

# **ESCUELA POLITÉCNICA NACIONAL**

**FACULTAD DE INGENIERÍA DE SISTEMAS**

**ELABORACIÓN DE UNA PROPUESTA DE PROCESO DE  
DESARROLLO PARA SISTEMAS WEB BASADA EN LAS  
PRINCIPALES METODOLOGÍAS EXISTENTES**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN  
SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN**

**CARLOS ANDRÉS FABARA DE LA PAZ**  
c\_a\_fabara@hotmail.com

**DIRECTOR: ING. MARCOS RAÚL CÓRDOVA BAYAS**  
raul.cordova@epn.edu.ec

**Quito, Febrero 2012**

## **DECLARACIÓN**

Yo, Carlos Andrés Fabara De la Paz, declaro bajo juramento que el trabajo aquí descrito es de mi autoría, que no ha sido previamente presentado para ningún grado o calificación profesional y que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo a la Escuela Politécnica Nacional, según lo establecido en la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad constitucional vigente.

---

Carlos Andrés Fabara De la Paz

## **CERTIFICACIÓN**

Certifico que el presente trabajo fue desarrollado por Carlos Andrés Fabara De la Paz bajo mi supervisión.

---

Ing. Raúl Córdova  
**DIRECTOR DEL PROYECTO**

## **AGRADECIMIENTOS**

*A Dios, por los dones que me ha dado  
y por haberme guiado en cada  
momento de mi vida.*

*A mis padres Patricio y María Cristina  
por sus enseñanzas y apoyo constante.  
En especial por la confianza que han  
tenido en mí.*

*A mis hermanos Daniel y Esteban por  
ser aquellos compañeros  
incondicionales que la vida me brindó.*

*A mi tutor Raúl y a todos aquellos  
maestros que durante estos años han  
compartido conmigo sus consejos,  
sabiduría y amistad.*

*A todos aquellos amigos y amigas que  
durante la carrera me supieron brindar  
su amistad, en especial a mí querida  
amiga Andrea*

*Carlos*

## **DEDICATORIA**

*A mi mamá por ser mi amiga y mi mayor apoyo, gracias por cada consejo que me diste y porque tus palabras ayudaron a forjar la persona que soy.*

*A mi papá por haberme motivado a seguir adelante y enseñarme a nunca rendirme. Y en especial gracias por haber confiado en mí.*

*A mi abuelito Pepe, que desde el cielo me dio la fuerza para avanzar en todo momento. Gracias.*

*Carlos*

## Tabla de Contenido

PRESENTACIÓN .....	1
RESUMEN .....	2
CAPÍTULO 1. DESCRIPCIÓN DEL PROBLEMA .....	3
1.1. ESTADO ACTUAL DEL DESARROLLO WEB.....	3
1.1.1. SISTEMAS BASADOS EN LA WEB.....	3
1.1.2. TECNOLOGÍAS INVOLUCRADAS.....	5
1.1.3. METODOLOGÍAS.....	14
1.2. DETERMINACIÓN DE LAS VENTAJAS Y DESVENTAJAS DE LAS PRINCIPALES METODOLOGÍAS UTILIZADAS EN EL DESARROLLO WEB.	16
1.2.1. WebML .....	16
1.2.2. WSDM .....	21
1.2.3. OOHDM.....	28
1.2.4. SCRUM .....	31
1.3. DETERMINACIÓN DE LOS ASPECTOS ADICIONALES APLICABLES AL DESARROLLO WEB.....	34
1.3.1. DISEÑO GRÁFICO PARA LA WEB .....	34
1.3.2. ARQUITECTURA DE INFORMACIÓN .....	37
1.3.3. DISEÑO DE INTERFACES WEB .....	38
1.3.4. DISEÑO DE LA NAVEGACIÓN.....	41
1.3.5. REUTILIZACIÓN DE SOFTWARE .....	42
CAPÍTULO 2. DEFINICIÓN DE LA PROPUESTA .....	43
2.1. DEFINICIÓN Y ESPECIFICACIÓN DE LAS FASES DEL PROCESO....	43
2.1.1. CONCEPTUALIZACIÓN.....	46
2.1.2. REQUERIMIENTOS.....	47
2.1.3. DISEÑO .....	48

2.1.4.	IMPLEMENTACIÓN .....	50
2.1.5.	PRUEBAS .....	51
2.1.6.	LIBERACIÓN .....	52
2.1.7.	MANTENIMIENTO Y EVOLUCIÓN .....	52
2.2.	DISEÑO DE ENTREGABLES DEL PROCESO .....	53
2.2.1.	CONCEPTUALIZACIÓN.....	53
2.2.2.	REQUERIMIENTOS.....	56
2.2.3.	DISEÑO.....	58
2.2.4.	IMPLEMENTACIÓN .....	59
2.2.5.	PRUEBAS .....	60
2.2.6.	LIBERACIÓN.....	62
2.2.7.	MANTENIMIENTO Y EVOLUCIÓN .....	63
2.3.	DEFINICIÓN DE PROCEDIMIENTOS Y HERRAMIENTAS PARA CADA FASE DEL PROCESO.....	64
2.3.1.	CONCEPTUALIZACIÓN.....	64
2.3.2.	REQUERIMIENTOS.....	65
2.3.3.	DISEÑO.....	66
2.3.4.	IMPLEMENTACIÓN .....	68
2.3.5.	PRUEBAS .....	69
2.3.6.	LIBERACIÓN.....	70
2.3.7.	MANTENIMIENTO Y EVOLUCIÓN .....	71
2.4.	FLUJOS DE APOYO AL PROCESO.....	71
2.4.1.	FLUJO DE GESTIÓN DEL PROYECTO .....	71
2.4.2.	FLUJO DE GESTIÓN DE CALIDAD.....	72
CAPÍTULO 3.	CASO DE ESTUDIO .....	74
3.1.	DESCRIPCIÓN DEL CASO DE ESTUDIO .....	74
3.2.	APLICACIÓN DE LA PROPUESTA .....	74

3.2.1. CONCEPTUALIZACIÓN.....	74
3.2.2. REQUERIMIENTOS.....	76
3.2.3. DISEÑO.....	81
3.2.4. IMPLEMENTACIÓN.....	85
3.2.5. PRUEBAS.....	89
3.2.6. LIBERACIÓN.....	91
3.2.7. MANTENIMIENTO Y EVOLUCIÓN.....	93
3.2.8. FLUJO DE GESTIÓN DEL PROYECTO.....	93
3.2.9. FLUJO DE GESTIÓN DE CALIDAD.....	97
3.3. ANÁLISIS DE RESULTADOS.....	97
3.3.1. ANÁLISIS INTERNO.....	97
3.3.2. ANÁLISIS EXTERNO.....	100
CAPÍTULO 4. CONCLUSIONES Y RECOMENDACIONES.....	107
4.1. CONCLUSIONES.....	107
4.2. RECOMENDACIONES.....	108
GLOSARIO.....	110
BIBLIOGRAFÍA.....	112
ANEXOS.....	114



## Lista de Figuras

Figura 1-1 Ejemplo de una página HTML simple. ....	7
Figura 1-2 Modelo de Datos de Información Discográfica.....	18
Figura 1-3 Modelo de Hipertexto para un Sistema de Información Discográfica..	19
Figura 1-4 Diagrama de fases y modelos de WebML .....	20
Figura 1-5 Representación de una clase y subclases de audiencia.....	23
Figura 1-6 Diagrama CTT para la tarea de buscar información .....	24
Figura 1-7 Ejemplo de un "object chunk" .....	25
Figura 1-8 Ejemplo de diagrama de navegación externo .....	25
Figura 1-9 Diagrama de fases y modelos de WSDM .....	27
Figura 1-10 Diagrama de fases y modelos de OOHDM .....	30
Figura 1-11 Fases de SCRUM .....	34
Figura 1-12 Ejemplo de evolución del diseño web (Apple).....	36
Figura 1-13 Los tres círculos de la Arquitectura de Información .....	38
Figura 1-14 Patrones Estándares de Interacción .....	40
Figura 1-15 Ejemplo de Wireframe.....	40
Figura 1-16 Ejemplo de Mapa del Sitio (bazuca.com).....	41
Figura 2-1 Diagrama del Proceso PRODEWEB.....	45
Figura 2-2 Formato del Documento de Modelado del Negocio .....	54
Figura 2-3 Formato del Checklist de Viabilidad .....	55
Figura 2-4 Formato del Documento de Especificación de Requerimientos.....	57
Figura 2-5 Formato del Documento de Caracterización de la Audiencia .....	57
Figura 2-6 Formato Documento de Diseño .....	59
Figura 2-7 Formato Bitácora de Pruebas .....	61
Figura 2-8 Formato Bitácora de Errores.....	61
Figura 2-9 Formato Listado de Cambios .....	62
Figura 2-10 Formato Bitácora de Tareas .....	72
Figura 3-1 Modelado de Negocio (Caso de Estudio).....	75
Figura 3-2 Checklist de Viabilidad (Caso de Estudio) .....	76
Figura 3-3 Especificación de Requerimientos (Caso de Estudio).....	80
Figura 3-4 Caracterización de la Audiencia (Caso de Estudio) .....	81

Figura 3-5 Diagrama de Arquitectura .....	82
Figura 3-6 Modelo de Datos (Caso de Estudio) .....	82
Figura 3-7 Mapa del Sitio (Caso de Estudio).....	83
Figura 3-8 Wireframes (Caso de Estudio) .....	84
Figura 3-9 Construcción Plantilla del sistema.....	86
Figura 3-10 Modelo de datos implementado .....	87
Figura 3-11 Captura del sistema de catálogo en funcionamiento.....	88
Figura 3-12 Captura de la sección administrativa del caso de estudio.....	88
Figura 3-13 Bitácora de Pruebas (Caso de Estudio) .....	90
Figura 3-14 Bitácora de Errores (Caso de Estudio).....	91
Figura 3-15 Listado de Cambios (Caso de Estudio).....	92
Figura 3-16 Bitácora de Tareas (Caso de Estudio) .....	96
Figura 3-17 Formato encuesta realizada.....	102
Figura 3-18 Niveles de Acuerdo con la propuesta.....	104
Figura 3-19 Disposición a usar la propuesta .....	105
Figura 3-20 Disposición a recomendar la propuesta .....	106

## Lista de Tablas

Tabla 1-1 Estadísticas de Uso de Navegadores Octubre 2011 .....	6
Tabla 1-2 Principales Frameworks y Librerías Javascript .....	10
Tabla 1-3 Complementos para la navegación .....	11
Tabla 2-1 Herramientas para la etapa "Conceptualización" .....	65
Tabla 2-2 Herramientas para la etapa "Requerimientos" .....	66
Tabla 2-3 Herramientas para la etapa "Diseño de Arquitectura" .....	67
Tabla 2-4 Herramientas para la etapa "Diseño de Datos y Lógica" .....	68
Tabla 2-5 Herramientas para la etapa "Diseño Gráfico y de Interfaces" .....	68
Tabla 2-6 Herramientas para la etapa "Diseño de la Navegación" .....	68
Tabla 2-7 Herramientas para la etapa "Selección de Componentes" .....	68
Tabla 2-8 Herramientas para la etapa "Implementación" .....	69
Tabla 2-9 Herramientas para la etapa "Pruebas" .....	70
Tabla 2-10 Herramientas para la etapa "Liberación" .....	70
Tabla 2-11 Herramientas para la etapa "Mantenimiento y Evolución" .....	71
Tabla 3-1 Uso de metodologías en desarrollo web .....	103
Tabla 3-2 Niveles de Acuerdo con la propuesta .....	103
Tabla 3-3 Disposición a usar la propuesta .....	104
Tabla 3-4 Disposición a recomendar la propuesta .....	106

## **PRESENTACIÓN**

El mundo de las tecnologías relacionadas a la web se encuentra en constante avance y evolución. Del mismo modo se requiere también que los procesos y metodologías para el desarrollo de software se adapten a las nuevas tendencias y necesidades.

En la práctica es muy difícil adaptar los procesos de desarrollo tradicionales, debido a su formalidad, es por este motivo que las metodologías ágiles, han tenido un auge, en especial dentro de los proyectos web, ya que su adaptabilidad permite que puedan acoplarse más fácilmente a múltiples entornos y escenarios.

El presente trabajo busca analizar las principales metodologías diseñadas exclusivamente para desarrollo web, y posteriormente basándose en una de ellas proponer un nuevo proceso de desarrollo, que incorpore lo mejor de las metodologías analizadas y los aspectos más relevantes para el desarrollo de sistemas web.

## **RESUMEN**

El Proyecto de Titulación tiene como finalidad la elaboración de una propuesta de proceso para el desarrollo de sistemas web, basándose en las principales metodologías existentes y complementándola con aquellas características utilizadas en el desarrollo web.

El primer capítulo brinda una visión global de la problemática que el presente trabajo busca resolver. Inicia con una introducción al estado actual del desarrollo web y sus tecnologías relacionadas. Continúa después con un análisis de las ventajas y desventajas de las principales metodologías utilizadas para el desarrollo web. Finalmente se determinan un conjunto de aspectos importantes que deben considerarse durante el desarrollo de un sistema web.

El segundo capítulo abarca el desarrollo de la propuesta como tal, tomando como base la investigación y el análisis realizado durante el capítulo uno. Tras seleccionar una metodología base se procede a adaptarla y a definir en detalle cada una de sus fases, incluyendo su documentación, procedimientos y herramientas recomendadas.

Para poder validar el proceso durante el capítulo tres se procede a aplicarla en un caso de estudio, que consistió en el desarrollo de un sistema de catálogo de productos. En este capítulo se detallan las experiencias obtenidas tras aplicar el proceso a un desarrollo real y posteriormente se analizan los resultados obtenidos.

Finalmente el capítulo cuatro contiene las conclusiones y recomendaciones obtenidas tras la elaboración y aplicación de la propuesta de proceso de desarrollo.

# **CAPÍTULO 1. DESCRIPCIÓN DEL PROBLEMA**

## **1.1. ESTADO ACTUAL DEL DESARROLLO WEB**

### **1.1.1. SISTEMAS BASADOS EN LA WEB**

Tradicionalmente los aplicativos de software eran construidos con la finalidad de que sean instalados en el computador del cliente de manera que estos puedan funcionar de manera independiente. Este modelo de distribución se acentuó a finales de los años 70 debido a la popularización de las computadoras personales.

Posteriormente a finales de la década de los 80, con el auge de las redes de comunicación y el aumento de las capacidades de procesamiento de los computadores, se dieron condiciones para permitir el desarrollo de aplicaciones distribuidas que podían comunicarse entre sí, a través de equipos en una red sin importar que estos se encuentren geográficamente dispersos.

Para la creación de aplicaciones distribuidas el modelo Cliente - Servidor es uno de los más difundidos. El modelo implica que existan:

- Un equipo o conjunto de equipos que actúen como Servidor.
- Uno o más equipos que se conecten al Servidor que se denominaran Clientes.
- Una infraestructura de comunicación entre los Clientes y el Servidor.

Dependiendo de cuál sea el equipo en el que se realiza la mayor carga de procesamiento se pueden reconocer varios tipos de aplicativos cliente, estos son:

- Cliente Pesado
- Cliente Ligero
- Cliente Híbrido

Un Cliente Pesado es aquel aplicativo que se instala en el equipo cliente, este software es el encargado de realizar la mayor carga de procesamiento. La interacción con el servidor es para consulta o almacenamiento de información. Un claro ejemplo de este tipo de software son los aplicativos de caja y facturación.

Los Clientes Ligeros son aquellos aplicativos que principalmente realizan tareas de presentación e ingreso de información ya que la mayor cantidad de procesamiento se lo realiza en el equipo servidor. El mayor representante de un cliente ligero es una página web, ya que casi todas las tareas de procesamiento se realizan en el servidor, tras las peticiones realizadas por el navegador web.

Cliente Híbrido es un término emergente que se refiere a los aplicativos de software que reparten la carga de procesamiento entre el servidor y el cliente. Se puede considerar que las aplicaciones RIA (Rich Internet Applications) son un ejemplo de clientes híbridos, ya que reparten ciertas tareas de procesamiento del servidor hacia el cliente.

El auge de las aplicaciones web se debe a las ventajas que ofrece sobre las aplicaciones tradicionales [1], que son:

- Capacidad de despliegue rápido.
- Acceso a los datos desde cualquier lugar.
- Se obtiene siempre la última versión de la aplicación.
- Cero instalación. Solo se requiere un navegador web para acceder.
- Independencia de la plataforma.
- Fácilmente adaptables.
- Alcance global (requiere publicación en internet).
- Entorno familiar para los usuarios.
- Gran importancia al contenido y aspecto visual.

Todas estas características que hacen útiles y versátiles a las aplicaciones web generan una serie de desafíos al momento de aplicar ingeniería al desarrollo de los mismos. Muchas metodologías tradicionales no consideran la adaptabilidad

como una prioridad dentro de su proceso. En otro aspecto, lo que generalmente se considera un prototipo, en el ambiente web debido a su característica de despliegue rápido, se considera un producto funcional; por citar un ejemplo, dentro de la primera iteración se podría construir la Página Inicial del sistema, la misma que puede ya publicarse, al ser un producto funcional.

A su vez existen muchos otros aspectos que se deben considerar para un desarrollo web de calidad, un claro representante de estos aspectos es el Diseño Gráfico, su importancia radica en que el aspecto visual del sitio es primordial para mejorar la experiencia de uso del usuario.

El proceso que se va a proponer, busca solventar las falencias que se dan durante el desarrollo de sistemas web, el proceso se basará en el estudio de las principales metodologías web de las cuales se tomará, lo mejor que tengan de forma que se obtenga un proceso de desarrollo adecuado para la creación de aplicaciones web de baja y media complejidad.

### **1.1.2. TECNOLOGÍAS INVOLUCRADAS**

Dentro de las tecnologías de software que conforman la web se pueden identificar dos grandes áreas:

- Front-end (Cliente)
- Back-end (Servidor)

#### **1.1.2.1. Front-end**






Las tecnologías dentro de esta categoría son aquellas utilizadas por los usuarios de un sitio o aplicación web. Generalmente estos usuarios son conocidos como clientes ya que son los que consumen los servicios ofrecidos por la aplicación.



### 1.1.2.1.1. Navegadores Web

Son aplicativos de software cuya función principal es la de representar el código HTML, CSS, Javascript, etc; codificado en una página web, interpretándolo y transformándolo en una representación gráfica de textos, imágenes, tablas, colores, botones y otros elementos, con la finalidad de permitir la visualización, utilización e interacción del usuario con el sistema web.

En la Tabla 1-1 se puede observar una lista de los navegadores más difundidos junto a sus porcentajes de uso a nivel Mundial y en Ecuador.

	<b>Navegador Web</b>	<b>% Uso Mundial</b>	<b>% Uso Ecuador</b>
	Internet Explorer	<b>40.18</b>	33.45
	Firefox	26.39	<b>34.36</b>
	Google Chrome	25	28.97
	Safari	5.93	2.45
	Opera	1.81	0.57

**Tabla 1-1 Estadísticas de Uso de Navegadores Octubre 2011**

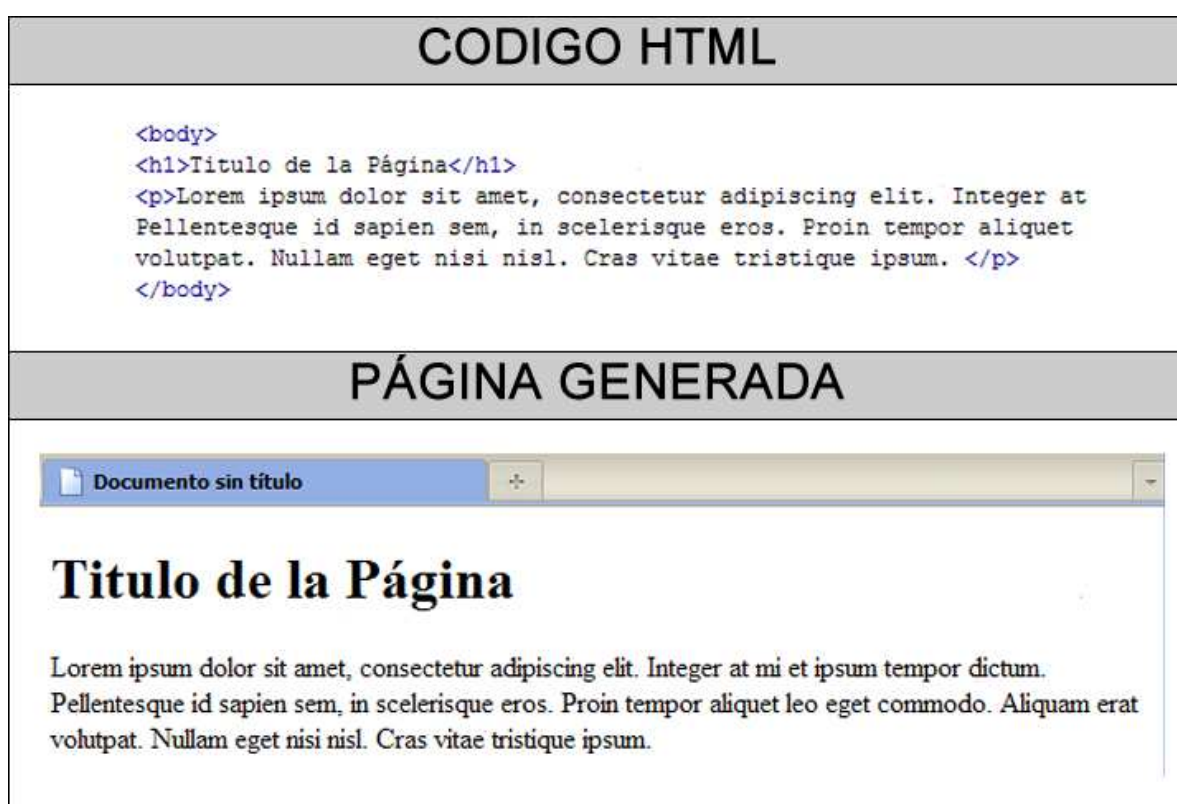
Fuente: <http://gs.statcounter.com/#browser-ww-monthly-201010-201110>

Realizado por: Carlos Fabara

La tecnología móvil ha crecido significativamente, por lo que en la actualidad es común la navegación a través de dispositivos móviles (teléfonos inteligentes, asistentes personales o PDA's) los cuales implementan versiones reducidas o específicas de los navegadores web que varían según el fabricante.

### 1.1.2.1.2. HTML

HTML proviene del acrónimo de Hypertext Text Markup Language; como su nombre lo indica, es un lenguaje de marcado o etiquetas similar al XML. Este lenguaje permite codificar la información, esquematizarla y organizarla dentro de un formato determinado por dichas etiquetas, en la Figura 1-1 se muestra un ejemplo de cómo se encuentra codificada una página web.



**Figura 1-1 Ejemplo de una página HTML simple.**  
Realizado por: Carlos Fabara

Actualmente se está trabajando con la especificación HTML 5 [2], en la cual se añaden múltiples y novedosas características con las que se busca disminuir la dependencia hacia Javascript y hacia complementos para la navegación (Plugins) para la realización de tareas comunes o repetitivas.

En el nuevo estándar de HTML5 se añadieron características como:

- Elementos de audio y video para la reproducción de elementos multimedia sin la necesidad de complementos externos.
- Capacidad de almacenamiento local de datos, con la característica de Local Storage.
- Elemento `<canvas>` que sirve como lienzo para la renderización, transformación y animación de elementos vectoriales y mapas de bits.
- Nuevo elemento `<figure>` que permite crear imágenes con un pie de imagen asociado semánticamente.
- Eliminación de la necesidad de definir el tipo (`type`) en elementos `<link>`, `<style>` y `<script>`.
- Simplificación del uso de la etiqueta `<!DOCTYPE>` y `<meta>`.
- Caja de texto específica para el ingreso de direcciones electrónicas `<input type="email">`.
- Atributo `required` para elementos de un formulario que deban encontrarse completos antes de enviarse.
- Uso de expresiones regulares en formularios a través del atributo `pattern`.
- Posibilidad de que algunos elementos sean editables a través del atributo `contenteditable="true"`.

Muchas de las características antes mencionadas no se encuentran completamente implementadas en algunos navegadores, y tomando en cuenta que se estima que la implementación del estándar HTML5 no será completa hasta el 2022 [3], se recomienda que antes de utilizar cualquiera de las nuevas características se analice cual sería el impacto hacia los usuarios.

#### *1.1.2.1.3. CSS*

Las hojas de estilo en cascada son archivos de texto que contienen la especificación del esquema visual de la página web (colores, esquema, tamaños, etc.). Surgió como una alternativa para permitir la separación entre el contenido representado en código HTML y la presentación representada en código CSS.

En este momento se trabaja en la especificación CSS3, misma que busca añadir múltiples características para evitar el uso excesivo de imágenes en tareas simples como degradados, bordes redondeados, entre otros. Cabe recalcar que la especificación 3 de CSS aún se encuentra en una etapa temprana de desarrollo, pero aun así ya se están implementando algunas características en los navegadores web.

Entre las principales novedades de la versión 3 se encuentran:

- Crear bordes redondeados con la propiedad `border-radius`.
- Crear sombras a través de la propiedad `box-shadow`.
- Aplicar un formato de múltiples columnas al texto.
- Uso de fuentes personalizadas a través de `@font-face`.
- Controlar la opacidad a través del atributo `opacity`.
- Posibilidad de trabajar con colores del tipo RGBA.






#### *1.1.2.1.4. Javascript*

Es un lenguaje de scripting, orientado a objetos y no tipado; su sintaxis es similar a la del lenguaje C++. Su principal uso es dentro de los navegadores web para permitir la ejecución de lógica del lado del cliente, liberando de tareas tediosas y repetitivas al servidor como son las alertas, validaciones, mensajes, etc.

Funciona dentro de un sandbox o entorno protegido dentro del navegador con lo que se elimina la posibilidad de ataques a través de este lenguaje. Al ser un lenguaje ligero, rápido y seguro ha extendido su uso y se encuentra implementado en todos los navegadores de la actualidad.

Los mayores usos del Javascript son dentro de la tecnología AJAX y para dotar de una mayor interactividad a las interfaces de usuario, permitiendo que se asemejen cada vez más a las interfaces de una aplicación de escritorio.

Una de las formas más comunes de encontrar implementado Javascript es a través del uso de los diversos frameworks y librerías existentes. Los dos agrupan código reutilizable que permite una fácil implementación de tareas repetitivas. Algunas de las librerías más utilizadas se resumen en la tabla Tabla 1-2.

