



UNIVERSIDAD MICHOACANA DE SAN NICOLAS DE HIDALGO

FACULTAD DE CONTADURÍA Y CIENCIAS ADMINISTRATIVAS

**DISEÑO E IMPLEMENTACIÓN DE UNA PÁGINA WEB DINÁMICA CON
ENVÍO AUTOMÁTICO DE EMAIL.**

CASO PRÁCTICO

PARA OBTENER EL TÍTULO DE:

LICENCIADO EN INFORMÁTICA ADMINISTRATIVA

PRESENTA:

CARLOS ALBERTO PINEDA GONZÁLEZ

ASESOR:

M.A. SALVADOR A. CASANOVA VALENCIA

Morelia, Mich., Noviembre de 2011

Índice

UNIDAD 1.- PROTOCOLO DE INVESTIGACIÓN

Introducción.....	5
Definición del problema.....	6
Preguntas de investigación.....	7
Justificación.....	8
Objetivos generales.....	10
Objetivos específicos.....	10

UNIDAD 2.- MARCO TEORICO

Capítulo 1: Análisis y diseño de sistemas.....11

1.1 Tipos de Sistemas.....11

1.1.1 Sistema de procesamiento de transacciones.....	12
1.1.2 Sistemas de automatización de la oficina y sistemas de trabajo del conocimiento.....	12
1.1.3 Sistemas de información gerencial.....	13
1.1.4 Sistemas de apoyo a la toma de decisiones.....	13

1.2 Integración de las tecnologías de sistemas.....14

1.2.1 Aplicaciones de comercio electrónico y sistemas web.....	14
--	----

1.3 El ciclo de vida del desarrollo de sistemas.....15

1.3.1 Identificación de problemas, oportunidades y objetivos.....	15
1.3.2 Determinación de los requerimientos de información.....	16
1.3.3 Análisis de las necesidades del sistema.....	17
1.3.4 Diseño del sistema recomendado.....	17
1.3.5 Desarrollo y documentación del software.....	18
1.3.6 Prueba y mantenimiento del sistema.....	19
1.3.7 Implementación t evaluación del sistema.....	19

Capítulo 2: Internet	21
2.1 World Wide Web	23
2.2 Protocolos	24
2.2.1 TCP/IP	24
2.2.2 Http	25
2.2.3 ARP	26
2.2.4 FTP	27
2.2.5 SMTP	28
2.2.6 POP	29
2.3 Página Web	31
2.3.1 Lenguaje de marcado	32
2.3.1.1 HTML	33
2.3.2 Lenguaje de programación	34
2.3.2.1 PHP	35
2.3.2.1.1 Características fundamentales del lenguaje	37
2.3.2.1.2 ¿Cómo y cuándo funciona el intérprete?	38
2.3.3 Pagina Web Estática	39
2.3.4 Pagina Web Dinámica	39
2.4 Bases de datos	40
2.4.1 Modelado de datos utilizando el modelo entidad-relación	41
2.4.1.1 Uso de modelos de datos conceptuales de alto nivel para el diseño de bases de datos	41
2.4.1.2 Entidades y atributos	42

2.4.2 MySQL.....	44
2.5 Direcciones IP.....	44
2.6 URL.....	45
2.7 Dominio.....	46
2.8 DNS.....	47
2.9 Metodologías para el análisis y diseño de una sistema de información web.....	47
2.9.1 Web Site Design Method (WSDM).....	48
2.9.2 Relationship Navigational Analysis (RNA).....	49
2.9.3 Object Oriented Hypermedia Design Model (OOHDM).....	50
2.9.4 Diseño y desarrollo de sistemas de información web.....	52
Capítulo 3: Marco metodológico.....	54
3.1 Análisis preliminar para sistemas de información web.....	54
3.1.1 Definir los objetivos del sistema.....	54
3.1.2 Tipo de almacenamiento.....	54
3.1.3 Describir el tipo de usuario del sistema.....	55
3.1.4 Definir los procesos a realizar.....	55
3.2 Diseño de interfaces.....	55
3.2.1 Boceto de pantallas.....	55
3.2.2 Diccionario de datos.....	55
3.2.3 Someter a evaluación.....	56
3.3 Selección de lenguaje y desarrollo de la aplicación.....	56
3.3.1 Seleccionar plataforma y base de datos.....	56
3.3.2 Lenguaje de programación.....	56

3.3.3 Organización de las carpetas.....	57
3.5 Implementación y pruebas.....	57
3.5.1 Implementación directa.....	57
3.5.2 Implementación en paralelo.....	57
UNIDAD 3.- DESARROLLO.....	58
Modelo Relacional.....	58
Modelo Entidad-Relación.....	59
Página del Administrador.....	60
Página Principal.....	71
Conclusiones.....	75
Bibliografía.....	76

INTRODUCCIÓN

Este trabajo está hecho con la finalidad de mostrar a las personas una página web dinámica, que contiene un catálogo de una variedad de productos usados y de bajo costo. Es importante y uno de los factores resaltantes de esta página, es la promoción que se dará a la empresa que ofrecerá estos productos.

El nombre que se le asignara a esta página web es: www.empenomorelia.com, la cual debe cumplir con el objetivo de informar por medio de internet y suscripción vía email y a sus clientes sobre los artículos que están a la venta a precios accesibles.

Las herramientas que se utilizaran en esta página web seran: **Dreamweaver**, con el lenguaje de programación **PHP**, el sistema para base de datos **MySQL** y **Photoshop**, para realizar la página el apoyo fueron tutoriales.

El internet en las empresas es vital ya que pueden llegar a conocer el perfil de sus consumidores, además de ofrecer una gran diversidad de productos, ya que su oferta puede llegar a cualquier parte del mundo, la empresa también tendrá la seguridad de que los clientes que vean su publicidad estarán interesados y centrados en esa publicidad ya que no pueden hacer el zapping que permite la televisión.

Una página web diseñada para una empresa debe contar con múltiples puntos básicos como lo es el logotipo de esta, colores acorde al logotipo, etc., también el usuario debe encontrar información sobre la empresa como lo es ubicación, contacto, y los servicios o productos que se ofrecen. Un diseño basado en los anteriores puntos obtendrá como resultado mantener un contacto directo y publicitario con los clientes, también mantener constantemente informados sobre las novedades en la empresa.

Definición del Problema

La empresa respuesta rápida, es una casa de empeño ubicada en Morelia, empresas como estas aceptan artículos a cambio de préstamos de dinero, cuando las personas dejan perder sus pertenencias, estos productos deben ponerse a la venta para recuperar el dinero prestado. Dichos productos son visibles únicamente en el establecimiento de la empresa, con lo cual hay que esperar a veces un mes o más para que algún producto se venda, esto ocasiona que el producto se devalúe dicho mes siendo esto un problema para la empresa.

Debido a que las personas no visitan seguido la empresa con la intención de encontrar algún producto que se desea y con los horarios de trabajo que no permiten a todas las personas esa disponibilidad de visitar el establecimiento, esto es un problema al tiempo de ofrecer un préstamo de dinero por algún producto, ya que conociendo que no se vende rápidamente crea un miedo para poder prestar más dinero, lo cual es un inconveniente ya que al prestar menos dinero se gana menos interés.

Es muy importante que mediante un medio electrónico al alcance de cualquier persona en cualquier parte del mundo, y en la comodidad del hogar, se le facilite al cliente conocer, comparar y decidir si adquiere un producto dotándolo de toda la información sobre este, mediante imágenes, precio y características. Otra de las facilidades que se debe ofrecer a los clientes es la opción para recordarle periódicamente sobre las novedades y promociones mediante correo electrónico al permitir suscripción en la página.

Preguntas de Investigación

¿Qué factores hacen interesantes al usuario el uso de una página web?

¿Qué características facilitan el uso de una página web?

¿Qué metodologías podemos utilizar en el desarrollo de una página web?

Justificación

El internet se ha vuelto una herramienta necesaria para llevar a cabo una compra, es por esto que mediante una página web, Empeño Morelia puede aumentar y agilizar las ventas de sus productos ya que son artículos usados y rápidamente se devengan, además de tener un contacto publicitario a la hora de mostrar los productos, esto sirve para el giro principal de la empresa lo cual su labor más importante es el empeño, el verdadero plus de la página es mediante el envío semanal de novedades y ofertas a los clientes que se inscriban en ella, lo cual mantendrá activos e interesados en los productos que están a la venta y en consecuencia la promoción y recordatorio sobre la necesidad de adquirir dinero prestado.

En la actualidad el internet se ha convertido en uno de los medios de comunicación más importante y necesaria para la sociedad, es por ello que esta página se diseñará con el propósito de transformarse en un espacio para brindar información acerca de los productos a la venta. Con una página web dinámica el administrador tendrá la facilidad de subir los artículos nuevos, dar de alta usuarios, etc, los usuarios novatos podrían tener un tremendo poder para hallar y tener acceso a la información acerca de sus productos y servicios. Esto es mucho más sencillo y económico que hacer llamadas telefónicas, imprimir nuevos catálogos o hacer publicaciones de prensa cada vez que lo requiera; Igualmente, puede mostrar nuevos productos e ideas y conseguir la opinión de los visitantes de su página, que son a la vez sus clientes potenciales. De esta manera, puede crear una base de datos de clientes (actuales y potenciales) para luego enviarle nuevas informaciones a través de correos electrónicos con costos irrisorios. Otorgar información detallada y específica acerca de sus productos. Si su público desea saber más sobre la empresa, la página Web es el mejor vehículo para hacerle llegar esa información.

Para las pequeñas empresas, ser capaz de vender productos de forma rápida y eficiente mediante el uso de la integración de comercio electrónico es una necesidad absoluta para mantenerse en la cima de las responsabilidades. Muchos empresarios utilizan el comercio electrónico como una forma de lograr que sus productos se puedan mantener en la mente de los clientes potenciales y además, para proporcionar información y apoyo para su empresa.

Pequeños sitios de comercio electrónico de negocio a menudo se llaman carros de compra o pasarelas de pago, ya que son, básicamente, una cesta virtual en la que los clientes pueden colocar los productos en venta.

Con las grandes empresas, la integración del comercio **electrónico** se está reflejando en consumidores cada vez en mayor número que están haciendo gran parte de sus compras en Internet. Eso significa que incluso los minoristas tradicionales han tenido que intensificar sus procesos de marketing online para proporcionar una manera conveniente para los clientes a hacer sus compras en línea a través de la tecnología web. Muchos consumidores utilizan un enfoque doble para tomar decisiones de compra hoy en día: en primer lugar mediante la búsqueda de los mejores precios en línea y luego mediante la colocación de órdenes en los sitios web de comercio electrónico.

Hay muchas industrias y organizaciones que mediante la integración de comercio electrónico como parte de sus procesos, están teniendo una mejor presencia en el mercado. Estas industrias incluyen sin fines de lucro, profesionales de la salud, proveedores de servicios educativos, las empresas de tecnología de la información, las empresas de investigación y desarrollo, empresas y asociaciones de redes, empresas de diseño web, **hoteles** y agencias de empleo, sólo para nombrar unas pocas.

Debido al aumento en la tecnología móvil, los consumidores y las empresas están confiando más en soluciones de comercio electrónico para proporcionar los medios de obtener información, productos y servicios a los demás de una manera más eficiente y a menor costo.

Objetivos Generales

- ✚ Se propone desarrollar una Pagina Web Dinámica para mostrar un catálogo de productos disponibles para la venta, y enviar información de estos vía email a suscriptores de manera automática.

Objetivos Específicos

- ✚ Crear una base de datos que contenga información de productos y de usuarios, relacionando una tabla de categorías para los productos.
- ✚ Crear un entorno amigable en la sección para el administrador, así como un buscador de usuarios y productos.
- ✚ Enviar semanalmente correos electrónicos a los usuarios mediante suscripción.
- ✚ Mostrar los productos por categoría.

Capítulo 1: Análisis y diseño de sistemas

1.1 Tipos de sistemas

Los sistemas de información se desarrollan con diversos propósitos, según las necesidades de la empresa. Los sistemas de procedimientos de transacciones (TPA, Transacción Processing Systems) funcionan al nivel operativo de una organización, los sistemas de automatización de la oficina (OAS, Office Automation Systems) y los sistemas de trabajo del conocimiento (KWS, Knowledge Work Systems) apoyan el trabajo al nivel del conocimiento. Los sistemas de información gerencial (DSS, Decision Support Systems) se encuentran entre los sistemas de alto nivel. Los sistemas expertos aplican el conocimiento de los encargados de la toma de decisiones para solucionar problemas estructurados específicos. Los sistemas de apoyo a ejecutivos (ESS, Executive Support Systems), se encuentran en el nivel estratégico de la administración. Los sistemas de apoyo a la toma de decisiones en grupo (GDSS, Group Decision Support Systems) y los sistemas de trabajo corporativo apoyado por computadora (CSCWS, Computer-Supported Collaborative Work Systems), descrito de manera más general, auxilian la toma de decisiones semiestructuradas o no estructuradas a nivel de grupo.

En este libro se emplean de manera indistinta los términos sistemas de información gerencial, sistemas de información (IS, information Systems), sistemas de información computarizados y sistemas de información de negocios computarizados, para denotar sistemas de información computarizados que apoyan el rango de actividades de negocios más amplio mediante la información que producen.

1.1.1 Sistema de procesamiento de transacciones

Los sistemas de procesamiento de transacciones (TPS, Transaction Processin Systems) son sistemas de información computarizada creados para procesar grandes cantidades de datos relacionadas con transacciones rutinarias de negocios, como las nóminas y los inventarios. Un TPS elimina el fastidio que representa la realización de transacciones operativas necesarias y reduce el tiempo que una vez fue requerido para llevarlas a cabo de manera manual, aunque los usuarios aún tienen que capturar datos en los sistemas computarizados.

Los sistemas de procesamiento de transacciones expenden los límites de la organización dado que le permiten interactuar con entornos externos. Es importante para las operaciones cotidianas de un negocio, que estos sistemas funcionen sin ningún tipo de interrupción, puesto que los administradores recurren a los datos producidos por los TPS con el propósito de obtener información actualizada sobre el funcionamiento de sus empresas.

1.1.2 Sistemas de automatización de la oficina y sistemas de trabajo del conocimiento

Existen dos clases de sistemas en el nivel del conocimiento de una organización, Los sistemas de automatización de la oficina (OAS, Office Automation Systems) apoyan a los trabajadores de datos, quienes por lo general no generan conocimientos nuevos, sino más bien analizan la información con el propósito de transformar los datos o manipularlos de alguna manera antes de compartirlos o, en sus caso, distribuirlos formalmente con el resto de la organización y en ocasiones más allá de ésta. Entre los componentes más comunes de un OAS están el procesamiento de texto, las hojas de cálculo, la autoedición, la calendarización electrónica y las comunicaciones mediante correo de voz, correo electrónico y videoconferencia.

