos sirve para identificar con quien o con que se va a conectar. En razón a que resulta más sencillo recordar una cadena de palabras que una de números, por ello se creo el DNS (Sistema de Nombre de Dominios) que contiene la equivalencia entre las dos series.

Dominio: Sistema de dominación de Hosts en Internet. Los dominios van separados por un punto y jerárquicamente están organizados de derecha a izquierda.

Download: Literalmente “Bajar Cargar”, se refiere al acto de transferir unos ficheros o archivos, desde un servidor a nuestro computador. En resumen significa bajarse un programa.

F

Firewall: Literalmente “Muro de Fuego”, se trata de cualquier programa que protege a una red de otra red.

Fireware: Se llama así al software de dominio público, es decir el que no es comercial y puede distribuirse gratuitamente.

Frame (Marco): En browser de www como Netscape se refiere a una estructura de subventanas dentro de un documento de HTML.

FTP: File Transfer Protocol. Protocolo para la transferencia de Archivos entre Pc's. Las paginas FTP se utilizan principalmente para que los usuarios puedan bajar programas o información desde los sitios de Internet hasta sus equipos.

G

GIF: Formato grafico muy comprimido que produce archivos muy pequeños. Es ideal para la web, en donde se requiere que las paginas carguen rápidamente.

GIF Animado: Un archivo que contiene una serie de imágenes gif que se muestra como una secuencia rápida en el navegador, lo que produce el efecto de animación.

GUI.- Interfaces gráficas de usuario.

H

HTML: Es el lenguaje que se utiliza para diseños de paginas y hay diferentes versiones.

Home Page (Pagina Principal): Es la página principal de un sitio web. Generalmente contiene un menú principal o una tabla de contenidos con hipervínculos a otras paginas del sitio.

Hosts: Literalmente anfitrión. En internet se llama así a un computador conectado a la red, que tiene su propio número IP y nombre de dominio, que sirve para dar información a través de la www.

HTTP: Protocolo de Transferencia de Hipertexto. Protocolo que proporciona a un equipo conectado a un servidor, la información de hipertexto que existe en Internet.

I

Internet: Net (Red) Inter (Internacional), Red Internacional.

Interfaz: Permite la comunicación entre un ordenador y una persona.

IP.- Identificador único que distingue una computadora de otra.

J

Java: Lenguaje de programación que permite crear programas que funcionan en cualquier tipo de Pc's y sistemas operativos.

Java Bridge: Es un protocolo de red el que puede ser usado para conectar Script nativo, por ejemplo PHP, con Java.

Java Bridge.war: Este es un archivo zip el que contiene las librerías "Java Bridge.jar", "php-script.jar" and "script-api.jar".

Java Script: Lenguaje derivado de java, pero las instrucciones mucho mas simples.

L

Link (Enlace): Conexión a otro documento web, por medio de la dirección URL los enlaces aparecen en el texto de un documento web en forma de texto subrayado y de distinto color. Permiten al usuario presionar el botón del mouse sobre dicho texto y automáticamente saltar a otro documento, a otro servidor o enlazar a otra parte del mismo documento.

Linux: Versión Shareware del conocido sistema operativo Unix. Es un sistema multitarea, multiusuario de 32 bits para PC.

Login: Nombre que se usa para acceder a un sistema de computadoras, cuando es secreto se llama Password o clave.

M

Maquina Virtual.- Software que permite la simulación de un computador.

Multiplataforma.- Se puede trabajar sobre diferentes sistemas operativos sin que afecte su desempeño.

N

Netscape: Es un browser (Navegador), y el nombre de una empresa.

O

Open Source. - Tecnología libre de distribución por la cual no se necesita licencia.

OrganizationalPerson.- Proporciona atributos para describir a una persona que pertenezca a una organización.

P

Página web: Cualquier documento que se puede ver en el web. Un sitio web consta de una o varias paginas.

Paquetes.- Segmento de datos que se transmite de un lugar a otro.

Parámetros.- Datos que se envía a una función para que cumpla con su propósito

Password o Contraseña: Palabra o cadena de caracteres, normalmente secreta, que se utiliza como herramienta de seguridad para identificar usuarios de una aplicación, archivo o red.

Protocolo: Conjunto de Instrucciones que permite la comunicación entre computadoras

POP: Protocolo de Oficina de Correos (Post Office Protocol)
Programa cliente que se comunica con el servidor, identifica la presencia de nuevos mensajes, solicita la entre de los mismos y utiliza al servidor como oficina despachadora de correo electrónico cuando el usuario envía una carta.

Proforma: Documento externo donde se detalla características y costos de los vehículos que se han consultado anteriormente.

Puerto: Interfaz en un dispositivo de internetworking (tal como un router). En la terminología IP, un proceso de capa superior que está recibiendo información de capas más bajas. Re-escribir software o microcodificar, de forma tal que correrá en una plataforma de hardware diferente o en un ambiente de software diferente para el que fue diseñado originalmente. Un conector hembra en un patch panel que acepta un enchufe del mismo tamaño que una ficha RJ45. Los patch cords se utilizan en estos puertos para interconectar computadoras cableadas al patch panel. Es esta interconexión la que permite el funcionamiento de una LAN.

R

RAM: Random Access Memory. Memoria de Acceso Aleatorio. Varios son los tipos de memoria que se usa en las computadoras. La más conocida son las RAM. Se les llama así porque es posible dirigirse directamente a la célula donde se encuentra almacenada la

información. Su principal característica es que la información se almacena en ellas provisoriamente, pudiendo ser grabadas una y otra vez, al igual que un cassette de sonido. La memoria RAM se puede comparar a un escritorio, donde se coloca los papeles con que se va a trabajar. Mientras más grande el escritorio más papeles soporta simultáneamente para ser procesados.

ROOT: Raíz. En sistemas de ficheros se refiere al directorio raíz. En Unix se refiere al usuario principal.

Red: Conjunto de computadoras, impresoras, routers, switches, y otros dispositivos, que pueden comunicarse entre sí por algún medio de transmisión.

S

Servidor: Nodo o programa de software que brinda servicios a los clientes.

Sesión: Conjunto relacionado de transacciones de comunicación entre dos o más dispositivos de red. **Schema (esquema).**- Define que clases de objetos se pueden almacenar en el directorio, que

atributos deben contener, que atributos son opcionales y el formato de los atributos.

Software.- Programas en ejecución.

SMTP (Simple Mail Transfer Protocol): Protocolo SMTP. Protocolo de Internet que brinda servicios de correo electrónico.

U

URL: Universal Resource Locator. Nombre genérico de la dirección en Internet, Indica al usuario dónde localizar un archivo HTML determinado, en la Web. UTILIDAD

W

WEB: Site. Sitio en el World Wide Web. Conjunto de páginas Web que forman una unidad de presentación, como una revista o libro. Un sitio está formado por una colección de páginas Web
WWW (World Wide Web).

Servidor de información, desarrollado en el CERN (Laboratorio Europeo de Física de Partículas), buscando construir un sistema distribuido hipermedia e hipertexto. También llamado WEB y W3. Existen gran cantidad de clientes WWW para diferentes plataformas.

BIBLIOGRAFIA

DAUM, BERTHOLD, Eclipse 3 para Desarrolladores Java (1ra Edición, Editorial Anaya Multimedia, 2005, 624 paginas).

KRIS JAMSA, Superutilidades para HTML y Diseño Web (1ra Edición, Editorial McGraw-Hill, 2002, Capitulo 12, 900 paginas).

Ayudas PHP, (Incluido instaladores).

<http://www.todojava.awardspace.com>

<http://www.sbcodigo.com/manuales/javabasico.pdf>



UNIVERSIDAD DE GUAYAQUIL

Facultad de Ciencias Matemáticas y Física

**Carrera de Ingeniería en Sistemas
Computacionales**

“Portal de Ventas Corporativas de Vehículos por Internet”

PROYECTO DE GRADO

Previo a la Obtención del Título de:

INGENIERO EN SISTEMAS COMPUTACIONALES

Autores:

Betty Diana Ronquillo Rodríguez

María Pamela Pin

Ruth Catalina Moreira Carvajal

GUAYAQUIL – ECUADOR

Año: 2007

AGRADECIMIENTO

Es prácticamente imposible dar gracias a todas aquellas personas que de una u otra manera nos han apoyado para que este proyecto se haya podido culminar con éxito, pero cabe recalcar que Dios fue nuestro pilar fundamental ya que él nos concedió la vida y nos dio fortaleza en los momentos más difíciles de nuestra carrera.

A nuestros padres quien con su gran preocupación, consejos, apoyo y paciencia nos formaron para ser mujeres de bien y emprendedoras.

Gracias a los amigos y familiares quienes con sus consejos y confianza en nosotras, nos motivaron a seguir adelante y a no detenernos en medio del camino.

DEDICATORIA

María Pamela Pin: A Dios por darme la fortaleza para culminar este proyecto. A mi madre por su apoyo incondicional a lo largo de esta etapa de mi vida.

Diana Ronquillo Rodríguez: A mis padres quienes me apoyaron sin dudarlo, ellos que siempre confiaron en mi capacidad e inteligencia, creyeron en mí y nunca me dejaron sola ante mis problemas.

Ruth Moreira Carvajal: A mis padres por haberme apoyado en mis estudios cuando los necesite, a Dios por darme la vida y a todas las personas que me ayudaron.

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”.

(Reglamento de Graduación de la Carrera de Ingeniería en sistemas Computacionales, Art. 26)

Betty Diana Ronquillo Rodríguez

María Pamela Pín

Ruth Catalina Moreira Carvajal

RESUMEN

El presente proyecto está orientado a la creación de un portal web el cual estará encargado de promocionar de manera centralizada la venta de vehículos, como también repuestos para vehículos, ya sea a nivel corporativo o por una persona natural, proporcionando a los usuarios obtener información de una manera fácil, y segura de aquellos productos (vehículos y repuestos) que se exhibirán en este portal. Hacemos énfasis en la calidad y en la seguridad de la información que se ofrece al usuario, ya que nuestro objetivo primordial es lograr que se vendan aquellos productos exhibidos en este sitio web. El motivo fundamental que nos lleva a realizar este proyecto, es contribuir con el mercado de vehículos a través del internet, por lo que hoy en día es una herramienta muy utilizada y de fácil acceso para obtener cualquier tipo de información.

ÍNDICE GENERAL

	Pág.
CAPITULO 1	1
MANUAL TECNICO	1
1.1 Configuración del Ambiente Web	1
1.1.1 Servidor Web	2
1.1.2 PHP	3
1.1.3 Integración de PHP 5.2.0 con Apache 2.2	4
1.1.4 Motor de Base de Datos	5
1.1.5 Java (J2EE)	7
1.1.6 Integración de PHP con Java (php-java-bridge 4.2.2_j2ee)	8
1.2 Desarrollo	10
1.2.1 Base de Datos	10
1.2.2 Clases y Objetos	22
1.2.3 Gui	37
CAPITULO 2	82
MANUAL DE USUARIO	82
2.1 Autenticación del Administrador	83
2.2 Navegación del Administrador	84
2.2.1 Mantenimiento de Persona	86

2.2.2 Mantenimiento de Identificación	89
2.2.3 Mantenimiento de Dirección	91
2.2.4 Mantenimiento de Productos	92
2.2.5 Mantenimiento de Categoría	96
2.2.6 Mantenimiento de País	97
2.2.7 Mantenimiento de Marca, Modelo, Provincia,	98
Tipo de Producto y Tipo de Identificación	
2.3 Activación de Pagos	99
2.4 Servicios	100
2.5 Reportes	101
2.5.1 Preferencias de Clientes	101
2.5.2 Consultas de Clientes	102

ÍNDICE DE GRÁFICOS

GRÁFICO #1	23
Estructura de las clases – Base de Datos	23
GRÁFICO #2	35
Clases y Paquetes.Java	35
GRÁFICO #3	82
Ingreso del Sistema.....	82
GRÁFICO #4	83
Navegacion del Administrador.....	83
GRÁFICO #5	84
Opciones Administradores	84
GRÁFICO #6	85
Registro de Personas.....	85
GRÁFICO #7	86
Busqueda de Personas	86
GRÁFICO #8	87
Actualizacion de Personas	87
GRÁFICO #9	88
Ingreso de Identificacion	88

GRÁFICO #10	89
Modificacion de Identificacion.....	89
GRÁFICO #11	90
Ingreso de Direcciones.....	90
GRÁFICO #12	91
Tipos de Productos	91
GRÁFICO #13	92
Eleccion de Vehiculos o Repuestos	92
GRÁFICO #14	93
Busqueda de Vehiculos.....	93
GRÁFICO #15	94
Tipo de Paquete	94
GRÁFICO #16	95
Ingreso Busqueda y Modificacion de Categoria	95
GRÁFICO #17	96
Ingreso, Busqueda y Modificacion de Pais.....	96
GRÁFICO #18	97
Ingreso, Busqueda y Modificacion de Marca, Modelo, Provincia, Tipo de Producto y Tipo de Identificacion	97
GRÁFICO #19	98
Activacion de Paquetes	98

GRÁFICO #20 99
 Ingreso del Servicio..... 99

GRÁFICO #21 100
 Preferncia de Cliente..... 100

GRÁFICO #22 101
 Consulta de Clientes 101

ÍNDICE DE TABLAS

TABLA # 1.....	11
Creacion de las Tablas – Base de Datos	11
TABLA # 2.....	17
Funciones – Base de Datos	17
TABLA # 3.....	21
Funcion ProformaRpt – Base de Datos.....	21
TABLA # 4.....	24
Clase Conexión.java	24
TABLA # 5.....	26
Clase Productos.java	26
TABLA # 6.....	29
Clase ProductosBo.java	29
TABLA # 7.....	32
Clase ProductosDao.java.....	32
TABLA # 8.....	37
Clase Autenticar Usuarios	37
TABLA # 9.....	39
Administrador.php	39

TABLA # 10.....	43
Index.php.....	43
TABLA # 11.....	47
Producto.php	47
TABLA # 12.....	60
ProductoBuscar.php	60
TABLA # 13.....	72
Pagos.php	72
TABLA # 14.....	78
Vigencia	78

CAPITULO 1

MANUAL TÉCNICO

1.1 Configuración del ambiente Web.

Tendremos una PC que tendrá la funcionalidad de servidor que constara con un Sistema Operativo XP, para nuestro caso podrá ser cualquier sistema operativo ya que nuestro sistema se adapta a cualquier plataforma (multiplataforma).

1.1.1 Servidor WEB

En primera instancia se instalará el servidor http, para nuestro proyecto utilizaremos Apache Http 2.

- Software (libre) de código abierto.
- Adaptable a plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras.
- Implementa el protocolo HTTP.
- Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido.

Proceso de instalación:

Los datos que ingresa el usuario lo resaltaremos con negrilla, los mismos que son datos de ejemplo, como se podrá apreciar en el nombre del servidor se ingresa el nombre de la máquina.

- ✓ Bajar el servicio “Administración de IIS”, en caso de tenerlo levantado.
- ✓ Instalar el Apache HTTP Server 2.2.4:
 - ⇒ Doble click en el archivo apache_2.2.4-win32-x86-openssl-0.9.8d, para iniciar el proceso de instalación.

⇒ **Next**

⇒ License Agreement. I accept the terms in the license agreement; **Next**

Server Information:

⇒ Network Domain: **servidor**

⇒ Server Name: **servidor**

⇒ Administrator's Email Address: **diana@yahoo.com**

⇒ Install Apache HTTP Server 2.2 programs and shortcuts for: for All Users, on Port 80, as a Service --

Recommended; **Next**

⇒ Setup Type: **Typical**

⇒ Destination Folder: Dejar el predeterminado. **Next**

⇒ Ready to Install the Program: **Install**

⇒ Installation Wizard Completed: **Finish**

- ✓ Reiniciar el servicio "Administración de IIS" (Opcional).
- ✓ Reiniciar Sistema Operativo (Opcional).

1.1.2 PHP

PHP es un lenguaje de programación usado frecuentemente para la creación de contenido para sitios web con los cuales se puede programar las páginas html y los códigos de fuente. PHP es un acrónimo recursivo que significa "PHP Hypertext Pre-

lenguaje interpretado, usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios web.

Proceso de instalación

- ✓ Copiar la Carpeta php 5.2.0 (que contiene los diferentes archivos de configuración), al directorio raíz "C:\".
- ✓ Agregar la ruta de este directorio (C:\php) a la variable de entorno Path.
- ✓ Modificar el archivo php.ini:

Cambiar extensión_dir="." por extensión_dir =
"C:\php\ext".

1.1.3 Integración de PHP 5.2.0 con Apache 2.2

- ✓ Configurar el archivo C:\Archivos de programa\Apache Software Foundation\Apache2.2\conf\httpd.conf con las siguientes instrucciones:

PHPIniDir "C:\\php\\"

LoadModule php5_module "C:\\php\\php5apache2_2.dll"

AddType application/x-httpd-php .php

AddType application/x-httpd-php-source .phps.

- ✓ Reiniciar Apache 2.2.0.

1.1.4 Motor de Base de Datos

Como medio de almacenamiento de datos para este portal Web se opto por instalar PostgreSQL.

PostgreSQL es un motor de base de datos, orientadas a objeto, capaz de manejar rutinas y reglas complejas del SQL.

Desarrollador: PostgreSQL Global Development Group
Versión: 8.0.0-beta4
S.O. : Multiplataforma
Licencia: BSD
Sitio Web: <http://www.postgresql.org/>

Durante el proceso de instalación siga los siguientes pasos:

- ✓ Installation options **(Dejar opciones predeterminadas).**
- ✓ Service configuration:
Install as a service: **Seleccionar**
Service name: **PostgreSQL Database Server 8.0.0-beta4.**
Account name: **postgres**
Account domain: **SERVIDOR** (por default - es el nombre del PC, no cambiar)
Account password: **dianarr**
Verify password: **dianarr**
- ✓ Account error: User SERVIDOR\postgres was not found.
Would you like the account to be created for you? **Yes.**

- ✓ The password you specified appears weak. Would you like the installer to replace ...? **No.**
- ✓ Successfully granted the 'Log on as a service' right.
Aceptar.
- ✓ Initialise database cluster.

Initialize database cluster:	Seleccionar
Port number:	5432
Address, accept connections on all address, not just localhost:	
	Seleccionar
Locale:	C spanish
Encoding:	UNICODE
Superuser name:	postgres
Password	dianarr
Password (again)	dianarr
- ✓ Enable procedural languages. **PL/pgsql**
- ✓ Enable contrib modules.

<u>DBsize</u>	(Seleccionar)
<u>pgAdmin Sopport</u>	(Seleccionar)
<u>Crypt. Functions</u>	(Seleccionar)
<u>Table Functions</u>	(Seleccionar)
- ✓ Installation complete! **Finish**

1.1.5 Java (J2EE)

Java Platform, Enterprise Edition es una plataforma de programación, parte de la Plataforma Java, para desarrollar y ejecutar software de aplicaciones en Lenguaje de programación Java con arquitectura de n niveles distribuidos, basándose ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones.

Para el desarrollo de este portal optamos por utilizar esta herramienta, ya que nos ayuda mantener un ambiente orientado a objetos y de esta manera logramos tener por separado lo que es la lógica de negocio.

Proceso de instalación:

- ✓ Doble clic sobre icono: jdk-1_5_0_03-windows-i586-p-1.exe. (10 min). J2SE Development Kit5.0 Update 3 - License. Seguir los pasos respectivo según lo que indiquen durante el proceso de la instalación.
- ✓ Agregar a la variable de entorno Path la siguiente ruta:
C:\Archivos de programa\Java\jdk1.5.0_03\bin

1.1.6 Integración de PHP con Java (php-java-bridge 4.2.2_j2ee).

El **PHP/Java Bridge** es un protocolo de red el que puede ser usado para conectar Script nativo, como PHP con Java. La implementación genérica PHP/Java Bridge es distribuido como un archivo web llamado "JavaBridge.war", esto es un archivo zip el que contiene las librerías "JavaBridge.jar", "php-script.jar" and "script-api.jar".

- ✓ Desempaquetar php-java-bridge_4.2.2_j2ee en C:\
- ✓ Desempaquetar JavaBridge.war

En la ventana de comandos, en el directorio C:\php-java-bridge-4.2.2_j2ee, ejecutamos la sentencia `java -jar JavaBridge.war` (este paso automáticamente muestra información por consola y abre una ventana con título `http://127.0.0.1:8081/server/tests.php5/`).

Luego verificar que en el directorio C:\php-java-bridge-4.2.2_j2ee\ contenga el archivo `RESULT.html`, este debe mostrar información de la versión actual de php y java.

- ✓ A continuación se ejecuta el archivo `JavaBridge.jar` que se encuentra en el directorio C:\php-java-bridge-4.2.2_j2ee\ext\, este proceso abre automáticamente una

ventana para la selección del puerto de servlet (elegir el predeterminado).

- ✓ Ejecutar en la consola de comandos la sentencia `java -jar JavaBridge.jar -help`.
- ✓ Desempaquetar el archivo `JavaBridge.war` que se encuentra en el directorio `C:\php-java-bridge-4.2.2_j2ee\` y copiar esta carpeta (`JavaBridge`) al directorio `C:\Archivos de programa\Apache Software Foundation\Apache2.2\htdocs\`, la misma que contendrá las paginas php a publicar, para uso de este proyecto a dicha carpeta se la denominó **“CarsMarketing”**.

