

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA



**“SISTEMAS DE APOYO A LOS SERVICIOS ACADEMICOS QUE
OFRECE LA COPADI”**

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO EN COMPUTACIÓN

P R E S E N T A :

EDUARDO BAUTISTA SALAZAR

DIRECTOR DE TESIS:

M. en A. MARÍA DE LOURDES CAMPOS LUNA



Ciudad Universitaria

2013

Dedicatorias y agradecimientos

Mamá y papá, este logro se los debo a ustedes, gracias por su incondicional apoyo perfectamente mantenido a través del tiempo, ustedes son el pilar fundamental de todo lo que soy. Muchas gracias por su cariño y amor, pero sobre todo por creer en mí.

A mis hermanas Aurora y Lorna Elizabeth, así como a ti hermanito Emiliano, por su cariño que me han brindado, por sus consejos y sobre todo por ser la parte de mi vida.

A mi chaparrita hermosa Marisol por estar siempre a mi lado, tanto en los buenos como los malos momentos.

Al Programa Universitario México Nación Multicultural, por haberme permitido ser parte de su distinguido grupo de becarios y preocuparse a cada momento por brindar apoyo para que cada uno de sus integrantes finalizara con éxito su carrera profesional.

A todos ustedes amigos y compañeros de la facultad por haber compartido clases con ustedes.

Eduardo

Índice

Capítulo I

INTRODUCCIÓN	1
1.1 Planteamiento del problema.....	1
1.2 Objetivo.....	2
1.3 Justificación.....	3
1.4 Limitaciones.....	3

Capítulo 2

MARCO TEÓRICO	5
2.1 Antecedentes.....	5
2.2 Internet y la Web.....	5
2.2.1 Diferencia entre Internet y la Web.....	6
2.2.2 Objetivo de la Web.....	6
2.2.3 Principios básicos de arquitectura.....	7
2.2.3.1 Identificador Uniforme de recursos (Uniform Resource Identifier URI).....	8
2.2.3.2 Localizador Uniforme de Recursos (Uniform Resource Locator - URL).....	9
2.2.3.3 Protocolo de Transferencia de Hipertexto (Hypertext Transfer Protocol - HTTP).....	9
2.2.3.4 Características del protocolo HTTP.....	10
2.2.3.5 Mensaje HTTP.....	10
2.2.3.6 Petición.....	11
2.2.3.7 Respuesta.....	12
2.2.3.8 Método.....	13
2.2.3.9 Lenguaje de marcado de hipertexto HTML (HyperText Markup Language).....	13
2.2.3.10 Nociones básicas de HTML.....	14
2.2.3.11 Etiquetas básicas de HTML.....	15
2.2.4 Evolución de la Web.....	16
2.2.4.1 La evolución.....	16
2.2.5 El consorcio World Wide Web.....	18
2.2.5.1 Historia del W3C.....	18
2.3. Lenguajes de programación.....	19
2.3.1 Lenguaje HTML.....	19

2.3.2	Lenguaje PHP.....	20
2.3.3	Lenguaje JavaScript.....	21
2.3.4	Lenguaje ASP.....	22
2.3.5	Lenguaje ASP.NET.....	23
2.3.6	Lenguaje JSP.....	23
2.3.7	Lenguaje Python.....	24
2.3.8	Lenguaje Ruby.....	25
2.3.9	Lenguajes Complementarios.....	26
2.3.9.1	Lenguaje CSS.....	27
2.3.9.1.1	Tipos de estilos CSS.....	27
2.3.9.1.2	Ventajas de usar las hojas de estilo.....	28
2.3.9.2	Lenguaje AJAX.....	29
2.3.9.2.1	Problemas e inconvenientes de usar AJAX.....	29
2.3.9.3	JQuery.....	30
2.4	Base de Datos.....	30
2.4.1	Definición de Base de Datos.....	31
2.4.2	Características.....	31
2.4.3	Propósito de los sistemas de bases de datos.....	31
2.4.4	Objetivos de las Bases de Datos.....	31
2.4.5	Sistemas de Administración de Bases de Datos (DBMS).....	32
2.4.6	Abstracción de Datos.....	33
2.4.7	Modelo de Datos.....	34
2.4.8	Lenguajes de Bases de Datos.....	34
2.4.8.1	Lenguaje de manipulación de datos, LMD.....	34
2.4.8.2	Lenguaje de definición de datos, LDD.....	35
2.4.9	Bases de Datos Relacionales.....	35
2.4.9.1	Componentes de una base de datos relacional y términos técnicos.....	35
2.5	Ingeniería del software.....	36
2.5.1	Metodología de trabajo.....	37
2.5.1.1	El Modelo de Cascada.....	37
2.5.1.2	El Modelo Incremental.....	39
2.5.1.3	Modelo de Espiral.....	40
2.5.2	Proceso Unificado (RUP).....	42
2.5.3	Programación extrema (eXtreme Programming- XP).....	46

Capítulo 3

ANÁLISIS DEL PROBLEMA

3.1	Descripción de la situación actual.....	49
3.2	Determinación de requerimientos del sistema.....	49

3.3	Herramientas a utilizar.....	52
-----	------------------------------	----

Capítulo 4

DISEÑO DEL SISTEMA CURSOS INTER-SEMESTRALES Y BIBLIOTECA DIGITAL

4.1	Desarrollo del sistema.....	55
4.2	Diseño de la página web Cursos Inter-semestrales.....	56
4.2.1	Estructura de Cursos Inter-semestrales.....	62
4.2.2	Objetivo del programa de cursos inter-semestrales.....	62
4.2.3	Proceso de Inscripción.....	63
4.2.4	Reportes.....	66
4.2.5	Programa de cursos.....	67
4.2.6	Conformación de la base de datos.....	68
4.3	Diseño del sistema de Biblioteca Digital.....	72
4.3.1	Menú Inicial.....	72
4.3.2	Sección Libro.....	77
4.3.3	Sección Estudiante.....	80
4.3.4	Sección Préstamo.....	82
4.3.5	Conformación de la base de datos.....	84
4.4	Implementación del Sistema.....	89
4.5	Pruebas y resultados.....	90

CONCLUSIONES

Conclusiones generales.....	97
Contribuciones y mejoras.....	97

BIBLIOGRAFÍA.....	99
-------------------	----

Apéndice A.....	103
-----------------	-----

Índice de figuras.....	126
------------------------	-----

Capítulo I

INTRODUCCIÓN

1.1. Planteamiento del problema

Las universidades hoy en día han cambiado la manera de realizar sus operaciones y cada vez más se implantan sistemas de información que logren mejoras y ayuden a realizar de manera automatizada las actividades.

La Universidad Nacional Autónoma de México como máxima casa de estudios de Latinoamérica desde hace ya varios años ha decidido aplicar este método en sus distintas facultades logrando una mejora en los procesos realizados en estas.

La Facultad de Ingeniería siendo una institución líder en la formación de profesionales e impulsora en el uso de tecnologías ha refrendado su posición destacada en la sociedad.

Asimismo podemos decir que la Facultad de Ingeniería cuenta con más de 200 años de fructífera historia, en la actualidad dentro de esta Facultad se imparten carreras como son: Ingeniería Civil, Mecánica, Geomática, Geofísica, Minas y Metalurgias, Petrolera, Eléctrica Electrónica, en Computación, Telecomunicaciones, Industrial y Mecatrónica. Cuenta también con el Centro de Docencia para la Formación de la Planta Académica y con la Coordinación de Programas de Atención Diferenciada para Alumnos (COPADI) encargada de la atención de estudiantes y con un gran número de publicaciones, programas y equipos para alentar e impulsar su vida académica.

Dado al gran número de alumnos la COPADI es la encargada de brindar servicios y apoyos a la comunidad estudiantil. Para lograr esta labor coordina un conjunto de programas en beneficio directo a los estudiantes de la Facultad.

La implementación del Sistema de Apoyo a los servicios académicos que ofrece la COPADI, tiene como finalidad ofrecer dos servicios que son:

- Publicación y registro vía web de los cursos intersemestrales impartidos por la COPADI y que son destinadas para toda la comunidad universitaria.
- Desarrollar e implementar un sistema computarizado de servicio bibliotecario que nos permitirá llevar el control de préstamo de los libros de la biblioteca de la COPADI a los alumnos pertenecientes al programa de alto rendimiento académico (PARA).

En la actualidad, para los cursos impartidos por COPADI las inscripciones se realizan de manera manual; primeramente se publican carteles en las mamparas con información de dichos cursos para que después cada alumno interesado haga su registro en las oficinas. Con esta metodología no se lleva un control detallado

sobre cuántos alumnos están inscritos en cada curso lo que ocasiona que algunos grupos estén saturados mientras que en otros las vacantes son muy notorias. En este proceso la coordinación de COPADI una vez terminado el registro de alumnos tiene que vaciar la información a formato electrónico aumentando con ello la carga de trabajo.

Además de que se realiza un proceso muy laborioso puede presentarse inconsistencia de datos al momento de estar tecleando los datos del alumno, pues puede darse el caso de que se tecleé mal la información.

