



UNIVERSIDAD TÉCNICA DEL NORTE

**FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES**

**TRABAJO DE GRADO, PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN SISTEMAS COMPUTACIONALES**

TEMA:

**“ESTUDIO DE LA TECNOLOGÍA CODEIGNITER APLICADA AL
DESARROLLO DE PORTALES WEB CON UNA ARQUITECTURA MVC”**

AUTOR: WILSON JAVIER CEVALLOS AVILÉS.

DIRECTOR: ING. MARCO PUSDÁ.

IBARRA – ECUADOR

2014



UNIVERSIDAD TÉCNICA DEL NORTE BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

La UNIVERSIDAD TÉCNICA DEL NORTE dentro del proyecto Repositorio Digital Institucional, determina la necesidad de disponer de textos completos en formato digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la Universidad.

Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto, para lo cual pongo a disposición la siguiente información.

DATOS DE CONTACTO	
CÉDULA DE IDENTIDAD:	100247738-6
APELLIDOS Y NOMBRES:	CEVALLOS AVILÉS WILSON JAVIER
DIRECCIÓN:	ISLA FERNANDINA 17-48 Y ZUMBA
EMAIL:	wilson_cevallos2@yahoo.es
TELEFONO FIJO:	062652610
TELÉFONO MÓVIL:	0990554781

DATOS DE LA OBRA	
TÍTULO:	“ESTUDIO DE LA TECNOLOGÍA CODEIGNITER APLICADA AL DESARROLLO DE PORTALES WEB CON UNA ARQUITECTURA MVC.”
AUTOR:	CEVALLOS AVILÉS WILSON JAVIER
FECHA:	19 DE MAYO DEL 2014
PROGRAMA:	PREGRADO
TÍTULO POR EL QUE OPTA:	INGENIERÍA EN SISTEMAS COMPUTACIONALES
DIRECTOR:	ING. MARCO PUSDÁ

2. AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, WILSON JAVIER CEVALLOS AVILÉS, con cédula de identidad Nro. 100247738-6 en calidad de autor y titular de los derechos patrimoniales de la obra o trabajo de grado descrito anteriormente, hago entrega del ejemplar respectivo en forma digital y autorizo a la Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y uso del archivo digital en la biblioteca de la Universidad con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación, investigación y extensión, en concordancia con la Ley de Educación Superior Artículo 144.



(Firma):

Nombre: Wilson Javier Cevallos Avilés

Cédula: 100247738-6

Ibarra, a los 19 días del mes de mayo del 2014.



UNIVERSIDAD TÉCNICA DEL NORTE

CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE GRADO A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

Yo, WILSON JAVIER CEVALLOS AVILÉS, con cédula de identidad N° 1002477386, manifiesto mi voluntad de ceder a la Universidad Técnica del Norte los derechos patrimoniales consagrados en la Ley de Propiedad Intelectual del Ecuador, artículos 4, 5 y 6, en calidad de autor del trabajo de grado denominado: "ESTUDIO DE LA TECNOLOGÍA CODEIGNITER APLICADA AL DESARROLLO DE PORTALES WEB CON UNA ARQUITECTURA MVC", que ha sido desarrollado para optar por el título de: INGENIERÍA DE SISTEMAS COMPUTACIONALES, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En mi condición de autor me reservo los derechos morales de la obra antes citada, aclarando que el trabajo aquí descrito es de mi autoría y que no ha sido previamente presentado para ningún grado o calificación profesional.

En concordancia suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la biblioteca de la Universidad Técnica del Norte.

(Firma):

Nombre: Wilson Javier Cevallos Avilés

Cédula: 100247738-6

Ibarra, a los 19 días del mes de mayo del 2014.

3.- CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros



(Firma):

Nombre: Wilson Javier Cevallos Avilés

Cédula: 100247738-6

Ibarra, a los 19 días del mes de mayo del 2014.

Ibarra, 27 de noviembre del 2013

CERTIFICACIÓN

Certifico que la Tesis previa a la obtención del título de Ingeniería en Sistemas Computacionales con el tema **“Estudio de la Tecnología Codeigniter Aplicada al Desarrollo de Portales Web con una Arquitectura MVC”** y con el aplicativo **“Implementación del Portal Web de Información de Precios y Stock para la empresa Megasystem”** ha sido desarrollada y terminada en su totalidad por el Sr. Wilson Javier Cevallos Avilés con C.C. 100247738-6 bajo mi supervisión para lo cual firmo en constancia.

Atentamente,



Ing. Marco Pusdá
DIRECTOR DE TESIS

Ibarra, 23 de noviembre del 2013

CERTIFICACIÓN

Señores
UNIVERSIDAD TÉCNICA DEL NORTE
Presente.

De mis consideraciones.-

Siendo auspiciantes del proyecto de tesis del Egresado WILSON JAVIER CEVALLOS AVILÉS con CI: 100247738-6 quien desarrolló su trabajo con el tema "**Estudio de la Tecnología Codeigniter Aplicada al Desarrollo de Portales Web con una Arquitectura MVC**" con el aplicativo "**Implementación del Portal Web de Información de Precios y Stock para la Empresa Megasystem**", me es grato informar que se han superado con satisfacción las pruebas técnicas y la revisión de cumplimiento de los requerimientos funcionales, por lo que se recibe el proyecto como culminado y realizado por parte del egresado WILSON JAVIER CEVALLOS AVILÉS. Una vez que hemos recibido la capacitación y documentación respectiva, nos comprometemos a continuar utilizando el mencionado aplicativo en beneficio de nuestra empresa.

El egresado WILSON JAVIER CEVALLOS AVILÉS puede hacer uso de este documento para los fines pertinentes en la Universidad Técnica del Norte.

Atentamente,



Sr. Lening Morán
GERENTE MEGASYSTEM

Dedicatoria

Este trabajo está dedicado a mis padres Félix Cevallos y María Avilés, quienes me apoyaron incansablemente para ver realizado el objetivo de culminar mis estudios.

A mis hermanos por su apoyo a lo largo de todo este tiempo que duró mis estudios.

A mi novia Anita Ipiales que es la persona que me infunde valor cada día para alcanzar mis sueños.

Agradecimiento

Agradezco a Dios por la vida, por todas las bendiciones que me ha otorgado y por su amor infinito.

Agradezco a la Universidad Técnica del Norte que me abrió sus puertas para realizar mis estudios superiores.

Agradezco a mi director de Tesis el Ing. Marco Pusdá por su valiosa ayuda en la realización de este trabajo.

Agradezco a todos aquellos que fueron mis profesores por haber compartido sus conocimientos conmigo.

Al Señor Lening Morán gerente de MEGASYSTEM por el apoyo prestado en el desarrollo de este proyecto.

ÍNDICE DE CONTENIDOS

Contenido	
RESUMEN	1
SUMMARY	2
INTRODUCCIÓN	3
CAPÍTULO I	4
1. Antecedentes del Proyecto	5
1.1. Información General de la Empresa.....	5
1.2. Problema del Proyecto	10
1.3. Objetivos del Proyecto	11
1.4. Justificación del Proyecto	12
1.5. Alcance del Proyecto.....	12
CAPÍTULO II	16
2. Estudio del Framework CodeIgniter y la Arquitectura MVC (Modelo Vista Controlador)	17
2.1. ¿Qué es CodeIgniter?	17
2.2. Estudio de CodeIgniter.....	22
2.3. Herramientas de Desarrollo de CodeIgniter.....	51
CAPÍTULO III	60
3. Análisis, Diseño e Implementación del Portal Web de información para la Empresa Megasystem.....	61
3.1 DOCUMENTO DE VISIÓN	61
3.2. ESPECIFICACIÓN DE REQUISITOS	68
3.3. DISEÑO	84
3.4 DESARROLLO E IMPLEMENTACIÓN	88
3.5. PRODUCTO	89
CAPÍTULO IV	92
4. Portal Web de Información de Precios y Stock	93
4.1 Diseño de la base de datos	93
4.2 Manual de Usuario	94
4.3. Manual Técnico.....	109

CAPÍTULO V	121
5. Validación y Pruebas de Funcionamiento.....	122
5.1 Prueba de Módulo Clientes	122
5.2 Prueba de Módulo Productos	122
5.3 Prueba de Módulo Información	123
5.4. Prueba de Módulo Servicios	123
5.5 Prueba de Módulo Reportes	124
5.6 Prueba de Módulo Seguridad.....	124
CAPÍTULO VI	126
6.1 Análisis de Impactos	127
6.2 Conclusiones	127
6.3 Recomendaciones.....	128
GLOSARIO	130
BIBLIOGRAFÍA	132
Bibliografía	133
ANEXOS	136
ANTEPROYECTO DE TESIS	137
ANEXO B	152
LISTA DE HELPERS DE CODEIGNITER.....	152
ANEXO C	155
LISTA DE LIBRERÍAS DE CODEIGNITER	155

ÍNDICE DE FIGURAS

Figura 1: Organigrama de Megasystem	9
Figura 2: Módulos del Portal web de la Empresa Megasystem	13
Figura 3: Arquitectura del Portal web de la Empresa Megasystem	15
Figura 4: Diagrama de flujo de una aplicación Codeigniter	22
Figura 5: Diagrama de Arquitectura MVC	24
Figura 6: Estructura de Codeigniter	27
Figura 7: Pantalla del IDE Netbeans	52
Figura 8: Pantalla del IDE CodeLobster PHP Edition	54
Figura 9: Hoja de Trabajo	69
Figura 10: Diagrama de Casos de Uso del Portal	71
Figura 11 : Diagrama de Casos de Uso del Administrador.....	72
Figura 12: Diagrama de Casos de Uso del Usuario Registrado	73
Figura 13: Diagrama de Casos de Uso del Invitado.....	74
Figura 14: Diagrama de Clases del Portal Web	75
Figura 15: Diagrama de Secuencias del Portal Web	76
Figura 16: Diagrama de Secuencia del Administrador	77
Figura 17: Diagrama de Secuencia del Usuario Registrado	78
Figura 18: Diagrama de Secuencia del Invitado no Registrado.....	79
Figura 19: Diagrama de estados del Administrador.....	80
Figura 20: Diagrama de estados del Usuario Registrado	80
Figura 21: Diagrama de estados del Invitado no registrado.....	81
Figura 22: Diagrama de Colaboración del Administrador.....	81
Figura 23: Diagrama de Estados del Usuario Registrado	82
Figura 24: Diagrama de Estados del Invitado no Registrado.....	82
Figura 25: Diagrama de Distribución del Portal	83
Figura 26: Vista Principal (Frontend) del Invitado	90
Figura 27: Vista Principal (Frontend) del Usuario Registrado	90
Figura 28: Vista Principal (Backend) del Administrador	91
Figura 29: Diseño de la Base de Datos	93

Figura 30: Ingreso como administrador	94
Figura 31: Página de administrador	95
Figura 32: Página Información de Empresa.....	95
Figura 33: Página administración de Instalaciones.....	96
Figura 34: Página administración de Asesores	96
Figura 35: Página administración de Noticias	97
Figura 36: Página administración de Líneas de productos.....	97
Figura 37: Página administración de Marcas	98
Figura 38: Página administración de Productos.....	98
Figura 39: Página administración de Precios	99
Figura 40: Página administración de Servicios.....	99
Figura 41: Página administración de Contactos Invitados.....	100
Figura 42: Página administración de Usuarios Registrados.....	101
Figura 43: Página administración de Tipo de Clientes	101
Figura 44: Página administración de Tipo de Documento.....	102
Figura 45: Página administración de Documento Sugerencias.....	103
Figura 46: Página administración de Encuestas.....	103
Figura 47: Página administración de Respuestas de Encuestas	104
Figura 48: Cuadro de Encuestas.....	104
Figura 49: Gráfico estadístico de encuesta	105
Figura 50: Página administración de Reportes.....	105
Figura 51: Página Frontend de Usuarios no Registrados.....	106
Figura 52: Formulario de ingreso a módulo usuarios registrados.....	107
Figura 53: Página Frontend de Usuarios Registrados	108
Figura 54: Ejecuta Instalador Easyphp	109
Figura 55: Pantalla de Bienvenida Easyphp.....	109
Figura 56: Acuerdo de Licencia Easyphp	110
Figura 57: Ruta de instalación	110
Figura 58: Selecciona Carpeta	111
Figura 59: Pantalla de Instalación	111
Figura 60: Fin de Instalación.....	112
Figura 61: Instalación del Proyecto.....	112

Figura 62: Crear Base de Datos	113
Figura 63: Configurar Base de Datos.....	114
Figura 64: Página de inicio de proyecto.....	114

ÍNDICE DE TABLAS

Tabla 1: Análisis del Problema	62
Tabla 2: Declaración del posicionamiento del proyecto.....	64
Tabla 3: Roles y Responsabilidades durante el desarrollo del Proyecto.....	65
Tabla 4: Resumen de Usuarios del Portal	66
Tabla 5: Costes del proyecto	67
Tabla 6: Esquema del menú principal del Usuario Administrador	84
Tabla 7: Esquema del menú principal del Usuario invitado	86
Tabla 8: Esquema del menú principal del Usuario registrado	87

RESUMEN

El presente Trabajo de grado tiene como objetivo estudiar la tecnología CodeIgniter aplicada al desarrollo de portales web con patrón de arquitectura MVC (Modelo-Vista-Controlador).

Los capítulos que componen la presente Trabajo de grado están descritos a continuación.

El capítulo I hace una introducción de los antecedentes del proyecto, en donde se verá la información general de la empresa “MEGASYSTEM” en la que se implementa el aplicativo, el problema que generó el estudio de esta herramienta, los objetivos que se plantearon en la realización del estudio, la justificación y los alcances de dicho proyecto. En el Capítulo II se realiza una fundamentación teórica sobre la herramienta CodeIgniter, se hace una introducción y se revisan las principales características de la herramienta, posteriormente un estudio de las principales funcionalidades del framework y de la arquitectura MVC en la que está construida, las herramientas de desarrollo que se acoplan con el framework y un estudio de la integración del gestor de base de datos MySQL a CodeIgniter.

El Capítulo III corresponde al desarrollo e implementación de la aplicación Portal web de la empresa “MEGASYSTEM”, se realiza el documento de visión, el análisis de requerimientos, el diseño de la aplicación en donde se hace uso de herramientas como: flujos de trabajo, casos de uso, diagramas de clases, diagramas de secuencias, diagramas de colaboración, diagrama de estados, muestra del diseño, se registra el desarrollo y la implementación del portal, y presenta las características principales de la aplicación. El Capítulo IV presenta el aplicativo terminado, el diseño de la base de datos y los documentos técnicos así como los manuales de usuario.

El Capítulo V corresponde a las anotaciones sobre las pruebas de funcionamiento realizadas a los diferentes módulos del portal. El Capítulo VI se refiere a las Conclusiones y Recomendaciones de esta investigación.

SUMMARY

The current dissertation has as objective to study the CodeIgniter technology applied to the development of web portals with architecture pattern MVC (Model-View-Controller).

The chapters that the current dissertation contains are described as follows.

Chapter I makes an introduction of the project's background, where the general information of the company MEGASYSTEM will be seen in which the applicative is implemented, the problem that merged the study of this tool, the objectives that were settle down when making this study, the justification and the scopes of this project. In Chapter II it is carried out a theoretical establishment about the CodeIgniter tool, an introduction is made and the main characteristics of the tool are revised, then a study of the main functionalities of the framework and the MVC architecture in which it is built, the development tools that are coupled with the framework and a study of the integration of the agent's database MySQL to CodeIgniter.

Chapter III corresponds to the development and implementation of the Web Portal application of the company MEGASYSTEM, the document of vision is carried out, the analysis of requirements, the design of the application where makes use of tools like: work flows, use cases, class diagrams, sequence diagrams, collaboration diagrams, states diagram, show of the design, it registers the development and the implementation of the portal, and it presents the main characteristics of the application. Chapter IV presents the finished applicative, the design of the database and the technical documents as well as the user's manuals.

Chapter V corresponds to the annotations about the operation tests made to the different portal modules. Chapter VI refers to the Conclusions and Recommendations of this research.

INTRODUCCIÓN

El objetivo principal del proyecto es realizar un estudio de la tecnología Codeigniter y su integración al gestor de base de datos Mysql. El segundo objetivo es construir un Portal Web aplicando la tecnología Codeigniter y base de datos Mysql para la empresa MEGASYSTEM que brinde la información necesaria de productos para sus clientes.

Para realizar esta aplicación se utilizó el lenguaje de programación PHP con el framework Codeigniter y la base de datos Mysql, para la programación se utilizó la herramienta de desarrollo Netbeans.

Para el desarrollo de la metodología se usó RUP (Rational Unified Process) con los artefactos que se presentan a continuación:

- Documento de Visión
- Flujo de Trabajo
- Casos de Uso
- Diagramas de Clases
- Diagramas de Secuencia
- Diagramas de Estado
- Diagramas de Colaboración
- Diagramas de Distribución

CAPÍTULO I

Antecedentes del Proyecto



Antecedentes del Proyecto

- 1.1. Información General de la empresa
- 1.2. Problema del Proyecto
- 1.3. Objetivos del Proyecto
- 1.4. Justificación del Proyecto
- 1.5. Alcance del Proyecto

1. Antecedentes del Proyecto

1.1. Información General de la Empresa

1.1.1. Datos Generales

Nombre de la empresa: MEGASYSTEM

Propietario: SR. Léning Morán

Ruc: 0401168513001

Dirección: calles Víctor Gómez Jurado 3-44 y DR. Luis Fernando Aguinaga

Teléfono: 062630894

Ciudad: Ibarra-Ecuador

Actividades económicas:

- Distribución al por mayor y menor de computadoras incluso piezas y partes.
- Venta al por menor de materiales y equipos de oficina.
- Mantenimiento y reparación de maquinaria de informática

1.1.2. Reseña Histórica de la Empresa

La empresa MEGASYSTEM fue creada en el año 2004, sus propietarios son el señor Rilmer Léning Morán Coral y su esposa la señora Mayra Marcela Muñoz Villarreal. La empresa empezó como un pequeño local arrendado ubicado en la ciudad de Ibarra en las calles José Ignacio Canelos 3-53 y Av. Mariano Acosta.

En sus orígenes era un local pequeño en donde se brindaban los servicios de digitación, utilizando los recursos obtenidos mediante un crédito bancario se adquirieron suministros e insumos de computación como cds, tinta de impresora, cartuchos, entre otros para dedicarlos a la venta en el local. Se empezó a recorrer por los locales que distribuyen material informático en el centro comercial de la ciudad para ofrecer sus productos, dándose a conocer como un pequeño distribuidor de suministros.

Con la ayuda de un técnico informático se empezó a ensamblar equipos de computación, entonces dio inicio la venta de computadores y suministros tecnológicos a pequeña escala, para lo cual se empezó a buscar distribuidores de piezas y partes de la línea tecnológica e informática.

Así surge la idea de crear una empresa que se dedique a la distribución de piezas, partes y suministros de computación en la ciudad de Ibarra y que abarque a la región norte del país en donde existe una gran cantidad de pequeños locales de venta de computadores.

Con el tiempo la empresa Megasystem se trasladó a las calles Víctor Gómez Jurado 3-44 y Luis Fernando Aguinaga, en donde funciona hasta estos momentos, ocupa toda la planta baja del edificio en un área aproximada de 110 metros cuadrados, en donde se distribuyen los diferentes departamentos.

La actividad principal de Megasystem es la venta de equipos de computación, accesorios y suministros, servicios de mantenimiento técnico preventivo y correctivo. Entre los productos que se ofrecen al mercado están: cases, mother board, procesadores, discos duros, memorias ram, monitores, teclados, mouses, tarjetas de video, tarjetas de sonido, computadores portátiles, cartuchos de impresora entre otros.

1.1.3. Misión de la Empresa Megasystem

“Comercializar partes y piezas de equipos de computación, para satisfacer las necesidades del mercado regional, brindando productos de alta calidad con garantía y asesoría oportuna y efectiva”.

1.1.4. Visión de la Empresa Megasystem

“Ser líder en la comercialización de piezas y partes de equipos de computación en la zona norte del país mejorando día a día la calidad y productividad de las personas que trabajan en la organización para satisfacer de manera eficiente y oportuna las necesidades de sus clientes”.

1.1.5. Valores de la Empresa Megasystem

A) Con los Clientes

“La satisfacción de los requerimientos y las expectativas de los clientes son fundamentales para la supervivencia y prosperidad de la empresa, por tanto la organización continuará estableciendo relaciones de largo plazo con los clientes, basadas en la calidad, el buen desempeño, la seriedad, la honorabilidad y el espíritu de servicio”.

B) Con los Colaboradores

“Las relaciones con los colaboradores están basadas en un permanente respeto, atención por la persona y la mutua confianza. La empresa está comprometida en mantener un ambiente donde prime la salud, seguridad, bienestar y estímulo para el desarrollo de todos sus colaboradores”.

C) Con los Proveedores

“Se considera a los proveedores como socios en la satisfacción de los requerimientos de los clientes. Las decisiones de compra estarán basadas en criterios de confiabilidad, eficiencia, calidad, servicio y costo, buscando

relaciones equitativas de mutuo beneficio a largo plazo. Los pagos a proveedores son prioritariamente cancelados a su debido tiempo”.

D) Con la Competencia

“La empresa cree en la conveniencia social de una competencia sana. Consecuentemente obrará con lealtad, transparencia y buena fe en sus relaciones comerciales”.

E) Con el Estado

“La empresa respeta las instituciones y autoridades reconocidas en el país y en donde tenga sus demás actividades y aplicará correctamente sus leyes y normas”.

F) Con la Comunidad

“La empresa es consciente de la necesidad de contribuir con el desarrollo social, y económico de la ciudad de Ibarra para lo cual brinda oportunidades de trabajo para los habitantes de las región”.

G) La Posición Ética

“La empresa actuará siempre sobre una base de alta moral, integridad, seriedad y buena fe. La empresa no hará transacciones comerciales de compra o venta de bienes de ninguna índole, ni celebrará contratos de asesoría o prestación de servicios con personas naturales o jurídicas, de quienes de alguna forma se tenga indicios o se presuma estén involucrados en actividades ilícitas o que tengan un dudoso comportamiento social o comercial”.

1.1.6. Estructura Organizacional de Megasystem

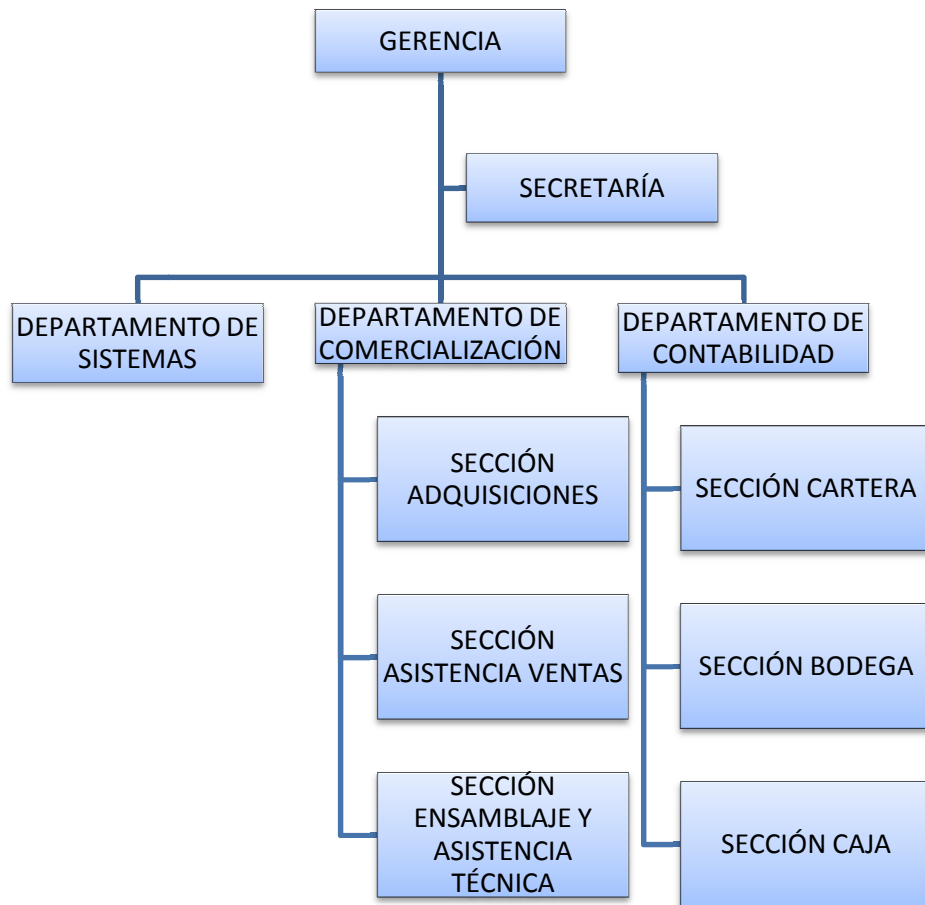


Figura 1: Organigrama de Megasystem

Fuente: Propia

A) GERENCIA.- toma decisiones concernientes a la empresa en general, se preocupa por el buen desarrollo de las actividades de todos los departamentos de la empresa.

B) SECRETARÍA.- se encarga de organizar la agenda de gerencia.

C) EL DEPARTAMENTO DE SISTEMAS.- se encarga del desarrollo del nuevo software contable de la empresa, la administración de la red de datos y del manejo del servidor.

D) EL DEPARTAMENTO DE COMERCIALIZACIÓN.- es el que se dedica a todo el proceso comercial, desde adquisición de mercadería hasta su venta al cliente, está dividido de la siguiente manera:

- **SECCIÓN DE ADQUISICIONES.-** encargado de buscar proveedores, escoger los mejores precios y la compra de la mercadería, así como también se encarga de la coordinación del transporte.
- **SECCIÓN VENTAS.-** conformado por vendedores que tienen a su cargo los clientes activos, su labor es atender al cliente, brindarle asesoramiento adecuado sobre los productos, cerrar las ventas y darle al cliente un servicio post venta.
- **SECCIÓN ENSAMBLAJE Y ASISTENCIA TÉCNICA.-** se encarga del ensamblaje de los equipos de cómputo y brinda el servicio de mantenimiento preventivo y correctivo de equipos.

E) EL DEPARTAMENTO DE CONTABILIDAD.- es el que se encarga de manejar las diversas cuentas que posee la empresa así como de realizar los análisis financieros, tiene las siguientes secciones:

- **SECCIÓN CARTERA.-** encargada de recuperación de la cartera vencida, análisis y autorización del crédito a los clientes.
- **SECCIÓN BODEGA.-** en donde se almacenan todos los productos, se realiza la recepción y despacho de los mismos.
- **SECCIÓN CAJA.-** en donde se realiza la cancelación de las facturas emitidas por los vendedores.

1.2. Problema del Proyecto

Las personas que se dedican al desarrollo de software y en especial de portales y páginas web, por lo general utilizan tecnologías como php, jsp o .net para el desarrollo de sus aplicaciones, también suelen utilizar

frameworks de desarrollo como Symfony, Zendframework, Ruby on Rails o gestores de contenido como Drupal, Joomla entre otros.

Los frameworks que se utilizan con regularidad para el desarrollo rápido de aplicaciones web, tienen características muy potentes, sin embargo poseen una curva de aprendizaje muy elevada, ya que se debe dedicar un buen tiempo a su estudio antes de poder utilizarlos, lo que retrasa el desarrollo y la culminación del proyecto.

Además puede ser que las herramientas no sean utilizadas en su real capacidad o en el peor de los casos que se abandonen los proyectos por no poder acoplarse al framework.

En consecuencia los frameworks ayudan al desarrollo rápido pero en su mayoría tienen una curva elevada de aprendizaje generando una inversión de tiempo adicional a la prevista que puede provocar el retraso o abandono del proyecto.

Adicionalmente la empresa Megasystem que es una empresa que distribuye equipos tecnológicos y al no tener una página web pierde oportunidades de venta y de ser conocida a nivel nacional y mundial, ventajas que ofrece la web.

1.3. Objetivos del Proyecto

1.3.1. Objetivo General: Estudiar la tecnología CodeIgniter aplicada al desarrollo de portales web con arquitectura MVC.

1.3.2. Objetivos Específicos:

- Estudiar el funcionamiento del framework CodeIgniter y la tecnología MVC (Modelo-Vista-Controlador).

- Implementar el portal web de información de precios y stock para la empresa Megasystem.
- Documentar el funcionamiento del portal mediante herramientas RUP.
- Realizar las pruebas del correcto funcionamiento del portal.

1.4. Justificación del Proyecto

El estudio del framework CodeIgniter beneficiará directamente a los estudiantes de la Carrera de Ingeniería en Sistemas y de la Facultad de Ingeniería en Ciencias Aplicadas FICA ya que tendrán un medio de apoyo para el desarrollo de las aplicaciones web y la integración al gestor de bases de datos MySQL utilizando esta herramienta, además podrán conocer una herramienta poco explotada a pesar de sus numerosas y muy buenas características ya que tiene una curva de aprendizaje muy baja lo que les permitirá desarrollar aplicaciones web muy completas en poco tiempo. Los profesores de la facultad también serán beneficiados ya que conocerán otra herramienta para enseñar en clase.

Indirectamente se beneficiará la empresa Megasystem y los usuarios que requieran productos informáticos ya que se creará una herramienta web que permitirá la difusión de los productos y servicios mediante el internet.

1.5. Alcance del Proyecto

1.5.1. Herramientas Tecnológicas

Se estudiará a la herramienta CodeIgniter con su arquitectura MVC en todo lo que tiene que ver a la realización de un portal web y su integración con el gestor de bases de datos MySQL.