1.2 Desarrollo

Este Portal Web se lo desarrollo en tres capas, en la primera tenemos a los datos que los almacenamos en una base de datos llamada Publicidad la cual trabajamos con la herramienta PostGres; La segunda capa se refiere a la lógica del negocio como lo son las clases Be, Bo y Dao aquellas son empaquetadas para poder ser reutilizadas a nivel de PHP, para desarrollar estas clases se utilizó la herramienta de Programación Java; Finalmente tenemos la tercera capa que es la GUI, que fue diseñada con el lenguaje HTML(diseño) y PHP(programación-páginas dinámicas).

1.2.1 Base de Datos

La base de datos utilizada en este proyecto se denomina Publicidad, contiene 17 tablas:

- Categoría
- Directorio
- Identificación
- Marca
- Modelo
- Orden
- OrdenItem
- Pago
- Pais
- Persona
- Productos
- Proforma
- ProformaItem
- Servicio
- Tipo
- TipoIdentificación
- Vigencia

En la siguiente tabla se ilustrará el script que se utilizó para la creación de la base de datos, tablas con sus restricciones, relaciones y funciones, para esto se utilizó sintaxis propias del Leguaje SQL.

Tabla 1
Creacion de las Tablas - Base de Datos

```
CREATE DATABASE "Publicidad"
WITH OWNER = postgres
ENCODING = 'UNICODE'
TABLESPACE = pg_default;

CREATE TABLE "Categoria"
(
    "Id" int8 NOT NULL,
    "Nombre" varchar(50) NOT NULL,
    "Descripcion" varchar(100) NOT NULL,
    "Estado" bit(1) NOT NULL,
    CONSTRAINT "PK_Categoria" PRIMARY KEY ("Id"),
    CONSTRAINT "U_Categoria" UNIQUE ("Nombre")
)
WITHOUT OIDS;
ALTER TABLE "Categoria" OWNER TO postgres;

CREATE TABLE "Directorio"
(
    "Id" int8 NOT NULL,
    "IdPersona" int8 NOT NULL,
    "IdCiudad" int8 NOT NULL,
    "Direccion" varchar(100) NOT NULL,
    "Telefono" varchar(9) NOT NULL,
    "Email" varchar(100) NOT NULL,
    "Estado" bit(1) NOT NULL,
    CONSTRAINT "PK_Directorio" PRIMARY KEY ("Id"),
    CONSTRAINT "FK_Directorio_Pais" FOREIGN KEY ("IdCiudad") REFERENCES "Pais"
("Id") ON UPDATE NO ACTION ON DELETE NO ACTION,
    CONSTRAINT "FK_Directorio_Persona" FOREIGN KEY ("IdPersona") REFERENCES
"Persona" ("Id") ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITHOUT OIDS;
ALTER TABLE "Directorio" OWNER TO postgres;

CREATE TABLE "Identificacion"
(
    "Id" int8 NOT NULL,
```

```

"IdPersona" int8 NOT NULL,
"IdTipoidentificacion" int8 NOT NULL,
"Codigo" varchar(13) NOT NULL,
"Estado" bit(1) NOT NULL,
CONSTRAINT "PK_Identificacion" PRIMARY KEY ("Id"),
CONSTRAINT "FK_Identificacion_Persona" FOREIGN KEY ("IdPersona")
REFERENCES "Persona" ("Id") ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT "FK_Identificacion_Tipoidentificacion" FOREIGN KEY
("IdTipoidentificacion") REFERENCES "Tipoidentificacion" ("Id") ON UPDATE NO
ACTION ON DELETE NO ACTION
)
WITHOUT OIDS;
ALTER TABLE "Identificacion" OWNER TO postgres;

```

```

CREATE TABLE "Marca"
(
    "Id" int8 NOT NULL,
    "Nombre" varchar(50) NOT NULL,
    "Descripcion" varchar(100) NOT NULL,
    "Estado" bit(1) NOT NULL,
    CONSTRAINT "PK_Marca" PRIMARY KEY ("Id"),
    CONSTRAINT "U_Marca" UNIQUE ("Nombre")
)
WITHOUT OIDS;
ALTER TABLE "Marca" OWNER TO postgres;

```

```

CREATE TABLE "Modelo"
(
    "Id" int8 NOT NULL,
    "Nombre" varchar(50) NOT NULL,
    "Descripcion" varchar(100) NOT NULL,
    "Estado" bit(1) NOT NULL,
    CONSTRAINT "PK_Modelo" PRIMARY KEY ("Id"),
    CONSTRAINT "U_Modelo" UNIQUE ("Nombre")
)
WITHOUT OIDS;
ALTER TABLE "Modelo" OWNER TO postgres;

```

```

CREATE TABLE "Orden"
(
    "Id" int8 NOT NULL,
    "IdPersona" int8 NOT NULL,
    "Fecha" date NOT NULL,
    "Estado" bit(1) NOT NULL,
    CONSTRAINT "PK_Orden" PRIMARY KEY ("Id"),
    CONSTRAINT "FK_Orden_Persona" FOREIGN KEY ("IdPersona") REFERENCES
"Persona" ("Id") ON UPDATE NO ACTION ON DELETE NO ACTION
)

```

```
WITHOUT OIDS;
ALTER TABLE "Orden" OWNER TO postgres;

CREATE TABLE "OrdenItem"
(
    "Id" int8 NOT NULL,
    "IdOrden" int8 NOT NULL,
    "IdProducto" int8 NOT NULL,
    "Cantidad" int8 NOT NULL,
    CONSTRAINT "PK_OrdenItem" PRIMARY KEY ("Id"),
    CONSTRAINT "FK_OrdenItem_Orden" FOREIGN KEY ("IdOrden") REFERENCES
"Orden" ("Id") ON UPDATE NO ACTION ON DELETE NO ACTION,
    CONSTRAINT "OrdenItem_IdProducto_fkey" FOREIGN KEY ("IdProducto")
REFERENCES "Productos" ("Id") ON UPDATE RESTRICT ON DELETE RESTRICT
)
WITHOUT OIDS;
ALTER TABLE "OrdenItem" OWNER TO postgres;
```

```
CREATE TABLE "Pago"
(
    "Id" int8 NOT NULL,
    "IdOrden" int8 NOT NULL,
    "Fecha" date NOT NULL,
    "Observacion" varchar(100) NOT NULL,
    "Estado" bit(1) NOT NULL,
    CONSTRAINT "PK_Pago" PRIMARY KEY ("Id"),
    CONSTRAINT "FK_Pago_Orden" FOREIGN KEY ("IdOrden") REFERENCES "Orden"
("Id") ON UPDATE RESTRICT ON DELETE RESTRICT
)
WITHOUT OIDS;
ALTER TABLE "Pago" OWNER TO postgres;
```

```
CREATE TABLE "País"
(
    "Id" int8 NOT NULL,
    "Nombre" varchar(50) NOT NULL,
    "Padre" int8 NOT NULL,
    "Estado" bit(1) NOT NULL,
    CONSTRAINT "PK_País" PRIMARY KEY ("Id")
)
WITHOUT OIDS;
ALTER TABLE "País" OWNER TO postgres;
```

```
CREATE TABLE "Persona"
(
    "Id" int8 NOT NULL,
    "Apellidos" varchar(50) NOT NULL,
    "Nombres" varchar(50) NOT NULL,
```

```

"Tipo" bit(1) NOT NULL,
"Sesion" varchar(20) NOT NULL,
"Estado" bit(1) NOT NULL,
CONSTRAINT "PK_Persona" PRIMARY KEY ("Id"),
CONSTRAINT "U_Sesion" UNIQUE ("Sesion") )
WITHOUT OIDS;
ALTER TABLE "Persona" OWNER TO postgres;

```

```

CREATE TABLE "Productos"

```

```

(
  "Id" int8 NOT NULL,
  "IdCategoria" int8 NOT NULL,
  "IdPropietario" int8 NOT NULL,
  "IdTipo" int8 NOT NULL,
  "IdMarca" int8 NOT NULL,
  "IdModelo" int8 NOT NULL,
  "Codigo" varchar(10) NOT NULL,
  "Anio" int8 NOT NULL,
  "Stock" int8 NOT NULL,
  "Color" varchar(50) NOT NULL,
  "Nombre" varchar(100) NOT NULL,
  "Descripcion" varchar(100) NOT NULL,
  "Precio" numeric(10,2) NOT NULL,
  "Imagen" varchar(100) NOT NULL,
  "Nuevo" bit(1) NOT NULL,
  "Estado" bit(1) NOT NULL,
  CONSTRAINT "Productos_pkey" PRIMARY KEY ("Id"),
  CONSTRAINT "Productos_IdCategoria_fkey" FOREIGN KEY ("IdCategoria")
REFERENCES "Categoria" ("Id") ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT "Productos_IdMarca_fkey" FOREIGN KEY ("IdMarca") REFERENCES
"Marca" ("Id") ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT "Productos_IdModelo_fkey" FOREIGN KEY ("IdModelo")
REFERENCES "Modelo" ("Id") ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT "Productos_IdPropietario_fkey" FOREIGN KEY ("IdPropietario")
REFERENCES "Persona" ("Id") ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT "Productos_IdTipo_fkey" FOREIGN KEY ("IdTipo") REFERENCES
"Tipo" ("Id") ON UPDATE RESTRICT ON DELETE RESTRICT)
WITHOUT OIDS;
ALTER TABLE "Productos" OWNER TO postgres;

```

```

CREATE TABLE "Proforma"

```

```

(
  "Id" int8 NOT NULL,
  "IdPersona" int8 NOT NULL,
  "Fecha" date NOT NULL,
  "Estado" bit(1) NOT NULL,
  CONSTRAINT "PK_Proforma" PRIMARY KEY ("Id"),
  CONSTRAINT "FK_Proforma_Persona" FOREIGN KEY ("IdPersona") REFERENCES
"Persona" ("Id") ON UPDATE NO ACTION ON DELETE NO ACTION

```

```

)
WITHOUT OIDS;
ALTER TABLE "Proforma" OWNER TO postgres;
CREATE TABLE "Proformaltem"
(
    "Id" int8 NOT NULL,
    "IdProforma" int8 NOT NULL,
    "IdProducto" int8 NOT NULL,
    "Cantidad" int8 NOT NULL,
    CONSTRAINT "PK_Proformaltem" PRIMARY KEY ("Id"),
    CONSTRAINT "FK_Proformaltem_Proforma" FOREIGN KEY ("IdProforma")
REFERENCES "Proforma" ("Id") ON UPDATE NO ACTION ON DELETE NO ACTION,
    CONSTRAINT "Fk_Proformaltem_Producto" FOREIGN KEY ("IdProducto")
REFERENCES "Productos" ("Id") ON UPDATE RESTRICT ON DELETE RESTRICT
)
WITHOUT OIDS;
ALTER TABLE "Proformaltem" OWNER TO postgres;

CREATE TABLE "Servicio"
(
    "Id" int8 NOT NULL,
    "IdPersona" int8 NOT NULL,
    "Nombre" varchar(50) NOT NULL,
    "Descripcion" varchar(100) NOT NULL,
    "Estado" bit(1) NOT NULL,
    CONSTRAINT "Pk_Servicio" PRIMARY KEY ("Id"),
    CONSTRAINT "Fk_Servicio_Persona" FOREIGN KEY ("IdPersona") REFERENCES
"Persona" ("Id") ON UPDATE RESTRICT ON DELETE RESTRICT
)
WITHOUT OIDS;
ALTER TABLE "Servicio" OWNER TO postgres;

CREATE TABLE "Tipo"
(
    "Id" int8 NOT NULL,
    "Nombre" varchar(50) NOT NULL,
    "Descripcion" varchar(100) NOT NULL,
    "Estado" bit(1) NOT NULL,
    CONSTRAINT "PK_Tipo" PRIMARY KEY ("Id"),
    CONSTRAINT "U_Tipo" UNIQUE ("Nombre")
)
WITHOUT OIDS;
ALTER TABLE "Tipo" OWNER TO postgres;

CREATE TABLE "Tipoidentificacion"
(
    "Id" int8 NOT NULL,
    "Nombre" varchar(50) NOT NULL,

```

```

"Descripcion" varchar(100) NOT NULL,
"Estado" bit(1) NOT NULL,
CONSTRAINT "PK_TipoIdentificacion" PRIMARY KEY ("Id"),
CONSTRAINT "U_TipoIdentificacion" UNIQUE ("Nombre")
)
WITHOUT OIDS;
ALTER TABLE "TipoIdentificacion" OWNER TO postgres;

CREATE TABLE "Vigencia"
(
    "Id" int8 NOT NULL,
    "IdOrdenItem" int8 NOT NULL,
    "IdProducto" int8 NOT NULL,
    "FechaCaducidad" date NOT NULL,
    CONSTRAINT "PK_Vigencia" PRIMARY KEY ("Id"),
    CONSTRAINT "FK_Vigencia_OrdenItem" FOREIGN KEY ("IdOrdenItem")
REFERENCES "OrdenItem" ("Id") ON UPDATE RESTRICT ON DELETE RESTRICT,
    CONSTRAINT "Vigencia_IdProducto_fkey" FOREIGN KEY ("IdProducto")
REFERENCES "Productos" ("Id") ON UPDATE RESTRICT ON DELETE RESTRICT
)
WITHOUT OIDS;
ALTER TABLE "Vigencia" OWNER TO postgres;

```

Para la obtención de datos de una manera ordenada y estructurada creamos funciones tales como actualizar, buscar, eliminar, insertar y seleccionar, para cada una de las tablas.

La siguiente tabla muestra como ejemplo algunas de las 85 funciones que se crearon, cuyas funciones hacen referencia a la tabla Productos

Tabla 2
Funciones - Base de Datos

```
-- Function: ProductosActualizar(id int8, idcategoria int8, idpropietario int8, idtipo  
int8, idmarca int8, idmodelo int8, codigo varchar, anio int8, stock int8, color varchar,  
nombre varchar, descripcion varchar, precio numeric, imagen varchar, nuevo bit,  
estado bit)
```

```
-- DROP FUNCTION "ProductosActualizar"(int8, int8, int8, int8, int8, int8, "varchar",  
int8, int8, "varchar", "varchar", "varchar", "numeric", "varchar", "bit", "bit");
```

-- De esta manera podemos ejecutar las funciones o eliminarlas.

```
CREATE OR REPLACE FUNCTION "ProductosActualizar"(int8, int8, int8, int8,  
int8, int8, "varchar", int8, int8, "varchar", "varchar", "varchar", "numeric", "varchar",  
"bit", "bit")  
  RETURNS int8 AS  
  $BODY$  
BEGIN  
  UPDATE  
    "Productos"  
  SET  
    "IdCategoria" = IdCategoria,  
    "IdPropietario" = IdPropietario,  
    "IdTipo" = IdTipo,  
    "IdMarca" = IdMarca,  
    "IdModelo" = IdModelo,  
    "Codigo" = Codigo,  
    "Anio" = Anio,  
    "Stock" = Stock,  
    "Color" = Color,  
    "Nombre" = Nombre,  
    "Descripcion" = Descripcion,  
    "Precio" = Precio,  
    "Imagen" = Id || '.gif',  
    "Nuevo" = Nuevo,  
    "Estado" = Estado  
  WHERE  
    "Id" = Id;  
  RETURN Id;  
END;  
$BODY$  
LANGUAGE 'plpgsql' VOLATILE;  
ALTER FUNCTION "ProductosActualizar"(int8, int8, int8, int8, int8, int8, "varchar",  
int8, int8, "varchar", "varchar", "varchar", "numeric", "varchar", "bit", "bit") OWNER TO  
postgres;
```

```

CREATE OR REPLACE FUNCTION "ProductosBuscar"("varchar", "varchar",
"varchar", "varchar", "varchar", "varchar", "varchar", "varchar",
"varchar", "varchar", "varchar", "varchar", "varchar", "varchar")
RETURNS SETOF "Productos" AS
$BODY$
    SELECT
        "Id",
        "IdCategoria",
        "IdPropietario",
        "IdTipo",
        "IdMarca",
        "IdModelo",
        "Codigo",
        "Anio",
        "Stock",
        "Color",
        "Nombre",
        "Descripcion",
        "Precio",
        "Imagen",
        "Nuevo",
        "Estado"
    FROM
        "Productos"
    WHERE
        "Id" LIKE $1
        AND "IdCategoria" LIKE $2
        AND "IdPropietario" LIKE $3
        AND "IdTipo" LIKE $4
        AND "IdMarca" LIKE $5
        AND "IdModelo" LIKE $6
        AND "Codigo" LIKE $7
        AND "Anio" LIKE $8
        AND "Stock" >= CAST($9 AS INT8) -- Desde Stock MINIMO
        AND "Color" LIKE $10
        AND UPPER("Nombre") LIKE '%' || UPPER($11) || '%'
        AND UPPER("Descripcion") LIKE '%' || UPPER($12) || '%'
        AND "Precio" <= CAST($13 AS NUMERIC(10,2)) --Hasta Precio Maximo
        AND UPPER("Imagen") LIKE '%' || UPPER($14) || '%'
        AND CAST("Nuevo" AS int8) LIKE $15
        AND CAST("Estado" AS int8) LIKE $16
    ORDER BY "Id";
$BODY$
LANGUAGE 'sql' VOLATILE;
ALTER FUNCTION "ProductosBuscar"("varchar", "varchar", "varchar", "varchar",
"varchar", "varchar", "varchar", "varchar", "varchar", "varchar",
"varchar", "varchar", "varchar", "varchar") OWNER TO postgres;

```

```

CREATE OR REPLACE FUNCTION "ProductosEliminar"(int8)
  RETURNS int8 AS
$BODY$
BEGIN
  DELETE FROM
    "Productos"
  WHERE
    "Id" = Id;
  RETURN Id;
END;
$BODY$
LANGUAGE 'plpgsql' VOLATILE;
ALTER FUNCTION "ProductosEliminar"(int8) OWNER TO postgres;

```

```

CREATE OR REPLACE FUNCTION "ProductosInsertar"(int8, int8, int8, int8, int8,
"varchar", int8, int8, "varchar", "varchar", "varchar", "numeric", "varchar", "bit")
  RETURNS int8 AS
$BODY$
DECLARE
  Id "Productos"."Id"%TYPE;
  Imagenes "Productos"."Imagen"%TYPE;
BEGIN
  Id:=(SELECT MAX("Id") + 1 FROM "Productos");
  IF Id IS NULL THEN
    Id:=1;
  END IF;
  Imagenes:= Id || '.gif';
  INSERT INTO "Productos"
  (
    "Id",
    "IdCategoria",
    "IdPropietario",
    "IdTipo",
    "IdMarca",
    "IdModelo",
    "Codigo",
    "Anio",
    "Stock",
    "Color",
    "Nombre",
    "Descripcion",
    "Precio",
    "Imagen",
    "Nuevo",
    "Estado"
  )
  VALUES
  (
    Id,
    IdCategoria,

```

```

        IdPropietario,
        IdTipo,
        IdMarca,
        IdModelo,
        Codigo,
        Anio,
        Stock,
        Color,
        Nombre,
        Descripcion,
        Precio,
        Imagenes,
        Nuevo, CAST(1 AS BIT));
    RETURN Id;
END;
$BODY$
    LANGUAGE 'plpgsql' VOLATILE;
ALTER FUNCTION "ProductosInsertar"(int8, int8, int8, int8, int8, "varchar", int8, int8,
"varchar", "varchar", "varchar", "numeric", "varchar", "bit") OWNER TO postgres;

CREATE OR REPLACE FUNCTION "ProductosSeleccionar"(int8)
    RETURNS SETOF "Productos" AS
$BODY$
    SELECT
        "Id",
        "IdCategoria",
        "IdPropietario",
        "IdTipo",
        "IdMarca",
        "IdModelo",
        "Codigo",
        "Anio",
        "Stock",
        "Color",
        "Nombre",
        "Descripcion",
        "Precio",
        "Imagen",
        "Nuevo",
        "Estado"
    FROM
        "Productos"
    WHERE
        "Id" = $1;
$BODY$
    LANGUAGE 'sql' VOLATILE;
ALTER FUNCTION "ProductosSeleccionar"(int8) OWNER TO postgres;

```

Para cada una de las tablas se crearon las 5 funciones esenciales, como las que se visualizaron en la anterior tabla, pero a diferencia de las tablas Orden y Proforma se creó una sexta función como lo es OrdenRpt y ProformaRpt respectivamente, que nos sirve para obtener los datos necesarios para generar una proforma u orden; y, otra tabla que tiene un tratamiento diferente en cuanto a funciones es la de OrdenItem y Proformaltem, solo tenemos la función OrdenItemSeleccionar y ProformaltemRpt, ya que de estas dos tablas solo requerimos seleccionar los datos para mostrar el detalle de una orden o proforma y a su vez generar un reporte de ella y no tenemos porque eliminar, actualizar y mucho menos insertar datos al detalle de una proforma u orden.

En la siguiente tabla tenemos como ejemplo las funciones ProformaRpt y ProformaltemRpt, ya que las OrdenRpt y ProformaltemRpt tienen igual estructura.

