

# UNIVERSIDAD NACIONAL DEL ALTIPLANO

FACULTAD DE INGENIERÍA MECÁNICA ELÉCTRICA, ELECTRÓNICA Y SISTEMAS

## ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS



**“PROTOTIPO DE SISTEMA DE ADMINISTRACIÓN DE  
CONTENIDOS PARA LA IMPLEMENTACIÓN Y  
MANTENIMIENTO  
DE PORTALES WEB ORIENTADOS AL E-COMMERCE”**

### **TESIS**

PRESENTADO POR:

**LISBETH DIANA GUILLÉN QUISCA**

PARA OPTAR EL TÍTULO PROFESIONAL DE:

**INGENIERO DE SISTEMAS**

**PUNO – PERÚ**

**2013**

# UNIVERSIDAD NACIONAL DEL ALTIPLANO

FACULTAD DE INGENIERÍA MECÁNICA ELÉCTRICA, ELECTRÓNICA Y SISTEMAS

## ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

“PROTOTIPO DE SISTEMA DE ADMINISTRACIÓN DE CONTENIDOS PARA  
LA IMPLEMENTACIÓN Y MANTENIMIENTO DE PORTALES WEB  
ORIENTADOS AL E-COMMERCE”

TESIS PRESENTADA POR:

LISBETH DIANA GUILLÉN QUISCA

PARA OPTAR EL TÍTULO PROFESIONAL DE:

INGENIERO DE SISTEMAS

APROBADA POR EL JURADO REVISOR CONFORMADO POR:

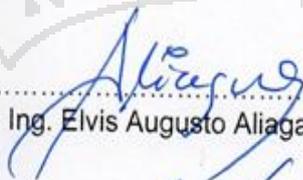
PRESIDENTE

  
M.Sc. Ing. Edelfré Flores Velásquez

PRIMER MIEMBRO

  
M.Sc. Ing. Fredy Aparicio Castillo Suaquita

SEGUNDO MIEMBRO

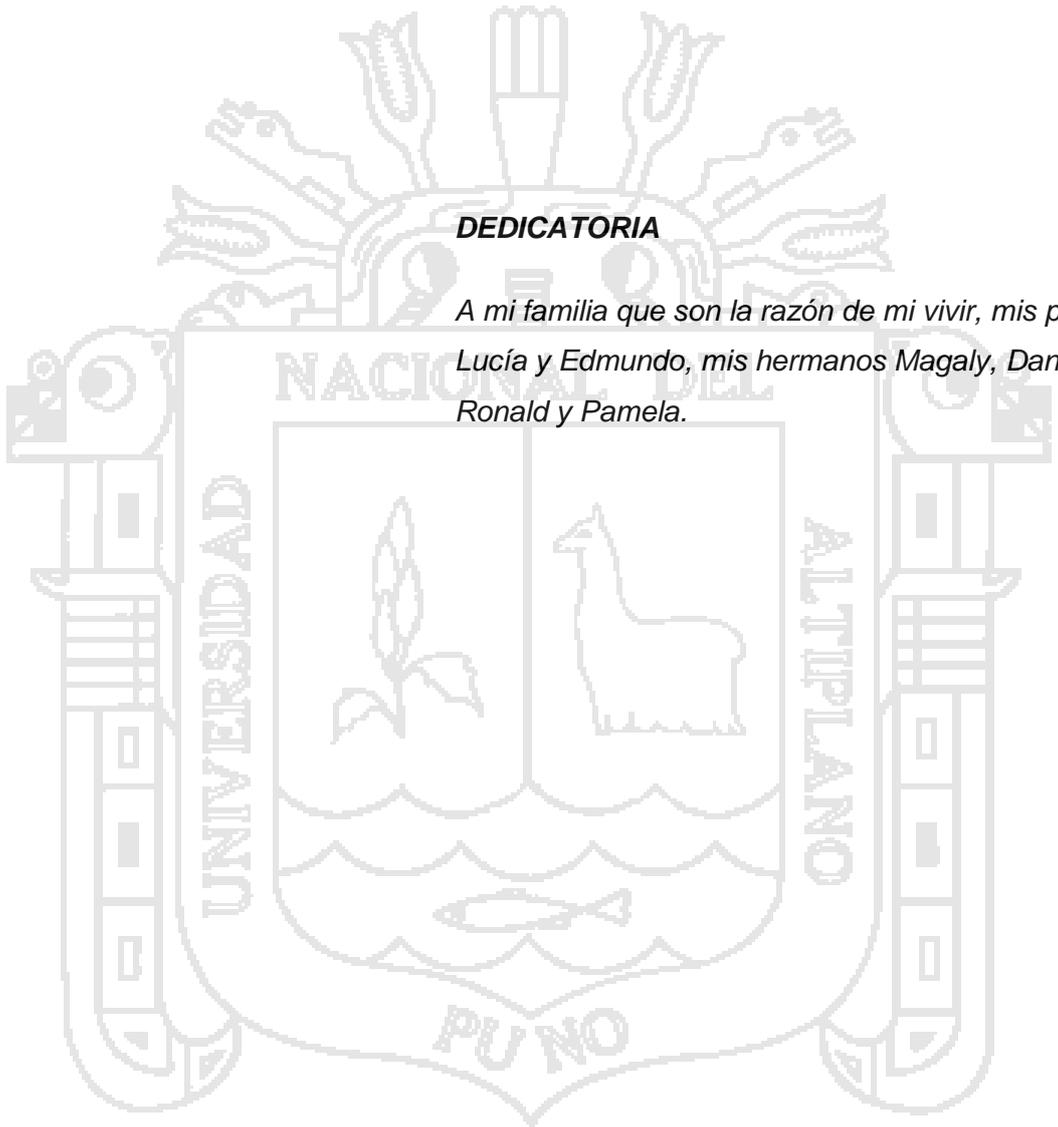
  
M.Sc. Ing. Elvis Augusto Aliaga Payehuanca

DIRECTOR DE TESIS

  
Mg. Ing. Robert Antonio Romero Flores

ÁREA: Informática

TEMA: Sistemas de información tradicionales y expertos



**DEDICATORIA**

*A mi familia que son la razón de mi vivir, mis padres  
Lucía y Edmundo, mis hermanos Magaly, Danny,  
Ronald y Pamela.*

### **AGRADECIMIENTOS**

*A Dios por bendecirme cada día de mi vida.*

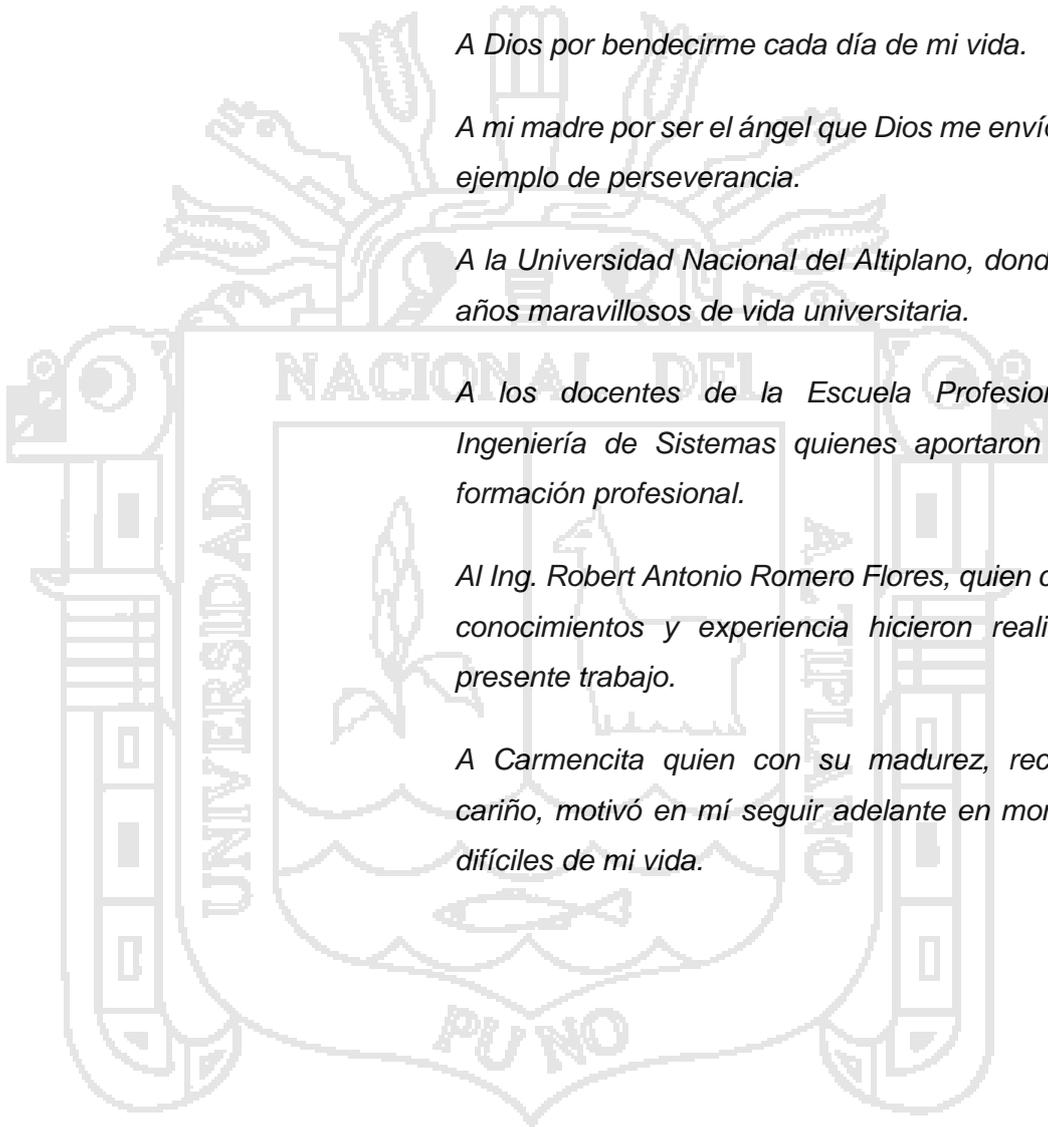
*A mi madre por ser el ángel que Dios me envió como ejemplo de perseverancia.*

*A la Universidad Nacional del Altiplano, donde pasé años maravillosos de vida universitaria.*

*A los docentes de la Escuela Profesional de Ingeniería de Sistemas quienes aportaron en mi formación profesional.*

*Al Ing. Robert Antonio Romero Flores, quien con sus conocimientos y experiencia hicieron realidad el presente trabajo.*

*A Carmencita quien con su madurez, rectitud y cariño, motivó en mí seguir adelante en momentos difíciles de mi vida.*



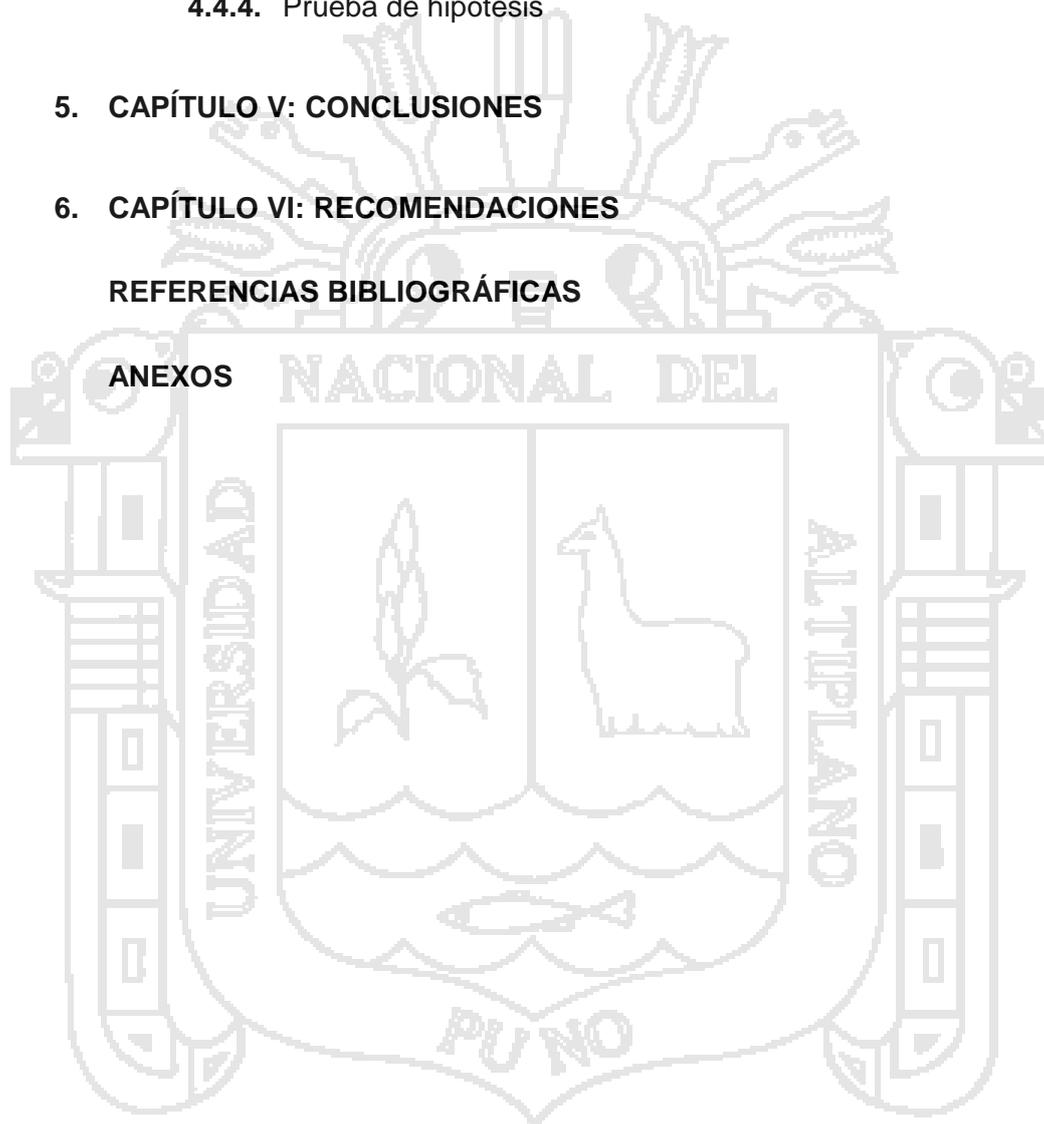
## ÍNDICE

<b>1. CAPITULO I: PLANTEAMIENTO DEL PROBLEMA OBJETO DE ESTUDIO</b>	<b>16</b>
1.1. PLANTEAMIENTO DEL PROBLEMA	16
1.2. JUSTIFICACIÓN DE LA INVESTIGACIÓN	17
1.3. OBJETIVOS DE LA INVESTIGACIÓN	19
1.3.1. Objetivo General	19
1.3.2. Objetivos Específicos	19
<b>2. CAPÍTULO II: MARCO TEÓRICO</b>	<b>20</b>
2.1. ANTECEDENTES DE LA INVESTIGACIÓN	20
2.2. MARCO TEÓRICO	22
2.2.1. E-Commerce (Comercio Electrónico)	22
2.2.1.1. Modalidades	22
2.2.1.2. Ventajas del Comercio Electrónico	23
2.2.1.3. Aspectos a Tomar en Cuenta en la Creación y Desarrollo de un Sitio de E-Commerce	24
2.2.2. El Comercio Electrónico en el Perú	25
2.2.2.1. Comercio Electrónico en Perú y su Situación Respecto a América Latina	29
2.2.2.2. Importancia del Ingreso de las Pequeñas y Medianas Empresas al Mundo de los Negocios Electrónicos en el Perú	29
2.2.2.3. Empresas exitosas en comercio electrónico en el Perú	31
2.2.2.4. Legislación Peruana Sobre Comercio Electrónico	32
2.2.3. Sistema Administrador de Contenidos (CMS)	33
2.2.3.1. Funciones principales de un CMS	33
2.2.3.2. Necesidad de un CMS	34
2.2.3.3. Beneficios del CMS	35
2.2.3.4. Anatomía del CMS	36
2.2.3.5. Metas del CMS	38
2.2.3.6. Tecnologías Web Utilizadas para el Desarrollo	38

de un CMS	39
2.2.4. Ingeniería Web	41
2.2.5. Metodologías de Desarrollo de Software Ágiles	42
2.2.6.1. Manifiesto Ágil	43
2.2.6. Metodología de Programación Extrema (XP)	45
2.2.6.1. Características Fundamentales de la Programación Extrema	47
2.2.6.2. Principios de la Programación Extrema	48
2.2.6.3. Prácticas de la Programación Extrema	51
2.2.6.4. Objetivos de la Programación Extrema	56
2.2.6.5. Valores de la Programación Extrema	56
2.2.6.6. Roles en la Programación Extrema	59
2.2.6.7. Fases de Desarrollo	60
2.2.7. Elementos Básicos de una Tienda Electrónica	69
2.2.7.1. Pilares de una Tienda Online	69
2.2.7.2. Elementos de una Tienda Online	71
2.2.8. Mecanismos de Seguridad en los Medios de Pago Electrónicos	76
2.2.9. Arquitectura MVC (Modelo-Vista-Controlador)	78
2.2.10. Web 2.0	80
2.2.10.1. Las Tecnologías 2.0	81
2.2.11. Framework CodeIgniter	82
2.2.11.1. Características de CodeIgniter	83
2.2.11.2. Diagrama de Flujo de Aplicación	84
2.2.12. Gestor de Base de Datos Mysql	85
2.2.12.1. ¿Qué es MySQL?	85
2.2.12.2. Características de MySQL	85
2.2.13. Lenguaje de Programación Php	86
2.2.14. Servidor Apache	86
2.2.14.1. Modelo Cliente-Servidor	87
2.2.14.2. Como obtenerlo y su licencia	87
2.2.15. SEO (Search Engine Optimization)	87
2.2.15.1. Ventajas de Ser Visibles en Buscadores	87
2.2.16. Diseño Web Adaptativo (Responsive Web Design)	88
2.2.16.1. Beneficios del Diseño Web Adaptativo	88
2.2.17. Métricas de Software	89

2.3. MARCO CONCEPTUAL	95
2.4. HIPOTESIS	99
2.4.2. Hipótesis general	99
2.4.3. Hipótesis específicas	99
2.5. OPERACIONALIZACIÓN DE VARIABLES	100
<b>3. CAPÍTULO III: MATERIALES Y MÉTODOS</b>	<b>102</b>
3.1. POBLACIÓN Y MUESTRA	102
3.2. MÉTODOS DE RECOPIACIÓN DE DATOS	103
3.3. MÉTODOS DE TRATAMIENTO DE DATOS	104
3.4. METODOLOGÍA PARA EL DESARROLLO DE SOFTWARE	105
3.5. MATERIALES Y EQUIPO UTILIZADO	106
<b>4. CAPÍTULO IV: RESULTADOS Y DISCUSIÓN</b>	<b>107</b>
4.1. ANÁLISIS DE REQUERIMIENTOS DEL PROTOTIPO DE SISTEMA DE ADMINISTRACIÓN DE CONTENIDOS	107
4.1.1. Planeación	107
4.1.1.1. Tipos de Usuario	108
4.1.1.2. Historias de Usuarios	108
4.1.1.3. Tareas a Realizar	113
4.1.1.4. Plan de entregas	121
4.1.1.5. Diagrama de Contexto de la Aplicación	124
4.2. DISEÑO DEL PROTOTIPO DE SISTEMA DE ADMINISTRACIÓN DE CONTENIDOS	125
4.2.1. Metáfora del Sistema	126
4.2.2. Tarjetas CRC	126
4.2.3. Diseño de la Base de Datos	130
4.2.4. Diseño de la Interfaz de Usuario	135
4.2.5. Arquitectura de Sección Pública	139
4.2.6. Arquitectura de la Sección Privada (Administración)	141
4.3. IMPLEMENTACIÓN DEL PROTOTIPO DE SISTEMA DE ADMINISTRACIÓN DE CONTENIDOS	142
4.3.1. Organización de la Aplicación	142
4.3.2. Codificación	143
4.3.3. Desarrollo del Modelo	144

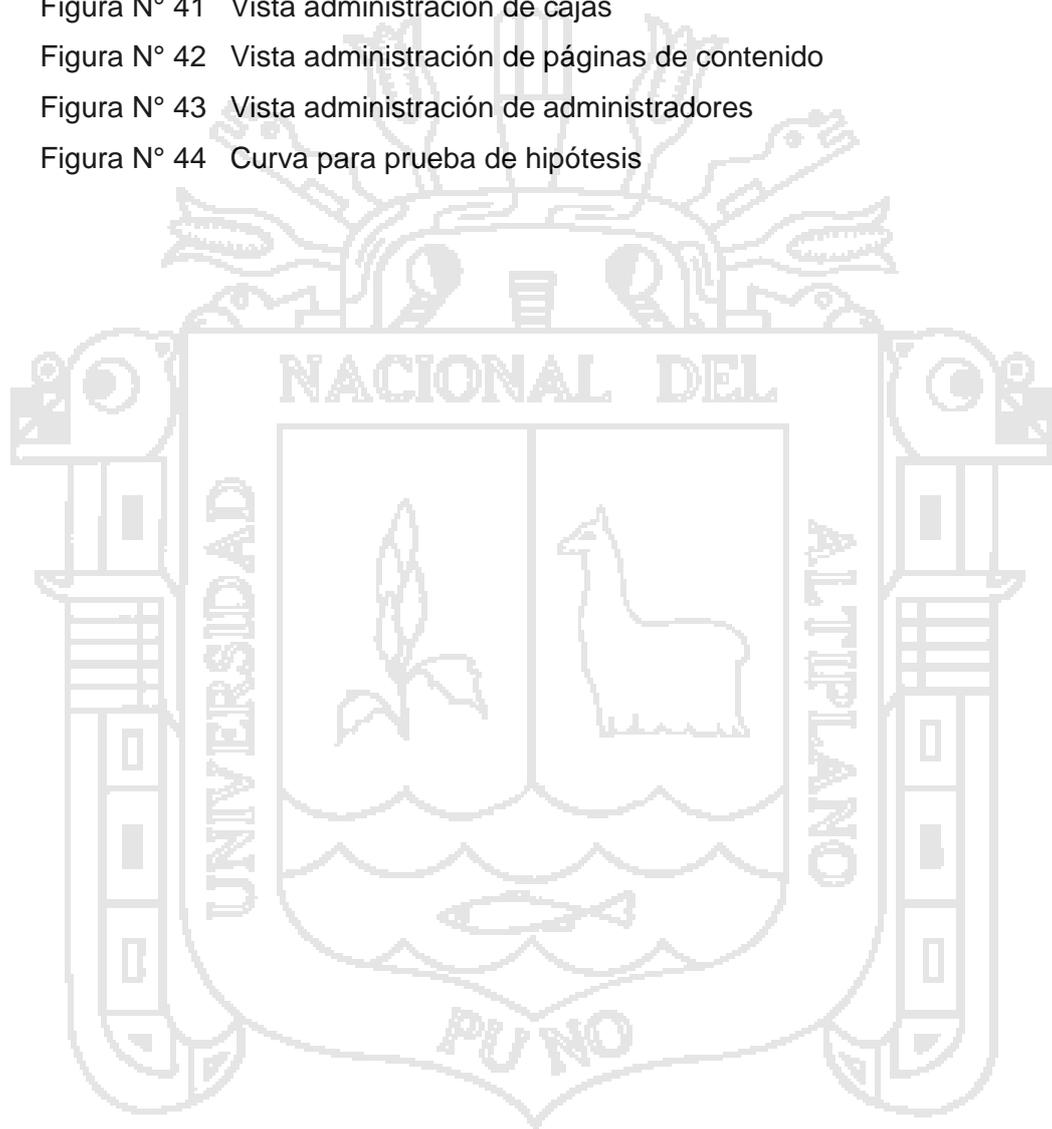
4.3.4.	Desarrollo de las Vistas	153
4.3.5.	Desarrollo del Controlador	169
4.4.	VALIDACIÓN DEL PROTOTIPO DE SISTEMA DE ADMINISTRACIÓN DE CONTENIDOS	174
4.4.1.	Pruebas de Aceptación	174
4.4.2.	Evaluación y Aceptación de la Aplicación por los Usuarios	174
4.4.3.	Métricas de Software	179
4.4.4.	Prueba de hipótesis	180
5.	<b>CAPÍTULO V: CONCLUSIONES</b>	183
6.	<b>CAPÍTULO VI: RECOMENDACIONES</b>	185
	<b>REFERENCIAS BIBLIOGRÁFICAS</b>	186
	<b>ANEXOS</b>	190



## ÍNDICE DE FIGURAS

Figura N° 1	E-Commerce en el Perú	26
Figura N° 2	Empresas según tipo de negocio	26
Figura N° 3	Empresas según canales de venta	27
Figura N° 4	Empresas según los principales medios de pago disponibles	27
Figura N° 5	Presencia en redes sociales	28
Figura N° 6	Anatomía de un CMS	37
Figura N° 7	Comparación de Metodologías de Desarrollo de Software	46
Figura N° 8	Prácticas de la programación XP	52
Figura N° 9	Costo de cambios en un proyecto XP	57
Figura N° 10	Arquitectura Modelo Vista Controlador	79
Figura N° 11	Diagrama de flujo de aplicación	84
Figura N° 12	Métricas del software. Categorías	90
Figura N° 13	Diagrama de contexto de la aplicación	124
Figura N° 14	Diagrama de casos de uso de alto nivel	126
Figura N° 15	Diseño de la base de datos	131
Figura N° 16	Estructura de diseño visual de interfaz de acceso a la administración	135
Figura N° 17	Estructura de diseño visual de la interfaz administrativa	136
Figura N° 18	Estructura de diseño visual de la interfaz de inicio del portal web	137
Figura N° 19	Estructura de diseño visual de la interfaz de navegación del catálogo	138
Figura N° 20	Estructura de diseño visual de la interfaz de productos	139
Figura N° 21	Diagrama de caso de uso del cliente	140
Figura N° 22	Arquitectura MVC de la Tienda Virtual	141
Figura N° 23	Diagrama de caso de uso del administrador	142
Figura N° 24	Organización de la aplicación según CodeIgniter	143
Figura N° 25	Vista compuesta	154
Figura N° 26	Vista página principal de tienda virtual	159
Figura N° 27	Vista detalles de categoría	160
Figura N° 28	Vista detalle de producto	161
Figura N° 29	Vista carro de compras	162
Figura N° 30	Vista realizar pedido	163
Figura N° 31	Vista acceso al administrador	164
Figura N° 32	Vista inicial de administración del sistema	164
Figura N° 33	Vista administración de clientes	165

Figura N° 34	Vista administración de grupos de clientes	165
Figura N° 35	Vista administración de pedidos	166
Figura N° 36	Vista reportes	166
Figura N° 37	Vista administración de cupones	166
Figura N° 38	Vista administración de categorías de productos	167
Figura N° 39	Vista administración de productos	167
Figura N° 40	Vista administración de banner	168
Figura N° 41	Vista administración de cajas	168
Figura N° 42	Vista administración de páginas de contenido	168
Figura N° 43	Vista administración de administradores	169
Figura N° 44	Curva para prueba de hipótesis	182



## ÍNDICE DE CUADROS

Cuadro N° 1	Fases de desarrollo y principios XP	46
Cuadro N° 2	Ejemplo de historia de usuario	48
Cuadro N° 3	Ejemplo de tarjeta CRC	51
Cuadro N° 4	Ejemplo de caso de prueba de aceptación	53
Cuadro N° 5	Ejemplo de tarea de ingeniería	54
Cuadro N° 6	Operacionalización de variables	85
Cuadro N° 7	Personal que intervendrán en el uso del sistema	86
Cuadro N° 8	Prioridad de las historias de usuario	106
Cuadro N° 9	Fechas de entregas para cada una de las historias de Usuario	107
Cuadro N° 10	Tabla categorías	116
Cuadro N° 11	Tabla clientes	117
Cuadro N° 12	Tabla pedidos	118
Cuadro N° 13	Tabla cupones	118
Cuadro N° 14	Tabla administrador	119
Cuadro N° 15	Resultado de opinión con respecto al diseño de interfaz	175
Cuadro N° 16	Resultado de opinión con respecto al manejo del sistema	175
Cuadro N° 17	Resultado de opinión con respecto a la rapidez del sistema	176
Cuadro N° 18	Resultado de opinión con respecto a la implementación del portal de E-commerce	176
Cuadro N° 19	Resultado de opinión con respecto a la dificultad en la implementación del portal de E-commerce	177
Cuadro N° 20	Estimación del Tiempo de Implementación del Sistema	177
Cuadro N° 21	Resultado de opinión con respecto al mantenimiento del portal de E-commerce	178
Cuadro N° 22	Resultado de opinión con respecto a la autodescripción del sistema	178
Cuadro N° 23	Resultado de opinión con respecto a la aceptación del sistema	179

## RESUMEN

El presente trabajo de investigación titulado: **“Prototipo de Sistema de Administración de Contenidos para la Implementación y Mantenimiento de Portales Web Orientados al E-commerce”**, tiene como objetivos analizar, diseñar, implementar y validar un prototipo de E-commerce (Comercio Electrónico) mediante el cual las empresas y Pymes de nuestra región Puno podrán promocionar y vender sus productos por internet a los mercados nacionales e internacionales.

Para el desarrollo del sistema se hizo uso de la metodología ágil Programación Extrema (XP), permitiendo que el proyecto se adapte a los requerimientos del usuario en cualquier punto de su ciclo de vida. La metodología XP hace énfasis en los siguientes aspectos: satisfacción del cliente y trabajo en equipo; además su ciclo de vida está conformada por los siguientes puntos: planificación, diseño, codificación y pruebas. Para la implementación del sistema se utilizó el framework CodeIgniter que es un conjunto de herramientas que permiten construir una aplicación web usando PHP, el gestor de base de datos MySQL y servidor Apache. Como arquitectura de desarrollo se hizo uso del Modelo Vista Controlador (MVC), que permite la separación entre la lógica y la presentación del sistema.

La población para la presente investigación fue conformada por las empresas registradas en el Centro de Innovación Tecnológica de Los Camélidos Sudamericanos de Puno, de los cuales se consideró como muestra 34 usuarios quienes validaron la operatividad del sistema. Las entrevistas, encuestas y observación fueron los principales métodos de recopilación de datos.

La principal conclusión a la que se llega es: ha sido factible desarrollar el prototipo de sistema de administración de contenidos ya que facilita la implementación y mantenimiento de portales web orientados al E-commerce.

**Palabras claves:** E-commerce, Comercio electrónico en el Perú, Sistema de administración de contenidos, Modelo vista contenedor, Metodologías ágiles.

## ABSTRACT

The present research work entitled "Prototype of Content Management System for Implementation and Maintenance of E-commerce Oriented Web Sites", has aims to analyze, design, implement and validate a prototype of E-commerce (Electronic Commerce) through companies and SMEs in our region Puno can promote and sell their products by internet to national and international markets.

For the system development we used the agil methodology named Extreme Programing (XP), it led that the project could adapt to the user requeriments in any point of the life cycle. The XP methodology makes emphasis in the root aspects: client satisfaction and team work; propose that a project must adaptate to the requeriment change any point of its life cycle. For systems implementation we used CodeIgniter framework it's a set of tools that led build a web application using PHP, the MySQL database system and apache server as development architecture we used the Model view controller (VCM) that lets a good operation between logic and presentation.

The population for this research was the enterprises registered in the Center for Technological Innovation The South American Camelids of Puno, from we considered as sampling only 34 users, who validate the systems operativity; the the interviews, surveys and observation were the main methods data collection.

The main conclusion reached is: it has been possible to develop a prototype system of content management as it facilitates the implementation and maintenance of web portals oriented E-commerce.

**Keywords:** E-commerce, E-commerce in Peru, Content management system, Model view controller, Agile methodologies.

## INTRODUCCIÓN

El presente trabajo de tesis es una contribución al desarrollo de nuestra región, promoviendo el uso de Tecnologías de Información por parte de las empresas y Pymes. Uno de los factores que afectan la competitividad de las Pequeñas y Medianas Empresas, es la tecnología, la cual ha cambiado notablemente el escenario económico mundial, regional y nacional. El comercio electrónico en nuestro país se ha centrado en el capital, ya que muchas empresas están incursionando en este tipo de comercio con éxito, los empresarios de nuestra región aún no cuentan con los suficientes conocimientos para incursionar en éste ámbito.

El presente proyecto de tesis busca desarrollar una aplicación Web que permita la implementación y mantenimiento de portales Web, enfocado al e-commerce, de manera que facilite el mantenimiento de sitios web y también aporte en cuanto a los pasos a seguir para conseguir un negocio electrónico exitoso.

Mediante esta investigación se prueba la medida en que el prototipo de sistema de administración de contenidos facilita la implementación y mantenimiento de portales Web orientados al e-commerce.

El objetivo general es desarrollar un prototipo de sistema de administración de contenidos para la implementación y mantenimiento de portales web orientados al e-commerce, y los objetivos específicos analizar, diseñar, implementar y validar el prototipo.

Se busca probar la hipótesis “El prototipo de sistema de administración de contenidos facilita la implementación y mantenimiento de portales web orientados al e-commerce”, para lo cual se realizó la validación por parte del usuario entre otros.

Para una mejor comprensión del presente trabajo de investigación científica, se ha dividido en cinco capítulos.

El primer capítulo trata sobre el planteamiento del problema, donde se justifica del por qué se realiza la presente investigación y se formulan los respectivos objetivos de la investigación: general y específicos.

El segundo capítulo aborda sobre el marco teórico y conceptual de la investigación, considerando los antecedentes de investigación, se enfatizó en temas como administración de contenidos, comercio electrónico y metodología de desarrollo de software programación extrema (XP), framework CodeIgniter, arquitectura Modelo

Vista Controlador; así mismo, se considera la hipótesis a probar y el cuadro de Operacionalización de variables dependientes e independientes.

El tercer capítulo está referido al método de investigación, que incluye el método de tratamiento de datos, métodos de recopilación de datos, metodología para el desarrollo de software, población y muestra de la investigación, también se enumera las técnicas e instrumentos que se han utilizado para el desarrollo de la tesis.

El cuarto capítulo desarrolla los resultados y discusión en el cual se detalla el análisis de requisitos, el diseño del sistema, la implementación del sistema y la validación del sistema. Para el desarrollo del sistema se ha utilizado utilizando la metodología de desarrollo de software programación extrema (XP) mediante sus fases de planificación, diseño, codificación y pruebas.

Finalmente se presenta la bibliografía utilizada, las conclusiones y recomendaciones, así mismo, se ha considerado la sección de anexos.



# CAPÍTULO I

## PLANTEAMIENTO DEL PROBLEMA OBJETO DE ESTUDIO

---

### 1.4. PLANTEAMIENTO DEL PROBLEMA

En nuestra región existen empresas y Pymes (pequeñas y medianas empresas), las cuales ven la necesidad de contar con un portal de e-commerce (comercio electrónico); según entrevistas y una encuesta<sup>1</sup> previa realizada a representantes de empresas y Pymes artesanales exportadoras y no exportadoras, los cuales están registrados en el Centro de Innovación Tecnológica de los Camélidos Sudamericanos Puno<sup>2</sup>; un 80% tienen conocimientos básicos de Internet y saben que la tecnología del e-commerce les ayudaría a mejorar sus ventas; un 40% los cuales constituyen en su mayoría las empresas exportadoras realizan sus transacciones por correo electrónico y no cuentan con un portal web; ellos indican que están en condiciones de asignar un presupuesto para el uso de nuevas tecnologías y de esta manera ser más competitivos, llegar a más mercados y promocionar sus productos.

Si bien es cierto que la mayoría de empresas de desarrollo Web entregan a sus clientes sitios que cubren sus expectativas, en cuanto a diseño y navegación, es también bastante cierto que para muchos de estos clientes, la información que se muestra en el momento de la entrega del sitio, semanas después, ya no será necesariamente la misma, o se mostrará del mismo modo, o tendrá la misma estructura, navegación y/o diseño; todo esto porque deberá evolucionar de acuerdo al ritmo de la empresa.

En la actualidad, el mantenimiento, es uno de los factores más importantes para el éxito de un portal Web. La actualización periódica de información y el mantenimiento del diseño son aspectos obligatorios a tomar en cuenta como parte de la implementación; de nada sirve tener un portal diseñado perfectamente con información desactualizada, ó

---

<sup>1</sup> Ver Anexo A.

<sup>2</sup> Se crea de acuerdo a la Ley N° 27627 y su modificatoria N° 27890 con la finalidad de promover la actividad artesanal desde un punto de vista de desarrollo de cadenas y beneficiar a los agentes económicos tanto directos como indirectos que participan en ella por su capacidad de generar oportunidades de negocios y puestos de trabajo.

información al día con un diseño poco atractivo. Muchas veces este factor es dejado de lado por las empresas, debido al elevado costo que puede alcanzar.

La desactualización en un portal Web ocasiona que el visitante desconfíe de la información mostrada y que a la larga desista de visitar la página nuevamente, en el caso de portales dedicados al e-commerce es necesario su actualización continua para promocionar sus productos e incrementar sus ventas.

Por otro lado también existe la falta de conocimiento por parte de los empresarios de nuestra región, en cuanto a las actividades que se realizan para lograr un proyecto exitoso de e-commerce.

Es por eso que se plantea la creación de un prototipo de administración de contenidos para e-commerce, el cual dará la oportunidad de mantener actualizado el portal Web de forma sencilla, rápida y sobre todo de forma autónoma. Esta alternativa facilitará la gestión del portal Web.

Según lo descrito en la formulación del problema se ha llegado a plantear la siguiente pregunta central de estudio:

¿En qué medida el prototipo de sistema de administración de contenidos facilitará la implementación y mantenimiento de portales Web orientados al e-commerce?

## 1.5. JUSTIFICACIÓN DE LA INVESTIGACIÓN

Uno de los factores que afectan la competitividad de las Pequeñas y Medianas Empresas, es la tecnología, la cual ha cambiado notablemente el escenario económico mundial, regional y nacional en los últimos años. En la medida que las empresas han incorporado a sus procesos productivos el uso de nuevas técnicas de producción, además de innovadoras formas de comunicación y acceso a información, el mercado se ha vuelto cada vez más competitivo. Corresponde entonces valorar la tecnología como una herramienta más que apoya el negocio y hace más competitiva a las empresas, facilitando la gestión de los negocios.

En nuestro país la artesanía es una actividad que pasó de ser una alternativa económica al sustento familiar, a constituirse una de las principales

fuentes de ingreso para los sectores más pobres. En nuestra región por ser turística y además de tener una artesanía atrayente, muchas familias se dedican a esta actividad; si bien tiene a su favor una gran riqueza cultural y estética, debe competir fuertemente con productos de otros países, donde existe una mayor experiencia en cuanto a comercialización internacional.

Estamos viviendo un entorno competitivo que exige nuevos canales de atención a las empresas y Pymes comercializadoras, para aprovechar el desarrollo de los e-bussines (negocios electrónicos) que estamos empezando a vivir aquí en el Perú y en la región en general.

La falta de conocimiento para realizar e-commerce por parte de las empresas y Pymes de nuestra región hace que no aprovechen de esta tecnología la cual ha llevado a muchas empresas nacionales e internacionales a un mayor éxito, en cuanto a las empresas nacionales se nota la centralización en la capital, debido a la falta de innovación por parte de los profesionales de nuestra región, allegados al tema.

El desarrollo del e-commerce en nuestra región crecerá en la medida que se desarrolle una mayor y mejor oferta (con estándares internacionales), la demanda existe, ya que en la actualidad hay un incremento en cuanto al uso del internet por parte de los usuarios.

El presente proyecto de tesis busca desarrollar una aplicación Web que facilite el mantenimiento y además, permita la implementación de portales Web, para aquellas empresas que deseen incursionar en el comercio electrónico. Por otro lado durante el desarrollo de la misma se muestran conceptos de las actividades a realizar para un proyecto exitoso de comercio electrónico.

El punto central de este trabajo de tesis es la orientación del software hacia usuarios no informáticos, no sólo en el mantenimiento de los contenidos sino también en la configuración del diseño; de forma que ya no sean necesarias las contrataciones de servicios informáticos, ni las aprobaciones de costosos presupuestos, ya que cualquier usuario podrá realizar las actualizaciones que desee al portal de su empresa.

Para facilitar el uso de la herramienta por parte de usuarios no informáticos, se plantea la diferenciación del diseño y del contenido; esta

diferenciación permite una total independencia entre ambos aspectos de la página, de manera que, cualquier modificación en el diseño de la página no afecte al contenido de la misma; y del mismo modo, cualquier cambio en el contenido de la página sea transparente para su diseño.

Entre las facilidades con que cuenta el portal, la implementación se puede resumir en descargar el instalador, instalarlo en el hosting y personalizarlo.

## 1.6. OBJETIVOS DE LA INVESTIGACIÓN

### 1.6.1. Objetivo General

Desarrollar un prototipo de sistema de administración de contenidos para la implementación y mantenimiento de portales web orientados al e-commerce.

### 1.6.2. Objetivos Específicos

- Analizar los requerimientos que intervienen en la construcción del prototipo de sistema de administración de contenidos para la implementación y mantenimiento de portales web orientados al e-commerce.
- Diseñar un prototipo de administración de contenidos para la implementación y mantenimiento de portales web orientados al e-commerce con las metodologías adecuadas.
- Implementar un prototipo de sistema de administración de contenidos utilizando la tecnología web.
- Validar el prototipo de sistema de administración de contenidos para la implementación y mantenimiento de portales web orientados al e-commerce.

## CAPÍTULO II

### MARCO TEÓRICO

---

## 2.3. ANTECEDENTES DE LA INVESTIGACIÓN

### 2.3.1. TESIS: “Implementación de un Content Management System (CMS) en la Escuela de Ingeniería de Sistemas y Telecomunicaciones”

Desarrollado por: Cárdenas Barrero Paula Carolina y Tocora Manrique Carlos Enrique.

Año: 2008 - Universidad Sergio Arboleda – Colombia.

En este proyecto se realizó una selección de CMS entre los más destacados del mercado, partiendo de algunas características específicas que se ajusten tanto para la escuela de Ingenierías como para la Universidad. Como respuesta a una necesidad inminente, surge el objetivo principal de este proyecto, consistente en la creación de un sistema, que permite la interacción virtual entre los docentes y los alumnos.

Principales conclusiones a las que llega:

- El manejo de esta herramienta incentiva a la comunidad a mostrar habilidades en sus campos de preferencia, ya que este gestor de contenidos proporciona opciones de administración, publicación y mantenimiento de la información, con lo cual facilita la organización y circulación de los contenidos importantes para los estudiantes y docentes.
- La implementación de un CMS para una institución de las proporciones de una universidad, demuestra que facilita la comunicación entre los diferentes actores que influyen en el funcionamiento de esta, ya que proporciona un mayor número de herramientas para la interacción entre docentes y alumnos, lo que hace que este gestor de contenidos sea medio masivo para el intercambio de información entre la comunidad. (Cárdenas y Tocora, 2008)

### 2.3.2. TESIS: “Prototipo de Sistema E-Marketplace para las Pequeñas Empresas Distribuidoras de Artesanía de la Ciudad de Puno”

Desarrollado por: Sucasaire Monroy, Wildo.

Año: 2007 – Universidad Nacional del Altiplano – Perú.

El proyecto consiste en desarrollar un prototipo de mercado virtual donde se promociona las diferentes artesanías de pequeñas empresas y de esta manera busca realizar ventas por Internet a los mercados nacionales e internacionales.

Principales conclusiones a las que llega:

- El desarrollo e implementación del prototipo ayuda a la promoción de los productos.
- Es muy necesario desarrollar la aplicación utilizando bases de la ingeniería Web, ya que permite a las empresas de artesanía, ingresar a un ambiente tecnológico en el área de la informática.
- El prototipo realizó mejoras en cuanto a la perspectiva de negocios y las ganas de emprender en el mercado internacional en las diferentes empresas de artesanía. (Sucasaire, 2007)

### **2.3.3. TESIS: “Diseño y promoción de un sitio web dirigido a pequeños y microempresarios interesados en implementar Internet y el comercio electrónico a las estrategias de la organización”**

Desarrollado por: Salas Delgado, Marcos.

Año: 2000 – Universidad Católica de Santa María – Perú.

El proyecto propone que el diseño de un sitio web, permite a las pequeñas y microempresas estar presentes en Internet en condiciones óptimas y asimismo, acceder al segmento del mercado que hace uso de los servicios que brinda este medio interactivo. Se pretende que el empresario se adecue al uso de esta tecnología, de tal manera que pueda afrontar la globalización de los mercados, aplicando la tecnología informática relacionado al comercio electrónico, la publicidad en Internet y la personalización en la atención al cliente.