	Librería	Sitio Web
	JQuery	<a href="http://jquery.com/">http://jquery.com/</a>
	MooTools	<a href="http://mootools.net/">http://mootools.net/</a>
	Dojo Toolkit	<a href="http://dojotoolkit.org/">http://dojotoolkit.org/</a>
	YUI	<a href="http://developer.yahoo.com/yui/">http://developer.yahoo.com/yui/</a>
	Prototype	<a href="http://www.prototypejs.org/">http://www.prototypejs.org/</a>




**Tabla 1-2 Principales Frameworks y Librerías Javascript**  
Realizado por: Carlos Fabara

#### 1.1.2.1.5. *Plugins o complementos*

Los complementos a la navegación son entornos que permiten la ejecución de código no-HTML con la finalidad de brindar un mayor grado de interactividad, usabilidad y la realización de tareas que el estándar HTML no permitía.

Una de las mayores limitaciones que se tienen al utilizarlos es que no siempre son independientes de la plataforma y requieren la instalación del complemento en los computadores del usuario.

Los plugins más utilizados en la actualidad dentro de los ambientes web se muestran en la Tabla 1-3.

	<b>Complemento</b>	<b>Plataformas</b>
	Adobe Flash	<ul style="list-style-type: none"> <li>• Windows</li> <li>• Mac OS</li> <li>• Linux</li> </ul>
	Microsoft Silverlight	<ul style="list-style-type: none"> <li>• Windows</li> <li>• Mac OS</li> </ul>
	Oracle Java (Applets y JavaFX)	<ul style="list-style-type: none"> <li>• Windows</li> <li>• Mac OS</li> <li>• Linux</li> </ul>

**Tabla 1-3 Complementos para la navegación**  
**Realizado por: Carlos Fabara**

### 1.1.2.2. Backend

#### 1.1.2.2.1. Servidores Web

Los servidores web son aplicaciones que permiten distribuir el contenido web hacia los navegadores de los usuarios. Los servidores se encuentran en espera de peticiones realizadas por los usuarios y cuando una petición HTTP llega, la analizan y devuelven el contenido de acuerdo a lo solicitado dentro de la petición. Actualmente los principales servidores web son:

- Apache
- Internet Information Services
- Nginx
- GWS (Google Web Server)

Adicionalmente existen aplicativos en la tecnología Java denominados Servidores de Aplicaciones, estos pueden funcionar como servidores web, aunque su principal función sea gestionar componentes y lógica de negocio.

Entre los principales servidores web encontramos a:

- Glassfish
- WebLogic
- JBoss
- WebSphere

Muchos de estos servidores web permiten implementar sistemas de alta disponibilidad y balanceo de carga, útil en escenarios de alta demanda.

#### *1.1.2.2.2. Bases de Datos*

Una base de datos es un repositorio de información, almacena la información de múltiples maneras, la más difundida en la actualidad es mediante tablas y relaciones entre las tablas (modelo relacional). Son piezas claves de la denominada web de contenido dinámico. Entre los principales gestores de bases de datos se tienen a:

- MySQL
- PostgreSQL
- SQL Server
- Oracle

En los últimos años han surgido como alternativas a las bases de datos relacionales denominadas bases NoSQL, ya que no utilizan el lenguaje de consulta SQL y su almacenamiento se basa en estructuras de datos. Estas brindan un mayor rendimiento, al no garantizar las propiedades ACID. Su uso es recomendado en entornos donde premie la velocidad de procesamiento antes que la integridad de los datos. Un ejemplo de la utilización de este tipo de bases es dentro de la red social Facebook.

#### *1.1.2.2.3. Lenguajes de programación*

Los lenguajes de programación permiten abstraer a los programadores de la lógica de máquina necesaria para el funcionamiento de un computador. Mediante el uso de un lenguaje de programación se puede codificar la lógica de la aplicación web, que en la mayoría de los casos puede incluir conexión a repositorios de información o sistemas externos. A continuación se listan los principales lenguajes de programación con los que se implementan las aplicaciones de servidor.

- C#, VB.net
- PHP
- Java
- Python
- Ruby

#### *1.1.2.2.4. Marcos de trabajo o Frameworks*

Los marcos de trabajo o frameworks liberan el verdadero potencial de un lenguaje de programación, así también como incrementan la productividad de los desarrolladores al facilitar o automatizar tareas complejas, brindando herramientas y librerías para este fin.

Los marcos de trabajo ofrecen un conjunto de lineamientos o mejores prácticas para aprovechar de una manera más eficiente la base tecnológica sobre la que se cimientan.

Entre los marcos de trabajo más utilizados se encuentran:

- ASP.Net, ASP.Net MVC (C#, VB.net)
- Symfony, CakePHP, etc. (PHP)
- Java EE, Spring MVC, etc. (Java)
- Django (Python)
- Ruby on Rails (Ruby)



### **1.1.3. METODOLOGÍAS**

#### **1.1.3.1. Enfoque Tradicional**

En este enfoque se busca organizar el desarrollo de software en un conjunto de procesos que pueden ejecutarse concurrente o secuencialmente según sea el caso. Estas metodologías promulgan:

- Alcance bien definido
- Requerimientos altamente detallados
- Diseño y arquitecturas de alta granularidad.
- Énfasis en la documentación y diagramas.

Los principales inconvenientes que se tienen al trabajar sobre estas metodologías es que:

- Existe una escasa interacción con el cliente, salvo en las etapas iniciales del proyecto.
- No se encuentran preparadas para adaptarse al cambio.

#### **1.1.3.2. Enfoque Ágil**

Los principios en los que se basan las metodologías ágiles son los siguientes:

- Se da mayor importancia a los individuos y su interacción, por encima de los procesos y las herramientas.
- Se da mayor importancia a que el software funcione antes que a la generación de documentación exhaustiva.
- Se promueve la colaboración con el cliente, por encima de la negociación contractual.
- Se prefiere la respuesta al cambio, por encima del seguimiento de un plan.

Estos cuatro principios fueron promulgados en el Manifiesto Ágil, redactado por los principales exponentes del movimiento ágil entre ellos Kent Beck, en este manifiesto se trató reflejar una nueva forma de hacer software. [4]

El término ágil se ha malinterpretado como una excusa para no documentar durante el proceso de desarrollo, sin embargo un proceso ágil no representa cero documentación, sino que la documentación no sea excesiva, pero que cumpla con su cometido de representar el sistema que se construirá.

### **1.1.3.3. Enfoque Web**

El desarrollo de aplicaciones web requiere considerar aspectos que no son tratados en la mayoría de procesos y metodologías tradicionales. Por este motivo han surgido un conjunto de métodos y procesos que buscan solventar estas falencias. Estas metodologías se adhieren al conjunto de reglas y convenciones del mundo web, con lo que proveen un marco de trabajo estable, robusto y escalable para el desarrollo y ejecución de aplicaciones basadas en la web. [5]

Los procesos específicos para la web se centran en:

- Caracterización del grupo de Usuarios
- Diseño de Navegabilidad
- Diseño de Interfaces
- Diseño de Contenidos
- Dirigidos por Datos
- Prototipos y evolución continua
- Despliegue rápido
- Adaptabilidad ante requerimientos cambiantes

Además surgen nuevos actores dentro del proceso de desarrollo como son:

- Diseñador Gráfico
- Webmaster

## **1.2. DETERMINACIÓN DE LAS VENTAJAS Y DESVENTAJAS DE LAS PRINCIPALES METODOLOGÍAS UTILIZADAS EN EL DESARROLLO WEB**

Para definir el proceso que se busca proponer, es necesario investigar acerca de las principales propuestas de metodologías aplicables al desarrollo de sistemas web. Dentro de este contexto se identificaron que los representantes más importantes en este sentido son [6] :

- WebML (Web Modeling Language)
- WSDM (Web Semantic Design Methods)
- OOHDM (Object Oriented Hypermedia Design Method)

Adicionalmente se desea complementar el proceso con aspectos puntuales y recomendaciones de las principales metodologías, entre ellas algunas de las que se espera utilizar son:

- Scrum
- RUP, OpenUP
- Extreme Programming - XP
- ICONIX

### **1.2.1. WEBML**

WebML (Web Modeling Language) es un lenguaje de modelado que propone un proceso de desarrollo del mismo nombre. Es un proceso de desarrollo dirigido por modelos que describe la estructura del contenido y el funcionamiento de la aplicación a través de modelos WebML. Es un proceso específico para aplicativos web que se conecten hacia una base de datos. [6]

Algunos roles característicos que se encuentran en WebML son:

- **Experto en Datos.-** Crea la estructura de datos que utilizará el sistema.
- **Arquitecto de la Aplicación.-** Diseña las páginas del sistema y la navegación entre ellas.
- **Arquitecto de Estilos.-** Se encarga de aplicar estilos gráficos y la presentación de las páginas.
- **Administrador del Sitio.-** Diseña las reglas del negocio, los grupos de usuarios y las opciones del sistema.

Fases:

- Análisis de Requerimientos
- Diseño de la Aplicación
- Implementación
- Pruebas y Evaluación
- Despliegue, mantenimiento y evolución

### Análisis de Requerimientos

En este proceso no se especifica un mecanismo para recoger requerimientos, pero recomienda el uso de tablas para escenarios informales y casos de uso junto con diagramas de actividad que describan el proceso del negocio, para escenarios que necesiten requerimientos mayormente detallados. En esta etapa se busca lograr lo siguiente:

- Identificación de los grupos de usuarios y objetos de información base.
- Especificación de requerimientos funcionales.
- Descomposición de la aplicación en vistas del sitio (site views), que son el conjunto de páginas por las que el usuario podrá navegar.

## Diseño de la Aplicación

El diseño de la aplicación se lo realiza a través de esquemas conceptuales basados en WebML, que permiten abstraer el sistema y representarlo independiente de la tecnología de implementación. Se identifican dos actividades en esta fase que son:

- **Diseño de Datos.-** Consiste en organizar los objetos básicos de información identificados durante la fase anterior en un esquema de datos coherente y completo. En general el diseño de datos permite generar el modelo entidad - relación de la base de datos.
- **Diseño de Hipertexto.-** Uno de los componentes claves de WebML es el modelado conceptual de las distintas vistas del sitio y su interacción, basándose en el modelo de datos definido anteriormente.

Por ejemplo en un sistema de información discográfica, los objetos de información base son los artistas, sus álbumes y las pistas de cada uno de ellos. Las vistas del sitio que se desea crear son Página principal, Menú de artistas, Bibliografía, Álbumes por artista y Pistas por álbum. El modelo de datos para este tipo de sistema se muestra en la Figura 1-2.

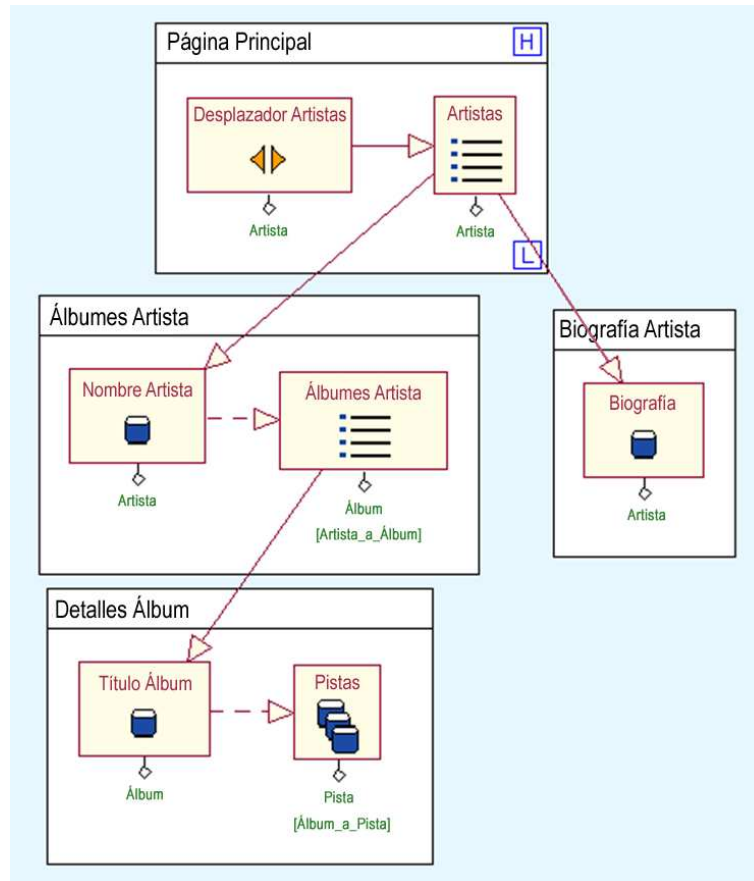


**Figura 1-2 Modelo de Datos de Información Discográfica**

Fuente: <http://www.webml.org>

Traducido y Adaptado por: Carlos Fabara

El modelo de hipertexto resultante será el mostrado en la Figura 1-3; en este diagrama se combinan los conceptos definidos en las vistas del sitio y el modelo de datos. Como se observa, se definió la Página Principal donde se listan los artistas y desde ahí se puede navegar ya sea hacia la biografía del artista o a la información de sus álbumes y pistas.



**Figura 1-3 Modelo de Hipertexto para un Sistema de Información Discográfica**

**Fuente:** <http://www.webml.org>

**Traducido por:** Carlos Fabara

### Implementación

En esta etapa se procederá a la construcción del sistema en base a los modelos generados durante la etapa de diseño.

En este punto WebML propone la utilización de herramientas de generación de código que permitan transformar modelos en código funcional. Existe un entorno de modelado y desarrollo propuesto por el proceso llamado WebRatio, el cual permite generar gran parte de la aplicación web, así como su base de datos, basándose en los modelos generados durante la fase anterior.

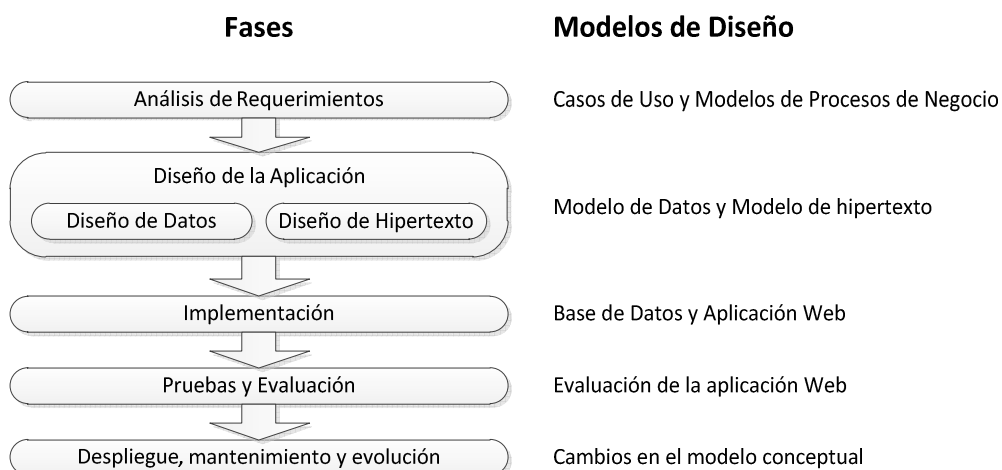
## Pruebas y Evaluación

Los modelos conceptuales de la aplicación permiten facilitar las pruebas, ya que si el modelo se encuentra correcto y describe el funcionamiento de la aplicación, la única preocupación consiste en asegurarse que la herramienta de generación de código interprete los modelos y trabaje de manera adecuada. Además WebML propone nuevas técnicas para el aseguramiento de la calidad que se agruparon en un framework, que posee técnicas de análisis estático (modelos) y dinámico (métricas).

## Mantenimiento y Evolución

WebML al ser un proceso dirigido por modelos permite que cualquier cambio a nivel de requerimientos pueda ser representado fácilmente en los modelos conceptuales que propone, de esta manera cualquier cambio realizado en el modelo se reflejará a nivel de código.

En la Figura 1-4 se resumen las distintas fases y los resultados que se producen en cada una de ellas. Al ser un proceso iterativo, estas fases se repiten en cada uno de las iteraciones o ciclos de diseño. [7]



**Figura 1-4 Diagrama de fases y modelos de WebML**  
**Fuente: [6] Sven Casteleyn, *Engineering Web Applications*.**  
**Traducido y Adaptado por: Carlos Fabara**

### **1.2.1.1. Ventajas**

- Toma en cuenta la navegación y los datos como piezas fundamentales de un sistema web.
- El modelo de hipertexto engloba en un solo diagrama a las vistas, los datos y la navegabilidad.
- Considera dentro del proceso al despliegue, mantenimiento y evolución.
- Incluye muchos aspectos concernientes al ciclo de vida del desarrollo de software.
- Puede utilizar casos de uso para la especificación de requisitos complejos.

### **1.2.1.2. Desventajas**

- WebML al no ser un estándar, no es un lenguaje de modelado muy difundido.
- Escasas herramientas de modelado, WebRatio es la más popular y es un producto de pago.
- Actualmente WebRatio solo puede realizar la implementación del sistema en Java, por lo que no se puede aprovechar todo el potencial de WebML con otras tecnologías.

## **1.2.2. WSDM**

WSDM es el acrónimo para Web Site Design Method (Método para el diseño de sitios web) fue propuesto en el año de 1998 y es considerado uno de los primeros modelos específicos para la web. Esto se evidencia en su nombre ya que en ese entonces aún no existía la separación clara entre el concepto de sitio web y sistema web. Por este motivo posteriormente fue rebautizado como Web Semantics Design Methods (Métodos para el diseño de webs semánticas). [6]

Es una metodología que ha evolucionado constantemente hasta alcanzar un modelo completo de diseño de aplicativos web, que abarca desde el diseño de la funcionalidad hasta aquellos aspectos esenciales para el desarrollo web, que no son considerados en las metodologías tradicionales.



Las características principales que posee son:

- Es una metodología, por lo que adicionalmente al proceso define un conjunto de guías y modelos que permiten una correcta representación del sistema a construir.
- Es dirigido por la audiencia, por lo cual se enfoca en satisfacer las necesidades del usuario final,
- Utiliza técnicas que permite el desarrollo de aplicaciones web con un alto contenido semántico.

Las fases de las que se compone WSDM son:

- Definición de la Misión
- Modelado de la Audiencia
- Diseño Conceptual
- Diseño de la Implementación
- Implementación

A continuación se explicará en que consiste cada una de las fases.

### Definición de la Misión

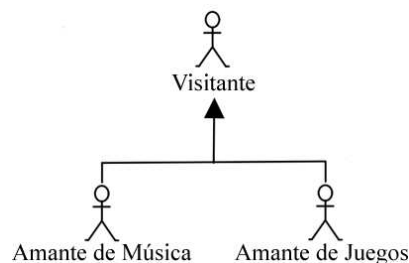
En esta fase se busca definir a la aplicación web en sí. Se analizan y documentan aspectos como la finalidad del sistema, la funcionalidad deseada, se analizan rápidamente los posibles grupos de usuarios y se especifican los límites de la aplicación. Se podría decir de forma concisa que en esta etapa se realiza la recopilación de requerimientos, así también como la especificación del alcance que tendrá el proyecto; cabe recalcar que WSDM no propone un mecanismo para la realización de esta etapa, pero si recomienda que la documentación de esta fase se la realice con lenguaje natural. Como ejemplo de WSDM se modelará un sitio que contiene información y servicios para entusiastas del cine y los videojuegos.

## Modelado de la Audiencia

En esta etapa se refinarán los grupos de usuario que se identificaron en la fase anterior, esto se realiza debido a que no todos los usuarios que acceden a la aplicación poseen las mismas necesidades e intereses.

El modelado de la audiencia consta de dos sub-fases:

- Clasificación de la Audiencia.- Consiste en agrupar a los distintos usuarios en "Clases de Audiencia", las cuales son grupos de usuario que comparten las mismas necesidades a nivel de información y de funcionalidad. En la Figura 1-5 se puede observar un ejemplo del modelamiento de clases de audiencia.
- Caracterización de la Audiencia.- En esta etapa se definen las características a nivel funcional, de usabilidad, navegabilidad, etc., que deberán cumplirse para satisfacer las necesidades de cada una de las Clases de Audiencia.



**Figura 1-5 Representación de una clase y subclases de audiencia**  
**Fuente: [8] Gustavo Rossi, *Web Engineering: Modelling and Implementing Web Applications*.**

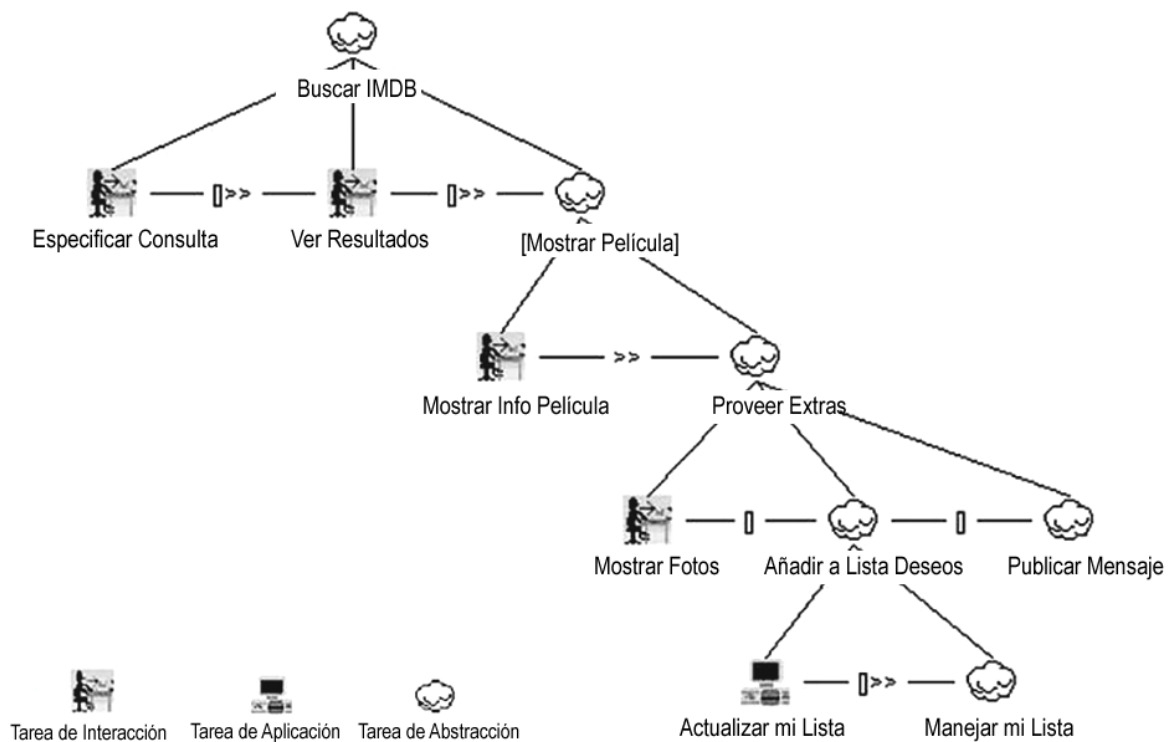
**Traducido por: Carlos Fabara**

## Diseño Conceptual

Esta etapa se divide en:

- Modelado de Tareas e Información
- Diseño de la Navegación

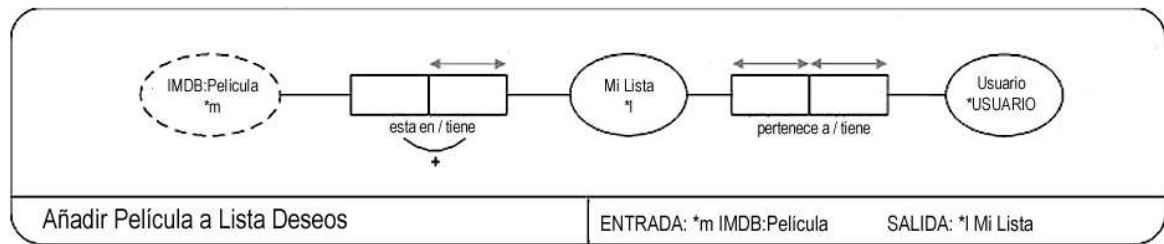
**Modelado de Tareas e Información.-** Consiste en modelar cada uno de los requerimientos y características funcionales definidos en la etapa anterior. Para realizar este modelamiento se utilizan los diagramas Arboles de Tareas Concurrentes CCT, un ejemplo de éste se muestra en la Figura 1-6, en este tipo de diagrama se descompone una tarea principal en subtareas, denominadas tareas elementales que se requieren para completarla y se representa su secuencia y relación en el tiempo.



**Figura 1-6 Diagrama CTT para la tarea de buscar información**  
**Fuente: [8] Gustavo Rossi, *Web Engineering: Modelling and Implementing Web Applications*.**

**Traducido por: Carlos Fabara**

Para cada tarea elemental detallada se deben desarrollar los denominados “object chunk” que se pueden traducir como un pedazo o trozo de objeto. Este pedazo describe que información y/o funcionalidad se necesita para completar la tarea elemental que representa. Para su modelado se utiliza OWL (Ontology Web Language), un ejemplo de un pedazo de objeto para la tarea de añadir una película a la lista del usuario se observa en la Figura 1-7.