Para el caso del sistema bibliotecario, esta coordinación no cuenta con un sistema que le ayude a llevar el control de manera computarizada sobre los libros a su disposición, todo se realiza de manera manual y los registros se realizan en formato de papel, cabe destacar que el préstamo bibliotecario se realiza solamente a alumnos pertenecientes al Programa de Alto Rendimiento Académico (PARA).

Por tales motivos, se detectó esta Coordinación como un área de oportunidad para la implementación de un Sistema de cómputo que ayude en la Inscripción a cursos intersemestrales y permita tener un control de la información manejada en éste y la implementación de un sistema bibliotecario que permita aminorar notablemente la carga que representa para el usuarios y personal todo el proceso que se lleva a cabo desde que se busca en el fichero hasta que, se hace entrega del material solicitado.

1.2. Objetivo

Diseño y la implementación de un sistema de inscripción a cursos intersemestrales y un sistema de control bibliotecario.

El sistema ayudará a los alumnos de la Facultad que pretendan inscribirse a un curso de su elección, brindándoles una mejor comodidad al momento de realizar este proceso, ya que, se podrá realizar desde cualquier parte donde cuente con Internet, así también, ayudará al encargado del área a eficientar su trabajo y esfuerzo, evitando que tenga que vaciar todos los datos de cada alumno a un formato electrónico, sino que esto lo realizará cada alumno desde el sistema.

De igual manera el implementar un sistema de información bibliotecario permitirá controlar el servicio de préstamos y el proceso de consultas a ficheros de la biblioteca de COPADI.

Contar con estos sistemas brindará a esta área un mejor manejo de la información la cual estará disponible en tiempo real.

1.3. Justificación

Las tecnologías de información hoy en día avanzan a pasos muy grandes, como se puede ver en la mayoría de los sistemas en línea que existen actualmente.

Las empresas han empezado a migrar sus procesos a software que permitan la automatización de sus procesos ayudando con ésto a la toma de decisiones y mejor control de sus operaciones.

La Facultad de Ingeniería de la UNAM desde hace tiempo ha implementado estas tecnologías tanto internamente para sus departamentos y coordinaciones, como externa diseñando sistemas para otras Facultades.

Aunque la mayoría de los departamentos y divisiones de la Facultad de Ingeniería ya cuenta con un sistema que automatice procesos y agilice la elaboración de estos, se ha podido ver la falta y necesidad que presenta el área de Atención Diferenciada para Alumnos.

Debido a la problemática que genera el uso de un sistema manual para el control de los préstamos de la biblioteca como lo es la lentitud del sistema, la falta de seguridad en los datos y el descontrol, la implementación de un sistema de información que controle el servicio de préstamos y el proceso de consultas a ficheros de la biblioteca de COPADI, permitirá mejorar en gran forma las actividades que se vinculan directamente con el proceso que representa solicitar el servicio de préstamos de material bibliográfico.

Al no contar con un sistema que les ayude tanto al control bibliotecario como a la inscripción y control de los alumnos que acuden a dichos cursos , se pretende realizar un sistema que ayude a dicha área eficientando los procesos y ayudando al encargado del área con el control de datos.

1.4. Limitaciones

El sistema de inscripciones a cursos intersemestrales COPADI tiene la limitante del tiempo, ya que los cursos se imparten cada 6 meses y no se podrán realizar las pruebas suficientes antes de poner en marcha la primera versión. No obstante se realizarán simulaciones pertinentes para corregir posibles fallas que pudieran existir.

El sistema automatizado de recepción y préstamos de la biblioteca de COPADI tiene como limitaciones: el bibliotecario sólo podrá acceder a los módulos de usuario, material bibliográfico, préstamos, multas, consultas y reportes para la realización de auditorías, asignación de claves y mantenimiento.

El sistema sólo puede ser manipulado por el bibliotecario, el usuario no podrá acceder al sistema ya que contará con un módulo de seguridad donde únicamente se le será asignada una clave de acceso.

Capítulo 2

MARCO TEÓRICO

2.1 Antecedentes

Antes de desarrollar un proyecto es importante tener como antecedentes los conceptos generales y los requerimientos necesarios que se aplicarán para llevarlo a cabo, todo esto nos servirán como herramientas y estrategias de trabajo que a su vez nos permitirán llegar y cumplir el objetivo planteado.

El desarrollo de este proyecto estará constituido por una parte, de una aplicación web donde el alumno accederá mediante el explorador donde podrá elegir el curso que sea de su agrado con el horario y fecha de inicio marcado en cada uno de ellos; y por otra parte un sistema bibliotecario el cual permitirá de una manera sistematizada el control de la misma.

Dentro de lo que abarca una aplicación Web, es necesario definir el servidor que será utilizado, tanto el manejador de la base de datos, como los lenguajes de programación requeridos y sobre todo un punto que es muy importante, la metodología que será implementada para llevar a cabo el sistema.

2.2 Internet y la Web

Internet surgió en el año de 1969, en principio se creó como una red exclusivamente militar llamada ARPANET, por parte de Estados Unidos, con el objetivo de que, en caso de un ataque ruso, se pudiera tener acceso a la información militar desde cualquier punto del país, esta red contaba con 4 ordenadores distribuidos entre distintas universidades del país. Dos años después, ya contaba con unos 40 ordenadores conectados. Tanto fue el crecimiento de la red que su sistema de comunicación se quedó obsoleto. Entonces dos investigadores crearon el Protocolo TCP/IP, que se convirtió en el estándar de comunicaciones dentro de las redes informáticas (actualmente seguimos utilizando dicho protocolo). ARPANET siguió creciendo y abriéndose al mundo, y cualquier persona con fines académicos o de investigación podía tener acceso a la red. Las funciones militares se desligaron de ARPANET y fueron a parar a MILNET, una nueva red creada por los Estados Unidos. La NSF (National Science Foundation) crea su propia red informática llamada NSFNET, que más tarde absorbe a ARPANET, creando así una gran red con propósitos científicos y académicos. El desarrollo de las redes fue abismal, y se crean nuevas redes de libre acceso que más tarde se unen a NSFNET, formando el embrión de lo que hoy conocemos como INTERNET.

Fue hasta el año de 1985 cuando la Internet ya era una tecnología establecida, aunque conocida por unos pocos. En ese tiempo la red era básicamente textual. El desarrollo de NSFNET fue tal que hacia el año 1990 ya contaba con alrededor de 100.000 servidores.

Para 1990, Tim Berners Lee del Centro Europeo de Investigaciones Nucleares (CERN) dirigía la búsqueda de un sistema de almacenamiento y recuperación de datos que permitía vincular información en forma lógica a través de las redes a este sistema lo nombró **World Wide Web (WWW)** o telaraña mundial. El contenido se programaba en un lenguaje de hipertexto con "etiquetas" que asignaban una función a cada parte del contenido. Luego, un programa de computación, un intérprete, era capaz de leer esas etiquetas para desplegar la información. Ese intérprete sería conocido como "navegador" o "browser". En 1993 Marc Andreessen produjo la primera versión del navegador "Mosaic", que permitió acceder con mayor naturalidad a la WWW. Poco después Andreessen encabezó la creación del programa Netscape. A partir de entonces Internet comenzó a crecer más rápido que otro medio de comunicación, convirtiéndose en lo que hoy todos conocemos.

2.2.1. Diferencia entre Internet y la Web

Partiendo del concepto de Internet tenemos que: "Internet es un conjunto descentralizado de redes de comunicación interconectadas que utilizan la familia de protocolos estándares TCP/IP".

De esta forma, Internet se define mediante los estándares TCP/IP.

Mientras que la Web se define como: "La World Wide Web (WWW, o simplemente Web) es un espacio de información donde los elementos de interés, denominados como recursos, se identifican a través de identificadores globales llamados Identificadores de Recurso Uniforme (URI)."

Así que la Web se define mediante otras especificaciones. Las tres primeras especificaciones para las tecnologías Web fueron URL, HTTP y HTML.

2.2.2. Objetivo de la Web

El objetivo de la WWW es convertir el acceso a Internet en algo sencillo para el público en general lo que da a ésta un crecimiento explosivo. Es relativamente sencillo recorrer la Web y publicar información en ella.

Permite unir información que está en un extremo del planeta con otro en un lugar

distante a través de algo que se denomina hipervínculo, al hacer clic sobre éste nos comunica con el otro sector del documento o con otro documento en otro servidor de información.

La Web está diseñada bajo los siguientes criterios (Barners-Lee, 1996):

- Un sistema de información que debe poder asociar registros de manera aleatoria entre objetos, en las distintas bases de datos.
- Si dos usuarios inician usando los sistemas independientemente, hacen un vínculo de un sistema a otro y no requieren desescalar las operaciones para fusionar los vínculos de las bases de datos.
- Intentar imitar el uso de lenguajes de particulares o sistemas operativos para evitar las fallas.
- La información está disponible en todas las plataformas.
- Intentar forzar a los usuarios a que manejen información tal como la manejan las computadoras para evitar fallas.
- Asegurar la precisión de la información para facilitar su ingreso y que sea correcta.

2.2.3. Principios básicos de arquitectura.

La World Wide Web utiliza tecnologías relativamente simples, con una escalabilidad suficiente, eficiente y de gran utilidad que han dado lugar a un espacio de información notable de los recursos relacionados entre sí, cada vez mayor en todos los idiomas, culturas y medios de comunicación.