1.5.2. Módulos del Portal Web de Información de Precios y Stock

EL portal web de información de precios y stock contendrá los siguientes módulos.

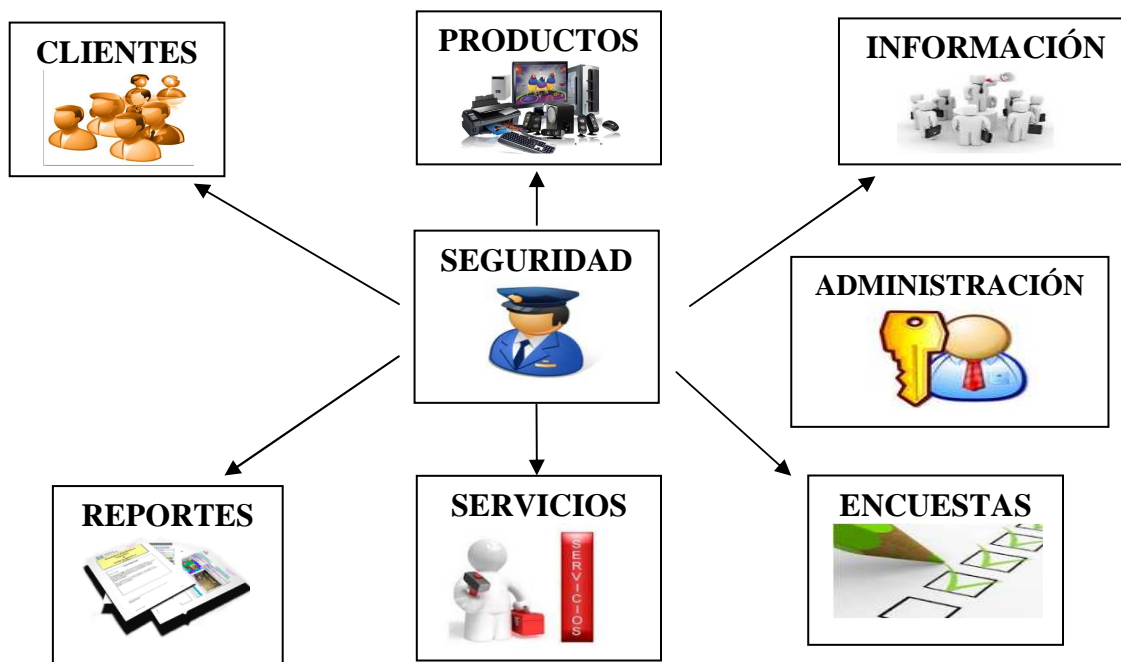


Figura 2: Módulos del Portal web de la Empresa Megasytem

Fuente: Propia

A) Módulo Clientes

- Registro de nuevos clientes
- Modificación de clientes
- Eliminación de clientes

B) Módulo de Productos

- Ingreso de productos
- Modificación de productos
- Eliminación de productos
- Catálogos de marcas
- Catálogo de productos por líneas
- Catálogo de productos novedosos
- Promociones de temporada

C) Módulo de Información

- Presentación
- Misión
- Visión
- Valores
- La Empresa-Instalaciones
- Información de contactos

D) Módulo de Servicios

- Servicios generales
- Realización de cotizaciones
- Formulario de Sugerencias
- Formulario de reclamos
- Formulario de contacto

E) Módulo de Reportes

- Reporte de clientes
- Reporte de productos
- Reporte de encuestas
- Reporte de cotizaciones
- Listas de precios

F) Módulo de Encuestas

- Encuestas de satisfacción del cliente
- Encuestas del producto

G) Módulo de Seguridad

- Login de ingreso
- Autenticación

H) Módulo de Administración

- Administración de usuarios
- Administración de contenidos

1.5.3. Arquitectura del Portal Web

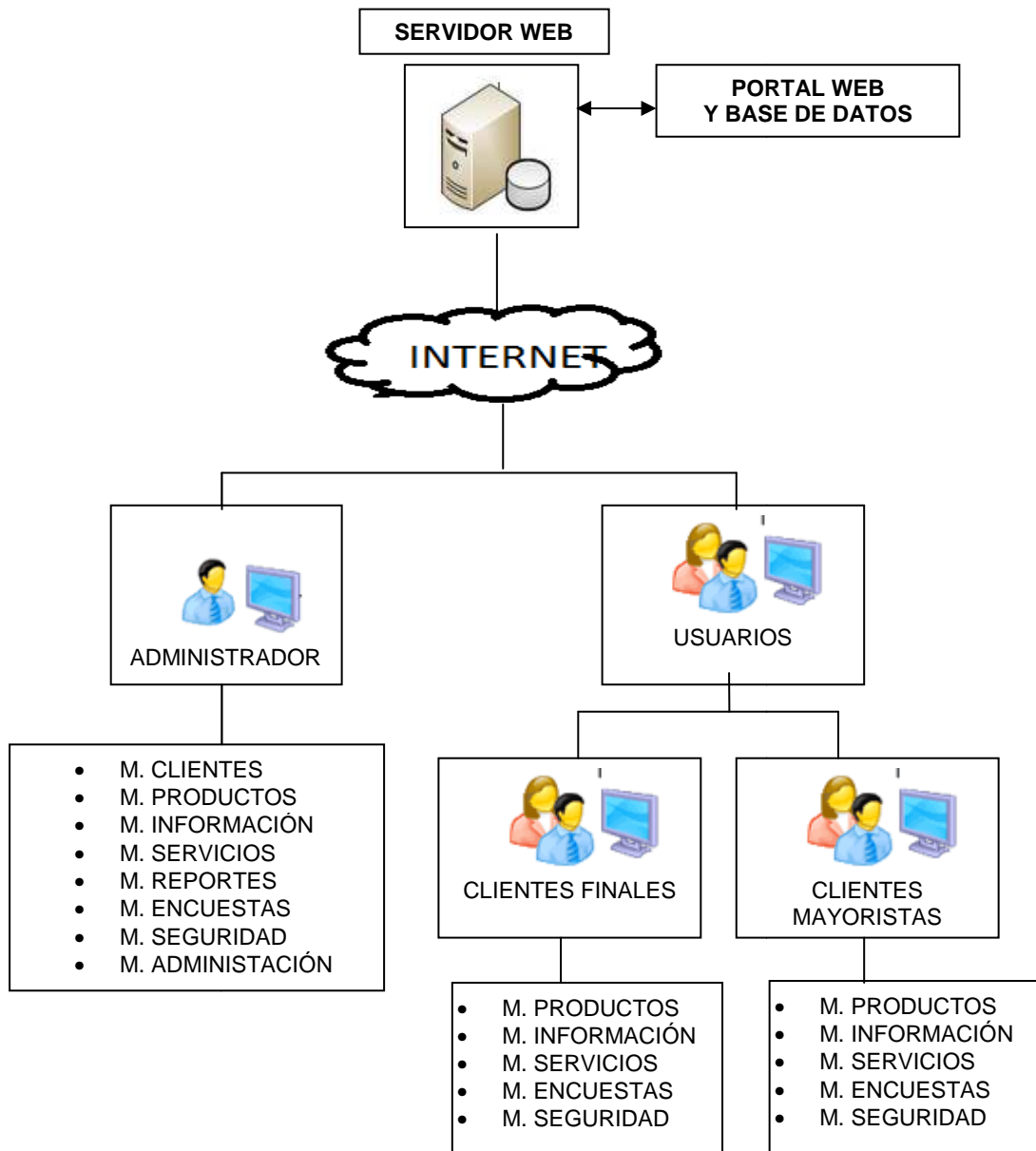


Figura 3: Arquitectura del Portal web de la Empresa Megasytem

Fuente: Propia

1.5.4. Restricciones

Solo se realizará la comunicación con clientes para proveer información de productos, precios, stock y cotizaciones on line, mas no se realizará ningún tipo de transacción económica, facturación o ventas en el Portal Web.

CAPÍTULO II

Estudio del Framework CodeIgniter y la Arquitectura MVC



Estudio del Framework CodeIgniter y la Arquitectura MVC

- 2.1 ¿Qué es CodeIgniter?
- 2.2 Estudio de CodeIgniter
- 2.3 Herramientas de Desarrollo de CodeIgniter

2. Estudio del Framework CodeIgniter y la Arquitectura MVC (Modelo Vista Controlador)

2.1. ¿Qué es CodeIgniter?

2.1.1. Introducción

En principio se debe indicar que es PHP¹:

“PHP es un lenguaje de alto nivel que se ejecuta en el servidor, un lenguaje de servidor es aquel que se ejecuta en el servidor donde están alojadas las paginas, al contrario de otros lenguajes que son ejecutados en el propio navegador, la principal ventaja es que, al ejecutarse el código en el servidor, todas las paginas van a poder ser vistas en cualquier ordenador, independientemente del navegador que tenga”. (Pavón, 2007, pág. 1).

Una vez entendido lo que es el lenguaje de programación PHP, se puede analizar lo que es un framework y codeigniter específicamente de la siguiente manera:

“Se puede decir que un framework es un programa para desarrollar otros programas, CodeIgniter² es un framework³, por tanto, es un programa o aplicación web desarrollada en PHP (Preprocesador de Hypertexto) para la creación de cualquier tipo de aplicación web bajo PHP. Es un producto de código libre y libre de uso para cualquier aplicación.

¹PHP es un acrónimo recursivo de PHP: Hypertext Preprocessor(Preprocesador de Hypertexto)

² CodeIgniter es un framework para desarrollo de aplicaciones web

³ Framework es una estructura de soporte definida, en la cual otro proyecto de software puede ser organizado y desarrollado.

Como cualquier otro framework, Codeigniter contiene una serie de librerías que sirven para el desarrollo de aplicaciones web y además propone una manera de desarrollarlas que se deben seguir para obtener provecho de la aplicación. Marca una manera específica de codificar las páginas web y clasificar sus diferentes scripts, que sirve para que el código esté organizado y sea más fácil de crear y mantener. CodeIgniter implementa el proceso de desarrollo llamado Modelo Vista Controlador (MVC), que es un estándar de programación de aplicaciones, utilizado tanto para hacer sitios web como programas tradicionales.

CodeIgniter contiene muchas ayudas para la creación de aplicaciones PHP avanzadas, que hacen el proceso de desarrollo más rápido. A la vez, define una arquitectura de desarrollo que hará que se programe de una manera más ordenada y contiene diversas herramientas que ayudan a hacer aplicaciones más versátiles y seguras.

CodeIgniter y otros frameworks PHP pueden ayudar a los desarrolladores PHP, creando aplicaciones web más profesionales y con código más reutilizable, con la diferencia que CodeIgniter está creado para que sea fácil de instalar en cualquier servidor y de empezar a usar que cualquier otro framework. Además muchas de sus utilidades y modos de funcionamiento son opcionales, lo que hace gozar de mayor libertad a la hora de desarrollar sitios web”.⁴ (Alvarez, n.d, pág. 2).

Rick Ellis es el creador del framework Codeigniter, quiso crear un framework con las características de Ruby on Rails pero manejando el lenguaje de programación PHP, para que los desarrolladores PHP no tengan el inconveniente de tener que aprender otro lenguaje. La empresa

⁴ Referencia tomada de la página www.desarrolloweb.com/manuales/manual-codeigniter.html

de desarrollo de software ExpressionEngine fundada por Rick Ellis es quien desarrolla y mantiene el framework Codeigniter en la actualidad.

El presente estudio está enfocado en aplicar todas las fortalezas del framework CodeIgniter en el desarrollo de un portal web, por lo que es importante entender que es un portal web y como funciona, lo que se hace a continuación con la siguiente definición:

“Un Portal Web es una página web o un conjunto de páginas web, orientado a lograr la participación del usuario final, con el fin de obtener algo de él, podría ser información más en detalle sobre sí mismo para poder utilizarla luego en campañas personalizadas, puede ser el cierre de una venta, en este caso el portal podría ser una tienda virtual que permita que el usuario final seleccione los productos o servicios, defina la forma de pago y realice el pedido, es posible que el portal esté dirigido a recibir la información de quejas y reclamos por parte de los usuarios, que discrimine la persona responsable y le haga llegar la información. Es mucho más que una página web, porque es completamente dinámico, un portal no solo provee al usuario información, sino que la recopila, lo que se espera es que éste interactúe con la empresa.

Un portal normalmente está desarrollado en algún lenguaje más poderoso y complejo que HTML, puede ser PHP o Java, y normalmente está asociado a una base de datos que almacena tanto la información que se quiere presentar como la que se obtiene del usuario del portal. Dicha información es totalmente aprovechable y en el caso de la empresa, la información que se quiere presentar se debe poder administrar desde un área diseñada para que un usuario común dentro de la empresa, pueda actualizarla, modificarla, añadir nuevo contenido, del tipo que sea, sin necesidad de expertos ya que el concepto de usuario abarca también al personal interno. Es una

aplicación de negocios que integra a la empresa con el mundo exterior y permite de una forma virtual desarrollar nuevas oportunidades y mejorar la eficiencia de los recursos, ampliar la cobertura sin necesidad de inversiones físicas, una herramienta para relacionarse más estrechamente con los clientes y crear fidelidad”.(GoDaddy Dominios)

2.1.2. Características de CodeIgniter

Algunas características fuertes que posee el framework, sobre todo en comparación con otros frameworks similares, son los siguientes:

Libre.-está liberado bajo licencias open source.

Liviano.- “El núcleo del sistema es muy liviano, sólo requiere algunas bibliotecas muy pequeñas para funcionar. Las bibliotecas adicionales se cargan dinámicamente cuando se van a utilizar, dando como resultado que el sistema base sea muy ligero y bastante rápido”.(EllisLab, 2008-2011, pág. 14)

Versátil: “trabaja en la mayoría de entornos o servidores, incluso en sistemas de alojamiento compartido, donde sólo se tiene acceso por FTP para enviar los archivos al servidor y donde no se tiene acceso a su configuración”. (Alvarez, n.d, pág. 2)

Rápido.- tiene un excelente desempeño y rapidez de respuesta a las peticiones.

Usa M-V-C.-usa el enfoque Modelo-Vista-Controlador, que permite la separación entre la lógica y la presentación.

Genera URLs Claras.- “Las URLs generadas son claras y amigables con los motores de búsqueda. En lugar de usar el enfoque estándar "query string" característico de sistemas dinámicos, CodeIgniter usa el enfoque basado en segmentos del siguiente modo: `www.ejemplo.com/controlador/función/id`”.(EllisLab, 2008-2011, pág. 14)

Trae varias Bibliotecas.- “contiene varias bibliotecas que facilitan las tareas de desarrollo web más comúnmente usadas, como acceso a base de datos, envío de correo electrónico, validación de datos de formularios, manejo de sesiones, manipulación de imágenes, entre otros”.(EllisLab, 2008-2011, pág. 14)

Flexibilidad: “es bastante menos rígido que otros frameworks. Define una manera de trabajar específica, pero se puede seguir o no sus reglas de codificación”.(Alvarez, n.d, pág. 2)

Extensible.- El sistema se puede extender fácilmente a través de bibliotecas, helpers o extensiones de clases creadas por el desarrollador.

Facilidad de instalación: “solo se necesita una cuenta de FTP para subir CodeIgniter al servidor y la configuración se realiza con apenas la edición de un archivo. No se necesita acceso con la línea de comandos para la configuración”.(Alvarez, n.d, pág. 2)

Completamente Documentado.- el código fuente viene con comentarios que indican lo que realiza cada clase o función para que el desarrollador entienda como está estructurado el código y si necesita también puede modificarlo.

2.2. Estudio de CodeIgniter

La finalidad al realizar un estudio del framework CodeIgniter es de dar a conocer una herramienta de desarrollo de aplicaciones web muy potente y rápida de aprender, además de estudiar y poner en práctica la estructuración de un proyecto usando la arquitectura de software Modelo-Vista-Controlador y sus ventajas de separación del código de la lógica con la presentación, la curva de aprendizaje de esta herramienta es muy pequeña combinada con la cantidad de librerías y helpers que posee permiten esperar que se reduzcan los tiempos de desarrollo al mismo tiempo de generar aplicaciones más seguras, flexibles y con componentes reutilizables.

2.2.1. Diagrama de Flujo de una Aplicación con CodeIgniter

CodeIgniter maneja el flujo de datos mediante la ejecución de varios procedimientos que se ejecutan internamente, para atender la solicitud de página del cliente y presentar la información en pantalla, utiliza módulos creados por el desarrollador como los controladores y las vistas y también módulos incorporados en el framework como librerías y helpers entre otros, el flujo de datos se maneja como indica el siguiente gráfico:



Figura 4: Diagrama de flujo de una aplicación CodeIgniter

Fuente: <http://www.etnassoft.com/biblioteca/codeigniter-guia-del-usuario-en-espanol-v-2-1-0/>

Los procedimientos se explican a continuación:

- “El index.php sirve como el controlador frontal, inicializando los recursos básicos que necesita CodeIgniter para ejecutar.
- El Ruteador examina la solicitud HTTP⁵ y se realiza un filtrado de la URL para saber cuál es el elemento que tiene que procesar esta página.
- Si la página se generó anteriormente y está en la caché, se lo envía directamente al navegador, sin pasar por la ejecución normal del sistema, para ahorrar procesamiento.
- Antes que se cargue el controlador de la aplicación, por razones de seguridad se filtran la solicitud HTTP y cualquier otro dato enviado por el usuario.
- El controlador adecuado carga el modelo, las bibliotecas del núcleo, helpers, y cualquier otro recurso requerido para procesar una solicitud específica.
- La Vista terminada se procesa y se envía al navegador para que se pueda ver. Si el caché está habilitado, la vista se guarda en cache primero para que las siguientes solicitudes que la necesiten puedan ser servidas”.(EllisLab, 2008-2011, pág. 17)

2.2.2. Arquitectura MVC (Modelo-Vista-Controlador) de CodeIgniter

CodeIgniter utiliza la arquitectura de software Modelo-Vista-Controlador (MVC) que es un enfoque de software que separa una página web en varias partes como son la lógica de la aplicación, la presentación y el acceso a la base de datos, este enfoque permite a las páginas web contener mínimo código PHP ya que la presentación de la vista está separada de la lógica, además permite tener varias ventajas como tener el código más

⁵ HTTP (HyperText Transfer Protocol- Protocolo de Transferencia de Hipertexto)

organizado, extensible y entendible por otros desarrolladores, es reutilizable y con más facilidad de mantenimiento, a continuación se muestra un esquema de cómo funciona la arquitectura MVC.

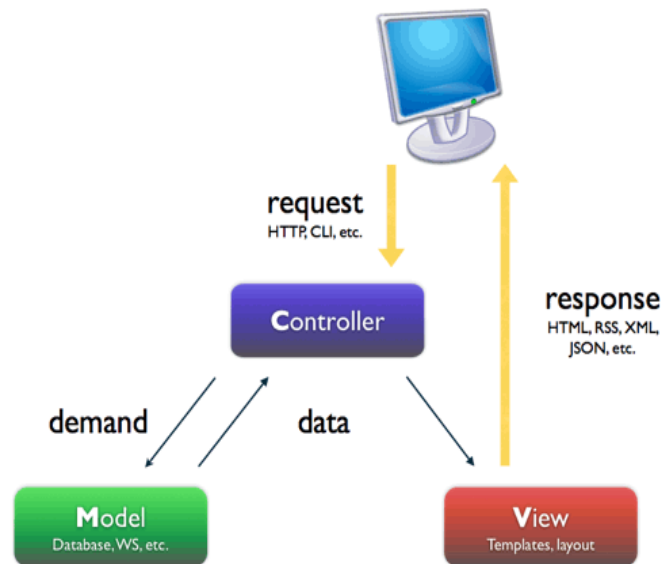


Figura 5: Diagrama de Arquitectura MVC

Fuente: <http://trib.us.es/ticpas/files/2011/04/Codeigniter.pdf>

El Modelo

Según los desarrolladores de CodeIgniter se dice que:

“El modelo representa las estructuras de datos, es decir todo el código que tiene que ver con el acceso a la base de datos. Las clases del modelo contienen funciones que ayudan a recibir, insertar, actualizar o borrar información de las tablas” (EllisLab, 2008-2011, pág. 18).

Una de las ventajas más importantes de utilizar los modelos es que todo el código que gestiona la base de datos está centralizado en una misma clase

y pueden ser utilizadas por diferentes funciones de los diferentes controladores.

La Vista

En lo que respecta a la vista, se puede definir como:

“la información que se presenta al usuario, una vista será normalmente una página web, un fragmento de página como el encabezado o pie de página o cualquier otro tipo de página. En la vista se coloca todo el código HTML, CSS, Java Script, etc. que sirve para generar la página” (Alvarez, n.d, pág. 7).

El Controlador

Por otra parte es necesario comprender que:

“El Controlador es la parte más importante del framework ya que sirve como un intermediario entre el Modelo, la Vista y cualquier otro recurso necesario para procesar la solicitud HTTP y generar una página web, en el controlador se guarda la lógica de las páginas” (Alvarez, n.d, pág. 7).

CodeIgniter tiene entre sus características el ser un framework flexible, por lo que le da la opción al desarrollador de no usar los Modelos, se puede construir una aplicación usando solamente Controladores y Vistas, pero con eso se pierden las ventajas de la separación del código.

2.2.3. Objetivos de Diseño y Arquitectura de CodeIgniter

Desde el punto de vista técnico y arquitectónico, CodeIgniter se creó con los siguientes objetivos:

Instanciación Dinámica.- “los componentes se cargan y las rutinas se ejecutan solamente cuando se necesita, en lugar de hacerlo globalmente. Los eventos, disparados por la solicitud HTTP, los controladores y las vistas que se diseñen determinarán lo que se invoque”.(EllisLab, 2008-2011, pág. 19)

Poco Acoplamiento.- “el acoplamiento es el grado en que los componentes de un sistema dependen unos de otros. Mientras menos componentes dependan unos de otros, más reusable y flexible será el sistema”.(EllisLab, 2008-2011, pág. 19)

Singularidad de Componentes.- “la singularidad es el grado en que los componentes tienen un propósito muy específico. En CodeIgniter, cada clase y sus funciones son muy autónomas a fin de permitir la máxima utilidad”.(EllisLab, 2008-2011, pág. 19)

En conclusión se puede decir que CodeIgniter es un sistema instanciado dinámicamente, poco acoplado y con alta singularidad de componentes. Todas estas características hacen que Codeigniter tenga un alto desempeño al momento de poner en producción una página web.

2.2.4. Estructura de CodeIgniter

Una vez instalados los archivos de CodeIgniter, se pueden ver los nuevos directorios que se han creado. Se puede observar claramente tres carpetas en donde se desarrollará toda la aplicación. La carpeta raíz debe contener algo como la figura siguiente.

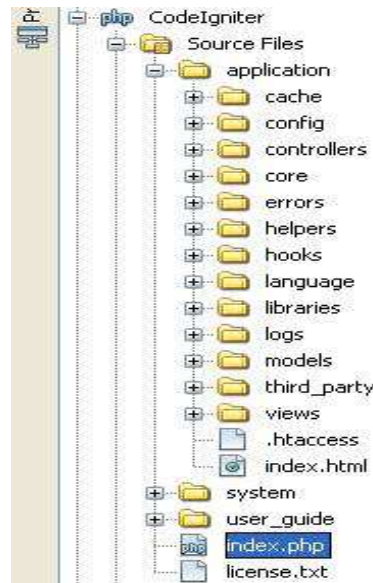


Figura 6: Estructura de CodeIgniter

Fuente: Propia

En la carpeta **system** es:

“Donde está todo el motor del framework. Los archivos en la carpeta **system** contienen el código del sistema de CodeIgniter (system / libraries, system / CodeIgniter, system / drivers, etc.), estos archivos se pueden leer y modificar si se desea, pero no es aconsejable. Si se altera el código base, al tener que modificar de nuevo cuando se descarga una actualización de CodeIgniter, la nueva versión puede sobrescribir las alteraciones” (Upton, 2007, pág. 23).

En la carpeta **user_guide**, es donde está toda la información de ayudas que nos ofrece CodeIgniter.

En la carpeta **application** es el lugar en donde:

“se encuentra todo el código de la aplicación que se está construyendo. Básicamente, esta carpeta contiene los modelos, vistas, controladores y otro tipo de código (helpers y extensiones de

clases). En otras palabras, esta carpeta es donde se hace el 99 por ciento del trabajo”(Myer, 2008, pág. 49).

Es básicamente el lugar donde se desarrolla la aplicación web y consta de las siguientes carpetas:

- **Config:** Ficheros de configuración del propio codeigniter.
- **Controller:** En donde se guardan los controladores.
- **Error:** Clases para poder gestionar los errores.
- **Helpers:** Funciones que ayudan a mostrar o generar el contenido.
- **Hooks:** Son funciones que proveen un medio para aprovechar y modificar el funcionamiento interno del framework sin alterar los archivos del núcleo.
- **Language:** Funciones para las traducciones a otros idiomas.
- **Libraries:** Donde se guardan las clases propias del desarrollador o de terceros para utilizarlas en los controladores.
- **Models:** Clases para trabajar con la capa de base de datos.
- **Views:** Donde se crean las vista para mostrar el contenido.

2.2.5. Funcionalidades de CodeIgniter

Entre las funcionalidades principales de CodeIgniter tenemos las siguientes:

- Sistema basado en Modelo-Vista-Controlador
- Extremadamente liviano
- Clases de base de datos con soporte para varias plataformas
- Soporte para base de datos con Active Record
- Validación de datos y formularios
- Seguridad y filtrado XSS⁶

⁶ Cross-site scripting (Secuencias de comandos en sitios cruzados)

- Administración de sesiones
- Clase para enviar Email, soporta adjuntos, email de HTML/Tex
- Bibliotecas de manipulación de imágenes (recorte, re dimensión, rotación, etc.).
- Clase para subir archivos
- Clase para FTP
- Localización
- Paginación
- Encriptación de datos
- Evaluación de rendimiento
- Caché de página completa
- Historial de errores
- Perfilado de la aplicación
- Clase para Calendarios
- Clase para Agente del Usuario
- Clase para codificación Zip
- Clase de motor de plantillas
- Clase para Trackback
- Biblioteca XML-RPC
- Clase para pruebas de unidad
- URLs amigables para los motores de búsqueda
- Ruteo URI flexible
- Soporte para hooks y extensiones de clase
- Amplia biblioteca de funciones "helper"(EllisLab, 2008-2011, pág. 16)

2.2.6. Instalación

Descargar CodeIgniter

La última versión de CodeIgniter se puede descargar de la página web <http://www.codeigniter.com>.

Requisitos del Servidor

Para la versión de codeigniter 2.1.0 son necesarios los siguientes requisitos de servidor:

- PHP versión 5.1.6 o más reciente.
- Las bases de datos que se soportan actualmente Codeigniter son MySQL (4.1+), MySQLi, MS SQL, Postgres, Oracle, SQLite, y ODBC.

Instrucciones de Instalación

CodeIgniter se instala en cuatro pasos:

- Descomprimir el paquete y renombrar la carpeta con el nombre del proyecto a crear.
- Subir la carpeta del proyecto al servidor, el archivo **index.php** será la raíz de la página web.
- Abrir el archivo **application/config/config.php** y establecer la URL⁷ base.
- Si se va a usar una base de datos, abrir el archivo **application/config/database.php** y establecer la configuración de la base de datos.

Luego de seguir estos pasos ya se tiene instalado codeigniter en el servidor, para probarlo se ingresa la URL con el nombre del proyecto en un navegador y se verá la página de bienvenida que confirma que está correctamente instalado y en funcionamiento.

2.2.7. Las URLs de CodeIgniter

En CodeIgniter se espera la facilidad tanto en su uso como en su entendimiento, por tal motivo al hablar de las URLs se puede decir que “las URLs se presentan en un formato amigable para los motores de búsqueda, también son fáciles de entender y recordar para las personas,

⁷ URL Uniform Resource Locator (Localizador Uniforme de Recursos)

esto quiere decir que los motores de búsqueda les darán una puntuación positiva” (Alvarez, n.d, pág. 8).

En lugar de usar el enfoque estándar de las "query string"(www.ejemplo.com/articulos.php?id=10), se usa un enfoque basado en segmentos(www.midominio.com/controlador/función/parametro por ejemplo www.ejemplo.com/artículos/muestra/10) que es más fácil de recordar para una persona.

Segmentos URI⁸

Según la arquitectura Modelo-Vista-Controlador, los segmentos en la URL normalmente se representan de la siguiente forma: **www.ejemplo.com/clase /función /ID**. Cada parte de la URL sirve para identificar y determinar cómo se procesa la página.

- El primer segmento es el nombre del dominio donde está instalado CodeIgniter.
- El segundo segmento representa la **clase** del controlador que se invoca para procesar la solicitud.
- El tercer segmento representa la **función** o método que está dentro del controlador y contiene el código que genera la página.
- El cuarto segmento, representa el **ID** y cualquier variable que se pasará al controlador, si se desea pasar varios parámetros, se los puede poner seguidos, separados por barras.

CodeIgniter tiene un archivo index.php que se encuentra en la raíz del framework por medio del cual pasan todas las solicitudes a las páginas. Es por eso que las URLs incluyen el nombre del archivo index.php (www.ejemplo.com/**index.php**/controlador/función/id). Se puede quitar este archivo index.php usando un archivo **.htaccess** en la raíz del framework con algunas reglas simples como la siguientes.

⁸ URI Uniform Resource Identifiers(Identificador de Recursos Uniforme)

```
Rewrite Engine on
Rewrite Cond $1 !^(index.php|images|robots.txt)
Rewrite Rule ^(.*)$ /index.php/$1 [L]
```

Cualquier solicitud HTTP distinta de **index.php**, imágenes, y robots.txt se trata como una solicitud al archivo **index.php**. Finalmente se edita la variable de configuración "index_page" en el archivo (**system/application/config/config.php**), poniéndola a la cadena vacía (`$config['index_page'] = "";`) y ya no saldrá en la URL del navegador.