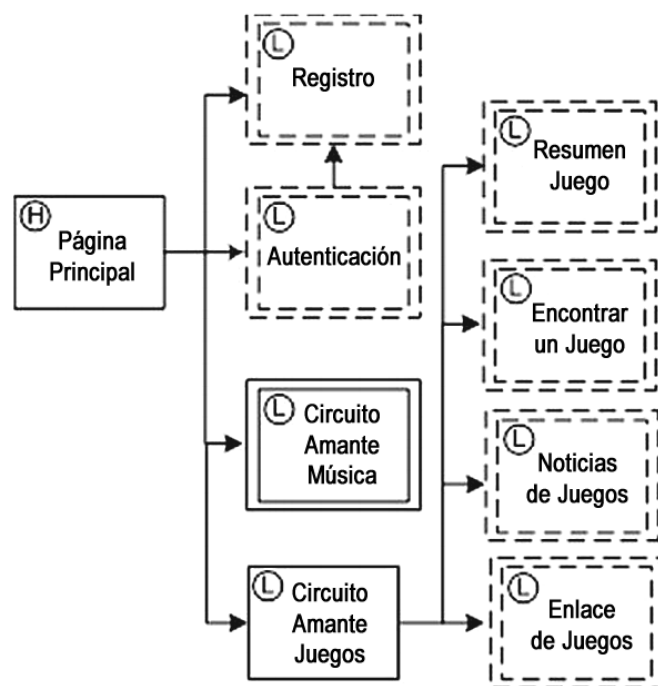


**Figura 1-7 Ejemplo de un "object chunk"**

**Fuente:** [8] Gustavo Rossi, *Web Engineering: Modelling and Implementing Web Applications*.

**Traducido por:** Carlos Fabara

**Diseño de la navegación.-** Se lo realiza en dos niveles: externo e interno. En el diseño externo se toma en cuenta la estructura del sitio partiendo de una página principal y a partir de esta se modelan los distintos caminos o flujos que seguirían los grupos de usuarios. El diseño de la navegación interna es a nivel de tarea y especifica el flujo de actividades definidas en el CTT y su relación con los pedazos de objeto definidos. Un ejemplo del diagrama producto del diseño externo se lo observa en Figura 1-8



**Figura 1-8 Ejemplo de diagrama de navegación externo**

**Fuente:** [8] Gustavo Rossi, *Web Engineering: Modelling and Implementing Web Applications*.

**Traducido por:** Carlos Fabara

## Diseño de la Implementación

En esta etapa se complementan los diagramas anteriores con todos los elementos necesarios para que con el uso de una herramienta se pueda realizar la generación automática de la implementación.

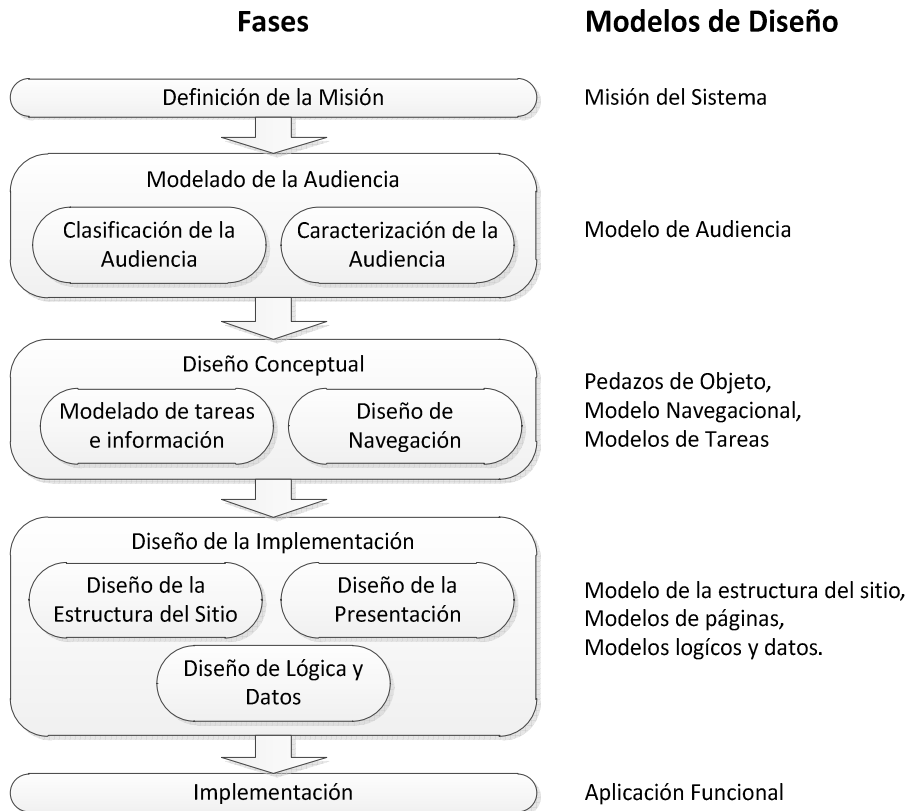
Consta de 3 subfases:

- **Diseño de la estructura del sitio.-** En esta etapa se mapean el diseño navegacional a las páginas del sitio. La estructura del sitio podrá variar de acuerdo a factores como el terminal de acceso, el contexto de uso y la plataforma sobre la que se implementará.
- **Diseño de la presentación.-** Consiste en que el diseñador organice en las páginas web los controles, su presentación, la estructura del contenido de forma que la interacción y usabilidad del sitio sea óptima. El resultado de esta fase son las plantillas de las páginas web.
- **Diseño de lógica y datos.-** Durante esta etapa se realiza un mapeo de los datos que se manejan en los pedazos de objeto, hacia entidades de un repositorio de información. Así también de la definición del mecanismo de implementación de las reglas del negocio. Este diseño se lo debe puede representar en un diagrama entidad relación, modelo RDF, XML u OWL.

## Implementación

Al tener los distintos modelos definidos, así también como el modelo de los datos, la implementación del sistema se la puede realizar automáticamente a través de herramientas de generación de código. Herramientas como Visual Wade pueden generar código a partir de modelos realizados en lenguajes estándar como RDF, OWL o XML.

Las distintas fases de WSDM junto con los productos de cada una de ellas se listan en la Figura 1-9



**Figura 1-9 Diagrama de fases y modelos de WSDM**  
**Fuente: [6] Sven Casteleyn, *Engineering Web Applications*.**  
**Traducido y Adaptado por: Carlos Fabara**

### 1.2.2.1. Ventajas

- El definir la misión o alcance claramente permite evitar inconsistencias durante el proceso de desarrollo.
- Al ser un proceso dirigido por la audiencia se enfoca en cumplir las necesidades de los distintos grupos de usuarios.
- WSDM utiliza OWL y RDF las cuales son recomendaciones del W3C (World Wide Web Consortium).
- Permite separar el diseño conceptual del diseño de implementación, ya que en la práctica pueden existir muchos escenarios de implementación para un mismo modelo conceptual.

### 1.2.2.2. Desventajas

- La gran cantidad de modelos necesarios para definir un sistema puede llegar a ser abrumadora aún más si se considera que los lenguajes de modelado que se utilizan son muy poco conocidos.
- Escasa documentación del proceso.
- Falta de apoyo de parte de un grupo u organización.
- No considera a las Pruebas dentro del proceso de desarrollo.
- No existe una herramienta específica o recomendada para realizar la implementación.

### 1.2.3. OOHDM

OOHDM (Object Oriented Hypermedia Design Method) es un Método de Diseño de Hipermedia Orientado a Objetos. Es uno de los primeros métodos creados para el desarrollo de sistemas web. Es un método dirigido por modelos, por lo que realiza abstracciones basadas en objetos para el análisis y diseño de aplicaciones que hacen uso intensivo de datos. Además provee una metodología que sirve como guía durante el proceso de desarrollo. [6]

Las características principales de este proceso son:

- **Vistas de Navegación.-** Especifican como los objetos de información deben agruparse cuando son explorados por un usuario en una sesión de navegación
- **Contextos de Navegación.-** Son abstracciones de agrupamiento para organizar el espacio de navegación.
- **Separación de Intereses.-** Esta separación permite que los aspectos conceptuales, de navegación y de presentación se encuentren completamente separados entre sí. Un lenguaje de consulta se utiliza para conectar los diferentes modelos y puntos de vista.

Es un proceso compuesto por cinco etapas:

- Análisis de Requerimientos
- Diseño Conceptual
- Diseño Navegacional
- Diseño de Interfaces Abstractas
- Implementación

### Análisis de Requerimientos

La meta principal de esta fase es la de entender y capturar los requerimientos funcionales y no funcionales de la aplicación, que se representan a través de Casos de Uso, los cuales son refinados hasta obtener Escenarios que reflejen las tareas que el usuario debe realizar. En esta fase también se producen los denominados Diagramas de Interacción de Usuario, los cuales capturan como el usuario debe interactuar con la aplicación para completar ciertos casos de uso.

### Diseño Conceptual

Esta etapa se enfoca hacia el diseño de la estructura de la información para representar el contenido del sistema. En este punto no se consideran a los usuarios ni a sus tareas, sino únicamente al dominio de la aplicación. El producto de esta fase es un diagrama de clases UML.

### Diseño Navegacional

Durante esta etapa se definen las estructuras de navegación necesarias para que el usuario pueda explorar la información que provee el sistema. El resultado del diseño produce vistas de las estructuras de información del sistema, las mismas que pueden variar dependiendo de la audiencia a la que se enfocan, esto se debe a que más de una vista se puede crear para un mismo modelo conceptual.



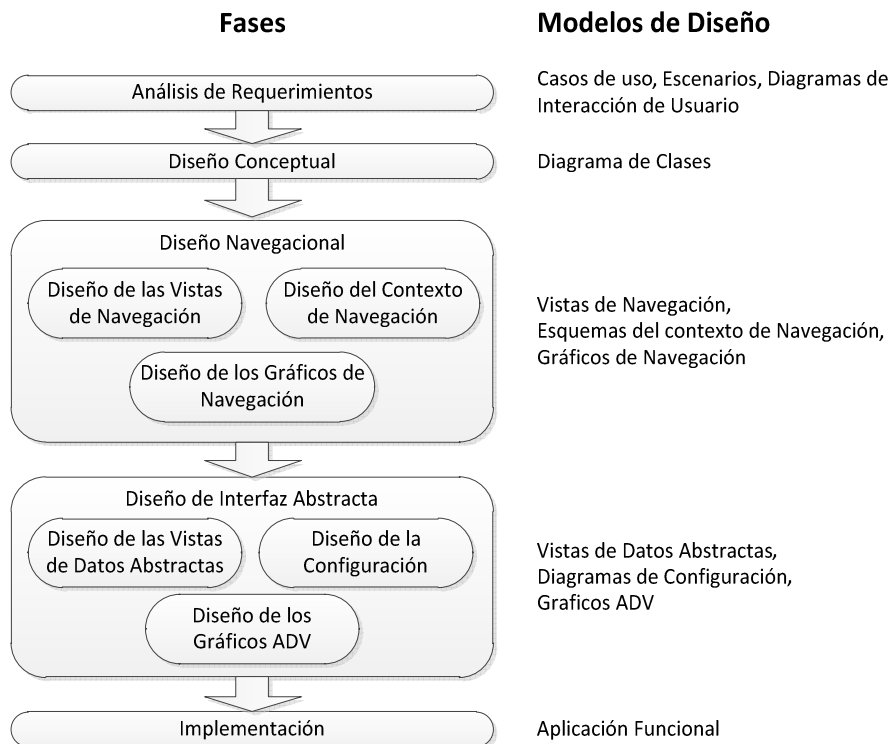
## Diseño de Interfaz Abstracta

Se centra en los objetos que pueden ser percibidos por el usuario, definiendo como las vistas de navegación se van a mostrar y como pueden ser complementadas a través de la adición de elementos de interacción como botones, listas desplegables, enlaces, etc.

## Implementación

OOHDM no utiliza un framework en particular para la realizar la implementación del sistema. Depende del equipo desarrollador escoger y planificar el mejor mecanismo para realizar la implementación de todo lo definido anteriormente. Deben definir la arquitectura, tecnologías de implementación, modelos de programación, herramientas, etc.

Un cuadro resumen de las fases y productos de OOHDM se observa en la Figura 1-10.



**Figura 1-10 Diagrama de fases y modelos de OOHDM**  
**Fuente: [6] Sven Casteleyn, *Engineering Web Applications*.**  
**Traducido y Adaptado por: Carlos Fabara**

### 1.2.3.1. Ventajas

- Es dirigido por casos de uso.
- Sus diagramas son realizados en UML.
- Es una metodología Orientada a Objetos.
- Es también una de las metodologías web más utilizada y difundida.

### 1.2.3.2. Desventajas

- Excesivo enfoque en la navegación y el diseño abstracto de la interfaz.
- No existe una herramienta que automatice la implementación.

## 1.2.4. SCRUM

“Scrum es una metodología para la gestión y desarrollo de software basada en un proceso iterativo e incremental utilizado comúnmente en entornos basados en el desarrollo ágil de software.” Tomado de [9].

Scrum es una de las metodologías ágiles más populares del momento, se caracteriza por su simplicidad, seguimiento continuo y efectividad, motivo por el cual es utilizada por algunas de las empresas más representativas de la industria como: Google, Sun, Microsoft, Siemens, entre otros [10].

Los documentos de las metodologías tradicionales son denominados artefactos en la metodología Scrum. A continuación se listan los principales artefactos junto con una breve explicación de su función:

- **Bitácora del Producto (Product Backlog).**- Documento en el que se agrupan el conjunto de requerimientos y características que deberá cumplir el producto final. Posteriormente se realiza la priorización de estos elementos de forma que sirva de guía para la creación de los sprints.

- **Bitácora del Sprint (Sprint Backlog).**- Documento en el que se agrupan los elementos o tareas de la Bitácora del Producto que se realizarán en un Sprint determinado.
- **Gráfico Burndown (Burndown Chart).**- Gráfico que representa de forma de una pendiente descendente la cantidad de trabajo restante, además su análisis permite anticipar fechas tentativas de la finalización del proyecto.

Algunos de los artefactos opcionales que se pueden utilizar dentro de Scrum son:

- **Bitácora de Errores (Bug Backlog).**- En esta bitácora se registran los errores que se encontraron de forma que se tenga un registro de lo que se debe corregir, dependiendo de su severidad se programa su resolución.
- **Bitácora de la Liberación (Release Backlog).**- En el caso de que el producto requiera de mucho trabajo para llegar a una liberación se seleccionan las características de la Bitácora del Producto que conformarán la liberación y posteriormente en base a ésta se crearán las distintas Bitácoras de Sprint.

Roles de Scrum:

- **Dueño del Producto (Product Owner).**- Se asegura de seleccionar los aspectos y características que conformaran la bitácora del producto y que debería contener el producto final. Es el encargado de representar los intereses de los clientes y los usuarios finales.
- **Maestro Scrum (Scrum Master).**- Analiza que el proyecto avance al ritmo adecuado y que los miembros del equipo dispongan de la infraestructura y herramientas necesarias para realizar su trabajo. Se encarga de evaluar los progresos, organizar reuniones, etc. Tiene características de lo que en otras metodologías se considera un Gerente de Proyecto.
- **Equipo.**- Son aquellos que construyen y prueban el producto (Desarrolladores, Diseñadores, Probadores, etc.).
- **Clientes (Customers).**- Son los usuarios finales del producto y aquellos que serán afectados directamente por la utilización del mismo.

El flujo de Scrum se lo puede observar representado en la Figura 1-11 y se lo describe a continuación: el dueño del producto crea la bitácora del producto, donde se especifican las características que debe cumplir el producto, de éstas se seleccionan las más importantes y se crea una bitácora de sprint. Posteriormente se empieza el proceso de iteración, durante este tiempo deben existir reuniones diarias denominadas “Scrum Meeting”, estas reuniones tienen una duración máxima de 15 minutos y se las debe realizar de pie, en estas reuniones los miembros del equipo deben responder 3 preguntas:

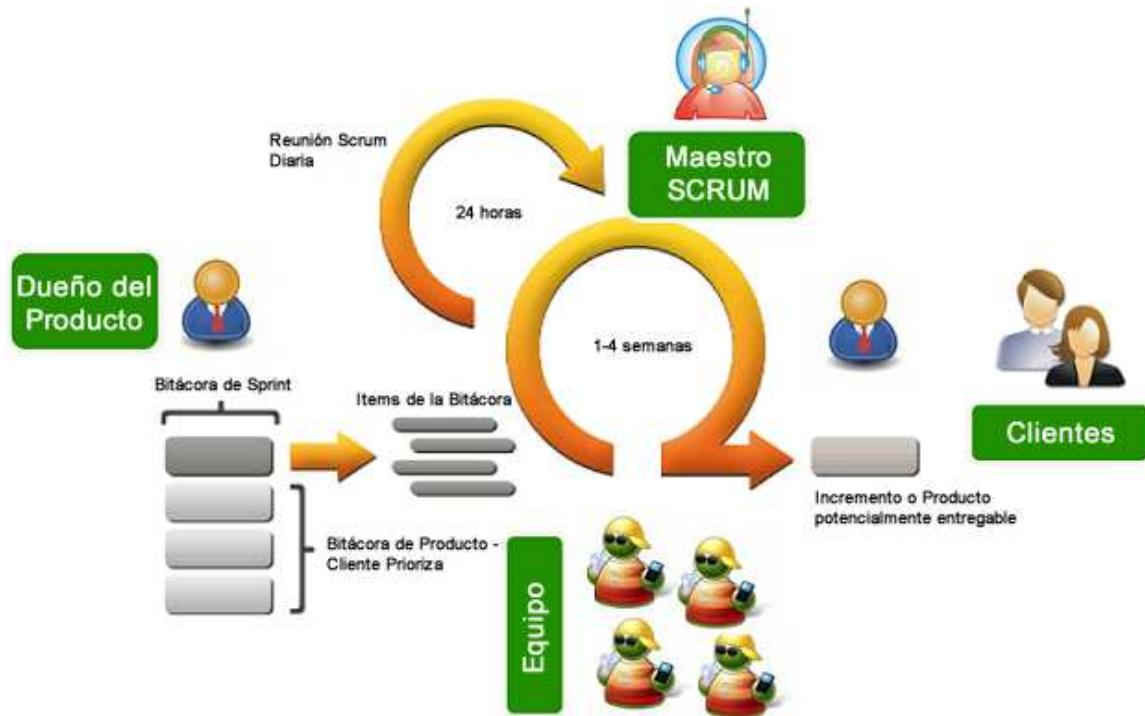
- ¿Qué hice ayer?
- ¿Qué voy a hacer hoy?
- ¿Existe algún impedimento para realizar mi trabajo?

A través de estas reuniones se logra el seguimiento continuo y se puede identificar posibles problemas y solucionarlos a tiempo. Tras un periodo de entre 1 y 4 semanas que dura el Sprint se obtiene un producto funcional que puede ser potencialmente entregado. El ciclo se repite hasta que se hayan terminado de implementar todas las características de la bitácora del producto.

Cabe recalcar que Scrum es una metodología que se orienta más hacia el manejo de proyectos en general, y que no es exclusivo para el manejo de proyectos de desarrollo de software.

#### **1.2.4.1. Ventajas**

- Es una metodología muy difundida y utilizada por compañías importantes.
- Tiene roles y artefactos bien definidos.
- Promueve las entregas continuas.
- Se centra en la simplicidad e interacción entre los miembros del equipo.
- Cumple con los principios del manifiesto ágil.



**Figura 1-11 Fases de SCRUM**

Fuente: <http://santimacnet.wordpress.com/category/metodologias-agiles/>

Traducido por: Carlos Fabara

#### 1.2.4.2. Desventajas

- No tiene elementos que sean específicos para un desarrollo web.
- No define fases similares a las otras metodologías en lo que respecta al desarrollo de software.

### 1.3. DETERMINACIÓN DE LOS ASPECTOS ADICIONALES APLICABLES AL DESARROLLO WEB

#### 1.3.1. DISEÑO GRÁFICO PARA LA WEB

“El diseño gráfico de una página web es tan solo una parte del diseño de la misma, ya que, además, hay que considerar un conjunto más o menos extenso de condicionantes que van a limitar la libre creatividad del diseñador.” Tomado de [11]

El diseño gráfico web se enfoca en los aspectos visuales de la página, busca lograr un acoplamiento de colores, imágenes, texturas, patrones, tipografías, entre otros; de forma que se logre una armonía visual que sea agradable hacia el usuario e incremente la usabilidad del sitio.

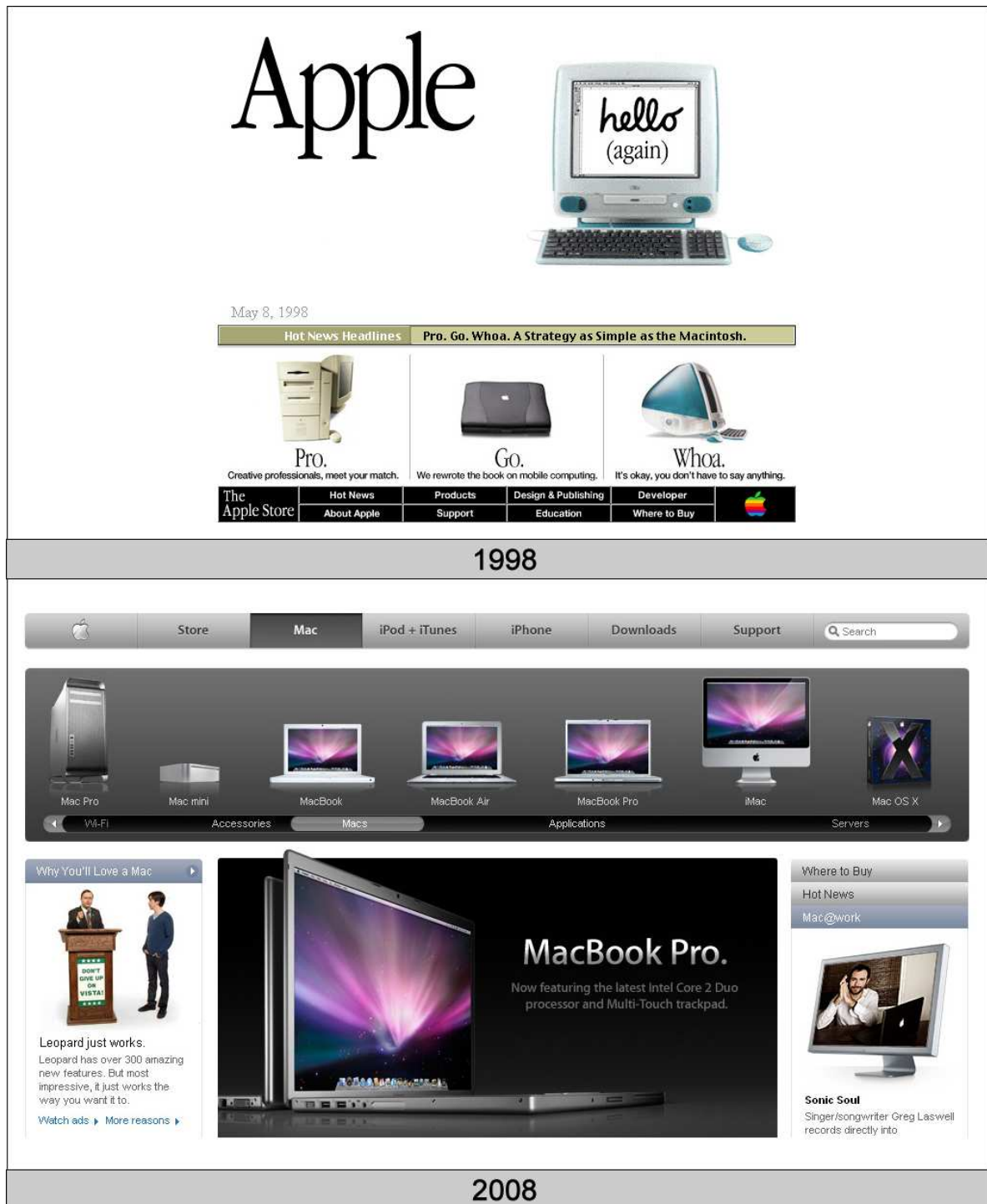
El uso del diseño gráfico en las interfaces web se ha incrementado y ha evolucionado a través del tiempo como se observa en la Figura 1-12; la aparición de nuevas herramientas (CSS, Javascript, Flash, etc.) permitió que las páginas web pasen de ser simples repositorios de contenido a ser cada vez más complejas, asemejándose a una aplicación de escritorio y teniendo el mismo cuidado a nivel de diseño que el que se tiene en una publicación impresa.

Existen gran cantidad de prácticas y aspectos concernientes al diseño gráfico que se pueden aplicar en el desarrollo web. Entre los más relevantes se encuentran:

**Uso del espacio.-** El espaciamiento entre los elementos de una página mejora la legibilidad de los párrafos, elimina la sensación de saturación del sitio o resalta la separación entre los elementos. Los tipos de espaciado más comunes que se utilizan en una página web son:

- Espacio entre líneas.
- Relleno (Padding).
- Espacio en blanco o vacío.

**Alineación.-** “Mantener los contenidos alineados es igual de importante en el diseño web como lo es en el diseño impreso. Esto no quiere decir que todo debe ubicarse en una línea recta, sino que se debe buscar mantener los ítems ubicados de forma consistente. La alineación de elementos hace que su diseño se vea más ordenado y se asimile más fácilmente, además de mostrar una apariencia más elegante” [12].



**Figura 1-12 Ejemplo de evolución del diseño web (Apple)**

**Fuente:** <http://www.wakeuplater.com/website-building/evolution-of-websites-10-popular-websites.aspx>

**Realizado por:** Carlos Fabara

**Tipografía.-** Los colores, tamaños y fuentes del texto pueden variar según el contexto en el que se usen (títulos, párrafos, citas, notas), pero es importante que el texto contraste del fondo, de forma que facilite la lectura. Adicionalmente, el

interlineado del texto debe ser el adecuado de forma que la página no se vea sobrecargada de texto y sea más legible. En lo que respecta a la tipografía, el uso de fuentes específicas puede causar problemas de compatibilidad en algunos equipos, si estos no tienen instalada dicha fuente, por esto se recomienda utilizar en lo posible fuentes estándar.

**Consistencia.-** “La consistencia significa que todo se acople. Encabezados, fuentes, colores, estilos, espaciado, elementos de diseño, ilustraciones, fotografías, etc. Todo debe tener un tema común de forma que el diseño sea consistente entre las páginas del sitio y el contenido de cada una de ellas.” [12].

**Diseñar para ser Construido.-** Un claro problema que ocurre al diseñar un sitio web es que, en la mayoría de los casos, el diseño que fue aprobado o que se recibió del diseñador, es de compleja implementación o poco práctico al momento de operar en entornos reales. Se deben diseñar páginas de forma que sean factibles de ser construidas y simplificar el diseño, evitando el uso de elementos que sean irrelevantes o extremadamente complejos. [12].