Las tres bases de la arquitectura web son:

- **Identificación.** URI se utilizan para identificar los recursos. Los URI son un texto corto que identifica unívocamente cualquier recurso (servicio, página, documento, dirección de correo electrónico, etc.).
- **Interacción.** Los agentes web se comunican mediante protocolos normalizados que permiten la interacción a través del intercambio de mensajes que se adhieren a una sintaxis y semántica definida. Un *mensaje* puede incluir datos, así como metadatos acerca de un recurso.
- **Formatos.** La mayoría de los protocolos utilizados para la recuperación de la representación y/o sumisión hacen uso de una secuencia de uno o más mensajes, que en conjunto contienen una carga útil de datos y metadatos

de representación, para transferir la representación de los agentes. La elección del protocolo de interacción pone límites a los formatos de representación de datos y metadatos que pueden ser transmitidos, por ejemplo HTTP por lo general transmite un flujo único de octetos además de los metadatos para identificar mejor el formato de la representación.

2.2.3.1. Identificador Uniforme de Recursos (Uniform Resource Identifier-URI).

Uno de los objetivos de la Web, desde su creación, ha sido la de construir una comunidad global en la que cualquiera de las partes puede compartir información con cualquier otra parte. Para lograr este objetivo, la Web hace uso de un sistema de identificación único y global: el URI. Los URI son una piedra angular de la arquitectura Web, que facilita la identificación de lo que es común en la Web. Un identificador aumenta cuanto más se usan de manera habitual.

Por su diseño un URI identifica cualquier recurso accesible en una red (Wikipedia 2012).

El poder de un vínculo en la Web es que éste puede ser cualquier documento (o cualquier otro tipo de recurso) en el universo de la información. Esto requiere un espacio global de identificadores, y los Identificadores Uniformes de Recursos (URI) son el primer elemento de la arquitectura Web. Es bastante conocido que la estructura comienza con el prefijo http:, que indica un espacio dentro del resto de la cadena.

La sintaxis de un URI consta de dos partes:

- Un identificador del método de acceso (protocolo) al recurso, por ejemplo http: , mailto: , ftp:
- El nombre del recurso.

La sintaxis universal permite el acceso a objetos disponibles utilizando protocolos existentes, y es capaz de ser extendida con el tiempo.

El espacio de los IUR es universal y se trata de un nuevo espacio que puede identificar, nombrar o direccionar la sintaxis, que puede ser mapeada dentro de una sintaxis impresa, puede dar un prefijo y puede llegar a ser parte del espacio de los IUR. Las propiedades de un identificador dependen de las propiedades del espacio dentro del cual se encuentren; algunos son conocidos como nombres de sitio, o bien, algunos como direcciones de sitio. Sin embargo, las propiedades de los sitios no dependen solo de factores técnicos, sino que también dependen de la definición de su sintaxis y del protocolo usado para referenciar los nombres. Cabe

señalar que las propiedades también dependen de factores sociales, de acuerdo a como estos identificadores fueron asignados o reasignados.

2.2.3.2. Localizador Uniforme de Recursos (Uniform Resource Locator-URL)

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. 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).

Existe un URL único para cada página de cada uno de los documentos de la World Wide Web. Un URL se clasifica por su esquema, que generalmente indica el protocolo de red que se usa para recuperar, a través de la red, la información del recurso identificado.

El formato general de una URL es:

`protocolo://máquina/directorio/fichero`

También pueden añadirse otros datos:

`protocolo://usuario:contraseña@máquina:puerto/directorio/archivo`

La especificación detallada se encuentra en la RFC 1738 (<http://www.ietf.org/rfc/rfc1738.txt>), titulada: Uniform Resource Locators (URL).

2.2.3.3. Protocolo de Transferencia de Hipertexto (Hypertext Transfer Protocol- HTTP)

Es el protocolo usado en cada transacción de la World Wide Web, fue desarrollado por el World Wide Web Consortium y la Internet Engineering Task Force. Es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor.

Toda la comunicación entre los clientes y servidores se realiza a partir de caracteres de 8 bits. De esta forma, se puede transmitir cualquier tipo de documento: texto, binario, etc., respetando su formato original.

2.2.3.4. Características del protocolo HTTP

- El protocolo usa referencias dadas por URI.
- Permite la transferencia de objetos multimedia codificando los archivos binarios en cadena de caracteres.
- HTTP se basa en el paradigma petición/respuesta que define la arquitectura cliente/servidor. Generalmente se apoya en TCP/IP (protocolos de transporte y red de Internet).
- El contenido de cada objeto intercambiado está identificado por su clasificación MIME (Multipart Internet Mail Extension) que proporcionan una representación de los datos y permiten que la comunicación entre clientes y servidores esté abierta a negociación.
- Existen ocho verbos que permiten que un cliente pueda dialogar con el servidor, pero los tres verbos más utilizados son: GET, para recoger un objeto, POST, para enviar información al servidor y HEAD, para solicitar las características de un objeto (por ejemplo, la fecha de modificación de un documento HTML).
- Cada operación HTTP implica una conexión con el servidor, que es liberada al término de la misma. No mantiene estado.
- De una petición a la siguiente no se conserva ningún tipo de información, cada conexión es independiente y en principio no hay una memoria de conexiones del cliente.
- Cada petición de un cliente a un servidor no es influida por las transacciones anteriores. El servidor trata cada petición como una operación totalmente independiente del resto.
- Cada objeto al que se aplican los verbos del protocolo está identificado a través de la información de situación del final de la URL.

2.2.3.5. Mensaje HTTP

Un mensaje http consiste en una petición de un cliente al servidor y en la respuesta del servidor al cliente.

Las peticiones y respuestas pueden ser simples o completas. La diferencia es que en las peticiones y respuestas completas se envían cabeceras y un contenido. Este contenido se pone después de las cabeceras dejando una línea vacía entre las cabeceras y el contenido. En el caso de peticiones simples, sólo se puede usar

el método GET y no hay contenido. Si se trata de una respuesta simple, entonces ésta sólo consta de contenido.

Esta diferenciación entre simples y completas se tiene para que el protocolo HTTP/1.0 pueda atender peticiones y enviar respuestas del protocolo HTTP/0.9.

2.2.3.6. Petición

Una petición de un cliente a un servidor ha de incluir el método que se aplica al recurso, el identificador del recurso y la versión del protocolo que usa para realizar la petición.

Para mantener la compatibilidad con el protocolo HTTP/0.9 se permite una petición simple con el formato:

"GET" SP URI CRLF

Donde SP es un espacio, URI es la URI del recurso al que hace referencia la petición y CRLF es un retorno de carro y nueva línea.

En el caso de que la petición se haga con el protocolo HTTP/1.0 o con el protocolo HTTP/1.1 la petición sigue el formato:

Línea de petición

*(Cabeceras)

CRLF

[Contenido]

La línea de petición comienza indicando el método, seguido de la URI de la petición y la versión del protocolo, finalizando la línea con CRLF:

Método SP URI de la petición SP Versión del protocolo CRLF

En el caso del protocolo HTTP/0.9 sólo se permite el método GET, con el protocolo HTTP/1.0 GET, POST y HEAD y con el protocolo HTTP/1.1 OPTIONS, GET, HEAD, POST, PUT, DELETE y TRACE.

2.2.3.7. Respuesta

Después de recibir e interpretar una petición, el servidor debe responder con un mensaje HTTP. Este mensaje tiene el siguiente formato:

Línea de estado

*(Cabeceras)

CRLF

[Contenido]

La línea de estado es la primera línea de la respuesta y consiste en la versión de protocolo que se utiliza, seguida de una indicación de estado numérica a la que puede ir asociada una frase explicativa. El formato es el siguiente:

Versión del protocolo SP Código de estado SP Frase explicativa CRLF

El código de estado es un número de 3 dígitos que indica si la petición ha sido atendida satisfactoriamente o no, y en caso de no haber sido atendida, indica la causa. Los códigos se dividen en cinco clases definidas por el primer dígito del código de estado.

Así tenemos:

- 1xx: Informativo. La petición se recibe y sigue el proceso. Esta familia de respuestas indican una respuesta provisional. Este tipo de respuesta está formada por la línea de estado y las cabeceras. Un servidor envía este tipo de respuesta en casos experimentales.
- 2xx: Éxito. La acción requerida por la petición ha sido recibida, entendida y aceptada.
- 3xx: Redirección. Para completar la petición se han de tomar más acciones.
- 4xx: Error del cliente. La petición no es sintácticamente correcta y no se puede llevar a cabo.
- 5xx: Error del servidor. El servidor falla al atender la petición que aparentemente es correcta.

2.2.3.8. Método

Un método se dice que es seguro si no provocan ninguna otra acción que no sea la de devolver algo (no produce efectos laterales). Estos métodos son el método GET y el método HEAD. Para realizar acciones inseguras (las que afectan a otras acciones) se pueden usar los métodos POST, PUT y DELETE. Aunque esto está definido así, no se puede asegurar que un método seguro no produzca efectos laterales, porque depende de la implementación del servidor.

Un método es idempotente si los efectos laterales para N peticiones son los mismos que para una sola petición. Los métodos idempotentes son los métodos GET, HEAD, PUT y DELETE.