Tabla 3
Funcion ProformaRpt - Base de Datos

```
CREATE OR REPLACE FUNCTION "ProformaRpt"(int8)
RETURNS SETOF record AS
$BODY$
SELECT
  a."Id", b."Apellidos", b."Nombres", d."Nombre", c."Codigo", a."Fecha", a."Estado"
FROM
  "Proforma" a,
  "Persona" b,
  "Identificacion" c,
  "Tipoidentificacion" d
WHERE
  a."Id" LIKE $1 AND
```

```

a."IdPersona" = b."Id" AND
b."Id" = c."IdPersona" AND
c."IdTipoidentificacion" = d."Id" AND
c."Estado" = CAST(1 AS BIT);
$BODY$
LANGUAGE 'sql' VOLATILE;
ALTER FUNCTION "ProformaRpt"(int8) OWNER TO postgres;

CREATE OR REPLACE FUNCTION "ProformaltemRpt"(int8)
RETURNS SETOF record AS
$BODY$
SELECT
    a."IdProducto", a."Cantidad", CAST(b."Nombre" || ' ' || c."Nombre" || ' ' || d."Nombre"
    || ' ' || e."Nombre" || ' ' || f."Nombres" || ' ' || f."Apellidos" AS VARCHAR) AS "Producto",
    b."Precio"
FROM
    "Proformaltem" a,
    "Productos" b,
    "Tipo" c,
    "Marca" d,
    "Modelo" e,
    "Persona" f
WHERE
    a."IdProforma" LIKE $1 AND
    a."IdProducto" = b."Id" AND
    b."IdTipo" = c."Id" AND
    b."IdMarca" = d."Id" AND
    b."IdModelo" = e."Id" AND
    b."IdPropietario" = f."Id"
$BODY$
LANGUAGE 'sql' VOLATILE;
ALTER FUNCTION "ProformaltemRpt"(int8) OWNER TO postgres;

```

1.2.2 Clases y Objetos

Para la construcción de la lógica del negocio optamos por desarrollar las clases en la plataforma Java, para esto nos ayudamos con una herramienta Open Source como lo es el Editor de Java Eclipse. Como lo explicamos en líneas anteriores las clases están divididas en paquetes como son Be, Bo, Dao

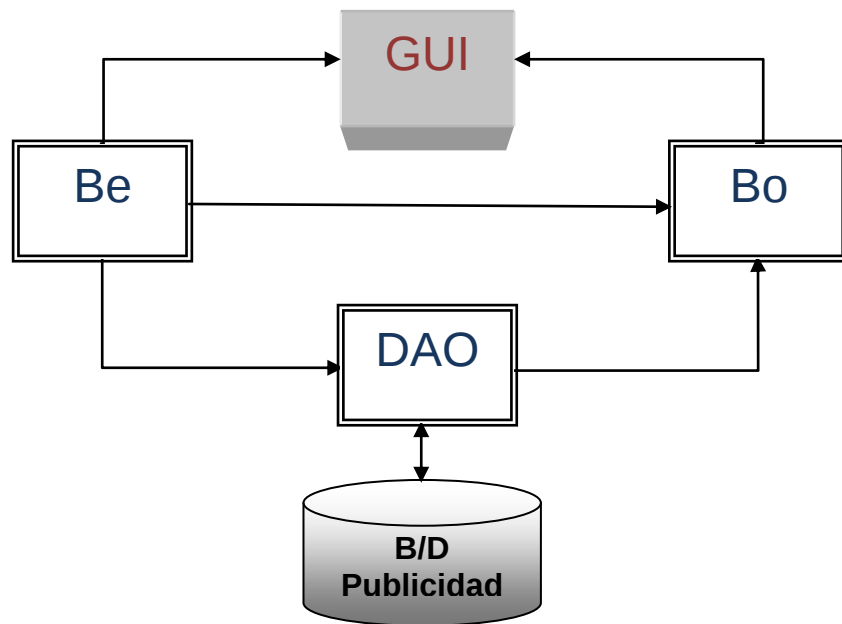


Grafico 1
Estructura de las Clases - Base de Datos

Este grafico es la representación exacta de cómo se comunican estas tres clases entre si, como y cuál es la clase que hace la conexión a la base de datos y cuales interactúan con la GUI.

Se construyo una clase denominada conexión, quien es la encargada de establecer comunicación con la base de datos por ende todas las clases existentes invocarán al método GetConexión para establecer conexión con la base, esta clase la ubicamos dentro del paquete Dao.

A continuación mostramos la clase conexión y las de Productos, como una referencia de las otras clases, ya que casi todas tienen la misma estructura.

Tabla 4
Clase Conexión.iava

```

package Dao;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class Conexion {
    private static String Provider;
    private static String DataSource;
    private static String InitialCatalog;
    private static String User;
    private static String Password;
    private static Connection con;
    public Conexion(){
        super();
    }

    public static Connection getConnection() throws ClassNotFoundException,
    SQLException{
        Provider = "postgresql";
        DataSource = "servidor";
        InitialCatalog = "Publicidad";
        User = "postgres";
        Password = "rroca";
        try {
            if ((con == null)||con.isClosed()) {
                Class.forName("org.postgresql.Driver");
                con = DriverManager.getConnection("jdbc:" + Provider + "://"
+ DataSource + "/" + InitialCatalog, User, Password);
                con.setAutoCommit(true);
            }
        } catch (ClassNotFoundException e) {
            throw new ClassNotFoundException("ClassNotFoundException:
Localición fallida del driver de conexión.");
        } catch (SQLException e) {
            throw new SQLException(e.toString());
        }
        return con;
    }

    public static Connection getConnection(String user, String password) throws
    ClassNotFoundException, SQLException {
        Provider = "postgresql";
        DataSource = "servidor";
        InitialCatalog = "Publicidad";
        User = user;
        Password = password;

        try{
            if((con == null)||con.isClosed()){
                Class.forName("org.postgresql.Driver");
                con = DriverManager.getConnection("jdbc:" +

```

```

        Provider + "://" + DataSource + "/" + InitialCatalog, User, Password);
        con.setAutoCommit(true);
    }
    } catch (ClassNotFoundException e) {
        throw new
ClassNotFoundException("ClassNotFoundException: Localición fallida del driver de
conexión.");
    } catch (SQLException e) {
        throw new SQLException(e.toString());
    }
    }
    return con;
    }

    public static Connection getConnection(String provider, String dataSource,
String initialCatalog, String user, String password) throws ClassNotFoundException,
SQLException {
        Provider = provider;
        DataSource = dataSource;
        InitialCatalog = initialCatalog;
        User = user;
        Password = password;

        try {
            if((con == null)||con.isClosed()){
                Class.forName("org.postgresql.Driver");
                con = DriverManager.getConnection("jdbc:" +
Provider + "://" + DataSource + "/" + InitialCatalog, User, Password);
                con.setAutoCommit(true);
            }
        } catch (ClassNotFoundException e) {
            throw new
ClassNotFoundException("ClassNotFoundException: Localición fallida del driver de
conexión.");
        } catch (SQLException e) {
            throw new SQLException(e.toString());
        }
        return con;
    }

    public static void disconnect() throws SQLException {
        try {
            if ( con != null ) {
                con.close();
            }
        }
        catch (SQLException e) {
            throw new SQLException("SQLException: Unable to close
the database connection.");
        }
    }
}

```

Tabla 5
Clase Productos.java

```
package Be;

public class Productos {
    private int Id;
    private int IdCategoria;
    private int IdPropietario;
    private int IdTipo;
    private int IdMarca;
    private int IdModelo;
    private StringCodigo;
    private int Anio;
    private int Stock;
    private String Color;
    private String Nombre;
    private String Descripcion;
    private float Precio;
    private String Imagen;
    private byte Nuevo;
    private byte Estado;

    public Productos () {
        this.Id = 0;
        this.IdCategoria = 0;
        this.IdPropietario = 0;
        this.IdTipo = 0;
        this.IdMarca = 0;
        this.IdModelo = 0;
        this.Codigo = "";
        this.Anio = 0;
        this.Stock = 0;
        this.Color = "";
        this.Nombre = "";
        this.Descripcion = "";
        this.Precio = 0.0f;
        this.Imagen = "";
        this.Nuevo = 1;
        this.Estado = 1;
    }

    public Productos (int id, int idCategoria, int idPropietario, int idTipo, int
idMarca, int idModelo, String codigo, int anio, int stock, String color, String nombre,
String descripcion, float precio, String imagen, byte nuevo, byte estado) {
        this.Id = id;
        this.IdCategoria = idCategoria;
        this.IdPropietario = idPropietario;
        this.IdTipo = idTipo;
        this.IdMarca = idMarca;
        this.IdModelo = idModelo;
        this.Codigo = codigo;
        this.Anio = anio;
        this.Stock = stock;
        this.Color = color;
        this.Nombre = nombre;
        this.Descripcion = descripcion;
```

```
        this.Precio = precio;
        this.Imagen = imagen;
        this.Nuevo = nuevo;
        this.Estado = estado;
    }

    public int getId() {
        return this.Id;
    }

    public void setId(int value) {
        this.Id = value;
    }

    public int getIdCategoria() {
        return this.IdCategoria;
    }

    public void setIdCategoria(int value) {
        this.IdCategoria = value;
    }

    public int getIdPropietario() {
        return this.IdPropietario;
    }

    public void setIdPropietario(int value) {
        this.IdPropietario = value;
    }

    public int getIdTipo() {
        return this.IdTipo;
    }

    public void setIdTipo(int value) {
        this.IdTipo = value;
    }

    public int getIdMarca() {
        return this.IdMarca;
    }

    public void setIdMarca(int value) {
        this.IdMarca = value;
    }

    public int getIdModelo() {
        return this.IdModelo;
    }

    public void setIdModelo(int value) {
        this.IdModelo = value;
    }
}
```

```
public String getCodigo() {  
    return this.Codigo;  
}  
  
public void setCodigo(String value) {  
    this.Codigo = value;  
}  
  
public int getAnio() {  
    return this.Anio;  
}  
  
public void setAnio(int value) {  
    this.Anio = value;  
}  
  
public int getStock() {  
    return this.Stock;  
}  
  
public void setStock(int value) {  
    this.Stock = value;  
}  
  
public String getColor() {  
    return this.Color;  
}  
  
public void setColor(String value) {  
    this.Color = value;  
}  
  
public String getNombre() {  
    return this.Nombre;  
}  
  
public void setNombre(String value) {  
    this.Nombre = value;  
}  
  
public String getDescripcion() {  
    return this.Descripcion;  
}  
  
public void setDescripcion(String value) {  
    this.Descripcion = value;  
}  
  
public float getPrecio() {  
    return this.Precio;  
}  
  
public void setPrecio(float value) {  
    this.Precio = value;  
}
```

```

    }

    public String getImagen() {
        return this.Imagen;
    }

    public void setImagen(String value) {
        this.Imagen = value;
    }

    public byte getNuevo() {
        return this.Nuevo;
    }

    public void setNuevo(byte value) {
        this.Nuevo = value;
    }

    public byte getEstado() {
        return this.Estado;
    }

    public void setEstado(byte value) {
        this.Estado = value;
    }
}

```

Tabla 6
Clase ProductosBo.iava

```

package Bo;

import java.sql.Connection;
import java.sql.SQLException;
import java.util.ArrayList;

import Be.Productos;
import Dao.Conexion;
import Dao.ProductosDao;

public class ProductosBo {
    public int guardar(Productos oProductos) throws SQLException, Exception {
        if(oProductos.getId() == 0) {
            return insertar(oProductos);
        } else {
            return actualizar(oProductos);
        }
    }

    private int insertar(Productos oProductos) throws SQLException, Exception {

```

```

        Connection con = null;
        int n = 0;

        try {
            con = Conexion.getConnection();
            ProductosDao oProductosDao = new ProductosDao(con);
            n = oProductosDao.insertar(oProductos);
            con.commit();
        } catch (Exception e) {
            if (con != null) {
                con.rollback();
                throw new Exception(e.getMessage());
            }
        } finally {
            if (con != null) {
                con.close();
            }
        }
        return n;
    }

    public Productos seleccionar(int id) throws SQLException, Exception {
        Productos oProductos = new Productos();
        Connection con = null;

        try {
            con = Conexion.getConnection();
            ProductosDao oProductosDao = new ProductosDao(con);
            oProductos = oProductosDao.seleccionar(id);
        } catch (Exception e) {
            throw new Exception(e.getMessage());
        } finally {
            if (con != null) {
                con.close();
            }
        }
        return oProductos;
    }

    private int actualizar(Productos oProductos) throws SQLException, Exception
    {
        Connection con = null;
        int n = 0;

        try {
            con = Conexion.getConnection();
            ProductosDao oProductosDao = new ProductosDao(con);
            n = oProductosDao.actualizar(oProductos);
            con.commit();
        } catch (Exception e) {
            if (con != null) {
                con.rollback();
                throw new Exception(e.getMessage());
            }
        }
    }

```

```

        } finally {
            if (con != null) {
                con.close();
            }
        }
        return n;
    }

    public int eliminar(int id) throws SQLException, Exception {
        Connection con = null;
        int n = 0;

        try {
            con = Conexion.getConnection();
            ProductosDao oProductosDao = new ProductosDao(con);
            n = oProductosDao.eliminar(id);
            con.commit();
        } catch (Exception e) {
            if (con != null) {
                con.rollback();
                throw new Exception(e.getMessage());
            }
        } finally {
            if (con != null) {
                con.close();
            }
        }
        return n;
    }

    public ArrayList buscar(String id, String idCategoria, String idPropietario,
        String idTipo, String idMarca, String idModelo, String codigo, String anio, String
        stock, String color, String nombre, String descripcion, String precio, String imagen,
        String nuevo, String estado) throws SQLException, Exception {
        ArrayList oArrayList;
        Connection con = null;

        try {
            con = Conexion.getConnection();
            ProductosDao oProductosDao = new ProductosDao(con);
            oArrayList = oProductosDao.buscar(id, idCategoria,
            idPropietario, idTipo, idMarca, idModelo, codigo, anio, stock, color, nombre,
            descripcion, precio, imagen, nuevo, estado);
        } catch (Exception e) {
            throw new Exception(e.getMessage());
        } finally {
            if (con != null) {
                con.close();
            }
        }
        return oArrayList;
    }
}

```

Tabla 7
Clase ProductosDao.iava

```
package Dao;

import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;

import Be.Productos;

public class ProductosDao {
    private Connection con;

    public ProductosDao (Connection con) {
        this.con = con;
    }

    public int insertar(Productos oProductos) throws SQLException, Exception {
        ResultSet rs;
        int n = 0;
        String oSQL = "SELECT \"ProductosInsertar\"(\"" +
oProductos.getIdCategoria() + "\", \"" + oProductos.getIdPropietario() + "\", \"" +
oProductos.getIdTipo() + "\", \"" + oProductos.getIdMarca() + "\", \"" +
oProductos.getIdModelo() + "\", \"" + oProductos.getCodigo() + "\", \"" +
oProductos.getAnio() + "\", \"" + oProductos.getStock() + "\", \"" + oProductos.getColor() +
 "\", \"" + oProductos.getNombre() + "\", \"" + oProductos.getDescripcion() + "\", \"" +
oProductos.getPrecio() + "\", \"" + oProductos.getImagen() + "\", \"" +
oProductos.getNuevo() + "\"));";
        Statement stmt = con.createStatement();
        rs = stmt.executeQuery(oSQL);
        while (rs.next()) {
            n = rs.getInt("ProductosInsertar");
        }
        rs.close();
        return n;
    }

    public Productos seleccionar(int id) throws SQLException, Exception {
        ResultSet rs;
        String oSQL = "SELECT * FROM \"ProductosSeleccionar\"(\"" + id +
");";
        Statement stmt = con.createStatement();
        rs = stmt.executeQuery(oSQL);
        Productos oProductos = new Productos();
        while (rs.next()) {
            oProductos.setId(rs.getInt("Id"));
            oProductos.setIdCategoria(rs.getInt("IdCategoria"));
            oProductos.setIdPropietario(rs.getInt("IdPropietario"));
            oProductos.setIdTipo(rs.getInt("IdTipo"));
            oProductos.setIdMarca(rs.getInt("IdMarca"));
            oProductos.setIdModelo(rs.getInt("IdModelo"));
            oProductos.setCodigo(rs.getString("Codigo"));
            oProductos.setAnio(rs.getInt("Anio"));
            oProductos.setStock(rs.getInt("Stock"));
        }
    }
}
```

```

        oProductos.setColor(rs.getString("Color"));
        oProductos.setNombre(rs.getString("Nombre"));
        oProductos.setDescripcion(rs.getString("Descripcion"));
        oProductos.setPrecio(rs.getFloat("Precio"));
        oProductos.setImagen(rs.getString("Imagen"));
        oProductos.setNuevo(rs.getBytes("Nuevo"));
        oProductos.setEstado(rs.getBytes("Estado"));
    }
    rs.close();
    return oProductos;
}

public int actualizar(Productos oProductos) throws SQLException, Exception
{
    ResultSet rs;
    int n = 0;
    String oSQL = "SELECT \"ProductosActualizar\"(" +
oProductos.getId() + ", " + oProductos.getIdCategoria() + ", " +
oProductos.getIdPropietario() + ", " + oProductos.getIdTipo() + ", " +
oProductos.getIdMarca() + ", " + oProductos.getIdModelo() + ", " +
oProductos.getCodigo() + ", " + oProductos.getAnio() + ", " + oProductos.getStock()
+ ", " + oProductos.getColor() + ", " + oProductos.getNombre() + ", " +
oProductos.getDescripcion() + ", " + oProductos.getPrecio() + ", " +
oProductos.getImagen() + ", " + oProductos.getNuevo() + ", " +
oProductos.getEstado() + ");";
    Statement stmt = con.createStatement();
    rs = stmt.executeQuery(oSQL);
    while (rs.next()) {
        n = rs.getInt("ProductosActualizar");
    }
    rs.close();
    return n;
}

public int eliminar(int id) throws SQLException, Exception {
    ResultSet rs;
    int n = 0;
    String oSQL = "SELECT \"ProductosEliminar\"(" + id + ");";
    Statement stmt = con.createStatement();
    rs = stmt.executeQuery(oSQL);

    while (rs.next()) {
        n = rs.getInt("ProductosEliminar");
    }
    rs.close();
    return n;
}

public ArrayList buscar(String id, String idCategoria, String idPropietario,
String idTipo, String idMarca, String idModelo, String codigo, String anio, String
stock, String color, String nombre, String descripcion, String precio, String imagen,
String nuevo, String estado) throws SQLException, Exception {
    ArrayList oArrayList = new ArrayList();
    ResultSet rs;

```

```

        String oSQL = "SELECT * FROM \"ProductosBuscar\"(\" + id + "\", \" + idCategoria + \"\", \" + idPropietario + \"\", \" + idTipo + \"\", \" + idMarca + \"\", \" + idModelo + \"\", \" + codigo + \"\", \" + anio + \"\", \" + stock + \"\", \" + color + \"\", \" + nombre + \"\", \" + descripcion + \"\", \" + precio + \"\", \" + imagen + \"\", \" + nuevo + \"\", \" + estado + \"\")";
        Statement stmt = con.createStatement();
        rs = stmt.executeQuery(oSQL);
        while (rs.next()) {
            Productos oProductos = new Productos();
            oProductos.setId(rs.getInt("Id"));
            oProductos.setIdCategoria(rs.getInt("IdCategoria"));
            oProductos.setIdPropietario(rs.getInt("IdPropietario"));
            oProductos.setIdTipo(rs.getInt("IdTipo"));
            oProductos.setIdMarca(rs.getInt("IdMarca"));
            oProductos.setIdModelo(rs.getInt("IdModelo"));
            oProductos.setCodigo(rs.getString("Codigo"));
            oProductos.setAnio(rs.getInt("Anio"));
            oProductos.setStock(rs.getInt("Stock"));
            oProductos.setColor(rs.getString("Color"));
            oProductos.setNombre(rs.getString("Nombre"));
            oProductos.setDescripcion(rs.getString("Descripcion"));
            oProductos.setPrecio(rs.getFloat("Precio"));
            oProductos.setImagen(rs.getString("Imagen"));
            oProductos.setNuevo(rs.getBytes("Nuevo"));
            oProductos.setEstado(rs.getBytes("Estado"));
            oArrayList.add(oProductos);
        }
        rs.close();
        return oArrayList;
    }
}

```

El gráfico que a continuación se muestra explica cómo se encuentran distribuidas las clases, es decir están divididas en paquetes.

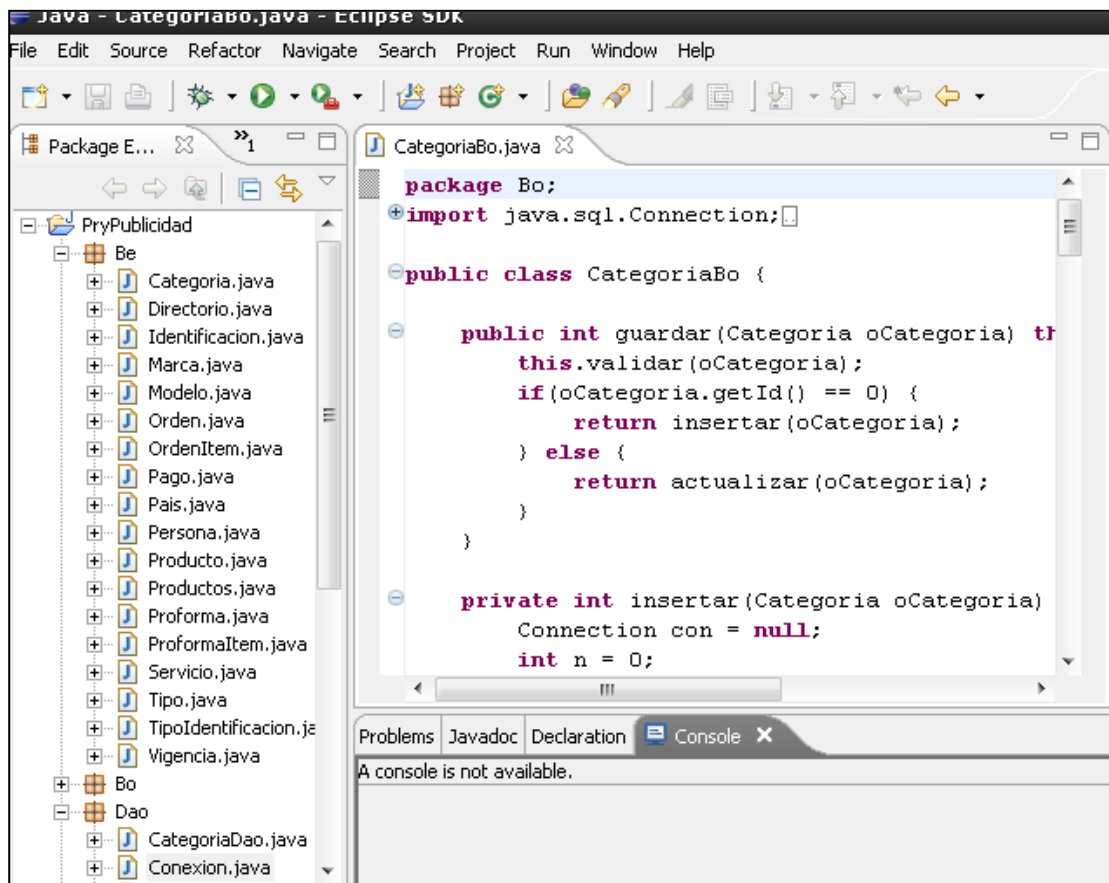


Gráfico 2
Clases y Paquetes Java

Luego para poder utilizar estas clases en el desarrollo de la GUI, es necesario empaquetar toda la carpeta PryPublicidad, resultado de esto obtendremos un archivo con extensión **jar** “**publicidad.jar**”, el mismo que se agrega a la carpeta lib que se encuentra en el directorio

Carsmarketing/WEB-INFO/lib, en esta carpeta tenemos todas las librerías que se requieren para el desarrollo de la Gui.