### 1.3.2. ARQUITECTURA DE INFORMACIÓN

La arquitectura de información se define como [13]:

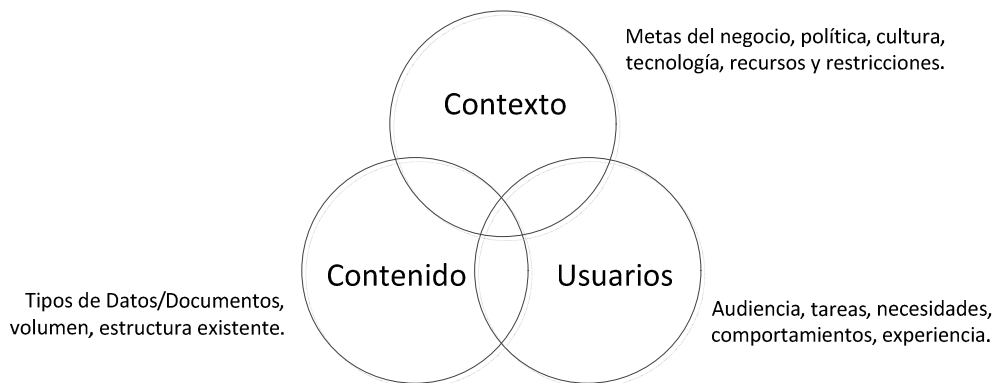
- El diseño estructural de ambientes de información compartida.
- La combinación de los sistemas de organización, etiquetado, búsqueda y navegación en sitios web e intranets.
- El arte y ciencia de concebir productos de información y experiencias que permitan la usabilidad y el uso de la intuición en un sistema web.
- Es una disciplina emergente y un conjunto de prácticas enfocadas en brindar principios de diseño y arquitectura dentro de un panorama digital.

El término información no se refiere únicamente al conocimiento o los datos que va a manejar el sistema. La información puede encontrarse en muchas formas y tamaños (sitios web, imágenes, documentos, etc.), y necesita ser estructurada y



categorizada de forma que se defina su nivel de granularidad y su relación con otros elementos.

La arquitectura de información se fundamenta en el análisis de tres aspectos primordiales, ejemplificados en la Figura 1-13. El correcto balance entre estos aspectos permite generar una estructura adecuada para el volumen de información, que se adapta a las necesidades de los usuarios, sin obviar las metas de la organización y las restricciones existentes.



**Figura 1-13 Los tres círculos de la Arquitectura de Información**  
**Fuente: [13] Peter Morville *Information Architecture for the World Wide Web*.**  
**Traducido por: Carlos Fabara**

Su importancia radica en que permite:

- Disminuir el costo de búsqueda y localización de la información.
- Minimizar las pérdidas al mejorar la satisfacción de los usuarios o clientes.
- Informar a los usuarios o clientes.
- Construir sitios escalables y de fácil mantenimiento.

### 1.3.3. DISEÑO DE INTERFACES WEB

El diseño de interfaces web requiere considerar los diversos factores que confluyen en la creación de una interfaz.

Los aspectos que se deben considerar al construir interfaces web de calidad son:

- **Diseño Gráfico.-** El diseño gráfico permite buscar una armonía visual entre los elementos de interfaz, de manera que sean agradables para el usuario final.
- **Diseño de Interfaces.-** Orienta en la ubicación y uso de controles específicos para las tareas que se requieren, de forma que se mejore la usabilidad del sitio.
- **Arquitectura de Información.-** Permite que se estructure adecuadamente el contenido que se desplegará en el sistema, para que sea más fácil difundir la información.

Lo que se busca a través de una interfaz web es mejorar la experiencia del usuario al utilizar el sistema, creando una interfaz que sea usable y fácilmente comprensible.

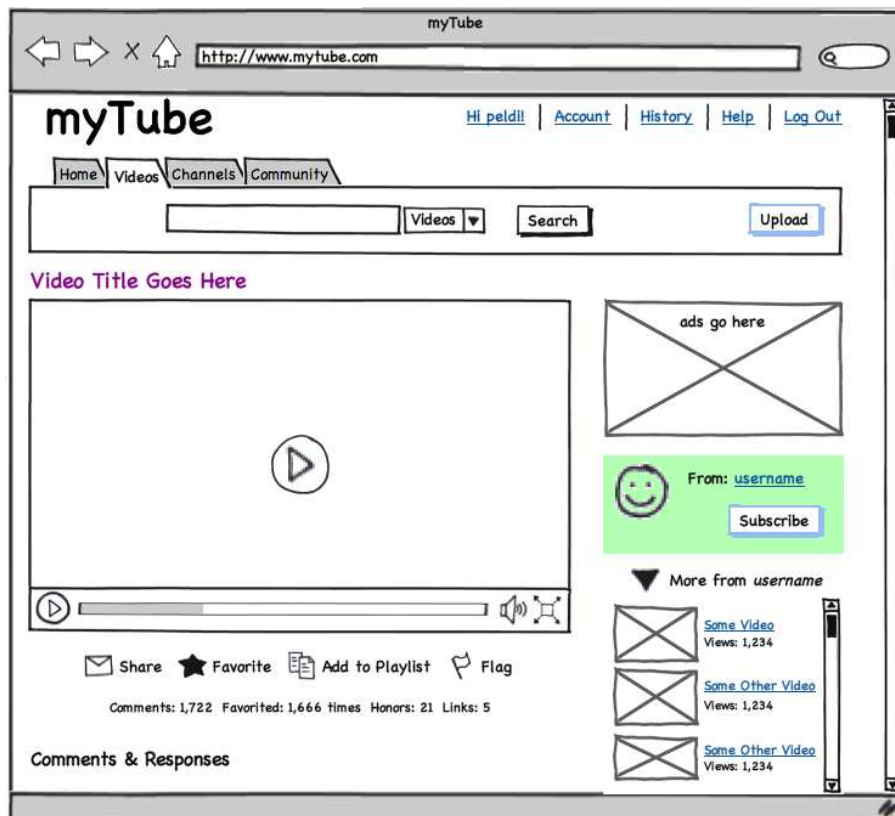
Dependiendo de la finalidad del sistema, existe un conjunto de patrones estándar para el desarrollo de interfaces [14], que, dependiendo de la finalidad del sistema, pueden ser más eficientes: es importante que los diseños se ubiquen dentro de alguno de ellos. En la Figura 1-14 se muestran algunos ejemplos de dichos patrones.

Una práctica común dentro del diseño de interfaces web es el prototipado a través de los denominados mockups [15], que son gráficos minimalistas que dan a conocer a los clientes la estructura que tendrá una página web, sin distraerlos con detalles como los colores e imágenes, tal como se observa en la Figura 1-15.

En etapas posteriores de diseño y tras haber seleccionado una estructura para el sitio, se pueden realizar prototipos que incorporen los elementos visuales y permitan que el cliente se forme una idea del aspecto que tendrá el sistema.



**Figura 1-14 Patrones Estándares de Interacción**  
**Fuente: [14] Theresa Neil. (2009, Jan.)**  
**Traducido por: Carlos Fabara**



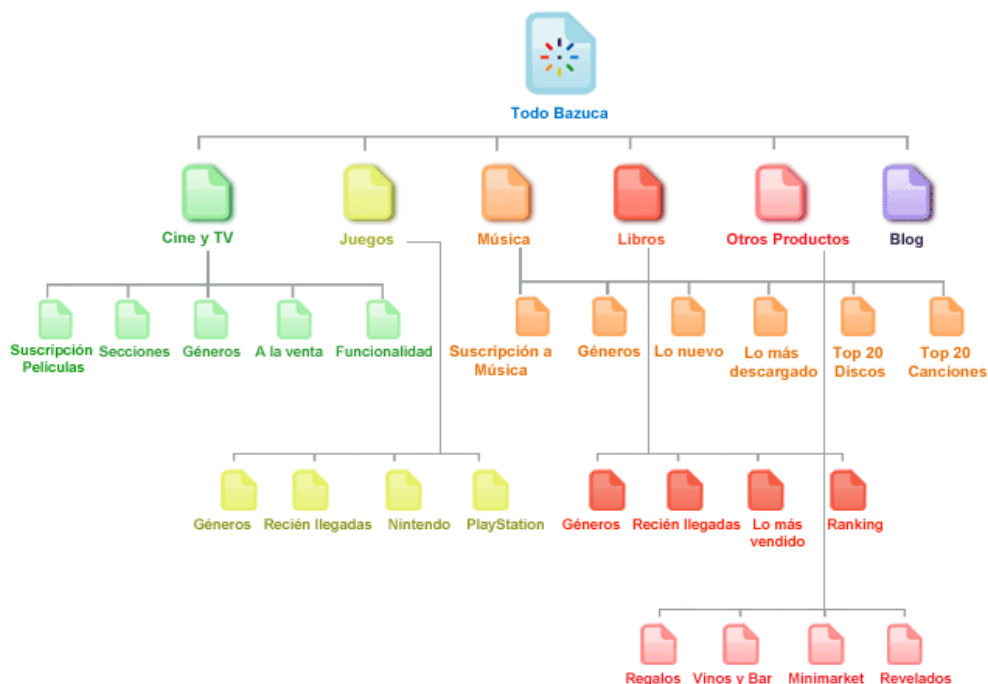
**Figura 1-15 Ejemplo de Wireframe**  
**Fuente: <http://balsamiq.com/products/mockups>**

### 1.3.4. DISEÑO DE LA NAVEGACIÓN

La navegación es una parte fundamental y clave de toda aplicación web, ya que desde sus albores la información se encontraba distribuida en múltiples páginas que se conectaban entre sí a través de hipervínculos; en la actualidad es común que la lógica y/o funcionalidades se encuentren distribuidas en diferentes páginas o contextos (Ej: Páginas de Login, Registro, Consultas, etc.).

Al existir un constante cambio de ubicación lógica, es importante registrar y diseñar correctamente el flujo que seguirá un usuario para realizar determinada tarea. El diseño de navegación busca definir los caminos posibles para navegar entre la información y funcionalidades que el sistema brinda.

Uno de los mecanismos comúnmente utilizados para ejemplificar las opciones de navegación del sistema son los mapas de sitio [16]. En ellos se ejemplifica las distintas páginas y caminos a los que se puede acceder dentro del sitio, como se observa en la Figura 1-16.



**Figura 1-16 Ejemplo de Mapa del Sitio (bazuca.com)**

Fuente: <http://www.bazuca.com/BazucaHTML/img/botones/img/mapa-del-sitio.gif>

### 1.3.5. REUTILIZACIÓN DE SOFTWARE

Dentro de la ingeniería de software existen principios como DRY (Don't Repeat Yourself), que busca evitar que dentro del proceso de desarrollo se reinvente la rueda, esto se logra con el uso de frameworks o al reutilizar código, librerías e incluso software completo para realizar el desarrollo de un sistema web. Algunos ejemplos de reutilización que se realizan en el desarrollo de sistemas web se muestran a continuación:

- Librerías (Enterprise Library, Log4j, Smarty, JQuery)
- Frameworks (Symfony, ASP.Net, RoR)
- Software Completo (Joomla!, Moodle, Drupal, Wordpress, Umbraco, OSCcommerce, DotNetNuke)

En lo referente al uso de un software completo es común en la actualidad el uso del manejador de contenido Joomla! como base para el desarrollo de sistemas más complejos. Y es que mucho software de este tipo tiene una buena capacidad de extensibilidad a través del desarrollo de complementos o plugins para el mismo. En el caso particular de Joomla! al ser un proyecto Open Source que trabaja sobre PHP y MYSQL, ha ganado una gran popularidad por ser una solución económica para el despliegue de sistemas web sencillos.

Por lo tanto, se puede concluir que la reutilización de software es un factor importante dentro de los sistemas web actuales, lo que justifica una especial consideración durante el proceso de desarrollo.

## CAPÍTULO 2. DEFINICIÓN DE LA PROPUESTA

### 2.1. DEFINICIÓN Y ESPECIFICACIÓN DE LAS FASES DEL PROCESO

La propuesta de proceso de desarrollo que se va a definir en el presente capítulo, toma el nombre de PRODEWEB, acrónimo proveniente de las palabras PROceso de DEsarrollo WEB, la propuesta tiene como finalidad, brindar una alternativa para el desarrollo de sistemas web, al incorporar a una metodología base, los diversos elementos y criterios que se utilizan actualmente para el desarrollo web.

Para este fin se analizaron durante el capítulo anterior algunas de las metodologías más emblemáticas que son específicas para el desarrollo de sitios y sistemas web. A partir de este análisis se seleccionó a WebML como la metodología base del proceso, debido a que abarca gran parte del ciclo de vida del desarrollo de software.

A través del presente capítulo se procederá a adaptar y complementar la metodología WebML, con los criterios descritos en el capítulo anterior.

Algunas de las ideas más importantes que contendrá el nuevo proceso serán:

- **Dirigido por Datos.-** Los sistemas web tienen como principal tarea el despliegue y recolección de datos, que generalmente se almacenan en un repositorio de información. Por lo tanto se considera a los datos como el eje principal del sistema.
- **Promover el desarrollo rápido de aplicaciones.-** Fomentar el uso de herramientas como un mecanismo de acelerar el proceso de desarrollo, al evitar la realización de tareas complejas o repetitivas. Entiéndase por herramientas: Entornos de Desarrollo, Generadores de Código, Generadores de Plantillas, Frameworks, Librerías, etc.

- **Ofrecer entregas y liberaciones continuas.-** Brindar un proceso que permita el desarrollo de aplicaciones cuyas funcionalidades puedan incrementarse paulatinamente. Esto se lo logra al ofrecer en el proceso un enfoque iterativo e incremental. Se enfatiza que las iteraciones sean breves y brinden productos susceptibles de entrega al usuario lo más rápido posible. Por citar un ejemplo, un sistema web puede iniciar a operar únicamente con una página principal entregada tras la primera iteración y posteriormente pueden añadirse nuevas funcionalidades y características generadas en las subsiguientes iteraciones.
- **Incorporar varios principios de desarrollo ágil.-** Dotar al proceso de características ágiles como: la versatilidad, adaptabilidad, respuesta al cambio, el énfasis en la interacción entre los individuos, reducción de la cantidad de documentación necesaria, entre otros.
- **Estar diseñado para grupos pequeños y proyectos de mediano alcance.-** Incentivar el uso de un proceso de desarrollo incluso en proyectos cuyo alcance no sea tan amplio. Y útil para grupos pequeños en los que el diálogo y difusión de ideas puede ser más fácil que redactar una documentación formal

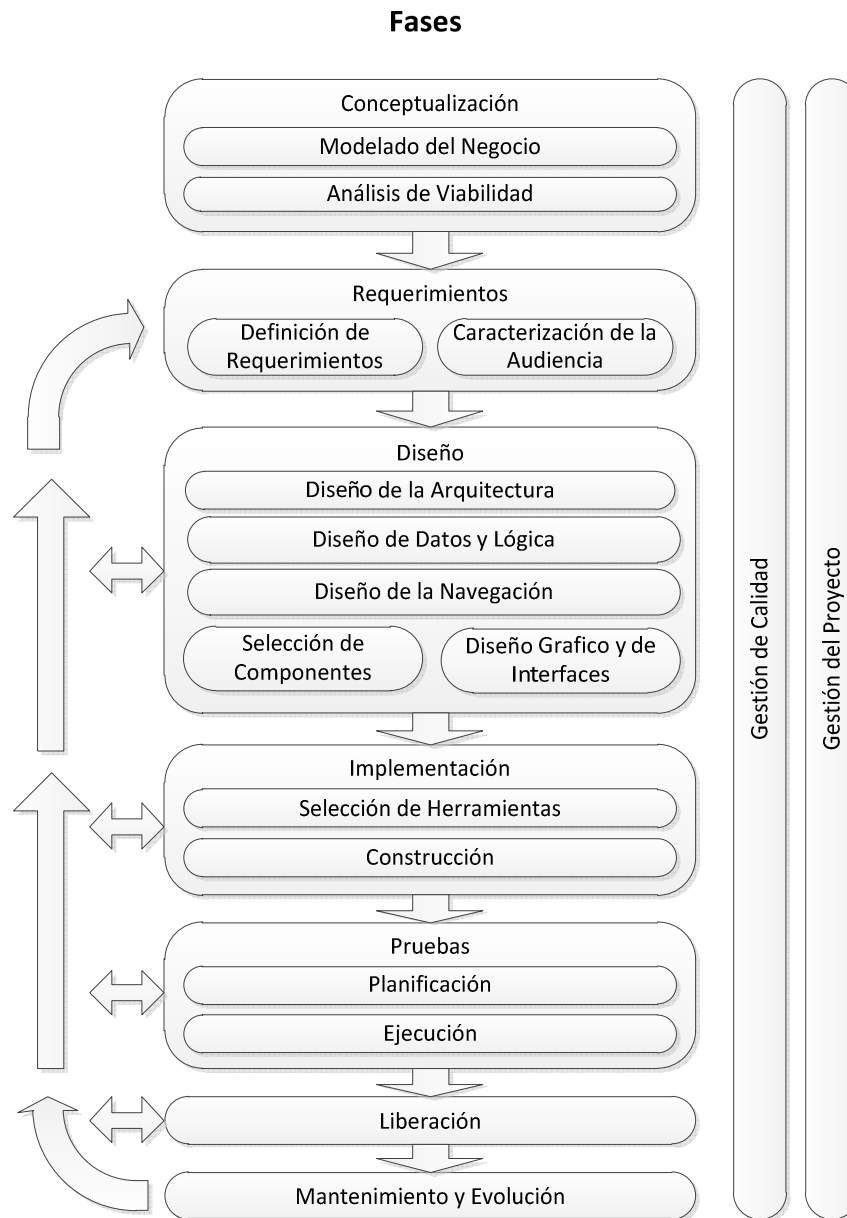
El proceso se describe a continuación y consta de las siguientes fases:

1. Conceptualización
2. Requerimientos
3. Diseño
4. Implementación
5. Pruebas
6. Liberación
7. Mantenimiento y Evolución

Adicionalmente se definieron dos flujos de apoyo que se ejecutan a lo largo de las diferentes fases del proceso. Éstos son:

1. Gestión de Calidad
2. Gestión del Proyecto

El diagrama del proceso PRODEWEB se lo puede observar en la Figura 2-1.



**Figura 2-1 Diagrama del Proceso PRODEWEB**  
Realizado por: Carlos Fabara

A continuación se especifican con mayor detalle cada una de las fases del proceso y como los flujos de apoyo interactúan con ellas:



### 2.1.1. CONCEPTUALIZACIÓN

La fase de conceptualización es una de las etapas que se añadieron a la propuesta con respecto al proceso base WebML, esta fase tiene dos objetivos primordiales, el primero es conocer al proyecto y el segundo es analizar su viabilidad, para determinar si emprenderlo o no, con esto se busca evitar que el equipo se comprometa en proyectos que no pueden ser cumplidos, ya sea por falta de recursos, infraestructura o alcances excesivamente extensos.

Esta fase se compone de dos subfases que son:

- a) Modelado del Negocio
- b) Análisis de Viabilidad

El modelado del negocio se realiza a través de la definición del problema, la finalidad y la visión del sistema, a través de éstos se busca conocer brevemente cual es el objetivo que tendrá el sistema, además de conocer el trasfondo y la realidad en la que se implementará. Este mecanismo de trabajo se lo hereda de la metodología WSDM la cual poseía una fase de “Definición de la Misión”, en la que en una frase o párrafo se definía toda la idea tras el sistema.

El análisis de viabilidad sirve, como su nombre lo indica, para determinar la viabilidad del proyecto que se plantea iniciar. Para que el Jefe del Proyecto se asegure de tomar la decisión correcta, es necesario difundir entre los miembros del equipo las ideas recabadas durante el modelado del negocio, para posteriormente retroalimentarse con las distintas opiniones y puntos de vista. Es importante recalcar que las opiniones de los miembros del equipo sirven únicamente, como una herramienta de apoyo para la toma de decisiones, ya que la decisión en sí, queda a criterio del Jefe del Proyecto.

### 2.1.2. REQUERIMIENTOS

La fase de requerimientos que propone el proceso PRODEWEB tiene como objetivos especificar formalmente los requerimientos del sistema y adicionalmente el detallar a la audiencia que lo operará. Esta fase se compone de dos subfases que son:

- a) Definición de Requerimientos
- b) Caracterización de la Audiencia

La etapa de definición de requerimientos tiene como finalidad profundizar en las ideas y limitaciones descritas durante el modelamiento del negocio y especificarlas de una manera formal. Tras el análisis de varias metodologías, se determinó que en el desarrollo web se utilizan diversos mecanismos para la especificación de requerimientos. El proceso PRODEWEB utilizará el mecanismo propuesto por WebML: los casos de uso; se recomienda su utilización debido a que son una herramienta ampliamente probada y utilizada por otras metodologías, además que los diagramas de casos de uso brindan información acerca de los usuarios (actores), sus tareas e interrelación entre ellos.

La granularidad en la especificación de los casos de uso dependerá de la complejidad de los mismos y del criterio del jefe del proyecto. Es recomendable para los casos de uso de tareas simples o de baja complejidad, no especificar flujos alternos, pre y post condiciones, etc., ya que se contradice con uno de los principios ágiles que es evitar la documentación excesiva. Adicionalmente con esta simplificación se busca unificar las ideas tras dos mecanismos de recolección de requerimientos las fortalezas de los casos de uso junto a la simplicidad de las historias de usuario.

La caracterización de la audiencia es un concepto heredado de WSDM; se define a “audiencia” como el grupo de usuarios finales que interactuarán con el sistema. En esta etapa se especificarán con mayor detalle a los actores definidos en los diagramas de casos de uso. Se deben definir las características y necesidades de

cada uno de ellos, ya que al conocer el perfil de los usuarios, el equipo de desarrollo puede tomar las mejores decisiones durante el diseño del sistema.

### **2.1.3. DISEÑO**

La fase de diseño es una de las más importantes del proceso, ya que en esta etapa se definirá la estructura de los distintos elementos, que posteriormente constituirán el sistema. Esta es una de las fases en las que se ha realizado el mayor número de cambios, con respecto a fases similares de otras metodologías.

Las subfases propuestas son:

- a) Diseño de la Arquitectura
- b) Diseño de Datos y Lógica
- c) Diseño Gráfico y de Interfaces
- d) Diseño de la Navegación
- e) Selección de Componentes

El Diseño de la Arquitectura es la etapa en la que se determina la estructura arquitectónica de la aplicación, por ejemplo capas y patrones; adicionalmente se puede definir información sobre la tecnología con que se construirá, sobre que plataforma se ejecutará, que base de datos, servidor web y mecanismos de conexión se utilizarán. El resultado de esta fase produce un diagrama de arquitectura en el que se especifica a nivel gráfico los diversos elementos y su interacción.

El Diseño de Datos y Lógica es una etapa primordial ya que el proceso propuesto es Dirigido por Datos, dado que gran parte de los sistemas basados en web, tienen como repositorio de información una base de datos. Durante esta fase, se modela a partir de la información recolectada durante la fase de requerimientos, el conjunto de datos con los que trabajará sistema y se los representa junto con sus relaciones en un modelo de datos.

Adicionalmente en esta subfase se realiza el diseño de la lógica y algoritmos necesarios para el funcionamiento del sistema. El proceso recomienda el uso de herramientas de mapeo objeto-relacional para la generación de clases del negocio, a partir de un modelo o una base de datos. En caso de que se especifiquen clases adicionales, y solo si el equipo de desarrollo lo considera necesario, se puede adicionar un diagrama de las nuevas clases para complementar la documentación.

El Diseño de la Navegación es una fase muy común en las distintas metodologías web, y es debido a que el paradigma de navegación es más evidente en una aplicación web, que en una aplicación de escritorio. Este concepto se encuentra incorporado en metodologías como WSDM y OOHDM. El proceso PRODEWEB propone el uso de un mapa del sitio como mecanismo para definir las diferentes opciones de navegación posibles.

El Diseño Gráfico y de Interfaces genera la estructura de las páginas y de su contenido, de manera que se adapten de mejor manera al flujo de navegación propuesto durante la etapa anterior. Durante esta etapa se utilizan varios criterios definidos en el Capítulo 1.3, uno de ellos es la arquitectura de información, que permite brindar un despliegue eficaz de los contenidos y los datos, así el diseño de interfaces busca que las pantallas del sistema se enmarquen en algún patrón de interacción que sea familiar a los usuarios.

El resultado de esta etapa son wireframes, mockups o prototipos de interfaz que se van complementando paulatinamente con elementos de diseño gráfico como son colores, patrones, logotipos y demás elementos gráficos que serán representativos para el sistema. Es importante que dichos elementos se encuentren asociados a la imagen de la organización o producto para el cual se construye.

La Selección de Componentes incorpora los criterios de reutilización de software al proceso, debido a que en la actualidad es una práctica común que los sistemas web utilicen librerías, frameworks y componentes de terceros, incluso existen

desarrollos que utilizan otros sistemas completos como base, este es el caso de los desarrollos realizados sobre manejadores de contenido como: Joomla!, Wordpress, DotNetNuke, etc.

El reutilizar software reduce el tiempo de desarrollo al “no reinventar la rueda” y permite que los desarrolladores se enfoquen en la implementación de la lógica de negocio. Se deben analizar durante esta fase, componentes o sistemas que se adapten a las necesidades y requerimientos definidos durante el diseño de la arquitectura. Y en caso de que los componentes seleccionados tengan una arquitectura diferente, sería necesario actualizar el diagrama de arquitectura.

#### **2.1.4. IMPLEMENTACIÓN**

La implementación es la fase en la que se realizará la construcción del sistema y los recursos necesarios para su funcionamiento. La fase propuesta consta de dos subfases que son:

- a) Selección de Herramientas
- b) Construcción

La etapa de Selección de Herramientas sirve para determinar un conjunto de aplicaciones y/o herramientas que se utilizarán para la construcción de los elementos determinados en dicha iteración. El proceso que se propone enfatiza en el uso de herramientas como elementos de apoyo para la realización de tareas, su uso permite que el desarrollo y construcción se lleven a cabo de una manera mucho más rápida y eficiente, si son usadas adecuadamente. Esta etapa se puede obviar en caso de que ya se la haya realizado previamente y no se requieran herramientas nuevas.

Durante la etapa de Construcción, se realiza la mayor carga de trabajo del proceso de desarrollo, en esta etapa se realiza la codificación, construcción y adaptación del código, basado en los modelos y artefactos definidos en la fase de diseño.

Es importante que durante la construcción, los programadores utilicen las herramientas seleccionadas previamente y se adhieran a estándares preestablecidos ya sean de organismos internacionales, del lenguaje de programación o de la organización en la que se realiza el desarrollo. Adicionalmente, es recomendable que el código sea comentado y documentado debidamente, de forma que se mantenga la consistencia a lo largo del proyecto y sea más fácil para cualquier miembro del equipo trabajar con código ajeno.

### **2.1.5. PRUEBAS**

Durante las pruebas se valida el producto elaborado durante la fase de construcción; se analiza que éste funcione correctamente y cumpla los requerimientos determinados en fases anteriores. Durante esta etapa se pretende identificar y eliminar errores que puedan existir dentro del sistema.