Principal conclusión a la que llega:

- Las pequeñas y microempresas frente a la facilidad de acceso del Internet, ven la oportunidad de tener una comunicación más rápida, económica y

directa con los clientes, ofreciéndoles un servicio confiable y eficaz. (Salas, 2000)

## 2.4. MARCO TEÓRICO

### 2.4.1. E-Commerce (Comercio Electrónico)

El e-commerce, es la actividad de compra y venta de bienes y servicios a través de la Internet donde las transacciones consisten del intercambio de una serie de documentos electrónicos que llevan información de los elementos negociados así como de los procesos, áreas y personajes involucrados. (Rodríguez, 2007:6)

#### 2.2.2.1. Modalidades

Es muy común referirse al comercio electrónico con el término e-commerce sin mayor importancia de los actores o proceso involucrados, pero existen variantes de comercio electrónico que dependen de la modalidad o naturaleza de la transaccionalidad. (Rodríguez, 2007:7)

Dichas variantes son identificadas como (Rodríguez, 2007:8):

- a. B2B: Business to Business (Negocio a Negocio). Es la modalidad de comercio electrónico destinado al comercio de mayoreo. Se caracteriza por el manejo de grandes volúmenes de mercancía, un mayor flujo de datos y enormes cantidades monetarias; todo aquello que precisamente se encontraría cuando las empresas hacen negocio con las empresas. Se manejan precios especiales por volumen, e inclusive por cliente (precios negociados). Puede o no recurrirse a un método de pago en línea y el proceso puede ser llevado automáticamente o involucrando personal. El proveedor de este servicio de comercio electrónico actúa más como un medio de transaccionalidad que como una tienda.
- b. B2C: Business to Consumer (Negocio a Consumidor). Es el más común para la mayoría de los usuarios de la Internet. Los precios son usualmente menores que los que se encuentran en la calle pero no son de mayoreo. Está enfocado a la venta en menudeo y para personas físicas. Los métodos de pago en línea están convirtiéndose en obligados, pero aún se conserva la modalidad COD.

- c. B2G: Business to Government (Negocio a Gobierno). Únicamente ventas a gobiernos locales, municipales y estatales es lo que contempla. Aplica reglas muy particulares para la licitación de contratos o la enajenación de bienes y servicios. El volumen y monto de ventas es el principal atractivo. Usualmente no hay pago en línea, pero la tendencia es incorporarlo.
- d. B2E: Business to Employee (Negocio a Empleado). Esta modalidad está vista como un medio para el ofrecimiento de prestaciones y beneficios que la compañía puede ofrecer a sus empleados. Usualmente son tiendas virtuales que sólo pueden ser vistas en la intranet de la organización. Formas de pago on-line están disponibles junto con la posibilidad de descuentos por nómina u otros esquemas.

#### **2.2.2.2. Ventajas del Comercio Electrónico**

El comercio electrónico realizado entre empresas es llamado en inglés Business-to-business o B2B. El B2B puede estar abierto a cualquiera que esté interesado (como el intercambio de mercancías o materias primas), o estar limitado a participantes específicos pre-calificados (mercado electrónico privado). Algunas de sus virtudes son (Gonzales, 2012:15):

- a. Mejoras en la distribución: La Web ofrece a ciertos tipos de proveedores (industria del libro, servicios de información, productos digitales) la posibilidad de participar en un mercado interactivo, en el que los costos de distribución o ventas tienden a cero, como por ejemplo en la industria del software, en la que los productos pueden entregarse de inmediato, reduciendo de manera progresiva la necesidad de intermediarios.
- b. Comunicaciones comerciales por vía electrónica: Actualmente, la mayoría de las empresas utiliza la Web para informar a los clientes sobre la compañía, aparte de sus productos o servicios, tanto mediante comunicaciones internas como con otras empresas y clientes; esto facilita las relaciones comerciales, así como el soporte al cliente, ya que al estar disponible las 24 horas del día, las empresas pueden fidelizar a sus clientes mediante un diálogo asincrónico que sucede a la conveniencia de ambas partes.

- c. Beneficios operacionales: El uso empresarial de la Web reduce errores, tiempo y sobrecostos en el tratamiento de la información. Los proveedores disminuyen sus costos al acceder de manera interactiva a las bases de datos de oportunidades de ofertas, enviar éstas por el mismo medio, y por último, revisar de igual forma las concesiones; además, se facilita la creación de mercados y segmentos nuevos, el incremento en la generación de ventajas en las ventas, la mayor facilidad para entrar en mercados nuevos, especialmente en los geográficamente remotos, y alcanzarlos con mayor rapidez.
- d. Facilidad para fidelizar clientes: Mediante la aplicación de protocolos y estrategias de comunicación efectivas que le permitan al usuario final del portal web de la compañía plantear inquietudes, levantar requerimientos o simplemente hacer comentarios con relación a los productos o servicios de la misma, y si y solo si estos comentarios son debidamente procesados se puede crear un elemento importante para lograr la fidelización de los clientes, y en consecuencia aumentar la re-compra de productos y servicios, así como también la ampliación del rango de cobertura en el mercado.

### **2.2.2.3. Aspectos a Tomar en Cuenta en la Creación y Desarrollo de un Sitio de E-Commerce**

La creación y el desarrollo de un sitio de comercio electrónico implicará por parte de la empresa tomar en cuenta dos aspectos (Iasoni, 2002:21):

- a. El sitio web: En efecto, el desarrollo de un sitio de comercio electrónico implica, en primer lugar, cuidar todos los temas directamente vinculados al desarrollo del sitio en sí (nombre del dominio, propiedad intelectual, política contractual referente al sitio y a su explotación).
- b. Los usuarios del sitio: El público como objetivo principal de un sitio de comercio electrónico, deberá ser el objeto de una atención particular por parte de la empresa. Esta toma en cuenta de los usuarios del sitio implica cuidar aspectos tan fundamentales como la protección al consumidor, la publicidad, la protección de los datos personales.

Sólo la consideración del conjunto de estos aspectos como esenciales, podrán garantizar a la empresa un sitio de éxito en la red.

### 2.2.3. El Comercio Electrónico en el Perú

El comercio electrónico no es una moda más en el mundo de la tecnología. Ahora se cuenta con cifras muy rigurosas acerca de la tendencia real que se está produciendo en este campo y con ello se tiene una idea clara de la evolución en los próximos años.

El reciente estudio que hizo público Futuro Labs<sup>3</sup> sobre “El Comercio Electrónico en el Perú”, en junio del 2012 de un total de 240 negocios analizados considerando como sitio de comercio electrónico a aquellas tiendas electrónicas que cuentan con un medio de pago por Internet, excluyendo a las que sólo incluyen como medio de pago a los pagos contra entrega o depósito en cuenta. Este estudio proporciona cifras muy significativas sobre cómo van evolucionando los negocios y los compradores a través de la web. Por una parte, los negocios, principalmente retails, han comenzado a realizar inversiones en la ampliación de la oferta de productos, en la difusión de sus plataformas. (Futuro Labs, 2012)

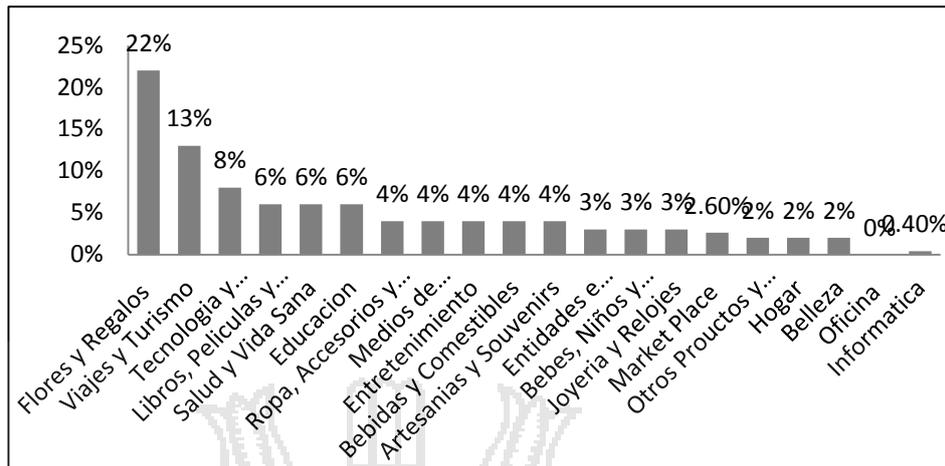
Por parte del comprador o usuario, cada vez se pierde el temor de hacer compras por Internet, a medida que los medios de pago se hacen más seguros.

Se llegó a las siguientes conclusiones:

- Centrándose en los sectores que más transacciones realizan, éstos tienen que ver con regalos (flores y otros), que representa el 21.8%. El segundo lugar lo ocupa el turismo. Hoy mucha gente compra boletos de avión, hace reservas y pagos a través de la web, llegando al 11.6%.

Figura N° 1  
E-Commerce en el Perú

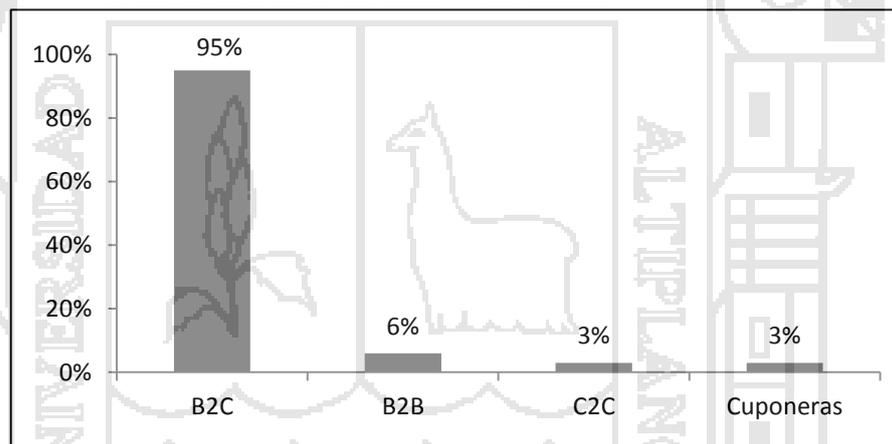
<sup>3</sup> Centro de investigación en internet aplicada a los negocios (<http://www.futurolabs.com>)



Fuente: Futuro Labs

- El 95% de las empresas analizadas están dirigidas hacia consumidores finales (B2C).

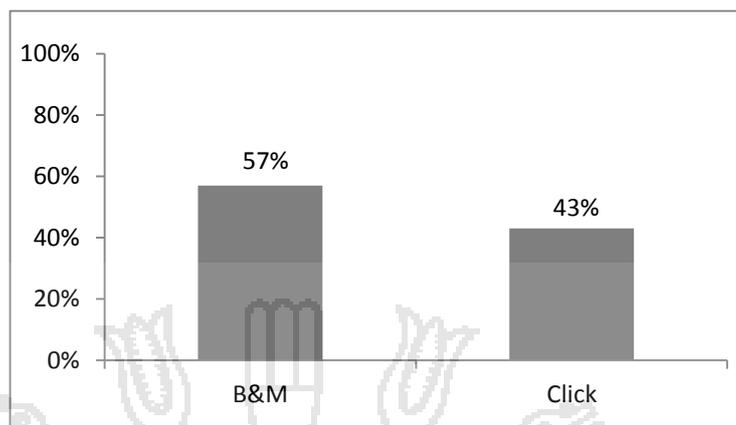
Figura N° 2  
Empresas según tipo de negocio



Fuente: Futuro Labs

- El 57% de los negocios involucrados en el estudio cuentan también con una tienda física además del canal virtual de ventas.

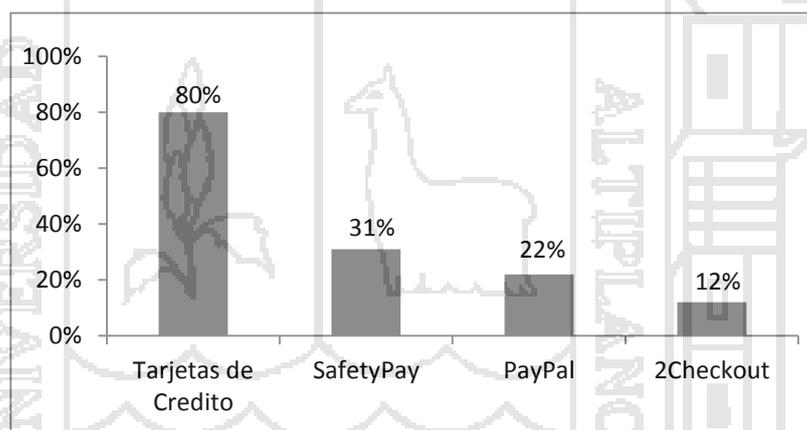
Figura N° 3  
Empresas según canales de venta



Fuente: Futuro Labs

- El medio de pago más utilizado por estas empresas es la Tarjeta de Crédito (80%), seguido por Safety Pay(31%), Pay Pal (22%) y 2CO (12%).

Figura N° 4  
Empresas según los principales medios de pago disponibles

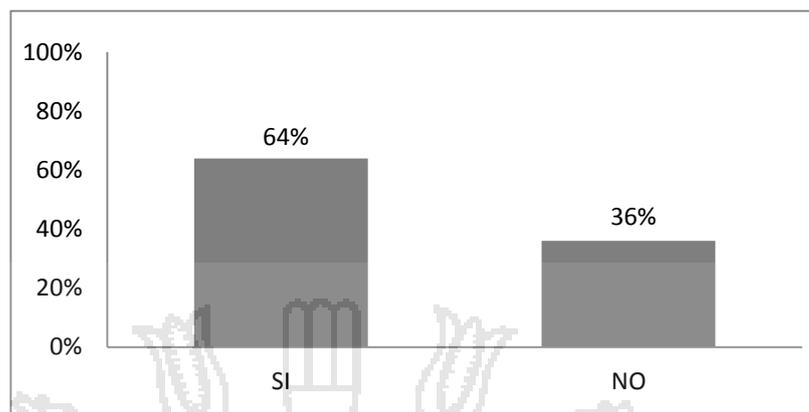


Fuente: Futuro Labs

2Checkout (2CO) resulta ser una plataforma de pagos apropiada para tiendas virtuales que aún no son muy grandes ya que tiene un costo más bajo que las afiliaciones a la plataforma de algunas tarjetas de crédito.

- El 64% de las empresas involucradas en el estudio tienen presencia en redes virtuales. El 63% de ellas posee una página en Facebook y el 34% una cuenta en Twitter.

Figura N° 5  
Presencia en redes sociales



Fuente: Futuro Labs

- Muchas de las empresas que están consideradas como un e-commerce en las asociaciones de comercio electrónico, no permiten concluir la transacción mediante un pago en línea a través de su sitio web (no se incluye transferencias bancarias). Esto se debe al alto costo de ciertos medios de pago y al bajo volumen de ventas que no justifica el pago por estos servicios y sus comisiones.
- Algunas páginas no cuentan con información de contacto con la empresa. Muchas otras no cuentan con medios de contacto inmediato como: live chat, skype o números telefónicos.

Esto podría llevarlos a perder oportunidades de venta debido a la falta de información puesto que no permiten a sus usuarios resolver sus dudas en el momento. No obstante, el uso de redes sociales es una forma de cubrir esta necesidad y el 64% de los negocios cuentan con presencia en por lo menos una.

- El desarrollo de un diseño amigable y agradable incrementa la confianza de los clientes potenciales.

### **2.5.2.1. Comercio Electrónico en Perú y su Situación Respecto a América Latina**

El Perú se ubica en el último lugar de Latinoamérica en cuanto a compras por Internet, según un estudio de comercio electrónico realizado por América Economía<sup>4</sup> en mayo del 2012. (América Economía, 2012)

El informe reveló que entre el 2010 y el 2011, el e-commerce creció 42.8% en América Latina y el Caribe, sobrepasando un gasto total de US\$43,000 millones, cifra que representa casi el doble de las ventas de comercio electrónico registradas en la región en 2009.

En orden porcentual, los países que presentaron una mayor participación en el gasto total son Brasil, responsable por el 59.1% de las ventas de comercio electrónico a consumidores, seguido por México, el cual registró 14.2%. Por su parte, el Caribe aportó un 6.4%; Argentina 6.2%; Chile 3.5%; Venezuela 3.3%; Centroamérica 2.4%; Colombia 2% y Perú 1.4%.

De acuerdo con el estudio, se espera que la región experimente un crecimiento de 26% en el comercio electrónico para fines de 2012, y del 28,5% en 2013. Igualmente, se prevé que para el 2015 el internet móvil genere una mayor actividad de compra, dado que la penetración de smartphones y tablets se estima que alcance un 50%.

### **2.5.2.2. Importancia del Ingreso de las Pequeñas y Medianas Empresas al Mundo de los Negocios Electrónicos en el Perú**

Según Morales (2012), gerente de Proyectos del Centro de Desarrollo Emprendedor de la Universidad ESAN, llega a las siguientes conclusiones:

- a. Se está viviendo un entorno competitivo que exige nuevos canales de atención. Las pequeñas y medianas empresas tienen las características de organización para aprovechar el desarrollo de los negocios electrónicos que se está empezando a vivir aquí en el Perú y en la región en general.
- b. Hay un mercado creciente, interno y externo,
  - Crecimiento promedio 8% anual PBI.

---

<sup>4</sup> Revista Internacional, AméricaEconomía Intelligence realiza los análisis más importantes de los que dispone la comunidad empresarial para apoyar la toma de decisiones ejecutivas.

- Además hay un mercado creciente que está siendo impulsado por los acuerdos comerciales y tratados de libre comercio.
- c. En la actualidad sin tecnología, la gestión de las empresas no es adecuada al entorno, es importante la innovación en productos y procesos.
- d. Crecimiento de usuarios en internet, gracias a proyectos como el de la Fibra Óptica de los Andes, se espera que se duplique la cantidad de usuarios de internet y además se duplicará la capacidad de oferta a la que podemos acceder a través del comercio electrónico, otro proyecto también es el de la Promoción de la banda ancha en el Perú<sup>5</sup>, que es Uno de los más grandes proyectos de infraestructura tecnológica del Ministerio de Transportes y Comunicaciones es la Promoción de la Banda Ancha y la construcción de la Red Dorsal de Fibra Óptica. Este proyecto de Ley se propone asegurar la conectividad con fibra óptica a las 195 capitales de provincia del país para el 2016<sup>6</sup>.
- e. Las pequeñas y medianas empresas en negocios electrónicos, consiguen buenos resultados, tan buenos como las grandes, a pesar de la diferencia tan enorme en inversiones que hacen porque a medida que los motores de búsqueda ( google, yahoo, bing, etc.) empiezan a ser la principal herramienta de investigación de los usuarios en internet al momento de tomar decisiones de compra, comercios más pequeños empiezan a tener probabilidades similares de llegar con su oferta al consumidor de las grandes cadenas, para esto tenemos que estar preparados.

Razones para el crecimiento del e-commerce, Paul Morales llega a las siguientes conclusiones:

- a. Aumento de la clase media; luego de años de crecimiento y recuperación económica en el Perú, somos más de clase media que lo que éramos hace 5 o 10 años.

---

<sup>5</sup> Según artículo del diario gestión: BID: Promoción de la banda ancha en el Perú es el proyecto más innovador de la región

( <http://gestion.pe/2012/06/25/tecnologia/bid-masificacion-banda-ancha-peru-proyecto-mas-innovador-region-2005892>)

<sup>6</sup> Ley de Promoción de la Banda Ancha y Construcción de la Red Dorsal Nacional de la Fibra Óptica (<http://gestion2.e3.pe/doc/0/0/0/0/4/4089.pdf>)

- b. Acceso a las tarjetas de crédito, actualmente existen facilidades de parte de las financieras para obtener una tarjeta de crédito y débito.
- c. Auge económico, propio de la dinámica de consumo del mercado interno.
- d. Cupones de descuento, existe ofertas que llegan al correo electrónico genuinas de las cuales las personas están interesadas en aprovecharlas.
- e. Aumento de la oferta, cada vez hay más empresas que entran al comercio electrónico, ya que empieza generarse un efecto retroalimentador.

Podríamos vender hasta cuatro veces esa cifra, falta oferta, las empresas deben entrar en el negocio, ya que el internauta peruano es el que más tiempo entra en internet, somos en la región el país que más pasa en horas promedio en internet.

### 2.5.2.3. Empresas exitosas en comercio electrónico en el Perú

Algunas de las empresas consideradas como exitosas en nuestro país son las siguientes:

- a. Saga Falabella: [www.falabella.com.pe](http://www.falabella.com.pe), impulsa el comercio electrónico en el país a través de sus campañas en Internet, su participación en redes sociales y publicidad en medios masivos. Actualmente brinda una de las mejores experiencias de compra en internet.
- b. Wong: [www.wong.com.pe](http://www.wong.com.pe), una de las primeras tiendas electrónicas en el Perú y que actualmente está incursionando en redes sociales y mejorando su plataforma para ofrecer una mejor experiencia al comprador.
- c. Plaza21: [www.plaza21.com](http://www.plaza21.com), considerado como el primer centro comercial online del Perú.
- d. Iquiero.com: [www.iquiero.com](http://www.iquiero.com), orientado al mercado extranjero, aprovecha el gran tráfico que tiene un portal como Peru.com, productos entregados en Lima, luego se ampliaron provincias, cuenta con vehículos de transporte para entrega.
- e. Idakoos.pe: [www.idakoos.pe](http://www.idakoos.pe), innovación y gran oferta de productos. Permite crear tus propios diseños; llegando a tener más de 3 billones de modelos para ofrecer en su tienda electrónica.

- f. Platanitos: [www.platanitos.com](http://www.platanitos.com), la oferta online dirigida al segmento femenino viene incrementándose, el jefe de proyecto web, Alexander Mont, revela cómo ha crecido en forma significativa la demanda y el incremento que puede llegar a tener en época de fiestas, destaca la seguridad que brinda, en este caso, el sistema Safety Pay.
- g. Tigress.com: [www.tigres.com.pe](http://www.tigres.com.pe), si abrir una tienda virtual no es posible aún, las empresas pueden recurrir a plataformas como Tigress.com, que les permitirán ingresar al mundo del comercio electrónico sin gastar demasiado. Solo necesitan registrarse, y luego de una breve evaluación, sus productos estarán a la venta en internet. Los clientes no solo podrán pagar por esta vía, sino que también recibirán lo que compren en la dirección que indiquen,

#### 2.5.2.4. Legislación Peruana Sobre Comercio Electrónico

Uno de estos grandes descubrimientos en el mundo de la tecnología y la informática es Internet la red de redes que ha permitido a todas las personas del mundo comunicarse en cuestión de segundos y, desde la óptica comercial, el instrumento que viene revolucionando la forma de hacer negocios con todo el mundo, debido a que mediante los medios electrónicos se ofrecen bienes y servicios a un mayor número de personas.

Conscientes de la importancia del e-commerce en la economía de las naciones y en el tráfico patrimonial de bienes y servicios, en el Perú se ha propugnado la dación de una legislación orientada a permitir y contribuir al desarrollo del comercio electrónico en el sector empresarial o comúnmente llamado business to business (B2B). (Soto, 2010:2)

Dentro de esta línea de pensamiento, se encuentran vigentes desde el año 2000 tres leyes que tienen por finalidad incentivar y promover el e-commerce dentro y fuera del Perú.

- a. En primer lugar, se encuentra la ley que permite a las personas declarar su voluntad por medios electrónicos, otorgándoles validez y eficacia jurídica (Ley N° 27291, publicada el 24 de junio de 2000);
- b. La ley sobre firmas y certificados digitales (Leyes N° 27269 y 27310, publicadas el 28 de mayo y el 17 de julio de 2000); y,

- c. Por último, la ley que incorpora los delitos informáticos al Código Penal (Ley Nº 27309, publicada el 17 de julio de 2000).

### **2.5.3. Sistema Administrador de Contenidos (CMS)**

Un administrador o gestor de contenidos (Content Management Systems o CMS) es un software que se utiliza principalmente para facilitar la gestión de sitios Web, ya sea en Internet o en una Intranet. Consiste en una interfaz que controla una o varias bases de datos donde se aloja el contenido del sitio, y que además permite independizar el contenido del diseño; de esta forma, es posible manejar el contenido y darle al sitio, un diseño distinto en cualquier momento, sin tener que darle formato al contenido nuevamente. (Robertson, 2003a:1)

El CMS tiene la facilidad de manejar la estructura de un sitio Web, junto con su apariencia, así como ofrecer a los usuarios la posibilidad de personalizar las aplicaciones. (Boiko, 2005:46)

#### **2.5.3.1. Funciones principales de un CMS**

Un CMS tiene dos funciones principales: (Escobedo, 2008:15)

- a. Almacenar contenido: provee una serie de herramientas para que publicar contenido sea más fácil, como rellenar un formulario, y haya, además, una sola fuente para todos ellos.
- b. Administrar contenido: Facilita la publicación de contenidos en múltiples formatos a partir de una sola fuente.

#### **2.5.3.2. Necesidad de un CMS**

Se podría pensar que no es necesario para un sitio Web relativamente pequeño o cuando no se necesitan tantas funcionalidades. Eso sólo podría ser cierto para un Web con unas pocas páginas estáticas para el que no se prevea un crecimiento futuro ni muchas actualizaciones, lo que no es muy realista. En cualquier otro caso, la flexibilidad y escalabilidad que permiten estos sistemas, justifican su utilización en prácticamente cualquier tipo de Web. (Robertson, 2003b:2)

Estos son algunos de los puntos más importantes que hacen útil y necesaria la utilización de un CMS: (Escobedo, 2008:15,16)

- a. Inclusión de nuevas funcionalidades en la Web: Con un gestor de contenidos esta operación puede ser tan simple como incluir un módulo realizado por terceros, sin que eso suponga muchos cambios en la Web. El sistema puede crecer y adaptarse a las necesidades futuras.
- b. Mantenimiento de gran cantidad de páginas: En una Web con muchas páginas hace falta un sistema para distribuir los trabajos de creación, edición y mantenimiento con permisos de acceso a las diferentes áreas. También se tienen que gestionar la metadata de cada documento, las versiones, la publicación y caducidad de páginas, entre otros aspectos. Todas estas tareas son soportadas por un gestor de contenidos.
- c. Reutilización de objetos o componentes: Un gestor de contenidos permite la recuperación y reutilización de páginas, documentos, y en general de cualquier objeto publicado o almacenado.
- d. Páginas interactivas: Las páginas estáticas llegan al usuario exactamente como están almacenadas en el servidor Web. En cambio, las páginas dinámicas no existen en el servidor tal como se reciben en los navegadores, sino que se generan según las peticiones de los usuarios. Para conseguir esta interacción, los gestores de contenido se conectan con una Base de Datos que hace de repositorio central de todos los datos de la Web.
- e. Cambios del aspecto de la Web: Debe existir independencia entre el contenido y la presentación, de manera que un cambio de diseño no implique la revisión de muchas páginas para su adaptación. Los gestores de contenido facilitan los cambios con la utilización por ejemplo, del estándar CSS (Cascading Style Sheets u hojas de estilo en cascada) con lo que se consigue la independencia de presentación y contenido.
- f. Consistencia de la Web: La consistencia en un sitio Web no quiere decir que todas las páginas sean iguales, sino que haya un orden visual. Las diferencias entre páginas en una misma Web, causan sensación de

desorden al usuario. Los gestores de contenido pueden aplicar un mismo estilo en todas las páginas con el mencionado CSS, y aplicar una misma estructura mediante patrones de páginas.

- g. Control de acceso: Controlar el acceso a un sitio Web no consiste simplemente en permitir la entrada a la Web, sino que implica gestionar los diferentes permisos a cada área de la Web aplicados a grupos o individuos. Un gestor de contenidos permite administrar perfiles y roles para los usuarios.

### 2.5.3.3. Beneficios del CMS

Un gestor de contenidos Web permite a un usuario común de la página actualizar el contenido de la misma, de manera fácil e intuitiva, sin necesidad de tener conocimientos en informática y desde cualquier parte del mundo donde se tenga acceso a Internet; evitando así la dependencia de terceros para su mantenimiento.

Algunas de las principales ventajas de la utilización de un gestor de contenidos son las siguientes: (Robertson, 2003c:1)

- a. Bajos costos en mantenimiento: Con un gestor de contenidos, se tiene total control sobre el contenido de su sitio actual. No requiere de gastos extras de subcontratación o capacitación, ya que todos los detalles técnicos son manejados por la herramienta de manera sencilla y transparente para el usuario.
- b. Ahorro de tiempo: Un gestor de contenidos permite controlar la publicación de páginas de manera inmediata: agregar y modificar páginas Web utilizando esta herramienta toma sólo minutos en comparación con complejos programas de diseño y editores de código HTML. Además un gestor de contenidos estará disponible las 24 horas del día de manera que estos cambios puedan realizarse a tiempo y sin demora.
- c. Mejor control de imagen de la página Web: Un gestor de contenidos separa el contenido de su presentación, usa plantillas y controladores de estilo para el diseño. De esta manera se puede manipular el diseño para

mantener una consistente y profesional imagen corporativa, esto sin afectar el contenido que existe actualmente.

- d. Mayor información para una mejor publicación: Un gestor de contenidos brinda una manera ágil de controlar y aprobar la publicación de temas e información en un sitio Web. Así, se podrán publicar las páginas cuando se verifique que la información es correcta. Se podrá seguir trabajando y manipulando la información aún cuando ésta no esté publicada.
- e. Mayor interacción con el usuario: Los gestores de contenidos ofrecen módulos y componentes que brindan al sitio Web mucha interacción con sus visitantes, tales como: foros de discusión, registro de usuarios, catálogos de productos en línea y carritos de compra, libro de visitas, gestores de formularios de contacto, galerías de imágenes, listas de correo, boletines de suscripción, etc.

#### **2.5.3.4. Anatomía del CMS**

Se encuentra estructurada en cuatro fases para lograr un alto nivel de funcionalidad.

Figura N° 6  
Anatomía de un CMS



A continuación, se explicará de manera detallada cada una de las partes de la anatomía del CMS de la figura N° 2.6, estas son (Robertson, 2003a:2,3):

- a. Creación de contenido: Un CMS aporta herramientas para que los creadores sin conocimientos técnicos en páginas Web puedan concentrarse en el contenido. Lo más habitual es proporcionar un editor de texto WYSIWYG, en el que el usuario ve el resultado final mientras escribe, al estilo de los editores comerciales, pero con un rango de formatos de texto limitado. Esta limitación tiene sentido, ya que el objetivo es que el creador pueda poner énfasis en algunos puntos, pero sin modificar mucho el estilo general del sitio Web.
- b. Gestión de contenido: Los documentos creados se depositan en una base de datos central donde también se guardan el resto de datos de la Web, cómo son los datos relativos a los documentos (versiones hechas, autor, fecha de publicación y caducidad, etc.), datos y preferencias de los usuarios, la estructura de la Web, etc.
- c. Publicación: Una página aprobada se publica automáticamente cuando llega la fecha de publicación, y cuando caduca se archiva para futuras referencias. En su publicación se aplica el patrón definido para toda la Web o para la sección concreta donde está situada, de forma que el resultado final es un sitio Web con un aspecto consistente en todas sus páginas. Esta separación entre contenido y forma permite que se pueda modificar el aspecto visual de un sitio Web sin afectar a los documentos ya creados y libera a los autores de preocuparse por el diseño final de sus páginas.
- d. Presentación: Un CMS puede gestionar automáticamente la accesibilidad de la Web, con soporte de normas internacionales de accesibilidad como

WAI, y adaptarse a las preferencias o necesidades de cada usuario. También puede proporcionar compatibilidad con los diferentes navegadores disponibles en todas las plataformas (Windows, Linux, Mac, Palm, etc.) y su capacidad de internacionalización lo permite adaptarse al idioma, sistema de medidas y cultura del visitante.

#### 2.5.3.5. Metas del CMS

El CMS es una herramienta que facilita el desarrollo e implementación de sitios Web dinámicos para cualquier tipo de organización. (Robertson, 2002:1)

Los objetivos y metas de un gestor de contenidos son: (Robertson, 2002:1,2)

- a. Incrementar flexibilidad en el sitio: Esta herramienta se adapta a los nuevos servicios, productos y estrategias que se presenten en la organización así como la actualización de los mismos.
- b. Mejorar exactitud de la información: Todas las páginas deben cumplir con ciertas características como exactitud de la información, actualización e información completa.
- c. Soporte de mercadeo: El CMS facilita el envío de la información de mercadeo de la organización ya que el sitio Web es una fuente de información para esta, así mismo también tiene soporte para este departamento de la organización.
- d. Reducir duplicidad de información: La información es almacenada una sola vez. Se mantiene un record de la misma para evitar este tipo de errores, y evitar costos más altos para la organización.
- e. Soporte de ventas: CMS maneja un módulo para el soporte de ventas llamado e-commerce que puede medir el impacto de las ventas en los sitios Web.
- f. Soporte de crecimiento del sitio Web: A medida que va creciendo la información que se debe colocar en un sitio Web el CMS contiene varios beneficios para poder administrar y soportar el crecimiento de la página.

- g. Reduce costos de mantenimiento: El CMS sustituye el incremento de la mano de obra por actualizaciones más sencillas que las anteriores. El gestor de contenidos en general, controla estos costos con su fácil manejo, por ello, no necesita de una persona calificada para poder realizar el mantenimiento pertinente a la página.
- h. Incrementa las visitas a la página: Debido a que es una página más dinámica e interactiva podrá ser visitada por cualquier usuario ya que es muy fácil de manejar.

### 2.5.3.6. Tecnologías Web Utilizadas para el Desarrollo de un CMS

Los gestores de contenidos utilizan para su desarrollo: tecnologías Web, tanto del lado cliente como del lado servidor; estas tecnologías deben ser escogidas de acuerdo a las necesidades y limitaciones que se presenten en las empresas o instituciones para las cuales son desarrollados. (Martinez y Tirano, 2008:13)

En la actualidad existen diferentes tecnologías organizadas de acuerdo al lado en donde es utilizada: lado cliente o lado servidor. A continuación se mencionan algunas de ellas:

#### A. Tecnologías Web del Lado del Cliente

##### a. Navegador Web

- Internet Explorer
- Mozilla
- Chrome
- Safari, etc

##### b. Tecnologías de Programación

- HTML
- JavaScript/Jscript
- VBScript
- Applets Java
- Componentes ActiveX en Visual C++, Visual Basic o .NET

#### B. Tecnologías Web del Lado del Servidor

## a. Servidores Web

- Internet Information Server (IIS)
- Apache, Apache-Tomcat
- WebSphere Webserver
- Motores Java, PHP, etc.

## b. Tecnologías de Programación

- PHP
- ASP
- JSP
- Servlets

La selección de las tecnologías Web a utilizar para el desarrollo de un gestor de contenidos es determinante, no solo para el éxito del gestor en sí, sino también para el éxito de las páginas que serán administradas por él. Esta selección de las tecnologías a utilizar dependerá de: la cantidad de información a publicar, la frecuencia de las actualizaciones y el poder adquisitivo de la empresa. (Martinez y Tirano, 2008:14)

Este último factor, el económico, es bastante importante para la selección de dichas tecnologías. Algunas combinaciones planteadas como soluciones bastante económicas (Open Source) y versátiles son:

- Linux, Apache, MySQL y PHP; y
- Linux, Apache-Tomcat, PostgreSQL y Java.

Existen además, soluciones que implican una mayor inversión, como es la ofrecida por Microsoft, que propone una combinación de sus productos: Windows, IIS, SQL Server y ASP; las ventajas de la utilización de estas tecnologías son la popularidad del sistema operativo Windows y la facilidad de aprendizaje del lenguaje ASP. (MSDN-Library, 2006)

Sin embargo la implantación de ASP está limitada para arquitecturas basadas en tecnología Microsoft, mientras que JSP sigue la filosofía de la arquitectura JAVA. (JSP-Overview, 2006)

#### 2.5.4. Ingeniería Web

Los sistemas y aplicaciones basados en Web (WebApps) ofrecen un complejo arreglo de contenido y funcionalidad a una amplia población de usuarios finales. La ingeniería web (IWeb) es el proceso con el que se crean WebApps de alta calidad. (Pressman, 2005:503)

La Iweb no es un clon perfecto de la ingeniería del software, pero toma prestados muchos principios fundamentales de ella. Además, el proceso IWeb acentúa actividades técnicas y administrativas similares. Existen sutiles diferencias en la manera como se dirigen dichas actividades.

En la mayoría de WebApps se encuentran los siguientes atributos (Pressman, 2005:504):

- a. Intensidad de red: Una WebApp reside en una red y debe satisfacer las necesidades de una variada comunidad de clientes
- b. Concurrencia: Un gran número de usuarios puede tener acceso a la WebApp al mismo tiempo.
- c. Carga impredecible: El número de usuarios de la WebApp puede variar en órdenes de magnitud de día con día.
- d. Desempeño: Si un usuario de la WebApp debe esperar demasiado, puede decidir irse a cualquier otra parte.
- e. Disponibilidad: Aunque la expectativa de una disponibilidad del total es poco razonable, los usuarios de las WebApps populares con frecuencia demandan acceso sobre una base de "24/7/365".
- f. Gobernada por los datos: Las WebApps se utilizan para tener acceso a información que existe en la base de datos.
- g. Sensibilidad al contenido: La calidad y naturaleza estética del contenido sigue siendo un importante determinante de la calidad de una WebApp.
- h. Evolución continua: A diferencia del software de aplicación convencional, que evoluciona a lo largo de una serie de planeadas liberaciones espaciadas cronológicamente, las WebApps evolucionan de manera continua.

- i. Inmediatez: Con las herramientas modernas se pueden producir elaboradas páginas web en cuestión de unas cuantas horas.
- j. Seguridad: Con la finalidad de proteger el contenido confidencial y ofrecer modos seguros de transmisión de datos, se deben implementar fuertes medidas de seguridad a lo largo de la infraestructura que sustenta una WebApp y dentro de la aplicación misma.
- k. Estética: Cuando una aplicación se diseña para comercializar o vender productos o ideas, la estética puede tener tanto que ver con el éxito como el diseño técnico.

### **2.5.5. Metodologías de Desarrollo de Software Ágiles**

El proceso de desarrollo de software en la vida real se describe como un proceso caótico donde muchas veces se realiza la corrección de errores muy apresuradamente sin que estos cambios consten en la documentación y según la naturaleza del proyecto se han propuesto varias metodologías que cuando muestran fallas se ajustan los cuellos de botella detectados, el punto crítico es cuando se exige reducir tiempos y mantener la calidad del producto (Ambler, 2002:29).

Las metodologías imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente. Lo hacen desarrollando un proceso detallado con un fuerte énfasis en planificar inspirado por otras disciplinas de la ingeniería. Las metodologías tradicionales han estado presentes durante mucho tiempo. No se han distinguido precisamente por ser muy exitosas en el resultado del proyecto ni para el bienestar del equipo. La crítica más frecuente a estas metodologías es que son burocráticas “muy estrictas”. Hay tanto que hacer para seguir la metodología que el ritmo entero del desarrollo se retarda.

Como una reacción a estas metodologías, un nuevo grupo de metodologías ha surgido en los últimos años. Durante algún tiempo se conocían como metodologías ligeras, pero el término aceptado ahora es metodologías ágiles. Para mucha gente el encanto de estas metodologías ágiles es su reacción ante la burocracia de las metodologías monumentales. Estos nuevos métodos buscan un justo medio entre ningún proceso y demasiado proceso,

proporcionando simplemente suficiente proceso para que el esfuerzo valga la pena. (Bair, 2002)

El resultado de todo esto es que los métodos ágiles cambian significativamente algunos de los énfasis de los métodos ingenieriles. La diferencia inmediata es que son menos orientados al documento, exigiendo una cantidad más pequeña de documentación para una tarea dada. De muchas maneras son más bien orientados al código: siguiendo un camino que dice que la parte importante de la documentación es el código fuente.

Entre las principales características de las metodologías ágiles tenemos (Auer y Roy, 2001:15):

- a. Los métodos ágiles son adaptables en lugar de predictivos: Los métodos ingenieriles tienden a intentar planear una parte grande del proceso del software en gran detalle para un plazo largo de tiempo, esto funciona bien hasta que las cosas cambian. Así que su naturaleza es resistirse al cambio. Para los métodos ágiles, no obstante, el cambio es bienvenido. Intentan ser procesos que se adaptan y crecen en el cambio, incluso al punto de cambiarse ellos mismos.
- b. Los métodos ágiles son orientados a la gente y no orientados al proceso: La meta de los métodos ingenieriles es definir un proceso que funcionará bien con cualquiera que lo use. Los métodos ágiles afirman que ningún proceso podrá nunca maquillar las habilidades del equipo de desarrollo, de modo que el papel del proceso es apoyar al equipo de desarrollo en su trabajo. Explícitamente puntualizan el trabajar a favor de la naturaleza humana en lugar de en su contra y enfatizan que el desarrollo de software debe ser una actividad agradable.

#### **2.2.6.1. Manifiesto Ágil**

En un taller de dos días en Snowbird, Utah, en febrero de 2001 se reunieron los representantes de cada una de las metodologías ágiles. Había mucho en común, y este reconocimiento era mucho mayor que las diferencias entre los procesos. Además de un contacto útil entre los líderes de procesos, existía también la idea de emitir una declaración conjunta en favor de procesos de desarrollo de software ágiles. El resultado es un Manifiesto para el

Desarrollo de Software Ágil, una declaración de los principios y valores comunes de los procesos ágiles. (Ambler, 2002:6)

El manifiesto fue sólo eso, una publicación que actuó como un punto de partida para aquellos que compartían estas ideas básicas. Uno de los frutos del esfuerzo fue la creación de un cuerpo más longevo, la Alianza Ágil. La Alianza Ágil es una organización sin fines de lucro que busca promover el conocimiento y la discusión de todos los métodos ágiles. Muchos de los líderes de metodologías de desarrollo de software ágiles son miembros y líderes de la Alianza Ágil.

Los principios que se establecieron en el manifiesto son (Ambler, 2002:6):

- a. La prioridad principal es satisfacer al cliente mediante tempranas y continuas entregas de software que le reporte un valor.
- b. Dar la bienvenida a los cambios. Las metodologías ágiles capturan los cambios para que el cliente tenga una ventaja competitiva.
- c. Entregar frecuentemente software que funcione, desde un par de semanas a un par de meses, con el menor intervalo de tiempo posible entre una entrega y la siguiente.
- d. La gente del negocio y los desarrolladores deben trabajar juntos a lo largo del proyecto.
- e. Construir el proyecto entorno a individuos motivados. Darles el entorno y el apoyo que necesitan y confiar en ellos para conseguir el trabajo.
- f. El diálogo cara a cara es el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo.
- g. El software que funciona es la medida principal de progreso.
- h. Los procesos ágiles promueven un desarrollo sostenible. Los promotores, desarrolladores y usuarios deberían ser capaces de mantener una paz constante.
- i. La atención continua a la calidad técnica y al buen diseño mejora la agilidad.

- j. La simplicidad es esencial.
- k. Las mejores arquitecturas, requisitos y diseños surgen de los equipos organizados por sí mismos.
- l. En intervalos regulares, el equipo reflexiona respecto de cómo llegar a ser más efectivo, y según esto ajusta su comportamiento.

### 2.5.6. Metodología de Programación Extrema (XP)

La programación extrema o eXtreme Programming (XP) es una metodología de desarrollo de software ágil presentada por Kent Beck autor del primer libro sobre programación extrema titulado: "Programación Extrema Explicada: Abrazando el cambio" (Extreme Programming Explained: Embrace Change). (Beck, 2004:1)

La programación extrema es la metodología ágil de desarrollo de software más utilizada debido a las destacadas innovaciones e incremento en la productividad en la industria de software, esta metodología se diferencia de las metodologías de software tradicionales pues pone énfasis en la entrega de resultados (prototipos) más que en otros procesos como la documentación del software.

Kent Beck conceptualiza la programación extrema de la siguiente forma: "La programación extrema comúnmente conocida como XP, es una disciplina del negocios de desarrollo de software que se enfoca en todo el equipo en común, metas razonables usando los valores y principios de la programación extrema, los equipos aplican apropiadamente practicas de XP en sus propios contextos, la práctica de XP es escogida por su valor para incrementar la creatividad humana, dar un mejor trato a los programadores, la programación extrema produce software de calidad y un ambiente sostenible" (Beck, 2004:18)

Kent Beck también menciona las metas de esta metodología: "Una de las metas de XP es traer responsabilidad y transparencia al desarrollo de software, para realizar el desarrollo de software como cualquier otra actividad empresarial, Otra meta es lograr resultados más eficientes con pocos defectos. Finalmente la programación extrema desea lograr estas metas sirviendo a las necesidades humanas de todos pasando por patrocinadores, administradores, evaluadores, usuarios y programadores" (Beck. 2004:23).