2.2.3.9. Lenguaje de marcado de hipertexto HTML (HyperText Markup Language)

Es el lenguaje de marcado (o marcas) predominante para la elaboración de páginas web que se utiliza para describir y traducir la estructura y la información en forma de texto, así como para complementar el texto con objetos tales como imágenes. El HTML se escribe en forma de «etiquetas».

El lenguaje HTML se basa en la sintaxis SGML (Standard Generalized Markup Language). Esto quiere decir que los diferentes elementos (párrafos, encabezamientos, tablas, listas...) de un documento para la WWW se señalan intercalando etiquetas que indican al navegador cómo debe mostrarlo.

El uso del hipertexto se ha convertido en una de las bases de internet, lo que ha popularizado enormemente el lenguaje HTML, hasta el punto de convertirlo en el estándar. Posteriormente, los navegadores son los encargados de leer este lenguaje y traducir los elementos que representan, mostrándolos directamente en pantalla.

Su funcionamiento se basa en el uso de etiquetas o códigos, que permiten dotar de unas determinadas características y propiedades a los textos y elementos que engloban.

La sintaxis de HTML es:

1. Una etiqueta inicial que señala el comienzo de un elemento
2. Un número determinado de atributos (y sus valores asociados)
3. Una cierta cantidad de contenido (caracteres y otros elementos)
4. Una etiqueta que marca el final

El HTML no es más que una aplicación del SGML (Standard Generalized Markup Language), un sistema para definir tipos de documentos estructurados y lenguajes

de marcas para representar esos mismos documentos. El término HTML se suele referir a ambas cosas, tanto al tipo de documento como al lenguaje de marcas.

2.2.3.10. Nociones básicas de HTML

El diseño en HTML, aparte de cumplir con las especificaciones propias del lenguaje, debe respetar ciertos criterios de accesibilidad web, siguiendo unas pautas o las normativas y leyes vigentes en los países donde se regule dicho concepto. Se encuentra disponible y desarrollado por el W3C a través de las Pautas de Accesibilidad al Contenido Web 1.0 WCAG (actualizadas recientemente con la especificación 2.0), aunque muchos países tienen especificaciones propias, como es el caso de España con la Norma UNE 139803.

El lenguaje HTML puede ser creado y editado con cualquier editor de textos básico, como puede ser Gedit en Linux, el Bloc de notas de Windows, o cualquier otro editor que admita texto sin formato como GNU Emacs, Microsoft Wordpad, TextPad, Vim, Notepad++, entre otros.

HTML utiliza etiquetas o marcas, que consisten en breves instrucciones de comienzo y final, mediante las cuales se determinan la forma en la que debe aparecer en su navegador el texto, así como también las imágenes y los demás elementos, en la pantalla del ordenador.

Un documento HTML comienza con la etiqueta `<html>`, y termina con `</html>`.

Dentro del documento hay dos zonas principales: el encabezamiento, delimitado por las marcas `<HEAD>` y `</HEAD>`, que sirve para definir algunos valores válidos para todo el documento, y el cuerpo, delimitado por las etiquetas `<BODY>` y `</BODY>`, donde reside la información del documento.

El elemento `<TITLE>` contenido dentro del encabezamiento permite especificar el título de un documento HTML. Este título no forma parte del documento en sí pues no se ve en la pantalla principal, sino que sirve como título de la ventana del programa que la muestra.

Existen muchos otros elementos que se engloban dentro del encabezamiento pero para la estructura básica del lenguaje HTML en su nivel básico no son necesarios.

El cuerpo de un documento HTML contiene el texto, imágenes, etc. que, con la presentación y los efectos que se decidan, se presentarán ante el usuario. Dentro del cuerpo se pueden aplicar una serie de efectos a través de diferentes marcas o etiquetas (también otros autores las denominan "directivas").

Así, la estructura de un documento HTML es la siguiente:

```
<HTML>
<HEAD>
<TITLE>Título de la página</TITLE>
</HEAD>
<BODY>
[Aquí se sitúan otras etiquetas que hacen posible visualizar la página]
</BODY>
</HTML>
```

2.2.3.11. Etiquetas básicas de HTML

`<html>`: define el inicio del documento HTML, le indica al navegador que lo que viene a continuación debe ser interpretado como código HTML.

`<script>`: incrusta un script en una web, o llama a uno mediante `src="url del script"`. Se recomienda incluir el tipo MIME en el atributo `type`, en el caso de JavaScript `text/javascript`.

`<head>`: define la cabecera del documento HTML; esta cabecera suele contener información sobre el documento que no se muestra directamente al usuario como, por ejemplo, el título de la ventana del navegador. Dentro de la cabecera `<head>` es posible encontrar:

- `<title>`: define el título de la página. Por lo general, el título aparece en la barra de título encima de la ventana.
- `<link>`: para vincular el sitio a hojas de estilo o iconos. Por ejemplo:`<link rel="stylesheet" href="/style.css" type="text/css">`.
- `<style>`: para colocar el estilo interno de la página; ya sea usando CSS u otros lenguajes similares. No es necesario colocarlo si se va a vincular a un archivo externo usando la etiqueta `<link>`.
- `<meta>`: para metadatos como la autoría o la licencia, incluso para indicar parámetros http (mediante `http-equiv=""`) cuando no se pueden modificar por no estar disponible la configuración o por dificultades con server-side scripting.

`<body>`: define el contenido principal o cuerpo del documento. Esta es la parte del documento html que se muestra en el navegador; dentro de esta etiqueta pueden definirse propiedades comunes a toda la página, como color de fondo y márgenes. Dentro del cuerpo `<body>` es posible encontrar numerosas etiquetas.

- `<h1>` a `<h6>`: encabezados o títulos del documento con diferente relevancia.

- `<table>`: define una tabla.
 - `<tr>`: fila de una tabla.
 - `<td>`: celda de una tabla (debe estar dentro de una fila).
- `<a>`: indica un hipervínculo o enlace, dentro o fuera del sitio web. Debe definirse el parámetro de pasada por medio del atributo *href*. Ejemplo: `Yahoo` se representa como Yahoo.
- `<div>`: división de la página. Se recomienda, junto con *css*, en vez de `<table>` cuando se desea alinear contenido.
- ``: imagen. Requiere del atributo *src*, que indica la ruta en la que se encuentra la imagen. Por ejemplo: ``. Es conveniente, por accesibilidad, poner un atributo *alt*="texto alternativo".

La mayoría de etiquetas deben cerrarse como se abren, pero con una barra (`</>`) tal como se muestra en los siguientes ejemplos:

- `<table><tr><td>Contenido de una celda</td></tr></table>`.
- `<script>Código de un [[script]] integrado en la página</script>`.

2.2.4. Evolución de la Web

La Web ha evolucionado a un punto en el que ya no es necesario ser el creador del sitio o saber de programación para hacerle modificaciones y crearle contenido a la página, y todo en tiempo real. Debido a esta evolución, cada vez más, las personas en todo el mundo disfrutan y necesitan estar conectados a Internet ya sea desde sus hogares, oficinas, desde su auto, mientras caminan en la calle, etc.

Hoy en día el Internet es una necesidad para la mayoría de las personas en el mundo. A través de nuestras computadoras, tablets, smartphones, etc. nos conectamos para buscar información, comunicarnos con otras personas, enviar documentos y hacer muchas otras tareas y actividades.

2.2.4.1. La evolución

Web 1.0. La Web 1.0 empezó en los años 60's de la forma más básica que existe, con navegadores de sólo texto bastante rápidos como ELISA. Después surgió el HTML (Hyper Text Markup Language) que hizo las páginas web más agradables a la vista, así como los primeros navegadores visuales tales como IE, Netscape, etc.

La Web 1.0 es de sólo lectura. El usuario no puede interactuar con el contenido de la página, (nada de comentarios, respuestas, citas, etc) estando totalmente limitado a lo que el Webmaster sube a ésta. Web 1.0 fue creada alrededor del año 1989, aunque se hizo pública más tarde.

Web 2.0. Esta web es la representación de la evolución de las aplicaciones tradicionales hacia aplicaciones web enfocadas al usuario final. *La web 2.0 es de lectura y escritura.* Es colaboración, comunicación e intercambio entre usuarios. Cualquier persona es capaz de agregar contenido sin la necesidad de ser el creador de la Web. Ejemplo de esto son las redes sociales, los Wikis, Foros, Blogs y básicamente cualquier sitio Web donde los usuarios podemos interactuar con la página y con las demás personas creando perfiles, agregando comentarios, imágenes o creando cualquier clase de contenido.

Aunque el término nació a mediados del 2004, la Web 2.0 apareció entre los años 1999 (con sitios como Napster y Blogger) y 2001 (con Wikipedia entre otros). Es la generación de hoy, la que todos usamos. Los usuarios son ahora los creadores de contenido.

Web 3.0. Es un neologismo que se utiliza para describir la evolución del uso y la interacción en la red a través de diferentes caminos. Ello incluye, la transformación de la red en una base de datos, un movimiento hacia hacer los contenidos accesibles por múltiples aplicaciones non-browser, el empuje de las tecnologías de inteligencia artificial, la web semántica, la Web Geoespacial, o la Web 3D. Frecuentemente es utilizado por el mercado para promocionar las mejoras respecto a la Web 2.0.