Los sistemas de trabajo del conocimiento (KWS, Knowledge Works Systems) sirven de apoyo a los trabajadores profesionales, como los científicos, ingenieros y médicos en sus

esfuerzos de creación de nuevo conocimiento y dan a éstos la posibilidad de compartirlo con sus organizaciones o con la sociedad.

1.1.3 Sistemas de información gerencial

Los sistemas de información gerencial (MIS, Management Information Systems) no reemplazan a los sistemas de procesamiento de transacciones, más bien, incluyen el procesamiento de transacciones. Los MIS son sistemas de información computarizados cuyo propósito es contribuir a la correcta interacción entre los usuarios y las computadoras. Debido a que requieren que los usuarios, el software y el hardware, funcionen de manera coordinada, los sistemas de información gerencial dan apoyo a un espectro de tareas organizacionales, mucho más amplio que los sistemas de procesamiento de transacciones, como el análisis y la toma de decisiones.

Para acceder a la información, los usuarios de un sistema de información gerencial comparten una base de datos común. Esta almacena datos y modelos que ayudan al usuario a interpretar y a aplicar los datos. Los sistemas de información gerencial producen información que se emplea en la toma de decisiones. Un sistema de información gerencial también puede contribuir a unificar algunas de las funciones de información computarizadas de una empresa, a pesar de que no existe como una estructura individual en ninguna parte de ésta.

1.1.4 Sistemas de apoyo a la toma de decisiones

Los sistemas de apoyo a la toma de decisiones (DSS, Decision Support Systems) constituyen una clase de alto nivel de sistemas de información computarizada. Los DSS coinciden con los sistemas de información gerencial en que ambos dependen de una base de datos para abastecer de datos. Sin embargo, difieren en que el DSS pone énfasis en el apoyo de la toma de decisiones en todas sus fases, aunque la decisión definitiva es responsabilidad exclusiva del encargado de tomarla. Los sistemas de apoyo a la toma de

decisiones se ajustan más al gusto de la persona o grupo que los utiliza, que a los sistemas de información gerencial tradicionales. En ocasiones se hace referencia a ellos como sistemas que se enfocan en la inteligencia de negocios.

1.2 Integración de las tecnologías de sistemas

En esta sección se describen algunas de las nuevas tecnologías de información que los analistas de sistemas utilizarán para empresas que buscan integrar sus aplicaciones de comercio electrónico con sus negocios tradicionales, o bien, iniciar negocios electrónicos completamente nuevos.

1.2.1 Aplicaciones de comercio electrónico y sistemas web

Muchos de los sistemas que se describen en este libro pueden dotarse de una mayor funcionalidad si se migran a la World Wide Web o si desde su concepción se implementan como tecnologías basadas en la web. En una encuesta reciente la mitad de todas las empresas pequeñas y medianas respondieron que internet fue su estrategia preferida para buscar el crecimiento de sus negocios. Esta respuesta duplico a la de aquellos que manifestaron su inclinación para realizar alianzas estratégicas como medio para crecer. Hay muchos beneficios derivados de la implementación de una aplicación en la web:

- 1.- Una creciente difusión de la disponibilidad de un servicio, producto, industria, persona o grupo.
- 2.- La posibilidad de que los usuarios accedan las 24 horas.
- 3.- La estandarización del diseño de la interfaz.
- 4.- La creación de un sistema que se puede extender a nivel mundial y llegar a la gente en lugares remotos sin preocuparse de la zona horaria en que se encuentren.

1.3 El ciclo de vida del desarrollo de sistemas

A lo largo de éste capítulo, nos hemos referido al enfoque sistemático que el analista toma en relación con el análisis y diseño de sistemas de información. Gran parte de este enfoque se incluye en el ciclo de vida del desarrollo de sistemas (SDLC, Systems Development Life Cycle). El SDLC es un enfoque por fases para el análisis y el diseño cuya premisa principal consiste en que los sistemas se desarrollan mejor utilizando un ciclo específico de actividades del analista y el usuario.

Los analistas no se ponen de acuerdo en la cantidad de fases que incluye el ciclo de vida del desarrollo de sistemas, pero en general alaban su enfoque organizado. Aquí hemos dividido el ciclo en siete fases, a pesar de que cada fase se explica por separado, nunca se realiza como un paso aislado. Más bien, es posible que varias actividades ocurran de manera simultánea, y algunas de ellas podrían repetirse. Es más práctico considerar que el SDLC se realiza por fases (con actividades en pleno apogeo que se traslapan con otras hasta terminarse por completo) y no en pasos aislados.

1.3.1 Identificación de problemas, oportunidades y objetivos

En esta primera fase del ciclo de vida del desarrollo de sistemas, el analista se ocupa de identificar problemas, oportunidades y objetivos. Esta etapa es crítica para el éxito del resto del proyecto, pues a nadie le agrada desperdiciar tiempo trabajando en un problema que no era el que se debía resolver.

La primera fase requiere que el analista observe objetivamente lo que sucede en un negocio. A continuación, en conjunto con otros miembros de la organización, el analista determina con precisión cuáles son los problemas. Con frecuencia los problemas son detectados por alguien más, y esta es la razón de la llamada inicial al analista. Las oportunidades son situaciones que el analista considera susceptibles de mejorar utilizando sistemas de información computarizados.

Los usuarios, los analistas y los administradores de sistemas que coordinan el proyecto son los involucrados en la primera fase. Las actividades de esta fase consisten en entrevistar a los encargados de coordinar a los usuarios, sintetizar el conocimiento obtenido, estimar el alcance del proyecto y documentar los resultados. El resultado de esta fase es un informe de viabilidad que incluye una definición del problema y un resumen de los objetivos. A continuación, la administración debe decidir si se sigue adelante con el proyecto propuesto.

1.3.2 Determinación de los requerimientos de información

La siguiente fase que enfrenta el analista es la determinación de los requerimientos de información de los usuarios. Entre las herramientas que se utilizan para determinar los requerimientos de información de un negocio se encuentran métodos interactivos como las entrevistas, los muestreos, la investigación de datos impresos y la aplicación de cuestionarios; métodos que no interfieren con el usuario como la observación del comportamiento de los encargados de tomar las decisiones y sus entornos de oficina, al igual que métodos de amplio alcance como la elaboración de prototipos.

En la fase de determinación de los requerimientos de información del SDLC, el analista se esfuerza por comprender la información que necesitan los usuarios para llevar a cabo sus actividades. Como puede ver, varios métodos para determinar los requerimientos de información implican interactuar directamente con los usuarios. Esta fase es útil para que el analista confirme la idea que tiene de la organización y sus objetivos. En ocasiones solo realizan las dos primeras fases del ciclo de vida del desarrollo de sistemas. Esta clase de estudio podría tener un propósito distinto y por lo general la lleva a la práctica un especialista conocido como analista de información (IA, Information Analyst).

Los implicados en esta fase son el analista y los usuarios, por lo general trabajadores y gerentes del área de operaciones. El analista de sistemas necesita conocer los detalles de las funciones del sistema actual: el quien (la gente involucrada), el qué (la actividad del negocio), el dónde (el entorno donde se desarrollan las actividades), el cuándo (el momento oportuno) y el cómo (la manera en que se realizan los procedimientos actuales) del negocio que se estudia. A continuación el analista debe preguntar la razón por la cual se utiliza el

sistema actual. Podría haber buenas razones para realizar los negocios con los métodos actuales, y es importante tomarlas en cuenta al diseñar el nuevo sistema.

1.3.3 Análisis de las necesidades del sistema

La siguiente fase que debe enfrentar el analista tiene que ver con el análisis de las necesidades del sistema. De nueva cuenta, herramientas y técnicas especiales auxilian al analista en la determinación de los requerimientos. Una de estas herramientas es el uso de diagramas de flujo de datos para graficar las entradas, los procesos y las salidas de las funciones del negocio en una forma gráfica estructurada. A partir de los diagramas de flujo de datos se desarrolla un diccionario de datos que enlista todos los datos utilizados en el sistema, así como sus respectivas especificaciones.

Durante esta fase el analista de sistemas analiza también las decisiones estructuradas que se hayan tomado. Las decisiones estructuradas son aquellas en las cuales se pueden determinar las condiciones, las alternativas de condición, las acciones y las reglas de acción. Existen tres métodos principales para el análisis de decisiones estructuradas: español estructurado, tablas y árboles de decisión.

1.3.4 Diseño del sistema recomendado

En la fase de diseño del ciclo de vida del desarrollo de sistemas, el analista utiliza la información recopilada en las primeras fases para realizar el diseño lógico del sistema de información. El analista diseña procedimientos precisos para la captura de datos que aseguran que los datos que ingresen al sistema de información sean correctos. Además, el analista facilita la entrada eficiente de datos al sistema de información mediante técnicas adecuadas de diseño de formularios y pantallas.

La concepción de la interfaz de usuario forma parte del diseño lógico del sistema de información. La interfaz conecta al usuario con el sistema y por tanto es sumamente importante. Entre los ejemplos de interfaces de usuario se encuentran el teclado (para teclear preguntas y respuestas), los menús en pantalla (para obtener los comandos de usuario) y

diversas interfaces gráficas de usuario (GUIs, *Graphical User Interfaces*) que se manejan a través de un ratón o una pantalla sensible al tacto.

La fase de diseño también incluye el diseño de archivos o bases de datos que almacenarán gran parte de los datos indispensables para los encargados de tomar las decisiones en la organización. Una base de datos bien organizada es el cimiento de cualquier sistema de información. En esta fase el analista también interactúa con los usuarios para diseñar la salida (en pantalla o impresa) que satisfaga las necesidades de información de estos últimos. Finalmente, el analista debe diseñar controles y procedimientos de respaldo que protejan al sistema y a los datos, y producir paquetes de especificaciones de programa para los programadores. Cada paquete debe contener esquemas para la entrada y la salida, especificaciones de archivos y detalles del procesamiento; también podría incluir árboles o tablas de decisión, diagramas de flujo de datos, un diagrama de flujo de sistema, y los nombres y funciones de cualquier rutina de código previamente escrita.

1.3.5 Desarrollo y documentación del software

En la quinta fase del ciclo de vida del desarrollo de sistemas, el analista trabaja de manera conjunta con los programadores para desarrollar cualquier software original necesario. Entre las técnicas estructuradas para diseñar y documentar software se encuentran los diagramas de estructura, los diagramas de Nassi-Shneiderman y el pseudocódigo. El analista se vale de una o más de estas herramientas para comunicar al programador lo que se requiere programar.

Durante esta fase el analista también trabaja con los usuarios para desarrollar documentación efectiva para el software, como manuales de procedimientos, ayuda en línea y sitios Web que incluyan respuestas a preguntas frecuentes (FAQ, *Frequently Asked Questions*) en archivos "Léame" que se integrarán en el nuevo software. La documentación indica a los usuarios cómo utilizar el software y lo que deben hacer en caso de que surjan problemas derivados de este uso.

1.3.6 Prueba y mantenimiento del sistema

Antes de poner el sistema en funcionamiento es necesario probarlo. Es mucho menos costoso encontrar los problemas antes que el sistema se entregue a los usuarios. Una parte de las pruebas las realizan los programadores solos, y otra la llevan a cabo de manera conjunta con los analistas de sistemas. Primero se realiza una serie de pruebas con datos de muestra para determinar con precisión cuáles son los problemas y posteriormente se realiza otra con datos reales del sistema actual.

El mantenimiento del sistema de información y su documentación empiezan en esta fase y se llevan a cabo de manera rutinaria durante toda su vida útil. Gran parte del trabajo habitual del programador consiste en el mantenimiento, y las empresas invierten enormes sumas de dinero en esta actividad. Parte del mantenimiento, como las actualizaciones de programas, se pueden realizar de manera automática a través de un sitio Web. Muchos de los procedimientos sistemáticos que el analista emplea durante el ciclo de vida del desarrollo de sistemas pueden contribuir a garantizar que el mantenimiento se mantendrá al mínimo.

1.3.7 Implementación y evaluación del sistema

Ésta es la última fase del desarrollo de sistemas, y aquí el analista participa en la implementación del sistema de información. En esta fase se capacita a los usuarios en el manejo del sistema. Parte de la capacitación la imparten los fabricantes, pero la supervisión de ésta es responsabilidad del analista de sistemas. Además, el analista tiene que planear una conversión gradual del sistema anterior al actual. Este proceso incluye la conversión de archivos de formatos anteriores a los nuevos, o la construcción de una base de datos, la instalación de equipo y la puesta en producción del nuevo sistema.

Se menciona la evaluación como la fase final del ciclo de vida del desarrollo de sistemas principalmente en aras del debate. En realidad, la evaluación se lleva a cabo durante cada una de las fases. Un criterio clave que se debe cumplir es si los usuarios a quienes va dirigido el sistema lo están utilizando realmente. Debe hacerse hincapié en que, con frecuencia, el trabajo de sistemas es cíclico. Cuando un analista termina una fase del

desarrollo de sistemas y pasa a la siguiente, el surgimiento de un problema podría obligar al analista a regresar a la fase previa y modificar el trabajo realizado.

Capítulo 2: Internet

Internet es una red de redes de millones de ordenadores en todo el mundo. Pero al contrario de lo que se piensa comúnmente, Internet no es sinónimo de World Wide Web. La Web es sólo una parte de Internet, es sólo uno de los muchos servicios que ofrece Internet.

Internet, la red de redes, suministra un foro de comunicación en el que participan millones de personas de todos los países del mundo, en mayor o menor medida. Internet aporta o soporta una serie de instrumentos para que la gente difunda y acceda a documentos y a la información (WWW, FTP, etc.), para que los individuos y los grupos se relacionen a través de una serie de medios de comunicación más o menos nuevos (correo electrónico, news, listas de distribución, videoconferencia, chats...) o más o menos viejos (como una conversación telefónica, poner un fax, etc.) y también incluye dentro de sí a los denominados medios de comunicación de masas (radio, televisión, periódicos y revistas "on line", cine, la omnipresente publicidad, etc). ¿Se trata de un nuevo medio de comunicación? ¿Pueden utilizarse las nociones habituales que se aplican a los medios de comunicación para definir y caracterizar la comunicación en Internet?

En realidad, Internet no es un medio de comunicación, sino muchos medios, una red que comprende distintos tipos y distintos sistemas de comunicación. La gente utiliza Internet para muy distintas finalidades. Muchas de ellas están relacionadas con diferentes y variadas categorías de comunicación, información e interacción. Algunas son nuevas y otras muy antiguas, pero estas categorías no se excluyen mutuamente, ya que no sólo los usuarios pueden participar en Internet mediante una combinación de comunicación, información e interacción al mismo tiempo, sino que también los distintos medios se entremezclan en el mismo canal.