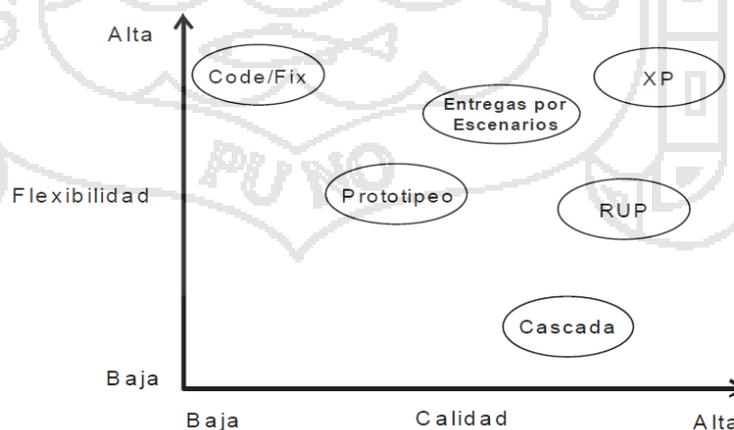
Por lo que se puede considerar la programación extrema como la adopción de las mejores metodologías de desarrollo de acuerdo a lo que se pretende llevar a cabo con el proyecto y aplicarlo de manera dinámica durante el ciclo de vida del software. La programación extrema es una metodología de desarrollo ligera basada en una serie de valores y una docena de prácticas que propician un aumento en la productividad a la hora de generar software. La meta real de XP es entregar el software requerido a tiempo.

Las características generales de XP es deliberadamente una metodología “liviana” que pasa por alto la utilización de elaborados casos de uso, la exhaustiva definición de requerimientos y la producción de una extensa documentación. (Anon, 2002)

Todo lo anterior puede parecer caótico según el enfoque tradicional de la ingeniería de software (aplicando metodologías como RUP); aunque XP tiene asociado un ciclo de vida y es considerado a su vez un proceso. La tendencia de entregar software en lapsos cada vez menores de tiempo y con exigencias de costos reducidos y altos estándares de calidad, hace que XP sea una opción importante a considerar.

Bair (2002:21) recopila información sobre la utilización de diversas metodologías de programación en comparación de Calidad y Flexibilidad, la misma se muestra en la figura N° 2.7.

Figura N° 7  
Comparación de metodologías de desarrollo de software



Como se puede destacar de la figura N°2.7 la metodología de desarrollo XP según los expertos garantiza mayor flexibilidad y calidad seguida por la más

popular de las metodologías de desarrollo de software tradicional la metodología RUP (Rational Unified Process).

### 2.5.6.1. Características Fundamentales de la Programación Extrema

Las características fundamentales del método son (Abrahamsson, 2006:25):

- Desarrollo iterativo e incremental: Consiste en realizar pequeñas mejoras, unas tras otras siempre con la retroalimentación de los usuarios.
- Pruebas unitarias continuas, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión: Se aconseja escribir el código de la prueba antes de la codificación. Véase, por ejemplo, las herramientas de prueba JUnit orientada a Java, DUnit orientada a Delphi y NUnit para la plataforma.NET. Estas dos últimas inspiradas en JUnit.
- Programación en parejas: La metodología XP recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo computador. La mayor calidad del código escrito de esta manera es más importante que la posible pérdida de productividad inmediata.
- Frecuente integración del equipo de programación con el cliente o usuario: Se recomienda que un representante del cliente trabaje junto al equipo de desarrollo, algunos procesos deben ser dirigidos por usuarios.
- Corrección de todos los errores antes de añadir nueva funcionalidad: Hacer entregas frecuentes.
- Refactorización del código: Significa: “reescribir ciertas partes del código para aumentar su legibilidad y mantenibilidad pero sin modificar su comportamiento”. Las pruebas han de garantizar que en la refactorización no se ha introducido ningún fallo. (Abrahamsson, 2006:12)
- Propiedad del código compartida: En vez de dividir la responsabilidad en el desarrollo de cada módulo en grupos de trabajo distintos, este método promueve el que todo el personal pueda corregir y extender cualquier parte del proyecto. Las frecuentes pruebas de regresión garantizan que los posibles errores serán detectados.

- Simplicidad en el código: Es la mejor manera de que las cosas funcionen. Cuando todo funcione se podrá añadir funcionalidad si es necesario. La programación extrema apuesta que es más sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado y quizás nunca utilizarlo.

### 2.5.6.2. Principios de la Programación Extrema

Los principios en los que se basa la programación extrema son:

#### a. Humanidad

Al respecto Beck (2004:33) menciona: “Los valores son demasiado abstractos para guiar directamente la conducta. Documentos de gran tamaño tienen la intención de comunicar, por lo que son las conversaciones diarias de comunicación. ¿Cuál es la más eficaz? La respuesta depende en parte del contexto, y en parte en los principios intelectuales”.

#### b. Economía

El proyecto de software que desarrolla debe brindar una utilidad; Beck (2004:34) acertadamente menciona: “Asegúrese de que lo que está haciendo tiene un valor de negocio, cumpla con los objetivos del negocio, y atienda las necesidades de negocio”. Dos aspectos de la economía que afectan el desarrollo de software son el valor del dinero en el tiempo y el valor de la opción de los sistemas y equipos.

#### c. Beneficio Mutuo

Cada actividad debe beneficiar a todos los involucrados en el proyecto de desarrollo de software. El beneficio mutuo es el principio más importante de XP y el más difícil de cumplir. Hay siempre soluciones a cualquier problema que le costó pérdida a una persona mientras se benefician otra. Cuando la situación es desesperada, estas soluciones parecen atractivas; Pero son siempre una pérdida neta; sin embargo, debido a la mala voluntad que se crea destruye las relaciones que debemos valorar. Se deben mantener las relaciones en el trabajo. (Beck, 2004:35)

#### d. Similitud

Beck (2004:36) da la siguiente recomendación: "Intente copiar la estructura de una solución en un nuevo contexto, incluso a diferentes escalas. Por ejemplo, el ritmo básico del desarrollo en que se escribe una prueba que falla y luego la hace funcionar. El ritmo opera en todas las escalas. En un trimestre liste los temas que desea tratar y luego refléjelos en historias de usuario. En una semana selecciona las historias de usuarios que quiere abordar, escriba pruebas para las historias de usuario seleccionadas y luego haga que funcionen. En pocas horas, se enumeran las pruebas que se necesitan escribir, escriba una prueba, haga que funcione, escriba otra prueba, y hacer que todo el trabajo listado se cumple".

**e. Mejora continua**

El ciclo de desarrollo de software en XP consiste en hacer lo mejor que se puede hoy en día, la búsqueda de la conciencia y el conocimiento necesario para hacerlo mejor mañana.

No hay un proceso perfecto. No existe un diseño perfecto. No hay historias perfectas; sin embargo, puede perfeccionar su proceso, su diseño, y sus historias. (Beck, 2004:37)

**f. Diversidad**

Los equipos de desarrollo de software donde todos son iguales, aunque sea cómodo, no son eficaces. Los equipos deben reunir una serie de habilidades, actitudes y puntos de vista para ver los problemas y dificultades y pensar en varias formas de resolver problemas e implementar nuevas soluciones; por tanto los equipos necesitan la diversidad.

**g. Reflexión**

Los buenos equipos no solo hacen su trabajo, ellos meditan acerca de lo que hacen. Analizan los puntos en los que acertaron o fallaron. No tratan de ocultar sus errores, exponen sus errores ya prende de ellos.

#### **h. Flujo**

Flujo en desarrollo de software consiste en liberar un flujo de software constante participando en todas las actividades de desarrollo simultáneamente. Las prácticas de XP son orientadas a un flujo continuo de actividades más que a fases discretas. (Beck, 2004:77)

#### **i. Oportunidades**

La programación XP permite ver a los problemas como oportunidades para el cambio. Lo cual no quiere decir que los problemas no existan en el desarrollo de software; Sin embargo, la actitud de "supervivencia" nos orienta a resolver los problemas a medida que se presentan para lograr la excelencia, los problemas necesitan convertirse en oportunidades para aprender y mejorar, no solo para sobrevivir. (Beck, 2004:78)

#### **j. Redundancia**

Los problemas difíciles en el desarrollo de software pueden ser resueltos de diferentes formas. Aún si la solución falla, la otra solución va a prevenir desastres. El costo de la redundancia es más que pagar por el ahorro de no tener el desastre. (Beck, 2004:79)

#### **k. Fracaso**

Si se sabe cómo implementar una historia de usuario y se tiene dos maneras; y sin embargo todas esas maneras fallan, ciertamente se aprendió algo importante.

Al respecto Kent Beck menciona: “¿Es un desperdicio el fracaso? No si imparte conocimiento, el conocimiento es valorable y algunas veces difícil de adquirir, si sabes la opción correcta para implementar una historia simplemente hazlo de esa forma, y si no supieras como hacerlo ¿Cuál sería la forma más económica de hacerlo?”. (Beck, 2004:80)

#### **l. Calidad**

Según Kent Beck calidad no es una variable de control. Los proyectos no son más rápidos por que acepten menos calidad. Tampoco van más lentos por que requieren de mayor calidad.

La calidad no es un factor netamente económico. Las personas necesitan hacer tareas de las cuales se sientan orgullosas. (Beck, 2004:81).

#### **m. Pasos de Bebé**

Los equipos siempre están tentando a hacer grandes cambios en grandes pasos después de todos hay un gran camino que recorrer y un poco tiempo para llegar a la meta, cambios hechos todos al mismo tiempo es peligroso.

Entonces hacer cambios en “pasos de bebe” permiten seguir en la dirección correcta lo cual no indica que el cambio es demasiado lento puesto que estos pasos pueden realizarse muy rápidamente. Este concepto toma importancia cuando un gran cambio es abortado. (Beck, 2004:82)

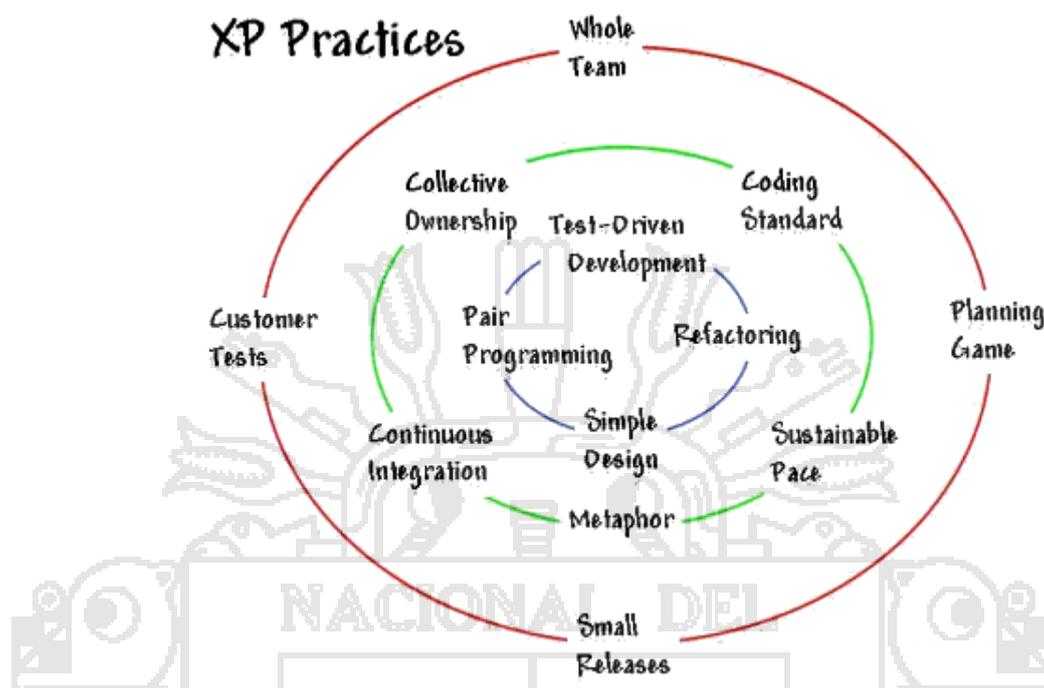
#### **n. Responsabilidad Aceptada**

Beck (2004:83) explica: “la responsabilidad no puede ser asignada, sólo puede ser aceptada, si alguien trata de darte responsabilidad, sólo tú decides si eres responsable de hacerlo o no”. Con la responsabilidad se adquiere autoridad. Cuando un experto en procesos te dice como debes trabajar pero no comparte el trabajo ni las consecuencias del mismo. Sólo las personas tienen la decisión de usar la retroalimentación para mejorar.

#### **2.5.6.3. Prácticas de la Programación Extrema**

Jeffries (2000:79) conceptualiza las prácticas de la programación extrema de la siguiente forma:

Figura N° 8  
Prácticas de la programación XP



#### a. El Juego de la Planificación

El alcance de la siguiente versión está definido por las consideraciones de negocios (prioridad de los módulos, fechas de entrega) y estimaciones técnicas (estimaciones de funciones, consecuencias). El objetivo del juego es maximizar el valor del software producido, La estrategia es poner en producción las características más importantes lo antes posible, Las Piezas clave son las Story Cards, Los Jugadores son los desarrolladores y el cliente y las Movidas son: Exploración, Selección y Actualización.

En esta fase, el usuario tendrá que escribir sus necesidades, definiendo las actividades que realizará el sistema. Se creará un documento llamado Historias del usuario (User Stories). Entre 20 y 80 historias (todo dependiendo de la complejidad del problema) se consideran suficientes para formar el llamado Plan de Liberación, el cual define de forma específica los tiempos de entrega de la aplicación para recibir retroalimentación por parte del usuario. Por regla general, cada una de las Historias del usuario suelen necesitar de una a tres semanas de desarrollo.

Son muy importantes y tienen que ser una constante las reuniones periódicas durante esta fase de planificación. Estas pueden ser a diario, con todo el equipo de desarrollo para identificar problemas, proponer soluciones y señalar aquellos puntos a los que se les ha de dar más importancia por su dificultad o por su punto crítico.

#### **b. Pequeñas Entregas**

Un sistema simple se pone rápidamente en producción. Periódicamente, se producen nuevas versiones agregando en cada iteración aquellas funciones consideradas valiosas para el cliente.

#### **c. Metáfora**

La metáfora es desarrollada por los programadores al inicio del proyecto, define una historia de cómo funciona el sistema completo. XP estimula historias, que son breves descripciones de un trabajo de un sistema en lugar de los tradicionales diagramas y modelos UML (Unified Modeling Language). La metáfora expresa la visión evolutiva del proyecto que define el alcance y propósito del sistema. Las tarjetas CRC (Clase, Responsabilidad y Colaboración) también ayudarán al equipo a definir actividades durante el diseño del sistema. Cada tarjeta representa una clase en la programación orientada a objetos y define sus responsabilidades (lo que ha de hacer) y las colaboraciones con las otras clases (cómo se comunica con ellas).

#### **d. Diseño Simple**

El sistema se diseña con la máxima simplicidad posible (YAGNY - "No vas a necesitarlo"), Se plasma el diseño en tarjetas CRC (Clase – Responsabilidad - Colaboración), no se implementan características que no son necesarias, con esta técnica, las clases descubiertas durante el análisis pueden ser filtradas para determinar qué clases son realmente necesarias para el sistema.

Beck (2004:147) al respecto menciona: “una buena idea puede ahorrar millones en costos y brindar una utilidad de millones”.

#### **e. Pruebas**

Los casos de prueba se escriben antes que el código. Los desarrolladores escriben pruebas unitarias y los clientes especifican pruebas funcionales.

Se tiene que establecer un período de pruebas de aceptación del programa (llamado también período de caja negra) donde se definirán las entradas al sistema y los resultados esperados de estas entradas. Es muy recomendable automatizar estas pruebas para poder hacer varias simulaciones del sistema en funcionamiento. Para hacer estas simulaciones automatizadas, se pueden utilizar Ambientes de Prueba (Unit testing frameworks). Un buen ejemplo de un ambiente de prueba es el JUnit para Java (JUnit, n.d.). Otros ambientes de pruebas para otros lenguajes como C, C++, JavaScript, XML y servicios Web, pueden encontrarse en: [www.xprogramming.com](http://www.xprogramming.com). (Jeffries, 1999)

#### **f. Refactorización**

Es posible reestructurar el sistema sin cambiar su comportamiento, por ejemplo eliminando código duplicado, simplificando funciones, mejorando el código constantemente, si el código se está volviendo complicado se debería modificar el diseño y volver a uno más simple. Refactoring (Modificar la forma del código sin cambiar su funcionamiento).

#### **g. Programación por Parejas**

El código es escrito por dos personas trabajando en el mismo computador. "Una sola maquina con un teclado y un mouse". uno de los principios más radicales y en el que la mayoría de gerentes de desarrollo pone sus dudas. Requiere que todos los programadores XP escriban su código en parejas, compartiendo una sola máquina. De acuerdo con los experimentos, este principio puede producir aplicaciones más buenas, de manera consistente, a iguales o menores costes. Aunque el pair-programming puede no ser para todo el mundo.

#### **h. Propiedad Colectiva**

Nadie es dueño de un modulo. Cualquier programador puede cambiar cualquier parte del sistema en cualquier momento, siempre se utilizan

estándares y se excluyen los comentarios, Los test siempre deben funcionar al 100% para realizar integraciones con todo el código permanentemente.

#### **i. Integración Continua**

Los cambios se integran en el código base, varias veces por día. Todos los casos de prueba se deben pasar antes y después de la integración, se dispone de una máquina para la integración y se realizan test funcionales en donde participa el cliente.

#### **j. Cuarenta Horas Semanales**

Cada Trabajador trabaja no más de 40 Horas por semana. Si fuera necesario hacer horas extra, esto no debería hacerse dos semanas consecutivas. Sin héroes, esto hace que se reduzca la rotación del personal y mejora la calidad del producto. Algunos estudios revelan que mientras más se trabaja se es menos productivo por tanto no tiene sentido trabajar más de la cuenta.

#### **k. Cliente en Casa**

El equipo de desarrollo tiene acceso todo el tiempo al cliente, el cual está disponible para responder preguntas, fijar prioridades, etc. Esto no siempre se consigue; un cliente muy Junior no sirve y un cliente muy sénior no es disponible. "Lo ideal es un cliente Analista". Esta estrategia también es importante para evitar la "resistencia al cambio" por parte de los usuarios y que estos se sientan comprometidos con el desarrollo y necesidad de implementar el proyecto de software.

#### **l. Estándares de Codificación**

Todo el código debe estar escrito de acuerdo a un estándar de codificación, como se observa los comentarios del código fuente, la nomenclatura de las variables (nombres, tipos, alcance, etc.), nomenclatura para funciones y procedimientos, etc.

#### **2.5.6.4. Objetivos de la Programación Extrema**

##### **a. La Satisfacción del Cliente**

El objetivo principal de la XP es la satisfacción del cliente. Se trata de dar al cliente lo que quiere y cuando quiere. Por tanto, se debe responder rápidamente a las necesidades del cliente por lo que la programación extrema opta por realizar periodos de despliegue mucho más cortos para que el usuario pueda validar si realmente es la solución que requiere y de esta forma retroalimentar y garantizar la calidad del producto, aunque realice cambios en fases avanzadas del proyecto. (Ambler, 2002:75)

##### **b. Potenciar el Trabajo en Grupo, Todos están Involucrados en el Desarrollo del Software**

Otro de los objetivos es el trabajo en grupo. Tanto los jefes del proyecto, clientes y desarrolladores forman parte del equipo y deben estar involucrados en el desarrollo, lo cual se refuerza con la propiedad común del código pues si se presenta un error éste debe ser solucionado por todo el equipo y no individualizar particularmente la responsabilidad en algún miembro del equipo. (Ambler, 2002:75)

#### **2.5.6.5. Valores de la Programación Extrema**

Los valores de la programación extrema son: “simplicidad, comunicación, retroalimentación, coraje y respeto”. A continuación detallamos cada uno de estos valores (Beck, 2004:31):

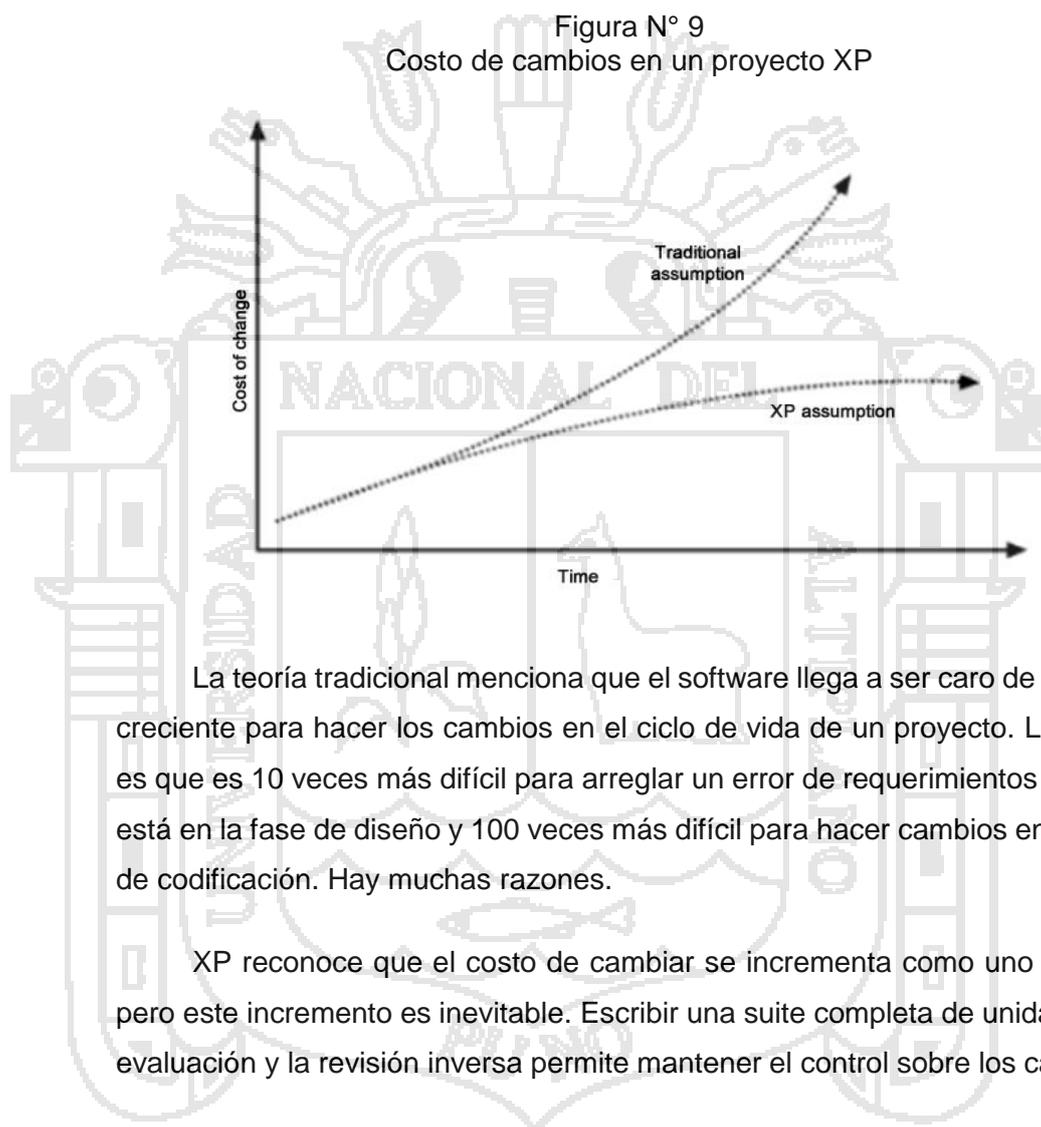
##### **a. Simplicidad**

La simplicidad es la base de la programación extrema. Se simplifica el diseño para agilizar el desarrollo y facilitar el mantenimiento. Un diseño complejo del código junto a sucesivas modificaciones por parte de diferentes desarrolladores hace que la complejidad aumente exponencialmente. Para mantener la simplicidad es necesaria la refactorización del código, ésta es la manera de mantener el código simple a medida que crece. También se aplica la simplicidad en la documentación, de esta manera el código debe comentarse en su justa medida, intentando eso sí que el código esté auto documentado. Para ello se deben elegir adecuadamente los nombres de las variables, métodos y clases. Aplicando la simplicidad junto con la autoría colectiva del

código y la programación por parejas se asegura que cuanto más grande se haga el proyecto, todo el equipo conocerá más y mejor el sistema completo.

Mantener el código simple hace los cambios de código simple, además adoptar técnicas simples para comunicar requisitos y el progreso de seguimiento maximiza las posibilidades que el equipo siga adecuadamente el proceso. La mayoría se enfoca en soluciones simples.

Figura N° 9  
Costo de cambios en un proyecto XP



La teoría tradicional menciona que el software llega a ser caro de manera creciente para hacer los cambios en el ciclo de vida de un proyecto. La teoría es que es 10 veces más difícil para arreglar un error de requerimientos cuando está en la fase de diseño y 100 veces más difícil para hacer cambios en la fase de codificación. Hay muchas razones.

XP reconoce que el costo de cambiar se incrementa como uno espera, pero este incremento es inevitable. Escribir una suite completa de unidades de evaluación y la revisión inversa permite mantener el control sobre los cambios.

## b. Comunicación

El incremento de la comunicación en el proceso de desarrollo de software gracias a esta metodología ha sido una de las claves para su éxito. La comunicación se realiza de diferentes formas para los programadores el código comunica mejor cuanto más simple sea. Si el código es complejo hay que esforzarse para hacerlo inteligible. El código auto documentado es más fiable que los comentarios ya que éstos últimos pronto quedan desfasados con el

código a medida que es modificado. Los programadores se comunican constantemente gracias a la programación por parejas. La comunicación con el cliente es fluida ya que el cliente forma parte del equipo de desarrollo. El cliente decide que características tienen prioridad y siempre debe estar disponible para solucionar dudas.

### **c. Retroalimentación**

Al estar el cliente integrado en el proyecto su opinión sobre los resultados se conoce en tiempo real. Al realizarse ciclos muy cortos tras los cuales se entregan prototipos, se minimiza el tener que rehacer partes que no cumplen con los requisitos que requiere el usuario y ayuda a los programadores a centrarse en lo que es más importante lográndose desarrollar con mayor celeridad módulos que pueden ir automatizando procesos de la empresa sin tener que esperar a la finalización del proceso de desarrollo de software, en caso contrario los problemas que derivan de tener ciclos muy largos que se resumen en meses de trabajo pueden tirarse por la borda debido a cambios en los criterios del cliente o malentendidos por parte del equipo de desarrollo. El código también es una fuente de retroalimentación gracias a las herramientas de desarrollo. Ejecutar las pruebas unitarias frecuentemente permite descubrir fallos debidos a cambios recientes en el código.

### **d. Coraje**

Para los programadores tradicionales muchas de las innovaciones de la metodología XP pueden ser difíciles de adoptarse como es el caso de la programación en parejas porque parece que la productividad se fuese a reducir a la mitad ya que solo la mitad de los programadores está escribiendo código. Hay que ser valiente para confiar en que la programación por parejas beneficia la calidad del código sin repercutir negativamente en la productividad. La simplicidad es uno de los principios más difíciles de adoptar. Se requiere coraje para implementar las características que el cliente quiere ahora sin caer en la tentación de optar por un enfoque más flexible que permita futuras modificaciones. No se debe emprender el desarrollo de grandes marcos de trabajo (frameworks) mientras el cliente espera. En ese tiempo el cliente no recibe noticias sobre los avances del proyecto y el equipo de desarrollo no recibe retroalimentación para saber si va en la dirección correcta. La forma de

construir marcos de trabajo es mediante la refactorización del código en sucesivas aproximaciones.

#### e. Respeto

Esta última característica fue añadida recién en la segunda edición del libro de Kent Beck, el respeto se manifiesta de varias formas. Los miembros del equipo se respetan los unos a otros, porque los programadores no pueden realizar cambios que hacen que las pruebas existentes fallen o que demore el trabajo de sus compañeros. Los miembros respetan su trabajo porque siempre están luchando por la alta calidad en el producto y buscando el diseño óptimo o más eficiente para la solución a través de la refactorización del código. Los miembros del equipo respetan el trabajo del resto no haciendo menos a otros, si no orientándolos a realizarlo mejor, obteniendo como resultado una mejor autoestima en el equipo y elevando el ritmo de producción en el equipo.

#### 2.5.6.6. Roles en la programación extrema

Aunque en otras fuentes de información aparecen algunas variaciones y extensiones de roles XP, en este apartado se describe los roles de acuerdo con la propuesta original de Beck. (Beck, 2000:4)

- a. Programador: Escribe las pruebas unitarias y produce el código del sistema. Debe existir una comunicación y coordinación adecuada entre los programadores y otros miembros del equipo.
- b. Cliente: Escribe las historias de usuario y las pruebas funcionales para validar su implementación. Además, asigna la prioridad a las historias de usuario y decide cuáles se implementan en cada iteración centrándose en aportar mayor valor al negocio. El cliente es sólo uno dentro del proyecto pero puede corresponder a un interlocutor que está representando a varias personas que se verán afectadas por el sistema.
- c. Encargado de pruebas (Tester): Ayuda al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.
- d. Encargado de seguimiento (Tracker): Proporciona realimentación al equipo en el proceso XP. Su responsabilidad es verificar el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, comunicando los

resultados para mejorar futuras estimaciones. También realiza el seguimiento del progreso de cada iteración y evalúa si los objetivos son alcanzables con las restricciones de tiempo y recursos presentes. Determina cuándo es necesario realizar algún cambio para lograr los objetivos de cada iteración.

- e. Entrenador (Coach): Es responsable del proceso global. Es necesario que conozca a fondo el proceso XP para proveer guías a los miembros del equipo de forma que se apliquen las prácticas XP y se siga el proceso correctamente.
- f. Consultor: Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto. Guía al equipo para resolver un problema específico.
- g. Gestor (Big boss): Es el vínculo entre clientes y programadores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas. Su labor esencial es de coordinación.

#### **2.5.6.7. Fases de Desarrollo**

Las fases de desarrollo están íntimamente ligadas con los principios de XP como se puede apreciar en el cuadro N° 2.1:

Cuadro N° 1  
Fases de desarrollo y principios XP

Fases	Principio
Planificación	El juego de la planificación
	Entregas pequeñas
	Metáfora
Diseño	Diseño simple
Codificación	Refactorización
	Programación en pares
	Propiedad colectiva
	Integración continua
	Semanas de 40 horas
	Cliente in situ
	Estándares de programación
Pruebas	Pruebas

#### A. Planificación del Proyecto

XP plantea la planificación como un permanente dialogo entre las partes la empresarial (deseable) y la técnica (posible), en esta fase se utilizan:

- a. Historias de usuario: El primer paso de cualquier proyecto que siga la metodología X.P es definir las historias de usuario con el cliente. Las historias de usuario tienen la misma finalidad que los casos de uso pero con algunas diferencias: Constan de 3 ó 4 líneas escritas por el cliente en un lenguaje no técnico sin hacer mucho hincapié en los detalles; no se debe hablar ni de posibles algoritmos para su implementación ni de diseños de base de datos adecuados, etc.

Son utilizadas para estimar tiempos de desarrollo de la parte de la aplicación que describen. También se utilizan en la fase de pruebas, para verificar si la aplicación cumple con lo que especifica la historia de usuario.

Cuando llega la hora de implementar una historia de usuario, el cliente y los desarrolladores se reúnen para concretar y detallar lo que tiene que hacer dicha historia. El tiempo de desarrollo ideal para una historia de usuario es entre 1 y 3 semanas.

Jeffries (1999:37) define historias de usuario de la siguiente forma: “definen los requerimientos con historias escritas en tarjetas”, los clientes tienen el

derecho de sacar el máximo provecho de cada momento de programación y los programadores tienen el derecho de conocer lo que es requerido. Estos dos derechos vienen juntos en las historias de usuario.

Cada historia de usuario es una pequeña descripción del comportamiento del sistema, desde el punto de vista del usuario del sistema. En XP el sistema es especificado íntegramente a través de historias de usuario.

El análisis puede ser descrito libremente como el proceso de averiguar lo que quiere el cliente. El análisis puede ser hecho muy formalmente, con objetos y diagramas o informalmente. XP elige el análisis informal de preguntar al cliente.

El análisis en XP no se limita al inicio del proyecto, lo que se denomina “conducido por análisis”. En XP se hace análisis todo el tiempo. Las historias de usuario son el medio de análisis, el medio de comunicación entre el cliente y el programador.

XP reconoce que el cliente debe aprender a pedir lo que necesita, basado en lo que se ha hecho rápidamente, manteniendo pequeñas historias de usuario, haciendo liberaciones frecuentes y manteniendo clientes y programadores juntos, el enfoque XP para analizar asegura que lo que sea necesario aprender pueda ser retroalimentado en el proceso de desarrollo y asegurando que el resultado sea lo más cercano posible a lo deseado, un esquema de historia de usuario puede verse en el cuadro N° 2.2.

Cuadro N° 2  
Ejemplo de historia de usuario

Historia de Usuario	
Número:	Nombre:
Usuario:	
Modificación de Historia Número:	Iteración Asignada:
Prioridad en el Negocios: Alta (Alta / Media /Baja)	Puntos estimados:
Riesgo en desarrollo: Baja (Alta / Media /Baja)	Puntos reales:
Descripción:	
Observaciones:	

Beck (2004) recomienda que estas historias de usuario sean llenadas a mano incluso en algunos proyectos mantienen sus historias de usuario pegadas en una pared la cual denominan "pared Gantt", las historias de usuario también pueden ser asociadas a pruebas de aceptación con lo que se confirma si la historia ha sido implementada correctamente.

- b. Planificación de entregas (Release planning): Tras definir las historias de usuario es necesario crear un plan de entregas, en inglés "Release planning", donde se indiquen las historias de usuario que se implementarán para cada versión de la aplicación y las fechas en las que se publicarán dichas versiones. Un "Release planning" es una planificación donde los desarrolladores y clientes establecen los tiempos de implementación ideales de las historias de usuario, la prioridad con la que serán implementadas y las historias de usuario que serán implementadas en cada versión del programa.

Después de un "Release planning" tienen que estar claros estos cuatro factores: los objetivos que se deben cumplir (que son principalmente las historias que se deben desarrollar en cada versión), el tiempo que tardarán en desarrollarse y publicarse las versiones de la aplicación, el número de personas que trabajarán en el desarrollo y cómo se evaluará la calidad del trabajo realizado.

- c. Iteraciones: Todo proyecto que siga la metodología XP se ha de dividir en iteraciones de aproximadamente 3 semanas de duración. Al comienzo de cada iteración los clientes deben seleccionar las historias de usuario definidas en el "Release planning" que serán implementadas. También se seleccionan las historias de usuario que no pasaron el test de aceptación que se realizó al terminar la iteración anterior. Estas historias de usuario son divididas en tareas de entre 1 y 3 días de duración cada una y serán asignadas a los programadores.
- d. Velocidad del proyecto: La velocidad del proyecto es una medida que representa la rapidez con la que se desarrolla el mismo; estimarla es muy sencillo, basta con contar el número de historias de usuario que se pueden implementar en una iteración; de esta forma, se sabrá el cupo de historias que se pueden desarrollar en las distintas iteraciones.

Con la velocidad del proyecto controlaremos si todas las tareas se pueden desarrollar en el tiempo dispuesto para la iteración. Es conveniente reevaluar esta medida cada 3 ó 4 iteraciones y si se aprecia que no es adecuada hay que negociar con el cliente un nuevo "Release Planning".

- e. Reuniones diarias: Es necesario que los desarrolladores se reúnan diariamente y expongan sus problemas, soluciones e ideas de forma conjunta. Las reuniones tienen que ser fluidas y todo el mundo debe tener voz y voto.

Las prácticas XP que en esta fase se desarrollan son:

- Juego de Planificación
- Pequeñas versiones
- Metáfora

## **B. Diseño del Proyecto**

El diseño incremental es una forma de lograr funcionalidad rápida y permanentemente en el ciclo de vida del proyecto. XP promueve el incrementalismo del diseño a sus límites sugiriendo que los proyectos se ejecutan más finamente si el diseño es parte del día a día. El diseño es la actividad que permite dar valor al software. A diferencia del mundo físico en las áreas del software podemos replicar elementos sin ningún costo, cuando se crean nuevas y beneficiosas relaciones entre elementos, podemos propagar

estas relaciones a todos los elementos existentes, una buena idea en los proyectos de desarrollo de software puede ahorrar millones en costos, las principales características de la fase de diseño en XP son (Beck, 2000:147):

- Glosarios de términos: Usar una correcta especificación de los nombres de clases, métodos y propiedades ayudará a comprender el diseño y facilitará futuras ampliaciones y la reutilización del código.
- Riesgos: Si surgen problemas potenciales durante el diseño, XP sugiere utilizar una pareja de desarrolladores para que investiguen y reduzcan al máximo el riesgo que supone ese problema.
- Funcionalidad extra: Nunca se debe añadir funcionalidad extra al programa aunque se piense que en un futuro será utilizada. Sólo el 10% de la misma es utilizada lo que demuestra que el desarrollo de funcionalidad extra es un desperdicio de tiempo y recursos.
- Refactorizar: Consiste en mejorar el código modificando su estructura sin alterar su funcionalidad. Refactorizar supone revisar de nuevo la codificación para procurar optimizar su funcionamiento.
- Es muy común reutilizar código ya creado que suele contener funcionalidades que no serán utilizadas y diseños obsoletos; esto es un grave error porque puede generar código inestable y mal diseñado; por este motivo, es necesario refactorizar cuando se va a reutilizar código.
- Tarjetas CRC: El uso de las tarjetas C.R.C (Class, Responsibilities and Collaboration) permiten al programador centrarse y apreciar el desarrollo orientado a objetos olvidándose de los malos hábitos de la programación procedural clásica. Las tarjetas C.R.C representan objetos; la clase a la que pertenece el objeto se puede escribir en la parte de arriba de la tarjeta, en una columna a la izquierda se pueden escribir las responsabilidades u objetivos que debe cumplir el objeto y a la derecha, las clases que colaboran con cada responsabilidad.

Cuadro N° 3  
Ejemplo de tarjeta CRC

Nombre de la clase	
Responsabilidades:	Colaboradores:

La práctica XP que en esta fase se desarrollan es diseño simple.

### C. Codificación

Una característica importante de la metodología XP es que el cliente es una parte más del equipo de desarrollo; su presencia es indispensable en las distintas fases de XP, a la hora de codificar una historia de usuario su presencia es aún más necesaria. No olvidemos que los clientes son los que crean las historias de usuario y negocian los tiempos en los que serán implementadas. Antes del desarrollo de cada historia de usuario el cliente debe especificar detalladamente lo que ésta hará y también tendrá que estar presente cuando se realicen las pruebas que verifiquen que la historia implementada cumple la funcionalidad especificada.

La codificación debe hacerse ateniendo a estándares y patrones de codificación ya creados. Programar bajo estándares mantiene el código consistente y facilita su comprensión y la escalabilidad. Crear pruebas que validen el funcionamiento de los distintos códigos implementados nos ayudará a desarrollar dicho código. Crear estos test antes nos ayuda a saber qué es exactamente lo que tiene que hacer el código a implementar y sabremos que una vez implementado pasará dichos test sin problemas ya que dicho código ha sido diseñado para ese fin. Se puede dividir la funcionalidad que debe cumplir una tarea a programar en pequeñas unidades, de esta forma se crearán primero los test para cada unidad y a continuación se desarrollará dicha unidad, así poco a poco conseguiremos un desarrollo que cumpla todos los requisitos especificados.

XP sugiere un modelo de trabajo usando repositorios de código donde los programadores publican cada pocas horas sus códigos implementados y corregidos junto a los test que deben pasar. De esta forma el resto de programadores que necesiten códigos ajenos trabajarán siempre con las

últimas versiones. Para mantener un código consistente, publicar un código en un repositorio es una acción exclusiva para cada pareja de programadores.

Los principios que deben tomarse en cuenta en esta fase son:

- a. Refactorización
- b. Programación en parejas
- c. Propiedad colectiva
- d. Integración continua
- e. Semana de 40 horas
- f. Cliente in situ
- g. Estándares de programación

#### **D. Pruebas**

Uno de los pilares de la metodología XP es el uso de test para comprobar el funcionamiento del código que estamos desarrollando. El uso de los test en XP es el siguiente:

- Se deben crear los test con frameworks específicos para tests (JUnit, por ejemplo para Java).
- Hay que someter a test las distintas clases del sistema omitiendo los métodos más triviales.
- Se deben crear los test que se aplicarán a una clase/método antes de implementarla; en el apartado anterior se explicó la importancia de crear antes los test que el código.

Un punto importante es crear test que no tengan ninguna dependencia del código que en un futuro evaluará. Hay que crear los test abstrayéndose del futuro código, de esta forma aseguraremos la independencia del test respecto al código que evalúa. Como se comentó anteriormente los distintos test se deben subir al repositorio de código acompañados del código que verifican. Ningún código puede ser publicado en el repositorio sin que haya pasado su test de funcionamiento, de esta forma, aseguramos el uso colectivo del código (explicado en el apartado anterior).

El uso de los test es adecuado para observar la refactorización. Los test permiten verificar que un cambio en la estructura de un código no tiene porqué

cambiar su funcionamiento. Para finalizar, un nuevo concepto. Las pruebas de aceptación son pruebas de caja negra definidas por el cliente para cada historia de usuario, y tienen como objetivo asegurar que las funcionalidades del sistema cumplen con lo que se espera de ellas. En efecto, las pruebas de aceptación corresponden a una especie de documento de requerimientos en XP, ya que marcan el camino a seguir en cada iteración, indicándole al equipo de desarrollo hacia donde tiene que ir y en qué puntos o funcionalidades debe poner el mayor esfuerzo y atención. “Las pruebas de aceptación permiten al cliente saber cuando el sistema funciona, y que los programadores conozcan qué es lo que resta por hacer.” (Jeffries et al., 2000: 45)

Las pruebas de aceptación tienen una importancia crítica para el éxito de una iteración. Por lo tanto el “evaluador” debe tenerlas lo antes posible a partir del comienzo de la iteración, y lograr que el cliente las apruebe para poder presentárselas cuanto antes al equipo de desarrollo.

Cuadro N° 4  
Ejemplo de caso de prueba de aceptación

Caso de prueba de aceptación	
Código	Historia de usuario (Número y Nombre)
Nombre:	
Descripción:	
Condiciones de ejecución:	
Entrada / Pasos de ejecución:	
Resultado esperado:	
Evaluación de la prueba:	

Posterior a la identificación de los casos de prueba el equipo debe de identificar las tareas de ingeniería cuya ficha de ejemplo se puede observar en el cuadro 2.5.

Cuadro N° 5  
Ejemplo de tarea de ingeniería

Tarea de Ingeniería	
Número de tarea:	Historia de usuario (Número y Nombre):
Nombre de tarea:	
Tipo de tarea: Desarrollo/Corrección/Mejora/Otra	Puntos estimados:
Fecha de inicio:	Fecha de finalización:
Programador responsable:	
Descripción:	

## 2.5.7. Elementos Básicos de una Tienda Electrónica

### 2.5.7.1. Pilares de una Tienda Online

Para la creación de una Tienda Online, se debe tener muy claro cuál es el público objetivo y qué se le quiere transmitir.

Se debe buscar la sencillez, un buen catálogo de productos y contenidos, así como la búsqueda de relaciones con otras Webs para conseguir notoriedad en Internet.

Pero ante todo, para realizar una Tienda Online, se debe tener en cuenta que los elementos que la conforman deben sustentarse en 3 pilares clave (AECCEM, 2009:29-31):

- Diseño.
- Usabilidad.
- Accesibilidad.

#### A. Diseño

El diseño de la página es un factor crucial. Debe ser intuitivo y accesible.

Algunas recomendaciones acerca del diseño son:

- La tienda online debe estar decorada con los propios contenidos de tal manera que los productos sean los que destaquen por encima del diseño.

- Combinar colores para crear una visión cálida y agradable.
- Los menús de navegación de las diferentes páginas que conforman la tienda online deben seguir un orden establecido, así conseguimos no despistar al usuario.

Disponer de un diseño orientado a la optimización de buscadores: la tienda online debe tener el contenido de forma que los buscadores puedan localizarnos.

### **B. Usabilidad**

Se dice que una página Web es “usable” cuando muestra todo el contenido de una forma clara y sencilla de entender por el usuario, favoreciendo la compra y consiguiendo la satisfacción del usuario en su experiencia de navegación.