Las tecnologías de la Web 3.0, como programas inteligentes, que utilizan datos semánticos; se han implementado y usado a pequeña escala en compañías para conseguir una manipulación de datos más eficiente. En los últimos años, sin embargo, ha habido un mayor enfoque dirigido a trasladar estas tecnologías de inteligencia semántica al público general.

Web 4.0. El desarrollo de la Web 3.0 y sus tecnologías llevarán hacia la Web 4.0, la Web Ubicua, donde el objetivo primordial será el de unir las inteligencias donde tanto las personas como las cosas se comuniquen entre sí para generar la toma de decisiones. Para el 2020 se espera que haya agentes en la Web que conozcan, aprendan y razonen como lo hacemos las personas.

Desde los inicios de la web, su primera versión fue la creación de websites estáticos sin más pretensión que dar información a quienes las visualizaban. En su segunda versión (web 2.0), es el usuario quien crea el contenido, y se conecta mediante redes sociales y comunidades. La evolución hacia la web 3.0 se basa en la semántica en la búsqueda de todo este contenido esparcido en el ciberespacio, dando como producto de las búsquedas un retorno ordenado y contextualizado a la situación del individuo. Pero el siguiente eslabón va a ser la suma de todo lo anterior. Serán las propias redes, que antaño fueron creadas por nosotros, las que

tomen la iniciativa en la estructuración de nuevos tejidos relacionales de forma 'inteligente', contextual, geo localizada y modulados a las necesidades del colectivo; siendo ellas las que decidan qué conexiones hacer, cuando y donde.

2.2.5. El consorcio World Wide Web

El consorcio World Wide Web (W3C) es una asociación internacional formada por organizaciones miembro del consorcio, personal y el público en general, que trabajan conjuntamente para desarrollar estándares Web. La misión del W3C es (W3C, 2012):

“Guiar la Web hacia su máximo potencial a través del desarrollo de los protocolos y pautas que aseguren el crecimiento futuro de la Web”

La principal actividad del W3C es desarrollar protocolos y directrices que aseguren el crecimiento de la Web a largo plazo. Los estándares del W3C definen las partes claves que hacen que la World Wide Web funcione.

El W3C crea Estándares Web y pautas para alcanzar su objetivo. En los primeros diez años de su existencia, el W3C ha publicado más de 80 estándares, como son las Recomendaciones del W3C. El W3C centra su trabajo en desarrollar tareas de educación y difusión, y en el desarrollo de software, ofreciendo a su vez un foro abierto para hablar sobre la Web. Con el objetivo de que la Web alcance su máximo potencial, las tecnologías Web más destacadas deben ser compatibles entre sí y permitir que cualquier hardware y software para acceder a la Web funcione conjuntamente. El W3C hace referencia a este objetivo denominándolo inter- operatividad Web. Al publicar estándares abiertos (no propietarios) para lenguajes Web y protocolos, el W3C busca evitar fragmentación del mercado y, por lo tanto, de la Web (W3C, 2012).

2.2.5.1. Historia del W3C

En 1989, Tim Berners-Lee creó la Web. Ya para octubre de 1990 acuñó el término "World Wide Web", desarrolló el primer servidor para la World Wide Web, y el primer programa de cliente (un navegador y un editor). De igual manera creó la primera versión del "Lenguaje de Etiquetado de Hipertexto" (HTML), que no era más que un lenguaje con enlaces de hipertexto que se convirtió en el formato de publicación principal para la Web. Sus especificaciones iniciales para URI, HTTP y HTML fueron mejoradas y debatidas en foros más amplios, mientras la tecnología Web se extendía (W3C, 2012).

En octubre de 1994, Berners-Lee fundó el Consorcio World Wide Web (W3C) en el Laboratorio de Ciencias Informática del Instituto de Tecnología de Massachusetts [MIT/LCS], en colaboración con el CERN, donde la Web tuvo su

origen, con la colaboración del DARPA y de la Comisión Europea. En Abril de 1995, INRIA (Instituto Nacional de Investigación en Informática y Automática) se convirtió en la primera sede Europea del W3C, seguida de la Universidad de Keio en Japón (Campus de Shonan Fujisawa) en Asia, en 1996. En 2003, ERCIM (Consortio Europeo para la Investigación en Informática y Matemáticas) sustituyó, en el papel de sede Europea, a INRIA.

Miembros del W3C, personal y expertos invitados, trabajan juntos para diseñar tecnologías con el objetivo de asegurar que la Web continuará creciendo en el futuro, adaptándose a la creciente diversidad de personas, situaciones, hardware y software (W3C, 2012).

Entre algunas de las iniciativas del W3C está la de mantener sus relaciones con casi 400 organizaciones nacionales y regionales. Estos contactos ayudan al W3C a establecer una cultura de participación global en el desarrollo de la World Wide Web (W3C, 2012).

Las operaciones realizadas por el W3C cuentan con el apoyo financiero procedente de las cuotas de los miembros, subvenciones para investigación y otros recursos de financiamiento, públicos y privados. La gestión de estas operaciones se realiza de forma conjunta por el Laboratorio de Ciencias de la Computación e Inteligencia Artificial del MIT (CSAIL) en los Estados Unidos, el Consortio Europeo para la Investigación en Informática y Matemáticas (ERCIM), en Francia y la Universidad de Keio, en Japón. El W3C tiene también oficinas mundiales en 18 países. Estas oficinas trabajan con sus comunidades Web regionales para ofrecer tecnologías del W3C en los idiomas locales, ayudar a ampliar la base geográfica del W3C y fomentar la participación internacional en las actividades del W3C (W3C, 2012).

2.3. Lenguajes de programación

Es un idioma artificial diseñado para expresar procesos que pueden ser llevadas a cabo por máquinas como las computadoras. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana. Está formado por un conjunto de símbolos, reglas sintácticas, semánticas que definen su estructura y el significado de sus elementos y expresiones. Al proceso por el cual se escribe, se prueba, se depura, se compila y se mantiene el código fuente de un programa informático se le llama programación.

2.3.1. Lenguaje HTML

Las páginas que nos encontramos en Internet, las páginas web, están construidas en un lenguaje de etiquetas denominado lenguaje HTML, que indica básicamente donde colocar cada texto, cada imagen o cada video y la forma que tendrán estos al ser colocados en la página.

El HTML se creó en un principio con objetivos divulgativos. No se pensó que la web llegara a ser un área de ocio con carácter multimedia, de modo que, el HTML se creó sin dar respuesta a todos los posibles usos que se le iba a dar y a todos los colectivos de gente que lo utilizarían en un futuro.

Ventajas:

- Es fácil de entender y utilizar
- Sencillo que permite describir hipertexto.
- Texto presentado de forma estructurada y agradable.
- No necesita de grandes conocimientos cuando se cuenta con un editor de páginas.
- Archivos pequeños.
- Despliegue rápido.
- Lenguaje de fácil aprendizaje.
- Lo admiten todos los exploradores.

Desventajas:

- Lenguaje estático.
- La interpretación de cada navegador puede ser diferente.
- Guarda muchas etiquetas que pueden convertirse en "basura" y dificultan la corrección.
- El diseño es más lento.
- Las etiquetas son muy limitadas.

2.3.2. Lenguaje PHP

PHP son las siglas "Personal Home Page". Es un lenguaje de programación pensado en el web de forma que es ideal para la creación de páginas dinámicas. PHP es la versión libre del sistema equivalente de Microsoft ASP.

PHP es un lenguaje encapsulado dentro de los documentos html (aunque los CGI hechos con PHP terminan en extensión .php y no .html) de forma que se pueden introducir instrucciones php dentro de las páginas. Gracias a esto el diseñador gráfico del web puede trabajar de forma independiente al programador. PHP es interpretado por el servidor (apache) generando un HTML con el resultado de substituir las secuencias de instrucciones PHP por su salida.

Por lo tanto una web dinámica con PHP contiene una serie de documentos php que el servidor apache interpreta proporcionando al cliente documentos html con el resultado de las ordenes php.

Un ejemplo básico del código es:

```
<?php  
echo "Hola Mundo";  
?>
```

Lo que se mostraría con estas líneas de código sería:

Hola Mundo.

Ventajas:

- Multiplataforma.
- Manejo de excepciones.
- Biblioteca nativa de funciones.
- Permite técnicas de programación orientada a objetos.
- Amplia documentación en su página oficial-> PHP.
- Destacada conectividad con MySQL
- Es libre

Desventajas:

- Promueve creación de código desordenado y con un mantenimiento complejo.
- No posee adecuado manejo de unicode.
- Es muy difícil de optimizar.
- Diseñado especialmente hacia un modo de realizar aplicaciones Web que es problemático y obsoleto.

2.3.3. Lenguaje JavaScript

Este es un lenguaje interpretado, no requiere compilación. Fue creado por Brendan Eich en la empresa Netscape Communications. Utilizado principalmente en páginas web. Es similar a java, aunque no es un lenguaje orientado a objetos, el mismo no dispone de herencias. La mayoría de los navegadores en sus últimas versiones interpretan código Javascript.