El proceso consta de dos subfases que son:

- Planificación
- Ejecución

La etapa de planificación se encarga de la definición de las pruebas que se van a realizar, mientras que la fase de ejecución se encarga de realizarlas.

Se pueden utilizar varios tipos de pruebas para verificar el software construido, se recomienda el uso de pruebas funcionales para validar las funcionalidades de cada módulo construido, pruebas de integración para validar la interacción y funcionamiento de los distintos módulos dentro de un flujo de trabajo y además se deben realizar pruebas de aceptación con el cliente durante las iteraciones.

Adicionalmente existen otros tipos de pruebas que se las considera opcionales dentro del proceso, pero que pueden ser utilizadas dependiendo de la complejidad del proyecto. Una de las más representativas son las pruebas

unitarias, las cuales se utilizan para automatizar el proceso de pruebas del código escrito por el equipo de desarrollo.

#### **2.1.6. LIBERACIÓN**

La fase de liberación representa la culminación del trabajo realizado a través de las otras fases del proceso. En esta etapa se realiza la entrega de un producto resultado de una iteración.

Cuando se completa una iteración cuyo resultado no es un módulo o producto que se va a entregar al cliente, se lo considera una liberación interna. En caso de ser una versión del producto apta para la entrega al cliente, esta se considerará una liberación externa.

Como las liberaciones externas son consideradas productos funcionales aptos para entrar a producción, es necesario realizar documentación que contenga información sobre la instalación, funcionamiento y operación del sistema. Adicionalmente es recomendable, más no imprescindible, generar un instalador para el sistema, de forma que se facilite el procedimiento de instalación y despliegue.

#### **2.1.7. MANTENIMIENTO Y EVOLUCIÓN**

Esta fase consta de dos grandes bloques, como su nombre mismo lo indica:

- a) Mantenimiento
- b) Evolución

En la subfase de mantenimiento se busca realizar dos tareas muy importantes: la corrección de errores no detectados y la implementación de mejoras de rendimiento, optimización, etc.

A través de la subfase de evolución se trata de evitar que el software caiga en el desuso, adaptándolo a los requerimientos y tecnologías siempre cambiantes. Esta es una subfase que lamentablemente no es realizada en gran parte de los proyectos de software, debido a que requiere de apoyo, compromiso y seguimiento por parte del cliente, quien muchas veces considera que el producto entregado es suficiente para sus necesidades y ve a la evolución del mismo como un gasto innecesario.

La etapa de evolución consiste en la especificación de nuevos requerimientos, por lo que el proceso la considera como el comienzo de un nuevo ciclo de iteraciones adicionales o de un nuevo proyecto en sí.

## **2.2. DISEÑO DE ENTREGABLES DEL PROCESO**

El proceso PRODEWEB, incorpora algunos aspectos de las metodologías ágiles y uno de ellos es disminuir la cantidad de documentación y entregables necesarios para llevar a buen término un proyecto de software.

Se utilizará para gran parte de la documentación, un concepto denominado “documento vivo”, el cual denota documentos que están en constante cambio y evolución a lo largo del proceso de desarrollo.

La documentación que el presente proceso utiliza es de carácter referencial y puede ser adaptada según las necesidades específicas de cada proyecto. Adicionalmente, el formato con el que se maneje la documentación física o digital, queda a criterio del jefe del proyecto y de la infraestructura que se disponga.

A continuación se procederá a detallar cada uno de los entregables, agrupados según la etapa en la que se construyen.

### **2.2.1. CONCEPTUALIZACIÓN**

La fase de conceptualización consta de dos entregables:



- a) Documento de Modelado de Negocio
- b) Checklist de Viabilidad

### Documento de Modelado de Negocio

El documento de Modelado de Negocio cuyo formato propuesto se lo puede observar en la Figura 2-2, debe incluir:

**Definición del Problema.-** Definición breve del problema que se busca solventar.

**Finalidad del Sistema.-** Definir en pocas líneas la finalidad y la misión del sistema.

**Visión del Sistema.-** Redactar en pocos párrafos, lo que el sistema hará y lo que no hará, de manera similar a un alcance, pero mucho más breve.

<p><b>Nombre del Proyecto:</b> <i>[Ingrese aquí el nombre del proyecto.]</i></p> <p><b>Definición del problema:</b></p> <p><i>[Defina en esta sección el problema que el proyecto busca solventar.]</i></p> <p><b>Finalidad del sistema:</b></p> <p><i>[Defina en esta sección la finalidad y la misión del sistema que se construirá.]</i></p> <p><b>Visión del sistema:</b></p> <p>El sistema debe:</p> <p><i>[Ingrese aquí en una lista de viñetas las tareas que el sistema SI realizará.]</i></p> <p>Es sistema no debe:</p> <p><i>[Ingrese aquí en una lista de viñetas las tareas que el sistema NO realizará.]</i></p>
--

**Figura 2-2 Formato del Documento de Modelado del Negocio**  
**Realizado por: Carlos Fabara**

## Checklist de Viabilidad

Es un documento de apoyo para la etapa de Análisis de Viabilidad, en este se detallan los criterios con los que se encuestará al equipo de desarrollo. El checklist debe contener el nombre del proyecto, el plazo de entrega (en caso de existir), el nombre del colaborador, los cargos que ejerce, si se disponen los conocimientos, la infraestructura, los recursos, el personal y finalmente el punto de vista con respecto a la viabilidad del proyecto junto con cualquier observación pertinente. El formato para el checklist, se muestra en la Figura 2-3.

<b>Nombre del Proyecto:</b> <i>[Ingrese aquí el nombre del proyecto.]</i>	
<b>Plazo de Entrega:</b> <i>[Ingrese aquí el plazo de entrega del proyecto, en caso de que exista.]</i>	
<b>Colaborador:</b> <i>[Ingrese aquí el nombre del colaborador a encuestar.]</i>	
<b>Cargo(s):</b> <i>[Ingrese aquí el o los cargos que el colaborador ejerce..]</i>	
<b>Checklist:</b>	
¿Se dispone de los conocimientos necesarios?:	<i>[Sí o No]</i>
¿Se dispone de la infraestructura para el desarrollo?:	<i>[Sí o No]</i>
¿Se dispone del personal para el desarrollo?:	<i>[Sí o No]</i>
¿Cree usted que el proyecto es factible?:	<i>[Sí o No]</i>
¿Cree usted se pueden cumplir con los plazos de entrega?:	<i>[Sí o No, en caso de existir plazos de entrega.]</i>
<b>Observaciones:</b>	
<i>[Redacte en esta sección las observaciones realizadas al proyecto.]</i>	

**Figura 2-3 Formato del Checklist de Viabilidad**  
Realizado por: Carlos Fabara

Algunas de las preguntas que se sugieren realizar en el checklist son:

- ¿Se dispone de los conocimientos necesarios?

- ¿Se dispone de la infraestructura para el desarrollo?
- ¿Se dispone de los recursos para el desarrollo?
- ¿Se dispone del personal para el desarrollo?
- ¿Cree usted que el proyecto es factible?
- ¿Cree usted que se pueden cumplir con los plazos de entrega?

### **2.2.2. REQUERIMIENTOS**

La fase de requerimientos consta de dos subfases, los entregables para cada una de ellas son:

- a) Documento de Especificación de Requerimientos
- b) Documento de Caracterización de la Audiencia

El formato del Documento de Especificación de Requerimientos dependerá del mecanismo de recolección de requerimientos seleccionado. La alternativa que el proceso PRODEWEB propone, es utilizar casos de uso para la especificación de requerimientos. El documento consta de los distintos diagramas de casos de uso que representan el comportamiento del sistema junto al detalle de cada uno de ellos. El formato para el documento se muestra en la Figura 2-4.

La especificación de los casos de uso debe contener únicamente el nombre del caso de uso, un código identificador y una descripción concisa del caso de uso utilizando lenguaje natural. Si la complejidad del caso de uso lo amerita, se puede también añadir la especificación de pre y post condiciones, así como la especificación de los flujos normales y alternativos.

El Documento de Caracterización de la Audiencia contiene información acerca de los usuarios que interactuarán con el sistema, una descripción de cada uno de ellos y las tareas que van a realizar en el sistema. El objetivo de este documento es brindar información sobre de los tipos de usuario y sus perfiles, de manera que se puedan tomar decisiones de diseño orientadas a dichos perfiles.

**Nombre del Proyecto:** *[Ingrese aquí el nombre del proyecto.]*

**Diagramas de Casos de Uso:**

*[Ingrese en esta sección las imágenes y/o los vínculos hacia los diagramas de casos de uso.]*

**Especificación de Casos de Uso:**

*[A continuación se brinda un ejemplo con el formato para la especificación del caso de uso, este debe duplicarse para cada uno de los casos de uso.]*

**Código:** *[Código identificador del caso de uso .]*

**Nombre:** *[El nombre asignado al caso de uso .]*

**Descripción:** *[ La descripción breve del caso de uso .]*

*[(Opcional) En caso de que la complejidad del caso de uso lo requiera se pueden añadir a este formato pre y post condiciones, flujos normales y alternativos y otros criterios usados en la especificación de casos de uso.]*

**Figura 2-4 Formato del Documento de Especificación de Requerimientos**  
**Realizado por:** Carlos Fabara

El formato del documento de Caracterización de la Audiencia se muestra en la Figura 2-5.

**Nombre del Proyecto:** *[Ingrese aquí el nombre del proyecto.]*

**Caracterización de la Audiencia:**

*[A continuación se brinda un ejemplo con el formato para la caracterización de los actores, este debe duplicarse para cada uno de los tipos de usuarios.]*

**Actor:** *[Nombre del actor.]*

**Descripción:** *[ La descripción del perfil del actor.]*

**Tareas:**

*[La lista de tareas que realizará el actor.]*

**Figura 2-5 Formato del Documento de Caracterización de la Audiencia**  
**Realizado por:** Carlos Fabara

### 2.2.3. DISEÑO

La etapa de diseño se compone de los siguientes documentos y artefactos:

- a) Diagrama de Arquitectura
- b) Modelo de Datos
- c) Prototipos de Interfaz
- d) Mapa del Sitio
- e) Lista de Componentes a Reutilizar

**Diagrama de Arquitectura.-** El diagrama de arquitectura es el primer paso en la etapa de diseño. En este documento se deben reflejar a modo de esquema, la arquitectura con la que será construida la aplicación, es decir módulos, patrones, capas, etc.

**Modelo de Datos.-** El modelo de datos se lo representa a través de un diagrama. Es necesario dar la importancia necesaria durante el diseño de los datos, ya que son una parte fundamental dentro del proceso. Un ejemplo común en la actualidad sería generar un diagrama entidad-relación para ser implementado posteriormente en una base de datos relacional.

**Mapa del Sitio.-** El mapa del sitio es la representación del esquema de navegación diseñado. Su finalidad es la de orientar a los miembros del equipo de desarrollo y a los usuarios del sistema acerca de las opciones por las que pueden navegar dependiendo del punto dentro del sitio en el que se encuentren.

**Prototipos de Interfaz (Wireframes).-** Los wireframes son la representación de la estructura y disposición de elementos que tendrá el sistema y sus diferentes páginas. El estilo de un wireframe debe ser minimalista, evitando cualquier detalle que pueda distraer u opacar la información contenida sobre la estructura del sitio. Los wireframes pueden ir evolucionando e incorporando paulatinamente elementos como colores, tipografías, imágenes, etc.

**Listado de Componentes por Reutilizar.-** Consiste en un listado de los componentes que se reutilizarán, junto con la descripción de la finalidad con la que se van a utilizar. Es recomendable también añadir la dirección web del componente y si es necesario el módulo o capa en la que se utilizarán.

Los entregables de esta fase se deben incorporar en el documento de Diseño, cuyo formato se muestra en la Figura 2-6

<p><b>Nombre del Proyecto:</b> <i>[Ingrese aquí el nombre del proyecto.]</i></p> <p><b>Diagrama de Arquitectura:</b></p> <p><i>[Inserte aquí el diagrama de arquitectura.]</i></p> <p><b>Diagrama de Datos:</b></p> <p><i>[Inserte aquí el Diagrama de Datos.]</i></p> <p><b>Mapa del Sitio:</b></p> <p><i>[Inserte aquí el Mapa del Sitio.]</i></p> <p><b>Prototipos de Interfaz:</b></p> <p><i>[Inserte aquí los prototipos de interfaz para el sistema.]</i></p> <p><b>Listado de Componentes:</b></p> <p><i>[Escriba aquí en una lista de viñetas los componentes seleccionados para reutilizar.]</i></p>
---

**Figura 2-6 Formato Documento de Diseño**  
**Realizado por: Carlos Fabara**

## 2.2.4. IMPLEMENTACIÓN

### Selección de Herramientas

**Listado de Herramientas.-** Es una lista con las herramientas que se utilizarán durante la fase de construcción. La finalidad de este documento es estandarizar

las herramientas que se utilizarán entre los miembros del equipo de desarrollo. Adicionalmente si es necesario se pueden especificar las características y usos de las herramientas seleccionadas.

### Construcción

Durante la fase de construcción es común que su resultado sea alguno de los siguientes tipos de artefactos:

**Lógica.-** Ésta puede ser representada en archivos fuente o en librerías compiladas. La lógica del sistema constituye una de las piezas más valiosas e importantes dentro del ciclo de vida del desarrollo de software.

**Script del Modelo de Datos.-** Archivo que contiene la información necesaria para implementar el modelo de datos en un repositorio de información. Debe generarse a partir del modelo definido durante la fase de diseño. Actualmente es común que se genere un script SQL para ser implementado en una base de datos relacional.

**Interfaces.-** En base a los wireframes creados durante la etapa de diseño, se construyen las plantillas y las interfaces del sistema. En éstas se integran los diferentes estilos, controles, recursos y demás elementos que las componen.

**Recursos.-** Los recursos son todos aquellos artefactos adicionales que requiere un sistema web. Estos pueden ser imágenes, artículos, videos, textos y otros contenidos.

#### **2.2.5. PRUEBAS**

La fase de pruebas consta de un par de entregables que evolucionan a lo largo de las iteraciones:

**Bitácora de Pruebas.-** Contiene la lista de las pruebas que deben realizarse, a cada prueba se le asigna un código identificador, un nombre, la descripción de la

prueba, el tipo al que pertenecen, el estado de la misma, el resultado obtenido y las observaciones generadas. La descripción puede ser tan extensa como se requiera y en caso de ser necesario, puede incluirse en este campo la secuencia de pasos necesarios para completar la prueba. El formato sugerido para la bitácora de pruebas se muestra en la Figura 2-7.

**Bitácora de Errores.-** Contiene la lista de errores encontrados, un código identificador, su importancia, un nombre, su descripción, el estado del error y la solución con la que se corrigió. Este documento se utiliza durante la etapa de pruebas y como complemento a la etapa de mantenimiento en caso de encontrarse nuevos errores cuando el sistema llegue a producción. El formato de la bitácora de errores se muestra en la Figura 2-8.

BITÁCORA DE PRUEBAS						
Proyecto:						
Código	Nombre	Descripción	Tipo	Estado	Resultado	Observaciones

**Figura 2-7 Formato Bitácora de Pruebas**  
Realizado por: Carlos Fabara

BITÁCORA DE ERRORES					
Proyecto:					
Código	Importancia	Nombre	Descripción	Estado	Solución

**Figura 2-8 Formato Bitácora de Errores**  
Realizado por: Carlos Fabara

En caso de que una prueba resulte fallida, se recomienda incluir en el campo de observaciones de la bitácora de pruebas, el código con el que se registró el falló en la bitácora de errores. De esta manera se interrelacionan ambos documentos. Si el error se encuentra solucionado, registrar este hecho en la bitácora de pruebas y actualizar el estado de la prueba a "Exitosa".



## 2.2.6. LIBERACIÓN

### Liberaciones Internas

**Listado de Cambios.-** El listado de cambios contiene información acerca de la evolución que ha tenido el producto. En este documento se registran los cambios más importantes realizados al sistema en cada liberación. De esta manera el equipo de desarrollo y los usuarios pueden conocer con mayor detalle los cambios realizados y las nuevas características implementadas. En la Figura 2-9, se muestra el formato sugerido.

**Producto de la Iteración.-** El producto de la iteración es el resultado del trabajo realizado durante la fase de construcción. Pueden ser sistemas, parte de ellos o únicamente fragmentos de código o algoritmos.

### Liberaciones Externas

Los artefactos resultado de la liberación externa deben ser entregados según el formato convenido con el cliente, ya sea físico (CD, DVD, Blu-Ray) o a través de medios digitales.

**Nombre del Proyecto:** *[Ingrese aquí el nombre del proyecto.]*

#### **Listado de Cambios**

- Versión *[Escriba aquí la versión del sistema.]*

*[En una lista especifique los cambios realizados en el sistema]*

**Figura 2-9 Formato Listado de Cambios**

**Realizado por:** Carlos Fabara

**Listado de Cambios.-** El listado de cambios contiene información acerca de la evolución que ha tenido el producto. En este documento se registran los cambios más importantes realizados al sistema en cada liberación. De esta manera el

equipo de desarrollo y los usuarios pueden conocer con mayor detalle los cambios realizados y las nuevas características implementadas.

**Manual de Instalación.-** Contiene la información y explicación de los procedimientos necesarios para realizar la instalación y puesta en marcha del sistema. Es recomendable que el documento sea lo suficientemente detallado como para evitar posibles errores durante la implantación del sistema.

**Manual de Usuario.-** El manual de usuario es un documento en el que se explican las funcionalidades y los procedimientos necesarios para operar el sistema. Es recomendable que se detallen los pasos utilizando lenguaje natural y acompañándolos de capturas e imágenes, de forma que sea lo más didáctico posible.

**Producto Software.-** Es el producto a entregar al cliente, puede ser el sistema completo, parches, módulos, plugins, etc.

### **2.2.7. MANTENIMIENTO Y EVOLUCIÓN**

El mantenimiento se lleva a cabo después de la liberación del producto y consiste principalmente en la corrección de errores, optimización y mejoras de rendimiento del sistema.

La corrección de errores se la realiza a partir de aquellos reportados en la bitácora de errores; cualquier error encontrado en producción o reportado por los usuarios debe ser registrado dentro de dicha bitácora para ser corregido durante esta etapa.

Las futuras características deben registrarse en un documento compuesto de una lista simple en la que se indican las nuevas características que se esperan del sistema en el futuro.

## **2.3. DEFINICIÓN DE PROCEDIMIENTOS Y HERRAMIENTAS PARA CADA FASE DEL PROCESO**

Los procedimientos y herramientas que se expondrán en el presente subcapítulo son de carácter referencial y no representan el único camino para realizar las tareas de cada fase. La adaptabilidad es una de las principales fortalezas del proceso PRODEWEB, por lo que si las personas interesadas en utilizar el proceso consideran necesario el uso de otras herramientas o realizar un procedimiento diferente, se encuentran en la libertad de hacerlo, siempre y cuando se enmarquen en los lineamientos definidos en el Capítulo 2.1.

### **2.3.1. CONCEPTUALIZACIÓN**

#### **Procedimientos**

##### Modelado de Negocio

1. Reunirse con la persona que conozca del negocio y/o cliente.
2. Escuchar los requerimientos.
3. Tras la reunión redactar el documento de Modelado de Negocio.

##### Análisis de Viabilidad

1. Realizar una reunión con los miembros del equipo de desarrollo, en la que se difunde el contenido del documento de modelado de negocio.
2. Realizar una encuesta a los miembros del equipo, utilizando el formato propuesto de checklist. Esta encuesta es únicamente de carácter referencial.
3. Analizar los resultados de la encuesta.
4. Dialogar con los miembros del equipo los resultados obtenidos.
5. Tras haberse analizado y considerado los puntos de vista, queda a criterio del jefe del proyecto determinar, si iniciar o no el proyecto.

## Herramientas

Tipo Herramienta	Finalidad	Ejemplos
Editores de Texto	Redactar documentos de la fase.	<ul style="list-style-type: none"> <li>• Microsoft Word</li> <li>• OpenOffice Writer</li> <li>• Google Docs</li> </ul>

**Tabla 2-1 Herramientas para la etapa "Conceptualización"**

Realizado por: Carlos Fabara

### 2.3.2. REQUERIMIENTOS

#### Procedimientos

##### Especificación de Requerimientos

1. Analizar el documento de Modelado de Negocio.
2. Definir los Casos de Uso necesarios para las funcionalidades del sistema.
3. Construir el Documento de Especificación de Requerimientos.
4. Validar el Documento con la persona que conozca del negocio y/o cliente.

##### Caracterización de la Audiencia.

1. Analizar el documento de Especificación de Requerimientos.
2. Describir con mayor detalle el perfil de los usuarios y/o actores de los casos de uso.
3. Construir el Documento de Caracterización de la Audiencia.
4. Validar el Documento con la persona que conozca del negocio y/o cliente.

#### Herramientas

Tipo Herramienta	Finalidad	Ejemplos
Editores de Texto	Redactar documentos de la fase.	<ul style="list-style-type: none"> <li>• Microsoft Word</li> <li>• OpenOffice Writer</li> <li>• Google Docs</li> </ul>
Modelado UML	Construcción de Diagramas de Casos de Uso	<ul style="list-style-type: none"> <li>• Enterprise Architect</li> <li>• Umbrello</li> <li>• Magic Draw</li> </ul>

		• StarUML
--	--	-----------

**Tabla 2-2 Herramientas para la etapa "Requerimientos"**

**Realizado por:** Carlos Fabara

### 2.3.3. DISEÑO

#### Procedimientos

##### Diseño de la Arquitectura

1. Realizar una reunión entre los miembros del equipo de desarrollo y con la persona que conozca del negocio y/o cliente.
2. Dialogar durante la reunión acerca de las limitantes a nivel técnico, de infraestructura, económicas, con la finalidad de conocer el presupuesto y sobre que tecnologías se podría trabajar.
3. Definir formalmente la arquitectura que tendrá el sistema y plasmarla en el diagrama de arquitectura.

##### Diseño de Datos y Lógica

1. Analizar los casos de uso
2. Identificar los elementos de información que manejará el sistema
3. Agruparlos dentro de Entidades de Datos
4. Generar un Diagrama de Datos (Entidad-relación, XML, etc.)
5. Diseñar la lógica del sistema (flujos y algoritmos)

##### Diseño de la Navegación

1. Identificar el conjunto de interfaces necesarias para cumplir las tareas especificadas en los requerimientos.
2. Determinar la jerarquía y caminos posibles entre las interfaces.
3. Plasmar las ideas en un mapa del sitio que especifique la información determinada previamente.

## Diseño Gráfico y de Interfaces

1. En base a los datos que manejará el sistema, encontrar cual es el patrón de interacción que mejor se adapte.
2. Construir una serie de wireframes para ilustrar el esquema que tendrán las páginas del sistema.
3. Validar con la persona que conozca del negocio y/o cliente el esquema propuesto.
4. A partir de los colores de la institución crear una paleta de colores para el sitio.
5. Construir o buscar en base a esta paleta los elementos gráficos que se incorporarán al sitio.

## Selección de Componentes

1. Tras conocer la arquitectura y los requerimientos de la aplicación, reunir a los especialistas técnicos y proponer alternativas para la reutilización.
2. Analizar cada una de las alternativas y seleccionar aquellas que más se adecuen a la realidad del proyecto.
3. Si existe un cliente, validar las alternativas propuestas con él.
4. Adquirir o descargar las alternativas de reutilización propuestas.

## **Herramientas**

### Diseño de la Arquitectura

<b>Tipo Herramienta</b>	<b>Finalidad</b>	<b>Ejemplos</b>
Diagramación	Construir el Diagrama de Arquitectura	<ul style="list-style-type: none"> <li>• Microsoft Visio</li> <li>• Dia</li> </ul>

**Tabla 2-3 Herramientas para la etapa "Diseño de Arquitectura"**

**Realizado por: Carlos Fabara**

### Diseño de Datos y Lógica

<b>Tipo Herramienta</b>	<b>Finalidad</b>	<b>Ejemplos</b>
Computer Aided Software	Creación del Modelo de	<ul style="list-style-type: none"> <li>• Power Designer</li> </ul>

Engineering CASE	Datos	• Rational Architect	Software
------------------	-------	----------------------	----------

**Tabla 2-4 Herramientas para la etapa "Diseño de Datos y Lógica"**  
Realizado por: Carlos Fabara

### Diseño Gráfico y de Interfaces

Tipo Herramienta	Finalidad	Ejemplos
Wireframing	Construcción de Wireframes.	<ul style="list-style-type: none"> <li>• Balsamiq Mockups</li> <li>• Frame Box</li> <li>• Microsoft Visio</li> </ul>
Paletas de Colores	Selección de Patrones de Colores.	<ul style="list-style-type: none"> <li>• Adobe Kuler</li> <li>• Color Scheme Designer</li> </ul>

**Tabla 2-5 Herramientas para la etapa "Diseño Gráfico y de Interfaces"**  
Realizado por: Carlos Fabara

### Diseño de la Navegación

Tipo Herramienta	Finalidad	Ejemplos
Diagramación	Construir el Mapa del Sitio	<ul style="list-style-type: none"> <li>• Microsoft Visio</li> <li>• Dia</li> </ul>

**Tabla 2-6 Herramientas para la etapa "Diseño de la Navegación"**  
Realizado por: Carlos Fabara

### Selección de Componentes

Tipo Herramienta	Finalidad	Ejemplos
Editores de Texto	Redactar documentos de la fase.	<ul style="list-style-type: none"> <li>• Microsoft Word</li> <li>• OpenOffice Writer</li> <li>• Google Docs</li> </ul>

**Tabla 2-7 Herramientas para la etapa "Selección de Componentes"**  
Realizado por: Carlos Fabara

## 2.3.4. IMPLEMENTACIÓN

### Procedimientos

1. Analizar los artefactos de diseño.
2. Reunir a los especialistas técnicos y proponer las herramientas que se podrían utilizar.
3. Seleccionar las herramientas necesarias.