2.2.8. Controladores en CodeIgniter

Dentro de Codeigniter y para cualquier otro tipo de framework “los controladores son el corazón de la aplicación, ya que determinan como se manejan las solicitudes HTTP. Un Controlador es un archivo de clase que se nombra de una forma en la que se puede asociar con una URI”(EllisLab, 2008-2011, pág. 36), por ejemplo **www.ejemlo.com/index.php/nombre_controller** y un ejemplo de código de un controlador sería el siguiente.

```
<?php
class Nombre_Controller extends CI_Controller
{
    public function index()
    {
        echo 'Hola Mundo!';
    }

    public function comentarios()
    {
        echo 'Esto es un comentario';
    }
}
?>
```

En la URI **www.ejemlo.com/index.php/nombre_controller**, se busca un controlador llamado **Nombre_Controller.php** para cargarlo y ejecutarlo,

siempre y cuando el nombre del Controlador coincida con el primer segmento de la URI. Por regla general los nombres de clases tienen que comenzar con una letra mayúscula, el nombre del archivo tiene que ser el mismo de la clase pero todo en minúsculas y finalmente se debe extender la clase del controlador padre. Todos los controladores se deben almacenar en el directorio **application/controllers**

Funciones en los Controladores

El segundo segmento de la URI después del dominio y del index.php indica la función del controlador que se está llamando. En el ejemplo **www.ejemlo.com/index.php/nombre_controller**, el nombre de la función que se llama es **index ()**, la función "index" se carga siempre por defecto si el segundo segmento de la URI está vacío. En el caso de este ejemplo **www.ejemlo.com/index.php/nombre_controller/comentarios**, se está llamando a la función **comentarios**. En el caso siguiente **www.ejemlo.com/index.php/nombre_controller/comentarios/2** el tercer segmento representa un parámetro que se envía a la función para ser procesado.

Constructores

Los constructores son métodos que se usan para establecer valores por defecto o ejecutar procesos por defecto al crear la instancia de la clase.

```
<?php
class Nombre_Controller extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();
        // Código del constructor
    }
    public function index()
    {
        echo 'Hola Mundo!';
    }
}
?>
```

2.2.9. Vistas en CodeIgniter

Una **vista** para CodeIgniter “es una página web completa o solo un fragmento de página, como un encabezado, pie de página, una barra lateral, etc. Las vistas se pueden embeber flexiblemente dentro de otras vistas”(EllisLab, 2008-2011, pág. 43).

Por ejemplo el siguiente código HTML es el de una página estática que solo realiza la presentación en pantalla de una cadena, pero se puede cargar como una vista.

```
<html>
<head>
<title>Mi Vista</title>
</head>
<body>
<h1>Mi primera Vista</h1>
</body>
</html>
```

En Codeigniter las vistas no se acceden desde una solicitud URL, se tienen que cargar a través de un controlador de la siguiente forma.

```
<?php
class Nombre_Controller extends CI_Controller {
    function index()
    {
        $this->load->view('mi_vista');//Carga la vista
        $this->load->view('mi_vista2');//Carga la vista2
    }
}
?>
```


De esta forma los controladores manejan la lógica de la petición y la vista muestra los resultados con un diseño de página agradable. Todos los archivos de vistas se almacenan en el directorio **application/views/**.

Se puede organizar mejor el contenido de una vista, dividiendo el código por partes y en varias vistas distintas, al organizar el contenido de la vista por partes como encabezado, pie, barra lateral, se puede hacer el llamado en el controlador a varias vistas para formar la página que verá el usuario.

Agregar Datos Dinámicos a la Vista

Para poder hacer que la información sea dinámica en la vista y esta muestre diferentes datos con el mismo código, en el controlador se almacenan los datos en un array y luego se los pasa a la vista usando el segundo parámetro en la función que carga la vista.

```
<?php
class Nombre_Controller extends CI_Controller {
function index()
{
    $data['titulo'] = "Mi Titulo";
    $data['cabeza'] = "Mi Encabezado";

    $this->load->view("mi_vista", $data);
}
}
?>
```

Y en el archivo de la vista ya se puede trabajar con las variables enviadas por el controlador:

```
<html>
<head>
<title><?php echo $titulo;?></title>
</head>
<body>
<h1><?php echo $cabeza;?></h1>
</body>
</html>
```

Devolver Vistas como Datos tipo String

Cuando se necesita cargar varias vistas simultáneamente, en ocasiones es necesario procesar de alguna forma el código HTML que genera la vista y no presentarla en pantalla directamente, esto se puede lograr almacenando la salida de la vista en una variable. Para que funcione correctamente se envía un tercer parámetro en la función que carga la vista con el valor “TRUE”:

```
$string = $this->load->view('mi_vista', '$data', TRUE);
```

2.2.10. Modelos en CodeIgniter

Se pueden definir los modelos en codeigniter de la siguiente manera:

“Los modelos son los módulos que tienen como responsabilidad el acceso y control de los datos que hay en la base de datos y mantienen encapsuladas todas las funciones de los accesos a las tablas para realizar cualquier tipo de operación como insertar, eliminar, actualizar y devolver los datos al controlador.

Al centralizar las acciones sobre la base de datos en los modelos, a los controladores no les interesa lo compleja o simple que sea la base de datos incluso de qué tipo de base de datos tiene el servidor con el que se está trabajando”(Alvarez, n.d, págs. 22-23).

En Codeigniter los modelos son opcionales, se puede gestionar la información de la base de datos desde el mismo controlador aprovechando la flexibilidad del framework, aunque es aconsejable usar los modelos especialmente si se realizan proyectos grandes, su estructura es la siguiente.

```
Class Nombre_Model extends CI_Model
{
    function __construct()
    {
        parent::__construct();
    }
    //Código de funciones
}
```

Al igual que en las clases controladores, los nombres de las clases de modelo deben empezar con una letra mayúscula y se debe extender la clase modelo padre, el nombre del archivo se escribe solo con minúsculas, los archivos de modelo se guardan en el directorio **application/models/**

Los modelos se cargan desde las funciones del controlador que los va a utilizar o desde el constructor de la clase de la siguiente forma:

```
$this->load->model ('Nombre_model');
```

Una vez que ya se tiene cargado el modelo se puede acceder a sus funciones en el controlador usando un objeto con el mismo nombre que la clase modelo y especificando el nombre de la función a utilizar:

```
$this->Nombre_model->función ();
```

2.2.11. Funciones Helper

Los archivos de helper se definen como:

“Una colección de funciones ordenados por temáticas en diferentes paquetes, que ayudan en la realización de tareas generales en el desarrollo de aplicaciones web, están creadas para que el

desarrollador no tenga que programar funciones que se utilizan frecuentemente en la mayoría de sitios web.

Los Helpers en Codeigniter no se escriben en formato Orientado a Objetos, son simples funciones procedimentales que ejecutan una tarea específica, sin dependencia con otras funciones”(Alvarez, n.d, pág. 34).

Los helpers se encuentran almacenados en el directorio **system/helpers** o **application/helpers**. La lista de Helpers se muestra en el anexo B.

Para aumentar el rendimiento del sistema, “CodeIgniter no carga de forma predeterminada ninguna librería excepto algunas librerías muy básicas. Por eso se deben cargar los helpers y cualquier otra librería en todas las páginas a través de los archivos de configuración, o en una página en concreto a través de las funciones de carga de helpers”(Alvarez, n.d, pág. 35). Se carga un archivo Helper usando la siguiente función:

```
$this->load->helper ('nombre_helper');
```

Un helper puede ser cargado en cualquier parte del controlador o en una vista. Una vez cargado ya se pueden utilizar las funciones que se necesiten del helper.

Para cargar automáticamente los Helpers en toda la aplicación y no tener que estar cargándolos cada vez que se los quiere usar, existen los archivos de configuración. Se configura el helper de array de carga automática del archivo **application/config/autoload.php** cargando todos los helpers que se vayan a necesitar.

```
$autoload['helper'] = array('url', 'html', 'form');
```

2.2.12. Bibliotecas o Librerías de CodeIgniter

Las librerías para el framework CodeIgniter “son clases de programación orientada a objetos creadas para realizar tareas típicas en el desarrollo de páginas web. Permiten resolver problemas similares a los que resuelven los helpers, pero son más especializadas en temas particulares”. (Alvarez, n.d, pág. 40)

Todas las bibliotecas o librerías se encuentran almacenadas en el archivo **system/libraries**. La lista de librerías se muestra en el anexo C.

Para poder usar una librería se debe inicializarla en un controlador de la forma siguiente:

```
$this->load->library ('nombre_de_clase');
```

Una vez cargada la librería en el controlador ya se pueden utilizar sus métodos y sus funcionalidades, de la forma en que indica el manual de cada librería.

Se pueden cargar varias bibliotecas al mismo tiempo para que estén disponibles en todas las páginas sin necesidad de cargarlas en los controladores. Esto se hace editando el archivo `autoload.php`, que está en la ruta: **system/application/config/autoload.php**.

En ese archivo se cargan todas las librerías a utilizar.

```
$autoload['libraries'] = array('database', 'session', 'form_validation');
```

Crear Bibliotecas Propias

Codeigniter permite hacer las siguientes operaciones con las bibliotecas:

- Crear bibliotecas o librerías nuevas.

- Reemplazar bibliotecas o librerías nativas.
- Extender bibliotecas o librerías nativas.

Las librerías que se crean se deben guardar en la carpeta **application/libraries** y no en **system/libraries**, porque es en ese lugar donde CodeIgniter las buscará primero para inicializarlas, además de permitir separarlas de las nativas.

Para reemplazar las librerías nativas que se encuentran en **system/libraries**, se nombra el nuevo archivo de clase y la declaración de clase igual que las librerías nativas a reemplazar y se guarda en la misma carpeta, con este método CodeIgniter usará la nueva clase en lugar de la nativa.

Para añadir alguna funcionalidad específica a una biblioteca que ya existe en CodeIgniter se extiende su clase nativa. Para extender una clase librería, se crea una clase nueva y en su declaración se extiende la clase nativa como clase padre y en el nuevo nombre de clase y nombre de archivo se pone el prefijo (**MY_**) seguido del nombre de clase de la librería a extender.

```
Class MY_Libreria extends CI_Libreria {  
  
    //Nuevas funciones  
  
}
```

Esta nueva clase se guarda en **application/libraries**. Para cargar y usar las funciones de una librería extendida, se lo hace de la misma forma como si fuera una librería nativa, sin poner el prefijo.

```
$this->load->library ('nombre_de_clase');
```

2.2.13. Drivers de CodeIgniter

Los Drivers se definen como:

“Un tipo especial de Biblioteca que tienen una clase padre y cualquier cantidad de clases hijas potenciales. Las clases hijas tienen acceso a la clase padre, pero no a sus hermanas. Los Drivers están ubicados en la carpeta **system/libraries**, en su propia carpeta que se llama igual que la clase de la biblioteca padre. También dentro de esa carpeta hay una subcarpeta llamada **drivers**, que contiene todos los posibles archivos de clases hijas”.⁹(EllisLab, 2008-2011, pág. 59)

Un driver es inicializado en un controlador de la siguiente manera.

```
$this->load->driver ('nombre de clase');
```

Y para invocar los métodos de las clases se usa la función:

```
$this->algun_padre->algun_metodo ();
```

2.2.14. Ruteo URI

El ruteo en Codeigniter sirve para hacer que se llame a un controlador y a una función diferente a la especificada en la URL, esto se logra editando las reglas de ruteo en el archivo **application/config/routes.php**. Se debe configurar el array **\$route** usando **Comodines** o **Expresiones Regulares**.

Usando Comodines

Para usar comodines en el array **\$route**, en los valores entre corchetes se especifica una URI que debe coincidir con la URL enviada por el

⁹Referencia tomada de

<http://www.etnassoft.com/biblioteca/codeigniter-guia-el-usuario-en-espanol-v-2-1-0/>

navegador, y en los valores que se le asigna al array \$route se determina la nueva ruta de destino a la que se va a redirigir la página.

```
$route ['1er_segmento/:num'] = "controlador/función";
```

En una URL el 1er_segmento, es el segmento que sigue después del index.php, en el segundo valor entre corchetes se puede poner una de las palabras num o any

(:num) es un segmento que contiene solamente números.

(:any) es un segmento que contiene cualquier caracter.

Por ejemplo en la siguiente regla:

```
$route ['clientes/:num'] = "productos/imprimir";
```

Indica que si en la URL se envía algo como www.dominio.com/index.php/clientes/5 se cargará la clase productos con la función imprimir.

Usando Expresiones Regulares

CodeIgniter permite utilizar cualquier expresión regular válida para definir las reglas de ruteo.

```
$route['productos/([a-z]+)/(\d+)'] = "$1/id_$2";
```

2.2.15. Manejo de Errores en CodeIgniter

Codeigniter maneja los errores que se producen mediante procedimientos globales en cualquier parte de la aplicación, estos procedimientos generan unos mensajes de error que se guardan en un archivo de texto.

Las siguientes funciones permiten generar errores:

- **show_error('mensaje' [, int \$codigo_estado= 500]):** Muestra el mensaje de error usando la plantilla de error:**application/errors/error_general.php**
- **show_404('página' [, 'log_error']):** Muestra el mensaje del error 404 usando la plantilla de error:**application/errors/error_404.php**
- **log_message('level' , 'message'):** Permite escribir mensajes en los archivos de registro.

2.2.16. Almacenamiento en Caché de Páginas Web

CodeIgniter para hacer más rápida la respuesta a las peticiones realiza lo siguiente:

“permite cachear las páginas con el fin de lograr un máximo rendimiento que se aproxima al de las páginas web estáticas, debido a que estas se guardan completamente renderizadas. Cuando se carga por primera vez una página, esta se guarda en la carpeta **application/cache**. En las siguientes cargas de la página, se recupera el archivo de caché y es enviado al navegador del usuario”(EllisLab, 2008-2011, pág. 73).

Para poder almacenar las páginas en caché, se usa la función de cacheo de la siguiente manera:

```
$this->output->cache(n);
```

Esta función se pone dentro de una función de controlador que genere una salida con una vista. La variable **n** representa la cantidad de minutos que la página debe permanecer en cache.

2.2.17. Administrar Aplicaciones en CodeIgniter

Se pueden tener varias aplicaciones diferentes compartiendo una sola instalación de CodeIgniter. Para lo cual se crea una subcarpeta y en esta se guardan todos los archivos que se encuentran dentro de la carpeta application, este proceso se repite para el número de aplicaciones que se necesiten.

```
application/aplicación1/
application/ aplicación1/config/
application/ aplicación1/controllers/
application/ aplicación1/errors/
application/ aplicación1/libraries/
application/aplicación1/models/
application/ aplicación1/views/

application/ aplicación2/
application/ aplicación2/config/
application/aplicación2/controllers/
application/aplicación2/errors/
application/aplicación2/libraries/
application/aplicación2/models/
application/aplicación2/views/
```

Para determinar cuál será la aplicación con la que se va a trabajar, se configura el archivo **index.php** principal, estableciendo la variable `$application_folder` con el nombre de la aplicación de la siguiente forma:

```
$application_folder = "application/nombre_aplicación";
```

2.2.18. Seguridad en CodeIgniter

CodeIgniter maneja varias formas para mantener la seguridad de las aplicaciones entre las que se pueden mencionar las siguientes:

Seguridad URI

CodeIgniter tiene mucho cuidado de que código o datos maliciosos sean ingresados a la aplicación a través de la URI de petición del navegador

web. Este control lo realiza restringiendo el tipo de caracteres que se permite en las cadenas URI.

Los caracteres permitidos para las cadenas URIs son los siguientes:

- Texto alfanumérico
- Tilde: ~
- Punto: .
- Dos puntos: :
- Guión bajo: _
- Guión medio: -

Register_globals

Otra manera para mantener la seguridad, es al iniciar el sistema debido a que “Durante la inicialización del sistema todas las variables globales se destruyen, excepto aquellas que se encuentran en los arrays **\$_POST** y **\$_COOKIE**. La rutina de destrucción es lo mismo que hacer **register_globals = off**”(EllisLab, 2008-2011, pág. 80).

error_reporting

Cuando la aplicación está en la etapa de desarrollo es importante para los desarrolladores tener el reporte de los errores surgidos, pero cuando ya está en entorno de producción lista para subirse a la red, se tiene que evitar que se presenten en pantalla los errores nativos de PHP, esto se realiza modificando el archivo `index.php` principal, poniendo en la constante **ENVIRONMENT** el valor de **production** (producción) en lugar de **development** (desarrollo).

Otras Prácticas de Seguridad

CodeIgniter recomienda las siguientes normas generales de seguridad para aplicaciones:

“Antes de aceptar los datos en la aplicación, ya sean datos POST enviados desde un formulario, datos de COOKIE¹⁰, datos de URI, datos XML-RPC¹¹ data, es recomendable seguir los pasos siguientes:

- Filtrar los datos como si estuvieran contaminados. CodeIgniter viene con un filtro contra XSS (Cross Site Scripting). Este filtro busca técnicas usadas para ingresar código Javascript malicioso en los datos o algún tipo de código que intente secuestrar cookies.
- Validar los Datos para asegurar que cumplen con el tipo correcto, longitud, tamaño, etc. CodeIgniter tiene una Clase para Validación de Formularios que ayuda en la validación, filtrado y preparación de los datos.
- Escapar los datos antes de ingresarlos en la base de datos.”(EllisLab, 2008-2011, pág. 80)

2.2.19. Sintaxis General

CodeIgniter maneja la sintaxis del lenguaje de programación PHP, sin embargo tiene unas reglas particulares que se indican a continuación.

Etiqueta de Cierre de PHP

Para indicar la finalización de un archivo se lo puede hacer mediante la etiqueta de cierre de PHP (?>), pero para evitar cualquier tipo de error CodeIgniter determina que lo más aconsejable es usar un bloque de comentario en donde se indica que es el fin de archivo y la ubicación del mismo, por ejemplo.

¹⁰ Cookie es una pequeña información enviada por el sitio web y almacenado en el navegador del usuario

¹¹ XML-RPC es una forma en que dos computadoras se comunican por Internet usando XML, es un protocolo para usar servicios web

```
<?php  
  
    echo "Codigo Codeigniter";  
  
/* Fin de archivo.php */  
/* Ubicación: ./application/controllers/modulo/archivo.php */
```

Nomenclatura de Clases y Métodos

Para declarar una clase se siguen las siguientes reglas, el nombre debe comenzar con una letra mayúscula y se separa con un guión bajo si contienen varias palabras. Los nombres de las funciones que se declaran dentro de la clase se escriben en minúsculas y debe indicar lo que realiza.

```
class Clase_estudiante {  
  
    function ingreso()  
    {  
        //Código de función  
    }  
}
```

Nombres de Variable

Las reglas para declarar variables son las siguientes, el nombre se escribe solo con minúsculas y si tiene varias palabras se separan con guión bajo, en lo posible el nombre de la variable debe indicar lo que tendrá como contenido. Cuando se va a usar ciclos, las variables que hacen de iteradores deben tener nombres cortos, los nombres demasiado largos no se deben usar ni en la declaración de variables ni tampoco en la declaración de las clases.

```
$clientes_actuales  
  
$cadena  
  
for ($j = 0; $j < 10; $j++)
```

Comentarios

Comentar el código es una práctica muy importante debido a que facilita las tareas de mantenimiento de la aplicación. En CodeIgniter se puede usar comentarios en bloque antes de las declaraciones de clases y funciones, o comentarios en línea simple dentro del código.

```
/**
 * Clase_estudiante
 * @package Nombre del
 paquete
 * @author Nombre del autor
 * @link http://ejemplo.com
 */
class Clase_estudiante {

function ingreso()
{
    //Suma dos valores
    $sum=$a+$b;
}
}
```

Constantes

La declaración de constantes sigue las mismas reglas de las variables, con la diferencia que los nombres de las constantes se escriben con mayúsculas y no existen constantes de una sola letra como en las variables de iteración.

```
MY_CONSTANT
NEWLINE
SUPER_CLASS_VERSION
```

TRUE, FALSE y NULL

En ocasiones se necesita almacenar en las variables valores booleanos, estos valores son **TRUE**, **FALSE**, **NULL** y siempre deben escribirse con mayúsculas.

Operadores Lógicos

Cuando se va a realizar operaciones lógicas se siguen las siguientes recomendaciones, no se debe usar el símbolo (| |) debido a que se puede confundir con el número 11, es mejor usar la palabra (**OR**). Tampoco usar la palabra **AND** es mejor usar el símbolo (**&&**) pero codeigniter acepta los dos. Finalmente cuando se va a usar el símbolo de negación (!), se debe poner un espacio antes y después del símbolo.

```
if ($var1 OR $var2)
if ($var1 && $var2) // recomendado
if ( ! $var1)
```

Comparar Valores de Retorno

Para realizar comparaciones de igualdad se debe usar la simbología (===) y para comparaciones de diferencia el símbolo (!==) según sea el caso.

Un Archivo por Clase

Cuando se crean las clases para la aplicación es recomendable guardar cada clase en un archivo diferente, con esto se tiene el código separado y mejor estructurado.

Espacios en Blanco

Según se aconseja, es mejor “usar tabuladores en lugar de espacios en blanco en el código. Esto permite tener indentación en los niveles que se prefiera y personalización en cualquier aplicación que se use” (EllisLab, 2008-2011, pág. 86).

Indentación de Código

Una buena práctica de codificación es mantener el código indentado, para lo cual las llaves se deben poner al mismo nivel que las sentencias de control, además las llaves de apertura y cierre tienen que estar en la misma línea.

```
for ($i = 0; $i < 10; $i++)  
{  
    for ($j = 0; $j < 10; $j++)  
    {  
        // Código  
    }  
}
```

Cadenas

Para el manejo de cadenas de caracteres se usan las reglas siguientes, por lo general usar comillas simples, si se necesita imprimir el valor de una variable en la cadena se usa comillas dobles y la variable se encierra entre llaves, si dentro de la cadena hay otra cadena en la cual hay comillas simples entonces usar comillas dobles.

```
'Mi cadena'  
"Mi cadena {$var}"  
"SELECT nombre FROM tabla WHERE var = 'ana' "
```

Consultas SQL

En CodeIgniter cuando se trabaja con base de datos “las palabras clave de MySQL se escriben en mayúsculas: SELECT, INSERT, UPDATE, WHERE, AS, JOIN, ON, IN, etc. Las consultas largas se deben dividir en varias líneas para que sean más entendibles”(EllisLab, 2008-2011, pág. 91).

```
$query = $this->db->query ("SELECT detalle_producto,precio  
FROM productos  
WHERE prod_nov == 'si'  
AND prod_prom == 'si'");
```


2.3. Herramientas de Desarrollo de CodeIgniter

2.3.1 Netbeans

Existen disponibles dos productos: el NetBeans IDE y NetBeans Platform.

NetBeans IDE

A NetBeans IDE se lo define como:

“Un entorno de desarrollo, una herramienta para escribir, compilar, depurar y ejecutar programas. Está escrito en Java, pero puede servir para cualquier otro lenguaje de programación como PHP. Existe además un número importante de módulos para extender el NetBeans IDE. NetBeans IDE es un producto libre y gratuito sin restricciones de uso” (Netbeans.org, Bienvenido a Netbeans, n.d).

Características

- Soporte para crear interfaces gráficas de forma visual
- Desarrollo de aplicaciones web
- Control de versiones
- Colaboración entre varias personas
- Creación de aplicaciones compatibles con teléfonos móviles
- Resaltado de sintaxis
- Sus funcionalidades son ampliables mediante la instalación de packs. (Netbeans.org, Bienvenido a Netbeans, n.d)



Figura 7: Pantalla del IDE Netbeans

Fuente: Propia

NetBeans Platform

A NetBeans Platform se lo define como:

“Una base modular y extensible usada como estructura de integración para crear grandes aplicaciones de escritorio. Empresas independientes asociadas, especializadas en desarrollo de software, proporcionan extensiones adicionales que se integran fácilmente en la plataforma y que pueden también utilizarse para desarrollar sus propias herramientas y soluciones” ¹²(Netbeans.org, Netbeans Platform, n.d).

2.3.2. CodeLobster

CodeLobster PHP Edition es:

“Un editor de código potente pero fácil de utilizar, es principalmente destinado a la creación y edición rápida y fácil de archivos de sitios web. Permite editar archivos PHP, HTML, CSS,

¹² Referencia tomada de <http://netbeans.org>

JavaScript con compatibilidad con el motor de plantillas CMS Drupal, Joomla CMS, Smarty, biblioteca de JQuery, framework CodeIgniter y plataforma de blogs de WordPress. El plugin para CodeIgniter ayuda a codificar más rápido con autocompletado y con ayuda dinámica y contextual para este framework” (Softpedia, n.d).

Características

- Auto completado de funciones del lenguaje en forma automática.
- HTML / CSS inspector de código: El Inspector de código hace que sea fácil de encontrar elementos HTML y sus estilos en la página.
- Asistente para crear sitios en Drupal que permite indicar la base de datos MySQL, datos para el acceso, seleccionar temas, módulos etc.
- Herramientas para depuración del código.
- Resaltado de sintaxis para el código PHP, Javascript y los demás lenguajes.
- Visualizar las secciones del archivo PHP en modo de árbol para acceder rápidamente a las diferentes secciones. Clases, interfaces, funciones, variables, includes, y los demás contenidos del código como los CSS, HTML etc.
- Vista previa de las páginas en los navegadores.
- Herramientas para convertir de HTML a PHP, PHP a HTML.
- En la pantalla de instalación, permite instalar además plugins o complementos para trabajar en conjunto con:

Wordpress, CodeIgniter, FaceBook, Joomla, jQuery, MySQL entre otros.¹³(Softpedia, n.d)

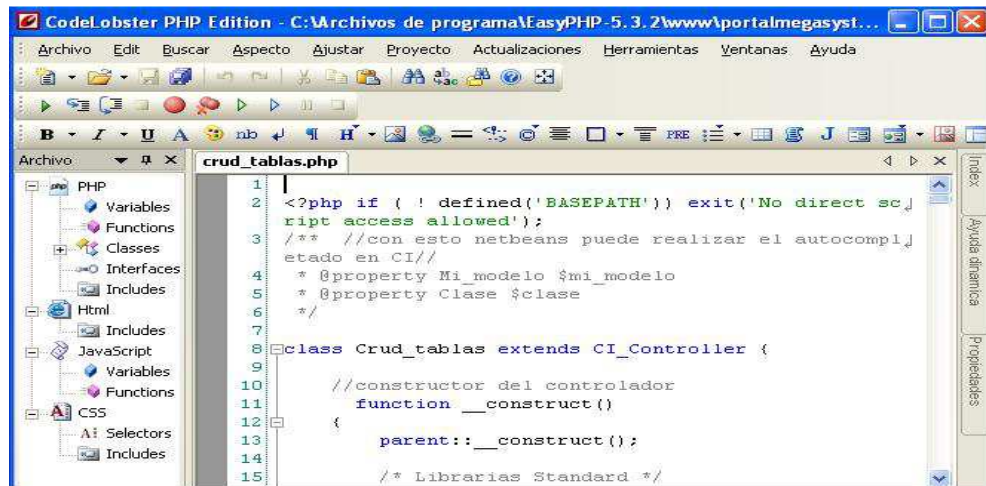


Figura 8: Pantalla del IDE CodeLobster PHP Edition

Fuente: Propia

2.3.3. Gestor de Base de Datos MySQL



Al gestor de base de datos MySQL se lo define como:

“Un sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. MySQL fue creada por la empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca.

¹³ Referencia tomada de <http://www.softpedia.es/programa-CodeLobster-PHP-Edition-76968.html>

Aunque MySQL es software libre, MySQL AB distribuye una versión comercial de MySQL, que no se diferencia de la versión libre más que en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario, ya que de no ser así, se vulneraría la licencia GPL.

Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración” (Pecos, n.d).

Características

- Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- Soporta gran cantidad de tipos de datos para las columnas.
- Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc.).
- Gran portabilidad entre sistemas.
- Soporta hasta 32 índices por tabla.
- Gestión de usuarios y passwords, manteniendo un muy buen nivel de seguridad en los datos¹⁴(Pecos, n.d).

2.3.4. Configurar MySQL en CodeIgniter

La configuración para que CodeIgniter acepte el gestor de bases de datos Mysql o cualquier otro, se lo hace en el archivo **database.php** que se encuentra en la ruta **application/config/database.php**, en este archivo es donde se especifican los valores de conexión de la base de datos tales

¹⁴ Referencia tomada de http://danielpecos.com/docs/mysql_postgres/x57.html

como el nombre de usuario, el password, el nombre de la base de datos, el tipo de gestor de bases de datos, entre otros.

Los valores de configuración se almacenan de la siguiente forma:

```
$db['default']['hostname'] = "localhost";  
$db['default']['username'] = "root";  
$db['default']['password'] = "";  
$db['default']['database'] = "nombre_de_base_de_datos";  
$db['default']['dbdriver'] = "mysql";  
$db['default']['dbprefix'] = "";  
$db['default']['pconnect'] = TRUE;  
$db['default']['db_debug'] = FALSE;  
$db['default']['cache_on'] = FALSE;  
$db['default']['cachedir'] = "";  
$db['default']['char_set'] = "utf8";  
$db['default']['dbcollat'] = "utf8_general_ci";  
$db['default']['swap_pre'] = "";  
$db['default']['autoinit'] = TRUE;  
$db['default']['stricton'] = FALSE;
```

CodeIgniter viene pre configurado para el uso del gestor de bases de datos Mysql, sin embargo se puede trabajar con otros gestores de bases de datos ya que CodeIgniter tiene soporte para la mayoría entre los que se pueden mencionar a los más importantes Postgres y Oracle.