La característica principal de Javascript, de hecho, es la de ser un lenguaje de scripting, pero, sobre todo, la de ser el lenguaje de scripting por excelencia y, sin lugar a dudas, el más usado. Esta particularidad conlleva una notable serie de ventajas y desventajas según el uso que se le deba dar y teniendo en cuenta la relación que se establece entre el mecanismo cliente-servidor.

La sintaxis de este lenguaje es:

```
<script type="text/javascript"> ... </script>
```

Ventajas:

- Lenguaje de scripting seguro y fiable.
- Los script tienen capacidades limitadas, por razones de seguridad.
- El código JavaScript se ejecuta en el cliente.

Desventajas:

- Código visible por cualquier usuario.
- El código debe descargarse completamente.
- Puede poner en riesgo la seguridad del sitio, con el actual problema llamado XSS (significa en inglés Cross Scripting renombrado a XSS por su similitud con las hojas de estilo CSS).

2.3.4. Lenguaje ASP

Active Server Pages (ASP), es una tecnología propietaria de Microsoft, nace aproximadamente en el año 1996. Se trata básicamente de un lenguaje de tratamiento de textos (scripts), Se utiliza casi exclusivamente en los servidores Web de Microsoft (Internet Information Server y Personal Web Server). Los scripts ASP se ejecutan, por lo tanto, en el servidor y puede utilizarse conjuntamente con HTML y Javascript para realizar tareas interactivas y en tiempo real con el cliente.

El lenguaje ASP a grandes rasgos funciona así: un computador cliente hace una petición de una página ASP. El computador servidor interpreta esta petición y le envía una página web. El resultado final es una página HTML que se le envía al cliente. El usuario no llega nunca a ver el código ASP, sino que ve el resultado de interpretar dicho código, es decir, una página HTML.

Ventajas:

- Usa Visual Basic Script, siendo fácil para los usuarios.
- Comunicación óptima con SQL Server.
- Soporta el lenguaje JScript (Javascript de Microsoft).

Desventajas:

- Código desorganizado.
- Se necesita escribir mucho código para realizar funciones sencillas.
- Tecnología propietaria.

- Hospedaje de sitios web costosos.

2.3.5. Lenguaje ASP.NET

Es un framework para aplicaciones web desarrollado y comercializado por Microsoft. Es usado por programadores para construir sitios web dinámicos, aplicaciones web y servicios web XML.

Apareció en enero de 2002 con la versión 1.0 del .NET Framework, y es la tecnología sucesora de la tecnología Active Server Pages (ASP). ASP.NET está construido sobre el Common Language Runtime, permitiendo a los programadores escribir código ASP.NET usando cualquier lenguaje admitido por el .NET Framework.

Mientras ASP se escribía en VBScript, ASP.net puede ser escrito en cualquier lenguaje soportado por el .net Framework, es decir: VB.net; C# y JScript.net. Otro cambio radical es que ASP.net es un lenguaje totalmente orientado a objetos.

Ventajas:

- Completamente orientado a objetos.
- Controles de usuario y personalizados.
- División entre la capa de aplicación o diseño y el código.
- Facilita el mantenimiento de grandes aplicaciones.
- Incremento de velocidad de respuesta del servidor.
- Mayor velocidad.
- Mayor seguridad.

Desventajas:

- Mayor consumo de recursos.

2.3.6. Lenguaje JSP

Es un lenguaje para la creación de sitios web dinámicos, acrónimo de Java Server Pages. Está orientado a desarrollar páginas web en Java. JSP es un lenguaje multiplataforma. Creado para ejecutarse del lado del servidor.

JSP fue desarrollado por Sun Microsystems (ahora perteneciente a ORACLE). Comparte ventajas similares a las de ASP.NET, desarrollado para la creación de aplicaciones web potentes. Posee un motor de páginas basado en los servlets de Java. Para su funcionamiento se necesita tener instalado un servidor Tomcat.

Sintaxis:

```
<%= new java.util.Date() %>
```

Características:

- Código separado de la lógica del programa.
- Las páginas son compiladas en la primera petición.
- Permite separar la parte dinámica de la estática en las páginas web.
- Los archivos se encuentran con la extensión (jsp).
- El código JSP puede ser incrustado en código HTML.

Elementos de JSP

Los elementos que pueden ser insertados en las páginas JSP son los siguientes:

Código: se puede incrustar código "Java".

Directivas: permite controlar parámetros de los servlets.

Acciones: permite alterar el flujo normal de ejecución de una página.

Ventajas:

- Ejecución rápida de los servlets.
- Crear páginas del lado del servidor.
- Multiplataforma.
- Código bien estructurado.
- Integridad con los módulos de Java.
- La parte dinámica está escrita en Java.
- Permite la utilización de servlets.

Desventajas:

- Complejidad de aprendizaje.

2.3.7. Lenguaje Python

Python es un lenguaje de programación moderno creado por Guido van Rossum a inicios de los años noventa. La implementación canónica, conocida como CPython, está bajo una licencia de software libre y se puede descargar del sitio oficial.

A continuación se listan algunas de las características más sobresalientes del lenguaje Python:

- Es orientado a objetos, pero soporta también los estilos de programación procedural y funcional.
- Corre en múltiples plataformas, incluyendo Windows, Mac OS y Linux.
- Su sintaxis y semántica es sencilla y consistente.
- Utiliza tipos dinámicos.
- Es adecuado tanto para programar scripts como aplicaciones de gran tamaño.
- Es muy modular.
- Cuenta con administración automática de memoria a través de recolección de basura.
- Incluye una poderosa y extensa biblioteca de clases.
- Cuenta con una gran comunidad que se dedica a promover su desarrollo y adopción.
- Por su naturaleza interactiva, resulta ideal para llevar a cabo programación experimental y desarrollo rápido.

Sintaxis:

Ejemplo de una clase en Python:

```
def dibujar_muneco(opcion):
    if opcion == 1:
        C.create_line(580, 150, 580, 320, width=4, fill="blue")
        C.create_oval(510, 150, 560, 200, width=2, fill='PeachPuff')
```

Ventajas:

- Libre y fuente abierta.
- Lenguaje de propósito general.
- Gran cantidad de funciones y librerías.
- Sencillo y rápido de programar.
- Multiplataforma.
- Licencia de código abierto (Opensource).
- Orientado a Objetos.
- Portable.

Desventajas:

- Lentitud por ser un lenguaje interpretado.

2.3.8. Lenguaje Ruby

Su creador, Yukihiro “matz” Matsumoto (comenzó a trabajar en Ruby en 1993, y lo presentó públicamente en 1995), mezcló partes de sus lenguajes favoritos (Perl,

Smalltalk, Eiffel, Ada, y Lisp) para formar un nuevo lenguaje que incorporara tanto la programación funcional como la programación imperativa.

Ruby es un lenguaje de programación interpretado en una sola pasada y su implementación oficial es distribuida bajo una licencia de software libre.

Características:

- Orientado a objetos
- Cuatro niveles de ámbito de variable: global, clase, instancia y local.
- Manejo de excepciones
- Expresiones regulares nativas similares a las de Perl a nivel del lenguaje
- Posibilidad de redefinir los operadores (sobrecarga de operadores)
- Recolección de basura automática
- Altamente portable
- Hilos de ejecución simultáneos en todas las plataformas usando green threads
- Carga dinámica de DLL/bibliotecas compartidas en la mayoría de las plataformas
- Amplia librería estándar
- Soporta inyección de dependencias
- Soporta alteración de objetos en tiempo de ejecución
- Continuaciones y generadores

Ventajas:

- Permite desarrollar soluciones a bajo Costo.
- Software libre.
- Multiplataforma.

Desventajas:

- No soporta polimorfismo de funciones (sobrecarga).
- Procesamiento lento.
- Comunidad de soporte no es tan grande.

2.3.9. Lenguajes Complementarios

Además de los lenguajes de programación mencionados anteriormente, hoy día se emplean lenguajes que permiten dar un mejor diseño a un sitio Web, así como realizar acciones que faciliten el uso del internet. Dentro de estos lenguajes tenemos:

2.3.9.1. Lenguaje CSS

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

La separación de los contenidos y su presentación representa numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos y con significado completo (también llamados “documentos semánticos”). Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes.

Mientras que el lenguaje HTML/XHTML se utiliza para marcar los contenidos, es decir, para designar lo que es un párrafo, lo que es un titular o lo que es una lista de elementos, el lenguaje CSS se utiliza para definir el aspecto de todos los contenidos, es decir, el color, tamaño y tipo de letra de los párrafos de texto, la separación entre titulares y párrafos, la tabulación con la que se muestran los elementos de una lista, etc.

CSS tiene una sintaxis muy sencilla, que usa unas cuantas palabras claves tomadas del inglés para especificar los nombres de sus selectores, propiedades y atributos.

Una hoja de estilos CSS consiste en una serie de *reglas*. Cada regla consiste en uno o más *selectores* y un *bloque de estilos* a aplicar para los elementos del documento que cumplan con el selector que les precede. Cada bloque de estilos se define entre llaves, y está formado por una o varias declaraciones de estilo con el formato propiedad: valor;

La información de estilo puede ser adjuntada tanto como un documento separado o en el mismo documento HTML. En este último caso podrían definirse estilos generales en la cabecera del documento o en cada etiqueta particular mediante el atributo "style".