- El catálogo de productos debe estar visible desde el primer momento.
- Fácil acceso a los productos mediante una clara navegación por categorías y subcategorías.
- El carrito de la compra siempre debe estar visible.
- El Proceso de la compra debe estar enfocado al producto de una forma clara y rápida. A su vez debe ser fácil y rápido para tratar de finalizar el mayor número de ventas posible.
- Disponer de un potente buscador que nos ofrezca la posibilidad de acceder a nuestro catálogo de producto por distintos criterios (precio, fecha, orden alfabético...).
- Facilitar el acceso a apartados de servicio de la tienda como Información de contacto, forma de comprar, condiciones generales, etc.
- La ficha de producto debe ser detallada e idealmente debe estar relacionada con otros productos similares para potenciar la compra indirecta

### **C. Accesibilidad**

Una tienda online es accesible si sus productos, contenidos y servicios pueden ser accedidos por el mayor número posible de personas. En este sentido planteamos las siguientes recomendaciones:

- El catálogo de productos debe ser accesible por categorías y escaparates comerciales. Es importante la correcta clasificación de productos utilizando categorías como novedades, productos más vendidos, productos en oferta, etc.
- Es también deseable mostrar productos destacados con un diseño algo distinto al resto de productos en especial si queremos promocionar la venta de un producto determinado.
- En todos los casos tendremos que escoger el tamaño del texto óptimo y destacando aquella información más relevante: nombre producto, botón comprar, precio, etc.
- Uso correcto del etiquetado de las imágenes favoreciendo su indexación.

#### **2.5.7.2. Elementos de una Tienda Online**

Como elementos básicos en una Tienda Online podemos destacar (AECCEM, 2009:32-38):

##### **A. Catálogo de Productos**

En la fase de creación de una tienda online una de las principales tareas que se realizan es dar forma al catálogo de productos. Es interesante el uso de escaparates de productos con un mensaje claro y directo al cliente como pueden ser: los más vendidos, últimas novedades, lista top, etc.

Factores en los que influye el catálogo de productos:

- Imagen de producto: la venta por internet es al fin y al cabo venta por catálogo.
- Productos en venta: no siempre los productos que existen en el portfolio de la empresa pueden ser vendidos en internet.

- Atributos o características de los productos: la categorización de los productos es un elemento clave en la organización del catálogo en la tienda.
- Descripción del producto: la decisión de compra en parte depende de una descripción lo más completa posible, lo cual no significa extensa.
- Información de producto como elemento de diferenciación y posicionamiento en buscadores: tanto los atributos del producto como su descripción, entre otros elementos, serán críticos en el posicionamiento de la tienda en los buscadores, los cuales actualmente son prácticamente el principal generador de visitas a las tiendas online.
- Profundidad de catálogo: el número de productos a vender puede condicionar el tipo de solución tecnológica a implantar. Es importante informarse bien de si la solución elegida se adapta a la cantidad y funcionamiento (productos configurables por color y tallas... etc).

## **B. Carrito de la Compra**

La cesta o carrito de la compra es un elemento indispensable en la Tienda Online. Este elemento debe ofrecer la posibilidad de añadir, eliminar o modificar los productos que durante la navegación hemos ido seleccionando e incorporando.

Es aconsejable, para que el usuario no pierda de vista en ningún momento lo que lleva comprado, que esté visible en todas las páginas de la Tienda Online, o al menos en aquellas donde se puedan seguir añadiendo productos.

De igual forma, se debe dar la posibilidad, con un solo clic, que el cliente pueda visualizar de una forma clara:

- Las referencias compradas (especificando la cantidad).
- Los gastos de envío.
- Impuestos aplicables de forma directa.
- Importe total del pedido

### **C. Mecanismos de Promoción y Ofertas**

Uno de los factores más importantes que atraen a los clientes de una Tienda Online e influyen en la decisión de compra es el precio. El precio de los productos debe estar siempre bien visible.

Pero alrededor de una captación de clientes por el precio tenemos varias estrategias comerciales que pasan por la aplicación de promociones o descuentos especiales sobre los productos.

Las promociones y las ofertas deben comunicarse de una forma clara, resaltando el precio de la oferta y mostrando junto a él, el precio no rebajado.

La Tienda Online debe disponer de un módulo de descuentos y promociones que permitan definir de forma puntual descuentos personalizados por categorías de producto, tipos de cliente, etc tanto a nivel de porcentaje como cantidad fija a aplicar sobre el precio de venta del producto.

### **D. Motor de Búsqueda**

Con la idea de facilitar al cliente encontrar nuestros productos y servicios es muy interesante e incluso indispensable disponer de un potente motor de búsqueda o buscador integrado que permita la búsqueda de productos por diversos criterios y parámetros de búsqueda.

Criterios de interés son las búsquedas por palabras clave, precio, categoría, nombre, etc.

El motor de búsqueda siempre debería ofrecer resultados para dar la sensación de robustez y buen funcionamiento. En caso de no poder mostrar resultados exactos a nuestra búsqueda es deseable que el buscador ofrezca otras sugerencias relacionadas con el producto buscado. De esta forma no comentemos el error de que el buscador parezca que no funciona, o que nuestro catálogo de productos es muy reducido.

### **E. Proceso de compra**

Se considera un buen proceso de compra aquel que es directo y se encuentra guiado mediante mensajes de información.

Se suele aconsejar que el registro de usuarios sea opcional. No es aconsejable que para efectuar una compra obliguemos a los visitantes a registrarse previamente ya que muchos compradores potenciales pueden ser sólo ocasionales. En cualquier caso, la información que solicitemos para que un cliente pueda realizar un pedido en nuestra tienda debe ser la mínima imprescindible.

Idealmente debemos plantearnos que la duración del proceso de compra implique el menor número posible de pasos y clicks de ratón (5 clicks de ratón o menos es un número idóneo).

Es importante que durante el proceso de compra se muestre información complementaria que transmita confianza a los usuarios, y que a éstos al terminar la compra no les quede ninguna duda respecto a la compra que acaban de hacer y respecto a:

- Gastos de envío asociados a la transacción
- Dirección de entrega del pedido
- Período de devolución y plazos de entrega
- Disponibilidad de los productos elegidos y plazo de envío de los mismos

#### **F. Medios de Pago**

El momento del pago de los artículos que el cliente ha ido añadiendo en el carrito de la compra es uno de los más críticos dentro de los procesos de una Tienda Online pues, estadísticamente, es en ese momento en el que se dan más abandonos.

Los medios de pago que podemos encontrar habitualmente en las tiendas de Internet son los siguientes:

- Contra reembolso
- Transferencia bancaria
- Domiciliación bancaria
- Tarjeta de crédito (A través de TPV virtual)
- Medios Alternativos: PayPal, Saftpay, Allopass

## **G. Impuestos**

Los precios de los productos o servicios que se publiquen en la página web de la Tienda Online deben ser los precios finales completos, incluidos todos los impuestos, tasas y demás gravámenes aplicables.

Asimismo, la Tienda Online debe tener en cuenta e informar de forma clara al consumidor de las posibles tasas, impuestos o gravámenes que puedan ser de aplicación en función del lugar de residencia del consumidor o del lugar de entrega del producto o servicio, antes de la realización del pedido. Finalmente, la tienda online en la factura o en el e-mail de confirmación del pedido deberá señalar el tipo y la cuota del impuesto, tasa o gravamen aplicados.

## **H. Logística**

Al igual que los impuestos, la Tienda Online debe ofrecer de forma clara el coste por enviarnos la mercancía adquirida. En este sentido, debe permitir configurar una matriz de gastos de envío para cada transportista y ofrecer la posibilidad de calcular los gastos de envío en función de:

- Cálculo de los gastos de envío empleando una tarifa fija.
- Cálculo de los gastos de envío en función del volumen de compra.
- Cálculo de los gastos de envío en función del peso de la compra, y coste adicional por cantidades excedidas.
- Otros

## **I. Información corporativa**

Dentro de una Tienda Online no toda la información deben ser escaparates de productos. Es necesario mostrar a nuestro comprador información que haga que nos conozca, que entienda la filosofía de negocio, trayectoria empresarial, etc,. Este tipo de información transmite transparencia, fiabilidad y confianza hacia nuestros clientes.

Esta información suele estructurarse en secciones como:

- Quiénes somos
- Qué ofrecemos
- Aviso Legal y Política de Privacidad

- Información de contacto
- Dónde estamos
- Condiciones de compra y contratación
- Preguntas Frecuentes
- Otros

## **J. Registro y Área de Usuario**

El registro de clientes es un aspecto muy importante de una Tienda Online tanto para que nuestros clientes realicen sus pedidos como a la hora de poder llevar a cabo acciones posteriores de fidelización y captación de nuevos clientes partiendo de una base de datos de clientes extensa.

En la zona de registro de un comercio electrónico los clientes deben poder consultar, al menos:

- Datos personales
- Datos de envío y facturación
- Estado del pedido realizado e histórico de pedidos
- Boletín de novedades

Finalmente, señalar que la aplicación informática en la que nos apoyemos para la gestión de nuestro comercio electrónico debe ser capaz de aportarnos toda la información que necesitemos para el seguimiento y control de nuestro negocio.

### **2.5.8. Mecanismos de Seguridad en los Medios de Pago Electrónicos**

Los mecanismos o requisitos de seguridad pueden variar ligeramente de un sistema de pago a otro dependiendo tanto de las características propias del sistema como por la confianza y seguridad que ofrezca el entorno donde va a interactuar el sistema. En cualquier caso, se puede decir que los requisitos de seguridad de un sistema de pago electrónico son los siguientes (AECM. 2009:45):

#### **A. Autenticación de los Datos**

En todo método de pago, tanto el comprador como la tienda deben identificarse para comprobar que no existe fraude. En un sistema de pago a través de TPV, existe una entidad certificadora que durante la operación de

pago garantiza la autenticidad de la transacción validando la información de tarjeta de crédito y titular de la misma.

La autenticación de otros datos que puedan afectar a la legalidad de la compra realizada se garantiza gracias a protocolos criptográficos de autenticación.

## **B. Disponibilidad y Fiabilidad**

Una característica fundamental de los sistemas electrónicos de pago es la alta disponibilidad y fiabilidad de las operaciones. Es un elemento clave en el proceso de pago de una transacción, ya que en ese instante no puede existir una pérdida del servicio.

No obstante, podría producirse algún fallo en el sistema, y una peculiaridad de las transacciones es que o se realizan correctamente o no se realizan. No hay un punto intermedio de recuperación o estado intermedio. De esta forma, la fiabilidad del sistema es total.

Por tanto, un sistema fiable y disponible no solo depende de que sus transacciones lo sean, sino de que toda la arquitectura y elementos del sistema cumplan los mismos requisitos.

## **C. Integridad**

Otro aspecto importante en todo sistema de pago electrónico es la integridad de los datos intercambiados entre los actores. Es decir, no puede haber forma de manipular la información o alterarla mientras se realiza una transacción electrónica.

Para ello existen mecanismos de seguridad como son códigos de autenticación, firma digital, comunicación segura, etc.

## **D. Confidencialidad**

Los datos que se envían en una operación electrónica no pueden ser visibles para terceros, de ahí que se empleen técnicas de encriptado y cifrado de la información por parte de los actores implicados en la transacción.

En este sentido, el comercio se dota de un Certificado de Seguridad emitido por una entidad colaboradora certificadora, que permite el cifrado,

encriptado y envío de la información de forma segura, empleando un canal de comunicación seguro (SSL).

Únicamente los extremos de la transacción, comercio y destinatario (entidad bancaria, autenticadora, etc) son conocedores de la información de forma clara tras un proceso inverso de desencriptado.

### **2.5.9. Arquitectura MVC (Modelo Vista Controlador)**

El patrón de diseño Modelo Vista Controlador (MVC) se utiliza para el diseño de aplicaciones en tres capas. Esto debido a que la lógica de una interfaz de usuario cambia con más frecuencia que los almacenes de datos y la lógica de negocio.

La arquitectura MVC es uno de varios modelos de infraestructura WebApp sugeridos para desacoplar la interfaz del usuario de la funcionalidad y el contenido de información de la WebApp. (Pressman, 2005: 589)

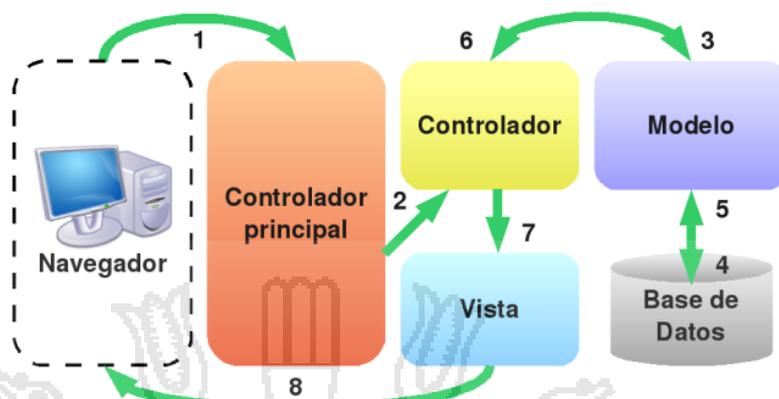
Si se realiza un diseño desordenado, es decir, algo que mezcle los componentes de interfaz y de negocio, entonces la consecuencia será que, cuando se necesite cambiar la interfaz, se tendrá que modificar trabajosamente los componentes de negocio, provocando un mayor trabajo y más riesgo de error.

Entonces el patrón Modelo Vista Controlador trata de realizar un diseño que desacople la vista del modelo, con la finalidad de mejorar la reusabilidad. De esta forma las modificaciones en las vistas impactan en menor medida en la lógica de negocio o de datos.

Elementos del patrón MVC:

- Modelo: datos y reglas de negocio.
- Vista: muestra la información del modelo al usuario.
- Controlador: gestiona las entradas del usuario.

Figura N° 10  
Arquitectura modelo vista controlador



Un modelo puede tener diversas vistas, cada una con su correspondiente controlador.

A continuación se detalla cada componente:

#### A. El modelo

El modelo es el responsable de:

- Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.
- Define las reglas de negocio (la funcionalidad del sistema). Un ejemplo de regla puede ser: "Si la mercancía pedida no está en el almacén, consultar el tiempo de entrega estándar del proveedor".
- Lleva un registro de las vistas y controladores del sistema.
- Si estamos ante un modelo activo, notificará a las vistas los cambios que en los datos pueda producir un agente externo.

#### B. El controlador

El controlador es responsable de:

- Recibir los eventos de entrada (un clic, un cambio en un campo de texto, etc.)

- Además contiene reglas de gestión de eventos, del tipo “SI Evento A, entonces Acción B”. Estas acciones pueden suponer peticiones al modelo o a las vistas. Una de estas peticiones a las vistas puede ser una llamada al método “Actualizar ()”. Una petición al modelo puede ser “Obtener\_Empleado\_PorCodigo(codigo\_Empleado)”.

### C. La Vista

La vista es responsable de:

- Recibir datos del modelo y mostrarlos al usuario.
- Tienen un registro de su controlador asociado (normalmente porque además lo instancia).
- Pueden dar el servicio de “Actualización()”, para que sea innovado por el controlador o por el modelo (cuando es un modelo activo que informa de los cambios en los datos producidos por otros agentes).

Un ejemplo clásico es el de la información de una base de datos que se puede presentar de diversas formas: diagrama de torta, de barras, tabular otro ejemplo de MVC con un modelo pasivo (aquel que no notifica cambios en los datos) es la navegación web, que responde a las entradas del usuario, pero no detecta los cambios en datos del servidor.

#### 2.5.10. Web 2.0

En otoño de 2001 se produjo el estallido de la burbuja tecnológica, lo que provocó que muchas empresas desaparecieran del mercado o vieran fuertemente reducidas sus expectativas, pero eso no pasó con todas. Algunas sobrevivieron y otras aparecieron con nuevas ideas y nuevos modelos de negocio. En el año 2004 dos empresas del sector informático, O'Reilly y MediaLive International, realizaron un brainstorming para analizar aquellas que habían superado la crisis y poder identificar las características claves de éxito. A las que no lo superaron las denominaron “web 1.0” y a las que sí, o estaban teniendo éxito en ese momento las llamaron “web 2.0”. (O'Reilly,2005)

Esto nos da idea de las primeras características del concepto. En primer lugar podemos decir que nace de la observación y no de un replanteamiento teórico de los servicios de internet. Había servicios web 2.0 antes de formularse

el concepto. En segundo lugar, nace en entorno empresarial y tiene, en su origen, una marcada impronta tecnológica y de marketing. La tercera característica es que no hay un elemento clave que sitúe un servicio web en el lado del 2.0, existe una serie de características que se pueden cumplir en mayor o menor medida, pero es difícil encontrarlas todas simultánea mente en un mismo servicio web.

En su primer trabajo O'Reilly nos da una definición de web 2.0, así que cuando se le pidió esa definición envió un mensaje al blog de su empresa definiéndola. Afirmó que las aplicaciones web 2.0 son aquellas que sacan partido a las ventajas intrínsecas de la web, ofreciendo un servicio continuamente actualizado que mejora cuanto más gente lo use, utilizando y remezclando los datos de múltiples recursos, incluyendo los usuarios individuales, a la vez que ofrecen sus propios datos y servicios de tal forma que pueden ser reutilizados por otros, creando una "arquitectura de participación" en red, yendo más allá de la página de la web 1.0 para ofrecer experiencias de usuario cada vez más ricas. (O'Reilly, 2005)

En esta definición puede verse claramente cómo el autor está relacionando la web 2.0 con la tecnología casi de forma exclusiva. Pero son los principios de compartir, reutilizar, mejora continua, consideración del usuario como fuente de información, confianza, aprovechamiento de la inteligencia colectiva, etc., los que han impulsado el establecimiento de la actitud 2.0, haciendo que la tecnología pase a un segundo plano.

En la actualidad cuando se habla de web 2.0 se está haciendo referencia al uso de determinadas tecnologías (ajax, mashups, software social, rss) y a la aplicación de una determinada actitud en el diseño de servicios web, basada en los principios citados.

#### **2.5.10.1. Las Tecnologías 2.0**

Hay algunas tecnologías consideradas 2.0, es decir, suelen estar presentes en los sitios web 2.0. No siempre son originarias de la web 2.0, pero son consideradas típicas de estos sitios (Margaix, 2007:99):

- Ajax: una combinación de xml y javascript que posibilita crear aplicaciones web ejecutables en el cliente, reduciendo considerablemente el tráfico de datos y la carga de trabajo del servidor, y con una mayor interactividad.

Muchos de los casos de “experiencias de usuario ricas” están programados con este lenguaje, como puede ser Google Maps, Gmail o Flickr.

- Los mashups o aplicaciones web híbridadas: Integran de manera transparente los datos de otros servicios web, gracias a que algunos de éstos han creado APIs que permiten desarrollar nuevas aplicaciones que accedan a sus datos gratuitamente. El ejemplo más citado es HousingMaps.com, donde se combina la información de una base de datos inmobiliaria con Google Maps.
- El software social supone un elemento clave en todo el desarrollo de la web 2.0, especialmente en lo que se refiere al aprovechamiento de la inteligencia colectiva. Se entiende como tal el conjunto de aplicaciones que permiten a los individuos comunicarse unos con otros y seguir las conversaciones a través de la web. Actualmente se suelen incluir también aquellas herramientas y servicios que permiten compartir información y objetos digitales, como los vídeos de YouTube o las fotografías de Flickr. Dentro de esta categoría hay tres componentes que han influido de manera decisiva en el la web 2.0: los blogs, los foros y las wikis

#### **2.5.11. Framework CodeIgniter**

CodeIgniter es un conjunto de herramientas para personas que construyen su aplicación web usando PHP. Su objetivo es permitirle desarrollar proyectos mucho más rápido de lo que podría si lo escribiese desde cero, proveyéndole un rico juego de librerías para tareas comúnmente necesarias, así como una interface simple y estructura lógica para acceder a esas librerías. CodeIgniter le permite creativamente enfocarse en su proyecto minimizando la cantidad de código necesaria para una tarea dada. (EllisLab, 2012.:19)

CodeIgniter se encuentra bajo una licencia open source Apache/BSD-style. Usa la arquitectura Modelo-Vista-Controlador, que permite una buena separación entre lógica y presentación. Esto es particularmente bueno para proyecto en los cuales diseñadores están trabajando con sus archivos de plantilla, ya que el código en esos archivos será mínimo.

CodeIgniter viene con un rango lleno de librerías que le permiten realizar las tareas de desarrollo web comúnmente necesarias, como acceder a una

base de datos, mandar un email, validar datos de un formulario, mantener sesiones, manipular imágenes. (EllisLab, 2012.:20)

### 2.5.11.1. Características de CodeIgniter

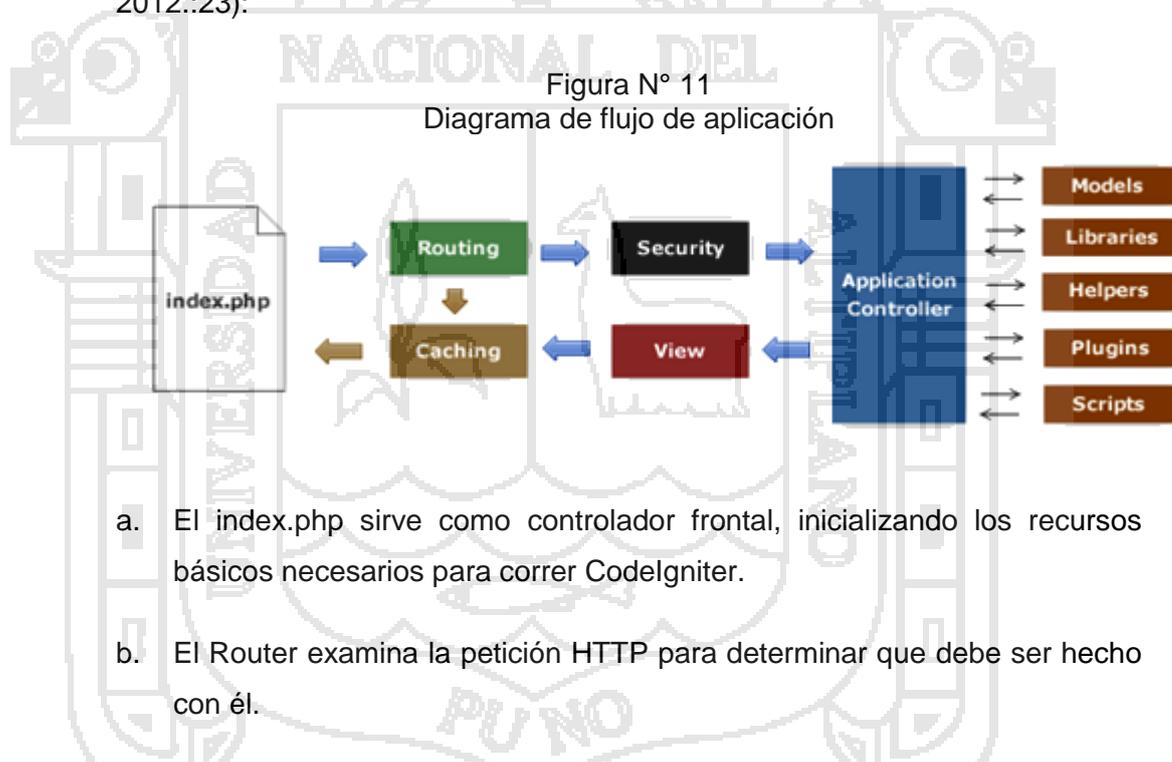
Las características en y de ellas son una forma muy pobre de juzgar una aplicación ya que no dicen nada acerca de la experiencia del usuario, o cuán intuitivamente o inteligentemente es diseñado. Las características no revelan nada acerca de la calidad del código, o la performance, o la atención a los detalles, o las prácticas de seguridad. La única forma de realmente juzgar una aplicación es probarla y llegar a conocer el código. Tenemos aquí una lista de las características principales de CodeIgniter (EllisLab, 2012.:22):

- Sistema Basado en Modelo-Vista-Controlador
- Compatible con PHP 4 y 5
- Extremadamente Liviano
- Clases de base de datos llenas de características con soporte para varias plataformas.
- Soporte de Active Record para Base de Datos
- Formulario y Validación de Datos
- Seguridad y Filtro XSS
- Manejo de Sesión
- Clase de Envío de Email. Soporta Archivos Adjuntos, email de texto/HTML, múltiples protocolos (sendmail, SMTP, and Mail) y más.
- Librería de Manipulación de Imagen (cortar, redimensionar, rotar, etc.). Soporta
- GD, ImageMagick, y NetPBM
- Clase de Carga (upload) de Archivo
- Clase de FTP
- Localización
- Paginación
- Encriptación de Datos
- Puntos de referencia
- Cacheo de páginas enteras
- Historial de Errores
- Perfilando la Aplicación
- Scaffolding
- Clase de Calendario

- Clase de Agente del Usuario
- Clase de Codificación Zip
- Clase de Motor de Plantillas
- Clase de Trackback
- Librería XML-RPC
- Clase de Prueba de Unidad
- URLs amigables a motores de búsqueda
- Ruteo de URI Flexible
- Soporte para Ganchos, Extensiones de Clase y Plugins
- Larga librería de funciones "asistentes"

### 2.5.11.2. Diagrama de Flujo de Aplicación

La figura 2.11 ilustra como fluyen los datos a través del sistema (EllisLab, 2012.:23):



- a. El index.php sirve como controlador frontal, inicializando los recursos básicos necesarios para correr CodeIgniter.
- b. El Router examina la petición HTTP para determinar que debe ser hecho con él.
- c. Si un archivo de caché existe, es enviado directamente al explorador, sobrepasando el sistema de ejecución normal.
- d. Seguridad, antes que el controlador sea cargado, la petición HTTP y cualquier dato suministrado por el usuario es filtrado por seguridad.
- e. El controlador carga los modelos, librerías, plugins, asistentes y cualquier otro recurso necesario para procesar la petición específica.

- f. La Vista finalizada es presentada entonces enviada al explorador web para ser vista. Si el cacheo está habilitado, la vista es cacheada primero para que las peticiones subsecuentes puedan ser servidas.

## **2.5.12. Gestor de Base de Datos Mysql**

### **2.5.12.1. ¿Qué es MySQL?**

MySQL es 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. 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, 2008:3)

### **2.5.12.2. Características de MySQL**

Las principales características de este gestor de bases de datos son las siguientes: (Pecos, 2008:4)

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

## **2.5.13. Lenguaje de Programación Php**

PHP son las iniciales de PHP Hypertext Procesor, y se trata de un lenguaje para la programación de páginas dinámicas de servidor. ¿Por qué PHP, si existen otros lenguajes similares que pueden resultar mejores?. Bien, es probable que esta pregunta sea cierta, pero hay dos características muy importantes que diferencian a PHP del resto de sus competidores. En primer lugar se trata de un lenguaje de tipo gratuito. Es decir, que no es necesario comprarlo, ni pagar nada para obtenerlo. Basta con ir a su website ([www.php.net](http://www.php.net)) y descargarlo. PHP forma parte de los programas que se conocen como código abierto (Opensource). Es decir, que ese posible realizar modificaciones y mejoras en el mismo y ponerlas a disposición del resto de los usuarios, de hecho también es posible obtener los ficheros de código fuente. Sus segunda característica importante es que se trata de un lenguaje multiplataforma, esto quiere decir que nuestra aplicación web desarrollada en PHP puede funcionar en casi cualquier tipo de plataforma: Windows, Linux, Unix (y sus diferentes versiones), Solaris, etc. Además de ofrecer soporte a las más importantes y usadas bases de datos (SQL Server, Oracle, mySQL, PostgreSQL, Informix, etc.). (Trigos, 2000:27)

Para poder crear una aplicación web basada en PHP son necesarios dos tipos de programas. El primero de ellos es el servidor web que va atender las peticiones de los usuarios y devolverá las páginas solicitadas. El servidor Apache, tanto su versión Windows como Linux, es el más utilizado cuando de PHP se trata. El segundo programa es el propio PHP, es decir el módulo que se va a encargar de interpretar y ejecutar las páginas PHP que se soliciten al servidor. (Trigos, 2000:28)

#### **2.5.14. Servidor Apache**

Apache es el número uno de los servidores web de todo el mundo. Aproximadamente el 50% de los servidores de la red utilizan Apache. El servidor Apache se basa en el modelo clásico de cliente-servidor, algunas claves para su éxito son su modularidad, potencia y disponibilidad. (Kabir, 2003:43)

##### **2.5.14.1. Modelo Cliente-Servidor**

- En las arquitecturas basadas en este modelo existen una serie de procesos llamados servidores que ofrecen una serie de servicios a otros procesos que se denominan clientes.
- El cliente debe de conocer la dirección del servidor y del proceso que nos va a atender dentro de la máquina. (Kabir, 2003:43)

#### **2.5.14.2. Como obtenerlo y su licencia**

- El código fuente de Apache es gratuito, así puede ser explorado y modificado.
- Este software gratuito se puede encontrar de dos formas: Mediante FTP: en el sitio web [Java.sun.com](http://Java.sun.com) y en la página web: [www.apache.org](http://www.apache.org) Aquí se puede encontrar las últimas versiones de Apache (estable y oficial), parches, módulos, etc. (Kabir, 2003:44)

#### **2.5.15. SEO (Search Engine Optimization)**

SEO (Search Engine Optimization) Optimización de motores de búsqueda, es la práctica de utilizar un rango de técnicas, incluidas la reescritura del código html, la edición de contenidos, la navegación en el site, campañas de enlaces y más acciones, con el fin de mejorar la posición de un website en los resultados de los buscadores para unos términos de búsqueda concretos. (IAB, 2010:9)

##### **2.5.15.1. Ventajas de ser visibles en buscadores**

La diferencia de la visibilidad con buscadores frente a otros modelos de publicidad en Internet es que el usuario busca de forma voluntaria un servicio, producto o información. Se asemeja a la búsqueda de información en las páginas amarillas tradicionales. Esto significa que la búsqueda responde a una necesidad y que, en algunos casos, puede suponer el inicio de un proceso de compra, ya sea en su fase de información, comparación o decisión final. (IAB, 2010:13)

Estar en el sitio y momento adecuado en función de un término de búsqueda adecuado puede convertirse en una oportunidad clara de negocio.

El SEO permite controlar, dentro de lo incontrolable (nadie puede asegurar un posicionamiento fijo en los buscadores), la posibilidad de aumentar las opciones de ser visible y de que se convierta en el primer paso para reconducir búsquedas en procesos de compra, apoyados por otras disciplinas como la analítica Web y la optimización de páginas de aterrizaje, para que, en la medida de lo posible, se pueda capitalizar esa visibilidad.

#### **2.5.16. Diseño Web Adaptativo (Responsive Web Design)**

Se describe como una técnica de diseño y desarrollo web que, mediante el uso de estructuras e imágenes fluidas, así como de media-queries en la hoja de estilo CSS, consigue adaptar el sitio web al entorno del usuario.

Toma las mejores prácticas para aplicarlas en la construcción de sitios, logrando buena calidad en las aplicaciones. La idea es que un solo sitio sea no solo adaptable a las características del recurso, sino que llegue a ser adaptativo. (Labrada y Salgado, 2013:4)

El diseño de la información es variable y relativo, denominado fluido. La composición se presenta en forma dinámica, puesto que cada elemento se ajusta a las condiciones técnicas de despliegue. Para hacer un diseño web adaptativo se debe cumplir con los siguientes aspectos:

- a. Diseño fluido con cuadrículas flexibles o fluid grids.
- b. Media Queries.
- c. Imágenes, objetos, videos o medios similares flexibles.
- d. Fuentes tipográficas con valores relativos.

##### **2.5.16.1. Beneficios del Diseño Web Adaptativo**

La utilización de un diseño web adaptativo trae consigo varios beneficios, tanto para los usuarios como para los desarrolladores, entre los que destacan (Labrada y Salgado, 2013:7):

- Reducción de costos: Se logra gracias a que el código es menor y no se requieren versiones distintas para su despliegue en cada uno de los dispositivos conocidos.
- Eficiencia en la actualización: Se utiliza una sola plantilla para la producción de la página, por lo que resulta eficiente la modificación.

- Mejora en la usabilidad: La legibilidad y características de uso del sitio se ajustan automáticamente en cada dispositivo.
- Capacidad de adaptación de la interfaz: Al ser un sitio con fluidez, la información es jerarquizada para presentar lo esencial de los contenidos para la consulta del usuario.
- Utilización de imágenes, videos y otros medios: Los recursos se redimensionan proporcionalmente, conservando una calidad óptima en pantalla.
- Tamaño relativo: Al estar basado en proporciones, es compatible con diferentes resoluciones y distintos dispositivos.
- Única dirección del sitio web (URL): La optimización de motores de búsqueda mejora, al aparecer en los resultados de búsqueda una sola URL.

#### **2.5.17. Métricas de Software**

Toda magnitud física es susceptible de ser medida, sin embargo, el software es una actividad fundamentalmente intelectual, no sólo en su origen, sino que su resultado final también lo es. El software no tiene de entidad física más que el soporte que lo registra: el papel, material magnético, los circuitos de la memoria. Las medidas realizadas sobre estos soportes no lo son estrictamente del software, sino que se trata únicamente de referencias indirectas. (Cuvillo, 2008:1)

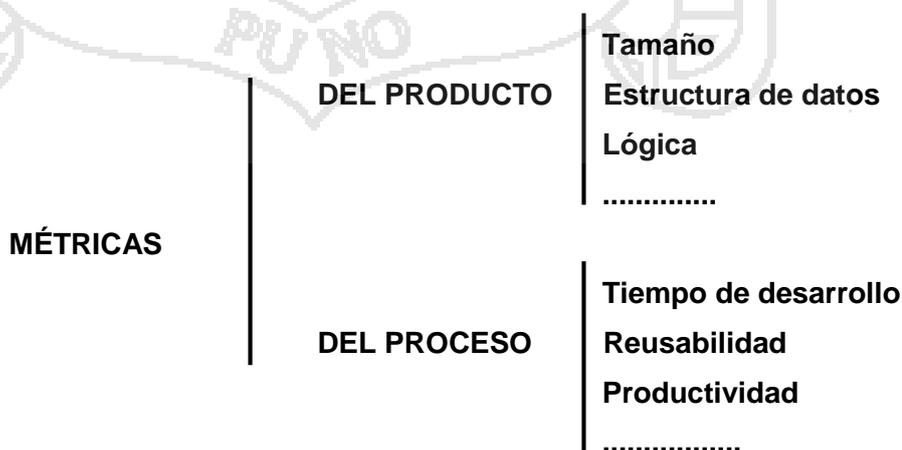
Las métricas del software responden a dos objetivos: valorar y estimar. Las magnitudes objeto de valoración son tres: la calidad, fiabilidad y productividad. La estimación parte de mediciones históricas para prever el esfuerzo y el tiempo que debe invertirse en un proyecto dado, y las características del resultado final. (Cuvillo, 2008:1)

Cualquiera de estas tareas (valoración y estimación) resulta tremendamente compleja, a pesar de contar con las métricas del software. Calidad y fiabilidad son conceptos importados de la ingeniería hardware, pero que en el software tienen un sentido muy distinto. La productividad es un concepto ampliamente desarrollado en otras áreas industriales, pero la carencia de un sistema métrico unificado, y la diversidad de factores que

influyen en ella, hacen que tampoco sea posible obtener valoraciones fiables de esa magnitud en el ámbito del software. Aún cuando hay medidas que son generalmente aceptadas (las líneas de código como métrica de tamaño, por ejemplo), no desaparecen los problemas por completo: matices de apreciación (considerar o no los comentarios y las líneas en blanco) y sistemas de referencia de compatibilidad dudosa (código generado en distintos lenguajes) hacen difícil encontrar una referencia consolidada en la que basar las observaciones. (Cuvillo, 2008:2)

Hay muchas magnitudes que pueden ser medidas en el software: el tamaño en líneas de código, el coste monetario del desarrollo, el tiempo de desarrollo en días de trabajo, el tamaño de la memoria precisada en bytes, e incluso el número de quejas del usuario antes de entregar el producto. Diferentes observadores del mismo producto, pueden obtener distintas medidas, incluso en una misma magnitud. Una posible clasificación de las métricas empleadas para valorar el software aparece en la figura N° 2.13 y establece dos categorías principales: métricas del producto, y métricas del proceso. Las primeras son las que tradicionalmente se han empleado en la medida del software, y pueden ser obtenidas automáticamente tomando como entrada el código fuente: tamaño, estructuras de datos y lógica. Las métricas del proceso por su parte dependen esencialmente del entorno de desarrollo. Un ejemplo de este tipo de métricas es el tiempo empleado para desarrollar un elemento software que depende de distintos factores externos (dificultad del problema, capacidad del personal, metodología empleada, etc...). (Cuvillo, 2008:2)

Figura N° 12  
Métricas del software. Categorías



Un problema asociado al empleo de métricas del software es el de comparar medidas diferentes, aunque aparentemente se refieran a la misma magnitud. Sólo la definición precisa de las condiciones y los elementos que intervienen en una medida determinada permitirán que la comparación de dos valores pueda realizarse con garantías y pueda servir de base a estudios más completos y fiables.

A continuación se enumeran algunas de las métricas del software más comunes ordenadas por el ámbito del software a que se refiere. A cada una de ellas, podría añadirse una larga serie de variaciones según los distintos autores, pero se pretende dar una visión de conjunto, más que hacer una enumeración exhaustiva. (Cuvillo, 2008:3-6)

#### **A. Métricas de tamaño**

Este tamaño se determina habitualmente tomando como referencia el programa en código fuente.

#### **B. Métricas de estructuras de datos**

Una de las razones fundamentales de la programación es el proceso de datos. Parte de estos datos constituyen la entrada del sistema, parte tiene un uso exclusivamente interno y, por último, una tercera parte constituye la salida del sistema.

#### **C. Métricas de estructuras de control**

La estructura lógica de un programa es el mecanismo que le permite realizar las distintas funciones para las que fue construido.

#### **D. Métricas compuestas**

Las medidas descritas hasta ahora miden una única magnitud para darle sentido como una característica del software. Sin embargo, ocurre con frecuencia que para describir una determinada cualidad del software es preciso componer (construir un par) de medidas simples.

#### **E. Métricas de esfuerzo**

El desarrollo del software es una actividad humana que depende en gran medida del trabajo personal. A la hora de valorar un sistema software debe

considerarse la cantidad de esfuerzo que debe invertir el equipo de desarrollo para culminar su construcción.

#### **F. Métricas de calidad y fiabilidad**

El estudio de la calidad y fiabilidad tiene una importancia cada vez mayor en el mundo de la Ingeniería del Software. No sólo se trata de obtener sistemas desarrollados correctamente, de acuerdo a los requerimientos y a los estándares establecidos, sino que se pretende conseguir programas fáciles de mantener y, lo que es más importante, sistemas fiables en tareas críticas.

#### **G. Métricas de diseño**

Entre las diversas ventajas de las técnicas de diseño se pueden destacar las siguientes:

- a. Comprensibilidad: programadores y usuarios pueden comprender fácilmente la lógica del programa.
- b. Manejabilidad: los gestores pueden asignar fácilmente personal y recursos a los distintos módulos representados por tareas.
- c. Eficiencia: el esfuerzo de implementación puede reducirse.
- d. Reducción de errores: los planes de prueba se simplifican notablemente.
- e. Reducción del esfuerzo de mantenimiento: la división en módulos favorece que las distintas funciones las lleven a cabo módulos diferenciados.

#### **H. Métricas de reusabilidad**

Las métricas de reusabilidad tratan de medir el grado en que un elemento software puede ser empleado por otros programas, en otras palabras, su independencia.

#### **I. Métricas de flexibilidad**

Las de flexibilidad tienen como componentes a la complejidad, la concisión, la consistencia, la expandibilidad, la generalidad, la autodocumentación, y la simplicidad.

También listamos otras métricas que son importantes:

### **K. Requerimientos de software**

Pressman (2010:37) conceptualiza requerimientos de software de la siguiente manera: “es una descripción completa del comportamiento de un sistema a desarrollar. Incluye un conjunto de casos de uso que describen todas las interacciones de los usuarios tendrán con el software. Los casos de uso también son conocidos como los requisitos funcionales. Además de los casos de uso, los requerimientos de software también pueden ser no funcionales (o complementos). Los requerimientos no funcionales son requisitos que imponen limitaciones en el diseño o la implementación (por ejemplo, requisitos de desempeño de ingeniería, normas de calidad, o restricciones de diseño)”.

### **L. Complejidad del Software**

La complejidad de software, que a menudo es también conocida como la complejidad de programación es un término que abarca numerosas propiedades de una pieza de software, todo lo cual afecta las interacciones internas. Según varios autores hay una distinción entre los términos complejo y complicado, complicado implica ser difícil de entender pero con el tiempo y esfuerzo en última instancia cognoscible. Complejo, por otro lado describe las interacciones entre varias entidades. Mientras el número de entidades aumenta, el número de interacciones entre ellos aumentaría exponencialmente y sería llegar a un punto en el que sería imposible conocer y comprender a todos ellos. Del mismo modo, los mayores niveles de complejidad en el software aumenta el riesgo de querer interferir en las interacciones y así se incrementa la posibilidad de la introducción de defectos cuando se hacen cambios. En casos más extremos la modificación del software se puede hacer prácticamente imposible. La idea de vincular la complejidad del software a la capacidad de mantenimiento del software se ha explorado ampliamente por el profesor Manny Lehman, que desarrolló sus “Leyes de la Evolución del Software” que podrían ser utilizados para medir el estado del software, llegando a la conclusión de que la única solución práctica sería utilizar uno de sus modelos deterministas de complejidad. (Abrahamson, 2006: 50)

### **M. Usabilidad del Software**

Pressman (2010:56) al respecto hace un interesante comentario: “El calificativo “amigable con el usuario” se ha convertido en un término omnipresente en las discusiones sobre productos de software. Si un programa no es “amigable con el usuario” frecuentemente está avocado al fracaso, incluso aunque las funciones que realice sean valiosas. La usabilidad es un intento de cuantificar la amistad del software con el usuario y que se puede medir en base a cuatro características: habilidad intelectual para aprender el sistema, tiempo requerido para llegar a ser moderadamente eficiente en el uso del sistema, aumento neto en productividad, valoración subjetiva de la disposición de los usuarios hacia el sistema”.

#### **N. Mantenimiento del Software.**

Las tareas de mantenimiento del software son un proceso delicado que algunos autores comparan con un “ice-berg” que se espera que lo que está inmediatamente visible sea todo lo que hay. Realmente sabemos que bajo la superficie se esconde una gran cantidad de problemas potenciales y de costes.

El mantenimiento del software existente puede llevarse hasta el 70% de todo el esfuerzo gastado por una organización de desarrollo. El porcentaje sigue subiendo a medida que se produce más software. En el horizonte podemos prever una “barrera de mantenimiento” para las organizaciones de desarrollo de software que no podrán producir nuevo software debido a que gastarán todos los recursos disponibles en el mantenimiento del antiguo software. (Pressman, 2010:693)

#### **O. Portabilidad**

Pressman (2010:578) menciona la portabilidad como un factor para determinar la calidad del software y menciona como definición: “El esfuerzo requerido para transferir el programa desde un hardware y/o un entorno de sistema de software a otro”.

#### **P. Longitud del Programa**

La teoría de Halstead sobre la ciencia de software propone las leyes analíticas de la ciencia de software, la ciencia de software propone las primeras “leyes” analíticas de la informática, Halstead usa las primitivas de medidas para desarrollar expresiones para la longitud total del programa determinada por la fórmula (Pressman, 2010:600):

$$N = n_1 \log_2 n_1 + n_2 \log_2 n_2$$

Donde:

$n_1$  : número de operadores distintos que aparecen en un programa.

$n_2$ : número de operandos distintos que aparecen en un programa.

#### Q. Volumen del Programa

Halstead dentro de las “leyes” analíticas de la informática también mencionada en el punto anterior propone la métrica volumen del programa que se define de acuerdo a la siguiente fórmula (Pressman, 2007: 600):

$$V = N n_1 \log_2(n_1 + n_2)$$

Donde:

N: longitud del programa definido por la fórmula

$n_1$ : número de operadores distintos que aparecen en un programa.

$n_2$ : número de operandos distintos que aparecen en un programa.

#### 4.2.18. Heurísticas de Usabilidad de Jakob Nielsen

Jakob Nielsen “el gurú de la usabilidad” es una de las personas más respetadas en el ámbito mundial sobre usabilidad en la web, un referente de Braintive a la hora de diseñar nuestros proyectos. (Braintive, 2013)

Jakob estudió 249 problemas de usabilidad y a partir de ellos diseñó lo que denominó las “**reglas generales**” para identificar los posibles problemas de usabilidad.