4. Adquirir o descargar las herramientas
5. Construir los elementos definidos para la iteración
6. Preparar el producto para la etapa de pruebas

## Herramientas

Tipo Herramienta	Finalidad	Ejemplos
Entornos Integrados de Desarrollo	Realizar la programación de los distintos módulos del sistema	<ul style="list-style-type: none"> <li>• Visual Studio</li> <li>• Eclipse</li> <li>• Netbeans</li> <li>• Aptana</li> <li>• XCode</li> </ul>
Control de Versiones	Mantener un control de los fuentes a lo largo del desarrollo	<ul style="list-style-type: none"> <li>• SVN</li> <li>• Team Foundation Server</li> <li>• Mercurial</li> <li>• Git</li> </ul>
Editores de Imágenes	Construcción y Edición de Imágenes y demás elementos del sistema.	<ul style="list-style-type: none"> <li>• Adobe Photoshop</li> <li>• GIMP</li> </ul>
Editores Vectoriales	Construcción de Logotipos y otros elementos.	<ul style="list-style-type: none"> <li>• Adobe Illustrator</li> <li>• Corel Draw</li> <li>• Inkscape</li> </ul>
Imágenes, Stocks y Recursos Varios	Recursos para complementar el diseño gráfico del sistema.	<ul style="list-style-type: none"> <li>• Photobucket</li> <li>• Stock Exchange</li> <li>• Open Clipart</li> </ul>

**Tabla 2-8 Herramientas para la etapa "Implementación"**  
Realizado por: Carlos Fabara

### 2.3.5. PRUEBAS

#### Procedimientos

1. Planificar en la Bitácora de Pruebas, aquellas pruebas que se consideren necesarias.
2. Ejecutar las pruebas descritas en la Bitácora de Pruebas.
3. En caso de encontrar algún error reportarlo en la Bitácora de Errores.
4. En caso de existir errores, corregirlos y realizar nuevamente las pruebas hasta solucionarlos.
5. Si las pruebas fueron exitosas continuar con la etapa de liberación.



## Herramientas

Tipo Herramienta	Finalidad	Ejemplos
Seguimiento de Errores	Ayuda en el control y manejo de errores	<ul style="list-style-type: none"> <li>• Bugzilla</li> <li>• JitterBug</li> </ul>
Hoja de Cálculo	Creación de la bitácora	<ul style="list-style-type: none"> <li>• Microsoft Excel</li> <li>• OpenOffice Calc</li> </ul>
Editores de Texto	Redactar documentos de la fase.	<ul style="list-style-type: none"> <li>• Microsoft Word</li> <li>• OpenOffice Writer</li> <li>• Google Docs</li> </ul>

**Tabla 2-9 Herramientas para la etapa "Pruebas"**

Realizado por: Carlos Fabara

### 2.3.6. LIBERACIÓN

#### Procedimientos

1. Tras obtener un producto susceptible de liberación ya sea interna o externa, es necesario etiquetarlo como una versión.
2. Realizar el documento de cambios realizados.
3. En caso de ser una liberación externa construir el material necesario.
4. Validar el contenido del producto y la documentación que lo acompaña.
5. Realizar la liberación del producto como tal.
6. En caso de ser necesario, se recomienda colaborar con el cliente durante la puesta a producción del sistema.

## Herramientas

Tipo Herramienta	Finalidad	Ejemplos
Editores de Texto	Redactar manuales y documentación de la fase.	<ul style="list-style-type: none"> <li>• Microsoft Word</li> <li>• OpenOffice Writer</li> <li>• Google Docs</li> </ul>
Control de Versiones	Mantener un control de las versiones del software y las liberaciones realizadas	<ul style="list-style-type: none"> <li>• SVN</li> <li>• Team Foundation Server</li> <li>• Mercurial</li> <li>• Git</li> </ul>
Compresión de Archivos	Construcción y Edición de Imágenes y demás elementos del sistema.	<ul style="list-style-type: none"> <li>• 7-Zip</li> <li>• WinRAR</li> <li>• WinZIP</li> </ul>

**Tabla 2-10 Herramientas para la etapa "Liberación"**

**Realizado por:** Carlos Fabara

### 2.3.7. MANTENIMIENTO Y EVOLUCIÓN

#### Procedimientos

##### Mantenimiento

1. Si se detectan errores o problemas de rendimiento tras la salida a producción, reportarlos en la bitácora de errores.
2. Analizar los errores en busca de una solución.
3. Corregir los errores.
4. Realizar una liberación con las correcciones realizadas.

##### Evolución

1. Identificar junto con el cliente las nuevas características que se podrían añadir al sistema.
2. Listar estas características en un documento.

#### Herramientas

<b>Tipo Herramienta</b>	<b>Finalidad</b>	<b>Ejemplos</b>
Seguimiento de Errores	Ayuda en el control y manejo de errores	<ul style="list-style-type: none"> <li>• Bugzilla</li> <li>• JitterBug</li> </ul>
Hola de Cálculo	Creación Bitácora	<ul style="list-style-type: none"> <li>• Microsoft Excel</li> <li>• OpenOffice Calc</li> <li>• Google Docs</li> </ul>
Editores de Texto	Redactar documentos de la fase.	<ul style="list-style-type: none"> <li>• Microsoft Word</li> <li>• OpenOffice Writer</li> <li>• Google Docs</li> </ul>

**Tabla 2-11 Herramientas para la etapa "Mantenimiento y Evolución"**

**Realizado por:** Carlos Fabara

## 2.4. FLUJOS DE APOYO AL PROCESO

### 2.4.1. FLUJO DE GESTIÓN DEL PROYECTO

El proceso propuesto se centra en los aspectos metodológicos necesarios para la realización de un proyecto de software web. Pero se observó que era necesario apoyarlo en el ámbito de la gestión de las tareas y del proyecto, de forma que se pueda llevar a cabo de manera organizada y eficiente.

El flujo de apoyo se basa en Scrum, una de las metodologías analizadas previamente, como se expresó en el Capítulo 1.2.4, Scrum es una metodología orientada a gestionar los aspectos no técnicos de un proyecto de software. En el proceso PRODEWEB, se encargara de gestionar las tareas y las iteraciones necesarias para culminar el proyecto con éxito.

Este flujo tiene un alcance global y abarca prácticamente todo el ciclo de vida del proyecto, su importancia es mayor durante la fase de implementación, ya que ahí ayudará a gestionar las micro-iteraciones necesarias, organizar, repartir y priorizar las tareas entre los miembros del equipo de desarrollo.

Este flujo se apoya en una bitácora donde por cada una de las fases se van añadiendo las distintas tareas que surgen, su descripción, estado, colaborador asignado, fechas de: inicio, fin estimado y fin real. De esta forma se puede tener un control de las tareas pendientes y realizadas, e información cronológica de los hechos. El formato propuesta para la bitácora se muestra en la Figura 2-10.

BITÁCORA DE TAREAS							
Proyecto:							
Fase	Tarea	Descripción	Estado	Asignado	Fecha Inicio	Fecha Estimada Fin	Fecha Fin

**Figura 2-10 Formato Bitácora de Tareas**  
Realizado por: Carlos Fabara

## 2.4.2. FLUJO DE GESTIÓN DE CALIDAD

El flujo de Gestión de Calidad es aquel encargado de garantizar y velar la calidad de los entregables, artefactos y productos del proyecto. Este flujo tiene un alcance global y se ejecuta paralelamente a lo largo de todo el ciclo de vida del proyecto, engloba aspectos tales como: validación de entregables, validación de documentos, cumplimiento de estándares, evitar ambigüedades, etc.

Este flujo tiene una importancia mayor durante la etapa de pruebas, ya que son ellas las que mayor incidencia tienen en verificar la calidad del producto final. Este flujo no tiene entregables, pero su utilidad surge en enfatizar la importancia de la calidad durante todo el proceso.

Dependiendo de la fase del proceso en la que se encuentre sus funciones variarán, por citar unos ejemplos:

**Conceptualización.-** Revisión y validación del documento de modelado del negocio.

**Requerimientos.-** Validación de los requerimientos especificados en los Casos de Uso, revisión de los Perfiles de Usuario

**Diseño.-** Validación de la arquitectura, revisión de la consistencia del modelo de datos, del mapa del sitio y de las interfaces.

**Implementación.-** Control, revisión, pruebas y validación de los entregables. Control del cumplimiento de estándares.

**Pruebas.-** Evaluación las pruebas planteadas y los resultados obtenidos.

**Liberación.-** Verificación de los contenidos a entregarse.

**Mantenimiento.-** Evaluación del sistema.

## **CAPÍTULO 3. CASO DE ESTUDIO**

### **3.1. DESCRIPCIÓN DEL CASO DE ESTUDIO**

Con el objetivo de validar el proceso se realizará un caso de estudio, que consistirá en el desarrollo de un sistema web que permitirá el manejo de un catálogo de productos para una compañía de suministros de equipos de computación.

El sistema dispondrá de una página principal, página de registro, página de autenticación, una vista de las categorías, un listado de productos por categoría y una página para descripción del producto. Para administrarlo dispondrá de interfaces para la gestión de productos, categorías y usuarios.

El nombre del sistema a desarrollar será “TechnoCatalog”. Los roles de cliente, usuario final, jefe del proyecto, especialista técnico, desarrollador, etc., serán llevados a cabo por el autor de este trabajo.

### **3.2. APLICACIÓN DE LA PROPUESTA**

#### **3.2.1. CONCEPTUALIZACIÓN**

En esta etapa se debe realizar el documento de Modelado del Negocio y el Análisis de Viabilidad del sistema. A continuación se detallará lo que se realizó en cada una de las subfases de esta etapa.

##### **3.2.1.1. Modelado del Negocio**

El documento de modelado del negocio, que se generó para el caso de estudio, se lo puede apreciar en la Figura 3-1.

**Nombre del Proyecto:** TechnoCatalog

**Definición del problema:**

Para una empresa es importante que sus clientes conozcan la mayor cantidad de productos del catálogo que tienen para ofrecerles, de forma que ellos se interesen por nuevos productos o por aquellos que desconocían que la empresa distribuía. Pero es difícil para una empresa dar a conocer a sus clientes todo su catálogo de productos, debido a que cada cliente tiene intereses particulares y las limitaciones del espacio físico en los almacenes no permiten que se pueda exponer toda la mercadería con la misma importancia, por lo que en muchos casos hay productos que terminan olvidados y relegados.

**Finalidad del sistema:**

El sistema que se busca construir debe permitir que los clientes o potenciales clientes conozcan más acerca de la empresa, además debe mostrar un catálogo organizado de los productos que oferta la empresa, que incluya información como precios, características, fabricante, etc., de cada uno de ellos.

**Visión del sistema:**

El sistema debe:

- Mostrar la información relevante de la empresa.
- Mostrar la ubicación e información de contacto.
- Presentar un catálogo de productos.
- Mostrar en una página la información relevante de un producto,
- Restringir cierta información a los clientes no registrados.
- Permitir la administración del catálogo.

El sistema no hará:

- La adquisición de productos vía web.
- El inventario de productos.
- Implementará un carrito de compras.

**Figura 3-1 Modelado de Negocio (Caso de Estudio)**

**Realizado por:** Carlos Fabara

Tras haberse redactado el documento es necesario analizar su viabilidad por lo que se procederá a realizar el análisis de viabilidad. Para este fin se llenará la

checklist de viabilidad, con el criterio que se tiene del proyecto. A continuación se muestra la checklist llena para el proyecto.

### 3.2.1.2. Análisis de Viabilidad

El checklist que se utilizó para el caso de estudio, se lo puede apreciar en la Figura 3-2.

<b>Nombre del Proyecto:</b> TechnoCatalog	
<b>Plazo de Entrega:</b> 45 días	
<b>Colaborador:</b> Carlos Fabara	
<b>Cargo(s):</b> Jefe del Proyecto, Especialista Técnico, Diseñador Gráfico, Desarrollador	
<b>Checklist:</b>	
¿Se dispone de los conocimientos necesarios?:	Sí
¿Se dispone de la infraestructura para el desarrollo?:	Sí
¿Se dispone del personal para el desarrollo?:	Sí
¿Cree usted que el proyecto es factible?:	Sí
¿Cree usted se pueden cumplir con los plazos de entrega?:	Sí
<b>Observaciones:</b>	
Ninguna	

**Figura 3-2 Checklist de Viabilidad (Caso de Estudio)**  
Realizado por: Carlos Fabara

La encuesta que se realiza para conocer el criterio de los involucrados debe ser analizada. En este caso particular, al ser el autor de este trabajo quien realiza todos los roles el criterio es único, y el análisis de los resultados muestra que se considera que existen los medios necesarios para llevar a cabo el proyecto.

### 3.2.2. REQUERIMIENTOS

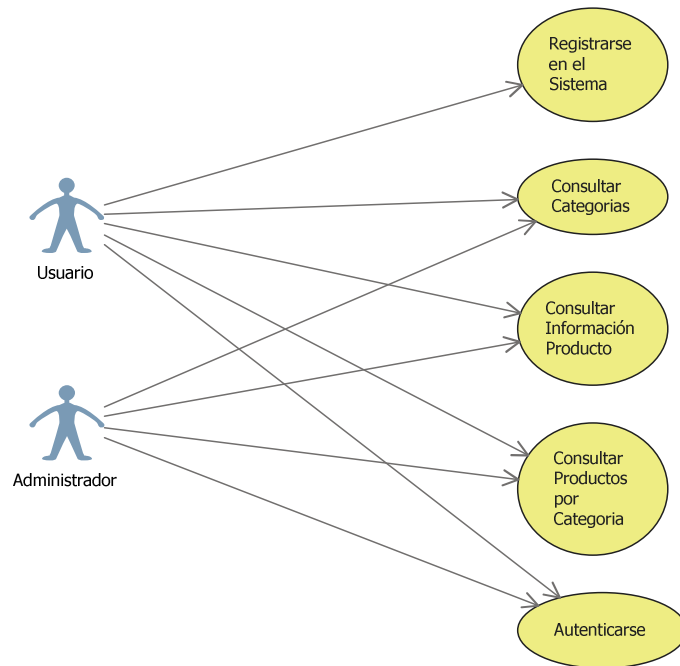
En esta etapa se debe realizar el documento de Especificación de Requerimientos y el documento de Caracterización de la Audiencia.

### 3.2.2.1. Especificación de Requerimientos

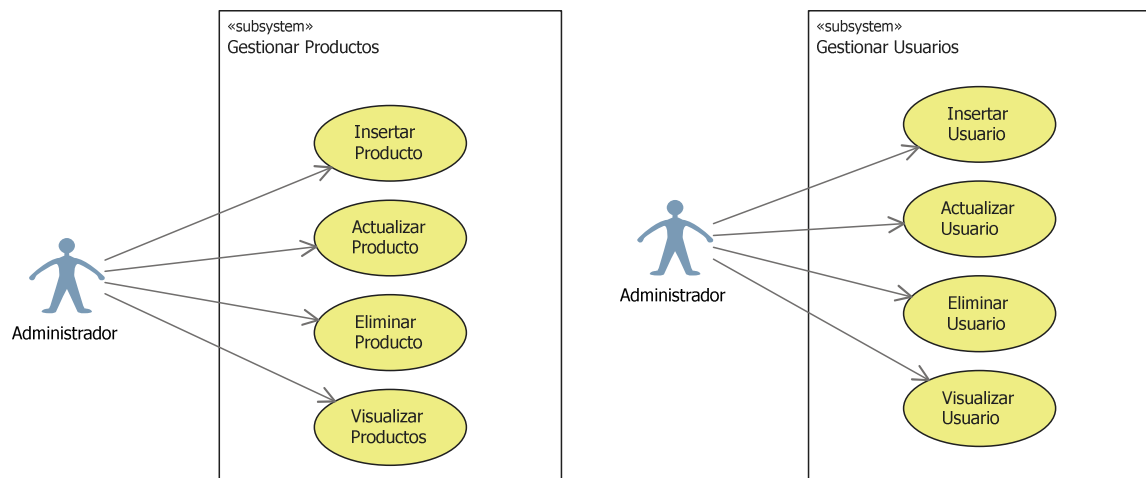
**Nombre del Proyecto:** TechnoCatalog

**Diagramas de Casos de Uso:**

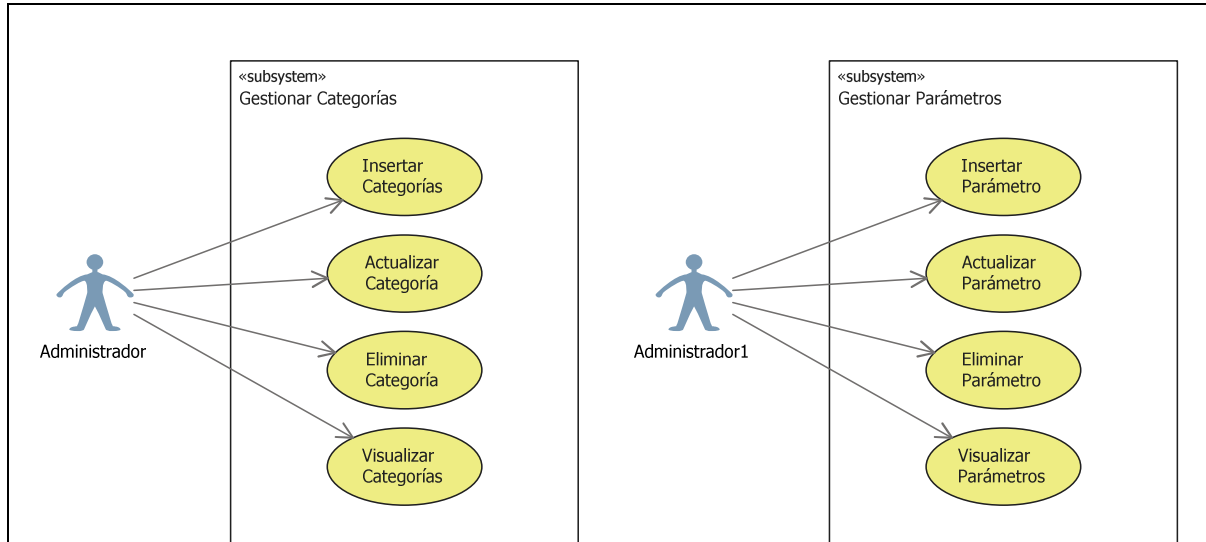
Catálogo



Sección Administrativa







### Especificación de Casos de Uso:

**Código:** CU-001

**Nombre:** Consultar Categorías

**Descripción:** Los usuarios podrán visualizar en una página las categorías del catálogo de productos.

**Código:** CU-002

**Nombre:** Consultar Información del Producto

**Descripción:** Al seleccionar un producto previamente, los usuarios podrán conocer las características del mismo.

**Código:** CU-003

**Nombre:** Consultar Productos por Categoría

**Descripción:** Se debe listar en una página los productos correspondientes a una categoría previamente seleccionado.

**Código:** CU-004

**Nombre:** Registrarse en el Sistema

**Descripción:** Se debe ofrecer a los usuarios la posibilidad de registrarse para acceder a funcionalidades especiales del sitio como visualizar precios.

**Código:** CU-005

**Nombre:** Autenticarse

**Descripción:** Para acceder a las funcionalidades especiales es necesario que un usuario previamente registrado se autentique en el sistema.

**Código:** CU-006

**Nombre:** Insertar Producto

**Descripción:** Dentro de la gestión de productos se permite al administrador insertar nuevos productos en el sistema.

**Código:** CU-007

**Nombre:** Actualizar Producto

**Descripción:** Dentro de la gestión de productos se permite al administrador realizar la actualización de la información de productos ya existentes.

**Código:** CU-008

**Nombre:** Eliminar Producto

**Descripción:** Dentro de la gestión de productos se permite al administrador eliminar productos del sistema.

**Código:** CU-009

**Nombre:** Visualizar Productos

**Descripción:** Dentro de la gestión de productos se permite al administrador visualizar los productos que se encuentran registrados en el sistema.

**Código:** CU-010

**Nombre:** Insertar Usuario

**Descripción:** Dentro de la gestión de usuarios se permite al administrador insertar nuevos usuarios en el sistema.

**Código:** CU-011

**Nombre:** Actualizar Usuario

**Descripción:** Dentro de la gestión de usuarios se permite al administrador realizar la actualización de la información de usuarios ya existentes.

**Código:** CU-012

**Nombre:** Eliminar Usuario

**Descripción:** Dentro de la gestión de usuarios se permite al administrador eliminar usuarios del sistema.

**Código:** CU-013

**Nombre:** Visualizar Usuarios

**Descripción:** Dentro de la gestión de usuarios se permite al administrador visualizar los usuarios que se encuentran registrados en el sistema.

**Código:** CU-014

**Nombre:** Insertar Categoría

**Descripción:** Dentro de la gestión de categorías se permite al administrador insertar nuevas categorías en el sistema.

**Código:** CU-015

**Nombre:** Actualizar Categoría

**Descripción:** Dentro de la gestión de categorías se permite al administrador realizar la actualización de la información de categorías ya existentes.

**Código:** CU-016

**Nombre:** Eliminar Categoría

**Descripción:** Dentro de la gestión de categorías se permite al administrador eliminar categorías del sistema, que no contengan productos.

**Código:** CU-017

**Nombre:** Visualizar Categorías

**Descripción:** Dentro de la gestión de categorías se permite al administrador visualizar las categorías que se encuentran registradas en el sistema.

**Código:** CU-018

**Nombre:** Insertar Parámetro

**Descripción:** Dentro de la gestión de parámetros se permite al administrador insertar nuevos usuarios en el sistema.

**Código:** CU-019

**Nombre:** Actualizar Parámetro

**Descripción:** Dentro de la gestión de parámetros se permite al administrador realizar la actualización de la información de parámetros ya existentes.

**Código:** CU-020

**Nombre:** Eliminar Parámetro

**Descripción:** Dentro de la gestión de parámetros se permite al administrador eliminar parámetros del sistema.

**Código:** CU-021

**Nombre:** Visualizar Parámetros

**Descripción:** Dentro de la gestión de parámetros se permite al administrador visualizar los parámetros que se encuentran ingresados en el sistema.

**Figura 3-3 Especificación de Requerimientos (Caso de Estudio)**  
**Realizado por:** Carlos Fabara

### 3.2.2.2. Caracterización de la Audiencia

<p><b>Nombre del Proyecto:</b> TechnoCatalog</p> <p><b>Caracterización de la Audiencia:</b></p> <p><b>Actor:</b> Usuarios</p> <p><b>Descripción:</b> Son todas aquellas personas que interactuarán con el sistema con la finalidad de obtener información del catálogo de productos. Existen un tipo especial de usuarios que son aquellos usuarios registrados, estos tienen privilegios que no disponen los usuarios normales.</p> <p><b>Tareas:</b></p> <ul style="list-style-type: none"> <li>• Consultar Categorías</li> <li>• Consultar Información del Producto</li> <li>• Consultar Productos por Categoría</li> <li>• Registro</li> <li>• Autenticación</li> </ul> <p><b>Actor:</b> Administrador</p> <p><b>Descripción:</b> El o los administradores del sistema deben ser personas que tengan conocimientos básicos de administración de sistemas a través de interfaces web. Serán los encargados de mantener el sistema a lo largo del tiempo.</p> <p><b>Tareas:</b></p> <ul style="list-style-type: none"> <li>• Consultar Categorías</li> <li>• Consultar Información del Producto</li> <li>• Consultar Productos por Categoría</li> <li>• Registro</li> <li>• Autenticación</li> </ul>
--

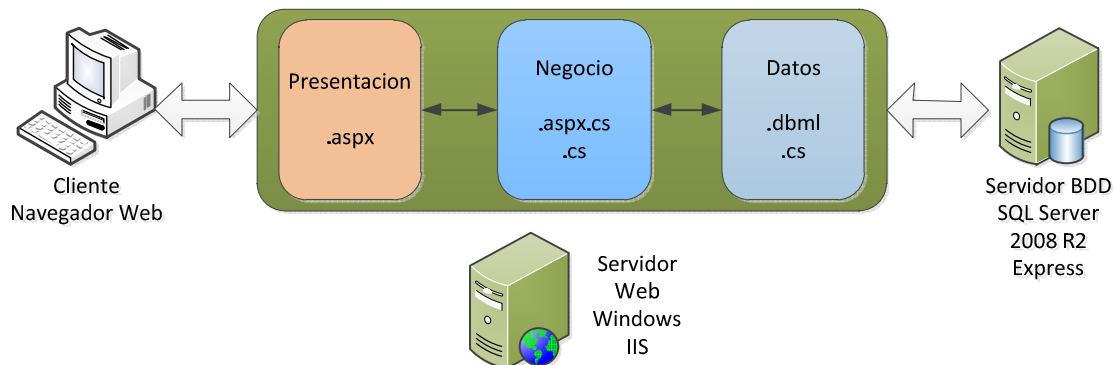
**Figura 3-4 Caracterización de la Audiencia (Caso de Estudio)**  
**Realizado por:** Carlos Fabara

### 3.2.3. DISEÑO

#### 3.2.3.1. Diseño de la Arquitectura

A nivel arquitectónico el sistema del caso de estudio utilizará el modelo de 3 capas, el cual es generalmente utilizado para desarrollos web, se utilizarán tecnologías Microsoft para su implementación, el sistema será construido utilizando el framework ASP.Net 4 con el patrón Webforms y el lenguaje C#, se implantará sobre un servidor web Internet Information Services y tendrá como

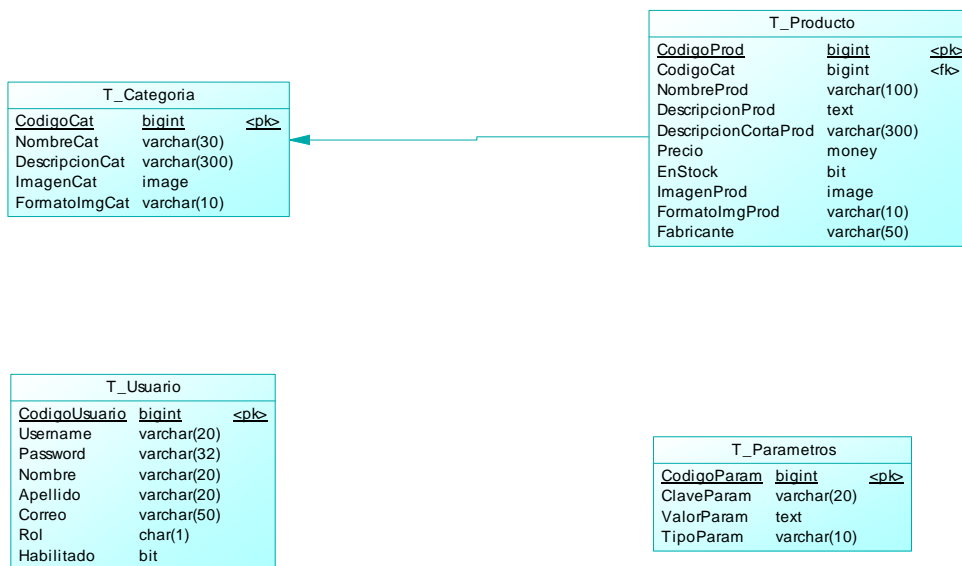
repositorio de Datos una Base de Datos en SQL Server 2008 R2 Express, como se puede observar en la Figura 3-5.