2.3.9.1.1 Tipos de estilos CSS

Para dar formato a un documento HTML, puede emplearse CSS de tres formas distintas:

- I. Una hoja de estilo interna, que es una hoja de estilo que está incrustada dentro de un documento HTML. (Va a la derecha dentro del elemento <head>). De esta manera se obtiene el beneficio de separar la información del estilo, del código HTML propiamente dicho. Se puede optar por copiar la hoja de estilo incrustada de una página a otra, (esta posibilidad es difícil de

ejecutar si se desea para guardar las copias sincronizadas). En general, la única vez que se usa una hoja de estilo interna, es cuando se quiere proporcionar alguna característica a una página Web en un simple fichero, por ejemplo, si se está enviando algo a la página web.

- II. Una hoja de estilo externa, que es una hoja de estilo que está almacenada en un archivo diferente al archivo donde se almacena el código HTML de la página Web. Esta es la manera de programar más potente, porque separa completamente las reglas de formateo para la página HTML de la estructura básica de la página.
- III. Un estilo en línea (inline), que es un método para insertar el lenguaje de estilo de página, directamente, dentro de una etiqueta HTML. Esta manera de proceder no es totalmente adecuada. El incrustar la descripción del formateo dentro del documento de la página Web, a nivel de código se convierte en una tarea larga, tediosa y poco elegante de resolver el problema de la programación de la página. Este modo de trabajo se podría usar de manera ocasional si se pretende aplicar un formateo con prisa, al vuelo. No es todo lo claro, o estructurado, que debería ser, pero funciona. Este es el método recomendado para maquetar correos electrónicos en HTML.

2.3.9.1.2 Ventajas de usar las hojas de estilo

Las ventajas de utilizar CSS (u otro lenguaje de estilo) son:

- Una página puede disponer de diferentes hojas de estilo según el dispositivo que la muestre o incluso a elección del usuario. Por ejemplo, para ser impresa, mostrada en un dispositivo móvil, o ser "leída" por un sintetizador de voz.
- Control centralizado de la presentación de un sitio web completo con lo que se agiliza de forma considerable la actualización del mismo.
- El documento HTML en sí mismo es más claro de entender y se consigue reducir considerablemente su tamaño (siempre y cuando no se utilice estilo en línea).
- Los navegadores permiten a los usuarios especificar su propia hoja de estilo local que será aplicada a un sitio web, con lo que aumenta considerablemente la accesibilidad. Por ejemplo, personas con deficiencias visuales pueden configurar su propia hoja de estilo para aumentar el tamaño del texto o remarcar más los enlaces.

2.3.9.2 Lenguaje AJAX

AJAX, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se solicitan al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página. JavaScript es el lenguaje interpretado (scripting language) en el que normalmente se efectúan las funciones de llamada de Ajax mientras que el acceso a los datos se realiza mediante XMLHttpRequest, objeto disponible en los navegadores actuales. En cualquier caso, no es necesario que el contenido asíncrono esté formateado en XML.

Las tecnologías que forman AJAX son:

- XHTML y CSS, para crear una presentación basada en estándares.
- DOM, para la interacción y manipulación dinámica de la presentación.
- XML, XSLT y JSON, para el intercambio y la manipulación de información.
- XMLHttpRequest, para el intercambio asíncrono de información.
- JavaScript, para unir todas las demás tecnologías.

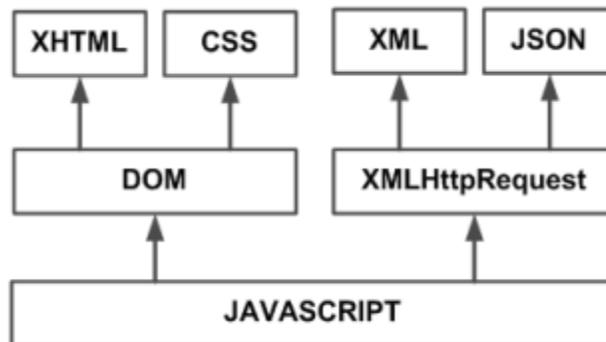


Fig. 1 Tecnologías agrupadas bajo el concepto AJAX

2.3.9.2.1 Problemas e inconvenientes de usar AJAX

- Las páginas con AJAX son más difíciles de desarrollar que las páginas estáticas.
- Las páginas creadas dinámicamente mediante peticiones sucesivas AJAX, no son registradas de forma automática en el historial del navegador, así

que haciendo clic en el botón de "volver" del navegador, el usuario no será devuelto a un estado anterior de la página, en cambio puede volver a la última página que visitó. Soluciones incluyen el uso de IFrames invisible para desencadenar cambios en el historial del navegador y el cambio de la porción de anclaje de la dirección (después de un #).

- Los motores de búsquedas no entienden JavaScript. La información en la página dinámica no se almacena en los registros del buscador.
- Hay problemas usando AJAX entre nombres de dominios. Eso es una función de seguridad.
- El sitio con AJAX usa más recursos en el servidor.
- Es posible que páginas con AJAX no puedan funcionar en teléfonos móviles, PDA u otros aparatos. AJAX no es compatible con todos el software para ciegos u otras discapacidades.

2.3.9.3 JQuery

jQuery es una biblioteca de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. Fue presentada el 14 de enero de 2006 en el BarCamp NYC.

jQuery es software libre y de código abierto y al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio. Además es compatible con todos los navegadores modernos como Firefox 2.0+, Internet Explorer 6+, Safari 2.0.2+, Opera 9 y Google Chrome.

2.4 Base de Datos

Una base de datos (cuya sigla es BD o abreviatura b.d.) es una entidad en la cual se pueden almacenar datos de manera estructurada, con la menor redundancia posible (repeticiones inútiles) con el fin de poder encontrarla después y utilizarla fácilmente.

En el ámbito informático se le llama base de datos a los bancos de información que contienen datos relativos a diversas temáticas y categorizados de distinta manera, pero que comparten entre sí algún tipo de vínculo o relación que busca ordenarlos y clasificarlos en conjunto.

Una base de datos está compuesta por unas estructuras lógicas de almacenamiento denominadas tablas. Una tabla está compuesta por columnas o campos y filas o registros.

2.4.1 Definición de Base de Datos

Se define una base de datos como una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información de una empresa o negocio en particular.

2.4.2 Características

Entre las principales características de los sistemas de base de datos podemos mencionar:

- Independencia lógica y física de los datos.
- Redundancia mínima.
- Acceso concurrente por parte de múltiples usuarios.
- Integridad de los datos.
- Consultas complejas optimizadas.
- Seguridad de acceso y auditoría.
- Respaldo y recuperación.
- Acceso a través de lenguajes de programación estándar.

2.4.3 Propósito de los sistemas de bases de datos.

Los sistemas de bases de datos se diseñan para gestionar grandes cantidades de datos. La gestión de los datos implica la definición de estructuras para almacenar la información así como la provisión de mecanismos para la manipulación de la misma.

Las aplicaciones más usuales son para la gestión de empresas e instituciones públicas. También son ampliamente utilizadas en entornos científicos con el objeto de almacenar la información experimental.

2.4.4 Objetivos de las Bases de Datos

Las BD's tienen como objetivos:

- *Control centralizado de la información.* Los sistemas tradicionales de ficheros nos permiten centralizar la información por medio de varios programas de diseño. Ahora bien, siguiendo las normas estandarizadas de las bases de datos actuales podemos acceder a todos los datos mediante un único programa -administrador de bases de datos.
- *Disminuir la redundancia y evitar la inconsistencia.* Son objetivos básicos de una base de datos eficiente. Disminuir la redundancia consiste en agrupar todos los datos en un mismo objeto sin repetir información. Esto no puede realizarse siempre, con lo cual hay ocasiones en las que se duplica información. Es en este punto donde aparece el concepto de inconsistencia. Una base de datos eficiente no puede tener datos contradictorios en aquellos puntos donde se repite la información (No pueden existir dos D.N.I. iguales asociados a nombres de personas diferentes). Cuanta menos redundancia existe, menos posibilidad de inconsistencia existe.
- *Posibilidad de compartición de datos.* Se consigue disminuyendo la redundancia.
- *Mantenimiento de la integridad.* Deben existir controles que verifiquen que los datos introducidos son correctos, para lo cual se comparan con otros datos, se crean redundancias de control, se hacen validaciones de rango y se permite al usuario modificar los datos.
- *Disponer de un acceso seguro.* Imponer controles para acceder o modificar las bases de datos tales como claves de acceso.
- *Proporcionar independencia de datos.* Establecer una separación entre programas y datos desde una perspectiva física y lógica, de tal forma que cualquier cambio físico o lógico en las estructuras de datos no afecten a los programas de aplicación. Como ejemplo de re-estructuración física estaría una división de uno de los ficheros de datos en dos ficheros. Un cambio lógico en la estructura sería añadir un nuevo campo en la base de datos.