- Visibilidad del estado del sistema
- Relación entre el sistema y el mundo real
- Control y libertad del usuario:
- Consistencia y estándares.
- Prevención de errores
- Reconocimiento antes que recuerdo:
- Flexibilidad y eficiencia de uso: Estética y diseño minimalista:
- Ayudar a los usuarios a reconocer.
- Ayuda y documentación

## 2.6. MARCO CONCEPTUAL

### 2.3.1. Comercio Electrónico

El comercio electrónico, también conocido como *e-commerce* (*electronic commerce* en inglés), consiste en la compra y venta de productos o de servicios a través de medios electrónicos, tales como Internet y otras redes informáticas.

### 2.3.2. Sistema Administrador de Contenido (CMS)

Un sistema de gestión de contenidos o CMS (por sus siglas en inglés, Content Management System) es una herramienta que permite a un editor crear, clasificar y publicar cualquier tipo de información en una página web.

### 2.3.3. Desarrollo Ágil de Software

El desarrollo ágil de software refiere a métodos de ingeniería del software basados en el desarrollo iterativo e incremental, donde los requisitos y soluciones evolucionan mediante la colaboración de grupos auto organizados y multidisciplinarios.

### 2.3.4. AJAX

AJAX son las siglas de Asynchronous JavaScript and XML (javascript y XML). No es un lenguaje, ni una plataforma, sino una técnica que permite que la aplicación realice conexiones con el servidor web y actualice la página sin necesidad de recargarla por completo como con un pedido HTTP tradicional. (Escobedo. 2008:31)

### 2.3.5. JavaScript

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. (Eguíluz, 2009a:5)

### **2.3.6. CSS**

CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas. (Eguíluz, 2009b:5)

### **2.3.7. Certificados Digitales**

Una certificación digital es un documento electrónico que proporciona al usuario un alto grado de confianza con una organización o persona con la que estén tratando.

### **2.3.8. Framework**

En el desarrollo de software un Framework es una estructura de soporte definida, en la cual otro proyecto de software puede ser organizado y desarrollado. (Escobedo, 2008:32)

### **2.3.9. XP (Programación Extrema)**

La Programación Extrema es una metodología ágil de desarrollo de software que se basa en la simplicidad, la comunicación y la realimentación o reutilización del código desarrollado. (Fernández, 2002:2)

### **2.3.10. Tecnologías de Información y Comunicación**

Las Tecnologías de la Información y Comunicación (TIC) son un conjunto de servicios, redes, software y dispositivos de hardware que se integran en sistemas de información interconectados y complementarios, con la finalidad de gestionar datos e información de manera efectiva, mejorando la productividad de los ciudadanos, gobierno, y empresas, dando como resultado una mejora en la calidad de vida. (PCM, 2011:8)

### **2.3.11. Firma Electrónica**

En términos genéricos, es cualquier método o símbolo basado en medios electrónicos utilizado o adoptado por una parte con la intención actual de vincularse o autenticar un documento, cumpliendo todas o algunas de las funciones características de una firma manuscrita. (Soto, 2010:16)

### 2.3.12. Prototipo

Es un sistema que funciona, no sólo una idea en el papel, desarrollado con la finalidad de probar ideas y suposiciones relacionadas con el nuevo sistema. Al igual que cualquier sistema basado en computadora, está constituido por software que acepta entradas, realiza cálculos, produce información ya sea impresa o presentada en una pantalla. Es la primera versión o iteración de un sistema; es el modelo original.

### 2.3.13. Certificados SSL

Las siglas SSL (Capa de Conexión Segura) responden a los términos en inglés (Secure Socket Layer), el cual es un protocolo de seguridad que hace que sus datos viajen de manera íntegra y segura, es decir, la transmisión de los datos entre un servidor y usuario web, y en retroalimentación, es totalmente cifrada o encriptada.

### 2.3.14. Dominio

Un dominio es una dirección de Internet. Para visitar una web tecleamos una dirección a través del navegador. El conjunto de caracteres que compone dicha dirección sirve para identificar este sitio concreto de la red. (Anetcom, 2004:12)

### 2.3.15. Hosting

El alojamiento web (en inglés *web hosting*) es el servicio que provee a los usuarios de Internet un sistema para poder almacenar información, imágenes, vídeo, o cualquier contenido accesible vía web.

### 2.3.16. Paypal

Consiste en la recepción y envío de dinero en Internet de forma rápida y segura entre comprador y vendedor. Para ello se tiene la posibilidad de registrarse gratis desde su web [www.paypal.es](http://www.paypal.es) y obtener servicios como suscripción a pagos periódicos, realizar el pago desde una cuenta bancaria, o incluso que el dinero se deposite en la propia cuenta de PayPal. (AECM, 2009:41)

### **2.3.17. GNU-GPL**

Pretende garantizar la distribución, modificación y uso de todo software libre, siempre y cuando se tengan en cuenta los derechos de cada autor y entrega de código fuente de cada modificación que se realice.

### **2.3.18. GPL**

Licencia creada para proteger el software de la libre distribución, modificación y uso. El Software que se encuentra bajo esta licencia es denominada Software Libre.

### **2.3.19. Sistema de pago TPV**

TPV (Terminal de pago virtual), Es el sistema de pago electrónico más común y aceptado hoy en día dado el uso generalizado de las tarjetas de crédito. Podemos distinguir dos tipos de TPVs: el tradicional a través del cual se puede usar cualquier tarjeta de crédito y los TPVs 3D Secure en los que el pago se realiza mediante conexión telemática directa con una pasarela de pago de una entidad bancaria. (AECEM, 2009:41)

## **2.4 HIPÓTESIS**

### **2.4.1. Hipótesis General**

El prototipo de sistema de administración de contenidos facilita la implementación y mantenimiento de portales web orientados al e-commerce.

### **2.4.2. Hipótesis Específicas**

- El análisis de requerimientos en el desarrollo del prototipo de sistema de administración de contenidos para la implementación y mantenimiento de portales web orientados al e-commerce es el adecuado.
- El diseño del prototipo de sistema de administración de contenidos para la implementación y mantenimiento de portales web orientados al e-commerce es el apropiado.
- La implementación del prototipo de sistema de administración de contenidos para la implementación y mantenimiento de portales web orientados al e-commerce utilizando la tecnología web es el adecuado.

- La validación del prototipo de sistema de administración de contenidos para la implementación y mantenimiento de portales web orientados al e-commerce lleva a un resultado satisfactorio.

## 2.5. OPERACIONALIZACIÓN DE VARIABLES

**Variable Independiente:** Prototipo de sistema de administración de contenidos.

**Variable Dependiente:** Implementación y mantenimiento de portales web orientados al e-commerce.

Cuadro N° 6  
Operacionalización de variables

VARIABLES	DIMENSIONES	INDICADORES	ITEMS
<p><b>Variable Independiente:</b> Prototipo de sistema de administración de contenidos.</p>	Análisis de Requerimientos	Grado de satisfacción de los requerimientos.	<ul style="list-style-type: none"> <li>- N° de requerimientos identificados.</li> <li>- N° de requerimientos desarrollados satisfactoriamente.</li> </ul>
	Diseño del Sistema	Nivel de Usabilidad	<ul style="list-style-type: none"> <li>- Esfuerzo necesario para la formación uso del sistema</li> <li>- Facilidad de operación.</li> </ul>
		Estética	<ul style="list-style-type: none"> <li>- Grado de atracción</li> </ul>
		Nivel de manejabilidad	<ul style="list-style-type: none"> <li>- Facilidad de navegación</li> </ul>
	Implementación del Sistema	Eficiencia	<ul style="list-style-type: none"> <li>- Nivel de rapidez.</li> </ul>
		Grado de mantenibilidad del sistema.	<ul style="list-style-type: none"> <li>- Índice de madurez del software.</li> </ul>
		Grado de portabilidad del sistema.	<ul style="list-style-type: none"> <li>- Independencia del hardware.</li> <li>- Independencia del sistema.</li> </ul>



<p><b>Variable Dependiente:</b></p> <p>Implementación y mantenimiento de portales web orientados al E-commerce.</p>	<p>Validación del Sistema</p>	<p>Implementación del portal Web</p>	<ul style="list-style-type: none"> <li>- Tiempo de implementación.</li> <li>- Facilidad de implementación</li> <li>- Facilidad de aprendizaje.</li> </ul>
		<p>Mantenimiento del portal Web</p>	<ul style="list-style-type: none"> <li>- Facilidad de actualización.</li> <li>- Es autodescriptible.</li> </ul>



## CAPÍTULO III

### MATERIALES Y MÉTODOS

---

#### 3.6. POBLACIÓN Y MUESTRA

##### 3.1.1. Población

La población de la presente investigación está conformada por representantes y personal de apoyo de la parte administrativa de las 88 empresas y Pymes artesanales registradas en el Centro de Innovación Tecnológica de los Camélidos Sudamericanos de Puno, los cuales son 176.

##### 3.1.2. Muestra

Se utilizó el muestreo por conveniencia que es una técnica no probabilística donde los sujetos son seleccionados dada la conveniente accesibilidad y proximidad para el investigador. Del total de empresas y Pymes de la población tenemos 16 las cuales están interesadas en contar con un portal de e-commerce.

La muestra está definida para el caso por los usuarios que deben validar la operatividad del sistema, para la presente investigación hemos considerado a 34 personas las cuales intervendrán en el uso del sistema de administración de contenidos, según el cuadro 3.1:

Cuadro N° 7  
Personal que intervendrán en el uso del sistema

N°	EMPRESA	Número de personas que intervendrán en el uso del sistema		
		Representantes	Personal de apoyo	Total
1	Artesanías Alpazury SRL.	1	1	2
2	Artesanías Sumac Perú	1	2	3
3	Arte y Moda SRL.	1	1	2
4	SOTEXPRO	1	1	2
5	Incatex	1	2	3
6	Juliaca Export	1	1	2
7	Mipsa Los Andes	1	1	2
8	Pimentel	1	1	2
9	Artesanías Kori Pacocha	1	1	2
10	Tikari Quilca	1	1	2
11	Alpatex.	1	1	2
12	Tikari Perú	1	1	2
13	Artesanías Antholla	1	1	2
14	Suri Alpaca	1	0	1
15	Artesanías Tunapa	1	2	3
16	Kactus Artesanía	1	1	2
<b>TOTAL</b>				<b>34</b>

Fuente: Elaboración propia, según entrevistas realizadas.

### 3.2. MÉTODOS DE RECOPIACIÓN DE DATOS

Las principales técnicas e instrumentos que se utilizaron en el presente proyecto de investigación son:

#### 3.2.1. Fuentes Primarias

Para la obtener información directa se utilizaron las siguientes técnicas:

- a. Entrevistas: Técnica utilizada con la finalidad de recabar información, especialmente para las creaciones de las historias de usuario para el presente proyecto.
- b. Encuestas: Técnica de recolección de datos y está conformado por un conjunto de preguntas escritas que el investigador administra o aplica a las personas o unidades de análisis, a fin de obtener la información necesaria para determinar los valores o respuestas de las variables motivo de estudio.

- c. Observación: Se define como una técnica de recolección de datos que permite acumular y sistematizar información sobre un hecho o fenómeno social que tiene relación con el problema que motiva la investigación.

### 3.2.2. Fuentes Secundarias

Para obtener información relacionada al tema se utilizaron los siguientes materiales:

- a. Documentos de la empresa
- b. Libros
- c. Revistas
- d. Información de internet (escritos, videos, etc.)

### 3.3. MÉTODO DE TRATAMIENTO DE DATOS

El método de investigación científica que se utilizó para verificar la veracidad de la hipótesis fue el método de investigación empírica experimental, siguiendo el proceso de:

- Recopilación y tabulación de datos.
- Análisis y consistencia de los datos.
- Interpretación de datos y validación de la hipótesis mediante la prueba de hipótesis estadística.

Para esta fase se empleó la prueba de la distribución normal como aproximación de la binomial, se utiliza esta distribución porque la variable en estudio es del tipo cualitativo y el tamaño de muestra es mayor de 30.

Otra condición que cumplir antes de probar la hipótesis, la muestra debe cumplir:

$$np \geq 5$$

$$nq \geq 5$$

El producto  $np$  y  $nq$  debe ser de al menos 5 para poder utilizar la distribución normal como aproximación de la binomial.

- a. Hipótesis:

$H_0$ :  $P \geq P_{ho}$ ; los resultados del prototipo de aplicación son satisfactorios.

$H_1$ :  $P < P_{ho}$ ; los resultados del prototipo de aplicación no son satisfactorios.

b. Nivel de significancia:  $\alpha = 0.01$

c. Estadística.

Calcular el error estándar:

$$\sigma_{\bar{P}} = ((P_{ho} q_{ho})/n)^{1/2}$$

Calculamos la proporción de muestra:

$$\bar{P} = x/n$$

Estandarizamos la proporción de la muestra mediante la siguiente fórmula:

$$Z = \frac{\bar{P} - P_{ho}}{\sigma_{\bar{P}}}$$

Donde:

$\sigma_P$  : Margen de error

$P_{ho}$  : Valor hipotético de la proporción de éxitos de la población.

$q_{ho}$  : Valor hipotético de la proporción de fracasos de la población.

$P$  : Proporción de la muestra.

$X$  : Número de éxitos en la muestra.

d. Región crítica

R.C. =  $] -\infty, -Z_{1-\alpha}[$

e. Decisión.

Se rechaza  $H_0$  si  $Z < -Z_{1-\alpha}$ , en caso contrario no se rechaza.

### 3.4. METODOLOGÍA PARA EL DESARROLLO DE SOFTWARE

La presente investigación se centra en aplicar la metodología de desarrollo ágil "Programación Extrema" (XP). Considera las siguientes fases de desarrollo:

a. Planeación

b. Diseño

c. Desarrollo

d. Pruebas

### 3.5. MATERIALES Y EQUIPO UTILIZADOS

Para el desarrollo del sistema administración de contenidos se utilizaron las siguientes herramientas:

#### A. RECURSOS SOFTWARE

- Microsoft Windows 7 Ultimate.
- Microsoft Office.
- WampServer
- Lenguaje de programación PHP 5.3.13
- Framework CodeIgniter versión 2.1.2.
- Gestor de Base de datos MySql 5.5.24
- Servidor Apache 2.2.22
- Editor de Programación Sublime text2

#### B. RECURSOS HARDWARE

- Desktop: Procesador AMD Athlon Dual Core 2.7 GHz, Memoria RAM 4GB. Disco Duro MAXTOR 250GB.
- Memoria USB 2GB.
- Impresora láser.

#### C. SERVICIOS

- Conexión a internet.

## CAPÍTULO IV

### RESULTADOS Y DISCUSIÓN

---

Para el desarrollo del presente capítulo se va a considerar la metodología de desarrollo de software ágil conocida como programación extrema; la misma que describimos en el marco teórico del presente documento.

#### 4.5. ANÁLISIS DE REQUERIMIENTOS DEL PROTOTIPO DE SISTEMA DE ADMINISTRACIÓN DE CONTENIDOS

##### 4.5.1. Planeación

Se describe la experiencia obtenida en la realización del proyecto. Inicialmente se comenta sobre cada uno de los aspectos que XP propone para la etapa de planeación. Las responsabilidades durante el escenario de la planificación son:

a. Cliente:

- Define las historias de usuario.
- Decide que valores del sistema tiene la historia de usuario.
- Decide que historias construir en las entregas.

b. Desarrollador:

- Estima que tan grande será de implementar cada historia de usuario.
- Advierte al usuario de posibles riesgos técnicos.
- Mide el progreso del equipo.

c. Las responsabilidades del negocio son:

- Definir el rango de la entrega.
- Define el orden de las entregas (que historias son desarrolladas primero).
- Define fechas y tiempos de las entregas.

d. Las responsabilidades técnicas son:

- Estimar cuanto demorará cada historia de usuario.
- Comunicar impactos técnicos que conllevan la implementación de requerimientos.
- Desarrolla las tareas a realizar para cada historia de usuario.

#### 4.5.1.1. Tipos de Usuarios

Para el uso del sistema se tienen dos tipos de usuarios los cuales a su vez que subdividen:

- a. Cliente. Son de dos tipos.
  - Cliente anónimo: es la persona que entra al portal sin estar registrado.
  - Cliente registrado: es la persona que está registrada en el sistema y en el momento logueado. Este usuario podrá realizar más acciones que el anónimo.
- b. Administrador. Son de dos tipos:
  - Administrador general: es la persona que administra el sistema. El administrador tiene control de acuerdo a privilegios asignados.
  - Administrador asignado: es la persona a quien el administrador general le ha asignado un determinado módulo a administrar.

#### 4.5.1.2. Historias de Usuario

Las historias de usuario principalmente identifican los requisitos funcionales que debe cumplir el software a desarrollar; se desea que el sistema gestione la información para una aplicación de e-commerce, ya que para esto al reunirse con el usuario se han definido las siguientes historias de usuario las cuales se dividen en dos tipos:

##### **Sección Privada (administrativa):**

1. Instalación de sistema e-commerce.
2. Acceso al administrador.
3. Administración de pedidos.
4. Administración de clientes.
5. Administración de categorías de productos.
6. Administración de productos.
7. Administración de cupones de descuentos.
8. Generación de reportes.
9. Configuración de contenidos de tienda.
10. Administración de usuarios del sistema (administradores).
11. Administración de ubicaciones geográficas.
12. Configuración de medios de pagos.

**Sección Pública:**

13. Interfaz principal de tienda.
14. Búsqueda de productos en el catálogo.
15. Consulta de catálogo.
16. Agregar producto al carro de compras.
17. Enviar pedido.
18. Acceso de clientes registrados.
19. Registro de cliente.

A continuación se muestran las historias de usuario detalladamente:

**Sección Privada**

Historia de Usuario	
<b>Número:</b> 1	<b>Nombre:</b> Instalación de sistema e-commerce
<b>Usuario:</b> Cliente	
<b>Prioridad en negocio:</b> Alta (Alta / Media /Baja)	<b>Riesgo en desarrollo:</b> Media (Alta / Media /Baja)
<b>Puntos estimados:</b> 4	<b>Iteración asignada:</b> 1
<b>Descripción:</b> Instalar sistema e-commerce, para su implementación y mantenimiento.	

Historia de Usuario	
<b>Número:</b> 2	<b>Nombre:</b> Acceso al administrador.
<b>Usuario:</b> Administrador del Sistema	
<b>Prioridad en negocio:</b> Alta (Alta / Media /Baja)	<b>Riesgo en desarrollo:</b> Media (Alta / Media /Baja)
<b>Puntos estimados:</b> 4	<b>Iteración asignada:</b> 1
<b>Descripción:</b> Acceder al administrador mediante validación de datos.	

Historia de Usuario	
<b>Número:</b> 3	<b>Nombre:</b> Administración de pedidos.
<b>Usuario:</b> Administrador del Sistema	
<b>Prioridad en negocio:</b> Media (Alta / Media /Baja)	<b>Riesgo en desarrollo:</b> Media (Alta / Media /Baja)
<b>Puntos estimados:</b> 3	<b>Iteración asignada:</b> 3
<b>Descripción:</b> Administrar los pedidos para poder controlar los estados en los que se encuentran y poder revisar sus respectivos datos.	

Historia de Usuario	
<b>Número:</b> 4	<b>Nombre:</b> Administración de clientes.
<b>Usuario:</b> Administrador del Sistema	
<b>Prioridad en negocio:</b> Alta (Alta / Media /Baja)	<b>Riesgo en desarrollo:</b> Media (Alta / Media /Baja)
<b>Puntos estimados:</b> 3	<b>Iteración asignada:</b> 2
<b>Descripción:</b> Administrar clientes registrados y poder registrar nuevos clientes por el administrador.	

Historia de Usuario	
<b>Número:</b> 5	<b>Nombre:</b> Administración de categorías de productos.
<b>Usuario:</b> Administrador del Sistema	
<b>Prioridad en negocio:</b> Alta (Alta / Media /Baja)	<b>Riesgo en desarrollo:</b> Media (Alta / Media /Baja)
<b>Puntos estimados:</b> 3	<b>Iteración asignada:</b> 2
<b>Descripción:</b> Administrar categorías de productos para organizar de manera adecuada el catálogo de productos.	

Historia de Usuario	
<b>Número:</b> 6	<b>Nombre:</b> Administración de productos.
<b>Usuario:</b> Administrador del Sistema	
<b>Prioridad en negocio:</b> Alta (Alta / Media /Baja)	<b>Riesgo en desarrollo:</b> Media (Alta / Media /Baja)
<b>Puntos estimados:</b> 3	<b>Iteración asignada:</b> 3
<b>Descripción:</b> Administrar los productos para poder tener control adecuado de sus respectivos datos.	

Historia de Usuario	
<b>Número:</b> 7	<b>Nombre:</b> Creación de cupones de descuentos.
<b>Usuario:</b> Administrador del Sistema	
<b>Prioridad en negocio:</b> Baja (Alta / Media /Baja)	<b>Riesgo en desarrollo:</b> Media (Alta / Media /Baja)
<b>Puntos estimados:</b> 4	<b>Iteración asignada:</b> 9
<b>Descripción:</b> Crear cupones de descuento para los productos, podría ser para todos o seleccionar algunos.	

Historia de Usuario	
<b>Número:</b> 8	<b>Nombre:</b> Generación de reportes.
<b>Usuario:</b> Administrador del Sistema	
<b>Prioridad en negocio:</b> Media (Alta / Media /Baja)	<b>Riesgo en desarrollo:</b> Media (Alta / Media /Baja)
<b>Puntos estimados:</b> 3	<b>Iteración asignada:</b> 9
<b>Descripción:</b> Visualizar productos vendidos por fechas y los más vendidos.	

Historia de Usuario	
<b>Número:</b> 9	<b>Nombre:</b> Configuración de contenidos de tienda.
<b>Usuario:</b> Administrador del Sistema	
<b>Prioridad en negocio:</b> Media (Alta / Media /Baja)	<b>Riesgo en desarrollo:</b> Media (Alta / Media /Baja)
<b>Puntos estimados:</b> 4	<b>Iteración asignada:</b> 4
<b>Descripción:</b> Configurar banner, logo, páginas y contenidos de la tienda.	

Historia de Usuario	
<b>Número:</b> 10	<b>Nombre:</b> Administración de usuarios del sistema (administradores)
<b>Usuario:</b> Administrador del Sistema	
<b>Prioridad en negocio:</b> Media (Alta / Media /Baja)	<b>Riesgo en desarrollo:</b> Media (Alta / Media /Baja)
<b>Puntos estimados:</b> 3	<b>Iteración asignada:</b> 4
<b>Descripción:</b> Crear administradores del sistema y asignarles determinados módulos para su administración.	

Historia de Usuario	
<b>Número:</b> 11	<b>Nombre:</b> Administración de ubicaciones geográficas.
<b>Usuario:</b> Administrador del Sistema	
<b>Prioridad en negocio:</b> Baja (Alta / Media /Baja)	<b>Riesgo en desarrollo:</b> Media (Alta / Media /Baja)
<b>Puntos estimados:</b> 3	<b>Iteración asignada:</b> 10
<b>Descripción:</b> Administrar ubicaciones geográficas para configuraciones de envío de productos.	

Historia de Usuario	
<b>Número:</b> 12	<b>Nombre:</b> Configuración de medios de pagos.
<b>Usuario:</b> Administrador del Sistema	
<b>Prioridad en negocio:</b> Media (Alta / Media /Baja)	<b>Riesgo en desarrollo:</b> Media (Alta / Media /Baja)
<b>Puntos estimados:</b> 4	<b>Iteración asignada:</b> 5
<b>Descripción:</b> Crear módulo para escoger y asignar medio de pago a la tienda.	

## Sección Pública

Historia de Usuario	
<b>Número:</b> 13	<b>Nombre:</b> Interfaz principal de tienda.
<b>Usuario:</b> Cliente	
<b>Prioridad en negocio:</b> Media (Alta / Media /Baja)	<b>Riesgo en desarrollo:</b> Media (Alta / Media /Baja)
<b>Puntos estimados:</b> 4	<b>Iteración asignada:</b> 6
<b>Descripción:</b> Diseñar interfaz de navegación para la tienda virtual.	

Historia de Usuario	
<b>Número:</b> 14	<b>Nombre:</b> Búsqueda de productos en el catálogo.
<b>Usuario:</b> Cliente	
<b>Prioridad en negocio:</b> Media (Alta / Media /Baja)	<b>Riesgo en desarrollo:</b> Media (Alta / Media /Baja)
<b>Puntos estimados:</b> 4	<b>Iteración asignada:</b> 6
<b>Descripción:</b> Mediante un buscador, encontrar una lista de productos de acuerdo a su interés.	

Historia de Usuario	
<b>Número:</b> 15	<b>Nombre:</b> Consulta de catálogo.
<b>Usuario:</b> Cliente	
<b>Prioridad en negocio:</b> Media (Alta / Media /Baja)	<b>Riesgo en desarrollo:</b> Media (Alta / Media /Baja)
<b>Puntos estimados:</b> 2	<b>Iteración asignada:</b> 6
<b>Descripción:</b> Consultar catálogo de acuerdo a categorías de productos, para poder seleccionar un producto de interés personal.	

Historia de Usuario	
<b>Número:</b> 16	<b>Nombre:</b> Agregar producto al carro de compras.
<b>Usuario:</b> Cliente	
<b>Prioridad en negocio:</b> Media (Alta / Media /Baja)	<b>Riesgo en desarrollo:</b> Media (Alta / Media /Baja)
<b>Puntos estimados:</b> 4	<b>Iteración asignada:</b> 7
<b>Descripción:</b> Agregar producto al carro de compras para poder concretar una compra.	

Historia de Usuario	
<b>Número:</b> 17	<b>Nombre:</b> Enviar pedido.
<b>Usuario:</b> Cliente	
<b>Prioridad en negocio:</b> Media (Alta / Media /Baja)	<b>Riesgo en desarrollo:</b> Media (Alta / Media /Baja)
<b>Puntos estimados:</b> 2	<b>Iteración asignada:</b> 7
<b>Descripción:</b> Enviar pedido, de productos seleccionados para su tramitación, registrando datos importantes para envío.	

Historia de Usuario	
<b>Número:</b> 18	<b>Nombre:</b> Acceso de clientes registrados.
<b>Usuario:</b> Cliente	
<b>Prioridad en negocio:</b> Media (Alta / Media /Baja)	<b>Riesgo en desarrollo:</b> Media (Alta / Media /Baja)
<b>Puntos estimados:</b> 3	<b>Iteración asignada:</b> 8
<b>Descripción:</b> Ingresar mediante un formulario de acceso a la parte privada y sección de usuarios registrados, también se ven los pedidos o compras realizadas.	

Historia de Usuario	
<b>Número:</b> 19	<b>Nombre:</b> Registro de cliente.
<b>Usuario:</b> Cliente	
<b>Prioridad en negocio:</b> Media (Alta / Media /Baja)	<b>Riesgo en desarrollo:</b> Media (Alta / Media /Baja)
<b>Puntos estimados:</b> 4	<b>Iteración asignada:</b> 8
<b>Descripción:</b> Realizar registro en el sistema y dar de alta.	

#### 4.5.1.3. Tareas a Realizar

Las historias de usuario son traducidas a tareas y de acuerdo a estas se estiman los tiempos para su realización. La estimación se refiere al esfuerzo necesario para implementar la historia de usuario en una escala de 1 – 5 (menos - más)

Cada tarea se estima de uno a tres días de programación, a continuación se muestran las tareas realizadas, correspondiente a cada historia de usuario:

**Historia de usuario N° 1:** Instalación de sistema e-commerce

Tarea de Ingeniería	
N° 1	Historia de usuario N° 1
<b>Nombre tarea:</b> Instalador de Diseño de interfaz de acceso.	
<b>Tipo de tarea:</b> Desarrollo Desarrollo/Corrección/Mejora/Otra	<b>Puntos estimados:</b> 5
Programador responsable:	
<b>Descripción:</b> Desarrollar formulario para instalar el sistema de administración de contenido, donde se ingresarán datos información de la base de datos a utilizar, datos de acceso al administrador; además el módulo generador de tablas de la base de datos.	

**Historia de usuario N° 2:** Acceso al administrador

Tarea de Ingeniería	
N° 2	Historia de usuario N° 2
<b>Nombre tarea:</b> Diseño de formulario de acceso.	
<b>Tipo de tarea:</b> Desarrollo Desarrollo/Corrección/Mejora/Otra	<b>Puntos estimados:</b> 4
Programador responsable:	
<b>Descripción:</b> Diseñar formulario para acceder al administrador (sección privada), validando los datos.	

Tarea de Ingeniería	
N° 3	Historia de usuario N° 2
<b>Nombre tarea:</b> Diseñar interfaz de administrador (sección privada)	
<b>Tipo de tarea:</b> Desarrollo Desarrollo/Corrección/Mejora/Otra	<b>Puntos estimados:</b> 4
Programador responsable:	
<b>Descripción:</b> Diseñar interfaz, la cual tendrá un menú principal, y area de visualización de datos.	

**Historia de usuario N° 3:** Administración de pedidos.

Tarea de Ingeniería	
N° 4	Historia de usuario N° 3
<b>Nombre tarea:</b> Administrar datos de pedidos.	
<b>Tipo de tarea:</b> Desarrollo Desarrollo/Corrección/Mejora/Otra	<b>Puntos estimados:</b> 3
Programador responsable:	
<b>Descripción:</b> Mostrar pedidos en una lista, cambiar estado de pedido.	

Tarea de Ingeniería	
N° 5	Historia de usuario N° 3
<b>Nombre tarea:</b> Buscar pedidos.	
<b>Tipo de tarea:</b> Desarrollo Desarrollo/Corrección/Mejora/Otra	<b>Puntos estimados:</b> 3
Programador responsable:	
<b>Descripción:</b> Se realizará búsqueda de pedidos por intervalos de fecha o por un determinado término.	

Tarea de Ingeniería	
N° 6	Historia de usuario N° 3
<b>Nombre tarea:</b> Enviar mensaje a cliente.	
<b>Tipo de tarea:</b> Desarrollo Desarrollo/Corrección/Mejora/Otra	<b>Puntos estimados:</b> 3
Programador responsable:	
<b>Descripción:</b> Se permitirá enviar mensaje a cliente que ha realizado pedido, para informarle sobre el estado de su pedido.	

**Historia de usuario N° 4:** Administración de clientes.

Tarea de Ingeniería	
N° 8	Historia de usuario N° 4
<b>Nombre tarea:</b> Crear grupos de clientes.	
<b>Tipo de tarea:</b> Desarrollo Desarrollo/Corrección/Mejora/Otra	<b>Puntos estimados:</b> 3
Programador responsable:	
<b>Descripción:</b> Administrar grupos de clientes, los cuales se mostraran en una lista, realizar acciones crear, editar, eliminar.	

Tarea de Ingeniería	
N° 7	Historia de usuario N° 4
<b>Nombre tarea:</b> Administrar datos de clientes.	
<b>Tipo de tarea:</b> Desarrollo Desarrollo/Corrección/Mejora/Otra	<b>Puntos estimados:</b> 3
Programador responsable:	
<b>Descripción:</b> Mostrar clientes registrados en una lista, realizar acciones crear, editar, eliminar.	

**Historia de usuario N° 5:** Administración de categorías de productos.

Tarea de Ingeniería	
<b>N° 9</b>	Historia de usuario N° 5
<b>Nombre tarea:</b> Administrar categorías de productos.	
<b>Tipo de tarea:</b> Desarrollo Desarrollo/Corrección/Mejora/Otra	<b>Puntos estimados:</b> 3
Programador responsable:	
<b>Descripción:</b> Mostrar categorías registradas en una lista, realizar acciones crear, editar, eliminar, organizar.	

**Historia de usuario N° 6:** Administración de productos.

Tarea de Ingeniería	
<b>N° 10</b>	Historia de usuario N° 6
<b>Nombre tarea:</b> Administrar datos de productos.	
<b>Tipo de tarea:</b> Desarrollo Desarrollo/Corrección/Mejora/Otra	<b>Puntos estimados:</b> 3
Programador responsable:	
<b>Descripción:</b> Mostrar productos registrados en una lista, realizar acciones activar o desactivar productos, asignar a productos destacados.	

Tarea de Ingeniería	
<b>N° 11</b>	Historia de usuario N° 6
<b>Nombre tarea:</b> Otras acciones para administración de productos.	
<b>Tipo de tarea:</b> Desarrollo Desarrollo/Corrección/Mejora/Otra	<b>Puntos estimados:</b> 3
Programador responsable:	
<b>Descripción:</b> Realizar acciones crear, editar, eliminar, agregar productos relacionados, asignar producto a una categoría.	

Tarea de Ingeniería	
<b>N° 12</b>	Historia de usuario N° 6
<b>Nombre tarea:</b> Buscar productos.	
<b>Tipo de tarea:</b> Desarrollo Desarrollo/Corrección/Mejora/Otra	<b>Puntos estimados:</b> 3
Programador responsable:	
<b>Descripción:</b> Buscar productos por categorías o por un término.	

**Historia de usuario N° 7:** Administración de cupones de descuentos.

Tarea de Ingeniería	
N° 13	Historia de usuario N° 7
<b>Nombre tarea:</b> Administrar datos de productos.	
<b>Tipo de tarea:</b> Desarrollo Desarrollo/Corrección/Mejora/Otra	<b>Puntos estimados:</b> 4
Programador responsable:	
<b>Descripción:</b> Mostrar cupones registrados en una lista, realizar acciones crear editar, eliminar.	

**Historia de usuario N° 8:** Generación de reportes.

Tarea de Ingeniería	
N° 14	Historia de usuario N° 8
<b>Nombre tarea:</b> Mostrar productos más vendidos.	
<b>Tipo de tarea:</b> Desarrollo Desarrollo/Corrección/Mejora/Otra	<b>Puntos estimados:</b> 3
Programador responsable:	
<b>Descripción:</b> Mostrar lista de productos más vendidos y realizar búsqueda por intervalos de fechas.	

Tarea de Ingeniería	
N° 15	Historia de usuario N° 8
<b>Nombre tarea:</b> Mostrar ventas por años y meses.	
<b>Tipo de tarea:</b> Desarrollo Desarrollo/Corrección/Mejora/Otra	<b>Puntos estimados:</b> 3
Programador responsable:	
<b>Descripción:</b> Mostrar lista de productos vendidos según año y mes.	

**Historia de usuario N° 9:** Configuración de contenidos de tienda.

Tarea de Ingeniería	
N° 16	Historia de usuario N° 9
<b>Nombre tarea:</b> Administrar banner de portal.	
<b>Tipo de tarea:</b> Desarrollo Desarrollo/Corrección/Mejora/Otra	<b>Puntos estimados:</b> 4
Programador responsable:	
<b>Descripción:</b> Mostrar banners guardados en una lista, subir imágenes, realizar acciones: crear, editar, eliminar.	

Tarea de Ingeniería	
N° 17	Historia de usuario N° 9
<b>Nombre tarea:</b> Administrar logo del portal.	
<b>Tipo de tarea:</b> Desarrollo Desarrollo/Corrección/Mejora/Otra	<b>Puntos estimados:</b> 4
Programador responsable:	
<b>Descripción:</b> Mostrar logo guardado, subir imagen, realizar acciones crear, editar, eliminar.	

Tarea de Ingeniería	
N° 18	Historia de usuario N° 9
<b>Nombre tarea:</b> Administrar páginas de contenidos del portal.	
<b>Tipo de tarea:</b> Desarrollo Desarrollo/Corrección/Mejora/Otra	<b>Puntos estimados:</b> 4
Programador responsable:	
<b>Descripción:</b> Mostrar páginas creadas en una lista, realizar acciones crear, editar, eliminar y ver página.	

**Historia de usuario N° 10:** Administración de usuarios del sistema (administradores).

Tarea de Ingeniería	
N° 19	Historia de usuario N° 10
<b>Nombre tarea:</b> Administrar usuario del sistema.	
<b>Tipo de tarea:</b> Desarrollo Desarrollo/Corrección/Mejora/Otra	<b>Puntos estimados:</b> 3
Programador responsable:	
<b>Descripción:</b> Mostrar administradores registrados en una lista, realizar acciones crear, editar, eliminar y asignarle módulo a administrar.	

**Historia de usuario N° 11:** Administración de ubicaciones geográficas.

Tarea de Ingeniería	
N° 20	Historia de usuario N° 11
<b>Nombre tarea:</b> Administrar ubicaciones geográficas.	
<b>Tipo de tarea:</b> Desarrollo Desarrollo/Corrección/Mejora/Otra	<b>Puntos estimados:</b> 3
Programador responsable:	
<b>Descripción:</b> Mostrar países registrados en una lista, realizar acciones crear, editar, eliminar, crear editar y eliminar zonas.	

**Historia de usuario N° 12:** Configuración de medios de pagos.

Tarea de Ingeniería	
N° 21	Historia de usuario N° 12
<b>Nombre tarea:</b> Configurar medio de pago.	
<b>Tipo de tarea:</b> Desarrollo Desarrollo/Corrección/Mejora/Otra	<b>Puntos estimados:</b> 4
Programador responsable:	
<b>Descripción:</b> Crear módulos de medio de pago y realizar acciones instalar o desinstalar.	

**Historia de usuario N° 13:** Interfaz principal de tienda.

Tarea de Ingeniería	
N° 22	Historia de usuario N° 13
<b>Nombre tarea:</b> Desarrollar interfaz principal del sistema	
<b>Tipo de tarea:</b> Desarrollo Desarrollo/Corrección/Mejora/Otra	<b>Puntos estimados:</b> 4
Programador responsable:	
<b>Descripción:</b> Crear interfaz de tienda virtual, mostrar menú principal, secciones de categorías, productos destacados, banner, logo, contenidos de páginas.	

**Historia de usuario N° 14:** Búsqueda de productos en el catálogo

Tarea de Ingeniería	
N° 23	Historia de usuario N° 14
<b>Nombre tarea:</b> Buscar producto por un término de búsqueda.	
<b>Tipo de tarea:</b> Desarrollo Desarrollo/Corrección/Mejora/Otra	<b>Puntos estimados:</b> 4
Programador responsable:	
<b>Descripción:</b> Desarrollar buscador de términos y mostrar lista de productos encontrados.	

**Historia de usuario N° 15:** Consulta de catálogo.

Tarea de Ingeniería	
N° 24	Historia de usuario N° 15
<b>Nombre tarea:</b> Implementar catálogo.	
<b>Tipo de tarea:</b> Desarrollo Desarrollo/Corrección/Mejora/Otra	<b>Puntos estimados:</b> 2
Programador responsable:	
<b>Descripción:</b> Mostrar lista de categorías de manera ordenada, realizar navegación por categorías y subcategorías, mostrar productos en cada categoría.	

**Historia de usuario N° 16:** Agregar producto al carro de compras.

Tarea de Ingeniería	
N° 25	Historia de usuario N° 16
<b>Nombre tarea:</b> Agregar producto al carro de compras.	
<b>Tipo de tarea:</b> Desarrollo Desarrollo/Corrección/Mejora/Otra	<b>Puntos estimados:</b> 4
Programador responsable:	
<b>Descripción:</b> El cliente selecciona la opción de añadir un producto al carro de compras y se muestra en la lista de productos a comprar, calcular costos.	

**Historia de usuario N° 17:** Enviar pedido.

Tarea de Ingeniería	
N° 26	Historia de usuario N° 17
<b>Nombre tarea:</b> Enviar pedidos.	
<b>Tipo de tarea:</b> Desarrollo Desarrollo/Corrección/Mejora/Otra	<b>Puntos estimados:</b> 7
Programador responsable:	
<b>Descripción:</b> Llenar datos de compra, direcciones de envío, datos de facturación y asignar forma de pago, enviar solicitud.	

**Historia de usuario N° 18:** Acceso de clientes registrados.

Tarea de Ingeniería	
N° 27	Historia de usuario N° 18
<b>Nombre tarea:</b> Acceso del cliente.	
<b>Tipo de tarea:</b> Desarrollo Desarrollo/Corrección/Mejora/Otra	<b>Puntos estimados:</b> 4
Programador responsable:	
<b>Descripción:</b> El cliente accede a su información mediante un formulario donde se validan sus datos.	

Tarea de Ingeniería	
N° 28	Historia de usuario N° 18
<b>Nombre tarea:</b> Consultar pedidos.	
<b>Tipo de tarea:</b> Desarrollo Desarrollo/Corrección/Mejora/Otra	<b>Puntos estimados:</b> 4
Programador responsable:	
<b>Descripción:</b> El usuario debe estar registrado y al acceder, el sistema muestra al usuario su historial de pedidos.	

**Historia de usuario N° 19:** Registro de cliente.

Tarea de Ingeniería	
N° 29	Historia de usuario N° 19
<b>Nombre tarea:</b> Registrar cliente.	
<b>Tipo de tarea:</b> Desarrollo Desarrollo/Corrección/Mejora/Otra	<b>Puntos estimados:</b> 4
Programador responsable:	
<b>Descripción:</b> El cliente se registrará mediante un formulario, el sistema valida los datos, la información se guarda en la base de datos y por lo tanto el cliente se da de alta.	

**4.5.1.4. Plan de Entregas**

De acuerdo a las historias de usuario consideradas para el desarrollo del sistema se ha realizado el plan de entregas, el cual tiene las iteraciones para cada tarea. Este plan se encuentra desarrollado tomando en cuenta el esfuerzo y la prioridad de cada historia de usuario.

Cuadro N° 8  
Prioridad de las historias de usuario

Nombre de Historia	Prioridad	Esfuerzo
<b>Historia 1</b> Instalación de sistema e-commerce.	Alta	5
<b>Historia 2</b> Acceso al administrador	Alta	4
<b>Historia 3</b> Administración de pedidos	Media	3
<b>Historia 4</b> Administración de clientes	Alta	3
<b>Historia 5</b> Administración del categorías de productos	Alta	3
<b>Historia 6</b> Administración de productos	Alta	3
<b>Historia 7</b> Creación de cupones de descuentos	Baja	4
<b>Historia 8</b> Generación de reportes	Baja	3
<b>Historia 9</b> Configuración de contenidos de tienda	Media	4
<b>Historia 10</b> Administración de usuarios del sistema (administradores)	Media	3
<b>Historia 11</b> Administración de ubicaciones geográficas	Baja	3
<b>Historia 12</b> Configuración de medios de pagos	Media	4
<b>Historia 13</b> Interfaz principal de tienda	Alta	4
<b>Historia 14</b> Búsqueda de productos en el catálogo	Media	4
<b>Historia 15</b> Consulta de catálogo	Media	2
<b>Historia 16</b> Agregar producto al carro de compras	Media	4
<b>Historia 17</b> Enviar pedido	Media	2
<b>Historia 18</b> Acceso de clientes registrados	Media	4
<b>Historia 19</b> Registro de cliente	Media	4

Fuente: Elaboración propia

Evaluando cada historia de usuario se puede determinar en qué iteración se desarrollará la misma, a partir de esto se determina las fechas de inicio y fin para las entregas. A continuación se muestra el cuadro de entregas:

Cuadro N° 9  
Fechas de entregas para cada una de las historias de Usuario

Iteraciones	Historia	Prioridad	Esfuerzo	Fecha Inicio	Fecha Final
Iteración 1	Historia 1	Alta	5	04-06-2012	22-06-2012
	Historia 2	Alta	4		
Iteración 2	Historia 4	Alta	3	25-06-2012	06-07-2012
	Historia 5	Alta	3		
Iteración 3	Historia 6	Alta	3	09-07-2012	20-07-2012
	Historia 3	Media	3		
Iteración 4	Historia 9	Media	4	23-07-2012	03-08-2012
	Historia 10	Media	3		
Iteración 5	Historia 12	Media	4	06-07-2012	17-08-2012
Iteración 6	Historia 13	Media	4	20-08-2012	07-09-2012
	Historia 14	Media	4		
	Historia 15	Media	2		
Iteración 7	Historia 16	Media	4	10-09-2012	21-09-2012
	Historia 17	Media	2		
Iteración 8	Historia 18	Media	4	24-09-2012	05-10-2012
	Historia 19	Media	4		
Iteración 9	Historia 7	Baja	4	08-10-2012	19-10-2012
	Historia 8	Baja	3		
Iteración 10	Historia 11	Baja	3	22-10-2012	02-11-2012

Fuente: Elaboración propia

La mayoría de historias de prioridad alta corresponden a la sección privada (administrador) del sistema, ya que con la información ingresada se podrá desarrollar la parte pública.

El proyecto fue dividido en diez iteraciones, por consiguiente se obtuvo un total de diez entregas.