**Figura 3-5 Diagrama de Arquitectura**  
Realizado por: Carlos Fabara

### 3.2.3.2. Diseño de Datos y Lógica

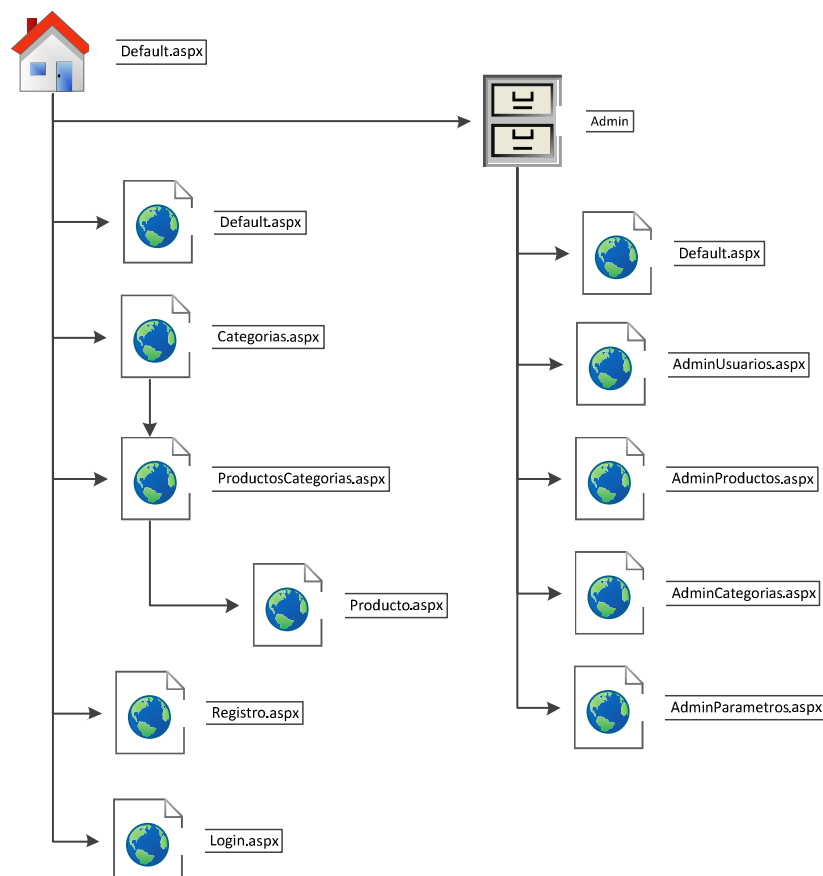
El diagrama de datos que se utilizará en el caso de estudio se representa en la Figura 3-6. Es un modelo de datos simple pero permite almacenar la información de las categorías con sus respectivos productos, la información de usuarios y los parámetros del sistema.



**Figura 3-6 Modelo de Datos (Caso de Estudio)**  
Realizado por: Carlos Fabara

### 3.2.3.3. Diseño de la Navegación

La navegación se plasma en un mapa del sitio el cual brinda la información necesaria acerca de las interfaces que se crearán y sus posibles caminos de navegación. El mapa del navegación para el caso de estudio se muestra en la Figura 3-7.

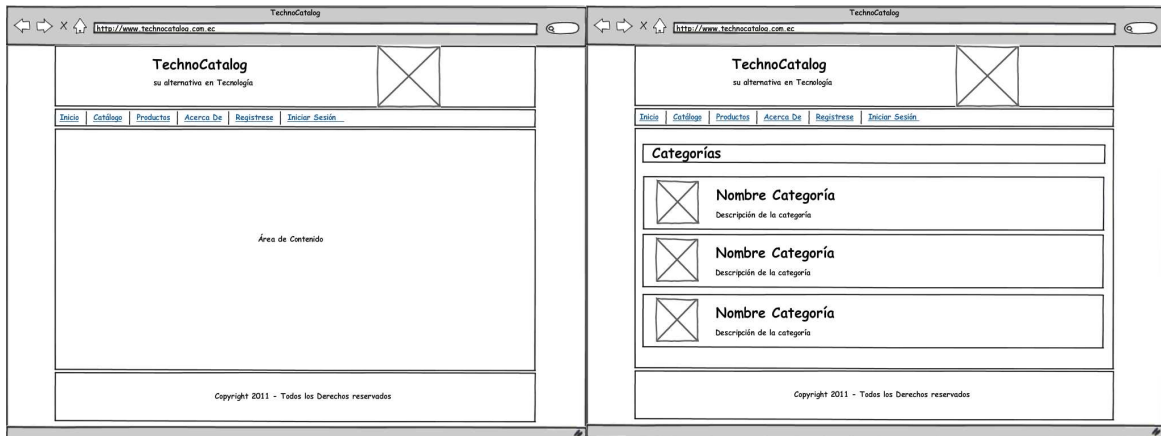


**Figura 3-7 Mapa del Sitio (Caso de Estudio)**  
Realizado por: Carlos Fabara

### 3.2.3.4. Diseño Gráfico y de Interfaces

Para la realización de los wireframes se utilizó la herramienta Balsamiq Mockups, ya que es una de las más populares para realizar esta tarea. Se decidió enmarcar al catálogo dentro de un patrón de interacción familiar, por lo que se seleccionó el patrón que incorpora un encabezado con información del sitio y una barra de

navegación, un área de contenido que ocupe gran parte de la superficie visible del navegador y finalmente un pie de página. Este diseño se puede observar en la Figura 3-8.



**Figura 3-8 Wireframes (Caso de Estudio)**  
Realizado por: Carlos Fabara

### 3.2.3.5. Selección de Componentes

**JQuery.-** Se utilizará para añadir efectos y mejorar la interactividad de las páginas del sistema. [<http://jquery.com>]

**JQuery Expander Plugin.-** Permitirá ocultar el texto en caso de que sea muy largo brindando la posibilidad de mostrarlo si el usuario lo desea. [<http://plugins.learningjquery.com/expander/>]

**uEditor.-** Editor de texto enriquecido para implementar en la página de administración del producto, para la redacción de la descripción del producto. [<http://www.upian.com/upiansource/ueditor/en>]

**Javascript MD5.-** Librería que calcula un hash md5 del lado del cliente para implementar en la página de administración de usuarios. [<http://pajhome.org.uk/crypt/md5/>]

### 3.2.4. IMPLEMENTACIÓN

#### 3.2.4.1. Selección de Herramientas

**Framework:** .Net Framework 4

**Framework Web:** ASP.Net 4

**Entorno de Desarrollo:** Visual Studio 2010 Ultimate

**Editor Imágenes:** Adobe Fireworks CS5

**Herramienta Wireframing:** Balsamiq Mockups

**Diseño Web:** Artisteer 2

**Herramienta CASE:** Power Designer 15

**Servidor Web:** Internet Information Services 7.5

**Base de Datos:** SQL Server 2008 R2 Express

**Control de Versiones:** UberSVN

**Herramienta Diagramación:** Microsoft Visio 2010

**Modelamiento UML:** Visual Studio 2010 Ultimate

#### 3.2.4.2. Construcción

El flujo de gestión del proyecto determinó, en base a los requerimientos que se realizarán 3 iteraciones, en cada una de las cuales se realizará una liberación del sistema, las 2 primeras liberaciones serán internas y la última será una liberación externa. A continuación se muestran las iteraciones con la descripción de que se realizó en cada una de ellas

##### *Iteración 1 – Sitio Base*

Una de las fortalezas de las metodologías ágiles es que fomentan la liberación continua de código funcional. Por este motivo se decidió que la forma más clara en la que el cliente observaría el avance sería teniendo su sitio base o informativo lo más pronto posible.



Los objetivos de esta iteración fueron:

- Crear el código HTML, CSS, Scripts y Recursos a partir de los Mockup
- Crear la Página Principal
- Crear una Página En Construcción

Para agilizar la construcción del diseño en base a los wireframes se utilizó el software Artisteer, que permite de forma visual (WYSIWYG) generar la plantilla para el sitio, ver Figura 3-9. Tras obtener el diseño fue necesario adaptarlo a la estructura de archivos y prepararla como una Master Page las cuales son plantillas en ASP.Net.



**Figura 3-9 Construcción Plantilla del sistema**  
Realizado por: Carlos Fabara

Tras crear la plantilla para las páginas del sistema se procede a crear la página principal del sistema y una página de “En construcción” donde se va a redirigir el resto de hipervínculos del menú aún no implementados.

### *Iteración 2 – Front Office*

Tras el primer entregable se consideró necesario brindar las funcionalidades de catálogo. Para este fin se creó la base de datos, donde se ingresó la información

de los productos. Sobre estos datos se construyeron las pantallas y sus respectivas funcionalidades.

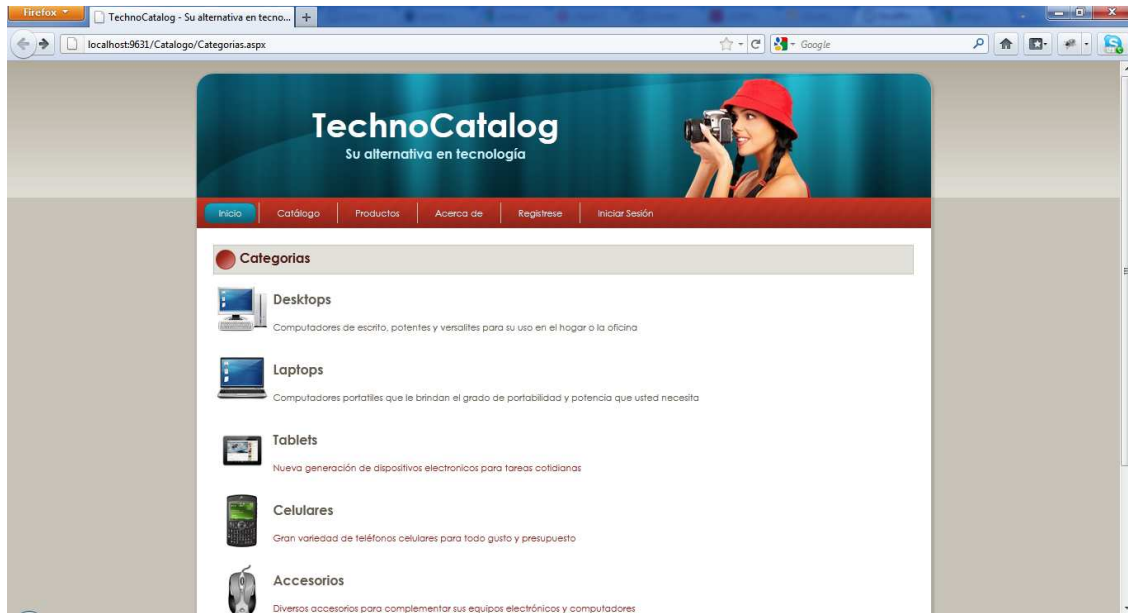
Los objetivos de esta iteración fueron:

- Crear Modelo de Datos ORM (.dbml)
- Crear Página Categorías
- Crear Página Productos por Categoría
- Crear Página Información Productos
- Crear Páginas de Autenticación y Registro



**Figura 3-10 Modelo de datos implementado**  
Elaborado por: Carlos Fabara

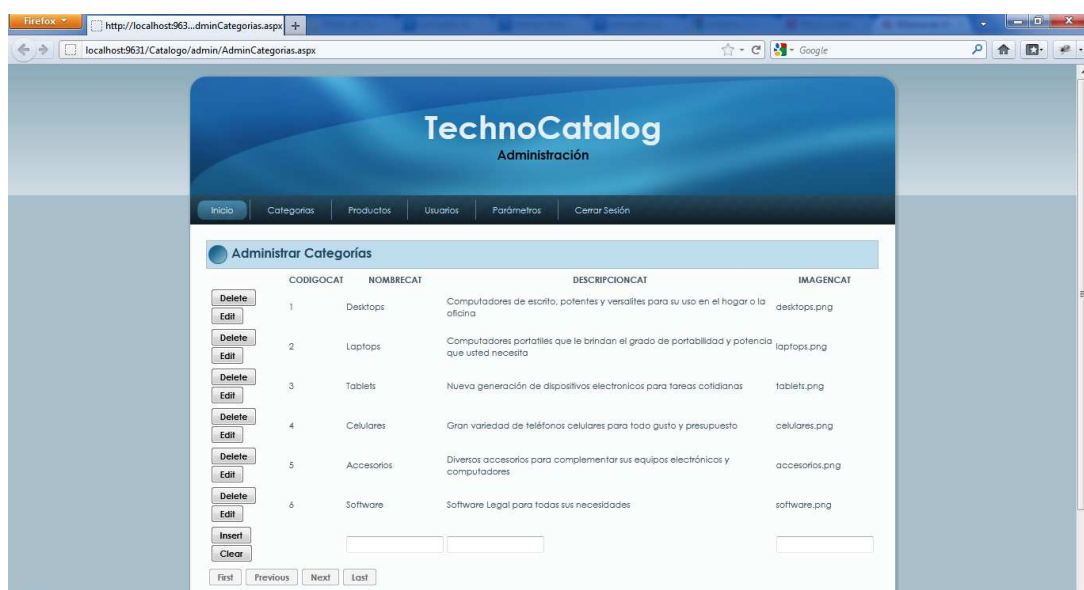
Al finalizar esta iteración se obtuvo las funcionalidades que los usuarios accederían y el sistema se encontraba operativo, aunque de momento con datos ingresados manualmente.



**Figura 3-11 Captura del sistema de catálogo en funcionamiento**  
Realizado por: Carlos Fabara

### *Iteración 3 – Back Office*

La tercera iteración buscaba crear las interfaces y funcionalidades para la administración del catálogo de productos. De esta manera se gestionaría la información ingresada previamente y se otorgaba la posibilidad de ingresar nuevos productos, categorías, usuarios, etc.



**Figura 3-12 Captura de la sección administrativa del caso de estudio**  
Realizado por: Carlos Fabara

Los objetivos de esta iteración fueron:

- Crear Página(s) Administración Usuarios
- Crear Página(s) Administración Categorías
- Crear Página(s) Administración Productos
- Crear Página(s) Configuración del Sistema

De esta manera tras tres iteraciones se concluyó la construcción del caso de estudio.

### 3.2.5. PRUEBAS

Para las pruebas se utilizaron los mecanismos y documentos descritos en el capítulo 2, a continuación se muestra la bitácora de pruebas y la bitácora de errores.

#### 3.2.5.1. Bitácora de Pruebas

En la Figura 3-13, se muestra la bitácora de pruebas que se generó durante la realización del caso de estudio.

BITÁCORA DE PRUEBAS						
<b>Proyecto:</b>		TechnoCatalog				
Código	Nombre	Descripción	Tipo	Estado	Resultado	Observaciones
PR001	Compatibilidad Navegadores	Revisar que las páginas generadas son compatibles entre los navegadores Internet Explorer, Firefox, Google Chrome	Funcional	Realizada	Exitosa	No se encontraron problemas de compatibilidad
PR002	Encontrar hipervínculos rotos	Probar que todos los hipervínculos expuestos en el menú se redirijan a páginas existentes o a la página de "En construcción"	Funcional	Realizada	(Fallida) Exitosa	Error ER001. SOLUCIONADO

PR003	Probar un flujo de consulta	Seleccionar una categoría, un producto de la misma y acceder a sus detalles	Integración	Realizada	Exitosa	El funcionamiento fue correcto.
PR004	Precios ocultos para no registrados	Verificar que el precio se muestre solo para clientes registrados.	Funcional	Realizada	Exitosa	El funcionamiento fue correcto.
PR005	Probar Producto.aspx sin parámetros	Probar que ocurre si no se ingresan parámetros en la URL	Integración	Realizada	(Fallida) Exitosa	Error ER004 SOLUCIONADO
PR006	Probar ProductosCategorias.aspx sin parámetros	Probar que ocurre si no se ingresan parámetros en la URL	Integración	Realizada	(Fallida) Exitosa	Error ER003 SOLUCIONADO
PR007	Probar Autenticación y Registro	Probar mecanismos de autenticación y registro	Funcional	Realizada	(Fallida) Exitosa	Error ER002 SOLUCIONADO
PR008	Probar Seguridad	Probar que ocurre al ingresar por URL a las páginas que requieren contraseña	No Funcional	Realizada	Exitosa	El funcionamiento fue correcto.
PR009	Probar la gestión de productos	Realizar operaciones CRUD en las interfaces de administración de productos	Funcional	Realizada	(Fallida) Exitosa	Error ER005 SOLUCIONADO
PR010	Probar la gestión de categorías	Realizar operaciones CRUD en las interfaces de administración de categorías	Funcional	Realizada	Exitosa	No se encontraron problemas.
PR011	Probar la gestión de usuarios	Realizar operaciones CRUD en las interfaces de administración de usuarios	Funcional	Realizada	Exitosa	No se encontraron problemas.
PR012	Probar la gestión de parámetros	Realizar operaciones CRUD en las interfaces de administración de parámetros	Funcional	Realizada	Exitosa	No se encontraron problemas.
PR013	Redirección a Pantallas Administrativas	Validar que se redirija a la sección administrativa en caso de que se autentique un usuario administrador	Integración	Realizada	Exitosa	No se encontraron problemas.
PR014	Flujo Completo	Crear una categoría nueva, un producto nuevo y consultarlo desde el front end	Aceptación	Realizada	Exitosa	No se encontraron problemas.

**Figura 3-13 Bitácora de Pruebas (Caso de Estudio)**  
Realizado por: Carlos Fabara

### 3.2.5.2. Bitácora de Errores

En la Figura 3-14, se muestra la bitacora de errores que se generó durante la realización del caso de estudio.

BITÁCORA DE ERRORES					
Proyecto:		TechnoCatalog			
Código	Importancia	Nombre	Descripción	Estado	Solución
ER001	Baja	Enlace Roto	Existía un enlace roto a un feed RSS no existente	Corregido	Eliminado enlace
ER002	Media	Iniciar sesión no cambia	El mensaje de iniciar sesión no se cambia a cerrar sesión después de una autenticación exitosa	Corregido	Se cambió el evento en el que se cambiaba el texto
ER003	Media	URL sin parámetros	Cuando se ingresa a ProductosCategorias.aspx sin parámetros en la URL da un error	Corregido	Redirección a Categorias.aspx
ER004	Media	URL sin parámetros	Cuando se ingresa a Producto.aspx sin parámetros en la URL da un error	Corregido	Redirección a Categorias.aspx
ER005	Alta	CRUD en Productos	No se pueden realizar operaciones CRUD en la pantalla de administración de productos	Corregido	Corrigió la definición del ListView y el problema se solucionó

**Figura 3-14 Bitácora de Errores (Caso de Estudio)**  
Realizado por: Carlos Fabara

### 3.2.6. LIBERACIÓN

#### 3.2.6.1. Liberaciones Internas

Para cada una de las liberaciones internas se actualizo el documento de Listado de Cambios. Las liberaciones consistieron en la página principal y el Front Office del sistema.

#### 3.2.6.2. Liberación Externa

En la tercera iteración se realizó una liberación externa. Tras esta iteración se disponía de un sistema completo con interfaces para la administración del catálogo. La documentación realizada se adjunta como anexos.

**Listado de Cambios.-** El documento con el listado de cambios se muestra en la Figura 3-15.

**Nombre del Proyecto:** TechnoCatalog

**Listado de Cambios**

- Versión 1.0

Liberación Externa

Implementación de la Pantalla de Administración de Categorías

Implementación de la Pantalla de Administración de Productos

Implementación de la Pantalla de Administración de Usuarios

Implementación de la Pantalla de Administración de Parámetros

Incorporado RichTextEditor en la pantalla de Administración de Productos

Incorporada interfaz para el cálculo de hash MD5 en la pantalla de Administración de Usuarios

- Versión 0.6

Liberación Interna

Creación Script BDD

Implementación de la Pantalla de Categorías

Implementación de la Pantalla de Productos por Categoría

Implementación de la Pantalla de Información de Productos

Implementación del Registro de Usuarios

Implementación de la Autenticación

Implementación del Precio Visible, solo para usuarios registrados

- Versión 0.3

Liberación Interna

Creación de la estructura de carpetas básica del proyecto

Creación de la Plantilla

Incorporación de Hojas de Estilos y librerías Javascript

Implementación de la Página Principal

Vínculos a la página En Construcción en el menú

**Figura 3-15 Listado de Cambios (Caso de Estudio)**  
**Realizado por:** Carlos Fabara

**Manual de Instalación.-** El manual de instalación contiene la información necesaria para la instalación del sistema, este incluye los prerequisites y los pasos para la instalación. Este documento se incluye en el Anexo 1.

**Manual de Usuario.-** El documento del manual de usuario contiene la información acerca del funcionamiento y operación del sistema. Este documento se incluye en el Anexo 2.

**Producto Software.-** El producto software al ser de tamaño reducido y al no requerirse una distribución en un medio físico, se empaquetaron en un archivo comprimido: los archivos del sistema, el script de la base de datos y los manuales.

### **3.2.7. MANTENIMIENTO Y EVOLUCIÓN**

#### **3.2.7.1. Futuras Características**

- Añadir componentes sociales al sitio para compartir información fácilmente.
- Añadir páginas para la gestión del perfil del usuario.
- Añadir listas de deseo (wish list) para que los usuarios puedan realizar el seguimiento de productos.
- Implementar el contenido del manual de usuario como una página del sistema.

### **3.2.8. FLUJO DE GESTIÓN DEL PROYECTO**

El flujo de gestión del proyecto es el encargado principalmente de las tareas y su organización. En la Figura 3-16, se detallan las tareas principales que se incorporaron en la bitácora para cada una de las fases.

El proyecto se desarrolló durante los fines de semana a partir del 6 de agosto de 2011, con una carga de trabajo diaria de 4 horas.



<b>BITÁCORA DE TAREAS</b>							
<b>Proyecto:</b>	TechnoCatalog						
<b>Fase</b>	<b>Tarea</b>	<b>Descripción</b>	<b>Estado</b>	<b>Asignado</b>	<b>Fecha Inicio</b>	<b>Fecha Fin Estimada</b>	<b>Fecha Fin</b>
Conceptualización	Definir problema	Definir el problema que se solventará con el caso de estudio	Realizada	CFA	06/08/2011	06/08/2011	06/08/2011
Conceptualización	Redactar modelado	Redactar el documento de modelado de negocio	Realizada	CFA	07/08/2011	07/08/2011	07/08/2011
Conceptualización	Analizar viabilidad	Analizar la viabilidad del caso de estudio	Realizada	CFA	13/08/2011	13/08/2011	13/08/2011
Requerimientos	Analizar audiencia	Analizar la audiencia que utilizará el sistema	Realizada	CFA	13/08/2011	13/08/2011	13/08/2011
Requerimientos	Casos de uso	Modelar los casos de uso en base al documento de modelado	Realizada	CFA	14/08/2011	14/08/2011	14/08/2011
Requerimientos	Diagrama casos de uso	Crear el diagrama de casos de uso	Realizada	CFA	20/08/2011	20/08/2011	20/08/2011
Requerimientos	Especificar casos de uso	Redactar el documento de especificación de casos de uso	Realizada	CFA	20/08/2011	21/08/2011	21/08/2011
Requerimientos	Caracterizar Audiencia	Redactar el documento de caracterización de la audiencia	Realizada	CFA	21/08/2011	21/08/2011	21/08/2011
Diseño	Análisis técnico	Analizar alternativas para la implementación del sistema	Realizada	CFA	27/08/2011	27/08/2011	27/08/2011
Diseño	Arquitectura	Diseñar la arquitectura del sistema e plasmarla en un diagrama	Realizada	CFA	27/08/2011	27/08/2011	27/08/2011
Diseño	Modelo de datos	Diseñar un modelo de datos que cumpla con los requerimientos	Realizada	CFA	28/08/2011	28/08/2011	28/08/2011
Diseño	Mapa del Sitio	Crear un mapa del sitio que demuestre las rutas de navegación	Realizada	CFA	03/09/2011	03/09/2011	03/09/2011
Diseño	Mockups de Interfaces	Crear los mockups para las interfaces propuestas en el mapa del sitio	Realizada	CFA	03/09/2011	03/09/2011	04/09/2011
Diseño	Seleccionar Componentes	Seleccionar las componentes que se reutilizaran en el sistema.	Realizada	CFA	04/09/2011	04/09/2011	04/09/2011
Implementación	Generar plantilla	Generar una plantilla para el sistema	Realizada	CFA	10/09/2011	10/09/2011	10/09/2011
Implementación	Adaptar plantilla	Adaptar la plantilla para convertirla en una Master Page	Realizada	CFA	10/09/2011	10/09/2011	17/09/2011

Implementación	Construir página "En Construcción"	Construir página "En Construcción"	Realizada	CFA	17/09/2011	17/09/2011	17/09/2011
Implementación	Construir página Principal	Construir página Principal	Realizada	CFA	17/09/2011	17/09/2011	18/09/2011
Implementación	Implementar la BDD	Crear un script a partir del modelo de datos y ejecutarlo	Realizada	CFA	17/09/2011	17/09/2011	17/09/2011
Implementación	Cargar Datos a la BDD	Cargar datos de prueba a la BDD	Realizada	CFA	24/09/2011	24/09/2011	01/10/2011
Implementación	Crear Lógica de Acceso a Datos	Crear Lógica de Acceso a Datos para las pantallas de cliente.	Realizada	CFA	25/09/2011	01/09/2011	02/09/2011
Implementación	Construir página Categorías	Construir la página de categorías basándose en los mockups previos	Realizada	CFA	08/10/2011	09/10/2011	09/10/2011
Implementación	Construir página Productos por Categoría	Construir página Productos por Categoría	Realizada	CFA	15/10/2011	15/10/2011	15/10/2011
Implementación	Construir página Producto	Construir página Producto	Realizada	CFA	16/10/2011	16/10/2011	16/10/2011
Implementación	Construir página Autenticación	Construir página Autenticación	Realizada	CFA	22/10/2011	22/10/2011	22/10/2011
Implementación	Construir página Registro	Construir página Registro	Realizada	CFA	23/10/2011	23/10/2011	23/10/2011
Implementación	Añadir validaciones	Añadir validaciones a la página de autenticación y registro	Realizada	CFA	29/10/2011	29/10/2011	30/10/2011
Implementación	Crear Sección Administración	Crear la carpeta /admin donde se ubicaran las interfaces administrativas	Realizada	CFA	29/10/2011	29/10/2011	29/10/2011
Implementación	Crear Plantilla para Administración	En base a la plantilla de cliente modificar los colores	Realizada	CFA	29/10/2011	29/10/2011	29/10/2011
Implementación	Crear MasterPage para Administración	Adaptar la nueva plantilla a una Master Page	Realizada	CFA	30/10/2011	05/11/2011	05/11/2011
Implementación	Crear Gestión de Usuarios	Crear Gestión de Usuarios	Realizada	CFA	05/11/2011	06/11/2011	06/11/2011
Implementación	Crear Gestión de Parámetros	Crear Gestión de Parámetros	Realizada	CFA	12/11/2011	12/11/2011	12/11/2011
Implementación	Crear Gestión de Categorías	Crear Gestión de Categorías	Realizada	CFA	13/11/2011	13/11/2011	13/11/2011
Implementación	Crear Gestión de Productos	Crear Gestión de Productos	Realizada	CFA	19/11/2011	19/11/2011	19/11/2011
Implementación	Implementar	Implementar RichTextBox en Admin. Productos	Realizada	CFA	20/11/2011	20/11/2011	20/11/2011