En puridad, se podría afirmar que Internet no es ni un canal ni un medio, sino un hipersistema, un macrosistema o un metasistema de comunicación en el que los distintos sistemas se mezclan y entretajan. La digitalidad no sólo reduce todas las morfologías de la información a una sola: texto, imagen y sonido se convierten en bits de información, sino que también los distintos medios de comunicación con sus distintos sistemas, se integran en uno al converger en la red.

Desde el punto de vista documental, Internet es un archivo y una biblioteca universal en la que el usuario, desde su pantalla de ordenador, tiene acceso inmediato a la mayor fuente de información que jamás ha existido.

Desde el punto de la vista de la comunicación, Internet se caracteriza por ser una red horizontal y en gran medida descentralizada. La comunicación se puede realizar de forma multidireccional (participar en un grupo de debate), unidireccional (leer, bajar archivos de la red, buscar información, etc.), bidireccional (conversar o jugar con alguien, intercambiar archivos o correos electrónicos) e interactiva. Y se establece tanto en ámbitos públicos como en privados. Al contrario que los medios de masas (radio, televisión, prensa, etc.) que no permiten interacciones transversales entre los participantes, los medios en Internet sí permiten este tipo de interacciones. El teléfono usa una comunicación uno a uno, en el ciberespacio un individuo es, potencialmente, emisor y receptor a la vez y todos pueden comunicar con todos en un espacio sincrónico o asíncrono, simultáneo o no concurrente.

Desde el punto de vista técnico, Internet es un gran conjunto de redes de ordenadores interconectadas cuyo funcionamiento interno no se ajusta a ningún tipo de ordenador concreto, a ningún tipo de red específica, a ninguna tecnología de conexión en exclusiva y a ningún medio físico privilegiado, se trata de una red flexible y dinámica, adaptable a diferentes contextos tecnológicos. Internet es, en sí misma, un universo tecnológico, una hipertecnología o una miríada de tecnologías que confluyen: telefonía, electrónica, microprocesadores, cables, satélites, fibra óptica, hipertexto, multimedia, televisión, escritura, dibujo, fotografía, vídeo, imágenes en 3D, realidad virtual, etc.

Desde el punto de vista sociológico, Internet es un fenómeno económico, social y cultural, que tiene repercusiones sobre las personas y las sociedades y sobre la forma en que éstas se comunican, interrelacionan, producen, comercian, trabajan, se divierten y se organizan. Se puede afirmar, incluso, que repercute sobre la propia identidad de las personas y de los grupos sociales, puesto que modifica la forma en que se ven a sí mismos, operan y se representan los individuos y las sociedades.

2.1 World Wide Web

Es básicamente un medio de comunicación de texto, gráficos y otros objetos multimedia a través de Internet, es decir, la web es un sistema de hipertexto que utiliza Internet como su mecanismo de transporte o desde otro punto de vista, una forma gráfica de explorar Internet.

La web fue creada en 1989 en un instituto de investigación de Suiza, la web se basa en buscadores y el protocolo de transporte de hipertexto (hypertext transport protocol (http)). La mayoría de los documentos de la web se crean utilizando lenguaje HTML (hypertext markup language).

Es importante saber que web o www no son sinónimo de Internet, la web es un subconjunto de Internet que consiste en páginas a las que se puede acceder usando un navegador. Internet es la red de redes donde reside toda la información. Tanto el correo electrónico, como FTPs, juegos, etc. son parte de Internet, pero no de la Web.

Para buscar hipertexto se utilizan programas llamados buscadores web que recuperan trozos de información (llamados *documentos* o *páginas web*) de los servidores web y muestran en la pantalla del ordenador de la persona que está buscando la información gráfica, textual o video e incluso audio.

Después se pueden seguir enlaces o hyperlinks en cada página a otros documentos o incluso devolver información al servidor para interactuar con él. Al acto de seguir un enlace tras otro a veces se le llama navegar en Internet.

La web se ha convertido en un medio muy popular de publicar información en Internet, y con el desarrollo del protocolo de transferencia segura (secured server protocol (https)), la web es ahora un medio de comercio electrónico donde los consumidores pueden escoger sus productos on-line y realizar sus compras utilizando la información de sus tarjetas bancarias de forma segura.

2.2 Protocolos

Un **protocolo** es un conjunto de reglas usadas por computadoras para comunicarse unas con otras a través de una red. Un protocolo es una regla o estándar que controla o permite la comunicación en su forma más simple, un protocolo puede ser definido como las reglas que dominan la sintaxis, semántica y sincronización de la comunicación. Los protocolos pueden ser implementados por hardware, software, o una combinación de ambos. A su más bajo nivel, un protocolo define el comportamiento de una conexión de hardware.

2.2.1 TCP/IP

Es uno de los protocolos fundamentales en Internet. Fue creado entre los años 1973 y 1974 por Vint Cerf y Robert Kahn.

Muchos programas dentro de una red de datos compuesta por computadoras pueden usar TCP para crear *conexiones* entre ellos a través de las cuales puede enviarse un flujo de datos. El protocolo garantiza que los datos serán entregados en su destino sin errores y en el mismo orden en que se transmitieron. También proporciona un mecanismo para distinguir distintas aplicaciones dentro de una misma máquina, a través del concepto de puerto.

TCP da soporte a muchas de las aplicaciones más populares de Internet (navegadores, intercambio de ficheros, clientes ftp,...) y protocolos de aplicación HTTP, SMTP, SSH y FTP.

En la pila de protocolos TCP/IP, TCP es la capa intermedia entre el protocolo de internet (IP) y la aplicación. Habitualmente, las aplicaciones necesitan que la comunicación sea fiable y, dado que la capa IP aporta un servicio de datagramas no fiable (sin confirmación), TCP añade las funciones necesarias para prestar un servicio que permita que la comunicación entre dos sistemas se efectúe libre de errores, sin pérdidas y con seguridad.

Los servicios provistos por TCP corren en el anfitrión (host) de cualquiera de los extremos de una conexión, no en la red. Por lo tanto, TCP es un protocolo para manejar conexiones de

extremo a extremo. Tales conexiones pueden existir a través de una serie de conexiones punto a punto, por lo que estas conexiones extremo-extremo son llamadas circuitos virtuales.

IP es un protocolo no orientado a conexión usado tanto por el origen como por el destino para la comunicación de datos a través de una red de paquetes conmutados no fiable de mejor entrega posible sin garantías.

Los datos en una red basada en IP son enviados en bloques conocidos como paquetes o datagramas (en el protocolo IP estos términos se suelen usar indistintamente). En particular, en IP no se necesita ninguna configuración antes de que un equipo intente enviar paquetes a otro con el que no se había comunicado antes.

IP provee un servicio de datagramas no fiable (también llamado del *mejor esfuerzo* (*best effort*), lo hará lo mejor posible pero garantizando poco). IP no provee ningún mecanismo para determinar si un paquete alcanza o no su destino y únicamente proporciona seguridad (mediante *checksums* o sumas de comprobación) de sus cabeceras y no de los datos transmitidos. Por ejemplo, al no garantizar nada sobre la recepción del paquete, éste podría llegar dañado, en otro orden con respecto a otros paquetes, duplicado o simplemente no llegar. Si se necesita fiabilidad, ésta es proporcionada por los protocolos de la capa de transporte, como TCP.

Si la información a transmitir ("datagramas") supera el tamaño máximo "negociado" (MTU) en el tramo de red por el que va a circular podrá ser dividida en paquetes más pequeños, y reensamblada luego cuando sea necesario. Estos fragmentos podrán ir cada uno por un camino diferente dependiendo de cómo estén de congestionadas las rutas en cada momento.

2.2.2 http

Todas las páginas web están escritas en lenguaje de hipertexto (hyper-text markup language (**HTML**)), por lo que el hipertexto es el contenido de las páginas web.

El protocolo de transferencia es el sistema mediante el cual se transfiere información entre los servidores y los clientes (por ejemplo los navegadores).

Hay una versión de http para la transferencia segura de información llamada **https** que puede utilizar cualquier método de cifrado siempre que sea entendido tanto por el servidor como por el cliente.

2.2.3 ARP

ARP son las siglas en inglés de **Address Resolution Protocol** (Protocolo de resolución de direcciones).

Es un protocolo de nivel de enlace responsable de encontrar la dirección hardware (Ethernet MAC) que corresponde a una determinada dirección IP. Para ello se envía un paquete (ARP request) a la dirección de difusión de la red (broadcast (MAC = FF FF FF FF FF FF)) que contiene la dirección IP por la que se pregunta, y se espera a que esa máquina (u otra) responda (ARP reply) con la dirección Ethernet que le corresponde. Cada máquina mantiene una caché con las direcciones traducidas para reducir el retardo y la carga. ARP permite a la dirección de Internet ser independiente de la dirección Ethernet, pero esto sólo funciona si todas las máquinas lo soportan.

ARP está documentado en el RFC (Request For Comments) 826.

El protocolo RARP realiza la operación inversa y se encuentra descrito en el RFC 903.

En Ethernet, la capa de enlace trabaja con direcciones físicas. El protocolo ARP se encarga de traducir las direcciones IP a direcciones MAC (direcciones físicas). Para realizar ésta conversión, el nivel de enlace utiliza las tablas ARP, cada interfaz tiene tanto una dirección IP como una dirección física MAC.

ARP se utiliza en 4 casos referentes a la comunicación entre 2 hosts:

1. Cuando 2 hosts están en la misma red y uno quiere enviar un paquete a otro.
2. Cuando 2 host están sobre redes diferentes y deben usar un gateway/router para alcanzar otro host.
3. Cuando un router necesita enviar un paquete a un host a través de otro router.
4. Cuando un router necesita enviar un paquete a un host de la misma red.

2.2.4 FTP

Es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP (Transmission Control Protocol), basado en la arquitectura cliente-servidor. Desde un equipo cliente se puede conectar a un servidor para descargar archivos desde él o para enviarle archivos, independientemente del sistema operativo utilizado en cada equipo.

El servicio FTP es ofrecido por la capa de aplicación del modelo de capas de red TCP/IP al usuario, utilizando normalmente el puerto de red 20 y el 21. Un problema básico de FTP es que está pensado para ofrecer la máxima velocidad en la conexión, pero no la máxima seguridad, ya que todo el intercambio de información, desde el login y password del usuario en el servidor hasta la transferencia de cualquier archivo, se realiza en texto plano sin ningún tipo de cifrado, con lo que un posible atacante puede capturar este tráfico, acceder al servidor, o apropiarse de los archivos transferidos.

Para solucionar este problema son de gran utilidad aplicaciones como scp y sftp, incluidas en el paquete SSH, que permiten transferir archivos pero cifrando todo el tráfico.

En el modelo, el intérprete de protocolo (IP) de usuario inicia la conexión de control en el puerto 21. Las órdenes FTP estándar las genera el PI de usuario y se transmiten al proceso servidor a través de la conexión de control. Las respuestas estándar se envían desde el PI del servidor al PI de usuario por la conexión de control como respuesta a las órdenes.

Estas órdenes FTP especifican parámetros para la conexión de datos (puerto de datos, modo de transferencia, tipo de representación y estructura) y la naturaleza de la operación sobre el sistema de archivos (almacenar, recuperar, añadir, borrar, etc.). El proceso de transferencia de datos (DTP) de usuario u otro proceso en su lugar, debe esperar a que el servidor inicie la conexión al puerto de datos especificado (puerto 20 en modo activo o estándar) y transferir los datos en función de los parámetros que se hayan especificado.

Vemos también en el diagrama que la comunicación entre cliente y servidor es independiente del sistema de archivos utilizado en cada computadora, de manera que no importa que sus sistemas operativos sean distintos, porque las entidades que se comunican entre sí son los PI y los DTP, que usan el mismo protocolo estandarizado: el FTP.

También hay que destacar que la conexión de datos es bidireccional, es decir, se puede usar simultáneamente para enviar y para recibir, y no tiene por qué existir todo el tiempo que dura la conexión FTP. Pero tenía en sus comienzos un problema, y era la localización de los servidores en la red. Es decir, el usuario que quería descargar algún archivo mediante FTP debía conocer en qué máquina estaba ubicado. La única herramienta de búsqueda de información que existía era Gopher, con todas sus limitaciones.

2.2.5 SMTP

(Protocolo simple de transferencia de correo) es el protocolo estándar que permite la transferencia de correo de un servidor a otro mediante una conexión punto a punto.

Éste es un protocolo que funciona en línea, encapsulado en una trama TCP/IP. El correo se envía directamente al servidor de correo del destinatario. El protocolo SMTP funciona con comandos de textos enviados al servidor SMTP (al puerto 25 de manera predeterminada). A cada comando enviado por el cliente (validado por la cadena de caracteres ASCII *CR/LF*, que equivale a presionar la tecla Enter) le sigue una respuesta del servidor SMTP compuesta por un número y un mensaje descriptivo.

A continuación se describe una situación en la que se realiza una solicitud para enviar correos a un servidor SMTP:

- Al abrir la sesión SMTP, el primer comando que se envía es el comando *HELO* seguido por un espacio (escrito *<SP>*) y el nombre de dominio de su equipo (para decir "hola, soy este equipo"), y después validado por Enter (escrito *<CRLF>*). Desde abril de 2001, las especificaciones para el protocolo SMTP, definidas en RFC 2821, indican que el comando HELO sea remplazado por el comando *EHLO*.
- El segundo comando es "*MAIL FROM:*" seguido de la dirección de correo electrónico del remitente. Si se acepta el comando, el servidor responde con un mensaje "250 OK".
- El siguiente comando es "*RCPT TO:*" seguido de la dirección de correo electrónico del destinatario. Si se acepta el comando, el servidor responde con un mensaje "250 OK".
- El comando *DATA* es la tercera etapa para enviar un correo electrónico. Anuncia el comienzo del cuerpo del mensaje. Si se acepta el comando, el servidor responde con un mensaje intermediario numerado 354 que indica que puede iniciarse el envío del cuerpo del mensaje y considera el conjunto de líneas siguientes hasta el final del mensaje indicado con una línea que contiene sólo un punto. El cuerpo del correo electrónico eventualmente contenga algunos de los siguientes encabezados:
 - Date (Fecha)
 - Subject (Asunto)
 - Cc
 - Bcc (Cco)
 - From (De)

2.2.6 POP

Post Office Protocol (POP3, *Protocolo de la oficina de correo*) en clientes locales de correo para obtener los mensajes de correo electrónico almacenados en un servidor remoto. Es un protocolo de nivel de aplicación en el Modelo OSI.

POP3 está diseñado para recibir correo, no para enviarlo; le permite a los usuarios con conexiones intermitentes o muy lentas (tales como las conexiones por módem), descargar su correo electrónico mientras tienen conexión y revisarlo posteriormente incluso estando desconectados. Cabe mencionar que la mayoría de los clientes de correo incluyen la opción de *dejar los mensajes en el servidor*, de manera tal que, un cliente que utilice POP3 se conecta, obtiene todos los mensajes, los almacena en la computadora del usuario como mensajes nuevos, los elimina del servidor y finalmente se desconecta. En contraste, el protocolo IMAP permite los modos de operación *conectado y desconectado*.