¿Cómo Empaquetar las Clases?

- Una vez compiladas las clases, seleccionamos solo los archivos con extensión **class** pero con sin alterar la distribución de los archivos es decir dentro de cada uno de los paquetes/carpetas respectivos.
- En la ventana de comandos nos ubicamos en el directorio raíz D:/PryPublicidad (Se requiere que la carpeta contenedora de dichos paquetes, en este caso como lo muestra la imagen se denomina PryPublicidad, se encuentre en el directorio raíz D:/)
- Una vez ubicados en dicho directorio tipear la siguiente instrucción:

```
D:/PryPublicidad> jar cvf publicidad.jar *.* ↵
```

- Luego de unos minutos completado 100% el empaquetamiento, encontraremos dentro del directorio D:/PryPublicidad> el archivo generado “**publicidad.jar**”, el mismo que se lo ubicó en el directorio de las librerías, como lo explicamos en líneas anteriores.

1.2.3 Gui

A continuación explicaremos la programación de los procesos más relevantes de este proyecto de tesis:

Para la autenticación de los usuarios tenemos el siguiente script, esta página es el punto de partida para un usuario que se identifica, es decir aquí se crea la sesión, luego del cual puede navegar dentro de una sesión, con lo que nos permite controlar los procesos por usuario.

```
<?php
session_start(); //iniciamos sesion
$idSession = session_id(); // asignamos el id de la sesion a una variable

if(!extension_loaded('java')) require_once("java/Java.inc"); // si no se encuentran
cargadas las extensiones java, se invoca al java.inc que es el fichero de
configuracion q contiene sentencias para la comunicaci3n de java con php
$session = java_session($idSession); // el id de la sesion le asignamos a una
variable
?>

<body link="#000000" vlink="#000000" alink="#000000" >
<table width="100%" border="1" bordercolor="#D8D8C4" style="border-collapse:
collapse" cellpadding="0" cellspacing="0">
    <tr>
        <td bgColor=#d8d8c4>
            <font face="Verdana" size="1"><b>Iniciar sesi3n</b></font></td>
        </tr>
        <tr>
            <td bgColor=#f6f6eb>
                <center>
                    <form method="POST" action="Autenticacion.php" >
                        <input type="hidden" name="hdnOpcion" value="">
                        <font face="Verdana" size="1">
                            <?php $mensaje = $session->get("mensaje"); if ($mensaje){
echo $mensaje; }?>
                                Usuario<br>
                                <input maxLength=19 size=10 name=txtUsuario><br>
                                Contrase1a<br>
                                <input type=password maxLength=19 size=10
name=txtContrasen1a><br>
                                C3digo de Seguridad<br>
                                <br>
                                Tipea C3digo de Seguridad<br>
                                <input maxLength=6 size=7 name=txtCodigo><br>
                                <input type=submit value=Login
                                </td>
            </tr>
        </table>
    </body>
</html>
```

```

OnClick="hdnOpcion.value='autenticar';"><br>
¿Todavía no tienes una cuenta? <A href="Index.php?op=4">Regístrate aquí</a>.
    </font>
</form>
</center>
</td>
</tr>
</table>
</body>

<?php
    global $oPersonaBo;

    $usuario = $_POST['txtUsuario'];
    $contrasenia = $_POST['txtContrasenia'];
    $cod = $_POST['txtCodigo'];
    //como el codigo de seguridad generado ya se encuentra pegado en la sesion
    entonces se preguntamos si es igual al que el usuario tipeó en la caja de texto
    txtCodigo
    if ($_SESSION['codigo'] == $cod){
        // si es igual valido que la contraseña pertenezca a ese usuario y por ende que
        este usuario exista
        autenticacion();
        $idUsuario = validar($usuario, $contrasenia);

    }
    else {
        $idUsuario = 0;
    }

    if ($_POST['hdnOpcion'] == 'autenticar') { //si es igual a autenticar es que se esta
    requiriendo autenticar un usuario
        if($idUsuario == 0) { //es cero cuando el usuario no existe o la contraseña no le
        pertenece a ese usuario
            $mensaje = '<font color=#FF0000><b>Los datos ingresados<br>son
            erroneos</b></font><br>';
        } else {
            $mensaje = ' ';
        }
        $session->put("mensaje", $mensaje); //el mensaje lo copiamos en la sesion para
        poder reutilizarlo
    }

    $session->put("idUsuario", $idUsuario); // el id de usuario lo copiamos en la sesion
    para poder reutilizarlo mientras dura la sesion

    header("Location: Index.php?"); //la funcion header nos permite cargar la pagina
    deseada sin perder la sesion

    function getDirectorio() { //esta función retorna la ruta especifica del directorio lib
        return realpath(dirname($_SERVER["SCRIPT_FILENAME"]))."/WEB-INF/lib";
    }

```

```

function autenticacion() { // creo un objeto PersonaBo
    global $oPersonaBo;

    java_require(getDirectorio()."/publicidad.jar");
    $oPersonaBo = new Java("Bo.PersonaBo");
}

function validar($usuario, $contrasenia) {
    global $oPersonaBo;
    return $oPersonaBo->autenticar($usuario, $contrasenia); //invoco el metodo
autenticar de PersonaBo y me retorna el id de ese usuario si es correcto sino retorna
cero.
}
?>

```

Tabla 8
Clase Autenticar Usuarios

Tenemos dos páginas índice tanto para el administrador como para los usuarios (compradores/vendedores); para el administrador es la página administrador.php y para los usuarios es la página index.php.

Administrador. Php

```

<!--esta pagina es invocada en index.php una vez el usuario administrador se haya
autenticado (autenticación.php) y escoja la tercera opcion que es Mantenimiento y
configuracion del sistema-->

<?php
session_start(); //iniciamos sesion
$idSession = session_id(); //asignamos la id de la sesion a una variable, esta sesion
se creo en autorización php

if(!extension_loaded('java')) require_once("java/Java.inc"); //cargamos todas las
extensiones java para poder hacer uso de las librerias propias de javas
java_require(realpath(dirname($_SERVER["SCRIPT_FILENAME"]))."/WEB-
INF/lib"."/publicidad.jar"); // invocamos la libreria publicidad.jar que fue creada por
nosotras

$session = java_session($idSession); //invoco el id de la sesion

$idUsuario=$session->get("idUsuario");
if($idUsuario){ if (($idUsuario->intValue() == 0) || ($idUsuario->intValue() > 1))
{header("Location: Index.php?");} } else {header("Location: Index.php?");}

// capturo las opciones por la que se solicito a esta pagina

```

```
if(is_null($_GET['op']))
{
    $opcion=$_POST['op'];
}else{
    $opcion=$_GET['op'];
}

//dependiendo de la opcion se incluye la pagina solicitada dentro de esta pagina
switch($opcion){
    case "1":
        $direccion = "IndexAdministrador.php";
        break;
    case "2":
        $direccion = "Categoria.php";
        break;
    case "3":
        $direccion = "CategoriaBuscar.php";
        break;
    case "4":
        $direccion = "Marca.php";
        break;
    case "5":
        $direccion = "MarcaBuscar.php";
        break;
    case "6":
        $direccion = "Modelo.php";
        break;
    case "7":
        $direccion = "ModeloBuscar.php";
        break;
    case "8":
        $direccion = "Tipo.php";
        break;
    case "9":
        $direccion = "TipoBuscar.php";
        break;
    case "10":
        $direccion = "Tipoidentificacion.php";
        break;
    case "11":
        $direccion = "TipoidentificacionBuscar.php";
        break;
    case "12":
        $direccion = "Pais.php";
        break;
    case "13":
        $direccion = "PaisBuscar.php";
        break;
    case "14":
        $direccion = "MenuCategoria.php";
        break;
    case "15":
        $direccion = "DatosPersonales.php";
        break;
```

```
case "16":
    $direccion = "PersonaBuscar.php";
    break;
case "17":
    $direccion = "Identificacion.php";
    break;
case "18":
    $direccion = "IdentificacionBuscar.php";
    break;
case "19":
    $direccion = "Directorio.php";
    break;
case "20":
    $direccion = "DirectorioBuscar.php";
    break;
case "21":
    $direccion = "Producto.php";
    break;
case "22":
    $direccion = "MenuTipo.php";
    break;
case "23":
    $direccion = "Provincia.php";
    break;
case "24":
    $direccion = "ProvinciaBuscar.php";
    break;
case "25":
    $direccion = "Ciudad.php";
    break;
case "26":
    $direccion = "CiudadBuscar.php";
    break;
case "27":
    $direccion = "Empresa.php";
    break;
case "28":
    $direccion = "Mision.php";
    break;
case "29":
    $direccion = "Vision.php";
    break;
case "30":
    $direccion = "Objetivos.php";
    break;
case "31":
    $direccion = "Pago.php"; //Link activación de pagos
    break;
case "32":
    $direccion = "Servicio.php";
    break;
case "33":
    $direccion = "ServicioBuscar.php";
    break;
```

```

        case "34":
            $direccion = "Preferencia.php"; //Link Preferencia Clintes
            break;
        case "35":
            $direccion = "Error.php";
            break;
        case "36":
            $direccion = "Cotizador.php"; //Link Consulta de Clientes
            break;
        default:
            $direccion = "IndexAdministrador.php";
            break;
    }

?>
<table border="0" cellpadding="0" cellspacing="0" style="border-collapse: collapse"
bordercolor="#111111" width="100%">
<!--ENCABEZADO-->
<tr>
<td width="100%" colspan="3"><?php include("Cabecera.php");?></td>
</tr>
<!--ESPACIO-->
<tr>
<td width="100%" colspan="3">&nbsp;</td>
</tr>
<!--CONTENIDO-->
<tr>
<td width="15%" colspan="3">
<!--MENU-->
<td width="15%" valign="top"><?php include("Menu1.php");?></td>
<!--PAGINA-->
<td width="70%" valign="top"><?php include($direccion);?></td>
<!--AUTENTICACION-->
<td width="15%" valign="top">&nbsp;</td>
</tr>
<!--ESPACIO-->
<tr>
<td width="100%" colspan="3">&nbsp;</td>
</tr>
<!--PIE-->
<tr>
<td width="100%" colspan="3"><?php include("Pie.php");?></td>
</tr>
</table>

```

Tabla 9
Administrador.Php

Index. Php

Como su nombre lo indica es el index del portal a través de esta pagina invoca y se muestra las otras paginas (interfaz) para un usuario ya sea vendedor o comprador. Tambien a través de esta pagina se enlaza al index del administrado, pagina (Script) lo explicamos en la anterior tabla.

```
<?php
session_start(); //Iniciar sesion
$idSession = session_id(); //Obtengo una id de sesion y asigno el id a una variable

if(!extension_loaded('java')) require_once("java/Java.inc");
java_require(realpath(dirname($_SERVER['SCRIPT_FILENAME']))."/WEB-INF/lib"."/publicidad.jar");
$session = java_session($idSession); //carga la sesion con el id
$idUsuario=$session->get("idUsuario"); //obtengo el idUsuario de la sesion
if ($idUsuario){
} else { // sino tiene un idUsuario la sesion, se la creamos
    $idUsuario=new Java('java.lang.Integer', '0');
    $session->put("idUsuario", $idUsuario);
}
//igualmente obtengo el modo de la sesion sino tengo le creo
$modo=$session->get("modo");
if($modo){
} else {
    $modo=new Java('java.lang.Integer', '0');
    $session->put("modo", $modo);
}
//Capturo la opcion con la que se invoco a esta pagina
if(is_null($_GET['op']))
{
    $opcion=$_POST['op'];
}else{
    $opcion=$_GET['op'];
}
if($idUsuario){ if ($idUsuario->intValue() == 0) { if ($opcion > 11) { if (($opcion != 21)
&& ($opcion != 27)) {header("Location: Index.php?");}}    } }
//si el id es cero, es un usuario no autenticado
switch($opcion){ //este case me permite asignar a la variable direccion el nombre de
la pagina que deseo incluir segun sea el caso
    case "0"://este caso es el inicio de la navegacion, es decir el home
        $session->destroy(); // se destruye la sesion en caso q exista alguna
        $session = java_session($idSession); // se carga una nueva sesion
        $idUsuario=new Java('java.lang.Integer', '0'); //se crea un idUsuario
con valor cero
        $session->put("idUsuario", $idUsuario); //se agrega este idUsuario a la
```

```
sesion
    header("Location: Index.php");
    break;
case "1": //Link Vendedor
    $modo=new Java('java.lang.Integer', '1'); //creo un modo con valor 1
    $session->put("modo", $modo); //agrego el modo a la sesion
    $direccion = "MenuCategoria.php";
    break;
case "2": //Link Comprador
    $modo=new Java('java.lang.Integer', '2'); //creo un modo con valor 2
    $session->put("modo", $modo); //agrego el modo a la sesion
    $direccion = "MenuCategoria.php";
    break;
    // estas dos opciones son el inicio de la nevegacion de un usuario
    comprador o vendedor, por lo que necesito de aqui restringir los procesos de los
    usuarios
case "3":
    $direccion = "ProductoBuscar.php";
    break;
case "4":
    $direccion = "DatosPersonales.php";
    break;
case "5":
    $direccion = "Bienvenida.php";
    break;
case "6":
    $direccion = "MenuTipo.php";
    break;
case "7":
    $direccion = "Empresa.php";
    break;
case "8":
    $direccion = "Mision.php";
    break;
case "9":
    $direccion = "Vision.php";
    break;
case "10":
    $direccion = "Objetivos.php";
    break;
case "11":
    $direccion = "Identificacion.php";
    break;
case "12":
    $direccion = "IdentificacionBuscar.php";
    break;
case "13":
    $direccion = "Directorio.php";
    break;
case "14":
    $direccion = "DirectorioBuscar.php";
    break;
case "15":
    $direccion = "Marca.php";
```

```

        break;
    case "16":
        $direccion = "MarcaBuscar.php";
        break;
    case "17":
        $direccion = "Modelo.php";
        break;
    case "18":
        $direccion = "ModeloBuscar.php";
        break;
    case "19":
        $direccion = "Vigencia.php";
        break;
    case "20":
        $direccion = "Proforma.php";
        break;
    case "21":
        $direccion = "Financiamiento.php";
        break;
    case "22":
        $direccion = "Producto.php";
        break;
    case "23":
        $direccion = "Vigencia.php";
        break;
    case "24":
        $direccion = "Servicio.php";
        break;
    case "25":
        $direccion = "ServicioBuscar.php";
        break;
    case "26":
        $direccion = "Orden.php";
        break;
    case "27":
        $direccion = "Error.php";
        break;
    case "28":
        $direccion = "Preferencia.php";
        break;
    case "29":
        $direccion = "Cotizador.php";
        break;
    default:
        $modo=new Java('java.lang.Integer', '0'); //por default el modo de la sesion
        tendra un valor cero
        $session->put("modo", $modo);
        $direccion = "Inicio.php"; //por default se incluye la pagina del home
        break;
    }
?>
<table border="0" cellpadding="0" cellspacing="0" style="border-collapse: collapse"
bordercolor="#111111" width="100%">
<!--ENCABEZADO-->

```

```

<tr>
  <td width="100%" colspan="3"><?php include("Cabecera.php");?></td>
</tr>
<!--ESPACIO-->
<tr>
  <td width="100%" colspan="3">&nbsp;</td>
</tr>
<!--CONTENIDO-->
<tr>
  <!--MENU-->
  <td width="15%" valign="top"><?php include("Menu.php");?></td>
  <!--PAGINA-->
  <td width="70%" valign="top">&nbsp;<a href="Index.php?op=21"><IMG
SRC="images/Financiamiento.gif" border=1 height="55"></a><p>
    <?php include($direccion); // en este bloque
incluimos la pagina invocada ?>
    <p><p>&nbsp;<br>
    <!--si navega como comprador o vendedor; o esta en el
home le mostramos en la parte inferior de la pagina banners diferentes-->
    <?php if ($modo->intValue() == 2) { ?>
      <IMG SRC="images/publicidadC.gif" border=1
width="130" height="126">
      <?php } if ($modo->intValue() == 1) { ?>
      <IMG SRC="images/publicidadV.gif" border=1
width="130" height="126">
      <?php } if ($opcion != 5) {
        if ($modo->intValue() == 0) { ?>
          <IMG SRC="images/Marcas.gif" border=1 width="130"
height="126">
          <?php } ?>
          <a href="http://www.autolasa.com.ec"><IMG
SRC="images/autolasa.jpg" border=1 height="126"></a>
          <?php } ?>
        </td>
      <!--AUTENTICACION-->
      <td width="15%" valign="top"><?php if (($opcion != 4) || ($opcion != 5)) { if
($idUser->intValue() == 0) {include("Autenticacion.php");}} ?></td>
      <!--Cuando idUsuario=0 (usuario no autenticado) y cuando no se incluya la
pagina de datosPersonales(opcion=5) y la pagina de bienvenida(opcion=4),
incluimos la pagina autenticacion para que se pueda autenticar-->
    </tr>
  <!--ESPACIO-->
  <tr>
    <td width="100%" colspan="3">&nbsp;</td>
  </tr>
  <!--PIE-->
  <tr>
    <td width="100%" colspan="3"><?php include("Pie.php");?></td>
  </tr>
</table>

```

Tabla 10
Index.Php

```

<table>
<tr>
<td width="5%">&nbsp;&nbsp;&nbsp;</td>
</tr>
//Producto.php
<tr>
<td width="90%">
<?php
global $opcion;
global $error;

switch($opcion){
    case "22":
        $pagina = "Index.php?op=22"; // se direcciona a la pagina producto
para agregar, mopdificary consultar todos aquellos productos que se encuentren en
un paquete.
        $error = "Index.php?op=27"; // para no generar toda la informacion respecto
al error se incluye la pagina error.php que muestra en forma general los posibles
errores.
        break;
    case "21":
        $paginaA = "Administrador.php?op=21"; // por parte del
administrador se presentaran todos los productos que promociona el portal.
        $error = "Administrador.php?op=35"; // genera errores en forma general.
        break;
} // validaciones de variables
global $oProducto;
global $oProductoBo;
global $id;
global $oCategoria;
global $oCategoriaBo;
global $oProveedor;
global $oProveedorBo;
global $oTipo;
global $oTipoBo;
global $oMarca;
global $oMarcaBo;
global $oModelo;
global $oModeloBo;
global $idUsuario;

global $oVigencia;
global $oVigenciaBo;

if(!extension_loaded('java')) require_once("java/Java.inc"); // obtenemos metodos
de java
$session = java_session($idSession); // creo una session con una id
$idUsuario=$session->get("idUsuario"); // obtengo la session con la id del usuario
Producto(); // funcion donde obtengo los metodo a utilizar en esta pagina.