	RichTextBox en Admin. Productos						
Implementación	Implementar Javascript MD5 en Admin, Clientes	Implementar Javascript MD5 en Admin, Clientes	Realizada	CFA	20/11/2011	20/11/2011	20/11/2011
Implementación	Modificar lógica de acceso a datos	Modificar lógica de acceso a datos	Realizada	CFA	20/11/2011	20/11/2011	20/11/2011
Pruebas	Ejecutar prueba PR001	Ejecutar prueba PR001	Realizada	CFA	29/10/2011	29/10/2011	29/10/2011
Pruebas	Ejecutar prueba PR002	Ejecutar prueba PR002	Realizada	CFA	29/10/2011	29/10/2011	29/10/2011
Pruebas	Ejecutar prueba PR003	Ejecutar prueba PR003	Realizada	CFA	29/10/2011	29/10/2011	29/10/2011
Pruebas	Ejecutar prueba PR004	Ejecutar prueba PR004	Realizada	CFA	30/10/2011	30/10/2011	30/10/2011
Pruebas	Ejecutar prueba PR005	Ejecutar prueba PR005	Realizada	CFA	30/10/2011	30/10/2011	30/10/2011
Pruebas	Ejecutar prueba PR006	Ejecutar prueba PR006	Realizada	CFA	30/10/2011	30/10/2011	30/10/2011
Pruebas	Ejecutar prueba PR007	Ejecutar prueba PR007	Realizada	CFA	05/11/2011	05/11/2011	05/11/2011
Pruebas	Ejecutar prueba PR008	Ejecutar prueba PR008	Realizada	CFA	05/11/2011	05/11/2011	05/11/2011
Pruebas	Ejecutar prueba PR009	Ejecutar prueba PR009	Realizada	CFA	26/11/2011	26/11/2011	26/11/2011
Pruebas	Ejecutar prueba PR010	Ejecutar prueba PR010	Realizada	CFA	26/11/2011	26/11/2011	26/11/2011
Pruebas	Ejecutar prueba PR011	Ejecutar prueba PR011	Realizada	CFA	26/11/2011	26/11/2011	26/11/2011
Pruebas	Ejecutar prueba PR012	Ejecutar prueba PR012	Realizada	CFA	27/11/2011	27/11/2011	27/11/2011
Pruebas	Ejecutar prueba PR013	Ejecutar prueba PR013	Realizada	CFA	27/11/2011	27/11/2011	27/11/2011
Pruebas	Ejecutar prueba PR014	Ejecutar prueba PR014	Realizada	CFA	27/11/2011	27/11/2011	27/11/2011
Liberación	Crear el listado de Cambios	Crear el listado de Cambios	Realizada	CFA	03/12/2011	03/12/2011	03/12/2011
Liberación	Crear el manual de instalación	Crear el manual de instalación	Realizada	CFA	03/12/2011	03/12/2011	10/12/2011
Liberación	Crear el manual de usuario	Crear el manual de usuario	Realizada	CFA	03/12/2011	04/12/2011	11/12/2011
Liberación	Empaquetar el producto a entregar	Empaquetar el producto a entregar	Realizada	CFA	17/12/2011	17/12/2011	17/12/2011

**Figura 3-16 Bitácora de Tareas (Caso de Estudio)**  
**Realizado por: Carlos Fabara**

### **3.2.9. FLUJO DE GESTIÓN DE CALIDAD**

Los aspectos que se consideraron durante el flujo de gestión de calidad fueron los mismos expuestos en el capítulo 2.4.2. Al considerar a la calidad como parte esencial del proceso de desarrollo se pudo lograr lo siguiente:

- Se evitó que los requerimientos sean ambiguos.
- Se consiguió que los documentos sean concisos y contengan toda la información necesaria.
- Se validó que el software cumpla con gran parte de los estándares del lenguaje C#, HTML y CSS.
- Se obtuvo un sistema probado que garantizaba la calidad del mismo.

## **3.3. ANÁLISIS DE RESULTADOS**

### **3.3.1. ANÁLISIS INTERNO**

En el Capítulo 2 se definió un conjunto de principios que serían las bases del proceso PRODEWEB, tras haberse aplicado la propuesta en el desarrollo de un caso de estudio, se procederá a evaluar el grado de cumplimiento como un criterio del análisis de resultados. Los principios que se expresaron en la propuesta fueron:

- Dirigido por Datos
- Promover el desarrollo rápido de aplicaciones
- Ofrecer entregas y liberaciones continuas
- Incorporar varios principios de desarrollo ágil
- Estar diseñado para grupos pequeños y proyectos de mediano alcance

## **Dirigido por datos**

El modelo de datos fue una pieza clave durante el desarrollo del caso de estudio. El análisis de la información que manejaría el sistema, permitió obtener un modelo de datos capaz de cumplir los requerimientos realizados. Adicionalmente el modelo tuvo impacto en el diseño de las interfaces, ya que se buscó la mejor manera y patrón para desplegar la información.

## **Promover el desarrollo rápido de aplicaciones**

El proceso promueve la reutilización de software y el uso de herramientas, como un mecanismo de acelerar y optimizar el tiempo de desarrollo. Por citar unos ejemplos:

- El desarrollo de interfaces a través del editor WYSIWYG de Visual Studio 2010, reduce considerablemente el tiempo necesario para su construcción, en comparación con el tiempo que toma escribir manualmente el código de las mismas.
- El uso de librerías Javascript abstraen al programador de las complejidades de trabajar con el DOM (Document Object Model), garantizan un funcionamiento prácticamente homogéneo entre los distintos navegadores web y permitieron la reutilización de componentes, algoritmos y controles.

## **Ofrecer entregas y liberaciones continuas**

El caso de estudio tuvo 3 liberaciones, de las cuales dos fueron internas y una externa. Cada una de las liberaciones complementaba con nuevas funcionalidades al producto de la anterior. El resultado de cada una de las iteraciones se consideraba un producto funcional, capaz de ser desplegado en un entorno de producción. De esta manera el proceso permitió la entrega de un sistema de catálogo de productos en un corto periodo de tiempo.

## **Incorporar varios principios de desarrollo ágil**

Como se indica en Capítulo 1.1.3.2 los principios en los que se basan las metodologías ágiles son:

1. Se da mayor importancia a los individuos y su interacción, por encima de los procesos y las herramientas.
2. Se da mayor importancia a que el software funcione antes que a la generación de documentación exhaustiva.
3. Se promueve la colaboración con el cliente, por encima de la negociación contractual.
4. Se prefiere la respuesta al cambio, por encima del seguimiento de un plan.

En el proceso propuesto se enfatiza en la interacción entre los individuos como un mecanismo efectivo de difusión de la información, ya sea durante la conceptualización del sistema o durante todas las etapas del desarrollo.

La importancia de un software que funciona radica en que al trabajar con grupos pequeños y proyectos no muy amplios, la documentación en vez de ayudar puede convertirse en un lastre que retrase al proyecto. Es importante recalcar que el proceso propuesto no elimina la documentación sino que la sintetiza en los que se consideran necesarios para llevar a buen término un proyecto de software.

La respuesta al cambio se enfatiza en el proceso al permitir regresar a etapas anteriores en caso de ser necesaria alguna modificación adicionalmente aunque suene contradictorio se incorporó el seguimiento de un plan como una característica esencial ya que uno de los inconvenientes de las metodologías ágiles es el hecho de dejar al albedrío del usuario el procedimiento necesario para realizar un desarrollo como en el caso de Scrum.

### **Estar diseñado para grupos pequeños y proyectos de mediano alcance**

El proceso fue pensado para grupos de trabajo pequeños en los que cada uno de los integrantes del equipo puede cumplir una o más funciones. Por este motivo se enfatizó en el uso de principios ágiles ya que al disponer de poco personal es necesario optimizar los recursos y el tiempo de trabajo. Para el caso de estudio se realizó un desarrollo unipersonal, ya que el autor de este trabajo realizó la totalidad de los roles.

#### **3.3.2. ANÁLISIS EXTERNO**

Con el afán de conocer la opinión y validar el proceso propuesto, se realizó una inducción al contenido de la propuesta a un conjunto de profesionales y académicos del medio, que tienen relación a las áreas de desarrollo web e ingeniería de software.

Posteriormente a la inducción se realizó una encuesta de opinión sobre la propuesta. El formato de la encuesta se muestra en la Figura 3-17.

Previa a la realización de la encuesta se realizó una inducción del contenido de la propuesta al encuestado/a, de manera que estuviera en condiciones de contestar las preguntas de la encuesta, que principalmente se encontraban enfocadas a su opinión del proceso PRODEWEB.

Nombre: \_\_\_\_\_

**Tipo Encuestado:**

Profesional       Docente

Cargo que desempeña

**1) Posee experiencia en el desarrollo de sistemas web:**

Si                     No

*Si su respuesta fue SI pase a la pregunta 2), si su respuesta fue NO pase a la pregunta 3)*

**2) Con qué frecuencia ha utilizado metodologías para sus desarrollos web:**

Nunca             Algunas Veces             Casi Siempre             Siempre

**3) En la siguiente pregunta marque con una (X) de acuerdo a la siguiente escala, su nivel de acuerdo con las fases del proceso propuesto y su documentación, de ser necesario escriba sus comentarios en la sección de observaciones:**

- 1 - Completamente en desacuerdo
- 2 - En desacuerdo
- 3 - De acuerdo
- 4 - Completamente de acuerdo

Fase	1	2	3	4	Observaciones
Conceptualización					
Requerimientos					
Diseño					
Implementación					
Pruebas					
Liberación					
Mantenimiento y Evolución					
(Flujo) Gestión del Proyecto					
(Flujo) Gestión de Calidad					

**4) Usted usaría este proceso**



<input type="checkbox"/> Si <input type="checkbox"/> No  ¿Por Qué?  <hr/> <hr/> <p><b>5) Usted recomendaría este proceso</b></p> <input type="checkbox"/> Si <input type="checkbox"/> No
--

**Figura 3-17 Formato encuesta realizada**  
**Realizado por: Carlos Fabara**

Los resultados que se obtuvieron fueron los siguientes:

Se entrevistaron un total de 20 personas, de las cuales el 70% son profesionales que trabajan en empresas del medio y el 30% son docentes de alguna de las principales universidades de la ciudad.

El 100% de los encuestados expresaron tener experiencia en desarrollo de sistemas web. Y de esta forma se puede garantizar que las personas encuestadas tienen experiencia en los temas tratados en el presente trabajo.

Al consultar a los encuestados sobre la frecuencia con la que ellos han utilizado metodologías en sus desarrollos web. Se obtuvieron los resultados que se muestran en la Tabla 3-1. Es importante recalcar que aunque la gran mayoría de encuestados utilizan con bastante frecuencia metodologías en sus desarrollos web, existió un porcentaje del 41.18% que expresaron que su frecuencia de uso era "Algunas Veces", por este motivo la propuesta PRODEWEB tiene como finalidad fomentar el uso de un proceso formal de desarrollo de software entre los usuarios que no se encuentran utilizando o que no utilizan constantemente un proceso o metodología.

Frecuencia de Uso	Porcentaje
Nunca	0%
Algunas Veces	41.18%
Casi Siempre	41.18%
Siempre	17.65%

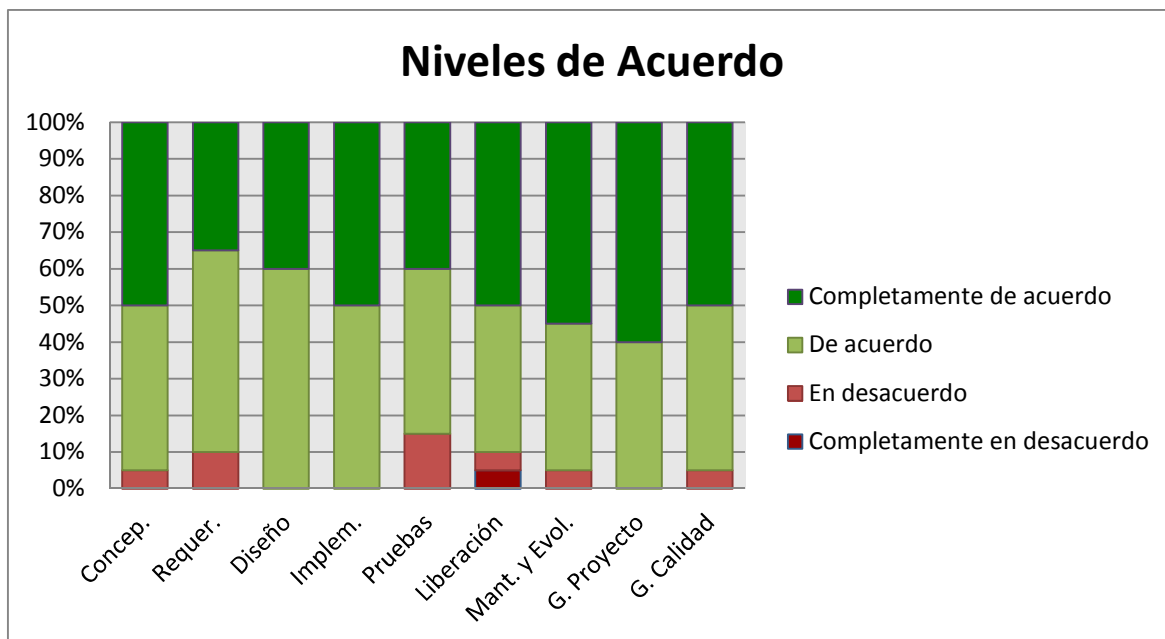
**Tabla 3-1 Uso de metodologías en desarrollo web**  
Realizado por: Carlos Fabara

La encuesta contenía una sección dedicada a conocer el nivel de acuerdo que tenía el encuestado con cada una de las fases del proceso PRODEWEB. Los resultados que se obtuvieron se detallan en la Tabla 3-2.

Fase	Nivel de acuerdo (%)			
	Completamente en desacuerdo	En desacuerdo	De acuerdo	Completamente de acuerdo
Conceptualización	0,00%	5,00%	45,00%	50,00%
Requerimientos	0,00%	10,00%	55,00%	35,00%
Diseño	0,00%	0,00%	60,00%	40,00%
Implementación	0,00%	0,00%	50,00%	50,00%
Pruebas	0,00%	15,00%	45,00%	40,00%
Liberación	5,00%	5,00%	40,00%	50,00%
Mantenimiento y Evolución	0,00%	5,00%	40,00%	55,00%
Gestión del Proyecto	0,00%	0,00%	40,00%	60,00%
Gestión de Calidad	0,00%	5,00%	45,00%	50,00%

**Tabla 3-2 Niveles de Acuerdo con la propuesta**  
Realizado por: Carlos Fabara

En la Figura 3-18 se sintetizan los resultados en un gráfico apilado de barras y como se puede observar la opinión de las fases del proceso fue muy positiva, desplegadas en color verde oscuro y verde claro todas aquellas opiniones que se encontraban “Completamente de acuerdo” o “De acuerdo”.



**Figura 3-18 Niveles de Acuerdo con la propuesta**  
Realizado por: Carlos Fabara

En la Figura 3-18 se sintetizan los resultados en un gráfico apilado de barras y como se puede observar la opinión de las fases del proceso fue muy positiva, desplegadas en color verde oscuro y verde claro todas aquellas opiniones que se encontraban “Completamente de acuerdo” o “De acuerdo”.

Se consultó además acerca de si estarían dispuestos a usar la propuesta en sus próximos desarrollos. El resultado se muestra en la Tabla 3-3.

Usaría la propuesta	Porcentaje
Si	90%
No	10%

**Tabla 3-3 Disposición a usar la propuesta**  
Realizado por: Carlos Fabara

El resultado reveló, como se resume en la Figura 3-19, que el 90% de los encuestados estarían dispuestos a utilizar el proceso, al analizar los comentarios del 10% que no la usaría, se pudo ver que expresaban que no podían utilizarla ya que en su empresa ellos disponían ya de una metodología adaptada a sus

necesidades. Sin embargo ellos estaban dispuestos a recomendar el uso de la propuesta.

Entre las principales razones del porque estarían dispuestos a utilizar el proceso los encuestados expresaron que:

- Les parecía un proceso interesante y novedoso.
- Era útil para proyectos de pequeño y mediano alcance.
- Se incorporaban criterios no considerados por otras metodologías
- La propuesta unificaba criterios de metodologías ágiles y tradicionales
- De ese modo podrían evaluarla con mayor detalle.
- Era factible de ser implementado en proyectos del mundo real.



**Figura 3-19 Disposición a usar la propuesta**  
Realizado por: Carlos Fabara

Finalmente se consultó si estarían dispuestos a recomendar el uso de la propuesta. Los resultados que se muestran en la Tabla 3-4 y se sintetizan en la Figura 3-20, reflejaron que un 85% de los encuestados, recomendarían la propuesta.

Recomendaría la propuesta	Porcentaje
Si	85%
No	15%

**Tabla 3-4 Disposición a recomendar la propuesta**  
Realizado por: Carlos Fabara

Al analizar los comentarios del 15% que no lo haría, se observó que algunas de las razones eran debido a que consideraban necesario utilizar la propuesta para poder recomendarla.



**Figura 3-20 Disposición a recomendar la propuesta**  
Realizado por: Carlos Fabara

## **CAPÍTULO 4. CONCLUSIONES Y RECOMENDACIONES**

### **4.1. CONCLUSIONES**

- El presente trabajo cumplió con los objetivos propuestos, al construir una nueva propuesta de proceso de desarrollo para sistemas web, que se basa en la metodología WebML, e incorpora criterios y procedimientos utilizados en la actualidad en el desarrollo web.
- Se seleccionó a WebML como metodología base para el proceso PRODEWEB, ya que tras el análisis se determinó que era una de las metodologías web que abarcaba una mayor cantidad de fases dentro del ciclo de vida de desarrollo de software.
- La documentación y entregables del proceso PRODEWEB buscan ser simples y fáciles de generar, de manera que puedan ser utilizados en el mayor número de casos y escenarios posibles.
- La adaptabilidad que poseen las metodologías ágiles es una de las características incorporadas al proceso PRODEWEB, ya que permite acoplarse a las distintas realidades de empresas o individuos que realizan desarrollo web.
- PRODEWEB busca convertirse en una alternativa viable para el desarrollo de sistemas web, al proporcionar una propuesta que se encuentra en el punto medio entre la formalidad de las metodologías tradicionales y la versatilidad de las metodologías web, además de incorporar las tendencias actuales utilizadas en el medio.
- El uso del proceso PRODEWEB en el desarrollo del caso de estudio, permitió validar la viabilidad del uso de la propuesta en un entorno real. Como

resultado se obtuvo un sistema funcional, de buena calidad y que cumplía los requerimientos planteados.

- A través de las encuestas realizadas, se pudo conocer que gran parte de los encuestados, se encontraban de acuerdo con el contenido de la propuesta y que estarían dispuestos a utilizar y/o recomendarla.
- A través de las encuestas, se recibió una retroalimentación muy importante de parte de los encuestados. Algunas de las sugerencias y observaciones realizadas fueron tomadas en cuenta en la sección de recomendaciones.

## **4.2. RECOMENDACIONES**

- La propuesta PRODEWEB es un proceso específico para el desarrollo de sistemas web, sin embargo comparte muchas similitudes con procesos tradicionales de desarrollo, sería recomendable analizar los elementos que deben ser incorporados o modificados para convertirla en un proceso orientado a todo tipo de proyectos de desarrollo de software.
- El proceso PRODEWEB se enfocó en incorporar muchos de los criterios actuales del desarrollo web, pero aun así sería recomendable profundizar en criterios emergente como son conceptos de las denominadas Web 2.0 y Web 3.0.
- Es recomendable realizar un proceso de difusión del proceso PRODEWEB, de forma que la comunidad técnica retroalimiente con opiniones y nuevos criterios, para que el proceso siga evolucionando y complementándose.
- Algunos de los encuestados coincidieron en el criterio de incorporar a la fase de diseño, una etapa dedicada al diseño de las pruebas. Adicionalmente se sugirió que se enfatice en un mecanismo para que las pruebas sirvan como herramienta de validación de requerimientos.

- Aunque se incluyen criterios de Gestión de la Configuración y Cambios en el proceso (listados de cambios y tipos de liberaciones), se sugirió durante las encuestas que estas se agrupen en una fase o flujo formal.
- Se recomienda describir con mayor detalle el flujo de gestión de calidad añadiendo criterios expuestos en alguna norma de evaluación de la calidad de software como es la norma ISO/IEC 9126.



## **GLOSARIO**

### **AJAX**

Asynchronous JavaScript And XML

### **BDD**

Base de Datos

### **CASE**

Computer Aided Software Engineering

### **CSS**

Cascading Style Sheets

### **DRY**

Don't Repeat Yourself

### **DOM**

Document Object Model

### **HTML**

HyperText Markup Language

### **IDE**

Integrated Development Enviroment

### **OOHDM**

Object Oriented Hypermedia Design Method

### **ORM**

Object-Relational mapping

### **OWL**

Ontology Web Language

### **RAD**

Rapid Application Development

### **RDF**

Resource Description Framework

### **RIA**

Rich Internet Applications

### **ROI**

Return on Investment

### **RoR**

Ruby on Rails

**SEO**

Search Engine Optimization

**SO**

Sistema Operativo

**UML**

Unified Modeling Language

**W3C**

World Wide Web Consortium

**WebML**

Web Modeling Language

**WSDM**

Web Semantic Design Methods

**WYSIWYG**

What You See Is What You Get

**XML**

Extensible Markup Language

## BIBLIOGRAFÍA

- [1] Google Chrome Team. (2010) 20 Things I learned about browsers and the web. [Online]. <http://www.20thingsilearned.com/>
- [2] World Wide Web Consortium. (2011, May) World Wide Web Consortium (W3C). [Online]. <http://www.w3.org/TR/html5/>
- [3] Justin James. (2008, Aug.) TechRepublic. [Online]. <http://www.techrepublic.com/blog/programming-and-development/html-5-editor-ian-hickson-discusses-features-pain-points-adoption-rate-and-more/718>
- [4] Kent Beck, Mike Beedle, and James Grenning. (2001) Manifiesto for Agile Software Development. [Online]. <http://agilemanifesto.org/>
- [5] Roger S. Pressman and David Lowe, *Web Engineering: A practioner's approach*. New York: McGraw-Hill, 2009.
- [6] Sven Casteleyn, Florian Daniel, Peter Dolog, and Maristella Matera, *Engineering Web Applications*. Berlin: Springer-Verlag, 2009.
- [7] Stefano Ceri, Piero Fraternali, and Aldo Bongio. (2000) 9th International World Wide Web Conference. [Online]. <http://www9.org/w9cdrom/177/177.html>
- [8] Gustavo Rossi, Oscar Pastor, Daniel Schwabe, and Luis Olsina, *Web Engineering: Modelling and Implementing Web Applications*. London: Springer-Verlag, 2008.
- [9] Fundación Wikimedia, Inc. (2011, Nov.) Wikipedia. [Online]. <http://es.wikipedia.org/wiki/Scrum>
- [10] Vikrama Dhiman. WiziQ education.online. [Online]. <http://www.wiziq.com/tutorial/12910-What-is-Scrum-Scrum-Overview-Scrum-Intro>
- [11] Luciano Moreno. (2003, Sep.) Desarrollo Web, Tu mejor ayuda para aprender a hacer webs. [Online]. <http://www.desarrolloweb.com/articulos/1278.php>
- [12] Collis. (2007, Dec.) Psdtuts+. [Online]. <http://psd.tutsplus.com/designing-tutorials/9-essential-principles-for-good-web-design/>
- [13] Peter Morville and Louis Rosenfeld, *Information Architecture for the World Wide Web*. Sebastopol: O'Reilly Media, 2007.

- [14] Theresa Neil. (2009, Jan.) THERESA NEIL. [Online]. <http://theresaneil.wordpress.com/2009/01/17/designing-web-interfaces-12-screen-patterns/>
- [15] Dan M. Brown, *Communicating Design: Developing Web Site Documentation for Design and Planning*. Berkeley: New Riders, 2011.
- [16] James Kalbach, *Designing Web Navigation*. Sebastopol: O'Reilly Media, 2007.
- [17] Dan M. Brown, *Communicating Design: Developing Web Site Documentation for Design and Planning*. New Berkley: New Riders, 2011.
- [18] 37signals, *Getting Real.*, 2006.
- [19] Luis Carlos Aceves. (2009) Slideshare. [Online]. <http://www.slideshare.net/laceves/diseo-de-informacin-navegacin-e-interfaces>

## **ANEXOS**

Los anexos que se numeran a continuación se encuentran en el disco adjunto.

**ANEXO 1** – Manual de Instalación (Caso de Estudio)

**ANEXO 2** – Manual de Usuario (Caso de Estudio)

**ANEXO 3** – Plantilla del Documento de Modelado de Negocio

**ANEXO 4** – Plantilla del Checklist de Viabilidad

**ANEXO 5** – Plantilla del Documento de Especificación de Casos de Uso

**ANEXO 6** – Plantilla del Documento de Caracterización de la Audiencia

**ANEXO 7** – Plantilla del Documento de Diseño

**ANEXO 8** – Plantilla del Listado de Cambios

**ANEXO 9** – Plantilla de la Bitácora de Pruebas

**ANEXO 10** – Plantilla de la Bitácora de Errores

**ANEXO 11** – Plantilla de la Bitácora de Tareas