Los clientes de correo electrónico que utilizan IMAP dejan por lo general los mensajes en el servidor hasta que el usuario los elimina directamente. Esto y otros factores hacen que la operación de IMAP permita a múltiples clientes acceder al mismo buzón de correo. La mayoría de los clientes de correo electrónico soportan POP3 o IMAP; sin embargo, solo unos cuantos proveedores de internet ofrecen IMAP como valor agregado de sus servicios.

Los clientes que utilizan la opción *dejar mensajes en el servidor* **por lo general utilizan la orden UIDL** (**U**nique **I**Dentification **L**isting). La mayoría de las órdenes de POP3 identifican los mensajes dependiendo de su número ordinal del servidor de correo. Esto genera problemas al momento que un cliente pretende dejar los mensajes en el servidor, ya que los mensajes con número cambian de una conexión al servidor a otra. Por ejemplo un buzón de correo contenía 5 mensajes en la última conexión, después otro cliente elimina el mensaje número 3, si se vuelve a iniciar otra conexión, ya el número que tiene el mensaje 4 pasará a ser 3, y el mensaje 5 pasará a ser número 4 y la dirección de estos dos mensajes cambiara. El UIDL proporciona un mecanismo que evita los problemas de numeración. El servidor le asigna una cadena de caracteres única y permanente al mensaje. Cuando un cliente de correo compatible con POP3 se conecta al servidor utiliza la orden UIDL para obtener el mapeo del identificador de mensaje. De esta manera el cliente puede utilizar ese mapeo para determinar qué mensajes hay que descargar y cuáles hay que guardar al momento de la descarga.

Al igual que otros viejos protocolos de internet, POP3 utilizaba un mecanismo de firmado sin cifrado. La transmisión de contraseñas de POP3 en texto plano aún se da. En la actualidad POP3 cuenta con diversos métodos de autenticación que ofrecen una diversa gama de

niveles de protección contra los accesos ilegales al buzón de correo de los usuarios. Uno de estos es **APOP**, el cual utiliza funciones MD5 para evitar los ataques de contraseñas. Mozilla, Eudora, Novell Evolution así como Mozilla Thunderbird implementan funciones APOP.

2.3 Pagina Web

En inglés **website** o **web site**, un **sitio web** es un sitio (localización) en la World Wide Web que contiene documentos (**páginas web**) organizados jerárquicamente. Cada documento (página web) contiene texto y o gráficos que aparecen como información digital en la pantalla de un ordenador. Un sitio puede contener una combinación de gráficos, texto, audio, vídeo, y otros materiales dinámicos o estáticos.

Cada sitio web tiene una **página de inicio (en inglés Home Page)**, que es el primer documento que ve el usuario cuando entra en el sitio web poniendo el nombre del dominio de ese sitio web en un navegador. El sitio normalmente tiene otros documentos (páginas web) adicionales. Cada sitio pertenece y es gestionado y por un individuo, una compañía o una organización.

Como medio, los sitios web son similares a las películas, a la televisión o a las revistas, en que también crean y manipulan imágenes digitales y texto, pero un sitio web es también un medio de comunicación. La diferencia principal entre un sitio web y los medios tradicionales es que un sitio web está en una red de ordenadores (Internet) y está codificado de manera que permite que los usuarios interactúen con él. Una vez en un sitio web, puedes realizar compras, búsquedas, enviar mensajes, y otras actividades interactivas.

Diferencia entre sitio web y página web

A veces se utiliza erróneamente el término **página web** para referirse a **sitio web**. **Una página web es** parte de un sitio web y es un único archivo con un nombre de archivo asignado, mientras que un sitio web es un conjunto de archivos llamados páginas web.

Si lo comparáramos con un libro, un sitio web sería el libro entero y una página web de ese sitio web sería un capítulo de ese libro. El título del libro sería el nombre del dominio del sitio

web. Un capítulo, al igual que una página web, tiene un nombre que lo define. Decimos que sería un capítulo y no una página del libro porque a menudo es necesario desplazarse hacia abajo en la pantalla para ver todo el contenido de una página web, al igual que en un libro te *desplazas* a través de varias páginas para ver todo el contenido de un capítulo. El índice de los capítulos del libro sería el equivalente al **mapa del sitio web (sitemap en inglés)**.

2.3.1 Lenguaje de marcado

Los lenguajes de marcas, también denominados lenguajes de marcado o lenguajes de descripción de documentos, construyen un conjunto de reglas que definen todo aquello que es parte de un documento digital, pero que no pertenece al texto del mismo. Los lenguajes de marcas no son lenguajes de formato similares a los lenguajes que se usan en Internet como los de descripción de páginas (archivos PostScript, archivos .pdf, etc.) ni son lenguajes de programación (Java, Perl, C++...), sino que se trata de lenguajes orientados a definir la estructura y la semántica de un documento. En realidad, más que de lenguajes, podríamos hablar de metalenguajes o sistemas formales mediante los cuales se añade información o codificación a la forma digital de un documento bien para controlar su procesamiento, bien para representar su significado.

En un documento existen distintos niveles de información: por un lado, los datos que conforman el contenido de un documento (caracteres de contenido), y por otro, una información superpuesta al contenido, que es lo que constituye el etiquetado, marcado o "markup" (caracteres de etiquetado).

Un lenguaje de marcado cumple con dos objetivos esenciales a la hora de diseñar y procesar un documento digital:

Especifica las operaciones tipográficas y las funciones que debe ejecutar el programa navegador/visualizador sobre dichos elementos. Las operaciones tipográficas son instrucciones de formato que se aplican a cada uno de los elementos de un documento digital como, por ejemplo, imprimir un título en negrita y a un determinado tamaño.

Separa un texto en los elementos de los que se compone, como por ejemplo un párrafo, un capítulo, un encabezamiento, etc.

Así, pues, hay 2 tipos de marcación:

específica: describe cómo ha de formatearse el documento: fuente, tamaño, color, etc.

estructural: describe la estructura del documento: titular, párrafo, etc.

En general los lenguajes de marcado siguen una sintaxis basada en el uso de marcas o etiquetas: una etiqueta que indica el principio de un elemento y otra el final del mismo.

2.3.1.1 HTML

(Lenguaje de marcado de hipertexto), es el lenguaje de marcado predominante para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML se escribe en forma de «etiquetas», rodeadas por corchetes angulares (<,>). HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un *script* (por ejemplo JavaScript), el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML.

HTML también es usado para referirse al contenido del tipo de MIME text/html o todavía más ampliamente como un término genérico para el HTML, ya sea en forma descendida del XML (como XHTML 1.0 y posteriores) o en forma descendida directamente de SGML (como HTML 4.01 y anteriores).

Elementos

Los elementos son la estructura básica de HTML. Los elementos tienen dos propiedades básicas: atributos y contenido. Cada atributo y contenido tiene ciertas restricciones para que se considere válido al documento HTML. Un elemento generalmente tiene una etiqueta de inicio (p.ej. <nombre-de-elemento>) y una etiqueta de cierre (p.ej. </nombre-de-elemento>). Los atributos del elemento están contenidos en la etiqueta de inicio y el contenido está

ubicado entre las dos etiquetas (p.ej. `<nombre-de-elemento atributo="valor">Contenido</nombre-de-elemento>`). Algunos elementos, tales como `<br>`, no tienen contenido ni llevan una etiqueta de cierre. Debajo se listan varios tipos de elementos de marcado usados en HTML.

Estructura general de una línea de código en el lenguaje de etiquetas HTML.

El marcado *estructural* describe el propósito del texto. Por ejemplo, `<h2>Golf</h2>` establece «Golf» como un encabezamiento de segundo nivel, el cual se mostraría en un navegador de una manera similar al título «Marcado HTML» al principio de esta sección. El marcado estructural no define cómo se verá el elemento, pero la mayoría de los navegadores web han estandarizado el formato de los elementos. Un formato específico puede ser aplicado al texto por medio de hojas de estilo en cascada.

2.3.2 Lenguaje de programación

Los lenguajes de programación son herramientas que nos permiten crear programas y software. Entre ellos tenemos Delphi, Visual Basic, Pascal, Java, etc. Una computadora funciona bajo control de un programa el cual debe estar almacenado en la unidad de memoria; tales como el disco duro.

Los *lenguajes de programación* de una computadora en particular se conocen como código de máquinas o lenguaje de máquinas. Estos lenguajes codificados en una computadora específica no podrán ser ejecutados en otra computadora diferente. Para que estos programas funcionen para diferentes computadoras hay que realizar una versión para cada una de ellas, lo que implica el aumento del costo de desarrollo.

Por otra parte, los lenguajes de programación en código de máquina son verdaderamente difíciles de entender para una persona, ya que están compuestos de códigos numéricos sin sentido nemotécnico. Los *lenguajes de programación* facilitan la tarea de programación, ya que disponen de formas adecuadas que permiten ser leídas y escritas por personas, a su vez resultan independientes del modelo de computador a utilizar.

Los lenguajes de programación representan en forma simbólica y en manera de un texto los códigos que podrán ser leídos por una persona. Los lenguajes de programación son independientes de las computadoras a utilizar.

Existen estrategias que permiten ejecutar en una computadora un programa realizado en un lenguaje de programación simbólico. Los procesadores del lenguaje son los programas que permiten el tratamiento de la información en forma de texto, representada en los lenguajes de programación simbólicos.

Hay lenguajes de programación que utilizan *compilador*.

La ejecución de un programa con compilador requiere de dos etapas:

- 1) Traducir el programa simbólico a código máquina.
- 2) Ejecución y procesamiento de los datos.

Otros lenguajes de programación utilizan un programa intérprete o traductor, el cual analiza directamente la descripción simbólica del programa fuente y realiza las instrucciones dadas. El *intérprete* en los lenguajes de programación simula una máquina virtual, donde el lenguaje de máquina es similar al lenguaje fuente.

La ventaja del proceso interprete es que no necesita de dos fases para ejecutar el programa, sin embargo su inconveniente es que la velocidad de ejecución es más lenta ya que debe analizar e interpretar las instrucciones contenidas en el programa fuente.

2.3.2.1 PHP

PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Se usa principalmente para la interpretación del lado del servidor (*server-side scripting*) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+.

PHP es un acrónimo recursivo que significa *PHP Hypertext Pre-processor* (inicialmente *PHP Tools*, o, *Personal Home Page Tools*). Fue creado originalmente por Rasmus Lerdorf en 1994; sin embargo la implementación principal de PHP es producida ahora por The PHP Group y sirve como el estándar de facto para PHP al no haber una especificación formal. Publicado bajo la PHP License, la Free Software Foundation considera esta licencia como software libre.

Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. El lenguaje PHP se encuentra instalado en más de 20 millones de sitios web y en un millón de servidores, el número de sitios en PHP ha compartido algo de su preponderante sitio con otros nuevos lenguajes no tan poderosos desde agosto de 2005. Este mismo sitio web de Wikipedia está desarrollado en PHP. Es también el módulo Apache más popular entre las computadoras que utilizan Apache como servidor web.

El gran parecido que posee PHP con los lenguajes más comunes de programación estructurada, como C y Perl, permiten a la mayoría de los programadores crear aplicaciones complejas con una curva de aprendizaje muy corta. También les permite involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones.

Aunque todo en su diseño está orientado a facilitar la creación de sitios webs, es posible crear aplicaciones con una interfaz gráfica para el usuario, utilizando la extensión PHP-Qt o PHP-GTK. También puede ser usado desde la línea de órdenes, de la misma manera como Perl o Python pueden hacerlo; a esta versión de PHP se la llama PHP-CLI (*Command Line Interface*).

Cuando el cliente hace una petición al servidor para que le envíe una página web, el servidor ejecuta el intérprete de PHP. Éste procesa el script solicitado que generará el contenido de manera dinámica (por ejemplo obteniendo información de una base de datos). El resultado es enviado por el intérprete al servidor, quien a su vez se lo envía al cliente. Mediante extensiones es también posible la generación de archivos PDF, Flash, así como imágenes en diferentes formatos.

Permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, PostgreSQL, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite.

XAMPP es un servidor independiente de plataforma, software libre, que consiste principalmente en la base de datos MySQL, el servidor Web Apache y los intérpretes para lenguajes de script: PHP y Perl. El nombre proviene del acrónimo de X (para cualquiera de

los diferentes sistemas operativos), Apache, MySQL, PHP, Perl. El programa está liberado bajo la licencia GNU y actúa como un servidor Web libre, fácil de usar y capaz de interpretar páginas dinámicas. Actualmente XAMPP está disponible para Microsoft Windows, GNU/Linux, Solaris, y MacOS X.

PHP también tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos, tales como Unix (y de ese tipo, como Linux o Mac OS X) y Microsoft Windows, y puede interactuar con los servidores de web más populares ya que existe en versión CGI, módulo para Apache, e ISAPI.

PHP es una alternativa a las tecnologías de Microsoft ASP y ASP.NET (que utiliza C# y Visual Basic .NET como lenguajes), a ColdFusion de la empresa Adobe, a JSP/Java y a CGI/Perl. Aunque su creación y desarrollo se da en el ámbito de los sistemas libres, bajo la licencia GNU, existe además un entorno de desarrollo integrado comercial llamado Zend Studio. CodeGear (la división de lenguajes de programación de Borland) ha sacado al mercado un entorno de desarrollo integrado para PHP, denominado 'Delphi for PHP. También existen al menos un par de módulos para Eclipse, uno de los entornos más populares.

2.3.2.1.1 Características fundamentales del lenguaje

PHP es un lenguaje para la creación de sitios web del que se pueden destacar las siguientes 3 características:

- Es un potente y robusto lenguaje de programación embebido en documentos HTML.
- Dispone de librerías de conexión con la gran mayoría de los sistemas de gestión de bases de datos para el almacenamiento de información permanente en el servidor.
- Proporciona soporte a múltiples protocolos de comunicaciones en internet (HTTP, IMAP, FTP, LDAP, SNMP, etc.).

Aparte de estas características básicas, existen otras no menos importantes:

- Código fuente abierto: el código del interprete esta accesible para permitir posibles mejoras o sugerencias acerca de su desarrollo (PHP ha sido escrito en lenguaje C).
- Gratuito: no es necesario realizar ningún desembolso económico para desarrollar sistemas de información empleando este versátil lenguaje.
- Portable y multiplataforma: existen versiones del interprete para múltiples plataformas (Windows 95, 98, NT, 2000, Unix, Linux, etc.). Esto permite que las aplicaciones puedan ser soportadas de una plataforma a otra sin necesidad de modificar ni una sola línea de código.
- Eficiente: PHP consume muy pocos recursos en el servidor, por lo que con un equipo relativamente sencillo es posible desarrollar interesantes aplicaciones.
- Alta velocidad de desarrollo: PHP permite desarrollar rápidamente sitios web dinámicos. Proporciona gran cantidad de librerías muy útiles y bien documentadas que ahorran mucho trabajo al programador.