```

```

if (is_null($_POST['IdOrdenItem']) || ($_POST['IdOrdenItem'] == 0) ) { // si no ha
seleccionado ningun producto
    if($idUser->intValue() == 1){
        $IdOrdenItem = 1;
        $Caduca = "3000/01/01";
        $limite = 99999999;
        $Modelo = 1;
    }
} else {          // pone informacion por defaul en las cajas de texto y los
checkbox
    $IdOrdenItem = $_POST['IdOrdenItem'];
    $Caduca = $_POST['Caduca'];
    $limite = $_POST['Limite'];
    $Modelo = $_POST['Modelo'];
}
?>

<form method="POST" action="<?php echo $pagina; ?>">
<input type="hidden" name="IdOrdenItem" value="<?php echo $IdOrdenItem; ?>">
<input type="hidden" name="Caduca" value="<?php echo $Caduca; ?>">
<input type="hidden" name="Limite" value="<?php echo $limite; ?>">
<input type="hidden" name="Modelo" value="<?php echo $Modelo; ?>">
<fieldset style="padding: 2">

    // diseño del criterio de busqueda por parte del administrador

    <legend><b><font face="Verdana" size="1">Criterios de
búsqueda...</font></b></legend>

    <table bgColor=#f6f6eb border="0" cellpadding="0" cellspacing="0" style="border-
collapse: collapse" bordercolor="#111111" width="100%" id="AutoNumber2">
        <tr>
            <td align="center"><b><font face="Verdana" size="1">Categoría</font></b></td>
            <td align="center"><b><font face="Verdana" size="1">Tipo</font></b></td>
            <td align="center"><b><font face="Verdana" size="1">Marca</font></b></td>
            <td align="center"><b><font face="Verdana" size="1">Modelo</font></b></td>
            <td align="center"><b><font face="Verdana" size="1">Año</font></b></td>
            <td align="center"><b><font face="Verdana" size="1">Nombre</font></b></td>
            <td align="center"><b><font face="Verdana"
size="1">Descripción</font></b></td>
            <td align="center"><b><font face="Verdana" size="1">Precio</font></b></td>
            <td align="center"><b><font face="Verdana" size="1">Estado</font></b></td>
            <td align="center"><b><font face="Verdana" size="1">Estado</font></b></td>
            <td align="center">&nbsp;</td>
            <td align="center">&nbsp;</td>
            <td align="center">&nbsp;</td>
        </tr>
        <tr>
            <td>
                <select size="1" name="cbxCategoria01" style="font-family: Verdana; font-
size: 8pt">

```

```

<?php
    $oArray = $oCategoriaBo->buscar("%", "%", "%", "1"); // se realiza la
    busqueda de los productos que promocionaremos en el portal
    $unaCategoria = new Java("Be.Categoria"); // se presentan en el check box
    los tipos de categorias que existen
    $unaCategoria->setNombre("Todo"); // en el chekc box se setea con la
    palabra todo por default.
    $oArray->add(0, $unaCategoria); // el primer registro esta designado solo
    para agregar un nuevo tipo de categoria que carsmarqueting promocionara.
    $oValor = java_values($oArray); //cargo lo agregado segun lo anterior.

    $cbxCategoria = $_POST['cbxCategoria01']; // dependiendo del tipo de
    producto que se promocionaran ya sea los diferentes paquetes que hemos elegido
    for ($i = 0; $i < count($oValor); $i++) {
        $oCategoria = $oValor[$i]; // guardo en un arreglo el tipo de categoria
        que hemos elegido
        $seleccionar = "";
        if ($oCategoria->getId() == $cbxCategoria) { // se el id de la categoria es igual
        al id del producto deseado
            $seleccionar = "selected"; // se selecciona el producto
        }
    }
    ?>
    <option value=<?php echo $oCategoria->getId(); ?> <?php echo
    $seleccionar; ?> > <?php echo $oCategoria->getNombre(); ?> </option> // cargo lo
    que se ha seleccionado en el check box
    <?php } ?>
</select>&nbsp;</td>

<td>
    <select size="1" name="cbxTipo01" style="font-family: Verdana; font-size: 8pt">
    // se vizualiza los tipos que le pertenecen a los tipos de categorias ingresados
    <?php
    $oArray = $oTipoBo->buscar("%", "%", "%", "1");
    $unTipo = new Java("Be.Tipo");
    $unTipo->setNombre("Todo");
    $oArray->add(0, $unTipo);
    $oValor=java_values($oArray);

    $cbxTipo= $_POST['cbxTipo01'];
    for ($i = 0; $i < count($oValor); $i++) {
        $oTipo = $oValor[$i];
        $seleccionar = "";
        if ($oTipo->getId() == $cbxTipo) {
            $seleccionar = "selected";
        }
    }
    ?>
    <option value=<?php echo $oTipo->getId(); ?> <?php echo $seleccionar; ?>
    > <?php echo $oTipo->getNombre(); ?> </option>
    <?php } ?>
</select>&nbsp;</td>

```

```

<td>
<select size="1" name="cbxMarca01" style="font-family: Verdana; font-size: 8pt">
<?php
$oSArray = $oMarcaBo->buscar("%", "%", "%", "1");
$oSunaMarca = new Java("Be.Marca");
$oSunaMarca->setNombre("Todo");
$oSArray->add(0, $oSunaMarca);
$oSValor=java_values($oSArray);

$ScbxMarca= $_POST['cbxMarca01'];
for ($Si = 0; $Si < count($oSValor); $Si++) {
    $oSMarca = $oSValor[$Si];
    $Sseleccionar = "";
    if ($oSMarca->getId() == $ScbxMarca) {
        $Sseleccionar = "selected";
    }
}
?>
<option value=<?php echo $oSMarca->getId(); ?> <?php echo $Sseleccionar;
?> > <?php echo $oSMarca->getNombre(); ?> </option>
<?php } ?>
</select>&nbsp;</td>

<td>
<select size="1" name="cbxModelo01" style="font-family: Verdana; font-size:
8pt">
<?php
$oSArray = $oSModeloBo->buscar("%", "%", "%", "1");
$oSunModelo = new Java("Be.Modelo");
$oSunModelo->setNombre("Todo");
$oSArray->add(0, $oSunModelo);
$oSValor=java_values($oSArray);

$ScbxModelo = $_POST['cbxModelo01'];
for ($Si = 0; $Si < count($oSValor); $Si++) {
    $oSModelo = $oSValor[$Si];
    $Sseleccionar = "";
    if ($oSModelo->getId() == $ScbxModelo) {
        $Sseleccionar = "selected";
    }
}
?>
<option value=<?php echo $oSModelo->getId(); ?> <?php echo $Sseleccionar;
?> > <?php echo $oSModelo->getNombre(); ?> </option>
<?php } ?>
</select>&nbsp;</td>

<td>
<input type="text" name="txtAnio01" maxlength=4 size="4"
OnKeyPress="numero();" OnBlur="numeroFormato(txtAnio01);" style="font-family:
Verdana; font-size: 8pt" value="<?php echo $_POST['txtAnio01']; ?>" >&nbsp;</td>
<td>
<input type="text" name="txtNombre01" maxlength=99 size="35" style="font-

```

```

family: Verdana; font-size: 8pt" value="<?php echo $_POST['txtNombre01']; ?>"
>&nbsp;</td>
<td>
<input type="text" name="txtDescripcion01" maxlength=99 size="35" style="font-
family: Verdana; font-size: 8pt" value="<?php echo $_POST['txtDescripcion01']; ?>"
>&nbsp;</td>
<td>
<input type="text" name="txtPrecio01" maxlength=9 size="9"
OnKeyPress="numero();" OnBlur="numeroFormato(txtPrecio01);" style="font-family:
Verdana; font-size: 8pt" value="<?php echo $_POST['txtPrecio01']; ?>" >&nbsp;</td>

<td>
<select size="1" name="cbxNuevo01" style="font-family: Verdana; font-size:
8pt">
<option value="-1" <?php if ($_POST['cbxNuevo01'] == "-1") {echo "selected";}?>
>Todo</option>
<option value="0" <?php if ($_POST['cbxNuevo01'] == "0") {echo "selected";}?>
>Usado</option>
<option value="1" <?php if ($_POST['cbxNuevo01'] == "1") {echo "selected";}?>
>Nuevo</option>
</select>&nbsp;</td>

<td><input type="submit" value="Buscar" name="btnBuscar" style="font-family:
Verdana; font-size: 8pt">&nbsp;</td>
<td><input type="reset" value="Cancelar" style="font-family: Verdana; font-size:
8pt" name="btnCancelar"></td>
<td>&nbsp;</td>
</tr>
</table>
</fieldset>
</form>

<?php
if ($_POST['hdnOpcion'] == 'guardar') {
    guardar();
}

if (is_null($_POST['cbxCategoria01']) || ($_POST['cbxCategoria01'] == 0) ) {
    $_POST['cbxCategoria01'] = "%";
}
if (is_null($_POST['cbxTipo01']) || ($_POST['cbxTipo01'] == 0) ) {
    $_POST['cbxTipo01'] = "%";
}
if (is_null($_POST['cbxMarca01']) || ($_POST['cbxMarca01'] == 0) ) {
    $_POST['cbxMarca01'] = "%";
}
if (is_null($_POST['cbxModelo01']) || ($_POST['cbxModelo01'] == 0) ) {
    $_POST['cbxModelo01'] = "%";
}
if (is_null($_POST['txtAnio01']) || ($_POST['txtAnio01'] == 0) ) {

```

```

$_POST['txtAnio01'] = "%";
}
if (is_null($_POST['txtNombre01']) || ($_POST['txtNombre01'] == "")) {
    $_POST['txtNombre01'] = "%";
}
if (is_null($_POST['txtDescripcion01']) || ($_POST['txtDescripcion01'] == "")) {
    $_POST['txtDescripcion01'] = "%";
}
if (is_null($_POST['txtPrecio01']) || ($_POST['txtPrecio01'] == 0)) {
    $_POST['txtPrecio01'] = "9999999.99";
}
if (is_null($_POST['cbxNuevo01']) || ($_POST['cbxNuevo01'] == -1)) {
    $_POST['cbxNuevo01'] = "%";
}

$oSimplesDateFormat = new Java("java.text.SimpleDateFormat");
$oSimplesDateFormat->applyPattern("yyyy-MM-dd");
$fFecha = $oSimplesDateFormat->format(new Java('java.util.Date'));

$oArrayVigencia = $oVigenciaBo->buscar("%", $idOrdenItem, "%", $fFecha);
$oValorVigencia=java_values($oArrayVigencia);

$oArrayList = $oProductoBo->buscar("%", $_POST['cbxCategoria01'], $idUsuario->intValue(), $_POST['cbxTipo01'], $_POST['cbxMarca01'], $_POST['cbxModelo01'],
"%", $_POST['txtAnio01'], "0", "%", $_POST['txtNombre01'],
$_POST['txtDescripcion01'], $_POST['txtPrecio01'], "%", $_POST['cbxNuevo01'],
"1");
$oValor=java_values($oArrayList);
$oArrayProducto = new Java('java.util.ArrayList');

for ($i = 0; $i < count($oValor); $i++) {
    $oProducto = $oValor[$i];

    for ($x = 0; $x < count($oValorVigencia); $x++) {

        $oVigencia = $oValorVigencia[$x];
        if ($oProducto->getId() == $oVigencia->getIdProducto()){
            $oArrayProducto->add(0, $oProducto);
        }
    }
}

if (count($oValorVigencia) < $limite){
    $oArrayProducto->add(0, new Java("Be.Productos"));
}

$oValor=java_values($oArrayProducto);
?>

<html>
<head>

```

```

<title>Producto</title>
</head>

<body>

</p>

<table border="1" cellpadding="0" cellspacing="0" style="border-collapse: collapse"
bordercolor="#d8d8c4" width="100%" id="AutoNumber1">
  <tr bgcolor=#d8d8c4>
    <td width="110%" align="center" colspan="17"><b>
      <font face="Verdana" size="1">PRODUCTO</font></b></td>
    </tr>
    <tr bgcolor=#f6f6eb>
      <td align="center"><b><font face="Verdana" size="1">No.</font></b></td>
      <td align="center"><b><font face="Verdana" size="1">Categoria</font></b></td>
      <td align="center"><b><font face="Verdana" size="1">Tipo</font></b></td>
      <td align="center"><b><font face="Verdana" size="1">Marca</font></b></td>
      <td align="center"><b><font face="Verdana" size="1">Modelo</font></b></td>
      <td align="center"><b><font face="Verdana" size="1">Placa</font></b></td>
      <td align="center"><b><font face="Verdana" size="1">Año</font></b></td>
      <td align="center"><b><font face="Verdana" size="1">Stock</font></b></td>
      <td align="center"><b><font face="Verdana" size="1">Color</font></b></td>
      <td align="center"><b><font face="Verdana" size="1">Nombre</font></b></td>
      <td align="center"><b><font face="Verdana"
size="1">Descripción</font></b></td>
      <td align="center"><b><font face="Verdana" size="1">Precio</font></b></td>
      <td align="center"><b><font face="Verdana" size="1">Imagen</font></b></td>
      <td align="center"><b><font face="Verdana" size="1">Nuevo</font></b></td>
      <td align="center"><b><font face="Verdana" size="1">Opción</font></b></td>
    </tr>
  <?php
    for ($i = 0; $i < count($oValor); $i++) {
      $oProducto = $oValor[$i];
    ?>

    <form enctype="multipart/form-data" name="frmProducto" method="POST"
    action="<?php echo $pagina; ?>">
      <tr bgcolor=#f6f6eb>

        <input type="hidden" name="hdnOpcion" value="buscar">
        <input type="hidden" name="hdnId" value="<?php echo $oProducto->getId();
        ?>">

        <input type="hidden" name="IdOrdenItem" value="<?php echo $IdOrdenItem;
        ?>">
        <input type="hidden" name="Caduca" value="<?php echo $Caduca; ?>">
        <input type="hidden" name="Limite" value="<?php echo $Limite; ?>">
        <input type="hidden" name="Modelo" value="<?php echo $Modelo; ?>">
        <td><b><font face="Verdana" size="1"><?php if (count($oValorVigencia) ==

```

```

$limite){ echo $i+1; } else { echo $i; } ?></font></b></td>
    <td>&nbsp;
        <select size="1" name="cbxCategoria" style="font-family: Verdana; font-size: 8pt"
OnFocus="<?php if(($oProducto->getId() > 0) && ($Modelo == 2)) { ?> alert('Campo
no modificable'); this.disabled = 1; <?php } ?>">
            <?php
                $categorias = $oCategoriaBo->buscar("%", "%", "%", "%");
                $lista = java_values($categorias);

                if ($idUsuario->intValue() > 1) {
                    for ($a = 0; $a < count($lista); $a++) {
                        $oCategoria = $lista[$a];
                        if($oCategoria->getId() == 3) {
                            $categorias->remove($a);
                            $lista = java_values($categorias);
                        }
                    }
                }

                for ($a = 0; $a < count($lista); $a++) {
                    $oCategoria = $lista[$a];

                    $seleccionar = "";

                    if ($oCategoria->getId() == $oProducto->getIdCategoria()) {
                        $seleccionar = "selected";
                    }
                }
            ?>
            <option value=<?php echo $oCategoria->getId(); ?> <?php echo $seleccionar;
?> ><?php echo $oCategoria->getNombre(); ?></option>
            <?php } ?>
        </select>&nbsp;</td>

        <td>&nbsp;
            <select size="1" name="cbxTipo" style="font-family: Verdana; font-size: 8pt"
OnFocus="<?php if(($oProducto->getId() > 0) && ($Modelo == 2)) { ?> alert('Campo
no modificable'); this.disabled = 1; <?php } ?>">
                <?php
                    $tipos = $oTipoBo->buscar("%", "%", "%", "%");
                    $lista = java_values($tipos);

                    for ($a = 0; $a < count($lista); $a++) {
                        $oTipo = $lista[$a];
                        $seleccionar = "";

                        if ($oTipo->getId() == $oProducto->getIdTipo()) {
                            $seleccionar = "selected";
                        }
                    }
                ?>
                <option value=<?php echo $oTipo->getId(); ?> <?php echo $seleccionar; ?>
><?php echo $oTipo->getNombre(); ?></option>

```

```

<?php } ?>
</select>&nbsp;</td>

<td>&nbsp;<select size="1" name="cbxMarca" style="font-family: Verdana; font-size: 8pt"
OnFocus="<?php if((\$oProducto->getId() > 0) && (\$Modelo == 2)) { ?> alert('Campo
no modificable'); this.disabled = 1; <?php } ?>">
<?php
\$marcas = \$oMarcaBo->buscar("%", "%", "%", "%");
\$lista = java_values(\$marcas);

for (\$a = 0; \$a < count(\$lista); \$a++) {
    \$oMarca = \$lista[\$a];
    \$seleccionar = "";

    if (\$oMarca->getId() == \$oProducto->getIdMarca()) {
        \$seleccionar = "selected";
    }
    ?>
    <option value=<?php echo \$oMarca->getId(); ?> <?php echo \$seleccionar; ?>
><?php echo \$oMarca->getNombre(); ?></option>
    <?php } ?>
</select>&nbsp;</td>

<td >&nbsp;<select size="1" name="cbxModelo" style="font-family: Verdana; font-size: 8pt"
OnFocus="<?php if((\$oProducto->getId() > 0) && (\$Modelo == 2)) { ?> alert('Campo
no modificable'); this.disabled = 1; <?php } ?>">
<?php
\$modelos = \$oModeloBo->buscar("%", "%", "%", "%");
\$lista = java_values(\$modelos);

for (\$a = 0; \$a < count(\$lista); \$a++) {
    \$oModelo = \$lista[\$a];
    \$seleccionar = "";

    if (\$oModelo->getId() == \$oProducto->getIdModelo()) {
        \$seleccionar = "selected";
    }
    ?>
    <option value=<?php echo \$oModelo->getId(); ?> <?php echo \$seleccionar; ?>
><?php echo \$oModelo->getNombre(); ?></option>
    <?php } ?>
</select>&nbsp;</td>

<td>&nbsp;<input type="text" name="txtCodigo" maxlength=9 size="7"
OnFocus="<?php if((\$oProducto->getId() > 0) && (\$Modelo == 2)) { ?> alert('Campo
no modificable'); this.disabled = 1; <?php } ?>" style="font-family: Verdana; font-size:
8pt" value="<?php echo \$oProducto->getIdCodigo(); ?>">&nbsp;</td>
<td>&nbsp;<input type="text" name="txtAnio" maxlength=4 size="4"

```

```

OnKeyPress="numero();" OnBlur="numeroFormato(txtAnio);" OnFocus="<?php
if(($oProducto->getId() > 0) && ($Modelo == 2)) { ?> alert('Campo no modificable');
this.disabled = 1; <?php } ?>" style="font-family: Verdana; font-size: 8pt" value=
"<?php echo $oProducto->getAnio(); ?>"&nbsp;</td>
<td>&nbsp;<input type="text" name="txtStock" maxlength=8 size="7"
OnKeyPress="numero();" OnBlur="numeroFormato(txtAnio);" style="font-family:
Verdana; font-size: 8pt" value= "<?php echo $oProducto->getStock();
?>"&nbsp;</td>
<td>&nbsp;<input type="text" name="txtColor" maxlength=49 size="10"
style="font-family: Verdana; font-size: 8pt" value= "<?php echo $oProducto-
>getColor(); ?>"&nbsp;</td>
<td>&nbsp;<input type="text" name="txtNombre" maxlength=99 size="15"
style="font-family: Verdana; font-size: 8pt" value= "<?php echo $oProducto-
>getNombre(); ?>"&nbsp;</td>
<td>&nbsp;<input type="text" name="txtDescripcion" maxlength=99 size="15"
style="font-family: Verdana; font-size: 8pt" value= "<?php echo $oProducto-
>getDescripcion(); ?>"&nbsp;</td>
<td>&nbsp;<input type="text" name="txtPrecio" maxlength=9 size="7"
OnKeyPress="numero();" OnBlur="numeroFormato(txtPrecio);" style="font-family:
Verdana; font-size: 8pt" value= "<?php echo $oProducto->getPrecio();
?>"&nbsp;</td>
<td>&nbsp;<input type="file" name="txtImagen" maxlength=99 size="15"
style="font-family: Verdana; font-size: 8pt">&nbsp;</td>
<td>&nbsp;<input type="checkbox" name="chbNuevo" value="1" <?php if
($oProducto->getNuevo() == 1 ) {echo "checked";} ?> ></td>
<td>&nbsp;<button name="btnEnviar" type="submit"
OnClick="hdnOpcion.value='guardar'"></button>&nbsp;</td>

</tr>
</form>
<?php
}
?>

</table>

&nbsp;<p>
<font face="Verdana" size="1">
<ul>
<li><b>Categoría:</b> Clase a la que pertenece el producto.</li>
<li><b>Tipo:</b> Tipo de producto.</li>
<li><b>Marca:</b> Marca del producto.</li>
<li><b>Modelo:</b> Modelo del producto.</li>
<li><b>Placa:</b> Placa asignada a un vehículo.</li>
<li><b>Año:</b> Año de producción.</li>
<li><b>Stock:</b> Cantidad de existencia.</li>
<li><b>Color:</b> Color del producto.</li>
<li><b>Nombre:</b> Nombre del producto, este campo es recomendado para
repuestos y vehículos varios.</li>