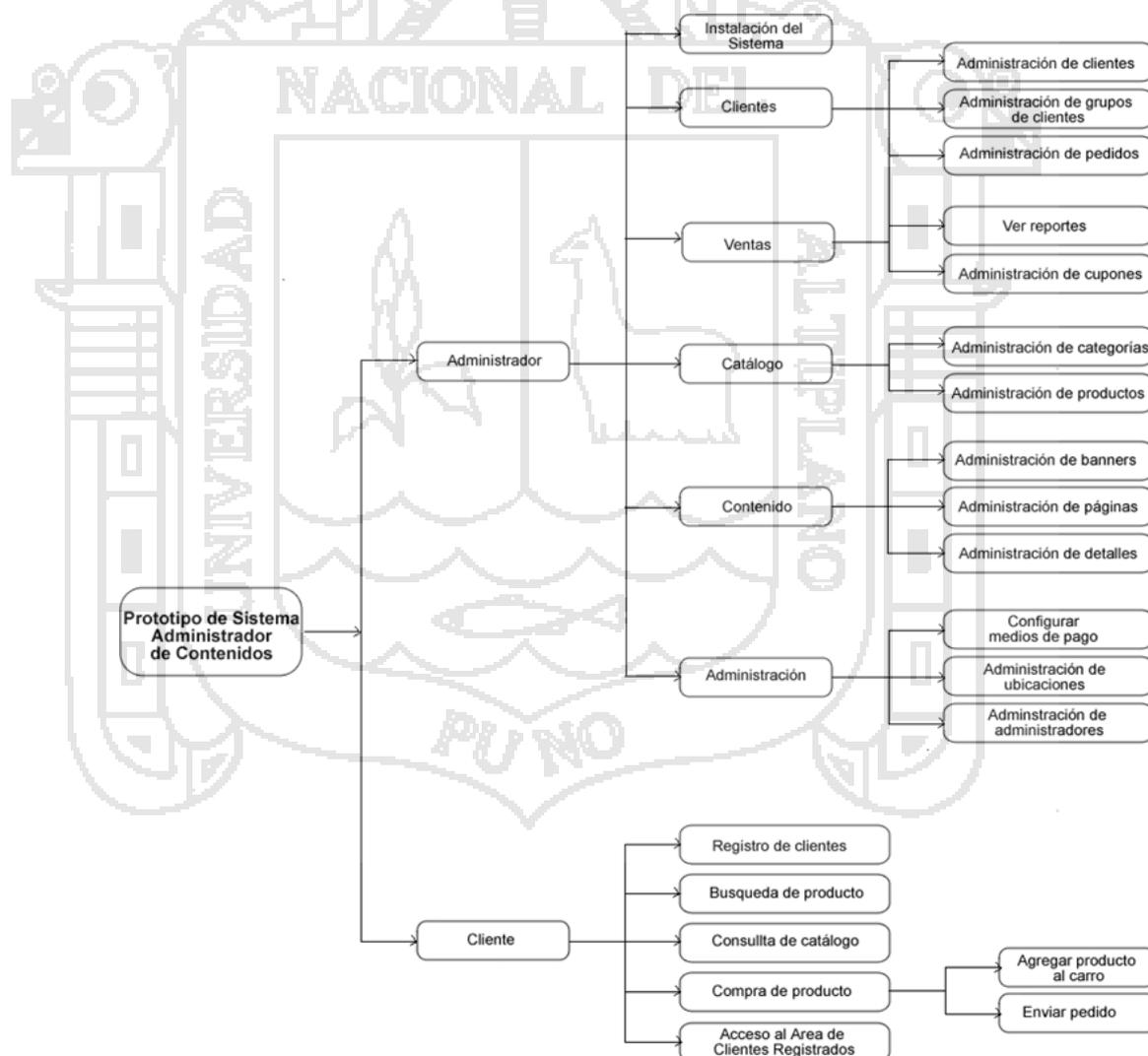
Para aproximar el tiempo que demoraría cada iteración, se tomó como medida en su mayoría dos semanas, esto fue correcto, ya que se cumplieron dichos plazos y se realizaron entregas completas, es decir, sin posponer historias para posteriores iteraciones.

En cada una de las entregas se realizaba las pruebas de aceptación correspondientes, con esto se aseguraba el desarrollo exitoso del sistema.

#### 4.5.1.5. Diagrama de Contexto de la Aplicación

Para poder identificar el entorno del sistema, es decir, sus alcances y límites, se realiza un diagrama de contexto formado según los requerimientos funcionales (historias de usuario), el cual puede dar una idea del manejo del sistema resultante al usuario interesado. En la figura 13 se presenta la manera cómo funciona un CMS, este es un gestor de contenidos conectada a una base de datos para poder integrar toda la información.

Figura N° 13  
Diagrama de contexto de la aplicación



Según las historias de usuario se diseñó el diagrama de contexto de la aplicación, como se observa en la figura N°4.1 el prototipo de Sistema de Administrador de Contenidos consta de dos partes: la parte del privada perteneciente al administrador y la parte pública correspondiente a la tienda virtual donde el cliente interactúa con la tienda.

La sección del administrador está dividida en los siguientes módulos:

- Clientes
- Ventas
- Catálogo
- Contenido
- Administración

La sección pública consta de los siguientes módulos:

- Registro de clientes
- Búsqueda de productos
- Consulta de catálogo
- Compra de producto
- Area de clientes registrados

#### **4.6. DISEÑO DEL PROTOTIPO DE SISTEMA DE ADMINISTRACIÓN DE CONTENIDOS**

A diferencia de las metodologías pesadas, el diseño se realiza durante todo el tiempo de vida del proyecto, siendo constantemente revisado y muy probablemente modificado debido a cambios presentados durante el desarrollo. Según XP el diseño debe ser sencillo donde se crearan sólo los diagramas necesarios.

El diseño del sistema de administración de contenidos comienza evaluando los requisitos funcionales los cuales se mostraron en las historias de usuario desarrolladas y determinando la implementación óptima para cumplir estos requisitos. Existen numerosas herramientas de análisis para reunir y evaluar requisitos de aplicación. El análisis de casos de uso es una de estas herramientas, e identifica los actores en un sistema y las operaciones que pueden realizar sobre éste.

El presente sistema es un caso de aplicación de comercio electrónico, el cliente selecciona elementos de un catálogo, los lleva al carro de compras donde selecciona la cantidad de cada producto y la aplicación muestra los totales de la compra. Cuando el cliente tiene los artículos que desea en el carro de compras, realiza el pedido, suministrando los datos de envío y el método de pago.

La siguiente figura N°14 muestra un diagrama de casos de uso de alto nivel, para el sistema. El diagrama muestra los actores potenciales del sistema y sus acciones.

- Un cliente navega por el catálogo, gestiona la cesta y realiza pedidos.
- Un administrador gestiona la tienda.

Figura N° 14  
Diagrama de casos de uso de alto nivel



#### 4.6.1. Metáfora del Sistema

Sistema de Administración de Contenidos que permite implementar y facilitar el mantenimiento de portales Web orientados al E-commerce.

#### 4.6.2. Tarjetas CRC

La principal funcionalidad que tienen estas, es ayudar a dejar el pensamiento procedimental para incorporarse al enfoque orientado a objetos. Cada tarjeta representa una clase con su nombre en la parte superior, en la sección inferior están descritas las responsabilidades y debajo las clases que le sirven de soporte.

Las tarjetas CRC más importantes se muestran a continuación:

<b>Clase: Administrador</b>
<b>Responsabilidades:</b>
Campos: Id Nombres Apellidos Correo Acceso Contraseña  Métodos: Crear nuevo administrador. Editar datos de administrador. Eliminar administrador. Asignar acceso restringido.
<b>Colaboración:</b>

<b>Clase: Productos</b>
<b>Responsabilidades:</b>
Campos: Id Código Nombre Id_ruta Descripción Precio Destacado Cantidad Envío Impuesto Productos relacionados Imágenes Activado  Métodos: Crear nuevo producto. Editar datos de producto. Eliminar producto. Realizar copia o duplicado de producto. Buscar un determinado producto en lista. Asignar a productos destacados. Asignar a una determinada categoría. Asignar productos relacionados. Activar y desactivar publicación de producto. Asignarle impuestos al producto. Asignarle montos de envío a producto.
<b>Colaboración:</b>



<b>Clase: Clientes</b>
<b>Responsabilidades:</b>
Campos: Id Nombres Apellidos Correo Telefono Empresa Dirección Usuario Contraseña Activo Confirmado Métodos: Crear nuevo cliente. Editar datos de cliente. Eliminar cliente. Activar o desactivar cliente. Asignar cliente a grupo de clientes. Crear direcciones de cliente.
<b>Colaboración:</b>

<b>Clase: Categorías de productos</b>
<b>Responsabilidades:</b>
Campos: Id Parent_id Nombre Descripción Secuencia Métodos: Crear nueva categoría. Editar datos de categoría. Eliminar categoría.
<b>Colaboración:</b>
Clase productos

<b>Clase: Pedidos</b>
<b>Responsabilidades:</b>
<p>Campos:</p> <ul style="list-style-type: none"> <li>Id</li> <li>Nro_orden</li> <li>Estado</li> <li>Fecha_pedido</li> <li>Fecha_envio</li> <li>Impuesto</li> <li>Total</li> <li>Subtotal</li> <li>Cupon_desc</li> <li>Envío</li> <li>Medio_pago</li> </ul> <p>Métodos:</p> <ul style="list-style-type: none"> <li>Cambiar estado de pedidos.</li> <li>Enviar correo a cliente.</li> <li>Eliminar pedidos.</li> <li>Visualizar datos de pedido.</li> <li>Crear etiqueta de embalaje con datos de pedido.</li> <li>Buscar pedidos por fechas y por palabras.</li> </ul>
<b>Colaboración:</b>
Clase productos, clase cupones
<b>Clase: Cupones</b>
<b>Responsabilidades:</b>
<p>Campos:</p> <ul style="list-style-type: none"> <li>Id</li> <li>Código</li> <li>Fecha_inicio</li> <li>Fecha_fin</li> <li>Cupon_todo</li> <li>Max_product</li> <li>Uso_max</li> <li>Modo_descuento</li> <li>Cantidad_descuento</li> </ul> <p>Métodos:</p> <ul style="list-style-type: none"> <li>Crear nuevo cupón.</li> <li>Editar datos de cupón.</li> <li>Eliminar cupón.</li> <li>Mostrar cantidad de uso de cupón.</li> </ul>
<b>Colaboración:</b>
Clase productos

<b>Clase: Páginas de contenido</b>
<b>Responsabilidades:</b>
Campos: Id Parent_id Titulo Titulo_menu Contenido Posicion
Métodos: Crear nueva página. Editar datos de página. Eliminar página.
<b>Colaboración:</b>

<b>Clase: Banners</b>
<b>Responsabilidades:</b>
Campos: Id Titulo Habilitado Deshabilitado Imagen
Métodos: Agregar nuevo banner. Editar banner. Eliminar banner.
<b>Colaboración:</b>

#### 4.6.3. Diseño de la Base de Datos

En la figura 4.4 se muestra el diseño de la base de datos del sistema de administración de contenidos.



**a. Tabla Categorías**

La tabla de categorías se utiliza para almacenar las categorías en que se organizar los productos de la tienda.

Cuadro N° 10  
Tabla categorías

Nombre del Campo	Tipo de Datos	Descripción
IdCategoria	Autonumérico	Clave primaria
CatPadre	Texto	Categoría padre
Nombre	Texto	Nombre de categoría
Descripcion	Texto	Descripción de categoría
Secuencia	Número	Orden de categoría

**b. Tabla Productos**

La tabla Productos almacena los productos que están disponibles en la tienda. Cada producto contiene la descripción y precio, así como la categoría a la que pertenece un producto. Además, es posible seleccionar que el producto aparezca en el escaparate de la tienda, mediante el campo destacado.

Cuadro N° 11  
Tabla productos

Nombre del Campo	Tipo de Datos	Descripción
IdProducto	Autonumérico	Clave principal
Codigo	Número	Código del producto
Nombre	Texto	Nombre del producto
Descripcion	Texto	Descripción del producto
Precio	Doble	Precio del producto
Destacado	Booleano	Si es 1 el producto aparece como destacado
Cantidad	Número	Cantidad en stock
Impuesto	Doble	Total de impuesto
Prod_relacionados	Texto	Productos relacionados
Imágenes	Texto	Imágenes del producto
Activado	Booleano	Si es 1 está activado

**c. Tabla Clientes**

Almacena información de los usuarios. Contiene el nombre de usuario y contraseña y otros datos administrativos.

Cuadro N° 12  
Tabla clientes

Nombre del Campo	Tipo de Datos	Descripción
IdCliente	Autonumérico	Clave principal
Nombres	Texto	Nombres del cliente
Apellidos	Texto	Apellidos del cliente
Correo	Texto	Correo del cliente
Teléfono	Texto	Teléfono del cliente
Empresa	Texto	Empresa del cliente
DNI	Texto	DNI del cliente
RUC	Texto	RUC del cliente
Dirección	Texto	Dirección del cliente
Usuario	Texto	Usuario del cliente
Contraseña	Texto	Contraseña del cliente
Grupo_ClienteID	Numero	Clave secundaria

**d. Tabla Pedidos**

La tabla de pedidos almacena la información acerca del pedido, como los datos de entrega del pedido y los datos del cliente.

Cuadro N° 13  
Tabla pedidos

Nombre del Campo	Tipo de Datos	Descripción
IdPedido	Autonumérico	Clave principal
Nro de Orden	Número	Nro de pedido
Fecha pedido	Fecha	Fecha de pedido
Fecha envío	Fecha	Fecha de envío de pedido
Impuesto	Doble	Impuesto aplicado al pedido
Total	Doble	Total del pedido
Subtotal	Doble	Subtotal del pedido
Cupón	Doble	Cupón de descuento del pedido
Medio de pago	Texto	Medio de pago del pedido
Clienteld	Número	Clave secundaria

**e. Tabla Cupones**

La tabla de cupones almacena la información acerca del cupón, como cantidad de descuento, inicio y fin de la promoción.

Cuadro N° 14  
Tabla cupones

Nombre del Campo	Tipo de Datos	Descripción
IdCupon	Autonumérico	Clave primaria
Código	Texto	Código del cupón
Fecha inicio	Fecha	Fecha de inicio del cupon
Fecha fin	Fecha	Fecha de fin del cupón
Uso máximo	Número	Cantidad máxima de usos
Modo de descuento	Texto	Modo de descuento del cupón
Cantidad de descuento	Doble	Cantidad de descuento del cupón

**f. Tabla Administrador**

Almacena información de los usuarios. Contiene el nombre de usuario y contraseña y otros datos administrativos.

Cuadro N° 15  
Tabla administrador

Nombre del Campo	Tipo de Datos	Descripción
IdAdministrador	Autonumérico	Clave primaria
Nombres	Texto	Nombres del administrador
Apellidos	Texto	Apellidos del administrador
Correo	Texto	Correo del administrador
Acceso	Texto	Acceso a determinados módulos
Contraseña	Texto	Contraseña del administrador

#### 4.6.4. Diseño de la Interfaz de Usuario

Como se ha especificado, las interfaces que maneja el CMS propuesto son dos: la interfaz privada (administrativa) y la interfaz pública (propia del portal).

##### A. Sección Privada (Administración)

###### Interfaz Acceso al Administrador

La figura N°16 muestra la estructura de diseño visual de la interfaz de acceso a la administración del portal web donde se puede ingresar a través del correo y la clave que se indicaron al momento de la instalación.

Figura N° 16  
Estructura de diseño visual de interfaz de acceso a la administración



### Interfaz Administrativa

En la figura N° 17 se muestra la estructura que consta de tres secciones, de las cuales, la sección del logo y el menú principal persisten con la misma información cuando se navega por las diversas funcionalidades del sistema.

- Logo: Muestra el logo del sistema
- Menú de navegación principal: Se encuentra en posición horizontal y es el centro de navegación de la zona de administración, cada menú tiene un menú desplegable con diversas opciones.
- Área de visualización de datos: Es el área que se actualiza cuando se hace alguna selección del menú.

Figura N° 17  
Estructura de diseño visual de la interfaz administrativa



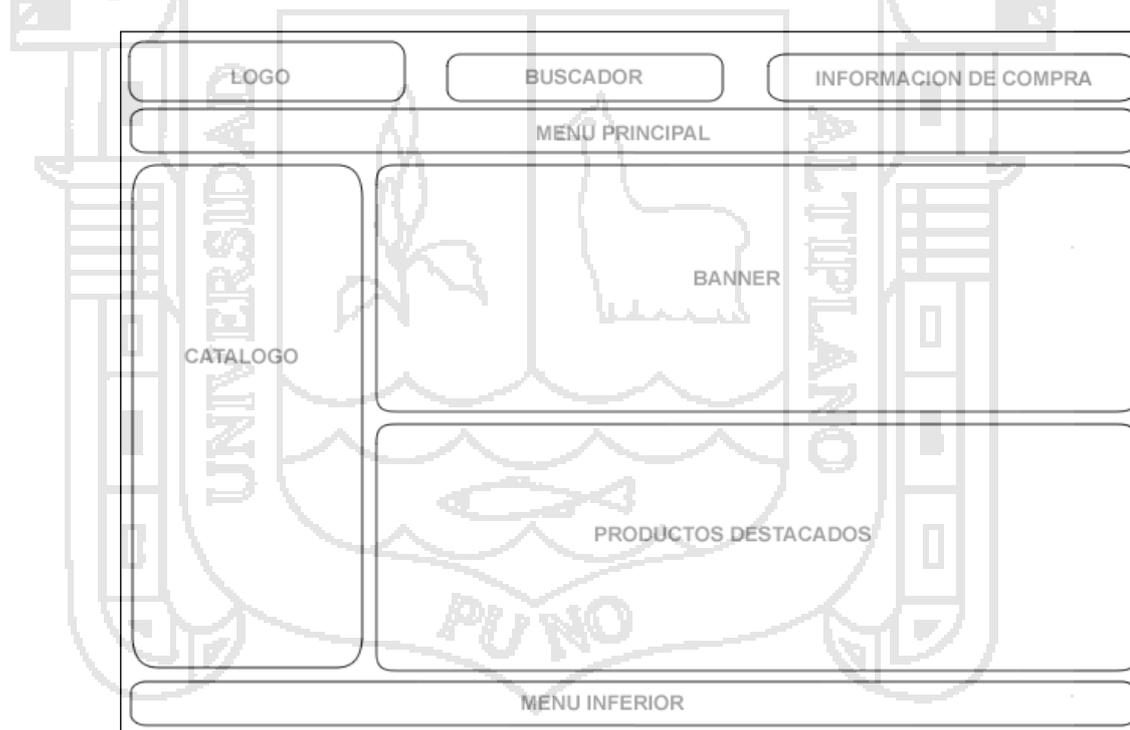
### B. Sección Pública (Portal Web)

### Interfaz de Inicio

Según la figura N° 18, corresponde a la interfaz principal para el usuario público. Se divide en las siguientes secciones:

- Cabecera: Se encuentran el logotipo de la tienda, el módulo de búsqueda y la información de compra de productos (registro, acceso, carro de compras).
- El menú principal: Se encuentra en posición horizontal, muestra contenidos sobre información de la empresa.
- Catálogo: Muestra las categorías de productos de la tienda.
- Banner: Muestra imágenes de acuerdo a las promociones de la tienda.
- Productos destacados: Lista los productos asignados como destacados.
- Menú inferior: Se encuentra en la parte inferior en posición horizontal.

Figura N° 18  
Estructura de diseño visual de la interfaz de inicio del portal web

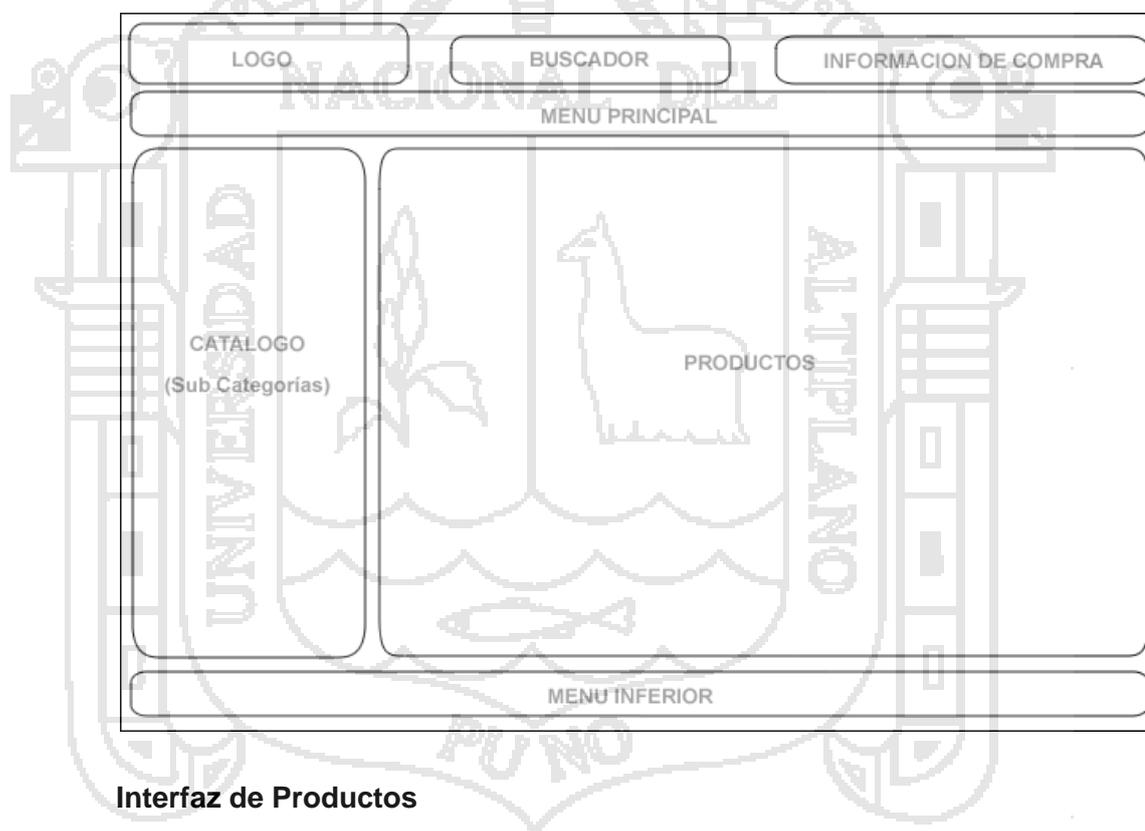


### Interfaz de Navegación de Catálogo

La siguiente interfaz, según la figura N° 19, es la que se muestra a los usuarios cuando realicen navegaciones por el catálogo de la tienda. La navegación se realizará entrando en cada categoría. Mantiene la estructura de

la interfaz principal, se sustituye la parte central e izquierda por la lista de subcategorías de la categoría seleccionada.

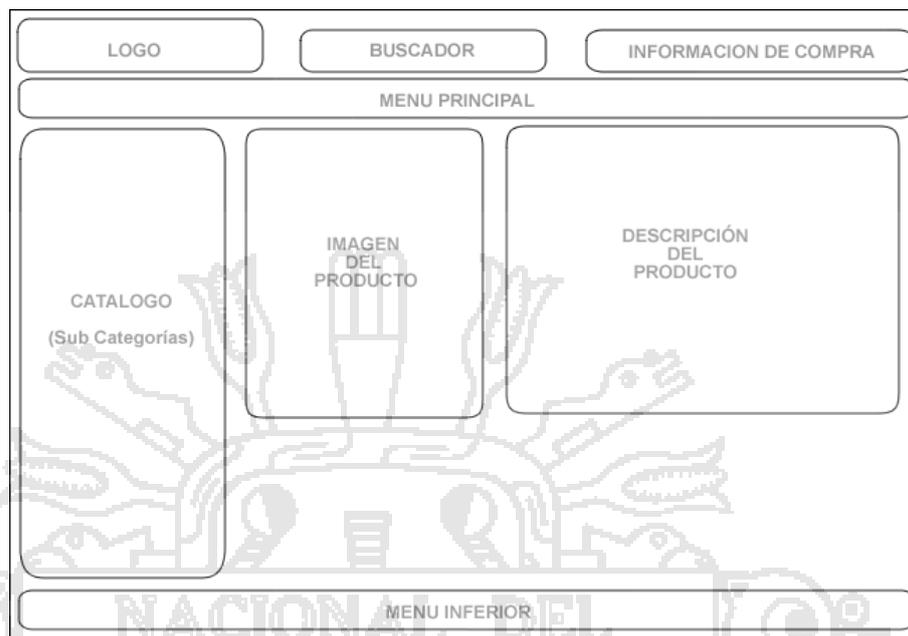
Figura N° 19  
Estructura de diseño visual de la interfaz de navegación del catálogo



### Interfaz de Productos

En la figura N° 20, se muestra la interfaz de productos, después de realizar una búsqueda de productos o al llegar a una categoría o sub categoría, donde ya se encuentran directamente catalogados los artículos.

Figura N° 20  
Estructura de diseño visual de la interfaz de productos

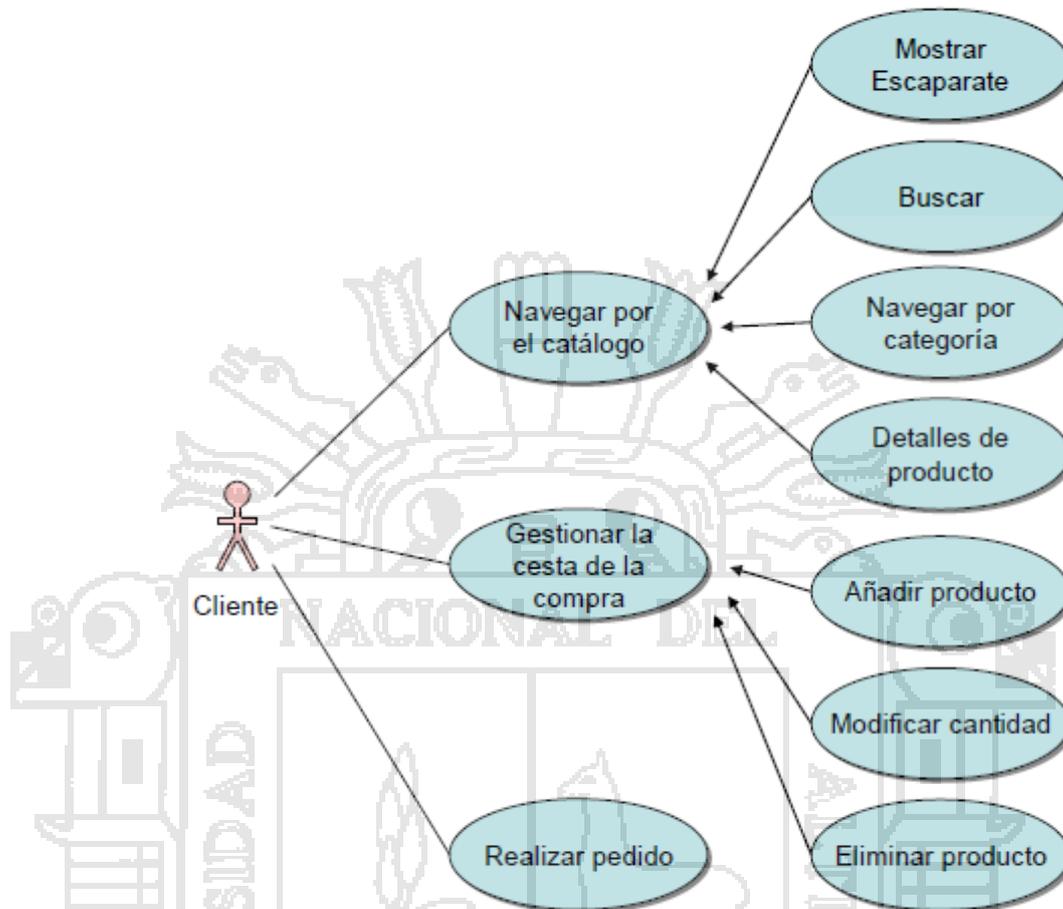


#### 4.6.5. Arquitectura de Sección Pública

La arquitectura MVC se adapta bien a la tienda virtual. Se puede descomponer el sistema en tres bloques lógicos, el primero es el relativo a los aspectos de presentación, el segundo es aquel que trata con el modelo y los datos, y el tercero es el que recibe e intercepta las peticiones del usuario, y controla el flujo para responder estas peticiones.

Partimos la aplicación en módulos y objetos que implementen los diferentes requisitos funcionales. La siguiente figura muestra los casos de uso de la interfaz de usuario del portal Web.

Figura N° 21  
Diagrama de caso de uso del cliente



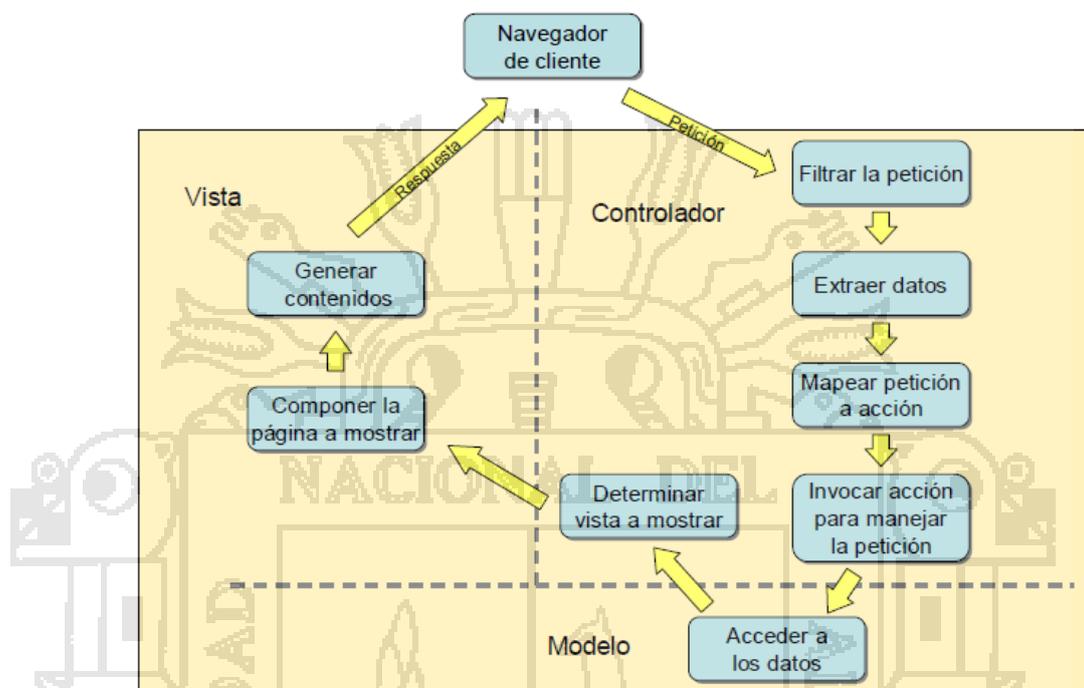
Según la figura N° 21, un cliente que accede al portal Web, espera ver los siguientes elementos:

- Barra de navegación de tareas comunes.
- Un catálogo que proporciona una vista organizada de los productos.
- Un mecanismo de búsqueda.
- Una vista detallada de cada producto.
- Una vista del carro de compras que deja a los clientes revisar y modificar los contenidos de su carro.
- Una vista de pedido, que permite a los clientes introducir información sobre la entrega del pedido y medio de pago.
- Una vista que confirma la compra.

El diseño de la aplicación consiste en, partir la aplicación en bloques pertenecientes al modelo, las vistas y el controlador, como se muestra en la siguiente figura:

Figura N° 22

Arquitectura MVC de la tienda virtual



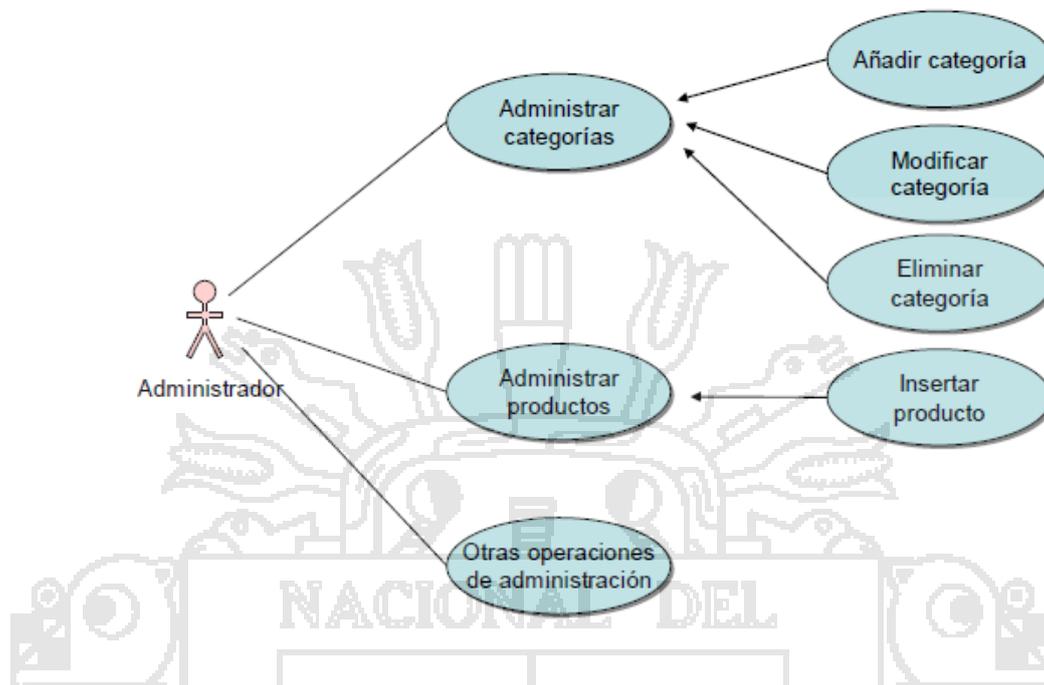
#### 4.6.6. Arquitectura de la Sección Privada (Administración)

Para el diseño de la aplicación de Administración, se han seguido los mismos pasos que para la Tienda Virtual, teniendo en cuenta las siguientes consideraciones:

- El controlador de administración gestiona la seguridad y limita el acceso a la aplicación a los usuarios registrados que tengan permiso de administración.

La siguiente figura muestra los casos de uso del parte de administración:

Figura N° 23  
Diagrama de caso de uso del administrador



El administrador ingresa a la administración y encuentra los siguientes elementos:

- Administrar categorías de productos.
- Administrar productos.
- Administrar clientes.
- Administrar pedidos.
- Configuraciones de tienda.

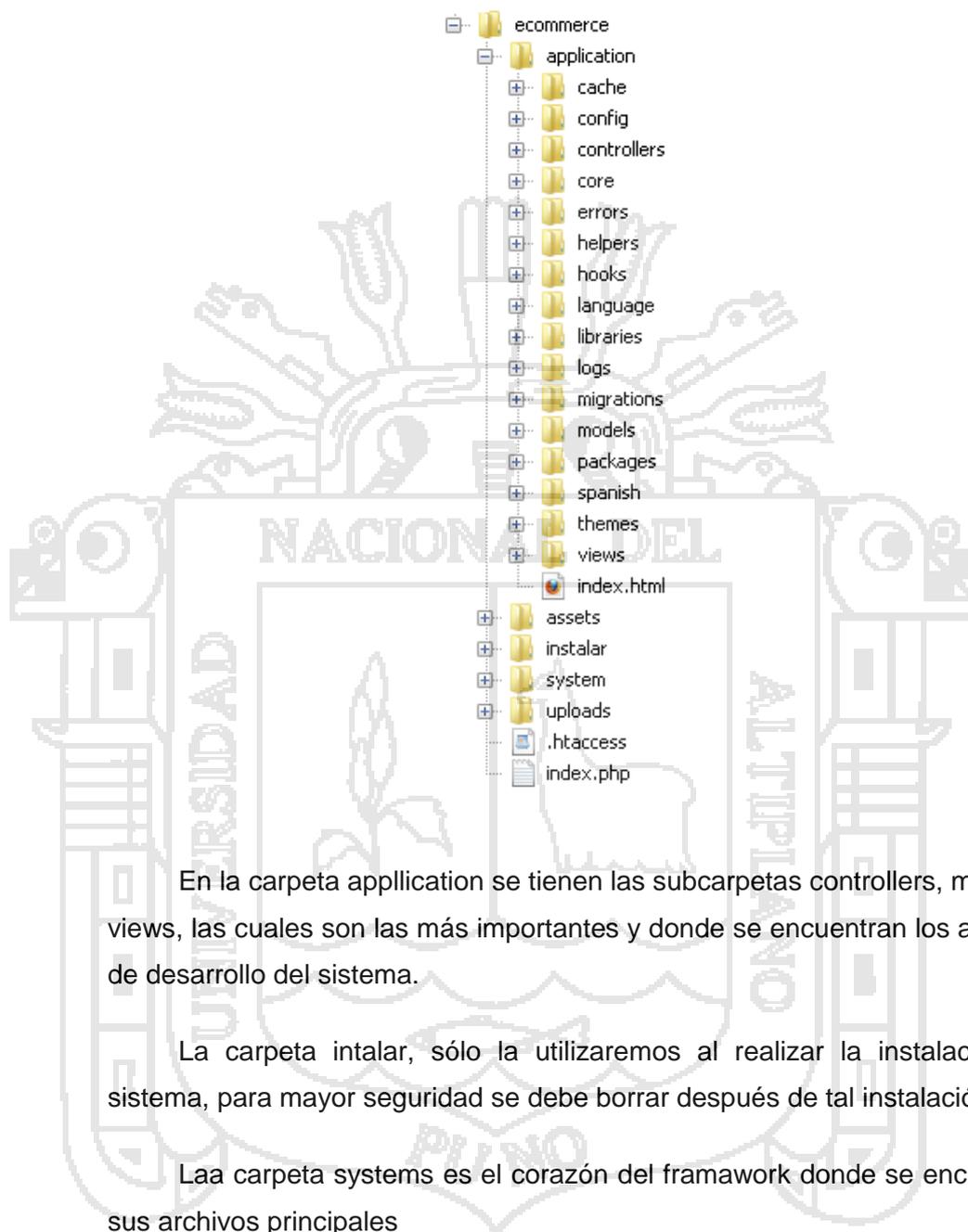
#### 4.7. IMPLEMENTACIÓN DEL PROTOTIPO DE SISTEMA DE ADMINISTRACIÓN DE CONTENIDOS

##### 4.7.1. Organización de la Aplicación

Antes de desarrollar la aplicación veremos la organización de la aplicación en cuanto a carpetas.

Para desarrollar la implementación del sistema, descargamos el framework CodeIgniter, el cual nos permite organizar la aplicación según el contexto MVC (Modelo Vista Controlador). Las carpetas que contiene son:

Figura N° 24  
Organización de la aplicación según CodeIgniter



En la carpeta application se tienen las subcarpetas controllers, models y views, las cuales son las más importantes y donde se encuentran los archivos de desarrollo del sistema.

La carpeta intalar, sólo la utilizaremos al realizar la instalación del sistema, para mayor seguridad se debe borrar después de tal instalación.

Laa carpeta systems es el corazón del framawork donde se encuentran sus archivos principales

#### 4.7.2. Codificación

En metodologías pesadas, la codificación es un proceso al cual sólo se llega después de largas fases de análisis y diseño de las que queda una gran cantidad de documentación a partir de la cual el proceso de codificación es relativamente sencillo. En XP el proceso es muy diferente. Prácticamente

desde un principio se inicia con la codificación, favoreciendo el logro del objetivo de estar haciendo entregas frecuentemente al cliente.

Los principios que se han tomado en cuenta para esta fase son:

- a. Refactorización.
- b. Propiedad colectiva.
- c. Integración continua.
- d. Semana de 40 horas
- e. Cliente in situ.
- f. Estándares de programación

#### 4.7.2.1. Desarrollo del Modelo

El modelo de la tienda contiene las clases necesarias para acceder a la base de datos, de una forma consistente.

Según el framework CodeIgniter las clases modelos reciben una herencia de la clase *CI\_Model*.

La sentencia para acceder a la base de datos es

```
$this->db->(sentencia).
```

La sentencia para llamar al modelo es

```
$this->load->model('Nombre Modelo');
```

A continuación se muestran los modelos principales que conforman el sistema administrador de contenidos:

##### a. La Clase Modelo Producto

En el siguiente código se muestran las funciones principales con las respectivas consultas realizadas a la base de datos, específicamente a la tabla productos, el archivo que lo contiene es *product\_model.php* el cual debe tener el mismo nombre de la clase.

```
Class Producto_model extends CI_Model
{
// Variable correspondiente al descuento por grupo
var $group_discount_formula = false;

function __construct()
{
    parent::__construct();

// Verificar si el usuario es registrado y si se le aplica descuento por
grupo

    $customer = $this->session->userdata('customer');
```

```

        if(isset($customer['group_discount_formula']))
        {
            $this->group_discount_formula = $customer
                                          ['group_discount_formula'];
        }
    }

    //Función productos
    function products($data=array(), $return_count=false)
    {
        if(empty($data))
        {
            //Listar productos de acuerdo a los datos ($data)
            $this->get_all_products();
        }
        else
        {
            //Ordenar productos
            if(!empty($data['order_by']))
            {
                $this->db->order_by($data['order_by'],
                                  $data['sort_order']);
            }
        }

        //Búsqueda por términos
        if(!empty($data['term']))
        {
            $search = json_decode($data['term']);
            if(!empty($search->term))
            {
                $this->db->like('name', $search->term);
                $this->db->or_like('description', $search->term);
                $this->db->or_like('excerpt', $search->term);
                $this->db->or_like('sku', $search->term);
            }

            if(!empty($search->category_id))
            {
                $this->db->join('category_products',
                              'category_products.product_id=products.id', 'right');
                $this->db->where('category_products.category_id', $search->
                               category_id);
                $this->db->order_by('sequence', 'ASC');
            }
        }

        if($return_count)
        {
            return $this->db->count_all_results('products');
        }
        else
        {
            return $this->db->get('products')->result();
        }
    }

    //Función para llamar todos los productos
    function get_all_products()
    {
        $this->db->order_by('name', 'ASC');
        $result = $this->db->get('products');
        $return = $result->result();
        return $return;
    }

    //Función para llamar productos de acuerdo a categorías
    function get_product_categories($id)
    {
        return $this->db->where('product_id', $id)->
            join('categories', 'category_id = categories.id')->
            get('category_products')->result();
    }

    //Función para guardar datos de productos en la base de datos
    function save($product, $options=false, $categories=false)
    {
        if ($product['id'])
        {
            $this->db->where('id', $product['id']);
            $this->db->update('products', $product);
        }
    }

```

```

        $id      = $product['id'];
    }
    else
    {
        $this->db->insert('products', $product);
        $id      = $this->db->insert_id();
    }

    return $id;
}
//Función para eliminar productos
function delete_product($id)
{
    $this->db->where('id', $id);
    $this->db->delete('products');
}
//Función para agregar a categorías
function add_product_to_category($product_id, $optionlist_id,
$sequence)
{
    $this->db->insert('product_categories',
array('product_id'=>$product_id, 'category_id'=>$category_id,
'sequence'=>$sequence));
}
//Función para buscar productos
function search_products($term, $limit=false, $offset=false,
$by=false, $sort=false)
{
    $results      = array();

    $this->db->select('*', LEAST(IFNULL(NULLIF(saleprice, 0), price),
price) as sort_price', false);
    $this->db->where('enabled', 1);
    $this->db->where('(name LIKE "%'.$term.'" OR description LIKE
"%'.$term.'" OR excerpt LIKE "%'.$term.'" OR sku LIKE "%'.$term.'"'));
    $results['count']      = $this->db->count_all_results('products');
    $this->db->select('*', LEAST(IFNULL(NULLIF(saleprice, 0), price),
price) as sort_price', false);
    $this->db->where('enabled', 1);
    $this->db->where('(name LIKE "%'.$term.'" OR description LIKE
"%'.$term.'" OR excerpt LIKE "%'.$term.'" OR sku LIKE "%'.$term.'"'));

    $results['products']  = $this->db->get('products', $limit,
$offset)->result();
    return $results;
}
}
}

```

## b. La Clase Modelo Categorías

En este modelo se muestran las funciones principales con las respectivas consultas realizadas a la base de datos, específicamente a la tabla categorías, Las funciones son referidas a búsquedas y a gestión de categorías (mostrar, guardar, eliminar), el archivo que lo contiene es category\_model.php el cual tiene el mismo nombre de la clase.