Explicación de Valores

A continuación se realiza una explicación de cada uno de los valores que se almacenan en las variables de configuración del archivo database.php:

- **“hostname.-** El nombre del host del servidor de base de datos. Frecuentemente es "localhost".
- **username.-** El usuario utilizado para conectar con la base de datos.
- **password.-** La contraseña utilizada para conectar con la base de datos.

- **database.-** El nombre de la base de datos con la que se quiere conectar.
- **dbdriver.-** El tipo de base de datos. Por ejemplo: mysql, postgres, odbc, etc. Tiene que especificarse en minúsculas.
- **dbprefix.-** Un prefijo opcional para tablas que se agregará al nombre de las tablas al ejecutar consultas del Active Record. Esto permite que varias instalaciones de CodeIgniter compartan una sola base de datos.
- **pconnect.-TRUE/FALSE** (booleano) - Si se desea usar una conexión persistente.
- **db_debug.-TRUE/FALSE** (booleano) - Si se tienen que mostrar los errores de la base de datos.
- **cache_on.-TRUE/FALSE** (booleano) - Si está habilitado el cacheo de consultas.
- **cachedir.-** La ruta absoluta en el servidor del directorio de cacheo de consultas de la base de datos.
- **char_set.-** El conjunto de caracteres usado en la comunicación con la base de datos.
- **dbcollat.-**La comparación de caracteres usado en la comunicación con la base de datos.
- **swap_pre.-** Un prefijo de tabla por defecto que se tiene que intercambiar con **dbprefix**. Esto es útil para aplicaciones distribuidas donde puede correr consultas escritas manualmente, y se necesita que el prefijo siga siendo personalizable por el usuario.
- **autoinit.-** Si conectar o no automáticamente a la base de datos cuando se carga la biblioteca. Si está establecido a **FALSE**, la conexión tomará lugar antes de ejecutar la primera consulta.
- **stricton.-TRUE/FALSE** (booleano) - Si forzar conexiones en "Modo Estricto", bueno para asegurar el SQL estricto mientras se desarrolla una aplicación.

- **port.-** El número de puerto de base de datos. Para usar este valor tiene que agregar una línea al array de configuración de la base de datos.”(EllisLab, 2008-2011, pág. 113)

Luego de realizada la configuración con los valores necesarios, se debe hacer la conexión a la base de datos, esto se realiza aumentando la librería **database** en el archivo de configuración **autoload.php** que se encuentra en la ruta **system/application/config/autoload.php** de la siguiente forma:

```
$autoload['libraries'] = array('database', 'otras_librerias...');
```

De esta forma se habilita automáticamente la conexión a la base de datos en toda la aplicación cargando la librería database en cada página.

Se puede realizar la conexión de forma manual si se necesita que solo ciertas páginas manejen la base de datos, llamando al procedimiento de carga de librerías dentro de una función de controlador o de modelo, de la forma siguiente:

```
$this->load->database();
```

2.3.5. Librería Grocery CRUD



Grocery CRUD¹⁵ es una librería que sirve para realizar el backend de la aplicación y se la define como:

“Una utilidad open source (Licencia GPL v3 y MIT) con el que hacer la vida de un desarrollador un poco más fácil con apenas unas líneas de código. Con Grocery CRUD y el poder del framework Codeigniter es posible crear en unos minutos interfaces que lleven a cabo las funciones básicas en base de datos (Create/Read/Update/Delete CRUD).

Una vez integrado en la instalación de Codeigniter, a través de unas líneas de código tendremos acceso a interfaces estables y adecuadas para administrar datos.

Esta librería contiene todo lo necesario para añadir, mostrar y editar datos. Incluye además validación de datos tanto en servidor como en cliente.

Con este generador de CRUD, no es necesario copiar todo el código CSS, Javascript, tablas, formularios, funciones, modelos, bibliotecas anuestro sistema de BackOffice”¹⁶(Martínez, 2012).

Para instalar Grocery CRUD en nuestra aplicación, hay que descargarlo de la página oficial www.grocerycrud.com, luego se descomprime el paquete y simplemente se van copiando los archivos de grocery crud dentro de las carpetas correspondientes en la estructura del proyecto en codeigniter y está listo para generar el backend de la aplicación.

¹⁵ CRUD (Create/Read/Update/Delete- Crear/Leer/Modificar/Borrar) en la base de datos.

¹⁶ Referencia tomada de http://www.desarrolloweb.com/de_interes/grocery-crud-interesante-extension-codeigniter-6635.html

CAPÍTULO III

Análisis, Diseño e Implementación del Portal Web de información para la Empresa Megasystem



Análisis, Diseño e Implementación del Portal Web de información para la Empresa Megasystem

- 3.1 Documento de Visión
- 3.2 Especificación de Requisitos
- 3.3 Diseño
- 3.4 Desarrollo e Implementación
- 3.5 Producto

3. Análisis, Diseño e Implementación del Portal Web de información para la Empresa Megasystem

3.1 DOCUMENTO DE VISIÓN

INTRODUCCIÓN.

A continuación se describe una configuración de RUP para este proyecto. Por las características del proyecto, se han incluido artefactos, roles y actividades de la metodología, manteniendo los más esenciales. Dicha configuración está basada en la siguiente selección de artefactos:

DOCUMENTACIÓN TÉCNICA DEL PROYECTO PORTAL WEB DE INFORMACION DE LA EMPRESA MEGASYSTEM

ENTREGABLES

A continuación se presentan y describen cada uno de los entregables. Esta lista se basa en los documentos indicados por RUP, como proceso de desarrollo de software.

Cabe destacar que todos los entregables citados han sido objeto de modificaciones a lo largo del proceso de desarrollo, finalizando con una versión definitiva y completa de cada uno de ellos.

VISIÓN.-

Este documento define la visión del producto desde la perspectiva del cliente, especificando las necesidades y características del producto. Constituye una base de acuerdo a los requisitos del sistema.

3.1.1. Análisis del Problema

El problema de	No poseer un portal web que brinde información a través de internet a los clientes sobre los productos que tiene la empresa Megasystem. Esto hace que se pierdan oportunidades de venta por desconocimiento.
afecta a	Empresa Megasystem, Vendedores de la empresa y a los clientes.
el impacto de este problema es	Social, no existe un adecuado flujo de información hacia los clientes mediante la Web. Los clientes no conocen la existencia de los productos que hay en la empresa lo que genera una pérdida de ventas.
una solución exitosa debería	Implementar una solución informática de calidad soportada por una metodología eficiente de desarrollo de software. Permitir un mejor desarrollo de los procesos relacionados con la presentación de la información de productos en la web. Economizar recursos en publicidad. Hacer conocer al público en general los productos y el contacto que les puede ayudar.

Tabla 1: Análisis del Problema

Fuente: Propia

3.1.2. Declaración del posicionamiento del proyecto

Para	<p>Empresa MEGASYSTEM</p> <p>Gerencia</p> <p>Dpto. de Ventas</p> <p>Clientes</p> <p>Público en general</p>
Quien(es)	<p>El gerente de la empresa MEGASYSTEM necesita conocer la información de las personas que se contactan con la empresa y que quieren ser nuevos clientes, además de saber qué es lo que piensan los clientes sobre la empresa.</p> <p>El Dto. De Ventas debe registrar y actualizar toda la información referente a los productos que se desean mostrar a través de la web.</p> <p>Los clientes necesitan conocer que productos nuevos llegan a la empresa, los precios y las ofertas, también pueden realizar cotizaciones en línea.</p> <p>El Público en general tendrá acceso a información de la empresa MEGASYSTEM, de productos en forma general, además de realizar sugerencias y reclamos.</p>
El (nombre del producto)	Portal Web de Información de Precios y Stock para la Empresa Megasystem
Que	Logra informar a los clientes sobre los productos, el stock y los precios, dependiendo del tipo de cuenta asignados a ellos.
Debido a que	La compañía no posee un portal web que muestre sus productos y servicios a sus clientes y a la comunidad en general.

<p>Nuestro producto</p>	<p>El Portal Web de Información de Precios y Stock de la Empresa Megasystem utiliza el entorno Web como medio de interacción con todos los implicados y público en general.</p> <p>Contendrá Reportes para toma de decisiones como por ejemplo (productos, clientes, cotizaciones, contactos, encuestas).</p> <p>Permitirá presentar páginas Web personalizadas para cada usuario del portal para ingresar, modificar, consultar, eliminar datos referentes a cada rol.</p>
-------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 2: Declaración del posicionamiento del proyecto

Fuente: Propia

3.1.3. Propuesta de Organización del Equipo de Desarrollo y definición de responsabilidades

Participantes en el Proyecto

Al ser un Proyecto de Tesis todos los roles necesarios para el desarrollo de este proyecto le corresponde a Wilson Javier Cevallos Avilés, excepto el control y seguimiento que es responsabilidad del Director del Proyecto, en este caso el Ing. Marco Pusdá.

MEGASYSTEM definirá los participantes del proyecto que proporcionarán los requisitos del sistema, y entre ellos quiénes serán los encargados de evaluar los módulos de acuerdo a cada subsistema si así lo amerita. El equipo de desarrollo interactuará activamente con los participantes de MEGASYSTEM para especificación y validación de los módulos entregados.

Roles y responsabilidades

A continuación se describen los principales roles y responsabilidades de cada uno de los puestos en el equipo de desarrollo durante el Análisis, Diseño, Codificación, Pruebas e Instalación.

Puesto	Responsabilidad
Director del Proyecto	El Director del proyecto se encargará del Control y Seguimiento del Proyecto. Responsable: Ing. Marco PUSDÁ
Analista de Sistemas	Captura, especificación y validación de requisitos, interactuando con el cliente y los usuarios mediante entrevistas. Elaboración del Modelo de Análisis y Diseño. Colaboración en la elaboración de las pruebas funcionales y el modelo de datos. Responsable: Wilson Javier Cevallos Avilés
Programador	Codificación del Sistema Propuesto. Colaboración en la elaboración de las pruebas del funcionamiento del Portal, modelo de datos y en las validaciones con el usuario. Responsable: Wilson Javier Cevallos Avilés
Ingeniero de Software	Gestión de requisitos, gestión de configuración y cambios, elaboración del modelo de datos, preparación de las pruebas del funcionamiento del Portal y elaboración de la documentación. Responsable: Wilson Javier Cevallos Avilés

Tabla 3: Roles y Responsabilidades durante el desarrollo del Proyecto

Fuente: Propia

3.1.4. Resumen de Usuarios

Nombre	Descripción	Stakeholder
Administrador del Portal	Administra toda la información de productos, clientes, vendedores, contactos, encuestas	Lening Morán(Gerente)
Usuarios clientes registrados	Pueden ingresar como clientes identificados a la información de productos, precios y stock, además de realizar cotizaciones en línea.	Clientes registrados
Invitados	Pueden ingresar a la información general de la empresa, los productos y sus características	Clientes no registrados

Tabla 4: Resumen de Usuarios del Portal

Fuente: Propia

3.1.5. Análisis de los Costes del Proyecto

El desarrollo del proyecto está sujeto a posibles cambios y utilización o no de ayuda de terceras personas y conforme se avance en el mismo se agregará o no personal.

Para el correcto uso del Portal Web, la empresa deberá contratar los servicios necesarios como un servidor o alojamiento Web, para realizar la implantación del Portal.

Para llevar a cabo el desarrollo del proyecto es necesario presentar los costes que implica llegar a su etapa de utilización para lo cual se presenta a continuación.

COSTOS		ACTUAL (USD)	REAL (USD)	
HARDWARE	1	COMPUTADORES DE ESCRITORIO	800	800
	1	IMPRESORA	180	180
	1	ALOJAMIENTO WEB	200	200
	SUBTOTAL		1180	1180
SOFTWARE		MySQL 5.5.8	0	0
		CodeIgniter 2.1.0	0	0
		Php5	0	0
		Apache 2.2.17	0	0
		JQuery 1.7.2	0	0
		Netbeans 6.9.1	0	0
		Gimp 2.6.11	0	0
	SUBTOTAL		0	0
CAPACITACIÓN A DESARROLLARSE	Cursos de Capacitación Continua		160	160
PROYECTO	Papelería y suministros de oficina		400	400
	Subtotal		1740	1740
	Imprevistos 10%		174	174
			1914	1914

Tabla 5: Costes del proyecto

Fuente: Propia

3.2. ESPECIFICACIÓN DE REQUISITOS

3.2.1. Flujo de trabajo

El flujo de trabajo es:

“El estudio de los aspectos operacionales de una actividad de trabajo: cómo se estructuran las tareas, cómo se realizan, cuál es su orden correlativo, cómo se sincronizan, cómo fluye la información que soporta las tareas y cómo se le hace seguimiento al cumplimiento de las tareas.

Una aplicación de flujos de trabajo automatiza la secuencia de acciones, actividades o tareas utilizadas para la ejecución del proceso, incluyendo el seguimiento del estado de cada una de sus etapas y la aportación de las herramientas necesarias para gestionarlo”(Wikipedia, 2013).

Para la representación de este modelo se utilizan diagramas de Actividad para modelar los Flujos de Trabajo (workflows) del área problema, y Hojas de Trabajo para analizar los requerimientos.

El aplicativo está enfocado principalmente en la presentación de la información de productos de la Empresa Megasystem a través del portal web. La aplicación Web es aplicada a la automatización de varios procesos, tales como:

- Administración de los contenidos y del portal.
- Presentación de información de productos.

El sitio Web cuenta con 2 módulos:

- Módulo de Administración de todo el sitio Web(Backend)
- Módulo de cliente (Frontend).

Descripción de las características del Sistema.-

Las características del sitio Web en el Módulo de Administración son:

- Ingreso, Modificación, Eliminación de los contenidos del Portal y de usuarios.

Las características del sitio Web en el Módulo del Cliente son:

- Mostrar al usuario de forma organizada la información contenida en el portal.

Descripción de las necesidades de la Empresa Megasystem.-

- Presentar información de los productos a los clientes.

Universidad Técnica del Norte Facultad de Ingeniería en Ciencias Aplicadas IBARRA-ECUADOR
HOJA DE TRABAJO VALIDACIÓN DEL PROYECTO CON EL CLIENTE

Cliente: Empresa Megasystem
Proyecto: DESARROLLO DE UN PORTAL WEB PARA LA EMPRESA MEGASYSTEM.
Tema a tratar: Especificación de requerimientos
Fecha: viernes 27 de abril del 2012

Observaciones:
Se acordó que el proceso sería el siguiente en cuanto a Megasystem: <ul style="list-style-type: none"> • Administración de contenidos y usuarios(ingreso, modificación y eliminación de contenidos y usuarios) • . Presentación organizada de información

Figura 9: Hoja de Trabajo

Fuente: Propia

3.2.2. Diagramas de Casos de Uso

Se puede definir de la siguiente manera:

“un caso de uso es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema. Los diagramas de casos de uso sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y otros sistemas”.(Wikipedia, 2014)

En el siguiente diagrama de casos de uso, se describe claramente el funcionamiento del Portal Web, los actores son el Administrador del Portal y de los contenidos, el Usuario que está registrado y el Invitado no registrado.

El Administrador del portal es el encargado de la presentación del contenido, modificar o eliminar el mismo, usará la generación de reportes para la toma de decisiones y además es el encargado de registrar a los nuevos usuarios del portal para que pueden acceder a los precios personalizados de los productos.

El Usuario es el cliente que está debidamente registrado en la empresa como tal, luego de presentar los requisitos y documentos exigidos. Puede navegar por el contenido de información general, de productos, precios, stock, responder a las encuestas y realizar cotizaciones en línea con los precios según el tipo de cliente que sea considerado.

El Invitado es un tipo de cliente que no está registrado en la empresa y solo se le autoriza a navegar por la información de productos y sus características, puede contactarse con la empresa, también realizar sugerencias, puede registrarse como usuario pero mientras no entregue sus documentos en la empresa solo observará precios de consumidor final.

Diagrama de Casos de Uso General del Portal Web

En el siguiente gráfico se muestran los casos de usos del portal completo en donde se puede ver tres actores, el administrador del portal, el usuario registrado y el usuario no registrado y cómo interactúan con las funciones de cada caso de uso.

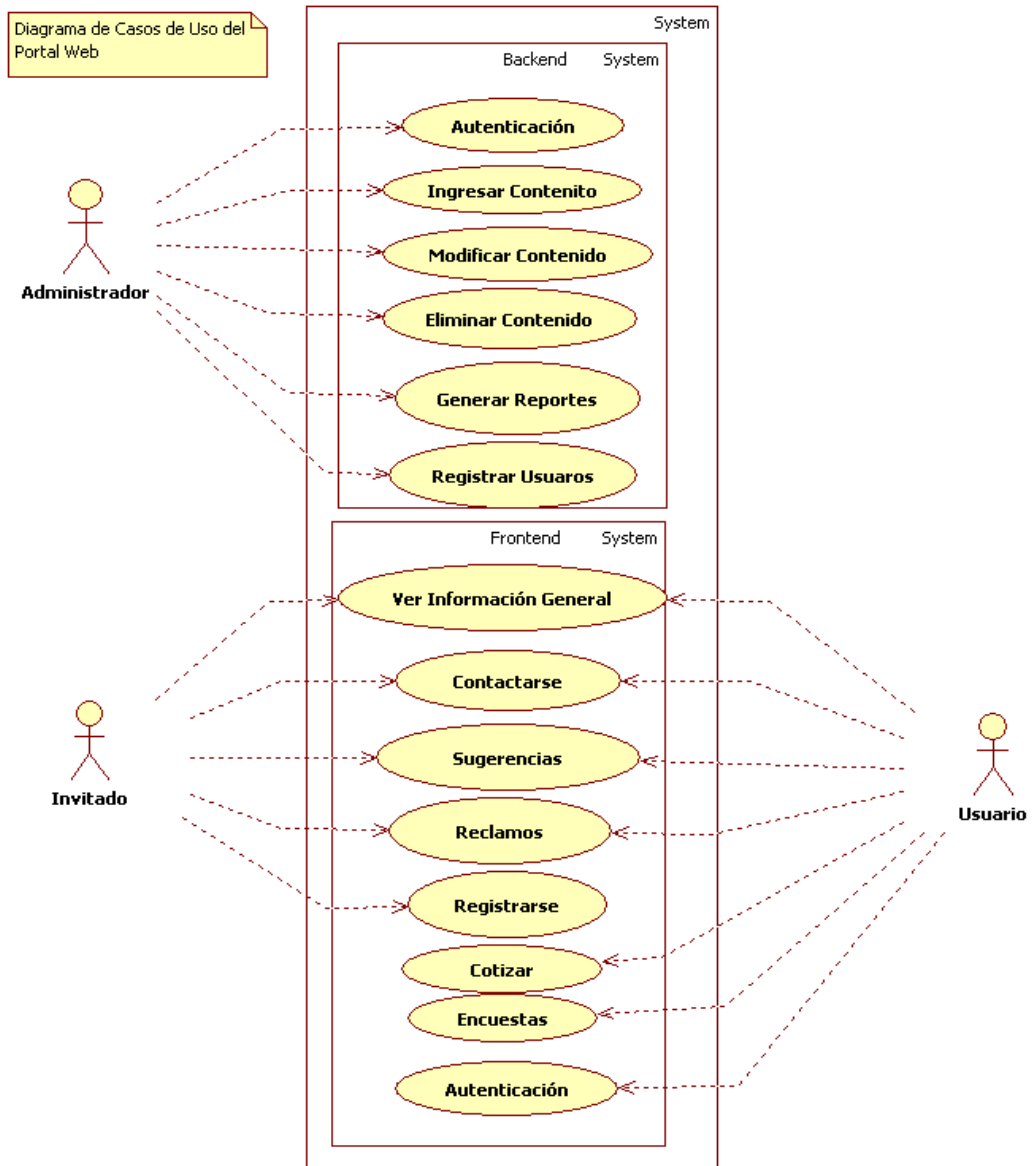


Figura 10: Diagrama de Casos de Uso del Portal

Fuente: Propia

Diagrama de Casos de Uso del Administrador del Portal Web

En este diagrama el actor es el administrador del portal, quien tiene que autenticarse para ingresar al sistema y luego poder realizar las acciones de ingresar, modificar y eliminar contenido y usuarios como se indica en la figura.

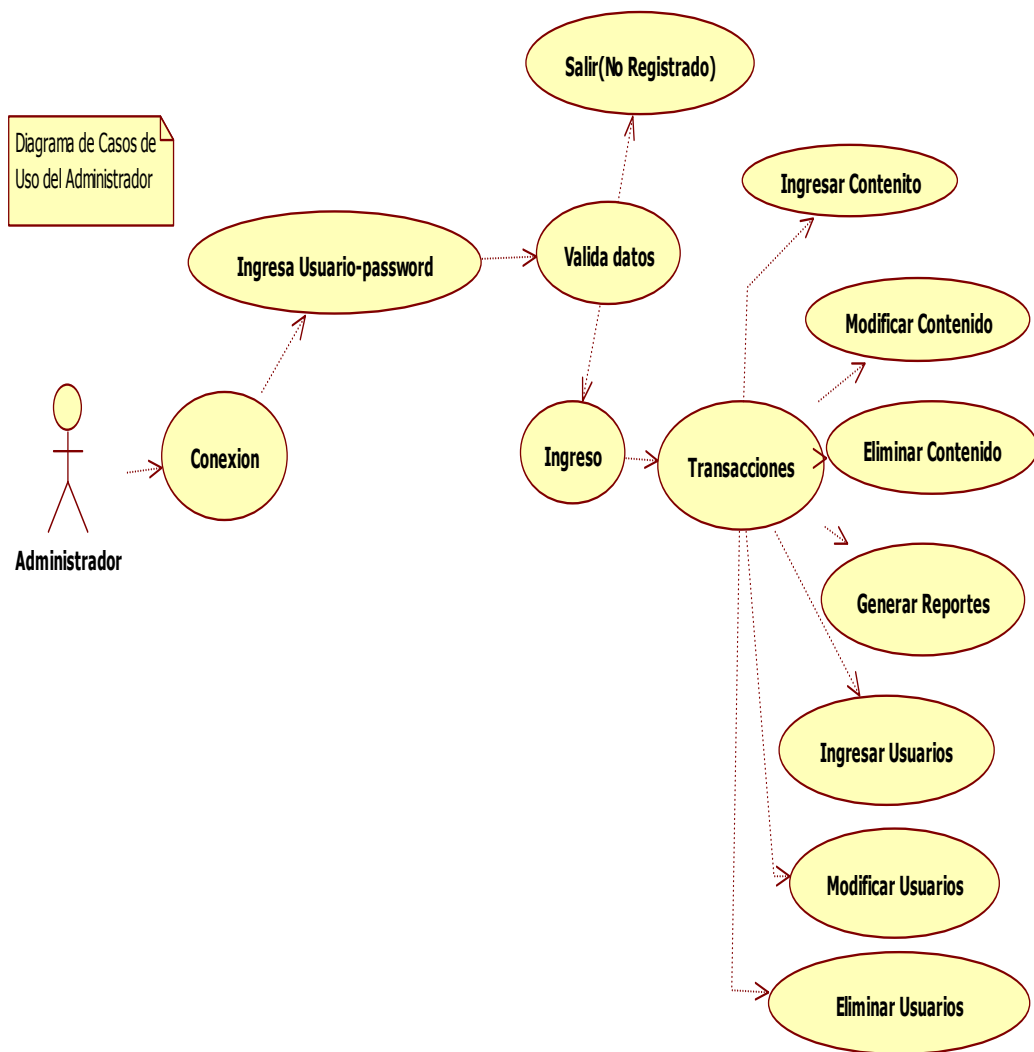


Figura 11 : Diagrama de Casos de Uso del Administrador

Fuente: Propia

Diagrama de Casos de Uso del Usuario Registrado

En este caso el actor usuario registrado puede realizar acciones que no necesitan autenticación y para ingresar a la parte del sistema que muestra información personalizada si tiene que ingresar su login y password como se indica a continuación.

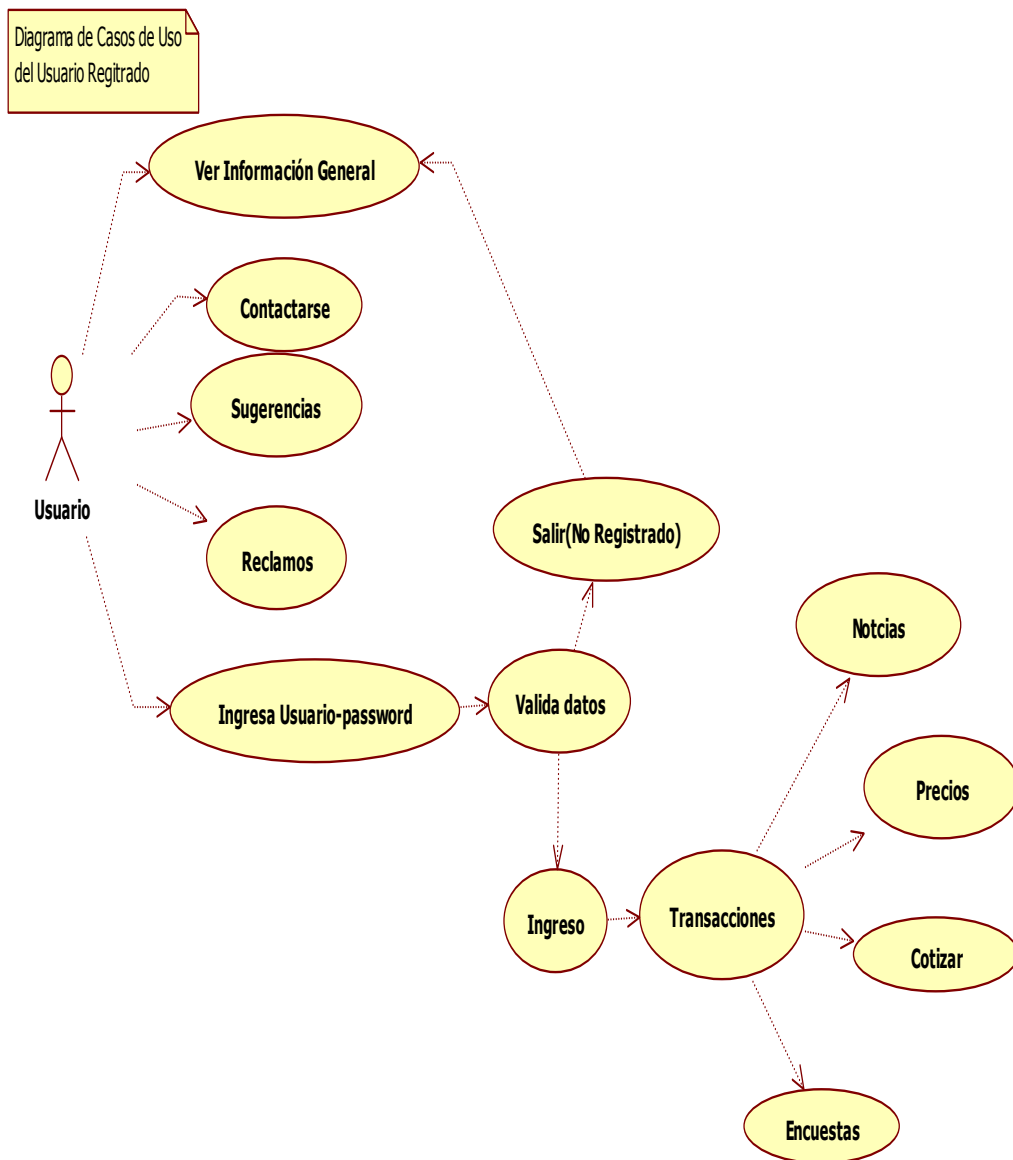


Figura 12: Diagrama de Casos de Uso del Usuario Registrado

Fuente: Propia

Diagrama de Casos de Uso del Invitado no Registrado

El actor en este diagrama es el usuario no registrado o invitado, el podrá realizar acciones generales que no necesitan ningún tipo de autenticación o logeo.

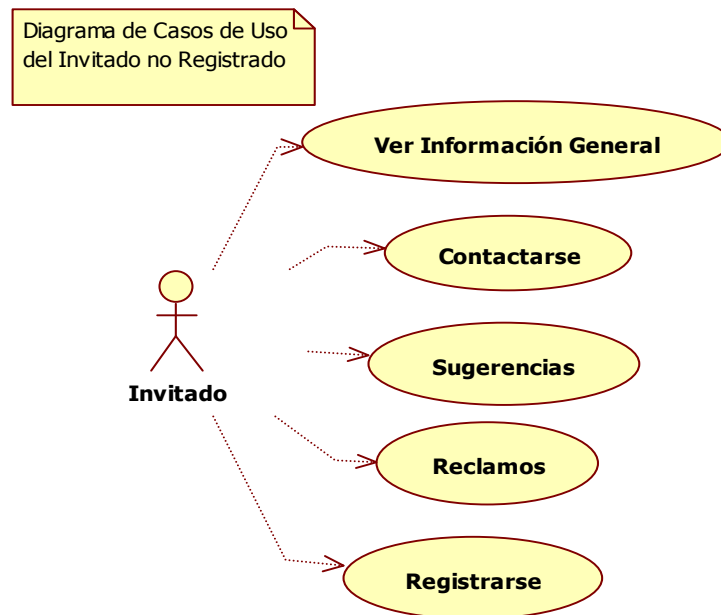


Figura 13: Diagrama de Casos de Uso del Invitado

Fuente: Propia

3.2.3. Diagrama de Clases

El Diagrama de Clases "muestra un conjunto de clases, interfaces y colaboraciones, así como sus relaciones. Los diagramas de clases cubren la vista de diseño estática de un sistema".(PEREZ, 2001)

En el siguiente gráfico se muestra el diagrama de clases del portal, el que contiene las principales clases de modelo y clases de controlador, las que realizan todo el funcionamiento del portal.