```

```
</li><b>Descripción:</b> Breve descripción de las características adicionales del
producto.</li>
</li><b>Precio:</b> Precio de venta al público incluyendo iva.</li>
</li><b>Imagen:</b> Foto del producto a publicar.</li>
</li><b>Nuevo:</b> Escoja esta opción si el producto es nuevo.</li>
</ul>
</font>
```

```
</body>
```

```
</html>
```

```
<?php
function getDirectorio() {
    return realpath(dirname($_SERVER["SCRIPT_FILENAME"]))."/WEB-INF/lib";
}

function Producto() {
    global $oProducto;
    global $oProductoBo;
    global $oCategoria;
    global $oCategoriaBo;
    global $oProveedor;
    global $oProveedorBo;
    global $oTipo;
    global $oTipoBo;
    global $oMarca;
    global $oMarcaBo;
    global $oModelo;
    global $oModeloBo;

    global $oVigencia;
    global $oVigenciaBo;

    java_require(getDirectorio())."/publicidad.jar");
    $oProducto = new Java("Be.Productos");
    $oProductoBo = new Java("Bo.ProductosBo");
    $oCategoria = new Java("Be.Categoria");
    $oCategoriaBo = new Java("Bo.CategoriaBo");
    $oProveedor = new Java("Be.Persona");
    $oProveedorBo = new Java("Bo.PersonaBo");
    $oTipo = new Java("Be.Tipo");
    $oTipoBo = new Java("Bo.TipoBo");
    $oMarca = new Java("Be.Marca");
    $oMarcaBo = new Java("Bo.MarcaBo");
    $oModelo = new Java("Be.Modelo");
    $oModeloBo = new Java("Bo.ModeloBo");

    $oVigencia = new Java("Be.Vigencia");
    $oVigenciaBo = new Java("Bo.VigenciaBo");
```

```

}

function encapsurar() {
    global $oProducto;
    global $oVigencia;
    global $idUsuario;

    $oProducto->setId($_POST['hdnId']);
    $oProducto->setIdCategoria($_POST['cbxCategoria']);
    $oProducto->setIdPropietario($idUsuario->intValue());
    $oProducto->setIdTipo($_POST['cbxTipo']);
    $oProducto->setIdMarca($_POST['cbxMarca']);
    $oProducto->setIdModelo($_POST['cbxModelo']);
    $oProducto->setCodigo($_POST['txtCodigo']);
    $oProducto->setAnio($_POST['txtAnio']);
    $oProducto->setStock($_POST['txtStock']);
    $oProducto->setColor($_POST['txtColor']);
    $oProducto->setNombre($_POST['txtNombre']);
    $oProducto->setDescripcion($_POST['txtDescripcion']);
    $oProducto->setPrecio($_POST['txtPrecio']);
    $oProducto->setImagen($_POST['txtImagen']);
    if (is_null($_POST['chbNuevo'])) {
        $oProducto->setNuevo(0);
    } else {
        $oProducto->setNuevo($_POST['chbNuevo']);
    }
    $oProducto->setEstado(1);

    $oVigencia->setIdOrdenItem($_POST['IdOrdenItem']);
    $oVigencia->setIdProducto($_POST['hdnId']);
    $oVigencia->setFechaCaducidad(new Java('java.util.Date', $_POST['Caduca']));
}

function guardar() {
    global $oProducto;
    global $oProductoBo;
    global $oVigencia;
    global $oVigenciaBo;
    global $id;

    global $error;

    try {
        encapsurar();
        $id = $oProductoBo->guardar($oProducto);

        if($oVigencia->getIdProducto()==0) {
            $oVigencia->setIdProducto($id);
            $idVigencia = $oVigenciaBo->guardar($oVigencia);
        }
    }
}

```


ProductoBuscar.php

Este archivo contiene el diseño y programación de la búsqueda, ingreso, modificaciones u otro proceso referente a los productos que se publican en este portal web.

```
<?php
global $oProducto;
global $oProductoBo;
global $id;
global $oCategoria;
global $oCategoriaBo;
global $oProveedor;
global $oProveedorBo;
global $oTipo;
global $oTipoBo;
global $oMarca;
global $oMarcaBo;
global $oModelo;
global $oModeloBo;
global $oVigencia;
global $oVigenciaBo;
global $idUsuario;

global $oProforma;
global $oProformaItem;
global $oProformaBo;

global $oOrden;
global $oOrdenItem;
global $oOrdenBo;

if(!extension_loaded('java')) require_once("java/Java.inc");
$session = java_session($idSession); //carga la sesion
$idUsuario=$session->get("idUsuario"); //obtengo el idUsuario de la sesion
$modo = $session->get("modo"); //obtengo tambien el modo de la sesion

if ($idUsuario->intValue() > 0){ //si es un usuario diferente al administrador
    $oProforma = $session->get("oProforma"); //obtenemos de la sesion la proforma
    $oOrden = $session->get("oOrden"); //obtenemos de la sesion la proforma

    if(is_null($oProforma)) {
        Proforma();
        $session->put("oProforma", $oProforma);
        $oProforma = $session->get("oProforma");
    }
    if(is_null($oOrden)) {
```

```

        Orden();
        $session->put("oOrden", $oOrden);
        $oOrden = $session->get("oOrden");
    }

}

Producto();

if (is_null($_POST['cbxCategoria01'])) {
    $cbxCategoria = $_GET['cbxCategoria01'];
} else {
    $cbxCategoria = $_POST['cbxCategoria01'];
}

if ($_POST['hdnOpcion'] == 'menu') { //para el caso que se desee seguir buscando
    if ($modo->intValue() == 1) { //si el modo=1 es vendedor
        header("Location: Index.php?op=1");
    }
    if ($modo->intValue() == 2) {
        header("Location: Index.php?op=2"); //si el modo=2 es comprador
    }
    exit;
}

if ($_POST['hdnOpcion'] == 'registrar') { // si es un usuario no autenticado o
registrado tiene la opcion de registrarse
    header("Location: Index.php?op=4");
    exit;
}

if ($_POST['hdnOpcion'] == 'cotizar') { //si desea cotizar se invoca la pagina
proforma.php
    header("Location: Index.php?op=20");
    exit;
}

if ($_POST['hdnOpcion'] == 'orden') { //si desea generar orden se invoca la pagina
orden.php
    header("Location: Index.php?op=26");
    exit;
}

if ($_POST['hdnOpcion'] == 'seleccionar') { //cuando se selecciona un producto se
setea con sus datos el objeto Proformaltem
    java_require(getDirectorio()."/publicidad.jar");
    $oProformaltem = new Java("Be.Proformaltem");
    $oProformaltem->setIdProducto($_POST['IdProducto']);
    $oProformaltem->setCantidad(1);
    $oProforma->agregarItem($oProformaltem);

    if ($cbxCategoria == 3){ //si el producto seleccionado es un paquete se genera

```

```

una proforma
    $oOrdenItem = new Java("Be.OrdenItem");
    $oOrdenItem->setIdProducto($_POST['IdProducto']);
    $oOrdenItem->setCantidad(1);
    $oOrden->agregarItem($oOrdenItem);
}
}

    $oCategoria = $oCategoriaBo->seleccionar($cbxCategoria);
?>
<body link="000000" vlink="#000000" alink="#000000">
<table>
<tr>
<td width="5%">&nbsp;&nbsp; </td>
</tr>

<tr>
<td width="90%">

<form name="frmProducto" method="POST" action="Index.php?op=3"> <!--
index.php?op=3 es productoBuscar.php-->
<input type="hidden" name="cbxCategoria01" value="<?php echo $cbxCategoria;
?>">
<input type="hidden" name="hdnOpcion" value="">
<!--Criterio de Búsqueda-->
    <fieldset style="padding: 2">
    <legend><b><font face="Verdana" size="1">Criterios de
búsqueda...</font></b></legend>

    <table border="0" bgColor=#f6f6eb cellpadding="0" cellspacing="0" style="border-
collapse: collapse" bordercolor="#111111" width="75%" id="AutoNumber2">
    <tr>
    <td width="8%" align="center"><font face="Verdana"
size="1"><b>Propietario</b></font></td>
    <td width="8%" align="center"><font face="Verdana"
size="1"><b>Tipo</b></font></td>
    <td width="8%" align="center"><font face="Verdana"
size="1"><b>Marca</b></font></td>
    <td width="8%" align="center"><font face="Verdana"
size="1"><b>Modelo</b></font></td>
    <td width="8%" align="center"><font face="Verdana" size="1"><b><?php
if($cbxCategoria == 3){ echo 'Duración';} else { echo 'Año';}?></b></font></td>
<?php
    if($cbxCategoria != 3){ // si es paquete no presenta el campo nombre
?>
    <td width="8%" align="center"><font face="Verdana"
size="1"><b>Nombre</b></font></td>
<?php } ?>
<?php
    if($cbxCategoria == 3){
?>
    <td width="8%" align="center"><font face="Verdana" size="1"><b>No. Productos
min.</b></font></td>

```

```

<?php } ?>
    <td width="9%" align="center"><font face="Verdana" size="1"><b>Precio
max.</b></font></td>
<?php
    if($cbxCategoria != 3){
?>
        <td width="9%" align="center"><font face="Verdana"
size="1"><b>Estado</b></font></td>
<?php } ?>
        <td width="9%" align="center">&nbsp;</td>
        <td width="9%" align="center">&nbsp;</td>
    </tr>

    <tr>
        <td width="8%"><select size="1" name="cbxProveedor01" style="font-family:
Verdana; font-size: 8pt">
            <?php //para cargar el ComboBox de Propietario
            if($cbxCategoria == 3){
                $oArray = $oProveedorBo->buscar("1", "%", "%", "%", "%", "1"); //en la categoria
paquetes solo deseo buscar los datos de la persona 1 es decir de Carsmarketing
que este en estado activo "1"
            }else {
                $oArray = $oProveedorBo->buscar("%", "%", "%", "%", "%", "1");
            }
            $unProveedor = new Java("Be.Persona");
            $unProveedor->setNombres("Todo");
            $oArray->add(0, $unProveedor);
            $oValor = java_values($oArray); //cargamos los valores obtenidos en la
busqueda en el arreglo valor

            $cbxProveedor= $_POST['cbxProveedor01'];

            //recorro el arreglo valor para setear el comboBox de Proveedor
            for ($i = 0; $i < count($oValor); $i++) {
                $oProveedor = $oValor[$i];
                $seleccionar = "";
                if ($oProveedor->getId() == $cbxProveedor) {
                    $seleccionar = "selected";
                }
            }
?>
            <option value=<?php echo $oProveedor->getId(); ?> <?php echo
$seleccionar; ?> > <?php echo $oProveedor->getNombres()." ".$oProveedor-
>getApellidos(); ?> </option>
            <?php } ?>
        </select>&nbsp;</td>

        <td width="8%"><select size="1" name="cbxTipo01" style="font-family: Verdana;
font-size: 8pt">

            <?php //Procesos para setear el ComboBox de Tipo
            if(is_null($_GET['cbxTipo01'])) {
                if (is_null($_POST['cbxTipo01']) || ($_POST['cbxTipo01'] == 0) ) {
                    $cbxTipo = "%";
                } else {

```

```

        $cbxTipo = $_POST['cbxTipo01'];
    }
    } else {
        $cbxTipo = $_GET['cbxTipo01'];
    }

    $oArray = $oTipoBo->buscar("", "", "", "1"); //se busca todos los tipos que
    esten en estado activo
    $oUnTipo = new Java("Be.Tipo");
    $oUnTipo->setNombre("Todo");
    $oArray->add(0, $oUnTipo); //agrego a mi arreglo en la posicion 0 la palabra Todo
    $oValor = java_values($oArray); //cargo los valores de mi arreglo a la variable
    valor

    for ($i = 0; $i < count($oValor); $i++) { //recorro el arreglo valor q contiene todos
    los tipos de productos
        $oTipo = $oValor[$i];
        $seleccionar = "";
        if ($oTipo->getId() == $cbxTipo) {
            $seleccionar = "selected";
        }
    }
    ?>
    <option value=<?php echo $oTipo->getId(); ?> <?php echo $seleccionar; ?>
    > <?php echo $oTipo->getNombre(); ?> </option>
    <?php } ?>
    </select>&nbsp;  </td>

    <td width="8%"><select size="1" name="cbxMarca01" style="font-family:
    Verdana; font-size: 8pt">
    <?php
    $oArray = $oMarcaBo->buscar("", "", "", "1");
    $oUnaMarca = new Java("Be.Marca");
    $oUnaMarca->setNombre("Todo");
    $oArray->add(0, $oUnaMarca);
    $oValor=java_values($oArray);

    $cbxMarca= $_POST['cbxMarca01'];
    for ($i = 0; $i < count($oValor); $i++) {
        $oMarca = $oValor[$i];
        $seleccionar = "";
        if ($oMarca->getId() == $cbxMarca) {
            $seleccionar = "selected";
        }
    }
    ?>
    <option value=<?php echo $oMarca->getId(); ?> <?php echo $seleccionar; ?> >
    <?php echo $oMarca->getNombre(); ?> </option>
    <?php } ?>
    </select>&nbsp;  </td>

    <td width="8%"><select size="1" name="cbxModelo01" style="font-family:
    Verdana; font-size: 8pt">
    <?php // procesos para setear el ComboBox Modelo en criterio de busquedas
    $oArray = $oModeloBo->buscar("", "", "", "1");
    $oUnModelo = new Java("Be.Modelo");
    $oUnModelo->setNombre("Todo");

```

```

$toArray->add(0, $unModelo);
$oValor=java_values($oArray);

$cbxModelo = $_POST['cbxModelo01'];
for ($i = 0; $i < count($oValor); $i++) {
    $oModelo = $oValor[$i];
    $seleccionar = "";
    if ($oModelo->getId() == $cbxModelo) {
        $seleccionar = "selected";
    }
}
?>
    <option value=<?php echo $oModelo->getId(); ?> <?php echo $seleccionar;
?> > <?php echo $oModelo->getNombre(); ?> </option>
    <?php } ?>
</select>&nbsp;</td>
    <td width="8%"><input type="text" name="txtAnio" style="font-family: Verdana;
font-size: 8pt" size="10" OnKeyPress="numero();"
OnBlur="numeroFormato(txtAnio);" value="<?php echo $_POST['txtAnio']; ?>"
>&nbsp;</td>
<?php
    if($cbxCategoria != 3){
?>
        <td width="8%"><input type="text" name="txtNombre" style="font-family:
Verdana; font-size: 8pt" size="10" value="<?php echo $_POST['txtNombre']; ?>"
>&nbsp;</td>
<?php } ?>
<?php
    if($cbxCategoria == 3){
?>
        <td width="8%"><input type="text" name="txtStock" style="font-family: Verdana;
font-size: 8pt" size="10" value="<?php echo $_POST['txtStock']; ?>" >&nbsp;</td>
<?php } ?>
        <td width="9%"><input type="text" name="txtPrecio" style="font-family: Verdana;
font-size: 8pt" size="10" OnKeyPress="numero();"
OnBlur="numeroFormato(txtPrecio);" value="<?php echo $_POST['txtPrecio']; ?>"
>&nbsp;</td>
<?php
    if($cbxCategoria != 3){
?>
        <td width="8%"><select size="1" name="cbxEstado01" style="font-family:
Verdana; font-size: 8pt">
            <?php $cbxEstado = $_POST['cbxEstado01']; ?>
            <option value="%" <?php if ($cbxEstado == '%') { echo 'selected'; } ?> > Todos
</option>
            <option value="1" <?php if ($cbxEstado == '1') { echo 'selected'; } ?> > Nuevo
</option>
            <option value="0" <?php if ($cbxEstado == '0') { echo 'selected'; } ?> > Usado
</option>
        </select>&nbsp;</td>
<?php } ?>

        <td width="9%"><input type="submit" value="Buscar" name="btnBuscar"
style="font-family: Verdana; font-size: 8pt"
OnClick="hdnOpcion.value='buscar'">&nbsp;</td>

```

```

        <td width="9%"><input type="reset" value="Cancelar" name="btnCancelar"
style="font-family: Verdana; font-size: 8pt"></td>
    </tr>
</table>
</fieldset>

<?php
//en estos procesos controlamos que si no se selecciono ningun criterio de
busqueda mostrara todos los productos existentes de dicha categoria
    if (is_null($_POST['cbxProveedor01']) || ($_POST['cbxProveedor01'] == 0) ) {
        $_POST['cbxProveedor01'] = "%";
    }
    if (is_null($_POST['cbxMarca01']) || ($_POST['cbxMarca01'] == 0) ) {
        $_POST['cbxMarca01'] = "%";
    }
    if (is_null($_POST['cbxModelo01']) || ($_POST['cbxModelo01'] == 0) ) {
        $_POST['cbxModelo01'] = "%";
    }
    if (is_null($_POST['txtAnio']) || ($_POST['txtAnio'] == 0) ) {
        $_POST['txtAnio'] = "%";
    }

    if (is_null($_POST['txtNombre']) || ($_POST['txtNombre'] == 0) ) {
        $_POST['txtNombre'] = "%";
    }
    if (is_null($_POST['txtStock']) || ($_POST['txtStock'] == 0) ) {
        $_POST['txtStock'] = "1";
    }
    if (is_null($_POST['txtPrecio']) || ($_POST['txtPrecio'] == 0) ) {
        $_POST['txtPrecio'] = "9999999.99";
    }
    if (is_null($_POST['cbxEstado01']) || ($_POST['cbxEstado01'] == 0) ) {
        $_POST['cbxEstado01'] = "%";
    }

    //invocamos el metodo buscar de la clase ProductoBo, lo cual me permite buscar
    un prodcto con ciertos criterios y lo asignamos en un arreglo
    $oArrayList = $oProductoBo->buscar("%", $cbxCategoria,
$_POST['cbxProveedor01'], $cbxTipo, $_POST['cbxMarca01'],
$_POST['cbxModelo01'], "%", $_POST['txtAnio'], $_POST['txtStock'], "%",
$_POST['txtNombre'], "%", $_POST['txtPrecio'], "%", $_POST['cbxEstado01'], "1");
    $oValor=java_values($oArrayList);
?>

<html>
<head>
<title>Producto</title>
</head>

<body>
<!--Resultado de las Búsquedas-->
</p>

```

```

<table border="1" cellpadding="0" cellspacing="0" style="border-collapse: collapse"
bordercolor="#d8d8c4" width="100%" id="AutoNumber1">
  <tr bgcolor=#d8d8c4>
    <td align="center" colspan="15"><b><font face="Verdana" size="1">Producto:
<?php echo $oCategoria->getNombre(); ?></font></b></td>
  </tr>
  <tr bgcolor=#f6f6eb>
    <td align="center"><b><font face="Verdana" size="1">Propietario</font></b></td>
    <td align="center"><b><font face="Verdana" size="1">Servicios</font></b></td>
    <td align="center"><b><font face="Verdana" size="1">Tipo</font></b></td>
    <td align="center"><b><font face="Verdana" size="1">Marca</font></b></td>
    <td align="center"><b><font face="Verdana" size="1">Modelo</font></b></td>

<?php
  if($cbxCategoria == 1){
  ?>
    <td align="center"><b><font face="Verdana" size="1">Placa</font></b></td>

<?php } ?>

    <td align="center"><b><font face="Verdana" size="1"><?php if($cbxCategoria ==
3){ echo 'Duración ds.'; } else { echo 'Año'; } ?></font></b></td>
<?php
  if($cbxCategoria != 3){
  ?>
    <td align="center"><b><font face="Verdana" size="1">Nombre</font></b></td>
<?php } ?>

    <td align="center"><b><font face="Verdana"
size="1">Descripción</font></b></td>
<?php
  if($cbxCategoria != 3){
  ?>
    <td align="center"><b><font face="Verdana" size="1">Color</font></b></td>
<?php } ?>
    <td align="center"><b><font face="Verdana" size="1"><?php if($cbxCategoria ==
3){ echo 'Cantidad'; } else { echo 'Stock'; } ?></font></b></td>

    <td align="center"><b><font face="Verdana" size="1">Precio</font></b></td>
<?php
  if($cbxCategoria != 3){
  ?>
    <td align="center"><b><font face="Verdana" size="1">Estado</font></b></td>
<?php } ?>
    <td align="center" colspan="2"><b><font face="Verdana"
size="1">Opciones</font></b></td>
  </tr>

<input type="hidden" name="IdProducto" value="">

<?php
  $oSimpleDateFormat = new Java("java.text.SimpleDateFormat"); // obtenemos un
formato de fecha

```

```

$oSimplifyDateFormat->applyPattern("yyyy-MM-dd");
$fFecha = $oSimplifyDateFormat->format(new Java('java.util.Date'));

for ($i = 0; $i < count($oValor); $i++) { //recorro el arreglo resultado de una
busqueda
    $oProducto = $oValor[$i];

    $oArray2 = $oVigenciaBo->buscar("%", "%", $oProducto->getId(), $fFecha); //
buscamos si este producto se encuentra en la tabla vigencia y que la fecha de
caducidad sea menor a la de la fecha actual
    $oValor2 = java_values($oArray2);
    if(count($oValor2) == 1){ // si se encontro ese producto en la tabla vigencia

        // obtenemos todos los datos complementarios de cada uno de los obtejos
obtenidos en la busqueda de productos
        $oCategoria = $oCategoriaBo->seleccionar($oProducto->getIdCategoria());
        $oProveedor = $oProveedorBo->seleccionar($oProducto->getIdPropietario());
        $oTipo = $oTipoBo->seleccionar($oProducto->getIdTipo());
        $oMarca = $oMarcaBo->seleccionar($oProducto->getIdMarca());
        $oModelo = $oModeloBo->seleccionar($oProducto->getIdModelo());
    }
    <tr bgcolor=#f6f6eb>
        <input type="hidden" name="hdnId" value="<?php echo $oProducto->getId();
    ?>">
        <td><font face="Verdana" size="1">&nbsp;<a href =
"javascript:ventana('Contactenos.php?op=<?php echo $oProveedor->getId(); ?>')">
<?php echo $oProveedor->getNombres(). " ". $oProveedor->getApellidos();
?></a></font>&nbsp;</td>
        <td><font face="Verdana" size="1">&nbsp;<a href =
"javascript:ventana('OtrosServicios.php?op=<?php echo $oProveedor->getId();
?>')">Otros Servicios</a></font>&nbsp;</td>
        <td><font face="Verdana" size="1">&nbsp;<?php echo $oTipo->getNombre();
?>&nbsp;</font></td>
        <td><font face="Verdana" size="1">&nbsp;<?php echo $oMarca->getNombre();
?>&nbsp;</font></td>
        <td><font face="Verdana" size="1">&nbsp;<?php echo $oModelo->getNombre();
?>&nbsp;</font></td>
    <?php
    if($ScbxCategoria == 1){
    ?>
        <td><font face="Verdana" size="1">&nbsp;<?php echo $oProducto->getCodigo();
?>&nbsp;</font></td>
    <?php }?>
        <td><font face="Verdana" size="1">&nbsp;<?php echo $oProducto->getAnio();
?>&nbsp;</font></td>
    <?php
    if($ScbxCategoria != 3){
    ?>
        <td><font face="Verdana" size="1">&nbsp;<?php echo $oProducto-
>getNombre(); ?>&nbsp;</font></td>
    <?php }?>
        <td><font face="Verdana" size="1">&nbsp;<?php echo $oProducto-
>getDescripcion(); ?>&nbsp;</font></td>
    <?php