Por último, y aparte de todas estas características citadas anteriormente, también dispone de facilidades para el procesamiento de ficheros, funciones de tratamiento de textos, generación dinámica de imágenes, tratamiento de documentos XML... y muchas más características.

2.3.2.1.2 ¿Cómo y cuándo funciona el intérprete?

Las páginas que contienen exclusivamente código HTML se pueden desarrollar y probar sin la intervención de un servidor web, ya que el código HTML es interpretado en el navegador del usuario. Esto es así incluso si las páginas incluyen código JavaScript, ya que también es una tecnología que se ejecuta en el navegador del usuario. Sin embargo, PHP se ejecuta en el servidor antes de que la pagina sea enviada al usuario que realizo la petición. A continuación se explica cómo y cuándo entra en funcionamiento el intérprete de PHP.

Cuando llega una petición a un servidor web, este localiza el documento solicitado por el cliente y, en función de una serie de parámetros de la propia configuración del servidor, decide la acción a realizar con el documento. Así, por ejemplo, en caso de tratarse de un documento simple HTML (sin código PHP) o un fichero con una imagen, el servidor se limita a enviar al cliente el documento que solicito y dicho cliente lo muestra en su navegador.

2.3.3 Pagina Web Estática

Una página web estática es una página web que no requiere de muchas actualizaciones constantes.

Está formada sólo por código xHTML y un estilo (CSS).

Los sitios Web estáticos son aquellos sitios enfocados principalmente a mostrar una información permanente, donde el navegante se limita a obtener dicha información, sin que pueda interactuar con la página Web visitada, las Web estáticas están construidas principalmente con hipervínculos o enlaces (links) entre las páginas Web que conforman el sitio, este tipo de Web son incapaces de soportar aplicaciones Web como gestores de bases de datos, foros, consultas on line, e-mails inteligentes.

Esta es una opción más que suficiente para aquellos sitios Web que simplemente ofrecen una descripción de su empresa, quiénes somos, donde estamos, servicios, etc... ideal para empresas que no quieren muchas pretensiones con su sitio Web, simplemente informar a sus clientes de sus productos y su perfil de empresa.

2.3.4 Pagina Web Dinámica

Los sitios **Web dinámicos** son aquellos que permiten crear aplicaciones dentro de la propia Web, otorgando una mayor interactividad con el navegante. Aplicaciones dinámicas como encuestas y votaciones, foros de soporte, libros de visita, envío de e-mails inteligentes, reserva de productos, pedidos on-line, atención al cliente personalizada.

El desarrollo de este tipo de Web son más complicadas, pues requieren conocimientos específicos de lenguajes de programación así como creación y gestión de bases de datos, pero la enorme potencia y servicio que otorgan este tipo de páginas hace que merezca la pena la inversión y esfuerzo invertidos respecto a los resultados obtenidos.

2.4 Bases de datos

Las bases de datos y los sistemas de bases de datos son un componente esencial de la vida cotidiana en la sociedad moderna. Actualmente, la mayoría de nosotros nos enfrentamos a diversas actividades que implican cierta interacción con una base de datos. Por ejemplo, ir al banco a depositar o retirar fondos, realizar una reserva en un hotel o una compañía aérea, acceder al catálogo computarizado de una biblioteca para buscar un libro, o comprar algo *online* (un juguete o un computador, por ejemplo), son actividades que implican que alguien o algún programa de computador acceda a una base de datos. Incluso la compra de productos en un supermercado, en muchos casos, provoca la actualización automática de la base de datos que mantiene el stock de la tienda.

Estas interacciones son ejemplos de lo que podemos llamar aplicaciones de bases de datos tradicionales, en las que la mayor parte de la información que hay almacenada y a la que se accede es textual o numérica. En los últimos años, los avances en la tecnología han conducido a excitantes aplicaciones y sistemas de bases de datos nuevos. La tecnología de los medios de comunicación nuevos hace posible almacenar digitalmente imágenes, clips de audio y flujos (*streams*) de vídeo. Estos tipos de archivos se están convirtiendo en un componente importante de las bases de datos multimedia. Los sistemas de información geográfica (GIS, *Geographic information systems*) pueden almacenar y analizar mapas, datos meteorológicos e imágenes de satélite. Los almacenes de datos y los sistemas de procesamiento analítico en línea (OLAP, *online analytical processing*) se utilizan en muchas compañías para extraer y analizar información útil de bases de datos mucho más grandes para permitir la toma de decisiones. Las tecnologías de tiempo real y bases de datos activas se utilizan para controlar procesos industriales y de fabricación. Y las técnicas de búsqueda en las bases de datos se están aplicando a la WWW para mejorar la búsqueda de la información que los usuarios necesitan para navegar por Internet.

2.4.1 Modelado de datos utilizando el modelo entidad-relación

El modelado conceptual es una fase muy importante para diseñar correctamente una aplicación de base de datos. Por lo general, el término aplicación de base de datos se refiere a una base de datos concreta y a los programas asociados encargados de implementar las consultas y las actualizaciones de la base de datos. Por ejemplo, una aplicación de base de datos BANCO que realiza el seguimiento de las cuentas de los clientes debe incluir programas encargados de implementar las actualizaciones de la base de datos correspondientes a los depósitos y las retiradas de fondos de los clientes. Estos programas ofrecen a los usuarios finales de la aplicación interfaces gráficas de usuario (GUIs) compuestas por formularios y menús. Por tanto, parte de la aplicación de base de datos requerirá el diseño, la implementación y la comprobación de esos programas de aplicación. Tradicionalmente, el diseño y la comprobación de los programas de aplicación se han considerado más como parte del dominio de la ingeniería de software que del dominio de las bases de datos.

Como las metodologías de diseño de bases de datos incluyen cada vez más conceptos que sirven para especificar las operaciones con los objetos de bases de datos, y como las metodologías de ingeniería de software especifican la estructura de las bases de datos que los programas utilizarán y a las que accederán, es evidente que estas actividades están estrechamente relacionadas.

2.4.1.1 Uso de modelos de datos conceptuales de alto nivel para el diseño de bases de datos

El primer paso es la **recopilación de requisitos y el análisis**. Durante este paso, los diseñadores de bases de datos entrevistan a los potenciales usuarios de la base de datos para comprender y documentar sus **requisitos en cuanto a datos**. El resultado de este paso es un conjunto por escrito de los requisitos del usuario. Estos requisitos se deben plasmar en un formulario lo más detallado y completo posible. En paralelo al estudio de estos requisitos, resulta útil especificar los **requisitos funcionales** de la aplicación, que consisten en las **operaciones** (o **transacciones**) definidas por el usuario que se aplicarán a la base de datos,

incluyendo las recuperaciones y las actualizaciones. En el diseño de software, es frecuente utilizar los *diagramas de flujo de datos*, *diagramas de secuencia*, *escenarios* y otras técnicas para especificar los requisitos funcionales. No vamos a explicar ninguna de estas técnicas; normalmente se explican en profundidad en los textos de ingeniería de software.

Una vez recopilados y analizados todos los requisitos, el siguiente paso es crear un **esquema conceptual** para la base de datos, mediante un modelo de datos conceptual de alto nivel. Este paso se denomina **diseño conceptual**.

El esquema conceptual es una descripción concisa de los requisitos de datos por parte de los usuarios e incluye descripciones detalladas de los tipos de entidades, relaciones y restricciones; se expresan utilizando los conceptos proporcionados por el modelo de datos de alto nivel. Como estos conceptos no incluyen detalles de implementación, normalmente son más fáciles de entender y se pueden utilizar para comunicar con usuarios no técnicos. El esquema conceptual de alto nivel también se puede utilizar como referencia para garantizar que se han reunido todos los requisitos de datos de los usuarios y que esos requisitos no entran en conflicto. Esta metodología permite a los diseñadores de bases de datos concentrarse en especificar las propiedades de los datos, sin tener que preocuparse por los detalles del almacenamiento. En consecuencia, es más fácil para ellos crear un buen diseño conceptual de bases de datos.

2.4.1.2 Entidades y atributos

El objeto básico representado por el modelo ER es una entidad, que es una cosa del mundo real con una existencia independiente. Una entidad puede ser un objeto con una existencia física (por ejemplo, una persona en particular, un coche, una casa o un empleado) o puede ser un objeto con una existencia conceptual (por ejemplo, una empresa, un trabajo o un curso universitario). Cada entidad tiene **atributos** (propiedades particulares que la describen). Por ejemplo, una entidad EMPLEADO se puede describir mediante el nombre, la edad, la dirección, el sueldo y el trabajo que desempeña. Una entidad en particular tendrá un valor para cada uno de sus atributos. Los valores de los atributos que describen cada entidad se convierten en la parte principal de los datos almacenados en la base de datos.

Atributos almacenados y derivados. En algunos casos, dos (o más) valores de atributo están relacionados (por ejemplo, los atributos Edad y FechaNac de una persona). Para una entidad de persona en particular, el valor de Edad puede determinarse a partir de la fecha actual (el día de hoy) y el valor de FechaNac de esa persona. El atributo Edad se denomina entonces **atributo derivado** y se dice que se ha **derivado** del atributo FechaNac, que es el denominado **atributo almacenado**. Algunos valores de atributo se pueden derivar de *entidades relacionadas*; por ejemplo, un atributo NumEmpleados de una entidad DEPARTAMENTO puede derivarse contando el número de empleados relacionados con (o que trabajan para) ese departamento.

Valores NULL (nulos). En algunos casos, es posible que una entidad en particular no tenga un valor aplicable para un atributo. Por ejemplo, el atributo NumApto de una dirección sólo se aplica a las direcciones correspondientes a edificios de apartamentos, y no a otros tipos de residencias, como las casas unifamiliares.

De forma parecida, un atributo Licenciaturas sólo se aplica a las personas con carrera universitaria. Para estas situaciones se ha creado un valor especial denominado NULL (nulo). La dicción de una casa unifamiliar tendría el valor NULL para su atributo NumApto, y una persona sin carrera universitaria tendría NULL para Licenciaturas. NULL también se puede utilizar cuando no se conoce el valor de un atributo para una entidad en particular.

El significado del tipo anterior de NULL no es aplicable, mientras que el significado del último es *desconocido*.

La categoría *desconocido* se puede clasificar en dos casos. El primero se da cuando se sabe que existe el valor del atributo pero *no se encuentra*: por ejemplo, si el atributo Altura de una persona aparece como NULL. El segundo caso se da cuando es *no conocido* si existe el valor del atributo: por ejemplo, si el atributo TlfCasa de una persona es NULL.

Atributos complejos. Los atributos compuestos y multivalor se pueden anidar arbitrariamente. Podemos representar el anidamiento arbitrario agrupando componentes de un atributo compuesto entre paréntesis O y separando los componentes con comas, y mostrando los atributos multivalor entre llaves {}. Dichos atributos se denominan **atributos complejos**. Por ejemplo, si una persona puede tener más de una residencia y cada residencia puede tener una sola dirección y varios teléfonos.

2.4.2 MySQL

MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C.

Al contrario de proyectos como Apache, donde el software es desarrollado por una comunidad pública y los derechos de autor del código están en poder del autor individual, MySQL es patrocinado por una empresa privada, que posee el copyright de la mayor parte del código.

Esto es lo que posibilita el esquema de licenciamiento anteriormente mencionado. Además de la venta de licencias privativas, la compañía ofrece soporte y servicios. Para sus operaciones contratan trabajadores alrededor del mundo que colaboran vía Internet.

2.5 Direcciones IP

Las direcciones IP (IP es un acrónimo para Internet Protocol) son un número único e irrepetible con el cual se identifica una computadora conectada a una red que corre el protocolo IP.

Una dirección IP (o simplemente *IP* como a veces se les refiere) es un conjunto de cuatro números del 0 al 255 separados por puntos. Por ejemplo, `users.net` tiene la dirección IP siguiente:

200.36.127.40

En realidad una dirección IP es una forma más sencilla de comprender números muy grandes, la dirección 200.36.127.40 es una forma más corta de escribir el número 3357835048. Esto se logra traduciendo el número en cuatro tripletes.

Anteriormente, las direcciones IP se asignaban usando o que se conocía como clases. Una clase C contenía 256 direcciones, una clase B contiene 16,384 direcciones y una clase A contiene unas 2,097,152 direcciones.

Debido a que las direcciones IP se agotaron súbitamente con este sistema, estos recursos se asignan hoy en día en bloques más pequeños usando un sistema conocido como CIDR (acrónimo de Classless Inter-Domain Routing). De esta forma se logra un mejor aprovechamiento de las direcciones.

Para dar direcciones a redes no conectadas directamente a Internet, se han reservado algunos bloques de direcciones privadas. Estas direcciones pueden ser usadas por cualquier persona en redes Internas pero no pueden ser ruteadas a la Internet global.

Los bloques de direcciones privadas son:

- **192.168.0.0** - 256 clases C o 65,536 direcciones
- **172.16.0.0** - 256 clases B o 4,194,304 direcciones
- **10.0.0.0** - una clase A o 2,097,152 direcciones

Para conectar una red con IPs internas a la Internet global existen varias tecnologías que pueden ser usadas. Estas incluyen los proxies o NAT.

2.6 URL

Un **localizador de recursos uniforme**, más comúnmente denominado **URL** (sigla en inglés de *uniform resource locator*), es una secuencia de caracteres, de acuerdo a un formato modélico y estándar, que se usa para nombrar recursos en Internet para su localización o identificación, como por ejemplo documentos textuales, imágenes, vídeos, presentaciones, presentaciones digitales, etc. Los localizadores uniformes de recursos fueron una innovación fundamental en la historia de la Internet. Fueron usadas por primera vez por Tim Berners-Lee en 1991, para permitir a los autores de documentos establecer hiperenlaces en la World Wide Web. Desde 1994, en los estándares de la Internet, el concepto de URL ha sido

incorporado dentro del más general de URI (*uniform resource identifier*, en español identificador uniforme de recurso), pero el término URL aún se utiliza ampliamente.

El URL es la cadena de caracteres con la cual se asigna una dirección única a cada uno de los recursos de información disponibles en la *Internet*. Existe un URL único para cada página de cada uno de los documentos de la World Wide Web, para todos los elementos de Gopher y todos los grupos de debate USENET, y así sucesivamente.

El URL de un recurso de información es su dirección en Internet, la cual permite que el navegador la encuentre y la muestre de forma adecuada. Por ello el URL combina el nombre del ordenador que proporciona la información, el directorio donde se encuentra, el nombre del archivo, y el protocolo a usar para recuperar los datos.