Se muestra el código de la clase Category\_Model:

```

Class Category_model extends CI_Model
{
//Funcion obtener categorias
function get_categories($parent = false)
{
    if ($parent !== false)
    {
        $this->db->where('parent_id', $parent);
    }
    $this->db->select('id');
    $this->db->order_by('categories.sequence', 'ASC');
    $this->db->order_by('name', 'ASC');
}
}

```

```

        $result = $this->db->get('categories');
        return $categories;
    }

    //Función para construir la lista de categorías
    function get_categories_tierd($parent=0)
    {
        $categories = array();
        $result = $this->get_categories($parent);
        foreach ($result as $category)
        {
            $categories[$category->id]['category'] =
            $category;
            $categories[$category->id]['children'] =
            $this->get_categories_tierd($category->id);
        }
        return $categories;
    }
    //Función para llamar todas las categorías
    function get_category($id)
    {
        return $this->db->get_where('categories',
        array('id'=>$id))->row();
    }
    //Función para llamar categorías en en administrador
    function get_category_products_admin($id)
    {
        $this->db->order_by('sequence', 'ASC');
        $result = $this->db->get_where('category_products',
        array('category_id'=>$id));
        $result = $result->result();
        $contents = array();
        return $contents;
    }
    //Función para llamar categorías
    function get_category_products($id, $limit, $offset)
    {
        $this->db->order_by('sequence', 'ASC');
        $result = $this->db->get_where('category_products',
        array('category_id'=>$id), $limit, $offset);
        $result = $result->result();
        $contents = array();
        return $contents;
    }
    //Función para guardar y actualizar datos de categorías en la base de datos
    function save($category)
    {
        if ($category['id'])
        {
            $this->db->where('id', $category['id']);
            $this->db->update('categories', $category);

            return $category['id'];
        }
        else
        {
            $this->db->insert('categories', $category);
            return $this->db->insert_id();
        }
    }
    //Función para eliminar categorías
    function delete($id)
    {
        $this->db->where('id', $id);
        $this->db->delete('categories');
        $this->db->where('category_id', $id);
        $this->db->delete('category_products');
    }
}

```

### c. La Clase Modelo Clientes

En este modelo se muestran las funciones principales (gestión) con las respectivas consultas realizadas a la base de datos, específicamente a la

tabla clientes, esta clase almacena de forma temporal los datos del cliente que provienen de la base de datos y se van a utilizar para la presentación, el archivo que lo contiene es customer\_model.php el cual tiene el mismo nombre de la clase.

El siguiente cuadro muestra el código de la clase Customer \_Model:

```

Class Customer_model extends CI_Model
{
//Tiempo de expiracion si no esta usando la aplicacion
var $session_expire = 7200;

function __construct()
{
parent::__construct();
}

//Función obtener lista de clientes
function get_customers($limit=0, $offset=0, $order_by='id',
$direction='DESC')
{
$this->db->order_by($order_by, $direction);
$result= $this->db->get('customers');
return $result->result();
}

//Función obtener un determinado cliente
function get_customer($id)
{
$result= $this->db->get_where('customers', array('id'=>$id));
return $result->row();
}

//Función para llamar direcciones de un determinado cliente
function get_address_list($id)
{
$addresses = $this->db->where('customer_id', $id)-
>get('customers_address_bank')->result_array();
return $addresses;
}

//Función para guardar direcciones de un determinado cliente
function save_address($data)
{
$this->db->where('id', $data['id']);
$this->db->update('customers_address_bank', $data);
return $data['id'];
}

//Función borrar direcciones
function delete_address($id, $customer_id)
{
$this->db->where(array('id'=>$id, 'customer_id'=>$customer_id))-
>delete('customers_address_bank');
return $id;
}

//Función para guardar y actualizar datos de un cliente
function save($customer)
{
if ($customer['id'])
{
$this->db->where('id', $customer['id']);
$this->db->update('customers', $customer);
return $customer['id'];
}
else
{
$this->db->insert('customers', $customer);
return $this->db->insert_id();
}
}

//Función desactivar un cliente

```

```

function deactivate($id)
{
    $customer      = array('id'=>$id, 'active'=>0);
    $this->save_customer($customer);
}
//Función borrar un cliente, es recomendable no borrar, sólo desactivar
function delete($id)
{
    $this->db->where('customer_id', $id);
    $this->db->delete('orders');
}

//Función salir
function logout()
{
    $this->session->unset_userdata('customer');
    $this->go_cart->destroy(false);
}
//Función de acceso de usuario registrado
function login($email, $password, $remember=false)
{
    $this->db->select('*');
    $this->db->where('email', $email);
    $this->db->where('active', 1);
    $this->db->where('password', sha1($password));
    $this->db->limit(1);
    $result = $this->db->get('customers');
    $customer      = $result->row_array();
    return $customer;
}
//Función para verificar si está logueado
function is_logged_in($redirect = false, $default_redirect =
'secure/login/')
{
    $customer = $this->go_cart->customer();
    if (!isset($customer['id']))
    {
        if ($redirect)
        {
            $this->session->set_flashdata('redirect',
$redirect);
        }
        if ($default_redirect)
        {
            redirect($default_redirect);
        }
        return false;
    }
    else
    {
        //Verificar si ha expirado la sesión
        if($customer['expire'] && $customer['expire'] < time())
        {
            $this->logout();
            if($redirect)
            {
                $this->session->set_flashdata('redirect',
$redirect);
            }
            if($default_redirect)
            {
                redirect('secure/login');
            }
            return false;
        }
        else
        {
            if($customer['expire'])
            {
                $customer['expire']=time()+$this->
                session_expire;
            }
        }
    }
}

```



```

Class Order_model extends CI_Model
{
    function __construct()
    {
        parent::__construct();
    }
//Función mostrar pedidos por años y meses
function get_gross_monthly_sales($year)
{
    $this->db->select('SUM(shipping) as shipping');
    $this->db->select('SUM(tax) as tax');
    $this->db->select('SUM(total) as total');
    $this->db->select('YEAR(ordered_on) as year');
    $this->db->select('MONTH(ordered_on) as month');
    $this->db->group_by(array('MONTH(ordered_on)'));
    $this->db->order_by("ordered_on", "desc");
    $this->db->where('YEAR(ordered_on)', $year);

    return $this->db->get('orders')->result();
}
//Función mostrar ventas por años
function get_sales_years()
{
    $this->db->order_by("ordered_on", "desc");
    $this->db->select('YEAR(ordered_on) as year');
    $this->db->group_by('YEAR(ordered_on)');
    $records = $this->db->get('orders')->result();
    $years = array();
    foreach($records as $r)
    {
        $years[] = $r->year;
    }
    return $years;
}
//Función obtener pedido al realizar una búsqueda
function get_orders($search=false, $sort_by='',
    $sort_order='DESC', $limit=0, $offset=0)
{
    if ($search)
    {
        if (!empty($search->term))
        {
            $term = explode(' ', $search->term);
            foreach($term as $t)
            {
                $not = '';
                $operator = 'OR';

                if(substr($t,0,1) == '-')
                {
                    $not = 'NOT ';
                    $operator = 'AND';
                    $t = substr($t,1,strlen($t));
                }

                $like = '';
                $like .= "( `order_number` ".$not."LIKE '%".$t."%' ";
                $like .= $operator." `bill_firstname` ".$not."LIKE '%".$t."%' ";
                $like .= $operator." `bill_lastname` ".$not."LIKE '%".$t."%' ";
                $like .= $operator." `ship_firstname` ".$not."LIKE '%".$t."%' ";
                $like .= $operator." `ship_lastname` ".$not."LIKE '%".$t."%' ";
                $like .= $operator." `status` ".$not."LIKE '%".$t."%' ";
                $like .= $operator." `notes` ".$not."LIKE '%".$t."%' )";
                $this->db->where($like);
            }
        }
        if (!empty($search->start_date))
        {
            $this->db->where('ordered_on >=', $search->start_date);
        }
        if (!empty($search->end_date))
        {
            $search->end_date = date('Y-m-d', strtotime($search->end_date)+86400);
            $this->db->where('ordered_on <', $search->end_date);
        }
    }
    return $this->db->get('orders')->result();
}
}

```

```

//Función obtener un pedido de un determinado cliente
function get_customer_orders($id, $offset=0)
{
    $this->db->join('order_items', 'orders.id' =
order_items.order_id');
    $this->db->order_by('ordered_on', 'DESC');
    return $this->db->get_where('orders', array('customer_id'=>$id),
15, $offset)->result();
}
//Función obtener pedidos
function get_order($id)
{
    $this->db->where('id', $id);
    $result = $this->db->get('orders');
    $order = $result->row();
    $order->contents = $this->get_items($order->id);
    return $order;
}
//Función borrar pedido
function delete($id)
{
    $this->db->where('id', $id);
    $this->db->delete('orders');
    $this->db->where('order_id', $id);
    $this->db->delete('order_items');
}
//Función guardar o actualizar pedido
function save_order($data, $contents = false)
{
    if (isset($data['id']))
    {
        $this->db->where('id', $data['id']);
        $this->db->update('orders', $data);
        $id = $data['id'];
        $order_number = $id;
    }
    else
    {
        $this->db->insert('orders', $data);
        $id = $this->db->insert_id();
        $order = array('order_number'=> date('U').$id);
        $this->db->where('id', $id);
        $this->db->update('orders', $order);
        $order_number = $order['order_number'];
    }
    if($contents)
    {
        $this->db->where('order_id', $id)-
>delete('order_items');
        foreach($contents as $item)
        {
            $save = array();
            $save['contents'] = $item;
            $item = unserialize($item);
            $save['product_id'] = $item['id'];
            $save['quantity'] = $item['quantity'];
            $save['order_id'] = $id;
            $this->db->insert('order_items', $save);
        }
    }
    return $order_number;
}
//Función para listar los productos más vendidos
function get_best_sellers($start, $end)
{
    $orders = $this->db->select('id')->get('orders')->result();
    $items = array();
    foreach($orders as $order)
    {

```

```

        $order_items = $this->db->select('product_id,
quantity')->where('order_id', $order->id)->get('order_items')-
>result_array();
        foreach($order_items as $i)
        {
            if(isset($items[$i['product_id']]))
            {
                $items[$i['product_id']] += $i['quantity'];
            }
            else
            {
                $items[$i['product_id']] = $i['quantity'];
            }
        }
        arsort($items);
        unset($orders);
        $return = array();
        foreach($items as $key=>$quantity)
        {
            $product = $this->db->where('id', $key)->get('products')->row();
            if($product)
            {
                $product->quantity_sold = $quantity;
            }
            else
            {
                $product = (object) array('sku'=>'Deleted', 'name'=>'Deleted',
'quantity_sold'=>$quantity);
            }
            $return[] = $product;
        }
        return $return;
    }
}

```

### 4.7.3. Desarrollo de las Vistas

Las vistas del sistema contienen código html, para poder mostrar las interfaces del sistema

La sentencia para llamar a la vista es:

```
$this->load->view('Nombre de la Vista');
```

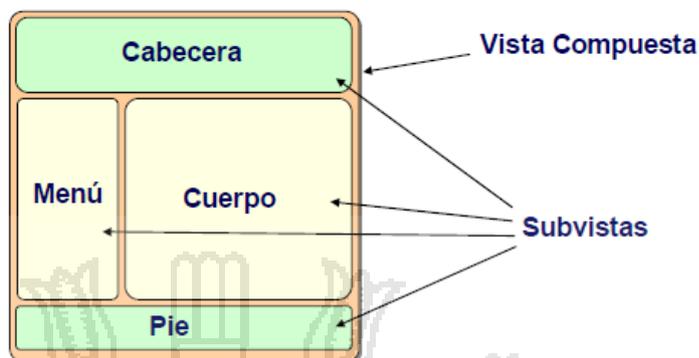
#### 4.7.3.1. Vistas compuestas

La vista compuesta se encarga de ensamblar la página que será enviada al usuario. Contiene la definición del aspecto que tendrá la aplicación completa.

Se hace uso de hojas de cascada de estilos (CSS) para especificar el aspecto de cada tipo de párrafo.

La vista compuesta hace uso del tag *include()*, para incluir una página de forma dinámica. Esta página es el denominado cuerpo, que realmente es el contenido principal de cada una de las páginas.

Figura N° 25  
Vista compuesta



#### a. Vista Compuesta del Portal Web

Se tiene el código de la vista de la sección pública, la cual está compuesta por la vista de cabecera (*header.php*) y la vista de pie de página (*footer.php*), y la vista categorías, el archivo que lo contiene es *homepage.php*.

```
<?php include('header.php'); ?>
<div class="row">
<div class="span3">
<?php include('categories_menu.php'); ?>
<br/>
//Ver cajas
<?php foreach($boxes as $box):?>
<div>
<?php
$box_image='';
if($box->link != '')
{
$target= false;
if($box->new_window)
{
$target = 'target="_blank"';
}
}
echo '<a href="'. $box->link.'"' . $target.' '>'. $box_image.'</a>';
}
else
{
echo $box_image;
}
?>
</div>
<br/>
<?php endforeach;?>
</div>
//Carrusel
<div class="span9">
<div id="myCarousel" class="carousel slide">
<!-- Carousel items -->
<div class="carousel-inner">
<?php
$active_banner = 'active ';
foreach($banners as $banner):?>
<div class="<?php echo $active_banner;?>item">
<?php
```

```

    $banner_image = 'link.' '.$target.'>'. $banner_image.'</a>';
        }
        else
        {
            echo $banner_image;
        }
    }
    ?>

        </div>
        <?php
        $active_banner = false;
        endforeach;?>
    </div>
    <a class="carousel-control left" href="#myCarousel" data-
    slide="prev">&lsaquo;</a>
        <a class="carousel-control right" href="#myCarousel"
    data-slide="next">&rsaquo;</a>
    </div>
    </div>
    //Productos destacados
    <div>
        <h3>Productos Destacados</h3>
        <br/>
        <ul class="thumbnails category_container">
            <?php foreach($products as $product){
                if($product->featured ==1)
                {
                    ?>
                    <li class="fproduct">
                        <?php
                            $base_url      = $this->uri->segment_array();
                            $photo         = theme_img('no_picture.png',
                            lang('no_image_available'));
                            $product->images = array_values((array)json_decode($product->
                            >images));
                            if(!empty($product->images[0]))
                            {
                                $primary      = $product->images[0];
                                foreach($product->images as $photo)
                                {
                                    if(isset($photo->primary))
                                    {
                                        $primary      = $photo;
                                    }
                                }
                            }
                            $photo         = 'seo_title.'"/>';
                            ?>
                            <a class="thumbnail" href="<?php echo site_url(implode('/',
                            $base_url).'/'.$product->slug); ?>">
                            <?php echo $photo; ?>
                            </a>
                            <h5 style="margin-top:5px;"><a href="<?php echo site_url(implode('/',
                            $base_url).'/'.$product->slug); ?>"><?php echo $product->name;?></a></h5>
                            <div>
                            <div>
                            <?php if($product->saleprice > 0):?>
                            <span class="price-slash"><?php echo lang('product_reg');?> <?php echo
                            format_currency($product->price); ?></span><br/>
                            <span class="price-sale"><?php echo lang('product_sale');?> <?php echo
                            format_currency($product->saleprice); ?></span>
                            <?php else: ?>
                            <span class="price-reg"><?php echo lang('product_price');?> <?php echo
                            format_currency($product->price); ?></span>
                            <?php endif; ?>
                            </div>
                        }
                    }
                }
            }
        }
    }
    ?>

```

```

<?php if((bool)$product->track_stock && $product->quantity < 1) { ?>
    <div
class="stock_msg"><?php echo lang('out_of_stock');?></div>
<?php } ?>
</div>

</li>
<?php }} ?>
</ul>
</div>
</div>
</div>

<script type="text/javascript">
$( '.carousel' ).carousel({
    interval: 5000
});
</script>

<?php include('footer.php'); ?>

```

## b. Vista Compuesta del Administrador

Se tiene el código de la vista del administrador, la cual está compuesta por la vista de cabecera (*header.php*) y la vista de pie de página (*footer.php*), y la vista categorías, el archivo que lo contiene es *dashboard.php*.

```

<?php include('header.php'); ?>

<?php if(!$payment_module_installed):?>
<div class="alert">
    <a class="close" data-dismiss="alert"></a>
    <strong><?php echo lang('common_note') ??:</strong> <?php echo
lang('no_payment_module_installed'); ?>
</div>
<?php endif;?>

<?php if(!$shipping_module_installed):?>
<div class="alert">
    <a class="close" data-dismiss="alert"></a>
    <strong><?php echo lang('common_note') ??:</strong> <?php echo
lang('no_shipping_module_installed'); ?>
</div>
<?php endif;?>
<div class="box">
<h2><?php echo lang('recent_orders') ??:</h2>
<table class="table table-striped">
    <thead>
        <tr>
            <th class="gc_cell_left"><?php echo lang('order_number') ??:</th>
            <th><?php echo lang('bill_to') ??:</th>
            <th><?php echo lang('ship_to') ??:</th>
            <th><?php echo lang('ordered_on') ??:</th>
            <th><?php echo lang('status') ??:</th>
            <th class="gc_cell_right"><?php echo lang('notes') ??:</th>
        </tr>
    </thead>
    <tbody>
        <?php foreach($orders as $order): ?>
            <tr>
                <td class="gc_cell_left"><a href="<?php echo site_url($this->config-
>item('admin_folder').'/orders/view/'.$order->id); ?>"><?php echo $order-
>order_number; ??:</a></td>
                <td><?php echo $order->bill_lastname.', '.$order->bill_firstname;
??:</td>
                <td><?php echo $order->ship_lastname.', '.$order->ship_firstname;
??:</td>
                <td><?php echo format_date($order->ordered_on); ??:</td>

```

```

        <td style="width:150px;">
        <?php echo $order->status ?>
        </td>
        <td class="gc_cell_right"><div class="MainTableNotes"><?php echo
        $order->notes; ?></div></td>
    </tr>
    <?php endforeach; ?>
    </tbody>
</table>

<div class="row">
    <div class="span12" >
        <a class="btn btn-primary" href="<?php echo $admin_url;?>orders"><?php
        echo lang('view_all_orders');?></a>
    </div>
</div>

<div class="box">
<h2><?php echo lang('recent_customers') ?></h2>
<table class="table table-striped">
    <thead>
    <tr>
    <?php /*<th>ID</th> uncomment this if you want it*/ ?>
    <th class="gc_cell_left"><?php echo lang('firstname') ?></th>
    <th><?php echo lang('lastname') ?></th>
    <th><?php echo lang('email') ?></th>
    <th class="gc_cell_right"><?php echo lang('active') ?></th>
    </tr>
</thead>

<tbody>
<?php foreach ($customers as $customer):?>
<tr>
<?php /*<td style="width:16px;"><?php echo $customer->id; ?></td>*/?>
    <td class="gc_cell_left"><?php echo $customer->firstname; ?></td>
    <td><?php echo $customer->lastname; ?></td>
    <td><a href="mailto:<?php echo $customer->email;?>"><?php echo
    $customer->email; ?></a></td>
    <td>
    <?php if($customer->active == 1)
    {
        echo lang('yes');
    }
    else
    {
        echo lang('no');
    }
    ?>
    </td>
</tr>
<?php endforeach; ?>
</tbody>
</table>
<div class="row">
    <div class="span12" >
    <a class="btn btn-primary" href="<?php echo $admin_url;?>customers"><?php
    echo lang('view_all_customers');?></a>
    </div>
</div>
<br/>
</div>
<?php include('footer.php');

```

#### 4.7.3.2. Visualización de Vistas del Portal Web

El diseño gráfico del portal se ha realizado manteniendo un aspecto común a través de todas las páginas de la aplicación. Todas las páginas presentan la misma estructura: una cabecera la parte más alta con un menú, una barra de navegación a la izquierda y un pie de página en la parte inferior

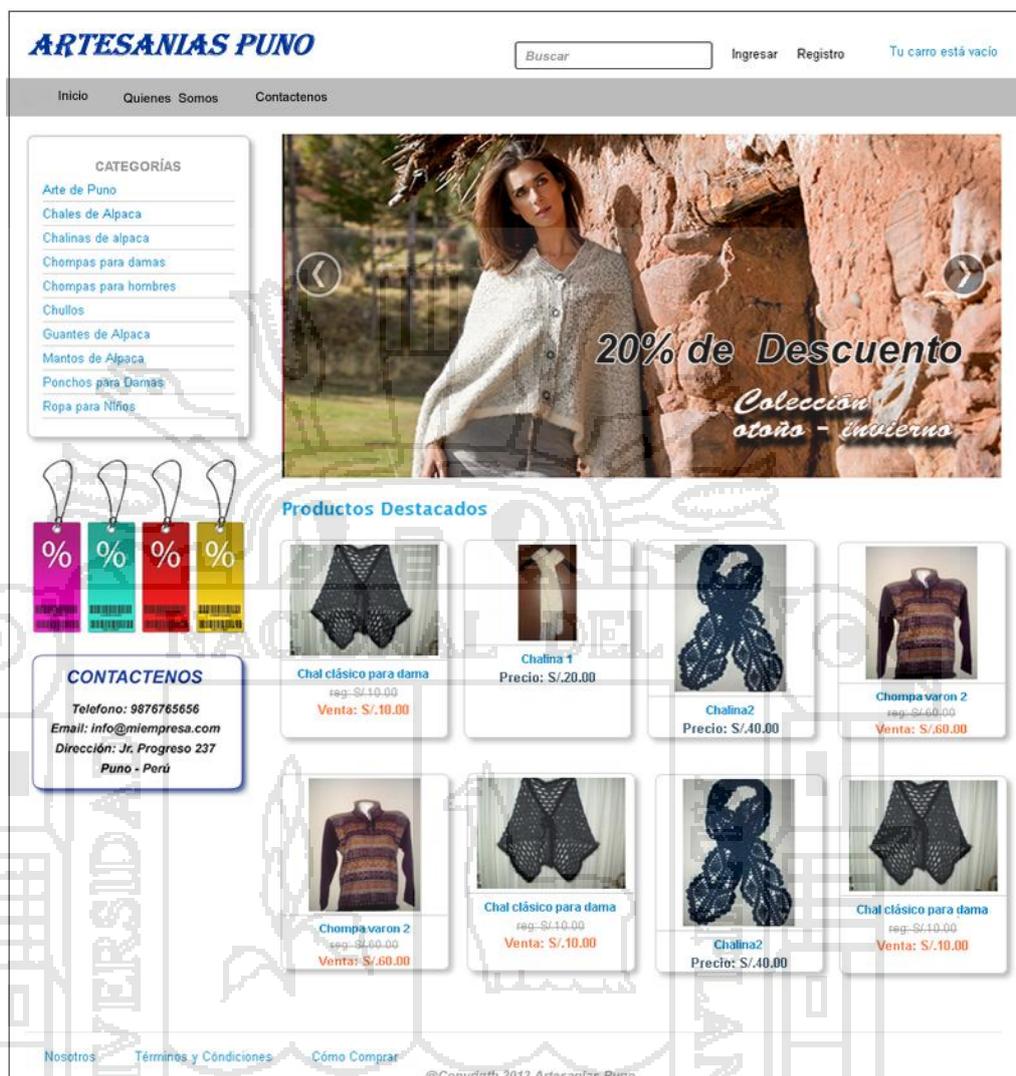
de la página. Los contenidos aparecen a la derecha de la barra de navegación entre la cabecera y el pie de página, son independientes unos de otros.

Cuando se teclea la dirección URL del Portal Web, la aplicación devuelve una página con los productos destacados de la tienda, es decir, la vitrina. De esta forma estamos atrayendo a los clientes, parecido a una tienda de una calle que intenta atraer clientes colocando productos en su vitrina. En esta primera página, se puede observar las partes comunes de que consta la interfaz gráfica.

- Se visualiza una cabecera donde se encuentra el menú principal, el logo, la herramienta de búsqueda, el menú de acceso a clientes para registro y un enlace para acceder al carro de compras.
- A la izquierda vemos una lista de categorías, el cliente de forma intuitiva comprende que puede buscar productos, navegando por las distintas secciones o categorías de productos. También se puede observar imágenes para mostrar otros datos e información.
- En la parte central se observa un banner y debajo la lista de productos destacados.
- En la parte inferior se observa un menú y pie de página.

Las siguientes figuras muestran las páginas del Portal Web. Aquí podemos ver que presentan un aspecto similar aunque el contenido es muy diferente.

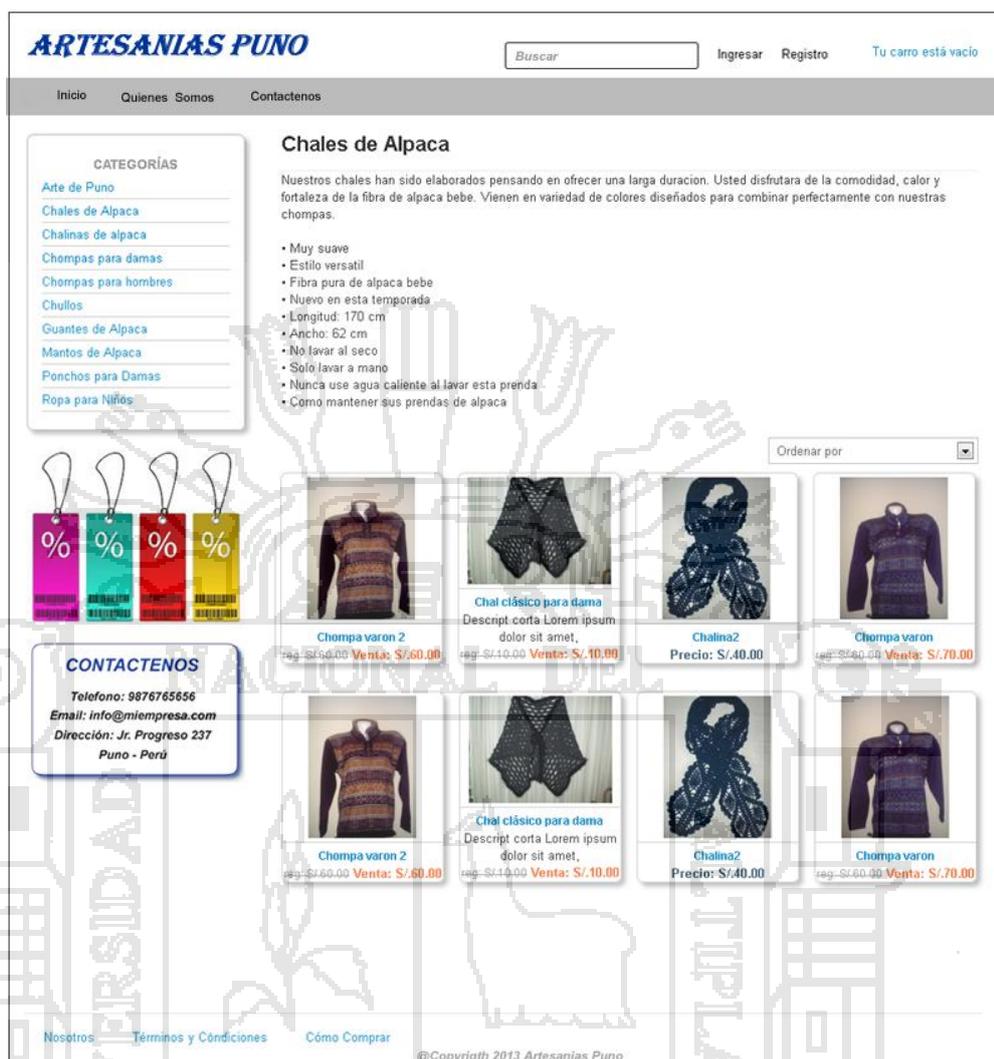
Figura N° 26  
Vista página principal de tienda virtual



Si utilizamos la opción de navegar por categorías, se muestra una página como la figura N° 26 con una lista de productos, pero con un rótulo de la categoría de productos que se ha seleccionado y su respectiva descripción; se tiene las opciones de ordenar los productos por nombre y por precios.

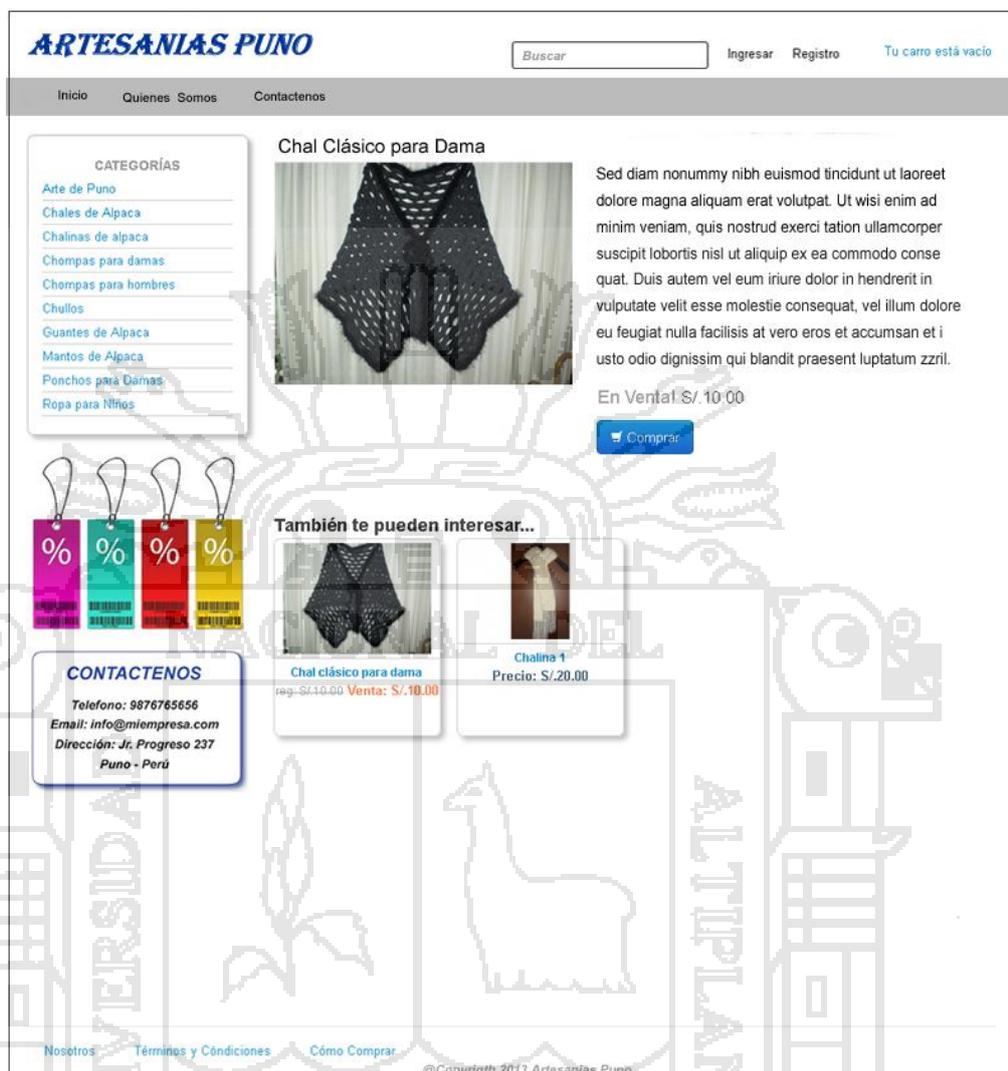
Podemos observar que la página anterior de vitrina y la de navegación por categorías comparten prácticamente, las mismas características y será posible reutilizar una parte importante de la página desarrollada.

Figura N° 27  
Vista detalles de categoría



Si seleccionamos un producto, podemos ver una página específica con la figura N° 27 donde se muestra el detalle del producto. Esta página contiene una descripción más completa de la anterior, contiene información adicional, el botón de comprar y los productos relacionados.

Figura N° 28  
Vista detalle de producto



The screenshot shows the website interface for 'ARTESANIAS PUNO'. At the top, there is a search bar with the text 'Buscar', and links for 'Ingresar', 'Registro', and 'Tu carro está vacío'. Below the header, there are navigation links: 'Inicio', 'Quienes Somos', and 'Contactenos'. The main content area features a sidebar with 'CATEGORÍAS' including 'Arte de Puno', 'Chales de Alpaca', 'Chalinas de alpaca', 'Chompas para damas', 'Chompas para hombres', 'Chullos', 'Guantes de Alpaca', 'Mantos de Alpaca', 'Ponchos para Damas', and 'Ropa para Niños'. The main product section displays 'Chal Clásico para Dama' with an image of a dark, patterned shawl. To the right of the image is a block of placeholder text: 'Sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo conse quat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril.' Below the text, it says 'En Venta! S/ 10.00' and a blue 'Comprar' button. Underneath, there are four colorful percentage tags (pink, green, red, yellow) and a section titled 'También te pueden interesar...' with two product thumbnails: 'Chal clásico para dama' (reg: S/10.00, Venta: S/10.00) and 'Chalina 1' (Precio: S/20.00). A 'CONTACTENOS' box provides contact information: 'Telefono: 9876765656', 'Email: info@miempresa.com', and 'Dirección: Jr. Progreso 237 Puno - Perú'. At the bottom, there are links for 'Nosotros', 'Términos y Condiciones', and 'Cómo Comprar', along with a copyright notice: '@Copyright 2013 Artesanias Puno'.

Una vez que hemos hallado el producto que se desea adquirir, se tiene la opción de llevar el producto a la cesta de la compra. Además, es posible que en cualquier momento deseemos ver el contenido del carro de compras. En ambas situaciones, nos aparecerá una página como la siguiente, en la que se listan los artículos de la cesta y podemos modificar la cantidad de cada uno de ellos, así como eliminar alguno o todos los artículos de nuestra cesta de la compra.

Figura N° 29  
Vista carro de compras

**ARTESANIAS PUNO**

Buscar Ingresar Registro Tienes 3 productos en tu carro

Inicio Quienes Somos Contactenos

**CATEGORÍAS**

- Arte de Puno
- Chales de Alpaca
- Chalinas de alpaca
- Chompas para damas
- Chompas para hombres
- Chullos
- Guantes de Alpaca
- Mantos de Alpaca
- Ponchos para Damas
- Ropa para Niños

**Tu carro**

Código	Nombre	Precio	Descripción	Cantidad	Totales
CHA	Chal clásico para dama	S/ 10.00	Descript corta Lorem ipsum dolor sit amet,	1	S/ 10.00
	Chalina2	S/ 40.00		1	S/ 40.00
	Chompa varon 2	S/ 60.00		1	S/ 60.00
Subtotal					S/ 110.00
<b>Total General</b>					<b>S/ 110.00</b>

Si tienes un cupon, ingresa el código aquí:

Aplicar cupon Ingresar Registrarse Ahora Actualizar el carro Pagar

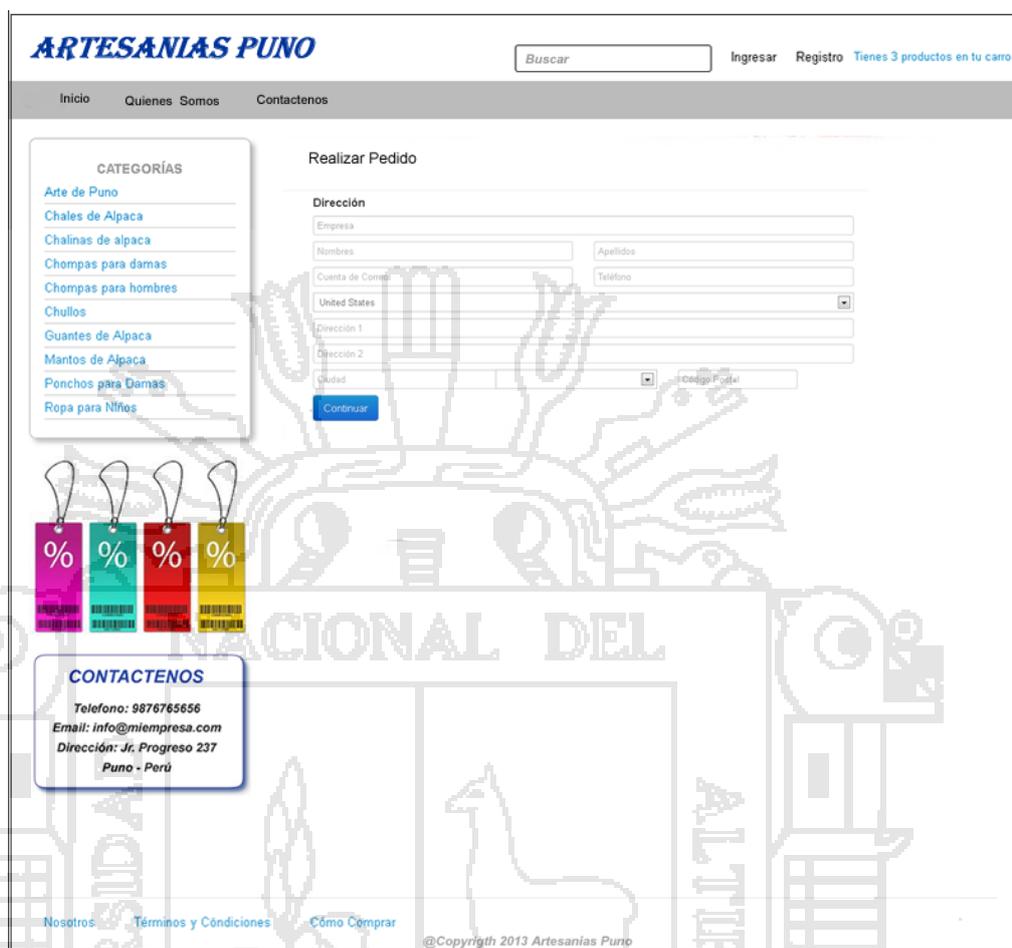
**CONTACTENOS**

Telefono: 9876765656  
Email: info@miempresa.com  
Dirección: Jr. Progreso 237  
Puno - Perú

Nosotros Términos y Condiciones Cómo Comprar @Copyright 2013 Artesanias Puno

Quando tenemos en el carro de compras todos los productos que se desea comprar, se puede utilizar la opción de realizar el pedido. Esta opción nos muestra una página que contiene un formulario de pedido para que el cliente proporcione información acerca del envío de los productos y de la forma de pago, según la figura N° 29.

Figura N° 30  
Vista realizar pedido



**ARTESANIAS PUNO**  [Ingresar](#) [Registro](#) [Tienes 3 productos en tu carro](#)

[Inicio](#) [Quienes Somos](#) [Contactenos](#)

**CATEGORÍAS**

- Arte de Puno
- Chales de Alpaca
- Chalinas de alpaca
- Chompas para damas
- Chompas para hombres
- Chullos
- Guantes de Alpaca
- Mantos de Alpaca
- Ponchos para Damas
- Ropa para Niños

**Realizar Pedido**

**Dirección**

Empresa

Nombres  Apellidos

Cuenta de Corriente  Teléfono

United States

Dirección 1

Dirección 2

Ciudad

**CONTACTENOS**

Telefono: 9876765656  
Email: info@miempresa.com  
Dirección: Jr. Progreso 237  
Puno - Perú

[Nosotros](#) [Términos y Condiciones](#) [Cómo Comprar](#)

@Copyright 2013 Artesanias Puno

Si llenamos correctamente los datos del formulario de pedido, la aplicación nos devolverá un mensaje de confirmación. Si falta algún dato, aparecerá un mensaje de error.

#### 4.7.3.3. Vistas del Administrador

Cuando tecleamos la dirección URL del portal Web, e incluimos la subcadena “/administrador/” en esta URL, la aplicación nos devuelve un formulario de acceso a la sección privada de la aplicación.

La siguiente imagen muestra el formulario de acceso de la aplicación de administración.

Figura N° 31  
Vista acceso al administrador



**CMS E-commerce**

Acceso al Administrador

Email

Contraseña

Recordarme

Acceder

En esta página debemos introducir una cuenta de correo y una contraseña de un usuario registrado cuando se hizo la instalación del Sistema. Cuando proporcionamos los datos que cumplen las condiciones anteriores, la aplicación nos deja acceder a la Administración devolviéndonos la siguiente página de bienvenida.

Figura N° 32  
Vista inicial de administración del sistema



**CMS E-commerce** Clientes ▾ Ventas ▾ Catálogo ▾ Contenido ▾ Administración ▾ Salir de Administrador

**Administrador**

**Pedidos Recientes**

Nro de Orden	Facturar en	Enviar a	Fecha de Pedido	Estado
13732549001	Guillen, Lisbeth	Guillen, Lisbeth	July 08 03:31:40, 2013	Pendiente

[Ver Todos los Pedidos](#)

**Clientes Recientes**

Nombres	Apellidos	Correo Electrónico	Activo
Lisbeth	Guillen	lisguillen1@gmail.com	Si

[Ver Todos los Clientes](#)

Tiene las siguientes opciones de menú:

### a. Administrar Clientes

- **Clientes:** Esta opción permite añadir, modificar, eliminar y agregar direcciones de los clientes. Mediante esta opción de administración, se modifica la tabla clientes de la base de datos.

Figura N° 33  
Vista administración de clientes



Apellidos	Nombres	Cuenta de Correo	Activo
Guillen	Lisbeth	lisguillen1@gmail.com	Yes

- **Grupos de Clientes:** Esta opción permite añadir, modificar y eliminar grupos de clientes. Mediante esta opción de administración, se modifica la tabla grupos de clientes de la base de datos.

Figura N° 34  
Vista administración de grupos de clientes



Nombre de Grupo	Descuento	Tipo de Descuento
Shoppers	0	fixed

### b. Administrar Ventas

- **Pedidos:** Esta opción permite administrar los pedidos, ver datos completos, cambiar estado de pedido y enviar mensajes al cliente. Mediante esta opción de administración, se modifica la tabla pedidos de la base de datos.

Figura N° 35  
Vista administración de pedidos



- **Reportes:** Esta opción nos permite ver reportes principales sobre las ventas, productos vendidos y más vendidos por fechas.

Figura N° 36  
Vista reportes



- **Cupones:** Esta opción permite añadir, modificar y eliminar cupones de descuentos. Mediante esta opción de administración, se modifica la tabla cupones de la base de datos.

Figura N° 37  
Vista administración de cupones



**c. Administrar Catálogo**

- **Categorías:** Esta opción permite añadir, modificar, organizar y eliminar categorías de la tienda virtual. Mediante esta opción de administración, se modifica la tabla categorías de la base de datos.

Figura N° 38  
Vista administración de categorías de productos

ID	Categoría	Editar	Organizar	Eliminar
10	Arte de Puno			
1	Chales de Alpaca			
2	Chalinas de alpaca			
3	Chompas para damas			
4	Chompas para hombres			
5	Chullos			
6	Guantes de Alpaca			
7	Mantos de Alpaca			
8	Ponchos para Damas			
9	Ropa para Niños			

- **Productos:** Esta opción permite añadir, modificar, y eliminar productos, permite activar o desactivar el productos, seleccionar como producto destacado, además esta opción muestra un formulario que permite añadir un nuevo producto en la tabla de Productos de la base de datos.

Figura N° 39  
Vista administración de productos

Código	Nombre	Precio	Precio de Venta	Cantidad Disponible	Activado	Editar	Copiar	Eliminar
CHA	Chal clásico para dama	10.00	10.00	8	Activado			
CHAL1	Chalina 1	20.00	0.00	11	Activado			
	Chalina2	40.00	0.00	30	Activado			
	Chompa varon	60.00	70.00	4	Activado			
	Chompa varon 2	60.00	60.00	8	Activado			

**d. Administrar Contenido**

- **Banners:** Esta opción permite añadir, modificar, ordenar y eliminar banners de la tienda virtual. Mediante esta opción de administración, se modifica la tabla banners de la base de datos.

Figura N° 40  
Vista administración de banner



- **Cajas:** Esta opción permite añadir, modificar, organizar y eliminar cajas de la tienda virtual. Mediante esta opción de administración, se modifica la tabla cajas de la base de datos.

Figura N° 41  
Vista administración de cajas



- **Páginas:** Esta opción permite añadir, modificar, organizar y eliminar páginas de la tienda virtual. Mediante esta opción de administración, se modifica la tabla páginas de la base de datos.

Figura N° 42  
Vista administración de páginas de contenido



#### e. Administración de Detalles de Tienda

- **Administradores:** Esta opción permite añadir, modificar y eliminar administradores. Mediante esta opción de administración, se modifica la tabla administradores de la base de datos.

Figura N° 43  
Vista administración de administradores



#### 4.7.4. Desarrollo del Controlador

El controlador, en la arquitectura MVC, controla el flujo de la aplicación y sirve como cola de unión entre el modelo y las vistas, ya que ejecuta lógica de negocio en el modelo en respuesta a peticiones de usuario y, a continuación, selecciona la siguiente vista a mostrar. El controlador desacopla la presentación de datos de la lógica de negocio y de datos. Los controladores forman las Url del sistema.