```

```

        if($cbxCategoria != 3){
        ?>
            <td><font face="Verdana" size="1">&nbsp;<?php echo $oProducto->getColor();
        ?>&nbsp;</font></td>
        <?php }?>
            <td><font face="Verdana" size="1">&nbsp;<?php echo $oProducto->getStock();
        ?>&nbsp;</font></td>

            <td><font face="Verdana" size="1">&nbsp;<?php echo $oProducto->getPrecio();
        ?>&nbsp;</font></td>
        <?php
            if($cbxCategoria != 3){
        ?>
            <td><font face="Verdana" size="1">&nbsp;<?php if($oProducto->getNuevo() ==
        1){echo 'Nuevo';} else {echo 'Usado';} ?>&nbsp;</font></td>

            <td><font face="Verdana" size="1"><a href =
        "javascript:ventana('Imagen.php?img=<?php echo $oProducto->getImagen();
        ?>')">Ver imagen</a></font></td>
        <?php }
        ?>
            <td><?php if ($idUsuario->intValue() > 0){ ?><button name="btnEnviar"
        type="submit" OnClick="hdnOpcion.value='seleccionar'; IdProducto.value='<?php
        echo $oProducto->getId(); ?>'; "></button> <?php } ?></td>

        </tr>
        <?php
            }//if de vigencia
        }//for de productos
        ?>

    </table>
    <table width="100%" border="0" cellpadding="0" cellspacing="0">
    <tr>
        <td>&nbsp;&nbsp;&nbsp;<p></td>
        <td align="center">
            <input type="submit" value="Búsquedas" OnClick="hdnOpcion.value='menu'"
            name="btnTipo" style="font-family: Verdana; font-size: 8pt">
        <?php
            if ($idUsuario->intValue() > 0){
        ?>
            <input type="submit" value="Ver Proforma"
            OnClick="hdnOpcion.value='cotizar'" name="btnCotizar" style="font-family: Verdana;
            font-size: 8pt">
        <?php if ($modo->intValue() == 1) {?>
            <input type="submit" value="Ver Orden" name="btnOrdenar"
            OnClick="hdnOpcion.value='orden'" style="font-family: Verdana; font-size: 8pt">
        <?php
            }
        }
        if ($idUsuario->intValue() == 0){
        ?>

```

[illegible]

```
$oModelo = new Java("Be.Modelo");
$oModeloBo = new Java("Bo.ModeloBo");
$oVigencia = new Java("Be.Vigencia");
$oVigenciaBo = new Java("Bo.VigenciaBo");

$oProformaBo = new Java("Bo.ProformaBo");

$oOrdenBo = new Java("Bo.OrdenBo");
}

function Proforma() { //setea los objetos de una proforma
    global $oProforma;
    global $idUser;

    java_require(getDirectorio()."/publicidad.jar");
    $oProforma = new Java("Be.Proforma");
    $oProforma->setIdPersona($idUser->intValue());
}

function Orden() {
    global $oOrden;
    global $idUser;

    java_require(getDirectorio()."/publicidad.jar");
    $oOrden = new Java("Be.Orden");
    $oOrden->setIdPersona($idUser->intValue());
}
?>
<script src="javascript/Utilitario.js" type=text/javascript></script>
</td>
</tr>
<tr>
<td width="5%">&nbsp;&nbsp;&nbsp;</td>
</tr>
</table>
</body>
```

Tabla 12
ProductoBuscar.php

Pagos.php

Este código diseña la interfaz la cual permite al administrador activar el pago de una orden, lo cual esto permite que un usuario pueda ingresar algún producto dentro de este paquete, cuyos productos que se ingresen estarán publicados en el portal web (producto activo) durante su tiempo de vigencia.

```
<body link="#000000" vlink="#000000" alink="#000000">
<table>
<tr>
<td width="5%">&nbsp;&nbsp;&nbsp;&nbsp;</td>
</tr>

<tr>
<td width="90%">
<?php

    global $oOrdenItem;
    global $oOrden;
    global $oOrdenBo;
    global $oPersona;
    global $oPersonaBo;
    global $oIdentificacion;
    global $oIdentificacionBo;
    global $oTipoIdentificacion;
    global $oTipoIdentificacionBo;
    global $oPago;
    global $oPagoBo;
    global $oProducto;
    global $oProductoBo;

    global $idUsuario;

    if(!extension_loaded('java')) require_once("java/Java.inc");

    Pago();

?>
<!-- el siguiente form se refiere al diseño del bloque del criterio de busqueda-->
<form method="POST" action="Administrador.php?op=31" name="Busqueda">
<fieldset style="padding: 2">
    <legend><b><font face="Verdana" size="1">Criterios de
búsqueda...</font></b></legend>
```

```

<table bgColor=#f6f6eb border="0" cellpadding="0" cellspacing="0" style="border-collapse: collapse" bordercolor="#111111" width="100%" id="AutoNumber2">
  <tr>
    <td align="center"><b><font face="Verdana" size="1">No.
Orden</font></b></td>
    <td align="center"><b><font face="Verdana" size="1">Fecha
Desde</font></b></td>
    <td align="center"><b><font face="Verdana" size="1">Fecha
Hasta</font></b></td>
    <td align="center"><b><font face="Verdana" size="1">Persona</font></b></td>
    <td align="center">&nbsp;</td>
    <td align="center">&nbsp;</td>
  </tr>
  <tr>
    <td><input type="text" name="txtOrden" maxlength=8 size="8"
OnKeyPress="numero();" OnBlur="numeroFormato(txtOrden);" style="font-family:
Verdana; font-size: 8pt" value="<?php echo $_POST['txtOrden']; ?>"
>&nbsp;</td></td>

    <td><input type="text" name="txtFechaD" size="10" maxlength="10"
value="<?php echo $_POST['txtFechaD']; ?>" style="font-family: Verdana; font-size:
8pt" onfocus="blur()" onclick=show_calendar('Busqueda.txtFechaD')></td>
    <!--al ejecutarse el evento onClick en cualquiera de las 2 cajas de texto de
fechas se muestra el calendario q permite escoger la fecha deseada y a su vez esta
se le asigna a su respectiva caja de texto, esto lo logramos utilizando la funcion
show_calendar creadas en javaScript-->
    <td><input type="text" name="txtFechaH" size="10" maxlength="10"
value="<?php echo $_POST['txtFechaH']; ?>" style="font-family: Verdana; font-size:
8pt" onfocus="blur()" onclick=show_calendar('Busqueda.txtFechaH')></td>

    <td><select size="1" name="cbxPersona" style="font-family: Verdana; font-size:
8pt">
    <?php
    //en los siguientes proceso logramos setear al checkBox de persona todos los
nombres de las personas existentes en nuestra base
    $oArray = $oPersonaBo->buscar("%", "%", "%", "%", "%", "1");
    $unPersona = new Java("Be.Persona");
    $unPersona->setNombres("Todo"); //agregamos al arreglo en la posición 0 la
palabra todo
    $oArray->add(0, $unPersona);
    $oValor = java_values($oArray);

    for ($i = 0; $i < count($oValor); $i++) {
      $oPersona = $oValor[$i];
      $seleccionar = "";
      if ($oPersona->getId() == $_POST['cbxPersona']) {
        $seleccionar = "selected";
      }
    }
    ?>

```

```

        <option value=<?php echo $oPersona->getId(); ?> <?php echo $seleccionar;
?> > <?php echo $oPersona->getNombres()." ".$oPersona->getApellidos(); ?>
</option>
        <?php } ?>
    </select>&nbsp;</td>

    <td><input type="submit" value="Buscar" name="btnBuscar" style="font-family:
Verdana; font-size: 8pt">&nbsp;</td>
    <td><input type="reset" value="Cancelar" style="font-family: Verdana; font-size:
8pt" name="btnCancelar"></td>
    <td>&nbsp;</td>
</tr>
</table>
</fieldset>
</form>

<?php
if ($_POST['hdnOpcion'] == 'guardar') {
    guardar();
}

$oSimpleDateFormat = new Java("java.text.SimpleDateFormat"); // invocamos este
metodo de java para obtener un formato de fecha
$oSimpleDateFormat->applyPattern("yyyy-MM-dd"); // asignamos el formato
especifico de fecha
$fecha = $oSimpleDateFormat->format(new Java('java.util.Date')); dicho formato
se lo asignamos a una variable

if (is_null($_POST['txtOrden']) || ($_POST['txtOrden'] == 0) ) {
    $_POST['txtOrden'] = "%";
}
if (is_null($_POST['cbxPersona']) || ($_POST['cbxPersona'] == 0) ) {
    $_POST['cbxPersona'] = "%";
}
if (is_null($_POST['txtFechaD']) || ($_POST['txtFechaD'] == 0) ) {
    $_POST['txtFechaD'] = $fecha;
}
if (is_null($_POST['txtFechaH']) || ($_POST['txtFechaH'] == 0) ) {
    $_POST['txtFechaH'] = $fecha;
}
$arrayList = $oOrdenBo->buscar($_POST['txtOrden'], $_POST['cbxPersona'],
$_POST['txtFechaD'], $_POST['txtFechaH'], "1");
$ordenes = java_values($arrayList); //cargar un arreglo en java

?>
<!--Las siguientes lineas se refiere al diseño y resultado del criterio de busquedas
deseado-->
<table bgColor=#f6f6eb border="1" cellpadding="0" cellspacing="0" style="border-
collapse: collapse" bordercolor="#d8d8c4" width="100%">
    <tr bgcolor=#d8d8c4>
        <td align="center" colspan="7"><b><font face="Verdana" size="1">Activación de
Pagos</font></b></td>
    </tr>

```

```

<tr>
  <td align="center"><b><font face="Verdana" size="1">No. Orden</font></b></td>
  <td align="center"><b><font face="Verdana" size="1">Fecha</font></b></td>
  <td align="center"><b><font face="Verdana"
size="1">Identificación</font></b></td>
  <td align="center"><b><font face="Verdana" size="1">Cliente</font></b></td>
  <td align="center"><b><font face="Verdana" size="1">Valor</font></b></td>
  <td align="center"><b><font face="Verdana"
size="1">Observación</font></b></td>
  <td align="center"><b><font face="Verdana" size="1">Opción</font></b></td>
</tr>
<?php

//oOrdenes es el arreglo que contiene todo el resultado de la busqueda, y con la
funcion count obtengo el numero de registros de dicho arreglo
for ($i = 0; $i < count($oOrdenes); $i++) { // recorro el arreglo oOrdenes
  $oOrden = $oOrdenes[$i];

  $oPagoArray = $oPagoBo->buscar("%", $oOrden->getId(), $fecha, "%", "1");
  //busca si esa orden ya tiene un pago en la tabla pagos
  $oPagos = java_values($oPagoArray);

  if(count($oPagos) == 0) { si no se encontro esa orden pagada, el arreglo no
constara con registro alguno y la funcion count retorna cer
  //en las siguientes lineas los procesos son para obtener los datos que se desean
  presentar
    $oPersona = $oPersonaBo->seleccionar($oOrden->getIdPersona());

    $identificacionItems = $oIdentificacionBo->buscar("%", $oPersona->getId(), "%",
"%", "1"); //obtengo todas la identificaciones que llegara a tener una persona
    $identificacionValores = java_values($identificacionItems);

    for ($j = 0; $j < count($identificacionValores); $j++) {
      $oIdentificacion = $identificacionValores[$j]; // contiene las distintas
identificaciones de tiene una persona
    }

    $oTipoidentificacion = $oTipoidentificacionBo->seleccionar($oIdentificacion-
>getIdTipoidentificacion());
    // hemos obtenido el nombre del tipo de identificacion es decir si es cedula,
RUC o pasaporte
    $items = $oOrden->getOrdenItems();
    $valores = java_values($items);

    for ($k = 0; $k < count($valores); $k++) {
      $oOrdenItem = $valores[$k];
      $oProducto = $oProductoBo->seleccionar($oOrdenItem->getIdProducto());
      $subtotal = $oProducto->getPrecio()*$oOrdenItem->getCantidad();
      $total = $total + $subtotal; //valor total a cancelar por dicha orden
    }

  ?>
<tr>

```

```

<form method="POST" action="Administrador.php?op=31">
  <input type="hidden" name="hdnOpcion" value="">
  <input type="hidden" name="hdnIdOrden" value="<?php echo $oOrden->getId();
?>">
  <td><font face="Verdana" size="1"><?php echo $oOrden->getId(); ?></font></td>
  <td><font face="Verdana" size="1"><?php echo $oSimpleDateFormat-
>format($oOrden->getFecha());?></font></td>
  <td><font face="Verdana" size="1"><?php echo $oTipoidentificacion-
>getNombre().": ".$oidentificacion->getCodigo(); ?></font></td>
  <td><font face="Verdana" size="1"><?php echo $oPersona->getNombres() ." ".
$oPersona->getApellidos(); ?></font></td>
  <td><font face="Verdana" size="1"><?php echo $total; ?></font></td>
  <td><input type="text" name="txtObservacion" maxlength=99 size="35"
style="font-family: Verdana; font-size: 8pt" value=""></td>
  <td>
    <button name="btnEnviar" type="submit"
OnClick="hdnOpcion.value='guardar';" style="width: 35; height: 26"></button></td>
</form>
</tr>

<?php
} //if de pagos
} // for de ordenes, en el q dentro de este se obtuvo los datos de cada orden y los
mostro en el bloque de activacion de pagos
?>

</table>
<?php
function getDirectorio() {
  return realpath(dirname($_SERVER["SCRIPT_FILENAME"]))."/WEB-INF/lib";
//retorna la ruta exacta del directorio donde se encuentra la libreria publicidad.jar
}

function Pago() { //esta funcion crea lo diferentes objetos que se requieren
  global $oOrden;
  global $oOrdenItem;
  global $oOrdenBo;
  global $oPersona;
  global $oPersonaBo;
  global $oidentificacion;
  global $oidentificacionBo;
  global $oTipoidentificacion;
  global $oTipoidentificacionBo;
  global $oPago;
  global $oPagoBo;
  global $oProducto;
  global $oProductoBo;

  java_require(getDirectorio()."/publicidad.jar");

  $oOrden = new Java("Be.Orden");

```

```

$ordenItem = new Java("Be.OrdenItem");
$ordenBo = new Java("Bo.OrdenBo");
$persona = new Java("Be.Persona");
$personaBo = new Java("Bo.PersonaBo");
$identificacion = new Java("Be.Identificacion");
$identificacionBo = new Java("Bo.IdentificacionBo");
$tipoidentificacion = new Java("Be.Tipoidentificacion");
$tipoidentificacionBo = new Java("Bo.TipoidentificacionBo");
$pago = new Java("Be.Pago");
$pagoBo = new Java("Bo.PagoBo");
$producto = new Java("Be.Productos");
$productoBo = new Java("Bo.ProductosBo");
}

function encapsurar() { //para almacenar el pago de una orden solo requerimos
enviar como parametros el id de la orden y el contenido del campo observacion y
este almacenamiento se lo hace a nivel de base de datos
    global $pago;

    $pago->setIdOrden($_POST['hdnIdOrden']);
    $pago->setObservacion($_POST['txtObservacion']);
}

function guardar() { //para activar una orden y a su vez se genera un registro en la
tabla pagos
    global $pago;
    global $pagoBo;

    try {
        encapsurar();
        $id = $pagoBo->guardar($pago); // el metodo guardar de la clase PagoBo
retorna el id del pago que se guardo
    } catch (Exception $ex) { //este proceso es solo para la captura de errores
        $cause = $ex->getCause();
        if(is_null($ex->getCause())) $cause = $ex;
        header("Location: Administrador.php?op=35");
        //echo "$cause <br>\nin file: {$ex->getFile()}<br>\nline:{$ex->getLine()}\n";
        exit (2);
    }
}

?>
<script src="javaScript/Utilitario.js" type="text/javascript"></script>
<script src="javaScript/date-picker.js" type="text/javascript"></script>

</td>
</tr>
<tr>
<td width="5%"></td>
</tr>
</table>
</body>

```

Tabla 13
Pagos.Php

Vigencia.php

Este script realiza la función de mostrar todas aquellas ordenes que se encuentran vigente,,,

```
<body link="000000" vlink="#000000" alink="#000000">
<table>
<tr>
<td width="5%">&nbsp;&nbsp;&nbsp;</td>
</tr>

<tr>
<td width="90%">
<?php

    global $oOrdenItem;
    global $oOrden;
    global $oOrdenBo;
    global $oPago;
    global $oPagoBo;
    global $oProducto;
    global $oProductoBo;

    global $oTipo;
    global $oTipoBo;
    global $oMarca;
    global $oMarcaBo;
    global $oModelo;
    global $oModeloBo;

    if(!extension_loaded('java')) require_once("java/Java.inc"); //carga las librerias java
    para poder reutilizarlas
    $session = java_session($idSession); //levantamos sesion
    $idUser=$session->get("idUser"); //obtengo el idUsuario de la sesion

    Vigencia(); //creo los objetos necesarios para utilizar los datos de la tabla vigencia

    $oSimpleDateFormat = new Java("java.text.SimpleDateFormat");
    $oSimpleDateFormat->applyPattern("yyyy-MM-dd");
    $fecha = new Java("java.util.Date"); //obtiene la fecha del sistema

    $oArrayList = $oOrdenBo->buscar("%", $idUser->intValue(), '1900-01-01',
    $fecha, "1"); //buscamos todas las ordenes del usuario que este en la sesion que se
    hayan generado hasta la fecha actual
    $oOrdenes = java_values($oArrayList);
```

```

?>
<table bgColor=#f6f6eb border="1" cellpadding="0" cellspacing="0" style="border-
collapse: collapse" bordercolor="#d8d8c4" width="100%">
  <tr bgcolor=#d8d8c4>
    <td align="center" colspan="8"><b><font face="Verdana" size="1">Vigencia de
Paquetes</font></b></td>
  </tr>
  <tr>
    <td align="center"><b><font face="Verdana"
size="1">No.<br>Orden</font></b></td>
    <td align="center"><b><font face="Verdana" size="1">Cantidad</font></b></td>
    <td align="center"><b><font face="Verdana" size="1">Paquete</font></b></td>
    <td align="center"><b><font face="Verdana" size="1">Límite
de<br>Elementos</font></b></td>
    <td align="center"><b><font face="Verdana" size="1">Días
de<br>Vigencia</font></b></td>
    <td align="center"><b><font face="Verdana"
size="1">Fecha<br>Cancelación</font></b></td>
    <td align="center"><b><font face="Verdana"
size="1">Fecha<br>Vencimiento</font></b></td>
    <?php if($opcion == 23) { ?><td align="center"><b><font face="Verdana"
size="1">Opción</font></b></td><?php } ?>
  </tr>
  <?php
for ($i = 0; $i < count($oOrdenes); $i++) { //recorro todo las ordenes que se obtuvo
en la busqueda anterior
    $oOrden = $oOrdenes[$i];

    $oPagoArray = $oPagoBo->buscar("%", $oOrden->getId(), $oSimpleDateFormat-
>format($fecha, "%", "1"); //buscamos si esa orden se encuentra cnelada
    $oPagos = java_values($oPagoArray);

    if(count($oPagos) == 1) {
        for ($j = 0; $j < count($oPagos); $j++) { // recorro aquellos pagos perteneciente
a las ordenes del usuario de la session
            $oPago = $oPagos[$j]; //guardo los valores en un arreglo
        }

        $items = $oOrden->getOrdenItems(); //obtengo el detalle de las ordeness
        $valores = java_values($items);

        for ($k = 0; $k < count($valores); $k++) {
            $oOrdenItem = $valores[$k];

            $oProducto = $oProductoBo->seleccionar($oOrdenItem-
>getIdProducto()); //obtengo los datos del producto que pertenece a la orden

            $fechaCad = new Java('java.util.Date');
            $fechaCad = $oPago->getFecha()->clone();

            //Aumenta a la fecha de pago el numero de días de caducidad
            //Se le suma 1 para que sea considerado hasta el ulñtimo dia de activación

```

```

$fechaCad->setDate($fechaCad->getDate() + $oProducto->getAnio() + 1);

if ($fecha->before($fechaCad)) {

    $fechaCad->setDate($fechaCad->getDate() - 1);

    $oTipo = $oTipoBo->seleccionar($oProducto->getIdTipo());
    $oMarca = $oMarcaBo->seleccionar($oProducto->getIdMarca());
    $oModelo = $oModeloBo->seleccionar($oProducto->getIdModelo());

    $limite = $oOrdenItem->getCantidad() * $oProducto->getStock();

    $oSimpleDateFormat2 = new Java("java.text.SimpleDateFormat");
    $oSimpleDateFormat2->applyPattern("yyyy/MM/dd");
?>
<tr>
    <form name="frmProducto" method="POST" action="Index.php?op=22">
        <input type="hidden" name="IdOrdenItem" value="<?php echo $oOrdenItem-
>getId(); ?>">
        <input type="hidden" name="Caduca" value="<?php echo $oSimpleDateFormat2-
>format($fechaCad); ?>">
        <input type="hidden" name="Limite" value="<?php echo $limite; ?>">
        <input type="hidden" name="Modelo" value="<?php echo $oProducto-
>getIdModelo(); ?>">

        <td><font face="Verdana" size="1"><?php echo $oOrden->getId(); ?></font></td>
        <td><font face="Verdana" size="1"><?php echo $oOrdenItem->getCantidad();
?></font></td>
        <td><font face="Verdana" size="1"><?php echo $oTipo->getNombre().',
'. $oModelo->getNombre().'.<br>'. $oProducto->getDescripcion(); ?></font></td>
        <td><font face="Verdana" size="1"><?php echo $limite; ?></font></td>
        <td><font face="Verdana" size="1"><?php echo $oProducto->getAnio();
?></font></td>
        <td><font face="Verdana" size="1"><?php echo $oSimpleDateFormat-
>format($oPago->getFecha()); ?></font></td>
        <td><font face="Verdana" size="1"><?php echo $oSimpleDateFormat-
>format($fechaCad); ?></font></td>
        <?php if($opcion == 23) { ?><td><button name="btnEnviar" type="submit"></button></td><?php } ?>
    </form>
</tr>
<?php
    } //validacion de fechas
    } //for ordenItem
    } //If pago
} // for orden
?>

</table>
<?php
function getDirectorio() {

```

```

    return realpath(dirname($_SERVER["SCRIPT_FILENAME"]))."/WEB-INF/lib";
}

function vigencia() {
    global $oOrden;
    global $oOrdenItem;
    global $oOrdenBo;
    global $oPago;
    global $oPagoBo;
    global $oProducto;
    global $oProductoBo;

    global $oTipo;
    global $oTipoBo;
    global $oMarca;
    global $oMarcaBo;
    global $oModelo;
    global $oModeloBo;

    java_require(getDirectorio()."/publicidad.jar");

    $oOrden = new Java("Be.Orden");
    $oOrdenItem = new Java("Be.OrdenItem");
    $oOrdenBo = new Java("Bo.OrdenBo");
    $oPago = new Java("Be.Pago");
    $oPagoBo = new Java("Bo.PagoBo");
    $oProducto = new Java("Be.Productos");
    $oProductoBo = new Java("Bo.ProductosBo");

    $oTipo = new Java("Be.Tipo");
    $oTipoBo = new Java("Bo.TipoBo");
    $oMarca = new Java("Be.Marca");
    $oMarcaBo = new Java("Bo.MarcaBo");
    $oModelo = new Java("Be.Modelo");
    $oModeloBo = new Java("Bo.ModeloBo");
}

?>
<script src="javaScript/Utilitario.js" type=text/javascript></script>
<script src="javaScript/date-picker.js" type=text/javascript></script>

</td>
</tr>
<tr>
<td width="5%"></td>
</tr>
</table>
</body>

```

Tabla 14
Vigencia.Php

CAPITULO 2

MANUAL DE USUARIO

El usuario al que nos referimos en este capítulo, es al administrador de este portal. A continuación se detallará, con capturas de pantalla y explicación de las mismas cada paso de su funcionalidad. El administrador será aquel que pueda ingresar, modificar y actualizar la información para el buen funcionamiento del portal.