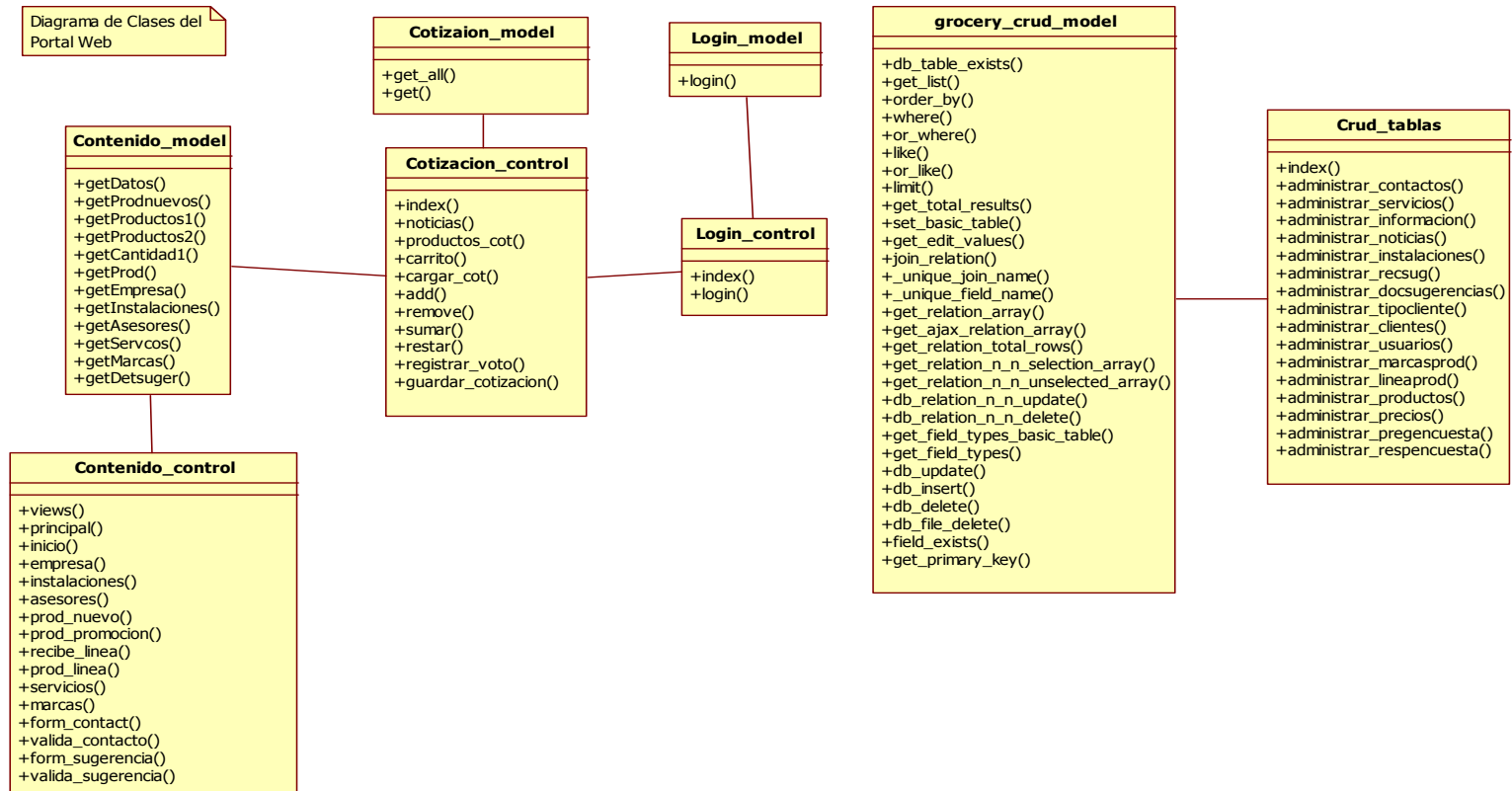


Figura 14: Diagrama de Clases del Portal Web

Fuente: Propia

3.2.4. Diagrama de Secuencias

Un diagrama de secuencia modela la mecánica de interacción entre objetos de una aplicación con base en tiempos y se modela para cada caso de uso.

Diagrama de Secuencia del Portal Web

En este diagrama se muestra como los diferentes usuarios del portal pueden interactuar con el sistema y las secuencias que se siguen desde que se abre la página web, se hace la solicitud de información, se ingresa a la base de datos, hasta que se presenta la información.

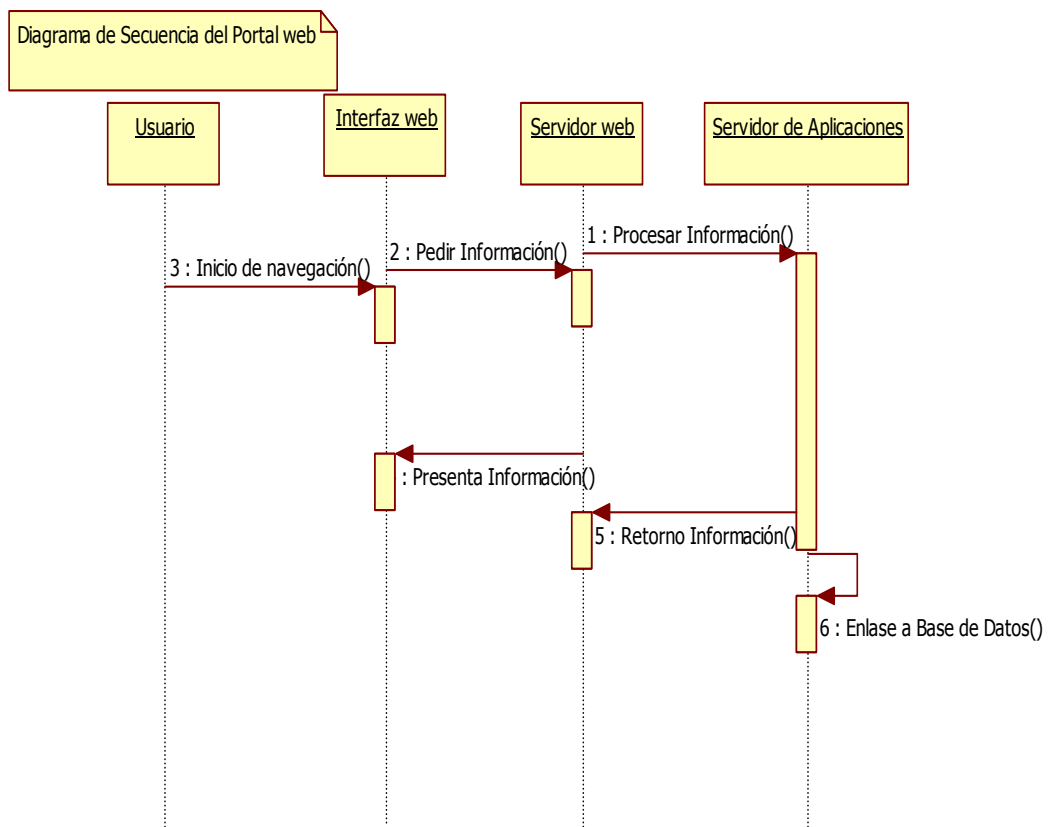


Figura 15: Diagrama de Secuencias del Portal Web

Fuente: Propia

Diagrama de Secuencia del Administrador

En este diagrama se presenta las secuencias que se siguen cuando el actor administrador ingresa al portal, primeramente tiene que logearse y autenticarse, luego puede realizar todos los procesos para administrar la información que se encuentra en la base de datos.

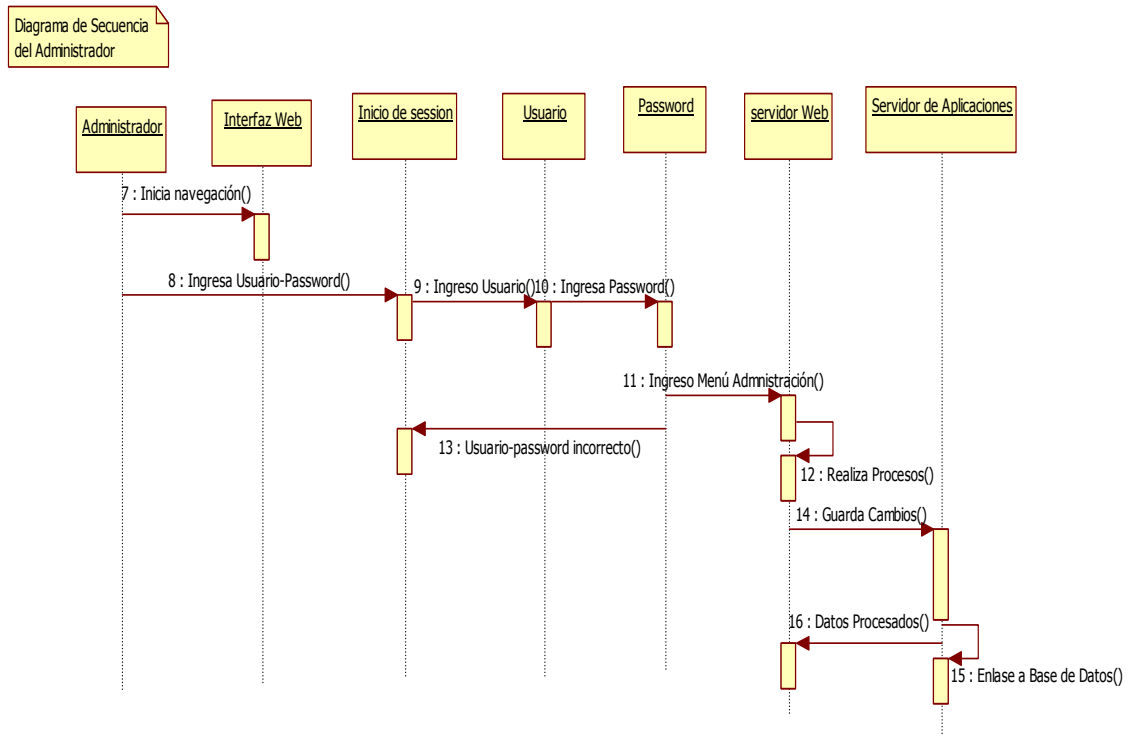


Figura 16: Diagrama de Secuencia del Administrador

Fuente: Propia

A continuación se presenta el diagrama de secuencia del usuario registrado, este usuario puede seguir la secuencia que implica solo pedir información general a la página y los procesos siguientes presentan esa información en pantalla, también puede logearse como parte de las secuencias de usuario registrado, para realizar las secuencias que implican pedir información más detallada y los procesos que se realizan para ingresar a la base de datos y presentar esa información.

Diagrama de Secuencia del Usuario Registrado

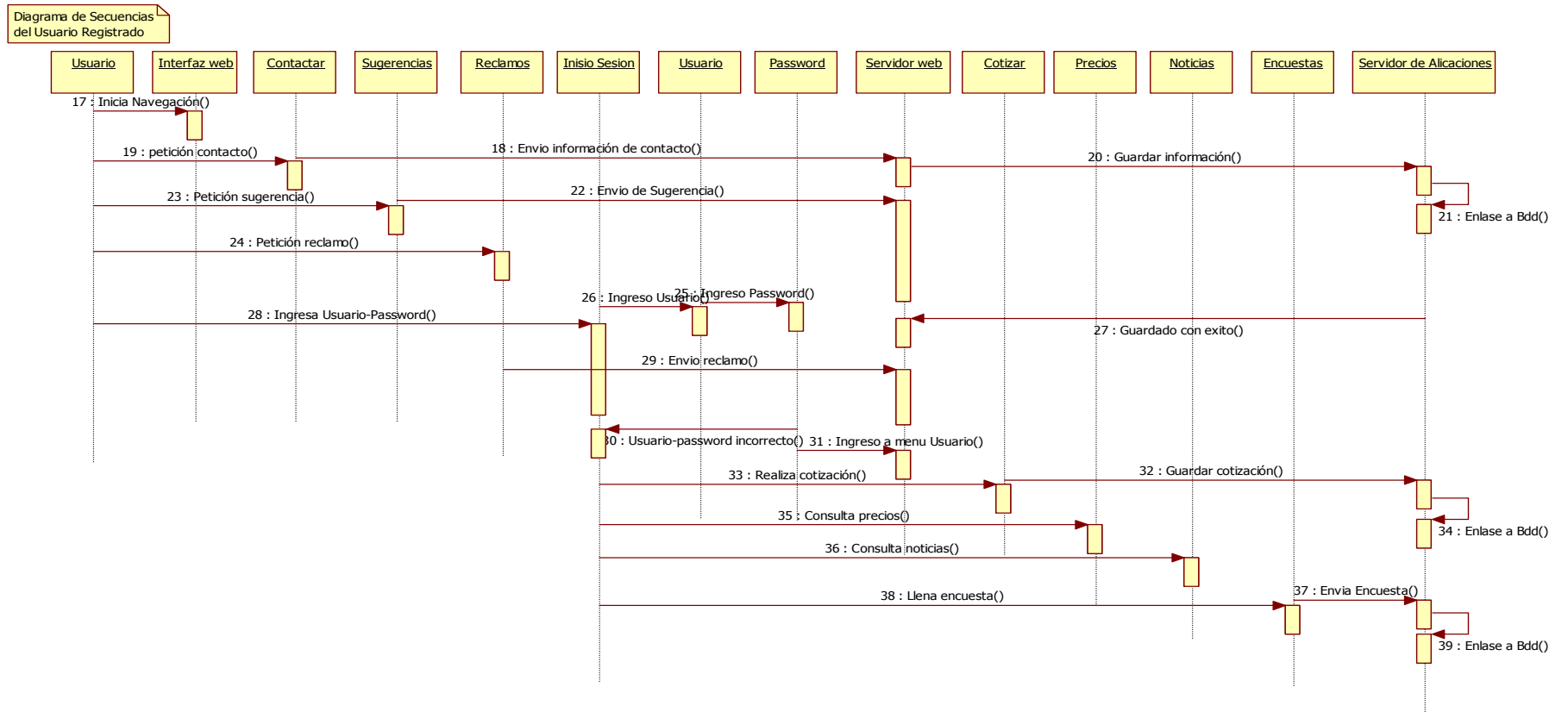


Figura 17: Diagrama de Secuencia del Usuario Registrado

Fuente: Propia

Diagrama de Secuencia del Invitado no Registrado

Cuando ingresa un usuario no registrado o invitado a la página web, hace el pedido solamente de información general, también puede logearse para ser considerado usuario registrado y tener información detallada, las secuencias a seguir son petición de información, presentación de la misma y si se registra guardarlo en la base de datos como cliente.

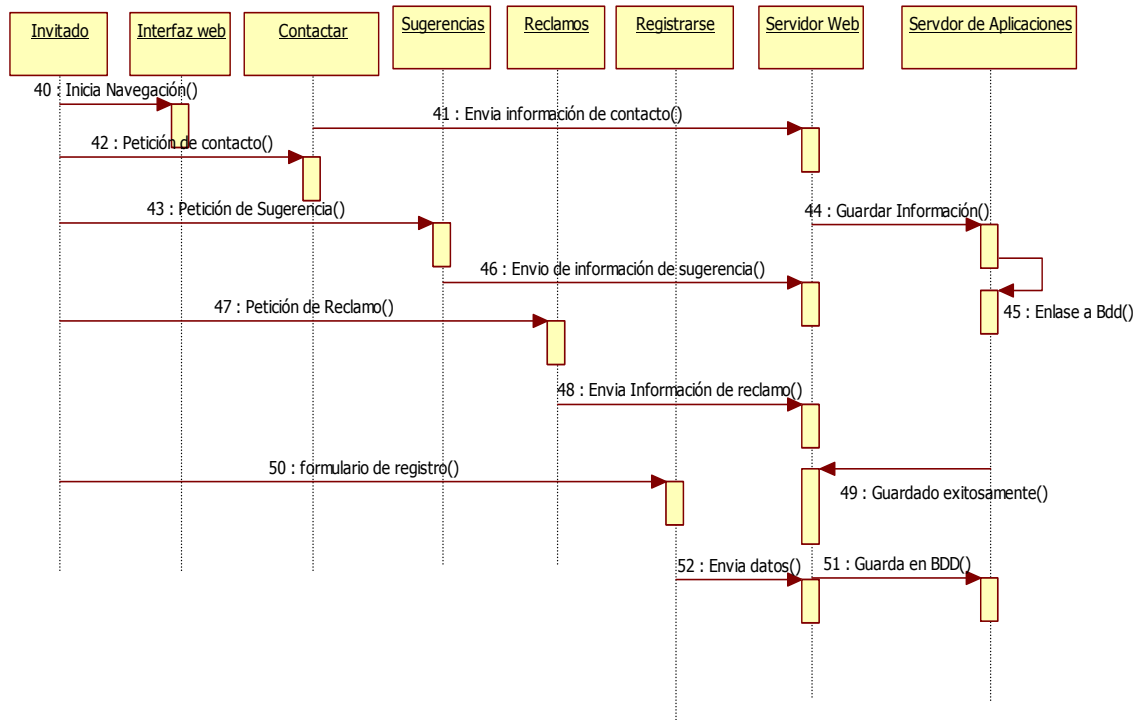


Figura 18: Diagrama de Secuencia del Invitado no Registrado

Fuente: Propia

3.2.5. Diagrama de estados

Los diagramas de estados muestran la secuencia de estados por los que pasa un objeto durante su vida y que se corresponden con los estímulos recibidos, junto con sus respuestas y acciones

Diagrama de Estados del Administrador

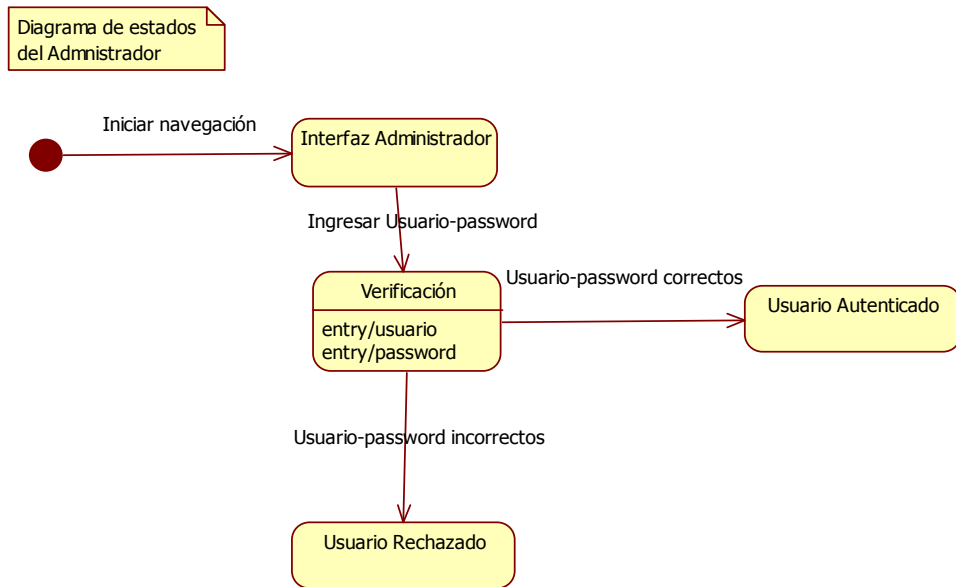


Figura 19: Diagrama de estados del Administrador

Fuente: Propia

Diagrama de Estados del Usuario Registrado

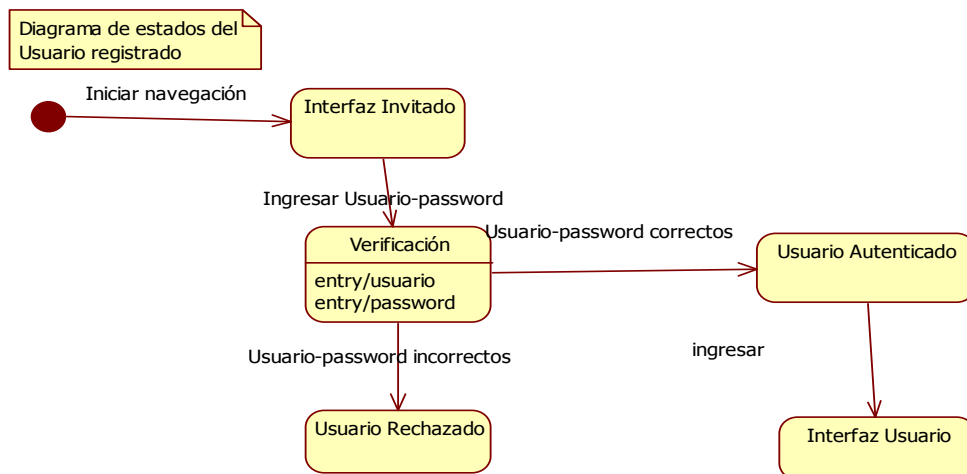


Figura 20: Diagrama de estados del Usuario Registrado

Fuente: Propia

Diagrama de Estados del Invitado no Registrado

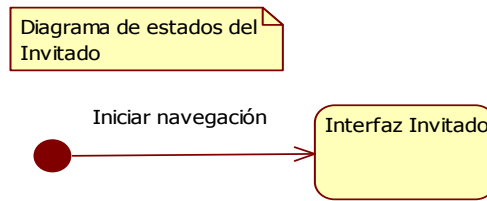


Figura 21: Diagrama de estados del Invitado no registrado

Fuente: Propia

3.2.6. Diagrama de Colaboración

Los diagramas de colaboración muestran la forma en que los objetos colaboran entre sí, tal como sucede con el diagrama de secuencias. Permiten profundizar en el nivel de detalle en los diagramas de Secuencia y expresan las colaboraciones de los objetos en tiempo de ejecución.

Diagrama de Colaboración del Administrador

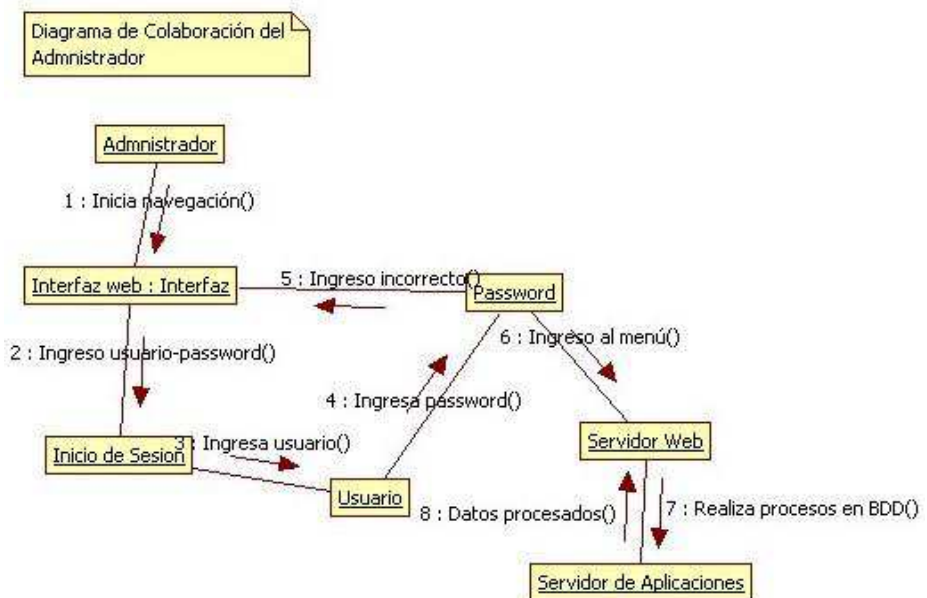


Figura 22: Diagrama de Colaboración del Administrador

Fuente: Propia

Diagrama de Colaboración del Usuario Registrado

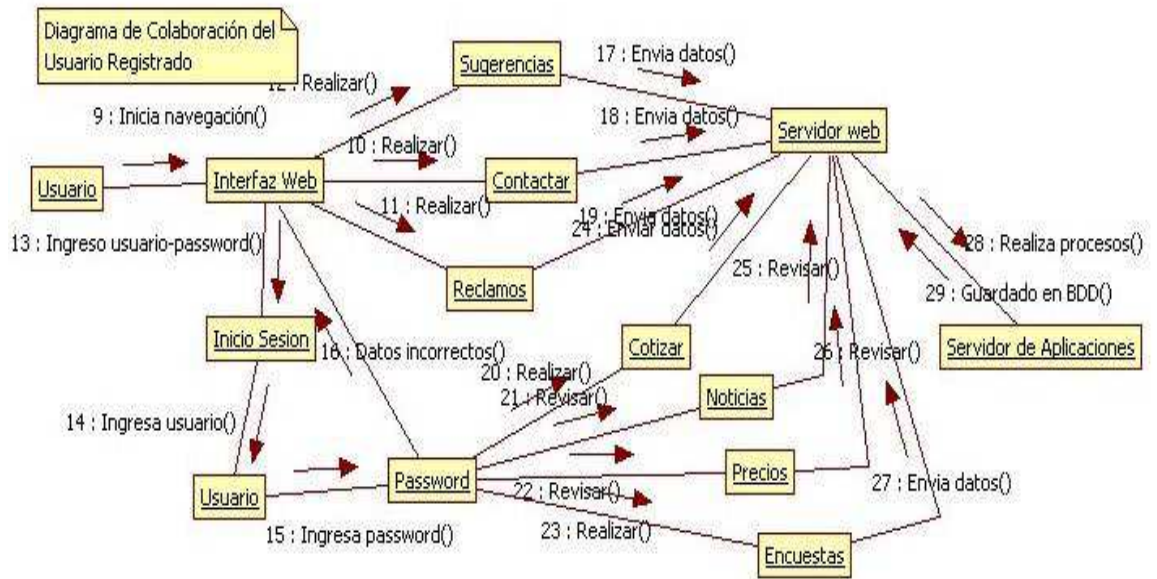


Figura 23: Diagrama de Estados del Usuario Registrado

Fuente: Propia

Diagrama de Colaboración del Invitado no Registrado



Figura 24: Diagrama de Estados del Invitado no Registrado

Fuente: Propia

3.2.7. Diagrama de Distribución

Un diagrama de distribución ilustra la distribución del hardware básico para el diseño del sistema.

Diagrama de Distribución del Portal

Para poner en funcionamiento el proyecto es necesario tener culminada la aplicación del portal web, creada la base de datos y estas deben almacenarse dentro de un servidor web que por medio del internet podrán ser accedidas por los diferentes usuarios.

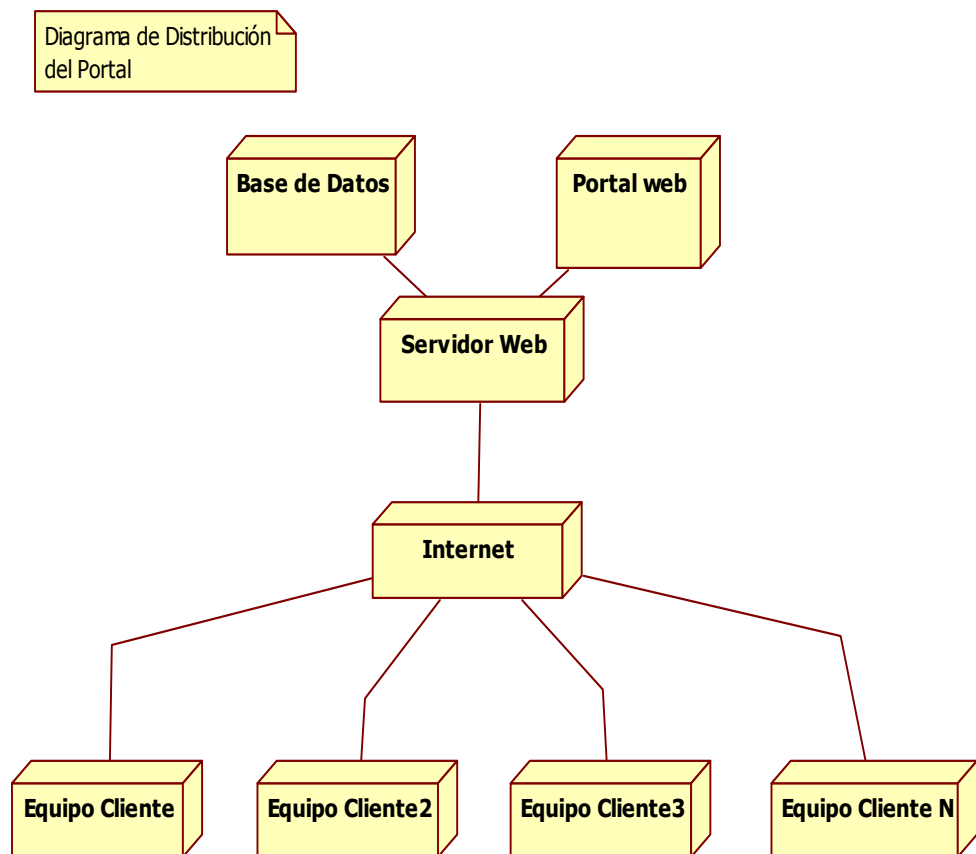


Figura 25: Diagrama de Distribución del Portal

Fuente: Propia

3.3. DISEÑO

Los tipos de usuarios que navegarán en el portal son tres: El Administrador del portal, el Usuario Invitado que son todas las personas que deseen visitar el sitio, pero que no son clientes registrados de la empresa y el Usuario Registrado que es el cliente de la empresa registrado en la base de clientes.