2.7 Dominio

Un dominio o nombre de dominio es el nombre que identifica un sitio web. Cada dominio tiene que ser único en Internet. Un solo servidor web puede servir múltiples páginas web de múltiples dominios, pero un dominio sólo puede apuntar a un servidor.

Un dominio se compone normalmente de tres partes: en www.empenomorelia.com, las tres uves dobles (www), el nombre de la organización ([empenomorelia](http://www.empenomorelia)) y el tipo de organización ([com](http://www.empenomorelia.com)).

Los tipos de organización más comunes son [.COM](http://www.com), [.NET](http://www.net), [.MIL](http://www.mil), y [.ORG](http://www.org), que se refieren a comercial, network, militar, y organización (originalmente sin ánimo de lucro, aunque ahora cualquier persona puede registrar un dominio [.org](http://www.org)).

Puesto que Internet se basa en direcciones IP, y no en nombres de dominio, cada servidor web requiere de un servidor de nombres de dominio (DNS) para traducir los nombres de los dominios a direcciones IP. Cada dominio tiene un servidor de nombre de dominio primario y otro secundario.

2.8 DNS

Es una base de datos distribuida, con información que se usa para traducir los nombres de dominio, fáciles de recordar y usar por las personas, en números de protocolo de Internet (IP) que es la forma en la que las máquinas pueden encontrarse en Internet.

Hay personas en todo el mundo manteniendo una parte de la base de datos, y esta información se hace accesible a todas las máquinas y usuarios de Internet. El Domain Name System (DNS), o Sistema de Nombres de Dominio, comprende personas, instituciones reguladoras, archivos, máquinas y software trabajando conjuntamente.

Una pieza fundamental en este sistema es el servidor de DNS, o en forma abreviada DNS. El servicio de DNS es indispensable para que un nombre de dominio pueda ser encontrado en Internet.

2.9 Metodologías para el análisis y diseño de un sistema de información web

Para el desarrollo de un sistema de información web hay que tomar en cuenta la realización de un proceso de actividades, las cuales son necesarias para el análisis y diseño del mismo, por mencionar alguna tenemos un sistema que gestiona la base de datos para poder interactuar con ella, y el almacenamiento de toda la información, también dichas actividades deben llevar a cabo la creación de una interfaz amigable, intuitiva y sencilla con un diseño agradable para el usuario, hay que crear una buena estructura de las funciones del sistema para que facilite el trabajo a realizar.

En la actualidad existen varias metodologías de diferentes autores que establecen un procedimiento estricto a seguir a la hora de diseñar un sistema de información web, en este tema se abordaran algunas de estas metodologías por autores que definen el proceso a seguir y se hará una breve explicación de cómo están compuestas:

- Web Site Design Method (WSDM).
- Relationship Navigational Analysis (RNA).

- Object Oriented Hypermedia Design Model (OOHDM).
- Diseño y desarrollo de sistemas de información web

2.9.1 Web Site Design Method (WSDM)

El Método para diseño de sitios web (WSDM) es una propuesta en la que el sistema se define en base a los grupos de usuarios. Fue propuesto por De Troyer y Le une en 1997[De Troyer 1997] y se compone de cuatro fases. Los autores de WSDM dividen los sitios web en dos grupos: Kiosco web y Aplicación Web [Isakowitz 1995]. Las que pertenecen al primer tipo ofrecen al usuario una determinada información y les permite navegar hacia ella. Las aplicaciones web englobarían a aquellos sistemas de información interactivos cuya interfaz de usuario es un conjunto de páginas web. Tras esta división, los autores de WSDM especifican que su propuesta es una propuesta orientada a la primera de ellas, no resultando adecuada para las aplicaciones web.

Fases:

- 1. Modelado de usuarios:** En esta primera fase se intenta detectar los perfiles de usuarios que se van a presentar en la aplicación. Para ello, se deben realizar dos tareas:
 - *Clasificación de usuarios:* en este paso se deben identificar y clasificar a los usuarios que va a tener el sistema. Para ello, los autores de WSDM proponen que se estudie el entorno de la organización donde se vaya a implantar el sistema y los procesos que se vayan a generar. De esta forma, se debe generar un diagrama similar al del diagrama de contexto.
 - *Descripción de los grupos de usuarios:* en esta segunda etapa se describen con más detalles a los grupos de usuarios detectados en la etapa anterior. Para ello, se debe elaborar un diccionario de datos, en principio con formato libre, en el que se indique los requisitos de almacenamiento de información para cada grupo de usuarios y sus características.

2. **Diseño conceptual:** esta fase se divide en dos etapas que son: modelado de objetos para determinar cuáles serán los requisitos del sistema de información que se basan en el modelado de usuarios, y la segunda etapa se estructura en el diseño navegacional que es donde se debe de proponer como el usuario puede navegar dentro del sistema.
3. **Diseño de la implementación:** Una vez definido el modelo es necesario diseñar la interfaz y el entorno de usuario del sistema. Para ello, los autores de WSDM no proponen nada. Únicamente referencian a las propuestas de December y Ginsberg para el desarrollo de interfaces.
4. **Implementación:** En esta fase se pretende conseguir la aplicación ejecutable final en base a los resultados de las fases anteriores. De nuevo tampoco se propone ninguna técnica en concreto para ello.

2.9.2 Relationship Navigational Analysis (RNA)

RNA plantea una secuencia de pasos para el desarrollo de aplicaciones web, centrándose fundamentalmente en la fase de análisis. Aunque la propuesta está orientada al desarrollo de aplicaciones que traten temas jurídicos o de leyes, sus ideas pueden aplicarse en otros entornos. Como veremos a continuación, RNA no hace nuevas propuestas de modelos o técnicas para desarrollar el proceso. Lo que RNA ofrece es una secuencia de pasos a seguir, sin indicar cómo hacer cada uno de ellos.

Fases:

1. **Análisis del entorno:** El propósito de esta fase es el de estudiar las características de la audiencia. Consiste en determinar y clasificar en grupos al usuario final de la aplicación según sus perfiles.
2. **Elementos de interés:** En esta fase se listan todos los elementos de interés de la aplicación. Por elementos de interés se entienden los documentos, las pantallas que se van a requerir, la información, etc. RNA da el objetivo de esta fase de esta forma tan genérica. No especifica cómo se deben listar estos elementos, sólo indica que hay que listarlos.

3. **Análisis del conocimiento:** Aquí hay que desarrollar y plantear un esquema que represente a la aplicación, hay que identificar los objetos, los procesos y las operaciones que se van a poder realizar en la aplicación y las relaciones entre estos.
4. **Análisis de la navegación:** En esta fase se enriquece el esquema obtenido, basándose en las relaciones y objetos definidos en la fase anterior, con las posibilidades de navegación dentro de la aplicación.
5. **Implementación del análisis:** Una vez obtenido el esquema final en el que ya se encuentran incluidos los aspectos de navegación, se pasa el esquema a un lenguaje entendible por la máquina.

2.9.3 Object Oriented Hypermedia Design Model (OOHDM)

OOHDM está basada en HDM, en el sentido de que toma muchas de las definiciones, sobre todo en los aspectos de navegación, planteadas en el modelo de HDM. Sin embargo, OOHDM supera con creces a su antecesor, ya que no es simplemente un lenguaje de modelado, sino que define unas pautas de trabajo, centrado principalmente en el diseño, para desarrollar aplicaciones multimedia de forma metodológica.

Fases:

1. **Diseño conceptual:** Se construye un modelo orientado a objetos que represente el dominio de la aplicación usando las técnicas propias de la orientación a objetos. La finalidad principal durante esta fase es capturar el dominio semántico de la aplicación en la medida de lo posible, teniendo en cuenta el papel de los usuarios y las tareas que desarrollan. El resultado de esta fase es un modelo de clases relacionadas que se divide en subsistemas.
2. **Diseño navegacional:** En OOHDM una aplicación se ve a través de un sistema de navegación. En la fase de diseño navegacional se debe diseñar la aplicación teniendo en cuenta las tareas que el usuario va a realizar sobre el sistema. Para ello, hay que partir

del esquema conceptual desarrollado en la fase anterior. Hay que tener en cuenta que sobre un mismo esquema conceptual se pueden desarrollar diferentes modelos navegacionales (cada uno de los cuales dará origen a una aplicación diferente). La estructura de navegación de una aplicación hipertexto está definida por un esquema de clases de navegación específica, que refleja una posible vista elegida. En OOHDM hay una serie de clases especiales predefinidas, que se conocen como clases navegacionales: Nodos, Enlaces y Estructuras de acceso, que se organizan dentro de un Contexto Navegacional. Mientras que la semántica de los nodos y los enlaces son comunes a todas las aplicaciones hipertexto, las estructuras de acceso representan diferentes modos de acceso a esos nodos y enlaces de forma específica en cada aplicación.

3. **Diseño de interfaz abstracta:** Una vez definida la estructura navegacional, hay que prepararla para que sea perceptible por el usuario y esto es lo que se intenta en esta fase. Esto consiste en definir qué objetos de interfaz va a percibir el usuario, y en particular el camino en el cuál aparecerán los diferentes objetos de navegación, qué objeto de interfaz actuarán en la navegación, la forma de sincronización de los objetos multimedia y el interfaz de transformaciones. Al haber una clara separación entre la fase anterior y esta fase, para un mismo modelo de navegación se pueden definir diferentes modelos de interfaces, permitiendo, así que el interfaz se ajuste mejor a las necesidades del usuario.
4. **Implementación:** Una vez obtenido el modelo conceptual, el modelo de navegación y el modelo de interfaz abstracta, sólo queda llevar los objetos a un lenguaje concreto de programación, para obtener así la implementación ejecutable de la aplicación. En la tabla 5 vemos un resumen de esta fase.

2.9.4 Diseño y desarrollo de sistemas de información web

En la página monografías.com el autor Jesús Humberto Martínez menciona que las páginas web no solo pueden ser como un medio publicitario como ocurrió en años anteriores, sino que también evoluciono su usabilidad para gestionar y vender por internet donde se comenzaron a introducir lenguajes de programación como ASP, C++, PHP, etc., que pueden interactuar con el lenguaje HTML como si fueran uno solo para dar como resultado sitios web que se convierten en sistemas de información muy poderosos y robustos.

Fases:

1. **Análisis preliminar para sistemas de información web:** esta fase se encarga de analizar y estudiar los procesos que se desean automatizar, formas de interactuar con el usuario de acuerdo a la empresa y definir sus requisitos.
2. **Diseño de interfaces:** en esta etapa se llevara a cabo todo lo analizado en la fase 1, como es el tipo de fuente, no saturar demasiado las pantallas con información y seccionarla detalladamente y tener una buena combinación de colores de acuerdo a la imagen de la empresa.
3. **Selección de lenguaje y desarrollo de la aplicación:** una vez evaluada la fase de diseño se seleccionara el lenguaje o plataforma donde se trabajara, para ello se deben tomar en cuenta las funciones que requiere el sistema, conectividad, equipo de cómputo, etc. Después de configurar y tener todo listo se procederá al desarrollo de la aplicación, la cual se realizara de acuerdo a los procedimientos y condiciones establecidas en la etapa de diseño.
4. **Implementación:** En esta fase procederemos a poner en función nuestra aplicación para ello podemos recurrir a las técnicas de implementación que conocemos como:

La **implementación directa:** es decir se va a proceder a trabajar directamente con la aplicación, ya sea porque no existe aplicación alguna o bien se sustituye directamente

la anterior por el nuevo sistema y en la práctica se evaluará para mejoras y ajustes a la aplicación.

Implementación en paralelo: En esta se van a trabajar con las dos aplicaciones, la nueva y la anterior, comparando resultados para ir evaluando la nueva y revisar que esté correcta o bien requiera de mejoras. En la actualidad en este tipo de sistemas se recurre más a la implementación directa ya que regularmente se recurren a aplicaciones diseñadas a innovar, para que la relación entre la aplicación y el usuario final sea amigable, es decir que el usuario se sienta cómodo con la aplicación y fácil de utilizar.

Capítulo 3: Marco metodológico

En este capítulo se hablara sobre la metodología que me pareció más importante de acuerdo al tipo de proyecto que se realizara, para esto se determinó que el mejor fue “**Diseño y desarrollo de sistemas de información web**”, tomando en cuenta esta decisión se procede a seguir las siguientes fases:

1. Análisis preliminar para sistemas de información web
2. Diseño de interfaces
3. Selección de lenguaje y desarrollo de la aplicación
4. Implementación

3.1 Análisis preliminar para sistemas de información web

En esta fase se realizaran entrevistas y se definirán los objetivos de la página, así como los requisitos y la información que pide el cliente que son los siguientes:

- Definir los objetivos del sistema
- Tipo de almacenamiento
- Describir el tipo de usuario del sistema
- Definir los procesos a realizar

3.1.1 Definir los objetivos del sistema

Esta actividad se refiere a la entrevista en donde se conocerá cual será el fin de la página web, que información se quiere mostrar y que tipo de sistema se requiere para las necesidades.

3.1.2 Tipo de almacenamiento

En este proceso ya se conocen los objetivos y la información que se requiere, así que se va a definir el tipo de almacenamiento, como lo es precio, fecha, y tipos enteros como el id.

3.1.3 Describir el tipo de usuario del sistema

Aquí es donde se debe definir al tipo de usuario que va dirigido, de acuerdo a la entrevista y al manejo de la empresa, esta página web no debe llevar un diseño tan vistoso y elegante, porque no va dirigido a usuarios de alto rango social.

3.1.4 Definir los procesos a realizar

Los procesos a realizar son aquellos que se refieren a las funciones que tendrá el sistema, es decir, desde subir un producto, listarlos, modificarlos y eliminarlos hasta enviar correos electrónicos.

3.2 Diseño de interfaces

Ahora entremos a esta etapa en la cual es donde todo lo que se ha analizado será aterrizado a la realidad. Para ello debemos tomar en cuenta los puntos expuestos dentro del análisis, pues dentro de esta etapa debemos realizar:

- Boceto de pantallas
- Diccionario de datos
- Someter a evaluación

3.2.1 Boceto de pantallas

Se definirán las plantillas para todas las páginas, así como la secuencia de flujo con la que se desplegarán según las opciones que se integren al sistema.

3.2.2 Diccionario de datos

Aquí es donde incluiremos la descripción de cada estructura de las tablas o bases de datos que interactuarán con el sistema. Debe llevar, nombre del archivo, fecha de creación, nombre

de campo, tipo de campo, longitud del campo, relaciones con otras bases de datos y nombrar los campos clave.

3.2.3 Someter a evaluación

Se debe mostrar cada avance considerable a la parte interesada, para que de sus opiniones a tiempo sobre cambios y preferencias sobre el tipo de interfaz y así quede satisfecho.