2.1 Autenticación del Administrador

Aquí podemos apreciar la pantalla principal para los diferentes tipos de usuarios y del administrador, el cual tiene que registrarse para poder ingresar a la administración del portal. Una vez que se haya autenticado aparecerá otra pantalla con diferentes opciones para proceder al mantenimiento de la página, ya sea la parte de vendedores de compradores o el mantenimiento y configuración del sistema.



Gráfico 3
Ingreso del sistema

2.2 Navegación del Administrador

El administrador tendrá la opción de ingresar ya sea por compradores, vendedores o por la parte de la administración del sistema. Por parte de vendedores o compradores solo verifica si se están cumpliendo con los procesos correctos del portal, una vez que el halla realizado alguna modificación en la parte administrativa la podrá visualizar en las otras particiones.



Gráfico 4
Navegación del Administrador

Si es coge la opción de administrador (Mantenimiento y configuraciones del sistema) se mostraran algunas de las opciones que el administrador podrá realizar para el ingreso y modificación como persona, identificación, dirección, producto, categoría, ciudad, país provincia, tipo de producto, y tipo de identificación.

<u>Inicio</u>
Opciones
<u>Activación de Pagos</u>
<u>Datos Personales</u>
<u>Direcciones</u>
<u>Identificaciones</u>
<u>Producto</u>
<u>Servicio</u>
Reportes
<u>Preferencia de Clientes</u>
<u>Consulta de Clientes</u>
Configuraciones
<u>Categoría</u>
<u>Ciudad</u>
<u>Marca</u>
<u>Modelo</u>
<u>País</u>
<u>Provincia</u>
<u>Tipo de Producto</u>
<u>Tipo de Identificación</u>

Gráfico 5
Opciones Administrador

2.2.1 Mantenimiento de Persona

Para dar mantenimiento a datos de los usuarios, se escogerá el link Persona, interfaz en la cual se podrá ingresar los datos personales de un usuario, o ya sea el caso de que se desee agregar otro administrador.

Configuración de Persona - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Dirección <http://localhost/CarsMarketing/Administrador.php?op=15> Ir Vínculos »

Opciones

- Inicio
- La empresa**
- Quiénes somos?
- Misión
- Visión
- Objetivos Generales
- Opciones**
- Persona**
- Identificaciones
- Direcciones
- Producto
- Configuraciones
- Categoría

Datos Personales

Id: 0

Nombres:

Apellidos:

Persona Jurídica: ☒

Identificación: C.I.

Usuario:

Contraseña:

Confirmar contraseña:

Directorio

País: Ecuador

Provincia: Azuay

Ciudad: Cuenca

Dirección:

Teléfono:

Email:

Habilitar Usuario

Estado: ☒

Guardar Buscar Cancelar

Listo Intranet local

Gráfico 6
Registro de personas

Si hacemos click en el botón buscar, se mostrará otra pantalla donde esté la información de todas las personas que se han registrado en nuestro portal.

noviembre 03, 2007 [inicio](#) | [recomiéndenos](#) | [contáctenos](#) | [ayuda](#)

CARSMARKETING

Bienvenido CarsMarketing S.A.! [Terminar sesión](#)

Búsqueda de Persona						
Id	Apellidos	Nombres	Tipo	Sesión	Estado	Opción
1	S.A.	CarsMarketing	P. Jurídica	administrador	1	Enviar
2	Pin	Pamela	P. Natural	ppin	1	Enviar
3	Ronquillo	Diana	P. Jurídica	dronquillo	1	Enviar
4	Moreira	Ruth	P. Natural	rmoreira	1	Enviar
5	Averos	Roger	P. Jurídica	s	1	Enviar
6	Lazo	Victor	P. Natural	vlazo	1	Enviar
7	Ibarra	Fanny	P. Natural	fibarra	1	Enviar
8	Luque	Daniel	P. Natural	dluque	1	Enviar
9	garzon	angel	P. Jurídica	angel	1	Enviar
10	Suarez	david	P. Natural	dsuarez	1	Enviar
11	Ronquillo	Diana	P. Natural	qq	1	Enviar
12	Mateus	Aurelio	P. Natural	amateus	1	Enviar
13	Villao	Eladio	P. Jurídica	evillao	1	Enviar

Inicio

La empresa

[Quiénes somos?](#)

[Misión](#)

[Visión](#)

[Objetivos](#)

[Generales](#)

Opciones

[Persona](#)

[Identificaciones](#)

Listo Intranet local

Gráfico 7
Búsqueda de personas

Para seleccionar cualquier registro, se deberá hacer click en el botón enviar, como resultado de esta acción aparecerá una pantalla donde el administrador pueda modificar y por ende guardar los cambios realizados. Estos datos automaticamente se actualizaran en la base de datos.

Configuración de Persona - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Dirección <http://localhost/CarsMarketing/Administrador.php?op=15> Ir Vinculos »

Datos Personales

Id: 1

Nombres: CarsMarketing

Apellidos: S.A.

Persona Jurídica: ☒

Identificación: R.U.C. 0918741745001

Usuario: administrador

Contraseña:

Confirmar contraseña:

Directorio

País: Estados Unidos

Provincia: New York

Ciudad: New York

Dirección: Av. 1

Teléfono: 042659879

Email: info@carsmarketing.com

Habilitar Usuario

Estado: ☒

Guardar Buscar Cancelar

Inicio

La empresa

Quiénes somos?

Misión

Visión

Objetivos Generales

Opciones

Persona

Identificaciones

Direcciones

Producto

Configuraciones

Categoría

Listo Intranet local

Gráfico 8
Actualización de Persona

2.2.2 Mantenimiento de Identificación:

Se ingresara el tipo de identificación que existan ya sea cedula, pasaporte o numero de Ruc.

Configuración de Identificación - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Atrás Búsqueda Favoritos Ir Vínculos »

Dirección <http://localhost/CarsMarketing/Administrador.php?op=17>

noviembre 03, 2007 [inicio](#) | [recomiéndenos](#) | [contáctenos](#) | [ayuda](#)

 **CARSMARKETING**

Bienvenido CarsMarketing S.A.! [Terminar sesión](#)

[Inicio](#)

La empresa

[Quiénes somos?](#)

[Misión](#)

[Visión](#)

[Objetivos Generales](#)

Opciones

[Persona](#)

[Identificaciones](#)

Identificación

Id: 0

Tipo de Identificación: C.I.

Código:

Estado: ☒

[Guardar](#) [Buscar](#) [Cancelar](#)

Listo Intranet local

Gráfico 9
Ingreso de Identificación

Igualmente se presiona el botón buscar se dispararan todos lo registros que existan, luego se da click en el I botón enviar para poder modificar y guardar la información correcta.

Configuración de Identificación - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Atrás Búsqueda Favoritos Ir Vínculos »

Dirección http://localhost/CarsMarketing/Administrador.php?op=17

noviembre 03, 2007 inicio | recomiéndenos | contáctenos | ayuda

CARSMARKETING

Bienvenido CarsMarketing S.A. Terminar sesión

Inicio

La empresa

Quiénes somos?

Misión

Visión

Objetivos Generales

Opciones

Persona

Identificaciones

Identificación

Id: 1

Tipo de Identificación: Pasaporte

Código: 123456789001

Estado: ☒

Guardar Buscar Cancelar

Intranet local

Gráfico 10
Modificación de Identificación

2.2.3 Mantenimiento de Dirección

Se ingresan o se modifica los países con sus respectivas provincias, dirección, teléfono, email etc. De igual manera para poder modificar los registros se escogen la opción “Buscar” y devuelve una ventana donde podemos modificar y actualizar los datos.

Configuración de Directorio - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Dirección <http://localhost/CarsMarketing/Administrador.php?op=19> Ir Vínculos »

noviembre 03, 2007 [inicio](#) | [recomiéndenos](#) | [contáctenos](#) | [ayuda](#)

 **CARSMARKETING**

Bienvenido CarsMarketing S.A.! [Terminar sesión](#)

Directorio

Id:

País:

Provincia:

Ciudad:

Dirección:

Teléfono:

Email:

Estado: ☒

Gráfico 11
Ingreso de Direcciones

2.2.4 Mantenimiento de Productos:

Aquí se mostraran los diferentes productos que se ofrecerá ya sea vehículos, repuestos, y paquetes. El administrador podrá modificar información de cualquiera de estas opciones.



Gráfico 12
Tipos de productos

Si presionamos vehículos o repuestos nos aparecerá una ventana con opciones de vehículos o repuestos para vehículos dependiendo de lo escogido anteriormente.



Gráfico 13
Elección de vehículos o repuestos

Una vez que el administrador escoja cualquiera de las diferentes opciones se desplegará una ventana con la búsqueda correspondiente para que pueda hacer la correspondiente modificación.

Criterios de búsqueda...

Categoría	Proveedor	Tipo	Marca	Modelo	Año	Descripción
Todo	Todo	Todo	Todo	Todo		

PRODUCTO						
Id	Categoría	Proveedor	Tipo	Marca	Modelo	Año
0	Vehículo	CarsMarketing S.A.	Camioneta	Toyota s.a.	Stoud	0
1	Repuesto	CarsMarketing S.A.	Auto	skoda	Luv	2006
2	Paquete	CarsMarketing S.A.	Camioneta	skoda	Stoud	2007
3	Paquete	CarsMarketing S.A.	Dinamico	CarsMarketing	Variable	2007
4	Repuesto	Pamela Pin	Auto	Toyota s.a.	Luv	2003
5	Vehículo	CarsMarketing S.A.	Camion	Toyota s.a.	Stoud	2005
6	Vehículo	CarsMarketing S.A.	Camioneta	Toyota s.a.	Stoud	0
7	Vehículo	CarsMarketing S.A.	Camion	Toyota s.a.	Stoud	2000
8	Vehículo	CarsMarketing S.A.	Camioneta	Toyota s.a.	Stoud	1
9	Vehículo	CarsMarketing S.A.	Camioneta	Toyota s.a.	Stoud	1
10	Vehículo	Pamela Pin	Auto	skoda	Luv	2003

Gráfico 14
Búsqueda de vehículos

Por otro lado al presionar paquetes se mostraran los diferentes paquetes donde el administrador podrá ingresar uno nuevo o modificar características de uno anterior.



Gráfico 15
Tipos de paquetes

2.2.5 Mantenimiento de Categoría:

También contaremos con algunas ventanas donde el administrador podrá ingresar o modificar las diferentes Categorías ya sea para aumentar un producto o servicio que nuestro portal ofrezca.

INGRESO

Categoría

Id: 0

Nombre:

Descripción:

Estado: ☒

Guardar

Buscar

Cancelar

BUSQUEDA

Búsqueda de Categoría				
Id	Nombre	Descripción	Estado	Opción
1	Vehiculo	Ninguna	1	Enviar
2	Repuesto	Repuesto s.a.	1	Enviar
3	Paquete	Publicidad	1	Enviar

MODIFICAR Y ACTUALIZAR

Categoría

Id: 2

Nombre: Repuesto

Descripción: Repuesto s.a.

Estado: ☒

Guardar

Buscar

Cancelar

Gráfico 16
Ingreso, búsqueda y modificación de categoría

2.2.6 Mantenimiento de País:

Existirá también interfaces donde pueda ingresar, modificar, buscar y actualizar Países.

The image displays two screenshots of a web application interface for managing countries. Both screenshots feature a light beige background and a title bar labeled 'País'.

The top screenshot, titled 'Ingreso de país', shows a form for entering a new country. It includes a text input for 'Id' with the value '0', a text input for 'Nombre' which is empty, and a dropdown menu for 'Estado' with a downward arrow. Below these fields are three buttons: 'Guardar', 'Buscar', and 'Cancelar'.

The bottom screenshot, titled 'Busqueda y actualizacion del país', shows a form for searching and updating an existing country. It includes a text input for 'Id' with the value '1', a text input for 'Nombre' with the value 'Ecuador', and a dropdown menu for 'Estado' with a downward arrow. Below these fields are three buttons: 'Guardar', 'Buscar', and 'Cancelar'.

Gráfico 17
Ingreso, búsqueda y modificación de país

2.2.7 Mantenimiento de Marca, Modelo, Provincia, Tipo de Producto y Tipo de Identificación:

De igual manera se tendrá opciones para ingresar, buscar, modificar y guardar registros de MARCA, MODELO, PROVINCIA, TIPO DE PRODUCTO Y TIPO DE IDENTIFICACION.

The image displays five separate forms stacked vertically, each for a different entity: Marca, Modelo, Provincia, Tipo de Producto, and Tipo de Identificación. Each form has a title bar with the entity name. The fields and values are as follows:

Entity	Id	Nombre	Descripción	Estado
Marca	3	Toyota s.a.	Toyota	☒
Modelo	3	Stoud	Stoud	☒
Provincia	4	Canar		☒
Tipo de Producto	3	Auto	Auto	☒
Tipo de Identificación	0			☒

Each form also includes three buttons at the bottom: Guardar, Buscar, and Cancelar.

Gráfico 18
Ingreso, búsqueda y modificación de Marca Modelo, Provincia, Tipo de producto, Tipo de identificación

2.3 Activación de pagos:

En esta sección el administrador deberá cumplir con la tarea de revisar y estar atento para la activación de pagos de los clientes que hayan adquirido paquetes para la promoción de sus productos.

The screenshot shows a web browser window with the URL `http://127.0.0.1/carsmarketing/Administrador.php?op=31`. The page header includes the date "diciembre 09, 2007" and navigation links: "inicio | servicios | contáctenos | recomiéndanos". The CarsMarketing logo is prominently displayed, along with a welcome message "Bienvenido CarsMarketing S.A.!" and a "Terminar sesión" link.

A left sidebar contains a menu with the following items: "Inicio", "Opciones", "Activación de Pagos", "Datos Personales", "Direcciones", "Identificaciones", "Producto", "Servicio", "Reportes", "Preferencia de Clientes", and "Consulta de Clientes".

The main content area features a search section titled "Criterios de búsqueda..." with fields for "No. Orden", "Fecha Desde", "Fecha Hasta", and "Persona" (set to "Todo"). "Buscar" and "Cancelar" buttons are present.

Below the search section is a table titled "Activación de Pagos".

No. Orden	Fecha	Identificación	Cliente	Valor	Observación	Opción
10	2007-12-09	C.I.: 1309579033	MARTHA MARIA DE LA TORRE QUIROGA	10		☒

Gráfico 19
Activación de paquetes

2.4 Servicios:

Esta interfaz nos permitirá crear algún servicio adicional para los usuarios del portal, estos servicios solo se podrán visualizar si el usuario ha adquirido algún paquete para promocionar su producto.

Configuración de Servicio - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Atrás Búsqueda Favoritos

Dirección <http://127.0.0.1/carsmarketing/Administrador.php?op=32>

diciembre 09, 2007 [inicio](#) | [servicios](#) | [contáctenos](#) |

CARSMARKETING

Bienvenido CarsMarketing S.A.! [Terminar sesión](#)

Inicio

Opciones

[Activación de Pagos](#)

[Datos Personales](#)

[Direcciones](#)

[Identificaciones](#)

[Producto](#)

[Servicio](#)

Reportes

[Preferencia de Clientes](#)

[Consulta de Clientes](#)

Servicio

Id: 0

Nombre:

Descripción:

Estado: ☒

[Guardar](#) [Buscar](#) [Cancelar](#)

Gráfico 20
Ingreso de servicios

2.5 Reportes:

En esta sección el administrador podrá visualizar dos clases de reportes PREFERENCIAS DE CLIENTES y CONSULTAS DE CLIENTES esto se mostrara en un archivo PDF para que no se lo pueda modificar esto le servirá al administrador para tener un estimativo de cuantas personas visitan nuestro portal y cual es el producto de mayor demanda.

2.5.1. Preferencias de Clientes:

The screenshot displays the CARSMARKETING web application interface. The top header shows the date 'diciembre 09, 2007' and navigation links: 'inicio | servicios | contáctenos'. The main header features the CARSMARKETING logo and a welcome message: 'Bienvenido CarsMarketing S.A.!' with a 'Terminar sesión' link.

The left sidebar contains a menu with the following sections:

- Inicio**
- Opciones**
 - [Activación de Pagos](#)
 - [Datos Personales](#)
 - [Direcciones](#)
 - [Identificaciones](#)
 - [Producto](#)
 - [Servicio](#)
- Reportes**
 - [Preferencia de Clientes](#)
 - [Consulta de Clientes](#)
- Configuraciones**
 - [Categoría](#)
 - [Ciudad](#)

The main content area displays the 'Reporte: Preferencia de Clientes' form. It includes the following fields and controls:

- Fecha Inicio:** 2007-11-01
- Fecha Fin:** (empty)
- Categoría:** Todo (dropdown menu)
- Tipo:** Todo (dropdown menu)
- Marca:** Todo (dropdown menu)
- Modelo:** Todo (dropdown menu)
- Generar Reporte** (button)

Overlaid on the bottom right is a calendar window titled 'http://127.0.0.1 - Cale...' showing 'Noviembre 2007'. The calendar includes navigation buttons: '<<', '<', '[Print]', '>', and '>>'. The days of the week are listed as Dom, Lun, Mar, Mie, Jue, Vie, Sab. The dates are displayed in a grid, with the 1st of the month highlighted in green.

Gráfico 21
Preferencias de clientes

2.5.2. Consultas de Clientes:

El administrador tendrá que ingresar la fecha de inicio y la del final para que pueda visualizar el respectivo reporte.

The screenshot displays the CarsMarketing web application interface. At the top, the date 'diciembre 09, 2007' is shown on the left, and navigation links 'inicio | servicios | contáctenos' are on the right. The CarsMarketing logo is prominently displayed. Below the logo, a welcome message 'Bienvenido CarsMarketing S.A.!' and a 'Terminar sesión' link are visible.

The left sidebar contains a menu with the following items: 'Inicio', 'Opciones' (with sub-items: 'Activación de Pagos', 'Datos Personales', 'Direcciones', 'Identificaciones', 'Producto', 'Servicio'), 'Reportes' (with sub-items: 'Preferencia de Clientes', 'Consulta de Clientes'), and 'Configuraciones' (with sub-items: 'Categoría', 'Ciudad').

The main content area is titled 'Reporte: Consulta de Clientes'. It features a form with the following fields: 'Fecha Inicio:', 'Fecha Fin:', 'Categoría:' (dropdown menu), 'Tipo:' (dropdown menu), 'Marca:' (dropdown menu), and 'Modelo:' (dropdown menu). A 'Generar Reporte' button is located at the bottom of the form.

Overlaid on the bottom right of the application is a calendar window titled 'Diciembre 2007'. The calendar includes navigation buttons: '<<', '<', '[Print]', '>', and '>>'. The days of the week are listed as 'Dom', 'Lun', 'Mar', 'Mie', 'Jue', 'Vie', and 'Sab'. The dates are displayed in a grid, with the 9th of the month highlighted in red. The calendar is part of a browser window titled 'http://127.0.0.1 - Cale...'.

Gráfico 22
Consultas de clientes