Una vez reconocidos los procesos de cada usuario en los módulos del Portal Web, se establecen los diseños de las pantallas a realizar, que permitan a cada usuario del portal ingresar a sus diferentes menús de manera ordenada utilizando un usuario y password en caso del administrador y del usuario registrado.

A continuación se realizan los prototipos de pantallas de cada usuario utilizando la información que se recuperó en el levantamiento de requerimientos de los módulos Backend y Frontend:

Menú Usuario Administrador del Portal (Backend):

Usuario: Administrador del Portal					
Empresa	Productos	Servicios	Clientes	Interacción	Reportes
Información de Empresa	Líneas	Servicios	Contactos	Tipo de Documento	Sugerencias
Instalaciones	Marcas		Invitados	DocumentoSuger	Contactos
Asesores	Productos		Usuarios	encia	Invitados
Noticias	Precios		Registrados	Encuestas	Usuarios
			Tipo de Cliente	Respuestas	Registrados
				Encuesta	Líneas
			Gráfico Encuesta	Marcas	
					Productos
					Encuestas
					Cotizaciones

Tabla 6: Esquema del menú principal del Usuario Administrador

Fuente: Propia

Este usuario es el que va administrar el contenido de todo el portal y también administrará a los usuarios registrados, tomando en cuenta que administrar se considera ingresar, modificar y eliminar datos, podrá acceder a los diferentes botones:

Empresa: Se administra información concerniente a la empresa

- Información de Empresa.- la información que se quiere presentar sobre la empresa al visitante.
- Instalaciones.- información de las distintas sucursales de la empresa.
- Asesores.- información de los asesores de cuenta que laboran en la empresa.
- Noticias.- las noticias que se quiere hacer conocer al cliente registrado.

Productos: Se administra la información de los productos que se quieren ofertar

- Líneas.- las diferentes líneas de productos que se manejan.
- Marcas.- las marcas variadas de los productos.
- Productos.- información en si de los productos
- Precios.- administra los diferentes precios asignados a cada cliente según el tipo de cliente que sea considerado.

Servicios: Se administra los diferentes servicios que la empresa brinda a los clientes.

Clientes: Se administra la información sobre los diferentes clientes que visitarán el portal.

- Contactos Invitados.- se administra la información enviada por los usuarios no registrados del portal y que quieren hacer contacto con la empresa.

- Usuarios registrados.- se administra a los clientes registrados en la base de clientes de la empresa.
- Tipo de Cliente.- asignación del tipo de cliente a cada cliente registrado

Interacción: Se administra la información que los usuarios ingresan a través del portal para comunicarse con los representantes de la empresa.

- Tipo documento.- administra los diferentes documentos que el usuario quiere hacer llegar a la empresa como sugerencias, reclamos, solicitudes entre otros.
- Documento Sugerencias.- administrar los diferentes tipos de documentos que los usuarios ingresan.
- Encuestas.- se administra las preguntas que se realizarán en las encuestas.
- Respuestas Encuestas.- se administran las diferentes respuestas asignadas a cada encuesta.
- Gráfico Encuesta- se muestra el gráfico estadístico de la encuesta.

Reportes: Generar los reportes de los distintos tipos de información que se administra en formato pdf.

Menú Usuario no Registrado del Portal (Frontend):

Usuario: Invitado no Registrado				
Inicio	Quienes Somos	Productos	Servicios	Marcas
Bienvenida	Empresa Instalaciones Asesores	Nuevos Promociones	Todos los servicios existentes	Todas las marcas existentes
Contáctenos	Tú Cuenta	Categorías		
Formulario de Contacto Sugerencias	Ingreso a tu Cuenta Olvido sus claves Registrarse	Todas las categorías existentes		

Tabla 7: Esquema del menú principal del Usuario invitado

Fuente: Propia

Este tipo de usuario es cualquier persona que desee navegar por el portal, pero solo lo hará a información general, tendrá acceso a los siguientes botones:

Inicio: aparece una bienvenida

Quienes Somos: mira información general de la empresa, de las instalaciones y de los asesores de cuenta.

Productos: verá información referente a los productos en general, los productos nuevos y los de promoción.

Servicios: se muestra información sobre los servicios que la empresa vende a los clientes.

Marcas: catálogos de productos diferenciados por marcas.

Contáctenos: Formularios para contactarse con la empresa y para realizar sugerencias.

Tu Cuenta: se presenta un formulario para ingresar al área de usuarios registrados.

Categorías: catálogos de productos clasificados por categorías o líneas.

Menú Usuario Registrado del Portal (Frontend):

Usuario: Usuario Registrado				
Inicio	Noticias	Quienes Somos	Productos	Servicios
Un menú general	Noticias para los clientes	Empresa Instalaciones Asesores	Nuevos Promociones Cotizaciones	Todos los servicios existentes
Marcas	Soporte	Categorías		
Todas las marcas existentes	Políticas de Garantía Lista de Precios	Todas las categorías existentes		

Tabla 8: Esquema del menú principal del Usuario registrado

Fuente: Propia

Es el usuario que se encuentra registrado en la base de clientes de la empresa y puede ingresar luego de logearse y autenticarse a una sección en donde se le muestra información más detallada de los productos, tiene acceso a los siguientes botones:

Inicio: se presenta un submenú con las principales páginas de este apartado.

Noticias: en este caso se presenta información sobre noticias que la empresa desea que se conozca.

Quienes Somos: mira información general de la empresa, de las instalaciones y de los asesores de cuenta.

Productos: verá información referente a los productos en general, los productos nuevos y los de promoción, además podrá realizar una cotización en línea.

Servicios: se muestra información sobre los servicios que la empresa vende a los clientes.

Marcas: catálogos de productos diferenciados por marcas.

Soporte: se presenta las políticas de garantía y las lista de precios según el tipo de cliente.

Categorías: catálogos de productos clasificados por categorías o líneas.

3.4 DESARROLLO E IMPLEMENTACIÓN

3.4.1 Especificaciones Tecnológicas

La implementación de la aplicación se la realizó con la herramienta Netbeans 6.9.1 con soporte para el lenguaje de programación PHP, el uso del lenguaje de programación PHP 5.1.6, el Framework CodeIgniter 2.1.0 y el gestor de bases de datos MySQL.4.1, herramientas libres para desarrollar el aplicativo de la pagina web de la empresa Megasystem.

3.4.2 Arquitectura de Aplicación

CodeIgniter está basado en el patrón de desarrollo Modelo-Vista-Controlador, en donde se separa la lógica de la aplicación de la presentación, generando código que puede ser reutilizado y evitando la reescritura del mismo, debido a esto, la arquitectura implementada para el desarrollo del portal es la arquitectura MVC.

3.5. PRODUCTO

3.5.1 Instalación

El portal se instala en tres pasos:

- Subir las carpetas y archivos del portal al servidor.
- Abrir el archivo **application/config/config.php** con un editor de texto y establecer la URL base.
- Para configurar la base de datos, abrir el archivo **application/config/database.php** con un editor de texto y establecer los parámetros de la base de datos.

3.5.2 Descripción del Portal Web

Administrativa

Para ver el portal en funcionamiento, en un navegador web ejecutar **<http://www.megasystemecuador.com/>**

Descripción del Portal

Vista Principal (Frontend) del Invitado

Esta es la página que verán todas las personas que accedan al portal y no estén registrados, estos solo verán información general, tienen la opción de registrarse para ver información más específica.



Figura 26: Vista Principal (Frontend) del Invitado

Fuente: Propia

Vista Principal (Frontend) del Usuario Registrado

Esta es la página que verán solo los usuarios registrados luego de logearse en donde podrán ver información de productos y precios.



Figura 27: Vista Principal (Frontend) del Usuario Registrado

Fuente: Propia

Vista Principal (Backend) del Administrador

A esta página ingresará solo el administrador del portal luego de logearse.



Figura 28: Vista Principal (Backend) del Administrador

Fuente: Propia

CAPÍTULO IV

Portal Web de Información de Precios y Stock



Portal Web de Información de Precios y Stock

- 4.1 Diseño de Base de Datos
- 4.2 Manual de Usuario
- 4.3 Manual Técnico

4. Portal Web de Información de Precios y Stock

4.1 Diseño de la base de datos

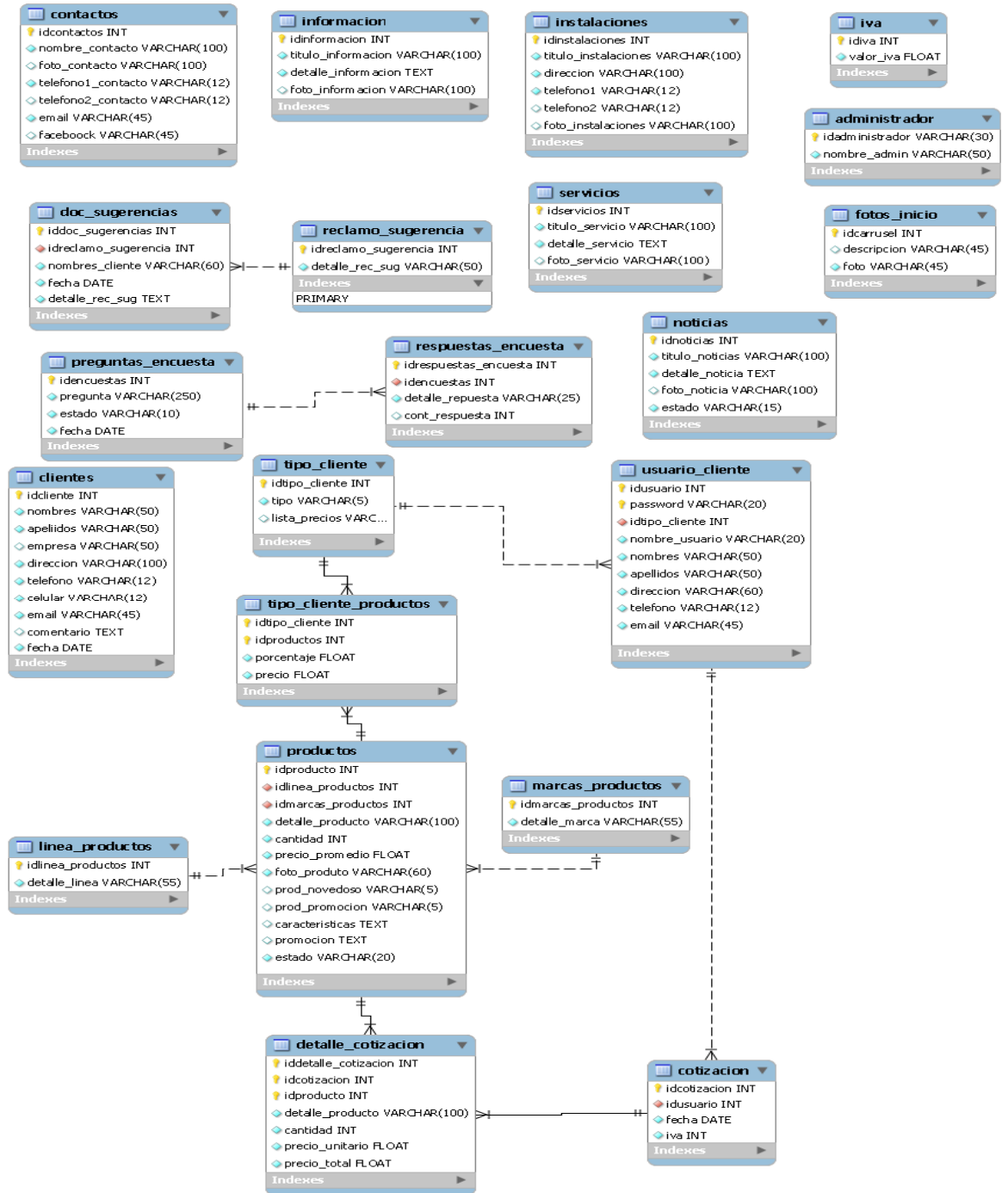


Figura 29: Diseño de la Base de Datos

Fuente: Propia

4.2 Manual de Usuario

Módulo de Administrador (Backend)

Para ingresar al módulo de administrador que es el Backend en donde se puede ingresar, eliminar, actualizar y consultar datos referentes a productos y usuarios, se tiene que ingresar un login y un password para su debida autenticación, colocando en el navegador la siguiente dirección: **www.megasystemecuador.com/crud_tablas**

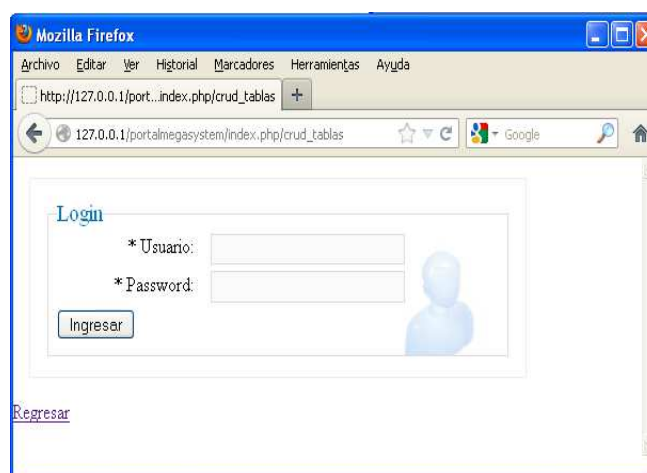


Figura 30: Ingreso como administrador

Fuente: Propia

Una vez ingresados los datos correctos de login y password, el sistema realiza internamente la autenticación y le permite ingresar a la página principal de administración en donde se pueden encontrar los submenús para administrar la información de:

- Empresa
- Productos
- Servicios
- Clientes
- Interacción
- Reportes



Figura 31: Página de administrador

Fuente: Propia

En la opción Empresa se encuentra un submenú para administrar información referente a la empresa: Información de Empresa, Instalaciones, Asesores, Noticias, Fotos Inicio.

Información de Empresa.-administra toda la información referente a la empresa que se quiera que los usuarios conozcan y vean en la página.



Figura 32: Página Información de Empresa

Fuente: Propia

Instalaciones.- se puede administrar información de las instalaciones físicas que posea la empresa.

Titulo instalaciones	Direccion	Telefono1	Telefono2	Foto instalaciones	Acciones
INSTALACIONES IBARRA	VICTOR GOMEZ JURADO 3.44 Y JUAN...	2630894	2630895	068b1-install1.JPG	

Figura 33: Página administración de Instalaciones

Fuente: Propia

Asesores.- administra los datos de los asesores de cuenta o vendedores.

Nombre contacto	Foto contacto	Telefono1	Telefono2	Email	Facebook	Acciones
LUIS VILLARREAL	02848-vendedor1.JPG	0990554781	0990554782	gordito@yahoo.es		
Santiago Tarambis	95fd7-vendedor2.JPG	0990554781	0990457782	santiagotarambis@yahoo.es		

Figura 34: Página administración de Asesores

Fuente: Propia

Noticias.- todas las noticias que la empresa quiere hacer conocer a los clientes registrados se ingresan en este apartado.



Figura 35: Página administración de Noticias

Fuente: Propia

En la opción Productos se encuentra un submenú para administrar información referente a los productos: Líneas, Marcas, Productos, Precios.

Líneas.- se ingresan las líneas de productos que maneja la empresa



Figura 36: Página administración de Líneas de productos

Fuente: Propia

Marcas.- administra las marcas que maneja la empresa.



Figura 37: Página administración de Marcas

Fuente: Propia

Productos.- se ingresa todos los productos que se maneja en stock



Figura 38: Página administración de Productos

Fuente: El Propia

Precios.- administra los precios diferenciados de los productos, según el tipo de cliente.



Idtipo cliente	Idproductos	Porcentaje	Precio	Acciones
B	Monitor LG 18.5"	10	0	 
B	MONITOR SANSUNG 21.5" LED	10	0	 
B	MOUSE GENIUS USB	10	0	 
B	TECLADO KLIPX 050-S	10	0	 
C	Monitor LG 18.5"	7	0	 
C	MONITOR SANSUNG 21.5" LED	7	0	 
C	MOUSE GENIUS USB	7	0	 
C	TECLADO KLIPX 050-S	7	0	 
D	Monitor LG 18.5"	5	0	 
D	MONITOR SANSUNG 21.5" LED	5	0	 

Figura 39: Página administración de Precios

Fuente: Propia

En la opción Servicios se puede administrar información referente a los servicios generales que ofrece la empresa.



Titulo servicio	Detalle servicio	Foto servicio	Acciones
REPARACIÓN DE EQUIPOS	Se repara toda clase de equipo...	1d1ba-tecnico.JPG	 
ELABORACIÓN DE REDES DE INFORMACIÓN	REALIZAMOS LA ELABORACIÓN...	36228-redes.JPG	 
Desarrollo de Software	Realizamos todo tipo de software...		 

Figura 40: Página administración de Servicios

Fuente: Propia

En la opción Clientes se encuentra un submenú para administrar información referente a los Clientes tales como:

- Contactos Invitados
- Usuarios Clientes
- Tipo de Clientes

Contactos Invitados.- son cualquier tipo de persona que se lo considera clientes invitado y que se contacta con la empresa mediante el portal.



Nombres	Apellidos	Empresa	Direccion	Telefono	Celular	Email	Comentario	Fecha	Acciones
JOSE	GALLEGOS	MUNICIPO DE ATUNTAQUI	ATUNTAQUI	062920452	09955447	jose@yahoo.es	Esta es una prueba	01/01/2013	 
ANITA	IPIALES	seguridad imbabura	TOBIAS MENA	062607452	0997292638	segimba@yahoo.es	Otra prueba para ver	01/01/2013	 
ULISES	DE LA CRUZ		CHOTA		0999578488	ulidelacruz@yahoo.es	xxxxxxxxxxxx	01/02/2013	 
andres	perez		los huertos familiares	062920430	0990554781	anfresperez@yahoo.es	una ultima prueba	01/01/2002	 
pepe	morales	eicono	barrio iess	062630894	0990554781	pepe@yahoo.es	para ver en donde se guarda	01/01/2002	 

Figura 41: Página administración de Contactos Invitados

Fuente: Propia

Usuarios Clientes.- administra a los usuarios o clientes de la empresa que constan en la base de datos como clientes registrados, en este módulo se puede cambiar el tipo de cliente que el sistema asignó automáticamente cuando un cliente invitado se registró

Nombre usuario	Nombres	Apellidos	Direccion	Telefono	Email	Empresa	Acciones
wilson	wilson	cevallos	alpachaca	0990514781	wilson_cevallos2@yahoo.es		
anita	anita	ipiales	caranqui	0989297477	anita@yahoo.es		
andres	andres	ayala	los celbos	2915009	andres@yahoo.es		
marcop	marco	pusda	el olivo	0990554781	marco@yaho.es		
mariom	mario	merlo	centro	062920426	mario@yahoo.com		
andreina	andreina	garrido	san antonio	062923458	andreina@hotmail.com		

Figura 42: Página administración de Usuarios Registrados

Fuente: Propia

Tipo Clientes.- administra los tipos de clientes según como se le considere dentro de la empresa.

Tipo	Lista precios	Acciones
A	a99e8-producto.docx	
B	9b521-producto.docx	
C	32b1c-listapreciosd.docx	
D	3848c-listapreciosd.docx	

Figura 43: Página administración de Tipo de Clientes

Fuente: Propia

En la opción Interacción se encuentra un submenú para administrar información referente a la interacción con la que los usuarios se comunican con la empresa por medio del portal como son:

- Tipo Documento
- Documento Sugerencias
- Encuestas
- Respuestas Encuesta

Tipo Documento.- se puede definir qué tipo de documento queremos crear como: sugerencias, reclamos, solicitudes para que sea presentado en la página al cliente.



Figura 44: Página administración de Tipo de Documento

Fuente: Propia

Documento Sugerencias.- administra la información enviada por los clientes en los documentos que se crearon como tipo de documentos, pueden ser sugerencias, reclamos u otros.



Figura 45: Página administración de Documento Sugerencias

Fuente: Propia

Encuestas.- ingreso de encabezado de las encuestas, en otras palabras es la pregunta que se presentará.



Figura 46: Página administración de Encuestas

Fuente: Propia

Respuestas Encuesta.- ingreso de las varias respuestas posibles a cada una de las preguntas de la encuesta.



Figura 47: Página administración de Respuestas de Encuestas

Fuente: Propia

Gráfico Encuesta.- realiza un grafico estadístico de cada una de las preguntas.



Figura 48: Cuadro de Encuestas

Fuente: Propia



Figura 49: Gráfico estadístico de encuesta

Fuente: Propia

En la opción Reportes se puede realizar reportes de la información que se está administrando en el módulo del administrador, se pueden realizar reportes de: Sugerencias, Contactos Invitados, Usuarios, Líneas, Marcas, Productos, Encuestas, Cotizaciones



Figura 50: Página administración de Reportes

Fuente: Propia

Módulo de Usuario (Frontend) para Usuarios no Registrados

En este módulo se presenta la información general de la empresa, se puede navegar por páginas que muestran catálogos de los productos categorizados por líneas y por marcas, permite interactuar con los responsables de la empresa por medio de formularios de solicitudes, de sugerencias y de contacto con la empresa, se puede buscar productos usando el buscador, en esta parte la información de productos no contienen precios. Para poder acceder a este módulo se pone en un navegador web la siguiente dirección www.megasystemecuador.com/



Figura 51: Página Frontend de Usuarios no Registrados

Fuente: Propia

Módulo de Usuario (Frontend) para Usuarios Registrados

Para ingresar a este módulo se debe entrar desde el botón Tu Cuenta, se despliega un submenú en el cual se debe escoger la opción Ingresar a tu

Cuenta, en donde saldrá un formulario para ingresar el login y el password suministrados por la empresa, si no está registrado tiene la opción en el botón Registrarse en donde se despliega un formulario para ingresar sus datos, si no recuerda los datos de login y password, tiene la opción Olvidaste tu Contraseña en donde se ingresa el correo con el que se registró y el sistema envía un correo automáticamente a esa dirección.



Figura 52: Formulario de ingreso a módulo usuarios registrados

Fuente: Propia

Luego de ingresar el login y el password correctos se ingresa al módulo de Usuarios Registrados en la base de datos de clientes. En este módulo se muestra información general de la empresa, noticias relevantes, información de los productos con precios y stock, catálogos de productos categorizados por líneas y por marcas, podrá realizar cotizaciones de los productos subidos a la base de datos, tendrá acceso a la lista de precios general, todos estos según el tipo de cliente, además puede responder a una encuesta.



Figura 53: Página Frontend de Usuarios Registrados

Fuente: Propia

4.3. Manual Técnico

INSTALACIÓN DEL SERVIDOR DE APLICACIONES

Para trabajar con el framework Codeigniter se debe instalar herramientas que contengan un servidor de aplicaciones como Wamp Server, EasyPhp, Xampp, estas herramientas tienen la facilidad de tener incorporado el servidor de aplicaciones Apache, el motor de PHP y el gestor de base de datos Mysql. Se pueden instalar dichas herramientas por separado si es necesario. Para este manual se usa la instalación de EasyPhp.

1.- Ejecutar el instalador, y en la primera pantalla elegir el idioma



Figura 54: Ejecuta Instalador Easyphp

Fuente: Propia

2.- En la pantalla de bienvenida, dar clic en siguiente



Figura 55: Pantalla de Bienvenida Easyphp

Fuente: Propia

3.- Aceptar el acuerdo de licencia y clic en siguiente

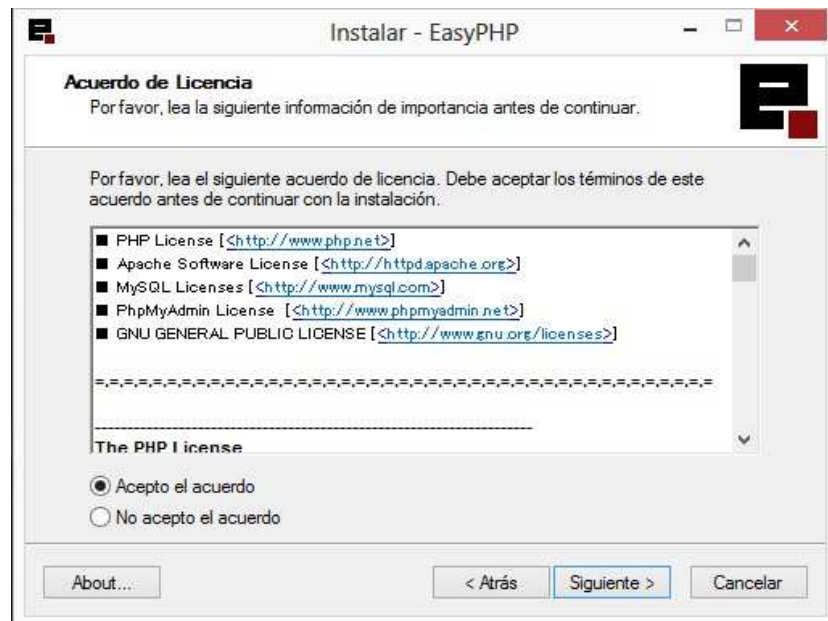


Figura 56: Acuerdo de Licencia Easyphp

Fuente: Propia

4.- Seleccionar la ruta en donde se va a instalar Easyphp y siguiente, dejar la ruta por defecto

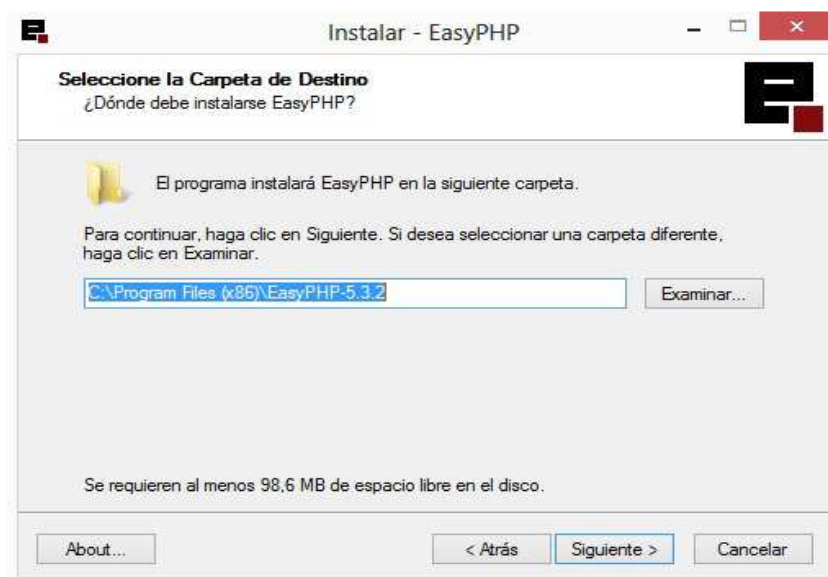


Figura 57: Ruta de instalación

Fuente: Propia

5.- Dejar el nombre de carpeta por defecto o cambiarlo si se desea y dar clic en siguiente

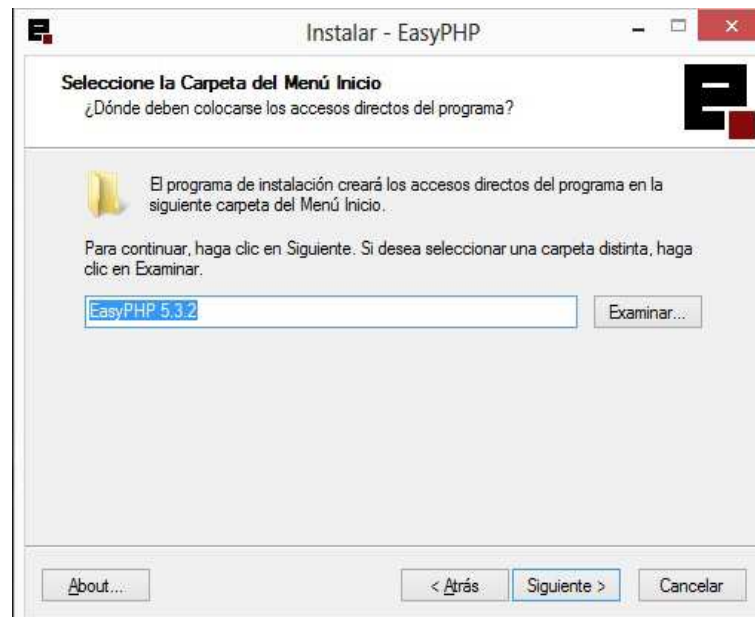


Figura 58: Selecciona Carpeta

Fuente: Propia

6.- Dar clic en instalar

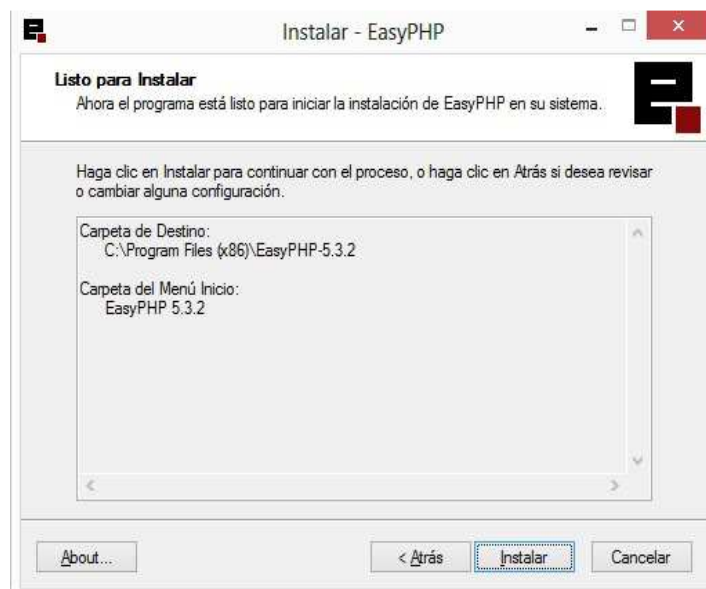


Figura 59: Pantalla de Instalación

Fuente: Propia

7.- Finalizar la instalación



Figura 60: Fin de Instalación

Fuente: Propia

INSTALACIÓN DEL PROYECTO EN EL SERVIDOR DE APLICACIONES

Una vez instalado el servidor de aplicaciones, el proyecto que se llama **portalmegasystem** se va alojar dentro de la carpeta EasyPHP/www

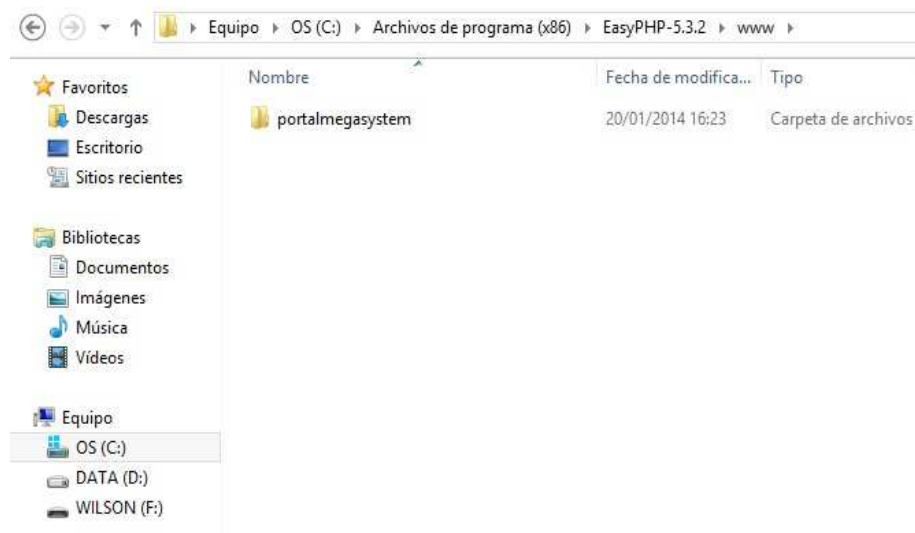


Figura 61: Instalación del Proyecto

Fuente: Propia

El proyecto trabaja con llamadas a la base de datos **bddmegasystem** por lo que hay que abrir PhpMyAdmin y crear la base de datos, crear las tablas e ingresar datos, la estructura de la base de datos se encuentra en los anexos presentados en el CD.