Como ejemplo detallamos el controlador de la página clientes del Portal Web:

```
<?php
//Clase clientes
class Customers extends Admin_Controller {
    var $customer_id = false;
    function __construct()
    {
        parent::__construct();

        $this->load->model(array('Customer_model', 'Location_model'));
        $this->load->helper('formatting_helper');
        $this->lang->load('customer');
    }
    //Función de inicio del controlador
    function index($field='lastname', $by='ASC', $page=0)
    {
        $data['page_title'] = lang('customers');
        $data['customers'] = $this->Customer_model->get_customers(50,$page, $field,
        $by);

        $this->load->library('pagination');

        $config['base_url'] = base_url().'/'.$this->config-
        >item('admin_folder').'customers/index/'.$field.'/'.$by.'/';
        $config['total_rows'] = $this->Customer_model->count_customers();
        $config['per_page'] = 50;
        $config['uri_segment'] = 6;
        $config['first_link'] = 'First';
        $config['first_tag_open'] = '<li>';
        $config['first_tag_close'] = '</li>';
        $config['last_link'] = 'Last';
        $config['last_tag_open'] = '<li>';
        $config['last_tag_close'] = '</li>';

        $config['full_tag_open'] = '<div class="pagination"><ul>';
        $config['full_tag_close'] = '</ul></div>';
        $config['cur_tag_open'] = '<li class="active"><a href="#">';
        $config['cur_tag_close'] = '</a></li>';
```

```

$config['num_tag_open']           = '<li>';
$config['num_tag_close']         = '</li>';
$config['prev_link']             = '&laquo;';
$config['prev_tag_open']         = '<li>';
$config['prev_tag_close']        = '</li>';
$config['next_link']             = '&raquo;';
$config['next_tag_open']         = '<li>';
$config['next_tag_close']        = '</li>';

$this->pagination->initialize($config);

$data['page']                    = $page;
$data['field']                   = $field;
$data['by']                      = $by;

$this->load->view($this->config->item('admin_folder').'customers', $data);
}

//Función para formulario de registro de clientes
function form($id = false)
{
    force_ssl();
    $this->load->helper('form');
    $this->load->library('form_validation');

    $data['page_title']           = lang('customer_form');

    //valores por defecto si el cliente es nuevo
    $data['id']                   = "";
    $data['group_id']             = "";
    $data['firstname']            = "";
    $data['lastname']            = "";
    $data['email']                = "";
    $data['phone']                = "";
    $data['company']              = "";
    $data['email_subscribe']      = "";
    $data['active']               = false;

    // unirse a un grupo de clientes
    $groups = $this->Customer_model->get_groups();
    foreach($groups as $group)
    {
        $group_list[$group->id] = $group->name;
    }
    $data['group_list'] = $group_list;

    $this->form_validation->set_rules('firstname', 'lang:firstname',
'trim|required|max_length[32]');
    $this->form_validation->set_rules('lastname', 'lang:lastname',
'trim|required|max_length[32]');
    $this->form_validation->set_rules('email', 'lang:email',
'trim|required|valid_email|max_length[128]|callback_check_email');
    $this->form_validation->set_rules('phone', 'lang:phone',
'trim|required|max_length[32]');
    $this->form_validation->set_rules('company', 'lang:company', 'trim|max_length[128]');
    $this->form_validation->set_rules('active', 'lang:active');
    $this->form_validation->set_rules('group_id', 'group_id', 'numeric');
    $this->form_validation->set_rules('email_subscribe', 'email_subscribe',
'numeric|max_length[1]');

    if ($this->form_validation->run() == FALSE)
    {
        $this->load->view($this->config->item('admin_folder').'customer_form',
$data);
    }
    else
    {
        $save['id']                = $id;
        $save['group_id']          = $this->input->post('group_id');
        $save['firstname']         = $this->input->post('firstname');
        $save['lastname']          = $this->input->post('lastname');
        $save['email']              = $this->input->post('email');
        $save['phone']              = $this->input->post('phone');
        $save['company']           = $this->input->post('company');
    }
}

```

```

        $save['active'] = $this->input->post('active');
        $save['email_subscribe'] = $this->input->post('email_subscribe');

        if ($this->input->post('password') != "" || !$id)
        {
            $save['password'] = $this->input->post('password');
        }

        $this->Customer_model->save($save);

        $this->session->set_flashdata('message', lang('message_saved_customer'));

        //regresar a la lista de clientes
        redirect($this->config->item('admin_folder').'/customers');
    }
}

//Función eliminar cliente
function delete($id = false)
{
    if ($id)
    {
        $customer = $this->Customer_model->get_customer($id);
        if (!$customer)
        {
            $this->session->set_flashdata('error', lang('error_not_found'));
            redirect($this->config->item('admin_folder').'/customers');
        }
        else
        {
            $delete = $this->Customer_model->delete($id);
            $this->session->set_flashdata('message', lang('message_customer_deleted'));
            redirect($this->config->item('admin_folder').'/customers');
        }
    }
    else
    {
        $this->session->set_flashdata('error', lang('error_not_found'));
        redirect($this->config->item('admin_folder').'/customers');
    }
}

//Función para ordenar lista de clientes
function order_list($status = false)
{
    $this->load->model('Order_model');
    $data['page_title'] = 'Order List';
    $data['orders'] = $this->Order_model->get_orders($status);
    $this->load->view($this->config->item('admin_folder').'/order_list', $data);
}

// Función grupos de clientes
function groups()
{
    $data['groups'] = $this->Customer_model->get_groups();
    $data['page_title'] = lang('customer_groups');

    $this->load->view($this->config->item('admin_folder').'/customer_groups', $data);
}

//Función editar grupos de clientes
function edit_group($id=0)
{
    $this->load->helper('form');
    $this->load->library('form_validation');

    $data['page_title'] = lang('customer_group_form');
    $data['id'] = "";
    $data['name'] = "";
    $data['discount'] = "";
    $data['discount_type'] = "";

    if($id)
    {

```

```

        $group = $this->Customer_model->get_group($id);

        $data['id']                = $group->id;
        $data['name']              = $group->name;
        $data['discount']          = $group->discount;
        $data['discount_type']     = $group->discount_type;
    }

    $this->form_validation->set_rules('name',                'lang:group_name',
'trim|required|max_length[50]');
    $this->form_validation->set_rules('discount', 'lang:discount', 'trim|required|numeric');
    $this->form_validation->set_rules('discount_type',     'lang:discount_type',
'trim|required');

    if ($this->form_validation->run() == FALSE)
    {
        $this->load->view($this->config->item('admin_folder').'/customer_group_form',
$data);
    }
    else
    {
        if($id)
        {
            $save['id'] = $id;

            $save['name']                = set_value('name');
            $save['discount']            = set_value('discount');
            $save['discount_type']       = set_value('discount_type');

            $this->Customer_model->save_group($save);
            $this->session->set_flashdata('message', lang('message_saved_group'));

            //go back to the customer group list
            redirect($this->config->item('admin_folder').'/customers/groups');
        }
    }

    //Función eliminar grupo
    function delete_group($id)
    {
        if(empty($id))
        {
            return;
        }

        $this->Customer_model->delete_group($id);

        //go back to the customer list
        redirect($this->config->item('admin_folder').'/customers/groups');
    }

    //Función lista de direcciones de clientes
    function address_list($customer_id)
    {
        $data['address_list'] = $this->Customer_model->get_address_list($customer_id);

        $this->load->view($this->config->item('admin_folder').'/address_list', $data);
    }

    // Función para formulario de direcciones
    function address_form($customer_id, $id = false)
    {
        $data['id']                = $id;
        $data['company']           = "";
        $data['firstname']         = "";
        $data['lastname']          = "";
        $data['email']             = "";
        $data['phone']             = "";
        $data['address1']          = "";
        $data['address2']          = "";
        $data['city']              = "";
        $data['country_id']        = "";
        $data['zone_id']           = "";
        $data['zip']               = "";

        $data['customer_id']       = $customer_id;
    }

```



```

$data['page_title']          = lang('address_form');
//get the countries list for the dropdown
$data['countries_menu']     = $this->Location_model->get_countries_menu();

$this->load->library('form_validation');
$this->form_validation->set_rules('company', 'lang:company', 'trim|max_length[128]');
$this->form_validation->set_rules('firstname', 'lang:firstname',
'trim|required|max_length[32]');
$this->form_validation->set_rules('lastname', 'lang:lastname',
'trim|required|max_length[32]');
$this->form_validation->set_rules('email', 'lang:email',
'trim|required|valid_email|max_length[128]');
$this->form_validation->set_rules('phone', 'lang:phone',
'trim|required|max_length[32]');
$this->form_validation->set_rules('address1', 'lang:address',
'trim|required|max_length[128]');
$this->form_validation->set_rules('address2', 'lang:address', 'trim|max_length[128]');
$this->form_validation->set_rules('city', 'lang:city', 'trim|required');
$this->form_validation->set_rules('country_id', 'lang:country', 'trim|required');
$this->form_validation->set_rules('zone_id', 'lang:state', 'trim|required');
$this->form_validation->set_rules('zip', 'lang:postcode',
'trim|required|max_length[32]');

if ($this->form_validation->run() == FALSE)
{
    $this->load->view($this->config-
>item('admin_folder').'/customer_address_form', $data);
}
else
{
    $a['customer_id']          = $customer_id; // this is
needed for new records
    $a['id']                  =
(empty($id))?"":$id;
    $a['field_data']['company'] = $this->input->post('company');
    $a['field_data']['firstname'] = $this->input->post('firstname');
    $a['field_data']['lastname'] = $this->input->post('lastname');
    $a['field_data']['email'] = $this->input->post('email');
    $a['field_data']['phone'] = $this->input->post('phone');
    $a['field_data']['address1'] = $this->input->post('address1');
    $a['field_data']['address2'] = $this->input->post('address2');
    $a['field_data']['city'] = $this->input->post('city');
    $a['field_data']['zip'] = $this->input->post('zip');
}

//Función eliminar direcciones
function delete_address($customer_id = false, $id = false)
{
    if ($id)
    {
        $address = $this->Customer_model->get_address($id);
        if (!$address)
        {
            $this->session->set_flashdata('error', lang('error_address_not_found'));
            if($customer_id)
            {
                redirect($this->config->item('admin_folder').'/customers/addresses/'.$customer_id);
            }
            else
            {
                redirect($this->config->item('admin_folder').'/customers');
            }
        }
        if($customer_id)
        {
            redirect($this->config->item('admin_folder').'/customers/addresses/'.$customer_id);
        }
        else
        {
            redirect($this->config->item('admin_folder').'/customers');
        }
    }
}
}

```

}

#### **4.8. VALIDACIÓN DEL PROTOTIPO DE SISTEMA DE ADMINISTRACIÓN DE CONTENIDOS**

##### **4.8.1. Pruebas de aceptación**

Cabe señalar que el diseño de pruebas se realiza para todas las partes del sistema como una práctica para garantizar el buen funcionamiento.

Se realizaron las siguientes pruebas de aceptación de acuerdo a las historias de:

- a. Pruebas de acceso al administrador.
- b. Pruebas de administración de pedidos.
- c. Pruebas de administración de clientes.
- d. Pruebas de administración de categorías de productos.
- e. Pruebas de administración de productos.
- f. Pruebas de administración de cupones de descuentos
- g. Pruebas de Administración de cupones de descuentos
- h. Pruebas de configuración de contenidos de tienda.
- i. Pruebas de administración de usuarios del sistema (administradores).
- j. Pruebas de administración de ubicaciones geográficas.
- k. Pruebas de configuración de medios de pagos.
- l. Pruebas de búsqueda de productos en el catálogo.
- m. Pruebas de consulta de catálogo.
- n. Pruebas de agregar producto al carro de compras.
- o. Pruebas de enviar pedido.
- p. Pruebas de consulta de pedidos.
- q. Pruebas de registro de cliente.
- r. Pruebas de acceso de cliente.
- s. Pruebas de navegación por la tienda.

En cada una de las pruebas se obtuvo un resultado satisfactorio.

##### **4.8.2. Evaluación y Aceptación de la Aplicación por los Usuarios**

La evaluación y validación del prototipo de sistema de administración de contenidos, se sondeó en base a una encuesta (Ver Anexo B) a la muestra que fueron un total de 34. Los resultados fueron los siguientes:

#### 4.8.2.1. Del Diseño

##### ¿Cómo califica el diseño de interfaz del sistema?

Cuadro N° 16  
Resultado de opinión con respecto al diseño de interfaz

Items	Usuarios	Porcentaje
Excelente	15	44.12%
Bueno	14	41.18%
Regular	5	14.70%
Malo	0	0%
Total	34	100%

Fuente: Elaboración propia

**Interpretación:** De los resultados a la pregunta efectuada se, resuelve que el 44.12% de los encuestados opina que la interfaz es excelente, el 41.18 % que es bueno y un 5% que es regular. Por lo tanto se concluye que la interfaz es agradable al usuario.

##### ¿Qué tan fácil le parece el uso o manejo del sistema?

Cuadro N° 17  
Resultado de opinión con respecto al manejo del sistema

Items	Usuarios	Porcentaje
Fácil	31	91.18%
Regular	3	8.82%
Difícil	0	0%
Total	34	100%

Fuente: Elaboración propia

**Interpretación:** De los resultados a la pregunta efectuada, se resuelve que el 91% de los encuestados opina que el manejo del sistema es fácil y un 8.82% es de manejo regular. Por lo tanto se concluye que el manejo del sistema es fácil para la mayoría de usuarios.

#### 4.8.2.2. De la Implementación del Sistema

¿Cuál es su opinión en cuanto a la rapidez del sistema?

Cuadro N° 18  
Resultado de opinión con respecto a la rapidez del sistema

Items	Usuarios	Porcentaje
Exelente	30	88.24%
Bueno	4	11.76%
Regular	0	0%
Malo	0	0%
Total	34	100%

Fuente: Elaboración propia

**Interpretación:** De los resultados a la pregunta efectuada, se resuelve que para un 88.24% de los encuestados, la rapidez del sistema es exelente y para un 11.76% es bueno. Por lo tanto se concluye que para la mayoría de usuarios los procesos en el sistema se realizan de manera rápida.

#### 4.8.2.3. De la Implementación del Portal de E-commerce

La implementación de la tienda virtual, consiste en la instalación, configuración de la tienda virtual e ingresar información (texto, imágenes, atributos) por el administrador.

¿Cómo considera Ud. el aprendizaje del sistema en cuanto a la implementación?

Cuadro N° 19  
Resultado de opinión con respecto a la implementación del portal de E-commerce

Items	Usuarios	Porcentaje
Fácil de aprender	28	82.35%
Moderadamente complejo de aprender	6	17.65%
Difícil de aprender	0	0%
Total	34	100%

Fuente: Elaboración propia

**Interpretación:** De los resultados a la pregunta efectuada, se resuelve que para un 82.35% de los encuestados el sistema es fácil de aprender para realizar su implementación y para un 17.65% es moderadamente complejo. Por lo tanto se concluye que para la mayoría de usuarios es fácil el aprendizaje para la implementación de la tienda virtual.

### ¿Cuál es su opinión con respecto al grado de dificultad en cuanto a la implementación de la tienda virtual?

Cuadro N° 20  
Resultado de opinión con respecto a la dificultad en la implementación del portal de E-commerce

Ítems	Usuarios	Porcentaje
Fácil	30	88.24%
Regular	4	11.76%
Difícil	0	0%
Total	34	100%

Fuente: Elaboración propia

**Interpretación:** De los resultados a la pregunta efectuada, se resuelve que para un 88.24% de los encuestados es fácil la implementación de una tienda virtual y para un 11.76% es regular. Por lo tanto se concluye que para la mayoría de usuarios es fácil implementar una tienda virtual con el uso del sistema.

### Estimación del Tiempo de Implementación

Para calcular el tiempo de implementación de un portal de E-commerce hizo uso de la observación directa de los cuales se tiene el siguiente resultado.

Cuadro N° 21  
Estimación del Tiempo de Implementación del Sistema

Horas	N° de participantes
0 – 8	25
8 – 16	7
16 – 24	2
24 – mas	0
Total	34

Fuente: Elaboración propia

**Interpretación:** De acuerdo a los resultados se observa que 25 encuestados ha implementado una tienda virtual en un promedio de 8 horas, 7 de 8 a 16 horas y 2 de 16 a 24 horas. Por lo tanto se concluye que se puede implementar una tienda virtual en 8 horas.

#### 4.8.2.4. Del Mantenimiento (actualización) del Portal de E-commerce

**¿Cómo considera Ud. la actualización de los contenidos del Portal de E-Commerce con el sistema administrador de contenidos?**

Cuadro N° 22  
Resultado de opinión con respecto al mantenimiento del portal de E-commerce

Items	Usuarios	Porcentaje
Fácil	29	85.29%
Regular	5	14.71%
Difícil	0	0%
Total	34	100%

Fuente: Elaboración propia

**Interpretación:** De los resultados a la pregunta efectuada, se resuelve que para un 85.29% de los encuestados es fácil realizar el mantenimiento (actualización) del portal de E-commerce y para un 14.71% es regular. Por lo tanto se concluye que para la mayoría de usuarios es fácil realizar el mantenimiento del portal de E-commerce con el uso del sistema administrador de contenidos.

**Sobre la autodescripción del sistema, usted cree que es:**

Cuadro N° 23  
Resultado de opinión con respecto a la autodescripción del sistema

Items	Usuarios	Porcentaje
Autodescriptible	32	94.12%
No autodescriptible	2	5.88%
Total	34	100%

Fuente: Elaboración propia

**Interpretación:** De los resultados a la pregunta efectuada, se resuelve que para una 94% de los encuestados el sistema de administración de contenidos es autodescriptible y para un 5.88% no lo es. Por lo tanto se concluye que para

la mayoría de usuarios el sistema de administración de contenidos es autodescriptible.

#### 4.8.2.5. De la aceptación del usuario sobre el sistema

#### ¿El sistema propuesto, le facilita el mantenimiento y la implementación de un portal de E-commerce?

Cuadro N° 24  
Resultado de opinión con respecto a la aceptación del sistema

Items	Usuarios	Porcentaje
Si	30	88.24%
No	4	11.76%
Total	34	100%

Fuente: Elaboración propia

**Interpretación:** De los resultados a la pregunta efectuada se resuelve que un 88.24% tiene aceptación del sistema y un 11.76% no. Por lo tanto se concluye que el sistema administrador de contenidos si facilita el mantenimiento e implementación de portales web para los usuarios.

#### 4.8.3. Métricas de Software

##### 4.8.3.1. Usabilidad

Tratándose de una aplicación de tipo visual, el usuario no tiene problemas en operar el sistema, tan solamente necesita tener referencias de Windows u Microsoft Office. En lo referente a la aplicación Web sucede de igual forma el usuario se guía por el sentido común ya que la interfaz que se le presenta es muy agradable.

##### 4.8.3.2. Mantenibilidad

La facilidad de mantenimiento lo da el framework CodeIgniter, ya que objetivo es permitirle desarrollar proyectos mucho más rápido de lo que podría si lo escribiese desde cero, proveyéndole un rico juego de librerías para tareas comúnmente necesarias, así como una interface simple y estructura lógica para acceder a esas librerías.

Por otro lado el patrón Modelo Vista Controlador trata de realizar un diseño que desacople la vista del modelo, con la finalidad de mejorar la reusabilidad. De esta forma las modificaciones en las vistas impactan en menor medida en la lógica de negocio o de datos.

#### 4.8.3.3. Portabilidad

Es una métrica de software que se refiere al esfuerzo requerido para transferir el programa desde un hardware y/o un entorno de sistemas de software a otro.

En cuanto a la portabilidad, se ha elegido herramientas que faciliten que el sistema sea independiente del hardware y software, al respecto Php es un lenguaje de programación multiplataforma, lo que significa que está preparado para trabajar sobre distintos sistemas operativos<sup>7</sup>.

El gestor de Base de Datos MySQL es soportada por 15 plataformas (Oracle Linux, Oracle Solaris, Suse, Debian, Microsoft, etc), esta información ha sido extraída del portal de MySQL<sup>8</sup>.

El servidor web Apache es soportado por 10 sistemas operativos (Unix, FreeBSD, Solaris, Novell, etc.), esta información ha sido extraída del portal de Apache<sup>9</sup>.

#### 4.8.4. Prueba de Hipótesis

Para el trabajo de investigación "Prototipo de Sistema de Administración de Contenidos para la Implementación y Mantenimiento de Portales Web Orientados al E-Commerce", se supuso que para un 80% de usuarios el prototipo de sistema de administración de contenidos le facilita la implementación y el mantenimiento de portales web. Es así que en base a la muestra y a las encuestas realizadas a los clientes consumidores se obtuvo que 30 usuarios de los 34 aceptan esta aplicación:

Para realizar la siguiente prueba de hipótesis, la muestra tiene que cumplir lo siguiente:

<sup>7</sup> <http://www.php.net> . [Acceso 01 de mayo 2013]

<sup>8</sup> <http://www.mysql.com/support/supportedplatforms/database.html>. [Acceso 01 de mayo 2013]

<sup>9</sup> <http://www.apache.org> . [Acceso 01 de mayo 2013]

$$np \geq 5$$

$$nq \geq 5$$

$$np = 34(0.80) = 27.2$$

$$nq = 34(0.20) = 6.8$$

Los resultados son mayores que 5 y la muestra es mayor que 30, entonces es posible realizar la prueba utilizando la distribución normal.

t. Establecemos la hipótesis:

**H<sub>0</sub>:**  $P \geq 0.80$ ; El prototipo de sistema de administración de contenidos facilita la implementación y mantenimiento de portales web orientados al e-commerce.

**H<sub>1</sub>:**  $P < 0.80$ ; El prototipo de sistema de administración de contenidos no facilita la implementación y mantenimiento de portales web orientados al e-commerce.

Nivel de significancia:  $\alpha = 0.01$

u. Estadística:

Calculamos el error estándar:

$$\sigma_{\bar{P}} = ((P_{ho} q_{ho})/n)^{1/2}$$

$$\sigma_{\bar{P}} = (((0.80)(0.20))/34)^{1/2}$$

$$\sigma_{\bar{P}} = 0.0685$$

Calculamos la proporción de muestra:

$$\bar{P} = 30/34 = 0.882$$

Estandarizamos la proporción de la muestra mediante la siguiente fórmula:

$$Z = \frac{\bar{P} - P_{ho}}{\sigma_{\bar{P}}}$$

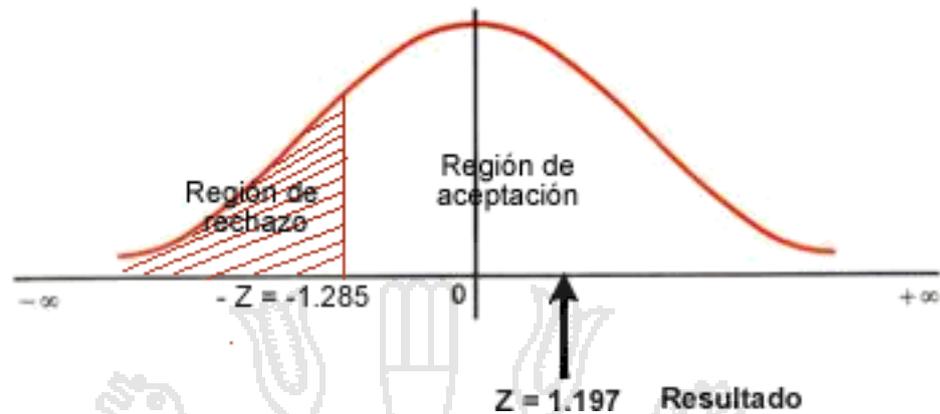
$$Z = \frac{0.882 - 0.80}{0.0685}$$

$$Z = 1.197$$

v. Región crítica:

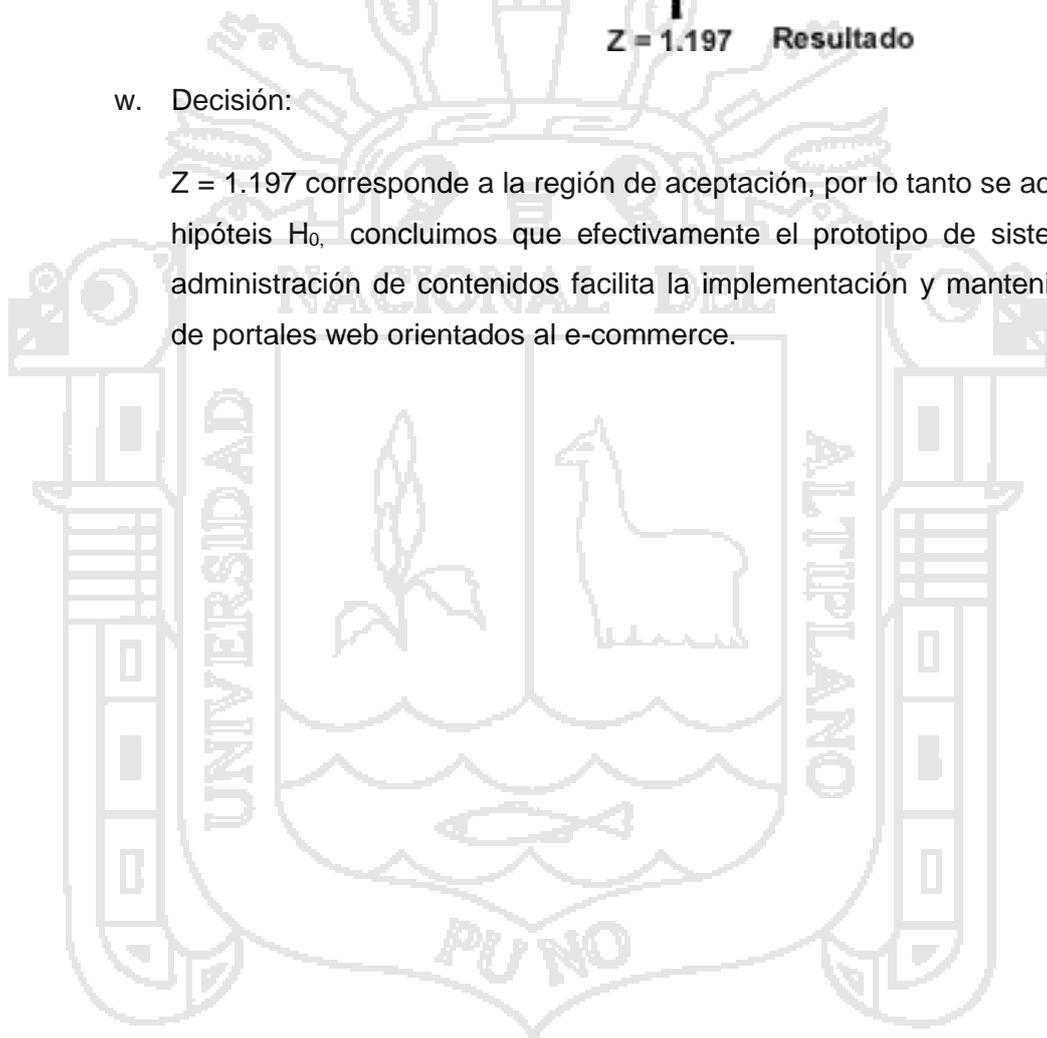
$$R.C. = ] -\infty, -1.285[$$

Figura N° 44  
Curva para prueba de hipótesis



w. Decisión:

$Z = 1.197$  corresponde a la región de aceptación, por lo tanto se acepta la hipótesis  $H_0$ , concluimos que efectivamente el prototipo de sistema de administración de contenidos facilita la implementación y mantenimiento de portales web orientados al e-commerce.



## CAPÍTULO V

### CONCLUSIONES

---

**PRIMERA:** Ha sido factible desarrollar el prototipo de sistema de administración de contenidos para la implementación y mantenimiento de portales web orientados al e-commerce, para lo cual se ha utilizado la metodología de desarrollo de software programación extrema y para su implementación el framework Codeigniter, las múltiples fases de desarrollo se muestran en la Capítulo 4 de la presente Tesis, lo cual también se confirma con el 80% de aceptación por parte de los usuarios, según la prueba estadística de hipótesis.

**SEGUNDA:** Todos los requerimiento identificados en las historias de usuario han sido desarrollados mostrando resultados satisfactorios en la fase de pruebas (pruebas de aceptación), los requerimientos funcionales en total son 19 y todos han sido desarrollados y entregados al usuario satisfactoriamente.

**TERCERA:** Según las encuestas realizadas sobre manejabilidad, estética y usabilidad el mayor porcentaje de usuarios quedó satisfecho.

**CUARTA:** En la implementación del sistema, las herramientas de software utilizadas como el framework codeigniter se han adecuado a las exigencias del sistema, sin presentar obstáculo durante su desarrollo.

En cuanto a la portabilidad, desde el análisis de requerimientos no funcionales se ha elegido herramientas que faciliten que el sistema sea independiente del hardware y software, al respecto Php es un lenguaje de programación multiplataforma.

El gestor de Base de Datos MySQL es soportada por 15 plataformas (Oracle Linux, Oracle Solaris, Suse, Debian, Microsoft, etc), esta información ha sido extraída del portal de MySQL<sup>10</sup>.

El servidor web Apache es soportado por 10 sistemas operativos (Unix, FreeBSD, Solaris, Novell, etc.), esta información ha sido extraída del portal de Apache<sup>11</sup>.

---

<sup>10</sup> <http://www.mysql.com/support/supportedplatforms/database.html>. [Acceso 01 de mayo 2013]

<sup>11</sup> <http://www.apache.org> . [Acceso 01 de mayo 2013]

**QUINTA:** Según las encuestas realizadas para la validación, se pudo observar y conocer la satisfacción de los usuarios con el sistema propuesto, por lo tanto les facilita la implementación y el mantenimiento de portales Web.



## CAPÍTULO VI

### RECOMENDACIONES

---

**PRIMERA:** Propiciar en las empresas y Pymes de nuestra región el uso de nuevas tecnologías, ya que muchas empresas en la capital han logrado ser más exitosas gracias a su uso.

**SEGUNDA:** Se recomienda a los profesionales de nuestra región el desarrollo de proyectos con el uso de las nuevas tecnologías, el desarrollo de aplicaciones web. Ya que la nueva tecnología es importante para el desarrollo del país.

**TERCERA:** Se recomienda utilizar las metodologías ágiles como la programación extrema para el desarrollo, modelación y elaboración de los sistemas de procesamiento de transacciones, sistemas de información y aplicaciones Web dinámicos donde se considera al usuario del sistema como elemento importante en el desarrollo del software, ya que cumpliendo sus requerimientos, se llega a desarrollar un sistema exitoso que satisfaga sus necesidades.

**CUARTA:** Se recomienda usar el patrón de desarrollo Modelo Vista Controlador que hace posible tener una buena organización de archivos y esto facilita el mantenimiento para el desarrollo de sistemas de información.

## REFERENCIAS BIBLIOGRÁFICAS

- Abrahamsson, P.**, (2006) *Extreme Programming and Agile Processes in Software Engineering*. 1ª Edición. Oulu, Springer.
- AECEM**, (2009) Asociación Española de Comercio Electrónico y Marketing Relacional. *Libro Blanco del Comercio Electrónico*. 1ª Edición. Licencia Creative Commons.
- AJAX Ya**, (2012) *AJAX*. [En línea]. Disponible en: <http://www.ajaxya.com.ar/> [Acceso 08 de agosto 2012].
- Ambler, S.**, (2002) *Agile Modeling: Effective Practices for eXtreme Programming and the Unified Process*. 1ª Edición. Nueva York, John Wiley & Sons.
- América Economía**, (2012) *Estudio de Comercio Electrónico en América Latina Mayo 2012*. [En línea]. Disponible en: [http://especiales.americaeconomia.com/2012/comercio-electronico-america-latina-2012/pdf/Estudio\\_de\\_comercio\\_electronico\\_en\\_America\\_Latina\\_Mayo\\_2012.pdf](http://especiales.americaeconomia.com/2012/comercio-electronico-america-latina-2012/pdf/Estudio_de_comercio_electronico_en_America_Latina_Mayo_2012.pdf) [Acceso 01 de agosto 2012].
- Anetcom**, (2004) *Los dominios en Internet*. [En línea]. Disponible en: <http://video.anetcom.es/editorial/librodominios.pdf> [Acceso 01 de abril 2012].
- Anon**, (2002) *Portal de Programación Extrema en Español*. [En línea]. Disponible en: <http://www.programacionextrema.org/> [Acceso 01 de agosto 2012].
- Auer, K. y Roy, M.**, (2001) *Extreme Programming Applied: Playing to Win*. 1ª Edición. Boston, Addison-Wesley Professional.
- Baintrive**, (2013) *10 Reglas Heurísticas de Usabilidad de Jakob Nielsen*. [En línea]. Disponible en: [web.baintrive.com/10-reglas-heuristicas-de-usabilidad-de-jakob-nielsen/](http://web.baintrive.com/10-reglas-heuristicas-de-usabilidad-de-jakob-nielsen/) [Acceso 20 de junio 2013]
- Bair, S.**, (2002) *Teach Yourself Extreme Programming*. 1ª Edición. Indianapolis, Sams Publishing.
- Beck, K.**, (2000) *Extreme Programming Explained. Embrace Change*. 1ª Edición. US, Addison Wesley.
- Beck K.**, (2004) *Extreme Programming Explained: Embrace Change*. 2ª Edición. US, Addison Wesley.
- Berzal, F.**, (2006) *Interfaces Web. Evolución de las aplicaciones Web*. [En línea]. Disponible en: <http://elvex.ugr.es/decsai/csharp/pdf/web/1-web.pdf> [Acceso 20 de julio 2011]
- Boiko, B.**, (2005) *Content Management Bible*. Segunda Edición. Indiana, Wiley.
- Bruegge, B. y Dutoit, A.**, (2002) *Ingeniería de Software Orientado a Objetos*. 1ª Edición. México, Prentice Hall.
- Cárdenas, P. y Tocora, C.**, (2008) *Implementación de un Content Management System (CMS) en la Escuela de Ingeniería de Sistemas y Telecomunicaciones*. Tesis de Licenciatura. Colombia, Universidad Sergio Arboleda.
- Cuvillo, C.**, (2005) *Métricas del Software*. [En línea]. Disponible en: <http://www.oei.eui.upm.es/Asignaturas/PIInformaticos/ficheros/temario/PROYINF-2.pdf> [Acceso 08 de junio 2012]
- Eguíluz Pérez, Javier**. (2009a). *Introducción a JavaScript*. [En línea]. Disponible en: [http://www.librosweb.es/javascript/pdf/introduccion\\_javascript.pdf](http://www.librosweb.es/javascript/pdf/introduccion_javascript.pdf) [Acceso 27 de enero 2012].

- Eguíluz, J.**, (2009b) *Introducción a CSS*. [En línea]. Disponible en: [http://www.librosweb.es/css/pdf/introduccion\\_css.pdf](http://www.librosweb.es/css/pdf/introduccion_css.pdf) [Acceso 27 de enero 2012].
- EllisLab.** (2012) *Manual de Code Igniter en Español*. [En línea]. Disponible en: <http://codeigniter.com/> [Acceso 08 de junio 2012].
- Escobedo, Q.** (2008) *Propuesta de un Sistema basado en estándares web para la gestión del contenido de documentos de formación académica*. Tesis de Pregrado. Perú, Universidad Católica de Santa María.
- Fernández, G.**, (2002) *Introducción a Extreme Programming*. [En línea]. Disponible en: <http://www.info-ab.uclm.es/asignaturas/42551/trabajosAnteriores/Presentacion-XP.pdf> [Acceso 14 de julio 2011].
- Futuro Labs**, (2012) *Estudio de Comercio Electrónico en el Perú*. [En línea]. Disponible en: <http://www.futurolabs.com/i-estudio-de-comercio-electronico-en-el-peru/> [Acceso 13 de enero 2013].
- Gonzales, M.**, (2012) *Proyecto E-Commerce*. [En línea]. Disponible en: <http://www.slideshare.net/PedroChavez1/estudio-sobre-comercio-electronico-en-el-peru> [Acceso 23 de octubre 2012].
- Hernandez, R. y Fernandez C.**, (2000) *Metodología de la investigación*. 1ª Edición. México, Ultra.
- Holdener, A.**, (2008) *Ajax: The definitive guide*. 1ª Edición. Beijing O'Reilly.
- IAB**, Interactive Advertising Bureau. (2010) *SEO: Optimización de Webs para Buscadores*. Madrid. [En línea]. Disponible en: <http://www.iabspain.net/libros-blancos/> [Acceso 12 de enero 2013].
- Iasoni, M.** (2002) *Comercio Electrónico, Aspectos Legales: Desafío para el Derecho Peruano*. 1ª Edición. Perú, Portocarrero.
- Jeffries, R.**, (1999) *xProgramming*. [En línea]. Disponible en: <http://xprogramming.com/> [Acceso 12 de enero 2013]
- Jeffries, R., Anderson, A. y Hendrickson, A.**, (2000) *Extreme Programming Installed*. 1ª Edición. Boston, Addison-Wesley Professional.
- JSP**, Java Server Pages Overview (2006). *JavaServer Pages Overview*. [En línea]. Disponible en: <http://java.sun.com/products/jsp/overview.html> [Acceso 04 de julio de 2011]
- Kabir, J.**, (2003) *La biblia del servidor Apache*. 1ª Edición. España: Anaya Multimedia.
- Labrada, E. y Salgado C.**, (2013) *Diseño Web Adaptativo o Responsivo*. Revista Digital Universitaria. [En línea]. Disponible en: <http://www.revista.unam.mx/vol.14/num1/art07/index.html> [Acceso 12 de enero 2013].
- Lawrence, S.**, (2002) *Ingeniería de Software: Teoría y Práctica*. 1ª Edición. Sao Paulo, Prentice Hall.
- Lorenzetti, R.**, (2000) *Tratado de los contratos*. Buenos Aires, Rubinzal-Culzoni Editores.
- Margaix, D.**, (2007) *Conceptos de Web 2.0 y Biblioteca 2.0: origen, definiciones y retos para las bibliotecas actuales*. [En línea]. Disponible en: <http://eprints.rclis.org/9521/1/kx5j65q110j51203.pdf> [Acceso 12 de enero 2013].

**Morales, P.**, (2012) *Es el Momento del E-Commerce en el Perú*. [Video]. Disponible en: <http://mass.pe/video/es-el-momento-del-e-commerce-en-el-peru> [Acceso 20 de agosto 2012].

**MSDN Library**, (en español). (2006) *Introducción ASP.NET*. [En línea]. Disponible en: <http://msdn.microsoft.com/library/default.asp> [Acceso 04 de julio 2011].

**O'Reilly, T.**, (2005) *What is web 2.0: design patterns and business models for the next generation of software*. [En línea]. Disponible en: <http://www.oreillynet.com/go/web2> [Acceso 01 de abril 2012].

**PCM, Presidencia del Consejo de Ministros**, (2011) *Plan de Desarrollo de la Sociedad de Información en el Perú La Agenda Digital 2.0*, promulgado el 26 de julio del 2001. [En línea]. Disponible en: [http://www.codesi.gob.pe/docs/AgendaDigital20\\_28octubre\\_2011.pdf](http://www.codesi.gob.pe/docs/AgendaDigital20_28octubre_2011.pdf) [Acceso 26 de julio 2012]

**Pecos, Daniel**. (2008). *MySQL*. [En línea]. Disponible en: [http://danielpecos.com/docs/mysql\\_postgres/index.html#AEN11](http://danielpecos.com/docs/mysql_postgres/index.html#AEN11) [Acceso 08 de junio 2011]

**Pressman, Roger S.** (2005). *Ingeniería de Software un enfoque práctico*. 6ª Edición. Madrid España: McGraw Hill.

**Pressman, Roger S.** (2010). *Ingeniería de Software un enfoque práctico*. 7ª Edición. Madrid España: McGraw Hill.

**Powers, D.**, (2008) *Desarrollo Web con Dreamweaver CS3, CSS, AJAX y Php*. 1ª Edición. España, Anaya Multimedia.

**Rahmel, D.**, (1999) *Professional Joomla!*. US, John Wiley & Sons, Inc.

**Robertson, J.**, (2002) *Cuáles son las metas del CMS*. [En línea]. Disponible en: [http://www.steptwo.com.au/papers/kmc\\_goals/index.html](http://www.steptwo.com.au/papers/kmc_goals/index.html) [Acceso 08 de junio 2012]

**Robertson, J.**, (2003a) *Entonces que es un Content Management System*. [En línea]. Disponible en: [http://www.steptwo.com.au/papers/kmc\\_what/index.html](http://www.steptwo.com.au/papers/kmc_what/index.html) [Acceso 08 de junio 2012].

**Robertson, J.**, (2003b) *How to Evaluate a Content Management System*. [En línea]. Disponible en: [http://www.steptwo.com.au/papers/kmc\\_evaluate/index.html](http://www.steptwo.com.au/papers/kmc_evaluate/index.html) [Acceso 08 de junio 2012].

**Robertson, J.**, (2003c) *Why Every Small Website Needs A Content Management System*. [En línea]. Disponible en: [http://www.steptwo.com.au/papers/cmb\\_needcms/index.html](http://www.steptwo.com.au/papers/cmb_needcms/index.html) [Acceso 08 de junio 2012].

**Robertson, J.**, (2004) *Ciclo de vida del Content Management System*. [En línea]. Disponible en: <http://www.steptwo.com.au/cm/lifecycle/index.html> [Acceso 08 de junio 2012].

**Rodríguez, E.**, (2007) *Comercio Electrónico*. [En línea]. Disponible en: <http://homepage.mac.com/eravila/eCommercel.pdf> [Acceso 08 de junio 2011].

**Salas, M.**, (2000) *Diseño y promoción de un sitio web dirigido a pequeños y microempresarios interesados en implementar Internet y el comercio electrónico a las estrategias de la organización*. Tesis de Pregrado. Perú, Universidad Católica Santa María Arequipa.

**Soto, C.,** (2010) *El Comercio Electrónico en el Derecho Peruano*. [En línea]. Disponible en: <http://derecho-uigv.com/AGORA/revistas/R000005.pdf> [Acceso 12 de mayo del 2012]-

**Sucasaire W.,** (2007) *Prototipo de Sistema E-Marketplace para las pequeñas empresas distribuidoras de artesanía de la ciudad de Puno*. Tesis de Pregrado. Perú, Universidad Nacional del Altiplano Puno.

**Trigos, E.,** (2000) *PHP 4*. 3ª Edición. España, Anaya Multimedia S.A.





## Anexo A

Encuesta a representantes de empresas artesanales de la región Puno para formulación del problema.

Nombre de la empresa: .....

Responsable: .....

Dirección: .....

RUC: .....Nro Artesanos.....

1. ¿Exporta su productos?

Si ( ) No ( )

2. ¿Cómo realiza el manejo de información (inventarios, caja, presupuestos, pedidos, facturación, etc.) en su empresa?.

Manual ( ) Automatizado ( ) Manual y automatizado ( )

3. ¿Hace Ud. uso del Internet? (Tiene pagina web)

Si ( ) No ( )

.....

4. ¿Cree usted que el uso del Internet ayudaría a mejorar sus ventas?

Si ( ) No ( )

Porque?.....

.....

5. ¿Tiene Ud, Conocimiento sobre el Comercio electrónico?

Si ( ) No ( )

6. ¿Esta en condiciones de asignar un presupuesto para el uso de nuevas tecnologías para su empresa? (Que tecnología es importante para usted)

Si ( ) No ( )

Porque?

.....

.....

.....

7. ¿Su empresa esta en condiciones de exportar?

8.

Si ( ) No ( ) Porque .....

.....

.....

9. ¿Como van sus ventas en comparación de años anteriores? (consecuencia)

**Anexo B**

Encuesta a usuarios del sistema, para validación del Prototipo de Sistema de Administración de Contenidos para la Implementación y Mantenimiento de Portales Web Orientados al E-Commerce.

1. ¿Cómo califica el diseño de interfaz del sistema?
  - a) Excelente
  - b) Bueno
  - c) Regular
  - d) Malo
2. ¿Qué tan fácil le parece el uso o manejo del sistema?
  - a) Fácil
  - b) Regular
  - c) Difícil
3. ¿Cuál es su opinión en cuanto a la rapidez del sistema?
  - a) Excelente
  - b) Bueno
  - c) Regular
  - d) Malo
4. ¿Cómo considera Ud. el aprendizaje del sistema en cuanto a la implementación?
  - a) Fácil de aprender
  - b) Moderadamente complejo
  - c) Difícil de aprender
5. ¿Cuál es su opinión con respecto al grado de dificultad en cuanto a la implementación de la tienda virtual?
  - a) Fácil
  - b) Regular
  - c) Difícil
6. ¿Cómo considera Ud. la actualización de los contenidos del Portal de E-Commerce con el sistema administrador de contenidos?

- a) Fácil
- b) Regular
- c) Difícil

7. Sobre la autodescripción del sistema, usted cree que es:

- a) Autodescriptible
- b) No autodescriptible

8. ¿El sistema propuesto, le facilita el mantenimiento y la implementación de un portal de E-commerce?

- a) Si
- b) No