3.3 Selección de lenguaje y desarrollo de la aplicación

Primero debemos seleccionar la plataforma donde se trabajara tomando en cuenta las funciones que se van a realizar. Después de configurar y tener todo listo, procedemos al desarrollo de la aplicación, la cual se realizará de acuerdo a los procedimientos y condiciones establecidas en la etapa de diseño, cada programa debemos poner, autor, nombre del programa, fecha en la que se realizó y la función del programa.

3.3.1 Seleccionar plataforma y base de datos

En esta actividad vamos a ver distintas plataformas para trabajar, y cuales utilizaremos para desarrollar la aplicación, en este caso se eligió Dreamweaver, Photoshop y para la base de datos se eligió MySQL.

3.3.2 Lenguaje de programación

El lenguaje de programación que más se adapta a esta aplicación web, es PHP ya que es una página web sencilla, únicamente mostrar información, dar de alta, modificar y eliminar.

3.3.3 Organización de las carpetas

Cabe mencionar que en la aplicación hay que definir las carpetas que contendrán los diferentes archivos y códigos para una mejor fluidez del sistema, como lo son las imágenes, la conexión a la base de datos y los menús de la barra lateral.

3.5 Implementación y pruebas

En esta fase procederemos a poner en función nuestra aplicación para ello podemos recurrir a las técnicas de implementación que conocemos como:

3.5.1 Implementación directa

Es decir se va a proceder a trabajar directamente con la aplicación, ya sea porque no existe aplicación alguna o bien se sustituye directamente la anterior por el nuevo sistema y en la práctica se evaluará para mejoras y ajustes a la aplicación.

3.5.2 Implementación en paralelo

En esta se van a trabajar con las dos aplicaciones, la nueva y la anterior, comparando resultados para ir evaluando la nueva y revisar que esté correcta o bien requiera de mejoras.

En este caso solo se tiene una aplicación, así que se tienen que hacer pruebas a todas las paginas, por ejemplo revisar que en la sección del administrador todas las paginas nieguen el acceso sin antes acceder al sistema, también hay que introducir mucha información para saber si la paginación funciona, etc.

Modelo Relacional

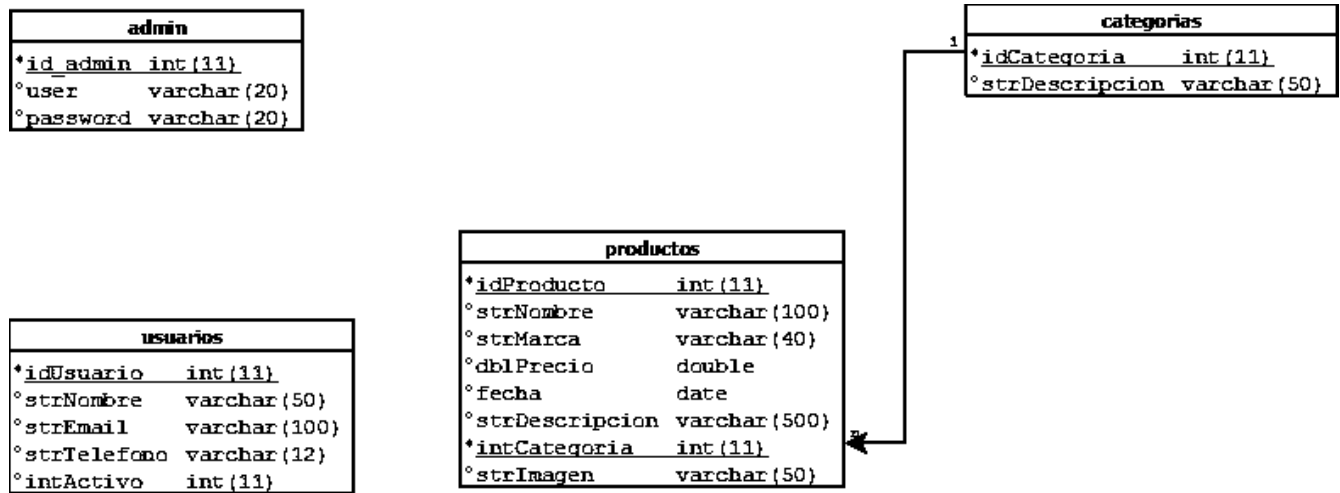


Diagrama que muestra el modelo relacional de las tablas.

Modelo Entidad-Relación

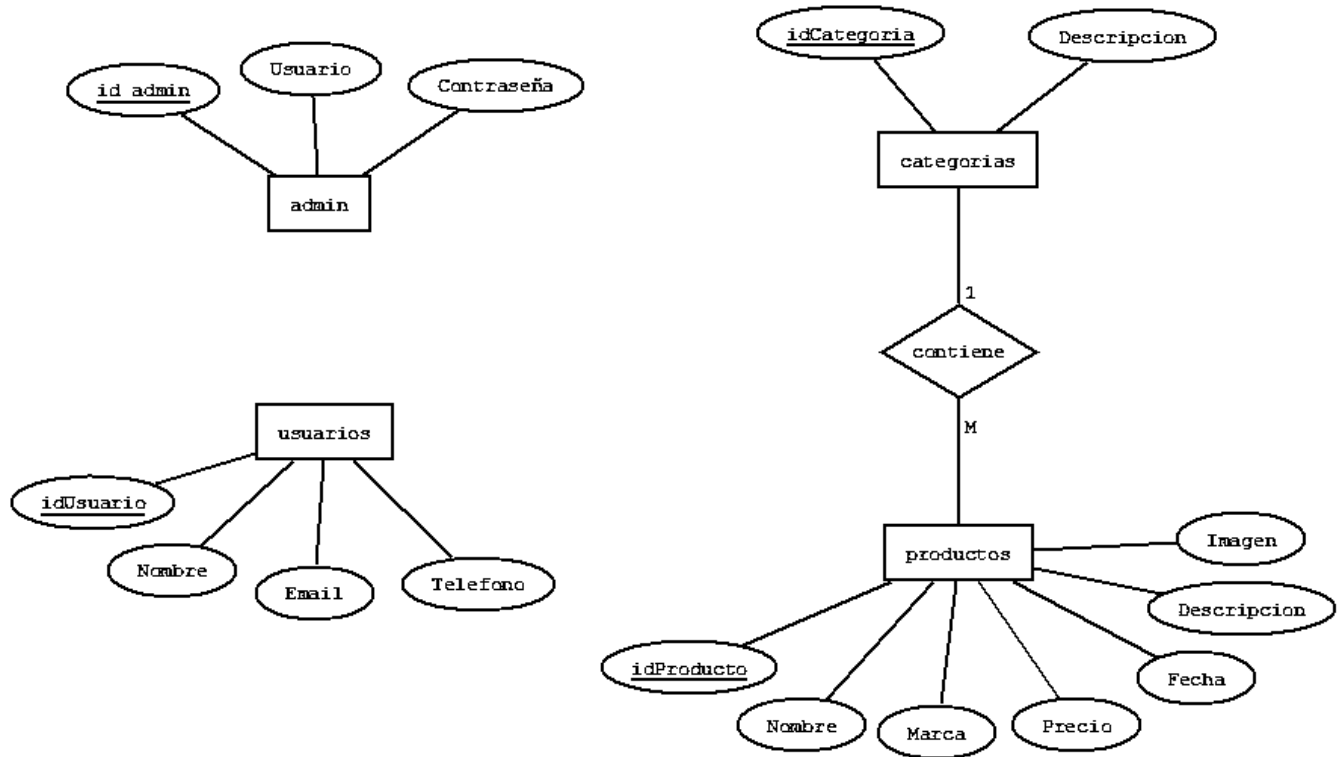
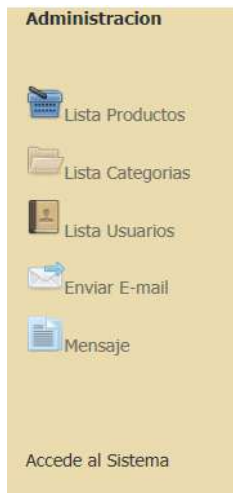


Diagrama que muestra la entidad-relación que hay entre los productos con su categoría para lo cual una categoría contiene muchos productos, y al mostrar cada producto contiene el nombre de su categoría.

Página del Administrador

```
<?php
# FileName="Connection_php_mysql.htm"
# Type="MYSQL"
# HTTP="true"
$hostname_conexionrespuesta = "localhost";
$database_conexionrespuesta = "empenomore_respuestarapida";
$username_conexionrespuesta = "empenomore";
$password_conexionrespuesta = "G4H1Wdt7";
$conexionrespuesta = mysql_pconnect($hostname_conexionrespuesta, $username_conexionrespuesta, $password_conexionrespuesta)
or trigger_error(mysql_error(),E_USER_ERROR);
?>
```

Aquí se muestra el código de acceso a la Base de Datos el cual se encuentra en un archivo que es usado en cada página que requiere dicha conexión.



Bienvenido a la Administración

A login form with a yellow background. At the top is a small icon of a person. Below it are two input fields: the first has a small icon of a person and the second has a small icon of a lock. Below the input fields is a dark gray button with the text "Acceder".

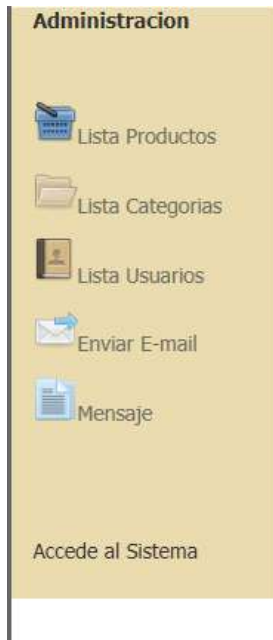
El siguiente código es utilizado para el acceso a la página del administrador, contiene el usuario y la contraseña:

```

if (isset($_POST['usuario'])) {
    $loginUsername=$_POST['usuario'];
    $password=$_POST['password'];
    $MM_fldUserAuthorization = "";
    $MM_redirectLoginSuccess = "bienvenido.php";
    $MM_redirectLoginFailed = "acceso_error.php";
    $MM_redirecttoReferrer = false;
    mysql_select_db($database_conexionrespuesta, $conexionrespuesta);

    $LoginRS_query=sprintf("SELECT `user`, password FROM `admin` WHERE `user`=%s AND password=%s",
        GetSQLValueString($loginUsername, "text"), GetSQLValueString($password, "text"));

```



Error de Acceso

A login form with a user icon above two input fields. The first field is for the username and the second for the password. Below the fields is a button labeled "Acceder".

Intenta de nuevo

Si el usuario o contraseña son incorrectos, entonces nos manda a una página con mensaje de error y el formulario para intentarlo de nuevo:

```

if (!isset($_SESSION)) {
    session_start();
}

$loginFormAction = $_SERVER['PHP_SELF'];
if (isset($_GET['accesscheck'])) {
    $_SESSION['PrevUrl'] = $_GET['accesscheck'];
}

if (isset($_POST['usuario'])) {
    $loginUsername=$_POST['usuario'];
    $password=$_POST['password'];
    $MM_fldUserAuthorization = "";
    $MM_redirectLoginSuccess = "bienvenido.php";
    $MM_redirectLoginFailed = "acceso_error.php";
    $MM_redirecttoReferrer = false;
    mysql_select_db($database_conexionrespuesta, $conexionrespuesta);

    $LoginRS_query=sprintf("SELECT `user`, password FROM `admin` WHERE `user`=%s AND password=%s",
        GetSQLValueString($loginUsername, "text"), GetSQLValueString($password, "text"));

```

Administración
 Lista Productos
 Lista Categorías
 Lista Usuarios
 Enviar E-mail
 Mensaje

 Hola pepe (Modificar) (Salir)

Lista de Productos

 Añadir Producto
 2 Productos Registrados

IMAGEN	NOMBRE	MARCA	PRECIO	CATEGORIA	EDITAR	ELIMINAR
	Celular	Motorola	\$1000	Celulares		
	Television	Samsung	\$8000	Televisiones		

Esta pantalla nos muestra todos los productos que se han dado de alta:

```
mysql_select_db($database_conexionrespuesta, $conexionrespuesta);
$query_Recordset1 = "SELECT * FROM productos, categorias WHERE categorias.idCategoria = productos.intCategoria
ORDER BY categorias.strDescripcion , productos.strNombre";
$Recordset1 = mysql_query($query_Recordset1, $conexionrespuesta) or die(mysql_error());
$row_Recordset1 = mysql_fetch_assoc($Recordset1);
$totalRows_Recordset1 = mysql_num_rows($Recordset1);
```

Contiene un campo de búsqueda para mayor fluidez pensado y usado también en la sección de usuarios, la consulta busca por Nombre o Marca:

```
$varBuscaProducto_Recordset2 = "-1";
if (isset($_GET["buscaproducto"])) {
    $varBuscaProducto_Recordset2 = $_GET["buscaproducto"];
}
mysql_select_db($database_conexionrespuesta, $conexionrespuesta);
$query_Recordset2 = sprintf("SELECT * FROM productos WHERE productos.strNombre
LIKE %s OR productos.strMarca LIKE %s ORDER BY productos.strNombre",
| GetSQLValueString($_ . $varBuscaProducto_Recordset2 . "%", "text"), GetSQLValueString($_ . $varBuscaProducto_Recordset2 . "%", "text"));
$Recordset2 = mysql_query($query_Recordset2, $conexionrespuesta) or die(mysql_error());
$row_Recordset2 = mysql_fetch_assoc($Recordset2);
$totalRows_Recordset2 = mysql_num_rows($Recordset2);
```

Administración

-  Lista Productos
-  Lista Categorías
-  Lista Usuarios
-  Enviar E-mail
-  Mensaje

Hola pepe (Modificar) (Salir)

Detalles del Producto



Nombre: Celular

Marca: Motorola

Precio: \$1000

Fecha: 2011-10-21

Categoría: Celulares

Editar

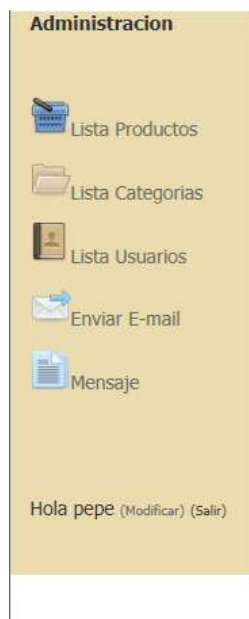
Descripción:
Celular en buenas condiciones

La lista de productos no contiene toda la información de dicho producto, así que para conocer toda la información solo hay que dar clic sobre el producto y como se muestra en el código, el recordID será la variable que recuerde el id del producto para poder mostrarlo:

```

$varProducto_Recordset1 = "0";
if (isset($_GET["recordID"])) {
    $varProducto_Recordset1 = $_GET["recordID"];
}
mysql_select_db($database_conexionrespuesta, $conexionrespuesta);
$query_Recordset1 = sprintf("SELECT * FROM productos WHERE productos.idProducto = %s", GetSQLValueString($varProducto_Recordset1, "int"));
$Recordset1 = mysql_query($query_Recordset1, $conexionrespuesta) or die(mysql_error());
$row_Recordset1 = mysql_fetch_assoc($Recordset1);
$totalRows_Recordset1 = mysql_num_rows($Recordset1);

```



Editar Producto

Nombre:
 Marca:
 Precio:
 Categoría:
 Imagen:
 Fecha:
 Descripción:

Para editar un producto hay que dar clic sobre esa opción vista a la derecha de cada uno de estos, se utilizó el siguiente código para ello.