Figura 62: Crear Base de Datos

Fuente: Propia

Una vez que la base de datos está funcional, se debe configurar Codeigniter para que acepte la base de datos, el framework trabaja con los más importantes gestores de bases de datos como Oracle o Postgres, pero para este proyecto se trabajará con el gestor de Mysql, esto se hace en el archivo que se encuentra en **application/config/database.php**, se debe poner el nombre de usuario y password con el que accede a Mysql, también el driver mysql, las otras variables se las deja con los valores por defecto.

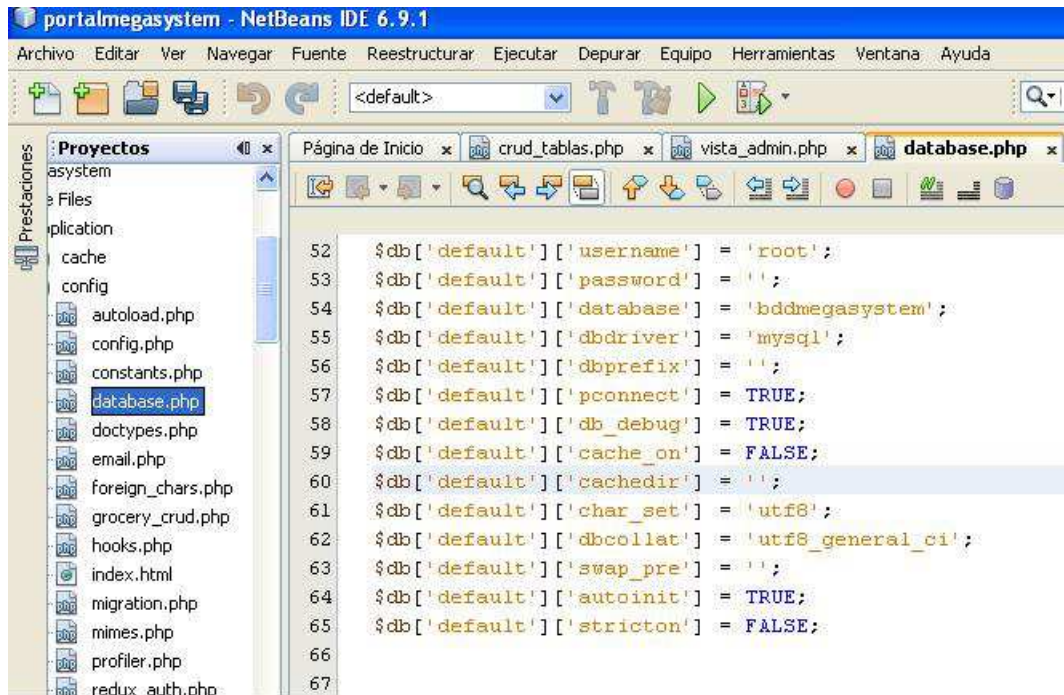


Figura 63: Configurar Base de Datos

Fuente: Propia

Y ya está listo, con los servicios de Apache y Mysql iniciados en Easyphp, poner la siguiente dirección en un navegador web <http://localhost/portalmegasystem/>, y saldrá la página principal del proyecto ejecutándose.



Figura 64: Página de inicio de proyecto

Fuente: Propia

CONTROLADORES

Los controladores son la base de una aplicación desarrollada con CodeIgniter, estos mantienen toda la lógica y hacen las llamadas a las vistas.

El proyecto contiene cuatro controladores crud_tablas, contenido_control, cotizacion_control, login_control.

CONTROLADOR ADMINISTRADOR CRUD_TABLAS (Backend)

Este controlador maneja la administración del módulo de Administrador del portal y contiene las siguientes funciones.

Función index: Función que llama al formulario de login de administrador

Función function _example_output: función privada que genera la llamada a las vistas

Función inicio: llama a la primera página de administrador

Función administrar_contactos: crea el crud¹⁷ de contactos

Función administrar_servicios: crea crud de servicios

Función administrar_informacion: crea crud de información

Función administrar_fotos: crea crud de carrusel para las fotos

Función administrar_noticias: crea crud de noticias

Función administrar_instalaciones: crea crud de instalaciones

Función administrar_recseg: crea crud de reclamo y sugerencias

Función administrar_docsugerencias: crea crud de doc_sugerencias

Función administrar_tipocliente: crea crud de tipo_cliente

Función administrar_clientes: crea crud de clientes

Función administrar_usuarios: crea crud de usuarios_cliente

Función administrar_marcasprod: crea crud de marcas

Función administrar_lineaprod: crea crud de líneas

Función administrar_productos: crea crud de productos

Función administrar_precios: crea crud para administrar de precios de productos

¹⁷ CRUD(Create, Read,Update,Delete)

Función administrar_pregcuesta: crea crud de las preguntas de la encuesta

Función administrar_respencuesta: crea crud de las respuestas a las preguntas de la encuesta

Función administrar_reportes: administra los reportes

Función encuesta_grap: genera tabla con las encuestas realizadas

Función administrar_grafico: llama a la vista que genera el gráfico

Función tarta: genera el gráfico de tarta

Función reporting: genera variosreportes

Función cot_fecha: genera reportes de información por fechas

Función rep_cot: genera el reporte de las cotizaciones

Función rep_sug: reporte de sugerencias

Función rep_continv: reporte de contactos invitados

Función rep_encuesta: reporte de encuestas

CONTROLADOR USUARIO NO REGISTRADO

CONTENIDO_CONTROL (Frontend)

Este controlador maneja toda la información general que verán los usuarios no registrados y tiene las siguientes funciones.

Función principal: genera la página principal con bienvenida

Función búsqueda: realiza búsquedas de productos

Función empresa: muestra la información de la empresa

Función instalaciones: muestra información de las instalaciones

Función asesores: muestra información de los asesores de cuenta

Función productos: muestra lista de todos los productos

Función prod_nuevo: genera lista de productos nuevos

Función prod_promocion: genera lista de productos de promoción

Función recibe_linea: recibe el pedido de generar productos por línea

Función prod_linea: genera productos por línea

Función servicios: información de los servicios de la empresa

Función marcas: muestra lista de productos por marcas

Función form_contact: genera formulario de contacto

Función valida_contacto: valida datos ingresados al formulario de contacto

Función valida_distribuidor: valida información ingresada en formulario para registro como distribuidor

Función form_sugerencia: genera el formulario de reclamos y sugerencia

Función valida_sugerencia: valida datos de formulario sugerencias y reclamos

CONTROLADOR USUARIO REGISTRADO COTIZACION_CONTROL (Frontend)

Este controlador se ejecuta una vez que el usuario ingresa su login y password, muestra al usuario la información de productos con más detalles, tiene las siguientes funciones.

Función inicio: presenta pantalla inicio, contiene submenús

Función noticias: muestra página de las noticias de la empresa

Función productos_cot: presenta página con productos para cotizar

Función cargar_cot: carga nuevamente la cotización luego de ingresar elementos al carro de compras

Función add: adiciona productos a la cotización

Función remove: elimina datos de la cotización

Función sumar: suma elementos de cada ítem al carrito

Función restar: resta elementos de cada ítem al carrito

Función registrar_voto: registra los votos de la encuesta

Función guardar_cotizacion: guarda la cotización en la BDD

Función pgarantia: muestra página de políticas de garantía

Función descargar_politicas: permite descargar las políticas de garantía

Función descargar: presenta la lista de precios

Función descargar_lista: permite descargar la lista de precios

CONTROLADOR DE LOGIN LOGIN_CONTROL

Este controlador realiza la verificación de login, usuario y autenticación para ingresar al modulo de administrador y usuario registrado, contiene las siguientes funciones.

Función index: verifica si ya se logeo anteriormente, si no, abre el formulario para logearse

Función olvido_passw: muestra mensaje de error cuando se olvida las claves de acceso

Función registro: llama al formulario registrarse

Función login: verifica el logeo de usuario y crea una sesión

Función login_admin: verifica el logeo de administrador y crea una sesión

Función cerrar_sesion: elimina la sesión

MODELOS

Los modelos permiten manejar la información que se encuentra en la base de datos y enviarla a los controladores.

MODELO PARA USUARIOS NO REGISTRADOS CONTENIDO_MODEL

Este modelo maneja la información de la base de datos considerada general y que presentan al usuario no registrado.

Función getDatos: saca productos de la tabla y genera la plantilla de vista y la paginación

Función getProd nuevos: saca productos de la base de datos, dependiendo de la consulta, pueden ser productos nuevos, de promoción o todos

Función getProductos: consulta todos los productos

Función getProductos1: consulta productos nuevos

Función getProductos2: consulta productos de promoción

Función getCantidad1: consulta la cantidad de elementos de tabla por productos

Función getProd: genera las páginas completas a ver por el usuario

Función getEmpresa: consulta información de la empresa

Función getInstalaciones: consulta información de las instalaciones

Función getAsesores: consulta información de los asesores

Función getServicios: consulta información de los servicios de la empresa

Función getMarcas: consulta información de las marcas

Función getDetsuger: consulta las sugerencias o reclamos

MODELO PARA USUARIOS REGISTRADOS COTIZACION_MODEL

Este modelo administra la información que se mostrará a los usuarios registrados.

Función get_all: reporte de todos los productos

Función get: reporte de productos por el id

MODELO PARA LOGIN LOGIN_MODEL

Este modelo administra los datos que se necesitan para el proceso de logeo.

Función login: consulta las claves del usuario

Función loginadmin: consulta las claves del administrador

VISTAS

Las vistas presentan la información generada por los controladores, el proyecto maneja tres vistas generales para cada uno de los tipos de usuarios, el administrador, el usuario no registrado y el usuario registrado.

VISTA PARA ADMINISTRADOR VISTA_ADMIN

Esta vista genera la página principal que se presenta al administrador del portal

VISTA PARA USUARIO NO REGISTRADO VISTA_GENERAL

Esta vista presenta la página principal que verán los usuarios no registrados

VISTA PARA USUARIO REGISTRADO VISTA_COTIZAR

Esta vista genera la página principal que verán los usuarios registrados luego de logearse correctamente

VISTA LATERAL IZQUIERDA (SIDEBAR)

Esta plantilla genera la vista del menú lateral izquierdo y es llamado en la vista general de usuarios no registrados.

VISTA LATERAL DERECHA (SIDEBAR2)

Esta vista genera la vista lateral derecha de la plantilla y es llamada en la vista general de usuarios no registrados.

VISTA DE PIE DE PÁGINA

Esta vista genera el pie de página que es llamado en la vista general

VISTA LATERAL IZQUIERDA DE VISTA DE COTIZACION (SIDEBAR_COT)

Esta vista genera el menú lateral que se muestra en la vista de usuarios registrados.

CAPÍTULO V

Validación y Pruebas de Funcionamiento



Validación y Pruebas de Funcionamiento

- 5.1 Prueba de Módulo Clientes
- 5.2 Prueba de Módulo Productos
- 5.3 Prueba de Módulo Información
- 5.4 Prueba de Módulo Servicios
- 5.5 Prueba de Módulo Reportes
- 5.6 Prueba de Módulo Seguridad

5. Validación y Pruebas de Funcionamiento

5.1 Prueba de Módulo Clientes

El módulo clientes maneja a los clientes invitados no registrados y usuarios registrados, además se puede asignar un tipo de cliente dependiendo de cómo la empresa lo califique, los requerimientos para este módulo son: ingreso, modificación y eliminación de clientes.

Las pruebas de requerimientos de este módulo indican que los datos se administran de forma adecuada dependiendo de la estructura de las tablas de usuarios registrados y no registrados.

Las pruebas funcionales de este módulo determinaron que la interfaz del módulo clientes es intuitiva y fácil de usar, el ingreso de los datos está correctamente validado y la información de clientes se presenta de forma entendible en pantalla.

5.2 Prueba de Módulo Productos

Este módulo maneja la información de productos, líneas de productos, marcas, precios de los productos diferenciados para cada tipo de cliente, los requerimientos para este módulo son: ingreso, modificación y eliminación de productos, líneas y marcas.

Las pruebas de requerimientos de este módulo indican que los datos se administran de forma adecuada dependiendo de la estructura de las tablas de productos, líneas y marcas.

Las pruebas funcionales realizadas a este módulo determinaron que se puede usar fácilmente, la interfaz es entendible, la información de

productos, líneas y marcas se ingresan correctamente y validados, la presentación de información está debidamente organizada.

5.3 Prueba de Módulo Información

El módulo de Información maneja todo lo que se refiere a información organizacional de la empresa, información de las instalaciones e información de los asesores de ventas, los requerimientos para este módulo son: ingreso, modificación y eliminación de información organizacional, instalaciones y asesores de venta.

Las pruebas de requerimientos de este módulo indican que los datos se administran de forma adecuada dependiendo de la estructura de las tablas de información, instalaciones y contactos.

Las pruebas funcionales realizadas a este módulo determinaron que la interfaz es intuitiva, tiene facilidad para usar, los datos se ingresan correctamente validados, la información se presenta al usuario de forma ordenada, se debió cambiar el campo que estaba determinado como Messenger por otro como facebook en la tabla contactos (asesores de venta) ya que Messenger esta en desuso.

5.4. Prueba de Módulo Servicios

En este módulo se maneja la información de los servicios que la empresa brinda a los clientes, los requerimientos para este módulo son: ingreso, modificación y eliminación de información de servicios.

Las pruebas de requerimientos de este módulo indican que los datos se administran de forma adecuada dependiendo de la estructura de la tabla servicios.

Las pruebas funcionales realizadas a este módulo indican que se puede usar fácilmente, la interfaz es entendible, la información de servicios ingresa correctamente y validada, la presentación de información de servicios está debidamente organizada.

5.5 Prueba de Módulo Reportes

Este módulo realiza los reportes de la información que se administra en el portal, los requerimientos para este módulo son: presentar la información en reportes.

Las pruebas de requerimientos de este módulo indican que se pueden realizar reportes en formato pdf de la información más relevante que existe en el portal.

Las pruebas funcionales realizadas a este módulo determinaron que la interfaz es intuitiva, es fácil de usar, la información se presenta al administrador de forma ordenada en formato pdf. No se estaba tomando en cuenta a los reportes de las cotizaciones, pero se consideró que sería importante para los administrativos tener ese reporte.

5.6 Prueba de Módulo Seguridad

En este módulo se maneja la seguridad y la autenticación al ingresar al modulo de administrador y al ingresar al módulo de usuarios registrados, se hicieron varias pruebas, se registraron varios errores que fueron corregidos.

Todos estos módulos forman parte un módulo más grande que es el Backend o módulo de administración en donde se realiza toda la gestión de la base de datos, las operaciones que se realizan están validadas y las

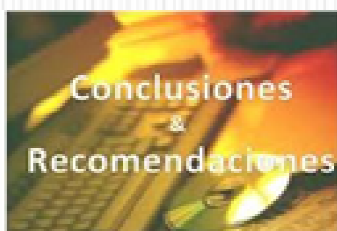
pruebas que se realizaron determinaron que la respuesta a las solicitudes es la esperada.

En el caso de los módulos Frontend de usuario no registrado que a pesar de mostrar información básica, a petición de los usuarios, se creó un formulario de búsqueda con el que se puede buscar información dinámicamente, la cual se probó correctamente, se mejoró el portal añadiendo un formulario para que este tipo de usuario se registre y pueda ingresar a información más detallada, el cual está validado, también se creó un método para recuperación de la clave y el password en caso de que el usuario los olvide y no tenga que llamar a la empresa para recuperarlos.

Para el módulo Frontend de usuario registrado, al que se ingresa luego de digitar correctamente un login y un password los cuales son validados, se accede a información más detallada de los productos y precios, se mejoró la presentación de este módulo para que sea más intuitiva a los usuarios.

CAPÍTULO VI

CONCLUSIONES Y RECOMENDACIONES



CONCLUSIONES Y RECOMENDACIONES

- 6.1 Análisis de Impactos
- 6.2 Conclusiones
- 6.3 Recomendaciones

6.1 Análisis de Impactos

- **En el área social**, el proyecto permite a las personas que normalmente compran en la empresa Megasystem y todas las personas en general tener la información que necesitan sobre los productos en cualquier momento mediante la web, sin tener que llamar a los teléfonos de la empresa o trasladarse a ella.
- **En el área económica**, el proyecto hace conocer al cliente la gama de productos, los productos nuevos que llagan y las promociones vigentes, por lo que genera a la empresa mayores oportunidades de ventas, también se puede gastar menos en marketing y en propaganda ya que se lo hace en línea mediante la página web, los clientes por su parte ahorran en llamadas telefónicas y en recursos para movilizarse hacia la empresa.
- **En el área ambiental**, el proyecto permite realizar cotizaciones en línea y guardarlas en formato pdf, hace igualmente la publicidad en línea, con esto ya no es necesario imprimir las cotizaciones ni la publicidad y por lo tanto genera el uso de menos papel, los clientes usan menos sus vehículos para trasladarse a la empresa.

6.2 Conclusiones

- El framework CodeIgniter tiene una curva de aprendizaje muy baja, lo que le hace una herramienta sencilla de aprender y de usar, con un corto tiempo que se dedique a su estudio, se estará en la capacidad de realizar aplicaciones web completas usando la arquitectura de software Modelo-Vista-Controlador con la que se separa el código de la lógica, los datos y la presentación, con esto se consiguen programas más claros y más fáciles de mantener, además la cantidad de librerías y helpers generan mayor rapidez de desarrollo de los proyectos.

- La implementación del portal web de precios y stock en la empresa Megasystem no ha tenido mayores dificultades considerando que la mayoría de servidores web pueden alojar aplicaciones realizadas en PHP que es el lenguaje que maneja Codeigniter y el gestor de base datos MySQL.
- Utilizar la metodología RUP ayudó a definir y entender las necesidades de los usuarios y definir los requerimientos del portal, al utilizar los artefactos de RUP como son casos de usos, diagramas de clases, diagramas de secuencias entre otros, se pudo entender mejor el análisis y el diseño del portal, es decir que nos permitió construir un sistema más eficiente, más seguro y con menos errores de programación.
- La validación y pruebas realizadas a los módulos del portal web permitieron determinar que el portal se ajusta a los requerimientos planteados inicialmente, que todos los módulos funcionan de acuerdo a lo esperado y que cumple con las expectativas de los diferentes usuarios.
- En general el uso del portal por parte de los clientes de la empresa les permitirá estar mejor informados sobre lo que la empresa Megasystem les puede ofrecer, creando mayores vínculos entre los dos.

6.3 Recomendaciones

- Se recomienda hacer un estudio de otras librerías realizadas por miembros de la comunidad y que no vienen integradas en Codeigniter tales como groceryCRUD que hace el backend de la aplicación, dompdf para convertir las páginas HTML en formato pdf, redux_auth para autenticación de usuarios, entre muchas otras librerías que pueden ser de mucha ayuda en los diferentes casos en que puedan necesitarse.

- Para las futuras versiones del portal se puede crear un módulo que se integre a facebook y con eso dar a conocer a la empresa usando las muy famosas y utilizadas redes sociales utilizando codeigniter.
- Antes de realizar cambios o mejoras en el portal es importante que se revisen los documentos que se obtuvieron utilizando los artefactos de RUP, con esto se entenderá mejor cómo funciona el sistema y será más fácil para el desarrollador hacer las mejoras.
- El portal tiene un módulo de encuestas y otro de sugerencias es aconsejable que el administrador haga preguntas acerca del funcionamiento del portal o del diseño y se indique a los usuarios que hagan sugerencias periódicas sobre el funcionamiento, así el portal estará en un constante ciclo de pruebas.
- Los estudiantes de la Carrera de Ingeniería de Sistemas deben aprovechar esta herramienta para el desarrollo de sus aplicaciones, los desarrolladores web tendrán muchos beneficios al poder entregar las páginas web a sus clientes en tiempos más cortos.
- Para los desarrolladores que usan otros frameworks como Symfony que es uno de los mejores, también deberían conocer Codeigniter, es considerado el hermano menor de Symfony pero tiene una gran potencialidad.

GLOSARIO

Array.- Es una zona de almacenamiento continuo de memoria, que contiene una serie de elementos del mismo tipo.

Cache.- En informática, un caché es un componente que almacena datos para que los futuros requerimientos a esos datos puedan ser servidos más rápidamente.

Cookie.- Es una pequeña información enviada por un sitio web y almacenado en el navegador del usuario, de manera que el sitio web puede consultar la actividad previa del usuario.

Diagrama de Flujo.- Son diagramas que emplean símbolos gráficos para representar los pasos o etapas de un proceso. También permiten describir la secuencia de los distintos pasos o etapas y su interacción.

Framework.- es una estructura de soporte definida, en la cual otro proyecto de software puede ser organizado y desarrollado, es un marco de aplicación o conjunto de bibliotecas orientadas a la reutilización a gran escala de componentes software para el desarrollo rápido de aplicaciones.

Helpers.- es simplemente una colección de funciones en una categoría particular, que ayudan a los desarrolladores con las tareas comunes de programación.

Http-Hipertexto Transfer Protocolo (Protocolo de transferencia de hipertexto) es el método más común de intercambio de información en la world wide web, el método mediante el cual se transfieren las páginas web a un ordenador.

Librerías.-en CodeIgniter son clases de programación orientada a objetos (POO) preparadas para realizar tareas típicas en el desarrollo de páginas web.

Plugins.- Un plugin es un programa o aplicación que añade funcionalidad al programa principal donde está hospedado.

Ruteo.- La tarea de determinar el camino que deben tomar los paquetes para llegar al punto de destino es lo que se conoce como ruteo (o encaminamiento).

Script.- Los scripts son un conjunto de instrucciones generalmente almacenadas en un archivo de texto que deben ser interpretados línea a línea en tiempo real para su ejecución.

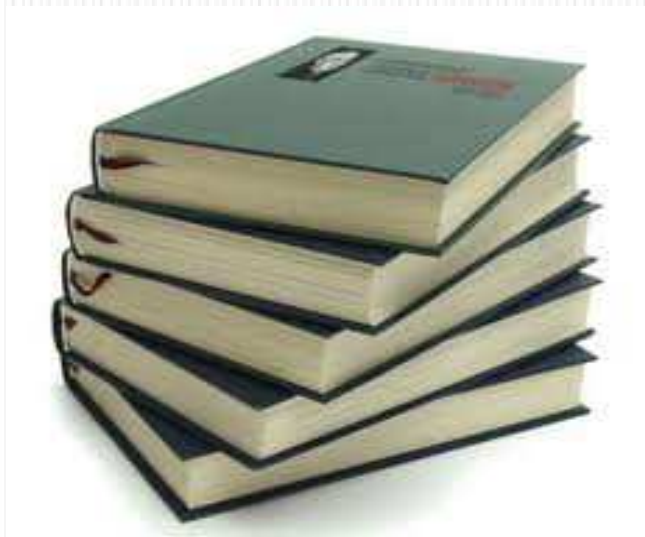
Uri.- Uniform Resource Identifiers (Identificador de Recurso Uniforme), son cadenas que funcionan como identificadores globales que hacen referencia a recursos en la Web tales como documentos, imágenes, archivos descargables, servicios, buzones de correo electrónico y otros.

Url.- Uniform Resource Locator (Localizador Uniforme de Recursos), es una dirección que permite acceder a un archivo o recurso como páginas .html, .php, .asp, o archivos gif, jpg, etc. Se trata de una cadena de caracteres que identifica cada recurso disponible en la Web.

Xss.- Cross-site scripting (Secuencias de comandos en sitios cruzados), es un tipo de inseguridad informática o agujero de seguridad típico de las aplicaciones Web, que permite inyectar en páginas web vistas por el usuario código JavaScript o en otro lenguaje de script similar.

BIBLIOGRAFÍA

BIBLIOGRAFÍA



Bibliografía

LIBROS

CAPÍTULO II

[LIB.001]

Cabezas, L. (2004). Manual Imprescindible de PHP5. Madrid: Artes Gráficas Guemo, S.L

[LIB.002]

Myer, T. (2008). *Professional CodeIgniter*. Indianapolis: Wiley Publishing, Inc.

[LIB.003]

Pavón, J. (2007). PHP. En J. P. Puertas, *Creación de un Portal con PHP y MySQL* (págs. 14-15). Mexico: Alfaomega Grupo Editor, S.A. de C.V.

[LIB.004]

Upton, D. (2007). *CodeIgniter for Rapid PHP Application Development*. Birmingham: Packt Publishing Ltd.

DOCUMENTOS DE SITIO WEB

CAPÍTULO II

[WWW.001]

Alvarez, M. A. (n.d de n.d de n.d). *Manual de Codeigniter*. Recuperado el 21 de Agosto de 2012, de [www.desarrolloweb.com](http://www.desarrolloweb.com/manuales/manual-codeigniter.html):
<http://www.desarrolloweb.com/manuales/manual-codeigniter.html>

[WWW.002]

EllisLab, I. (n.d de n.d de 2008-2011). *Guia de Usuario de Codeigniter Versión 2.1.0*. Recuperado el 25 de Julio de 2012, de Codeigniter:
www.codeigniter.com (F. Velo., Trad) Recuperado de
http://areadepruebas.com.ar/downloads/CodeIgniter_Guia_Usuario_2.1.0.pdf

SITIOS WEB

CAPÍTULO II

[WWW.003]

GoDaddy Dominios. (s.f.). *Diferencias entre paginas web y portales*. Recuperado el 05 de 08 de 2013, de Diferencias entre paginas web y portales:
<http://www.grypus.com/design/internet/49-redes-sociales/79-diferencias-entre-paginas-web-y-portales>

[WWW.004]

Martínez, A. (05 de Marzo de 2012). *Grocery CRUD: interesante extensión de CodeIgniter*. Recuperado el 25 de Julio de 2012, de desarrolloweb.com:
http://www.desarrolloweb.com/de_interes/grocery-crud-interesante-extension-codeigniter-6635.html

[WWW.005]

Netbeans.org. (n.d de n.d de n.d). *Bienvenido a Netbeans*. Recuperado el 15 de enero de 2013, de Bienvenido a Netbeans: https://netbeans.org/index_es.html

[WWW.006]

Netbeans.org. (n.d de n.d de n.d). *Netbeans Platform*. Recuperado el 15 de enero de 2012, de Netbeans Platform:
<https://netbeans.org/features/platform/index.html>

[WWW.007]

Pecos, D. (n.d de n.d de n.d). *PostGreSQL vs. MySQL*. Recuperado el 14 de septiembre de 2012, de PostGreSQL vs. MySQL:
http://danielpecos.com/docs/mysql_postgres/x57.html

[WWW.008]

Softpedia. (n.d de n.d de n.d). *Codelobster PHP*. Recuperado el 6 de febrero de 2013, de Codelobster PHP: <http://www.softpedia.es/programa-CodeLobster-PHP-Edition-76968.html>

CAPÍTULO III

[WWW.009]

PEREZ, A. (2001). *Diagramas de clases*. Recuperado el 15 de 09 de 2013, de Diagramas de clases: <http://www.monografias.com/trabajos88/diagramas-clases/diagramas-clases.shtml>

[WWW.010]

Wikipedia. (05 de 01 de 2014). *Caso de uso*. Recuperado el 15 de 01 de 2014, de Caso de uso: http://es.wikipedia.org/wiki/Caso_de_uso

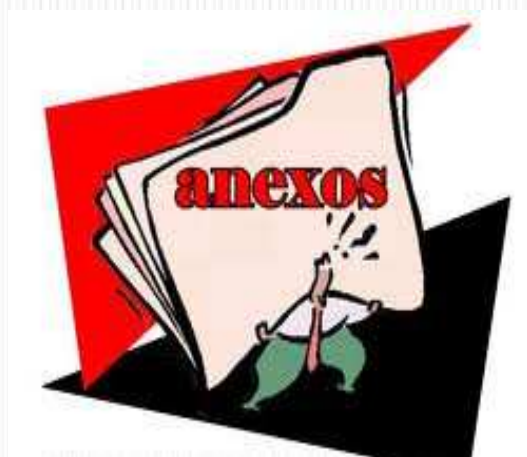
[WWW.011]

Wikipedia. (03 de 12 de 2013). *Flujo de trabajo*. Recuperado el 28 de 12 de 2013, de Flujo de trabajo: http://es.wikipedia.org/wiki/Flujo_de_trabajo

.

ANEXOS

ANEXOS



ANTEPROYECTO DE TESIS
LISTA DE HELPERS DE CODEIGNITER
LISTA DE LIBRERIAS DE CODEIGNITER

ANEXO A

ANTEPROYECTO DE TESIS



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CARRERA DE INGENIERÍA DE SISTEMAS COMPUTACIONALES

PLAN DEL PROYECTO DE TITULACIÓN

Propuesto por: Wilson Javier Cevallos Avilés	Áreas Técnicas del Tema: Innovación Tecnológica y en Productos
Director sugerido: Ing. Marco Pusdá	Fecha: 10 de Mayo del 2012

1. TEMA: ESTUDIO DE LA TECNOLOGÍA CODEIGNITER APLICADA AL DESARROLLO DE PORTALES WEB CON UNA ARQUITECTURA MVC

2. PROBLEMA:

Las personas que se dedican al desarrollo de software y en especial de portales y páginas web, por lo general utilizan tecnologías como php, jsp o .net para el desarrollo de sus aplicaciones, también suelen utilizar frameworks de desarrollo como Symfony, Zendframework, Ruby on Rails o gestores de contenido como Drupal, Joomla entre otros.

Los frameworks que se utilizan con regularidad para el desarrollo rápido de aplicaciones web, tienen características muy potentes, sin embargo poseen una curva de aprendizaje muy elevada, ya que se debe dedicar un buen tiempo a su estudio antes de poder utilizarlos, lo que retrasa el desarrollo y la culminación del proyecto.

Además puede ser que las herramientas no sean utilizadas en su real capacidad o en el peor de los casos que se abandonen los proyectos por no poder acoplarse al framework.

En consecuencia los frameworks ayudan al desarrollo rápido pero en su mayoría tienen una curva elevada de aprendizaje generando una inversión de tiempo adicional a la prevista que puede provocar el retraso o abandono del proyecto.



3. OBJETIVOS:

3.1. GENERALES: Estudiar la tecnología Codeigniter aplicada al desarrollo de portales web con arquitectura MVC.

3.2. ESPECÍFICOS:

- Estudiar el funcionamiento del framework Codeigniter y la tecnología MVC (Modelo-Vista-Controlador).
- Implementar el portal web de información de precios y stock para la empresa Megasystem.
- Documentar el funcionamiento del portal mediante herramientas RUP.
- Realizar las pruebas del correcto funcionamiento del portal.

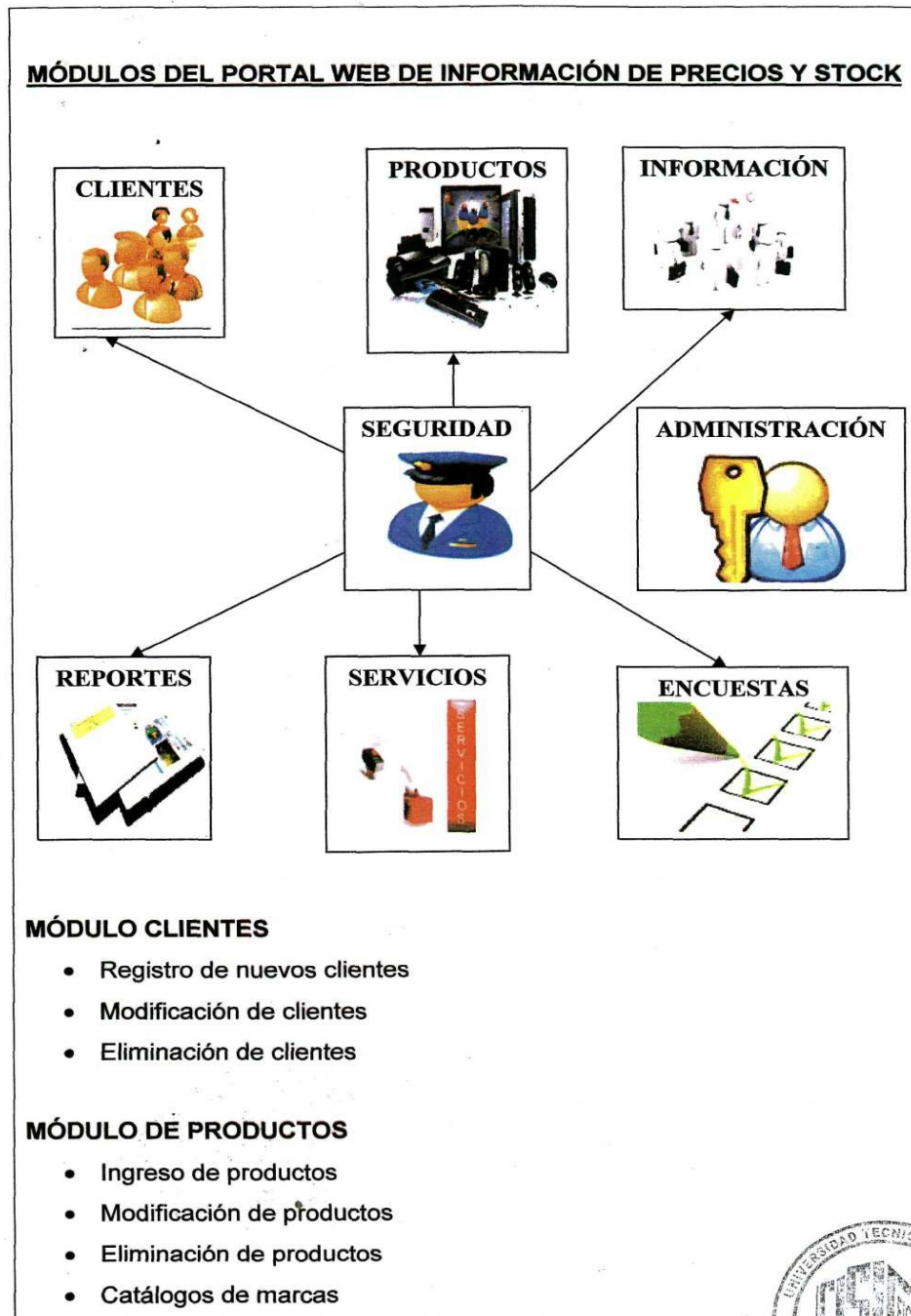
4. ALCANCE:**HERRAMIENTAS TECNOLÓGICAS**

Se estudiará a la herramienta Codeigniter con su arquitectura MVC en todo lo que tiene que ver a la realización de un portal web y su integración con el gestor de bases de datos MySQL.

MÓDULOS DEL PORTAL

EL portal web de información de precios y stock contendrá los siguientes módulos tentativos.





- Catálogo de productos por líneas
- Catálogo de productos novedosos
- Promociones de temporada

MÓDULO DE INFORMACIÓN

- Presentación
- Misión
- Visión
- Valores
- La Empresa-Instalaciones
- Información de contactos

MÓDULO DE SERVICIOS

- Servicios generales
- Realización de cotizaciones
- Formulario de Sugerencias
- Formulario de reclamos
- Formulario de contacto

MÓDULO DE REPORTES

- Reporte de clientes
- Reporte de productos
- Reporte de encuestas
- Reporte de cotizaciones
- Listas de precios

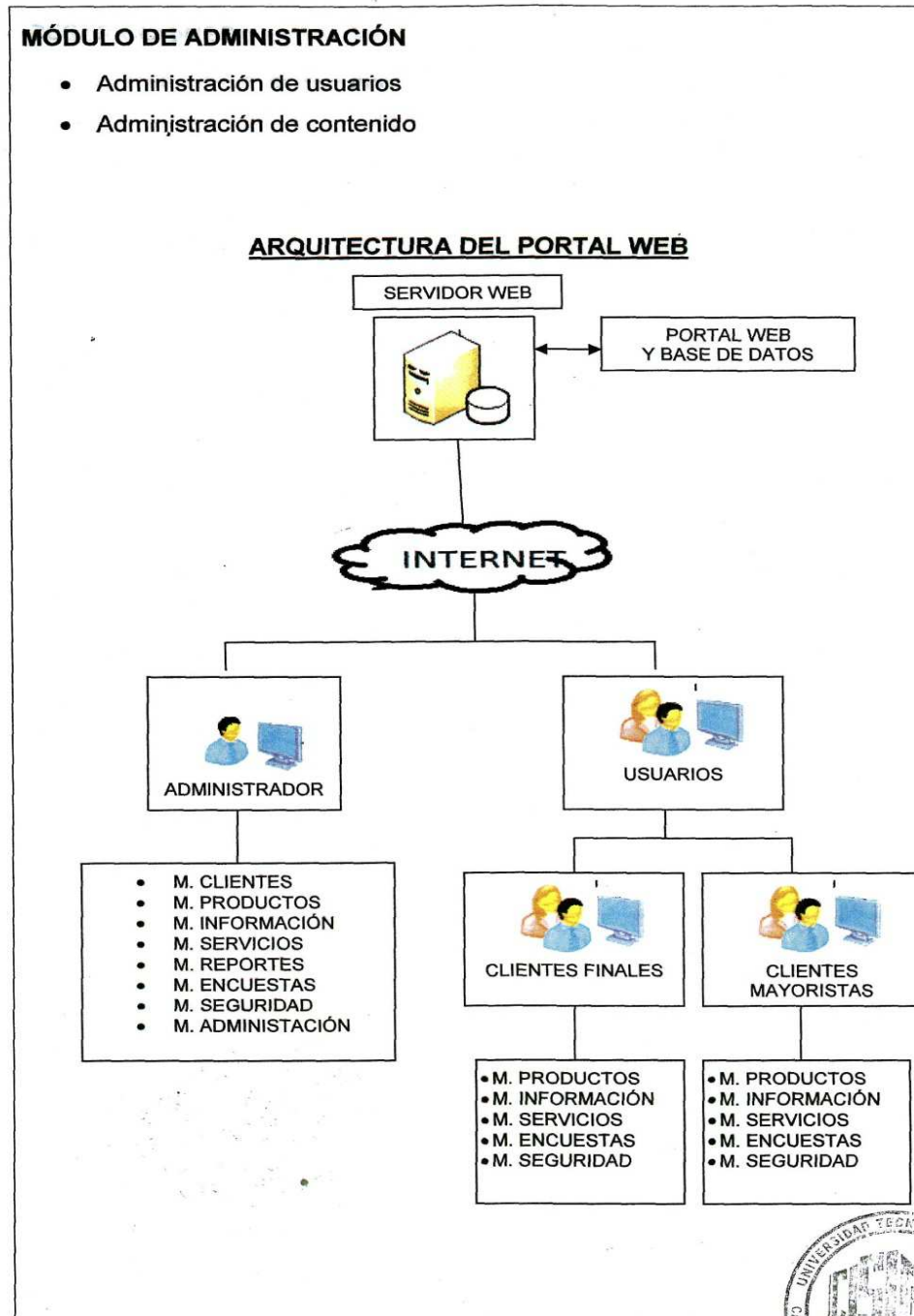
MÓDULO DE ENCUESTAS

- Encuestas de satisfacción del cliente
- Encuestas del producto

MÓDULO DE SEGURIDAD

- Login de ingreso
- Autenticación





RESTRICCIONES

Solo se realizará la comunicación con clientes para proveer información de productos, precios, stock y cotizaciones on line, mas no se realizará ningún tipo de transacción económica, facturación o ventas.

5. JUSTIFICACIÓN:

El estudio del framework Codeigniter beneficiará directamente a los estudiantes de la carrera de ingeniería en sistemas y de la Facultad de Ingeniería en Ciencias Aplicadas FICA ya que tendrán un medio de apoyo para el desarrollo de las aplicaciones web y la integración al gestor de bases de datos MySQL utilizando esta herramienta, además podrán conocer una herramienta poco explotada a pesar de sus numerosas y muy buenas características ya que tiene una curva de aprendizaje muy baja lo que les permitirá desarrollar aplicaciones web muy completas en poco tiempo. Los profesores de la facultad también serán beneficiados ya que conocerán otra herramienta para enseñar en clase.

Indirectamente se beneficiará la empresa Megasystem y los usuarios que requieran productos informáticos ya que se creará una herramienta web que permitirá la difusión de los productos y servicios mediante el internet.

6. CONTEXTO:

Existen los siguientes proyectos que tienen cierto grado de similitud con el proyecto a realizarse:



PROYECTO	AUTOR	SEMEJANZA	DIFERENCIA
Desarrollo de una Aplicación Web utilizando la tecnología JSF con una arquitectura MVC para el negocio de Servicio de Catering y eventos de La Viandé.	Ana Belén Obando Varas	Se desarrolla una aplicación web con arquitectura MVC	Proyecto realizado: Se usa la tecnología JSF y se aplica a servicios de eventos. Proyecto propuesto: Se usa la tecnología Codeigniter y se aplica a la información de productos computacionales.
Análisis y Estudio de Tecnología Ruby on Rails con bases de Datos Postgres para Aplicaciones Web 2.0, con el aplicativo: Implementación del Portal Web 2.0 para la Mancomunidad de la Cuenca del Río Mira.	Christian Fernando Realpe Rosero	Se realiza un estudio de un framework para desarrollo web	Proyecto realizado: Se usa la tecnología Ruby on Rails y se aplica al portal de una mancomunidad. Proyecto propuesto: Se usa la tecnología Codeigniter y se aplica a la información de productos computacionales.
<p>7. CONTENIDOS</p> <p>CAPÍTULO I</p> <p>INTRODUCCIÓN</p> <ul style="list-style-type: none"> • Datos generales de la empresa • Reseña histórica de la empresa • Misión • Visión • Valores • Estructura organizacional 			



CAPÍTULO II**ESTUDIO DEL FRAMEWORK CODEIGNITER Y LA ARQUITECTURA MVC**

- Instalación
- URLs de CodeIgniter
- Controladores
- Vistas
- Modelos
- Funciones Helper
- Bibliotecas de CodeIgniter
- Drivers de CodeIgniter
- Ruteo
- Manejo de Errores
- Administrar aplicaciones
- Seguridad
- Sintaxis

CAPÍTULO III**DISEÑO DE LOS PROCESOS MEDIANTE RUP**

- Documento Visión
- Especificación de Requisitos
- Diagramas de caso de uso
- Diagrama de clases
- Diagrama de Secuencia
- Diagrama de estados
- Diagrama de Colaboración
- Diagrama de Distribución

CAPÍTULO IV**PORTAL WEB DE INFORMACIÓN DE PRECIOS Y STOCK**

- Diseño de la base de datos



- Diseño del Manual de Usuario
- Diseño del Manual Técnico

CAPÍTULO V

VALIDACIÓN Y PRUEBAS DE FUNCIONAMIENTO

- Prueba módulo clientes
- Prueba módulo productos
- Prueba módulo de información
- Prueba módulo de servicios
- Prueba módulo reportes
- Prueba módulo seguridad
- Prueba módulo de administración
- Certificación de funcionamiento

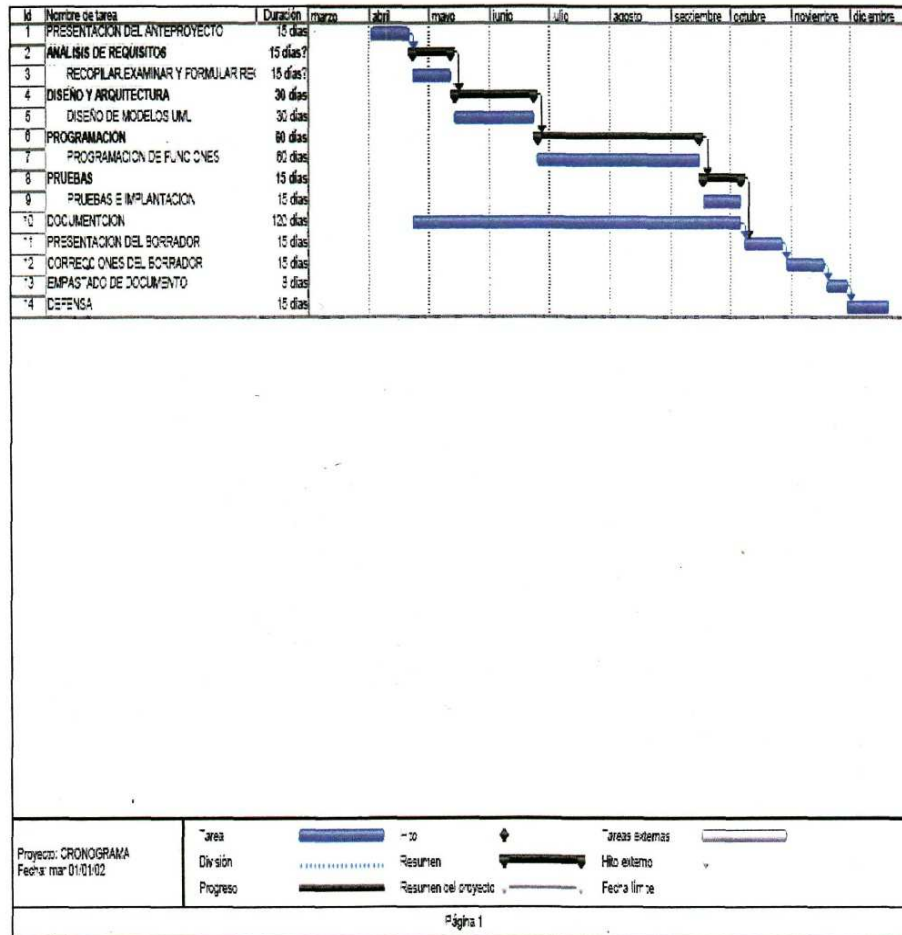
CAPÍTULO VI

CONCLUSIONES Y RECOMENDACIONES

- Conclusiones
- Recomendaciones



8. CRONOGRAMA DE ACTIVIDADES:



9. PRESUPUESTO:

COSTOS		ACTUAL (USD)	REAL (USD)	
HARDWARE	1	COMPUTADORES DE ESCRITORIO	800	800
	1	IMPRESORA	180	180
	1	ALOJAMIENTO WEB	200	200
		SUBTOTAL	1180	1180
SOFTWARE		MySQL 5.5.8	0	0
		CodeIgniter 2.0.1	0	0
		Php5	0	0
		Apache 2.2.17	0	0
		JQuery 1.7.2	0	0
		Netbeans 6.9.1	0	0
		Gimp 2.6.11	0	0
		SUBTOTAL	0	0
CAPACITACIÓN A DESARROLLARSE		Cursos de Capacitación Continua	160	160
PROYECTO		Papelería y suministros de oficina	400	400
		Subtotal	1740	1740
		Imprevistos 10%	174	174
			1914	1914



10. BIBLIOGRAFÍA:

Upton, D (2007). CodeIgniter for Rapid PHP Application Development.
Birmingham-Mumbai: Packt Publishing Ltd.

Myer, T (2008). Professional Codeigniter. Indianápolis: John Wiley & Sons Inc.

Gilfillan, I (2010). La Biblia de MySQL. Buenos Aires: Anaya Multimedia.

Pavón, J (2010). Creación de un portal con php y mysql. 4ª edición. Madrid:
RA-MA EDITORIAL

Álvarez, M (2012). Manual de CodeIgniter. Manuales de programación. 1(1),
1-43, Recuperado de: <http://www.codeigniter.com>

Comunidad De Desarrollo (2012). Guía de usuario en español.
Codeigniter. 1(1). Recuperado de: <http://www.codeigniter.com>

FUOC (2012). Introducción al Lenguaje de Modelado Unificado. Lenguaje
UML.1 (1). Recuperado de: <http://www.freeliros.com>

Murphey, R (2012). Fundamentos de JQuery. Tutorial de JQuery. 1(1).
Recuperado de: <http://librojquery.com/>

Bátiz, J (2012). Desarrollo orientado a objetos con UML. Tutorial de UML. 1(1)
Recuperado de: <http://es.scribd.com/Samirak/d/2458870-Desarrollo-Orientado-a-Objetos-con-UML-librobookespanolspanish>


DIRECTOR


ESTUDIANTE


Ibarra, 02 de Mayo del 2012

Ingeniero.
Edwin Rosero.
DECANO DE LA FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
UNIVERSIDAD TÉCNICA DEL NORTE.

Presente.

De mis consideraciones.

Por medio de la presente, certifico haber revisado el anteproyecto de titulación del Señor estudiante CEVALLOS AVILÉS WILSON JAVIER con el tema "ESTUDIO DE LA TECNOLOGÍA CODEIGNITER APLICADA AL DESARROLLO DE PORTALES WEB CON UNA ARQUITECTURA MVC"; y después de realizar las correcciones necesarias, manifiesto estar de acuerdo con los planteamientos propuestos, por lo que me comprometo a realizar la tutoría del mismo como Director de Proyecto hasta su culminación

Por la atención prestada a la presente anticipo mis sinceros agradecimientos.

Atentamente.



Ing. Marco Pusedá
DIRECTOR DEL PROYECTO

RECIBIDO 03 MAY 2012





Dirección: Víctor Gómez Jurado 3-44 y Dr. Luis Fernando Aguinaga
Telf. 06 2632007 – 06 2630894
Ibarra – Ecuador

Ibarra, 03 de mayo de 2012

Ing. Edwin Rosero.

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS.

UNIVERSIDAD TÉCNICA DEL NORTE.

Presente.

De mi consideración:

Por medio del presente, **MEGASYSTEM, "RILMER LENING MORAN CORAL"**, mayorista de hardware, autoriza y auspicia la elaboración del proyecto de grado **"ESTUDIO DE LA TECNOLOGÍA CODEIGNITER APLICADA AL DESARROLLO DE PORTALES WEB CON UNA ARQUITECTURA MVC"**; del Señor **CEVALLOS AVILÉS WILSON JAVIER**, con cédula **100247738-6**

Por la atención prestada al presente anticipo mis sinceros agradecimientos.

Atentamente,



Lening Morán Coral
Gerencia
PBX 2630894 / ext 111



ANEXO B

LISTA DE HELPERS DE CODEIGNITER



Helper Array.- El archivo del Helper Array contiene funciones que ayudan a trabajar con arrays.

Helper CAPTCHA.- El archivo del Helper CAPTCHA contiene funciones que ayudan en la creación de imágenes CAPTCHA.

Helper Cookie.- El archivo del Helper Cookie contiene funciones que ayudan a trabajar con cookies.

Helper Date.- El archivo del Helper Date contiene funciones que ayudan a trabajar con fechas.

Helper Directory.- El archivo del Helper Directory contiene funciones que ayudan a trabajar con directorios.

Helper Download.- El Helper Download le permite descargar datos a su escritorio.

Helper Email.- El Helper Email provee algunas funciones que lo ayudan a trabajar con emails. Para una solución de email más robusta, vea la Clase Email de CodeIgniter.

Helper File.- El archivo del Helper File contiene funciones que ayudan a trabajar con archivos.

Helper Form.- El archivo del Helper Form contiene funciones que ayudan a trabajar con formularios.

Helper HTML.- El archivo del Helper HTML contiene funciones que ayudan a trabajar con HTML.

Helper Inflector.- El archivo del Helper Inflector contiene funciones que permiten cambiar las palabras a plural, singular, etc.

Helper Language.- El archivo del Helper Language contiene funciones que ayudan a trabajar con archivos de idiomas.

Helper Number.- El archivo del Helper Number contiene funciones que ayudan a trabajar con datos numéricos.

Helper Path.- El archivo del Helper Path contiene funciones que permiten trabajar con rutas de archivos en el servidor.

Helper Security.- El archivo del Helper Security contiene funciones relacionadas con la seguridad.

Helper Smiley.- El archivo del Helper Smiley contiene funciones que permiten administrar smileys (emoticones).

Helper String.- El archivo del Helper String contiene funciones que ayudan a trabajar con cadenas de caracteres.

Helper Text.- El archivo del Helper Text contiene funciones que ayudan a trabajar con texto.

Helper Typography.- El archivo del Helper Typography contiene funciones que ayudan a formatear texto de forma semántica.

Helper URL.- El Helper URL contiene funciones que ayudan a trabajar con URLs.

Helper XML.- El archivo del Helper XML contiene funciones que ayudan a trabajar con datos XML.

ANEXO C

LISTA DE LIBRERÍAS DE CODEIGNITER



Clase Benchmark.- CodeIgniter tiene una clase para evaluar el desempeño que está siempre activa, permitiendo que se calcule la diferencia de tiempo entre dos puntos cualesquiera marcados.

Clase Calendar.- La Clase Calendar permite crear calendarios dinámicamente.

Clase Cart.- permite que los ítems se agreguen a la sesión que permanece activa mientras un usuario está navegando en el sitio. Estos ítems se puede recuperar y mostrar en un formato estándar de "carrito de compras", permitiéndole al usuario actualizar o quitar ítems del carrito.

Clase Config.- provee un medio para recuperar preferencias de configuración, estas preferencias pueden venir de un archivo de configuración por defecto (**application/config/config.php**) o de sus propios archivos de configuración personalizados.

Clase Database.- CodeIgniter viene con una clase de base de datos abstracta muy rápida y completa que soporta tanto las estructuras tradicionales como los patrones Active Record.

La Clase Active Record.- CodeIgniter usa una versión modificada del Patrón de Base de Datos Active Record. Este patrón permite que la información sea obtenida, insertada y actualizada en la base de datos con mínimo código. CodeIgniter no necesita que cada tabla de base de datos tenga su propio archivo de clase. Permite crear aplicaciones independientes de las bases de datos, ya que cada adaptador de base de datos genera la sintaxis de la consulta.

Clase Email.- La robusta Clase Email de CodeIgniter soporta las siguientes características:

- Varios Protocolos: Mail, Sendmail y SMTP
- Varios receptores
- CC y BCCs
- Correo HTML o de texto plano
- Adjuntos
- Salto de Línea Automático
- Prioridades
- Herramientas de Depuración de Email

Clase Encrypt.- provee encriptación de datos de dos vías. Usa un esquema quecompila el mensaje usando un algoritmo hash aleatorio, o encripta usando la biblioteca Mcrypt.

Validación de Formularios.- CodeIgniter provee una clase para preparación de datos y validación de formularios que ayuda a minimizar la cantidad de código que se escribe.

Clase FTP.- permite transferir archivos a un servidor remoto.

Clase Image_lib.- permite realizar las siguientes acciones:

- Redimensionamiento de Imágenes
- Creación de Miniaturas
- Recorte de Imágenes
- Rotación de Imágenes
- Marca de Agua en Imágenes

Clase Input.- La Clase Input sirve para dos propósitos:

- Por seguridad, preprocesa los datos de entrada globales.
- Provee algunas funciones helper para recuperar datos de entrada y preprocesarlos.

Clase Lang.- provee funciones para recuperar archivos de idioma y líneas de texto a los efectos de la internacionalización.

Clase Load.- se usa para cargar elementos, estos elementos pueden ser bibliotecas(clases), archivos de vistas, helpers, modelos, o sus propios archivos.

Clase Output.- es una clase pequeña con la función principal de Enviar la página web terminada al navegador. Es responsable de almacenar en caché las páginas web, si se usa esta funcionalidad.

Clase Pagination.- La clase Pagination de CodeIgniter provee funciones para la paginación de elementos.

Clase Security.- contiene métodos que ayudan a crear una aplicación segura, procesando los datos de entrada para brindar seguridad.

Clase Session.- La Clase Session permite mantener el "estado" del usuario y seguir su actividad mientras visita el sitio. Almacena la información de las sesiones para cada usuario como datos serializados (y opcionalmente encriptados) en una cookie. También puede almacenar los datos de sesión en una tabla de base de datos para mayor seguridad, ya que esto permite que el ID de sesión en la cookie del usuario se compare con el ID de sesión almacenado.

Clase Table.- provee funciones para permitirle generar tablas HTML automáticamente desde arrays o conjuntos de resultados de base de datos.

Clase Typography.- provee funciones que le ayudan a formatear texto.

Clase Unit_test.- Las pruebas de unidad son un enfoque de la ingeniería del software en el que se escriben pruebas para cada función en la aplicación.

Clase Upload.- permite subir archivos.

Clase URI.- provee funciones que ayudan a recuperar información de las cadenas URI.

Clase User_agent.- proporciona funciones que ayudan a identificar información acerca de navegadores, dispositivos móviles, o robots que visitan el sitio.

Clase Zip.- permite crear archivos Zip. Los archivos se pueden descargar al escritorio o guardarlos en un directorio.