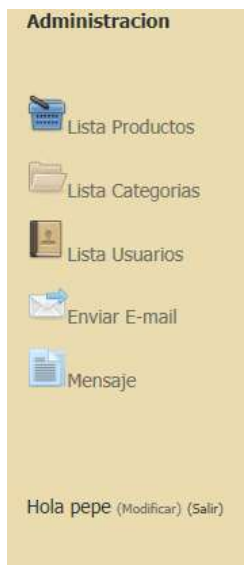
```

if ((isset($_POST["MM_update"])) && ($_POST["MM_update"] == "form1")) {
    $updateSQL = sprintf("UPDATE productos SET strNombre=%s, strMarca=%s, dblPrecio=%s, intCategoria=%s,
    strImagen=%s, fecha=%s, strDescripcion=%s WHERE idProducto=%s",
        GetSQLValueString($_POST['strNombre'], "text"),
        GetSQLValueString($_POST['strMarca'], "text"),
        GetSQLValueString($_POST['dblPrecio'], "double"),
        GetSQLValueString($_POST['intCategoria'], "int"),
        GetSQLValueString($_POST['strImagen'], "text"),
        GetSQLValueString($_POST['fecha'], "date"),
        GetSQLValueString($_POST['strDescripcion'], "text"),

        GetSQLValueString($_POST['idProducto'], "int"));

    mysql_select_db($database_conexionrespuesta, $conexionrespuesta);
    $Result1 = mysql_query($updateSQL, $conexionrespuesta) or die(mysql_error());
}

```



Eliminando..

Eliminar Producto

Para eliminar, igualmente se encuentra esa opción a la derecha de cada producto, en el código se muestra que para eliminar hay que ingresar la variable recordID, la cual recuerda únicamente el id del producto que se eliminara:

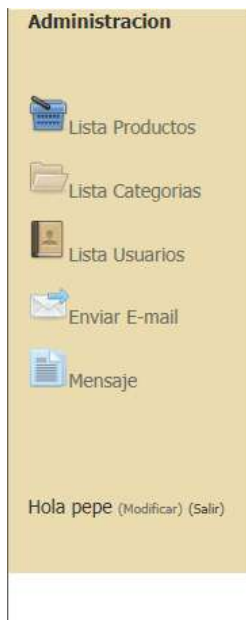
```

if ((isset($_GET['recordID'])) && ($_GET['recordID'] != "")) {
    $deleteSQL = sprintf("DELETE FROM productos WHERE idProducto=%s",
        GetSQLValueString($_GET['recordID'], "int"));

    mysql_select_db($database_conexionrespuesta, $conexionrespuesta);
    $Result1 = mysql_query($deleteSQL, $conexionrespuesta) or die(mysql_error());

    $deleteGoTo = "productos_lista.php";
    if (isset($_SERVER['QUERY_STRING'])) {
        $deleteGoTo .= (strpos($deleteGoTo, '?')) ? "&" : "?";
        $deleteGoTo .= $_SERVER['QUERY_STRING'];
    }
    header(sprintf("Location: %s", $deleteGoTo));
}

```



Añadir Categoría

Descripción:

Para poder ingresar un producto hay que agregar antes una categoría que lo va contener, para eso se utilizó el siguiente código:

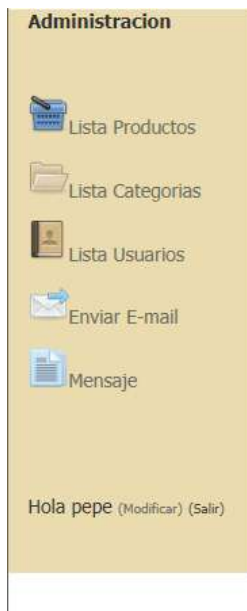
```

if ((isset($_POST["MM_insert"])) && ($_POST["MM_insert"] == "form1")) {
    $insertSQL = sprintf("INSERT INTO categorias (strDescripcion) VALUES (%s)",
        GetSQLValueString($_POST['strDescripcion'], "text"));

    mysql_select_db($database_conexionrespuesta, $conexionrespuesta);
    $Result1 = mysql_query($insertSQL, $conexionrespuesta) or die(mysql_error());

    $insertGoTo = "categorias_lista.php";
    if (isset($_SERVER['QUERY_STRING'])) {
        $insertGoTo .= (strpos($insertGoTo, '?')) ? "&" : "?";
        $insertGoTo .= $_SERVER['QUERY_STRING'];
    }
}

```



Añadir Producto

Nombre: *
 Marca: *
 Precio: *
 Categoría:
 Imagen:
 Fecha: aa/mm/dd
 Descripción:

Para ingresar el producto hay que seleccionar la categoría disponible en un menú que es sacado de la base de datos y que se relaciona con un id en la tabla productos:

```

if ((isset($_POST["MM_insert"])) && ($_POST["MM_insert"] == "form1")) {
    $insertSQL = sprintf("INSERT INTO productos (strNombre, strMarca, dblPrecio, intCategoria, strImagen,
    fecha, strDescripcion) VALUES (%s, %s, %s, %s, %s, %s, %s)",
        GetSQLValueString($_POST['strNombre'], "text"),
        GetSQLValueString($_POST['strMarca'], "text"),
        GetSQLValueString($_POST['dblPrecio'], "double"),
        GetSQLValueString($_POST['intCategoria'], "int"),
        GetSQLValueString($_POST['strImagen'], "text"),
        GetSQLValueString($_POST['fecha'], "date"),
        GetSQLValueString($_POST['strDescripcion'], "text"));

    mysql_select_db($database_conexionrespuesta, $conexionrespuesta);
    $Result1 = mysql_query($insertSQL, $conexionrespuesta) or die(mysql_error());
}

```



Para dar de alta un producto también hay que asignarle una imagen, la cual se sacara del disco duro y será enviada a una carpeta en el servidor, para esto se utilizó un script:

```

<?php if ((isset($_POST["enviado"]))) && ($_POST["enviado"] == "form1")) {
    $nombre_archivo = $_FILES['userfile']['name'];
    move_uploaded_file($_FILES['userfile']['tmp_name'], "../documentos/productos/".$nombre_archivo);

    ?>
    <script>
        opener.document.form1.strImagen.value="<?php echo $nombre_archivo; ?>";
        self.close();
    </script>
    <?php
}
else
{?>

```

Administracion

-  [Lista Productos](#)
-  [Lista Categorías](#)
-  [Lista Usuarios](#)
-  [Enviar E-mail](#)
-  [Mensaje](#)

Hola pepe (Modificar) (Salir)


Envio de E-mail

Desde:

Asunto:

Mensaje:

El siguiente código es utilizado para el envío de correos, el cual saca los correos de la base de datos y envía de uno por uno:

```

<?php
if(isset($_POST['asunto']) && !empty($_POST['asunto']) &&
isset($_POST['desde']) && !empty($_POST['desde']) &&
isset($_POST['mensaje']) && !empty($_POST['mensaje']))
{
    $cab.="Content-type: text/html\r\n";
    mysql_connect("localhost","empenomore","G4H1Wdt7")or die("error en la conexion".mysql_error());
    mysql_select_db("empenomore_respuestarapida")or die("no se conecto a la base de datos".mysql_error());
    $resultado=mysql_query("select * FROM usuarios");
    while($f=mysql_fetch_array($resultado)){
        $destinatario =$f['strEmail'];
        $asunto = $_POST['asunto'];
        $mensaje = $_POST['mensaje'];
        $desde = "From:" . $_POST['desde'];

        mail($destinatario,$asunto,$mensaje,$desde);
    }
}
?>

```

Página Principal



Este código se usa para mostrar las categorías de todos los productos en la barra izquierda de la página principal, en donde llama la opción includes.

```
mysql_select_db($database_conexionrespuesta, $conexionrespuesta);
$query_Recordset1 = "SELECT * FROM categorias ORDER BY categorias.strDescripcion";
$Recordset1 = mysql_query($query_Recordset1, $conexionrespuesta) or die(mysql_error());
$row_Recordset1 = mysql_fetch_assoc($Recordset1);
$totalRows_Recordset1 = mysql_num_rows($Recordset1);
?>
<?php do { ?>
    <p><a href="categorias_lista.php?cat=<?php echo $row_Recordset1['idCategoria']; ?>"><?php echo $row_Recordset1['strDescripcion']; ?></a></p>
    <?php } while ($row_Recordset1 = mysql_fetch_assoc($Recordset1)); ?>
|
<p>
    <?php
mysql_free_result($Recordset1);
?>
```

[Celulares](#)
[Cómputo](#)
[Herramienta](#)
[Televisiones](#)
[Novedades](#)

Novedades



Para mostrar las novedades en la página principal, se usó una consulta donde hay que sacar del campo fecha únicamente los datos de 7 días de antigüedad:

```
mysql_select_db($database_conexionrespuesta, $conexionrespuesta);
$query_DatosProductos = "SELECT * FROM productos WHERE productos.fecha >= SUBDATE( CURDATE( ), INTERVAL 3 DAY) ORDER BY productos.strNombre";
$query_limit_DatosProductos = sprintf("%s LIMIT %d, %d", $query_DatosProductos, $startRow_DatosProductos, $maxRows_DatosProductos);
$DatosProductos = mysql_query($query_limit_DatosProductos, $conexionrespuesta) or die(mysql_error());
$row_DatosProductos = mysql_fetch_assoc($DatosProductos);

if (isset($_GET['totalRows_DatosProductos'])) {
    $totalRows_DatosProductos = $_GET['totalRows_DatosProductos'];
} else {
    $all_DatosProductos = mysql_query($query_DatosProductos);
    $totalRows_DatosProductos = mysql_num_rows($all_DatosProductos);
}
```

[Celulares](#)
[Cómputo](#)
[Herramienta](#)
[Televisiones](#)
 [Novedades](#)

Suscripción

 Nombre:

 Email: *

 Confirma tu Email: * El Email no coincide

 Teléfono:

En la sección de alta para los usuarios, se añadió un campo para repetir el email, esto evitara equivocarse al darse de alta; para detectar en el formulario que sean iguales los emails se usó el siguiente script:

```
<script type="text/javascript">
var sprytextfield1 = new Spry.Widget.ValidationTextField("sprytextfield1", "none", {isRequired:false});
var sprytextfield2 = new Spry.Widget.ValidationTextField("sprytextfield2", "email", {validateOn:["blur"]});
var spryconfirm1 = new Spry.Widget.ValidationConfirm("spryconfirm1", "repetiremail", {validateOn:["blur"]});
</script>
```

Tu Email ya estaba registrado en la Base de Datos.

Si un usuario olvido que ya estaba dado de alta en el sistema, se detectara y se le notificara con la siguiente consulta:

```
$MM_flag="MM_insert";  
if (isset($_POST[$MM_flag])) {  
    $MM_dupKeyRedirect="alta_emailrepetido.php";  
    $loginUsername = $_POST['strEmail'];  
    $LoginRS_query = sprintf("SELECT strEmail FROM usuarios WHERE strEmail=%s", GetSQLValueString($loginUsername, "text"));  
    mysql_select_db($database_conexionrespuesta, $conexionrespuesta);  
    $LoginRS=mysql_query($LoginRS_query, $conexionrespuesta) or die(mysql_error());  
    $loginFoundUser = mysql_num_rows($LoginRS);  
}
```

CONCLUSIONES

Este proyecto se realizó con la finalidad que los lectores estén informados acerca de un desarrollo y diseño sencillo de página web como una búsqueda de información, que requiere de conocimientos de lenguajes de programación particulares así como creación y administración de base de datos, que permiten elevar el potencial de un sitio estándar.

Es importante también mencionar que dentro del proyecto se ha utilizado una metodología para un mejor funcionamiento y estructura de una página web, como lo es el análisis y diseño de sistemas, los tipos de información que deben llevar las tablas en una base de datos y el lenguaje de programación en PHP. El punto de vista desde el cual se analiza el proceso de diseño no es desde la interrelación usuario-tecnología, sino como interacción humana, mediatizada por la tecnología. Es decir, entender a la Web como medio de comunicación entre personas. Esto nos debe ayudar a pensar permanentemente en "el otro", la persona invisible y desconocida para quien hacemos la página Web.

Al realizar el proyecto se tuvieron varios problemas y errores, uno de los problemas que se me presentó fue fusionar la consulta para que mostrara la información del producto con su categoría; pero gracias a ese tipo de inconvenientes en el transcurso de la elaboración de éste se dejó un gran aprendizaje, que ayudó a que fuera favorable el cumplir con los objetivos de éste.

Bibliografía

E. KENDALL, Kenneth y E. KENDALL, Julie. Análisis y diseño de sistemas, Sexta edición. PEARSON EDUCACIÓN, México, 2005.

ELMASRI, Ramez y B. NAVATHE, Shamkant. Fundamentos de sistemas de bases de datos, PEARSON EDUCACIÓN S.A., Madrid, 2007.

GIL RUBIO, Fco. Javier y TEJEDOR CERBEL, Jorge A. Creación de sitios web con php4, Primera edición. MC GRAW HILL, España, 2001.

SENN A., James. Análisis y diseño de sistemas de información. Segunda edición, MCGRAW HILL, Mexico, 1990.

PEÑA AYALA, Alejandro. Ingeniería de software: una guía para crear sistemas de información. EXPEDICION A LA ANTARTIDA, 2001.

SILBERSCHATZ, Abraham. Fundamentos de bases de datos. Quinta edición, MACGRAW HILL, Madrid, 2006.

SCRIBD (Octubre, 2011) <http://es.scribd.com/doc/55449781/31/WSDM-Web-Site-Design-Method>

ENCICLOPEDIA (Octubre, 2011) <http://enciclopedia.us.es/index.php/Internet>

WIKIPEDIA (Octubre, 2011) [http://es.wikipedia.org/wiki/Protocolo_\(inform%C3%A1tica\)](http://es.wikipedia.org/wiki/Protocolo_(inform%C3%A1tica))

HOSTING PERU (Octubre, 2011) http://www.hosting-peru.net/que_es_pagina_web_estatica.html

CLUB DE DESARROLLADORES (Octubre, 2011) <http://www.clubdesarrolladores.com/articulos/categoria/16-mysql>

MONOGRAFIAS (Octubre, 2011) <http://www.monografias.com/trabajos62/sistemas-informacion-web/sistemas-informacion-web.shtml>