2.4.5 Sistemas de Administración de Bases de Datos (DBMS).

La creación y puesta en práctica del correcto sistema de base de datos asegura que éste dará soporte a las actividades y objetivos de la empresa. Un sistema de administración de base de datos es un grupo de programas que se usan como una interfaz entre una base de datos y programas de aplicaciones, o entre una base de datos y el usuario. Con independencia del modelo que respalden, los DBMS comparten algunas funciones comunes como proporcionar una vista de datos para el usuario, almacenar y recuperar físicamente los datos en una base de datos, permitir la modificación de la base de datos, manipular datos y elaborar informes.

2.4.6 Abstracción de Datos

Se usan estructuras de datos complejos para la representación de los ítems en la base de datos. Esta complejidad se oculta a los usuarios mediante varios niveles de abstracción, lo cual simplifica la interacción de los usuarios con el sistema, como lo muestra la figura 2.

- *Nivel físico.* Es el nivel más bajo, el cual describe cómo se almacenan realmente los datos. Describe en detalle las estructuras de datos complejas de bajo nivel. Los registros se pueden describir como bloques de posiciones consecutivas de almacenamiento.
- *Nivel lógico.* Describe qué datos se almacenan en la base de datos y qué relaciones existen entre esos datos. Describe toda la base de datos en términos de un número pequeño de estructuras relativamente simples. Los registros de este tipo se describen mediante una definición de tipo así como la relación entre estos tipos de registros.
- *Nivel de vistas.* El nivel más elevado de abstracción sólo describe parte de la base de datos. Aunque el nivel lógico usa estructuras más simples, queda algo de complejidad debido a la variedad de información almacenada, en las grandes bases de datos. Este nivel existe para simplificar su interacción con el sistema, puede proporcionar muchas vistas para la misma base de datos. Los usuarios ven un conjunto de programas de aplicación que ocultan los detalles de los tipos de datos.

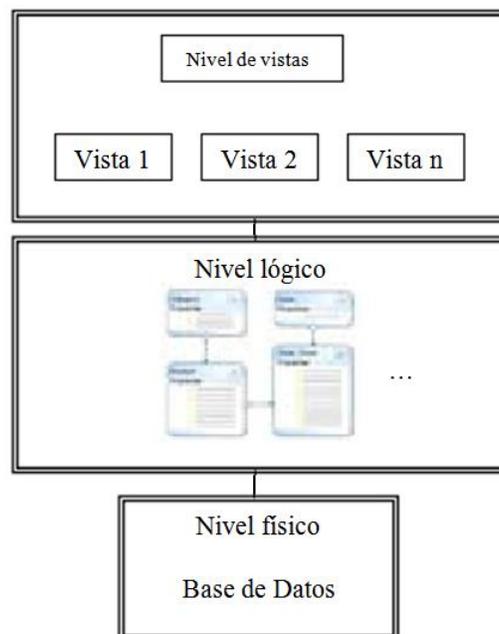


Fig. 2 Los tres niveles de abstracción de datos

2.4.7 Modelo de Datos

El modelo de datos es una colección de herramientas conceptuales para describir los datos, sus relaciones, su semántica y las restricciones de inconsistencia. Ofrecen un modo de describir el diseño de las bases de datos en los niveles físico, lógico y de vistas

- *Modelo relacional*: Usa una colección de tablas para representar tanto los datos como sus relaciones. Cada tabla tiene varias columnas y cada columna tiene un nombre único. Es un ejemplo de un modelo basado en registros ya que la base de datos se estructura en registros de formato fijo de varios tipos.
- *Modelo entidad-relación (E-R)*: Se basa en una percepción del mundo real que consiste en una colección de entidades y de las relaciones entre ellos. Las entidades se describen en las bases de datos mediante un conjunto de atributos. Una relación es una asociación entre varias entidades.
- *Modelo de datos orientado a objetos*: Se puede considerar como una extensión del modelo E-R con los conceptos de la encapsulación, los métodos y la identidad de los objetos.
- *Modelo de datos semiestructurado*: Permite la especificación de datos donde los elementos de datos individuales del mismo tipo pueden tener diferentes conjuntos de los atributos.

2.4.8 Lenguajes de Bases de Datos

Los sistemas de bases de datos proporcionan un lenguaje de definición de datos para especificar el esquema de la base de datos, y un lenguaje de manipulación de datos para expresar las consultas y las modificaciones de la misma.

2.4.8.1 Lenguaje de manipulación de datos, LMD

Un lenguaje de manipulación de datos (LMD) es aquel lenguaje proporcionado por el sistema de gestión de base de datos que permite a los usuarios manipularlos o tener acceso a los datos organizados mediante el modelo de datos correspondientes. Los tipos de acceso son:

- La recuperación de la información almacenada en la base de datos.
- La inserción de información nueva en la base de datos.
- El borrado de la información de la base de datos.
- La modificación de la información almacenada.

Existen fundamentalmente dos tipos:

- Los LMDs procedimentales necesitan que el usuario especifique qué datos se necesitan y cómo obtenerlos.
- Los LMDs declarativos necesitan que el usuario especifique qué datos se necesitan sin que haga falta que especifique cómo obtenerlos.

Una consulta es una instrucción que solicita que se recupere información. La parte de los LMDs implicada en la recuperación de información se denomina lenguaje de consultas

2.4.8.2 Lenguaje de definición de datos, LDD

La estructura de almacenamiento y los métodos de acceso usados por el sistema de bases de datos se especifican mediante un conjunto de instrucciones en un tipo especial de LDD denominado lenguaje de almacenamiento y definición de datos, cuyas instrucciones definen los detalles de implementación de los esquemas de las bases de datos.

2.4.9 Bases de Datos Relacionales

Una base de datos relacional es un conjunto de tablas, bajo una misma identificación, que trabajan en base a relaciones las cuales pueden ser entre dos o más tablas, generando una nueva tabla a partir de los registros que cumplen con el criterio de correspondencia

2.4.9.1 Componentes de una base de datos relacional y términos técnicos

Una base de datos relacional consiste de varios componentes.

Tablas y relaciones. Cada base de datos consiste de componentes fundamentales llamados relaciones. Las relaciones son las asociaciones entre tablas, las tablas consisten en varias líneas y en un número fijo de columnas.

Llaves primarias. Para identificar una fila de datos, cada tabla debe tener una columna que contenga valores únicos. Esta columna es conocida como la llave primaria de una tabla. La columna que es llave primaria no debe tener valores duplicados. Cada base de datos sofisticados deberá de poner atención en esto, para satisfacer las demandas del concepto de una base de datos relacional.

Llaves foráneas. Las llaves foráneas son columnas que están relacionadas a ciertas columnas de otra tabla. Los valores de la llave foránea en la columna son tratados de acuerdo a las restricciones definidas para la columna.

Constraints. Las constraints son atributos definidos para un cierto objeto. Constraints pueden ser incluso restricciones o reglas para definir como se llevarán a cabo ciertas operaciones.

Tipo de datos. Cada columna en una tabla debe de tener un tipo de dato predefinido.

Índices. Los índices son utilizados para acelerar las consultas a la base de datos. Internamente los índices son estructuras complejas de árbol que permiten el rápido acceso a registros individuales.

Secuencias. Las secuencias son utilizadas para generar una lista consecutiva de números que son únicos.

Joins. En el modelo de Base de datos relacional, los datos están almacenados en tablas. Si la información de más de una tabla necesita ser recuperada de la base de datos, los joins necesitan ser llevados a cabo. Joining significa que las tablas están conectadas con otras tablas utilizando una columna en común.

Disparadores. Todas las bases de datos sofisticadas soportan los disparadores. Los disparadores son una característica que puede ser usada para empezar ciertas funciones cada vez que un evento predefinido ocurra. Los disparadores son definidos en las tablas y pueden ser disparados cuando alguna de estas operaciones ocurra: INSERT, UPDATE, o DELETE.

Vistas. Las vistas son tablas virtuales. Esto significa que los usuarios pueden seleccionar datos de una vista justo como si lo seleccionaran de una tabla. Sin embargo, en realidad una vista no es una tabla real, sino que es el resultado de ejecutar una declaración SQL.

Expresiones combinadas. Son utilizadas para realizar operaciones con múltiples líneas devueltas por una consulta.

Funciones combinadas. Son utilizadas para realizar cálculo de datos, como obtener el máximo, el mínimo o el promedio.

2.5 Ingeniería del software

Al igual que otros tipos de ingeniería, la ingeniería de software no solamente consiste en producir productos, más bien significa producir los productos de la mejor relación costo-beneficio posible. Asignando una cantidad de recursos

ilimitados, la mayoría de los problemas de software podrían ser resueltos, pero el reto de la ingeniería de software es producirlo con alta calidad y una cantidad finita de recursos en el tiempo estimado.

Sommerville (1989) también menciona cuatro atributos que cualquier producto de software bien diseñado debe tener:

- *Fácil Mantenimiento*: Un software con una larga vida útil, está sujeto a un cambio regular, por lo que es importante que el software sea escrito y documentado de modo que los cambios puedan ser hechos sin costos inmerecidos
- *Confiable*: Un nivel apropiado de confiabilidad es esencial, si el software es de uso general
- *Eficiente*: Esto no necesariamente significa que el desempeño debe ajustarse al hardware donde el software se ejecuta. Ciertamente, tratar de maximizar la eficiencia puede hacer al software mucho más difícil de cambiar. Sin embargo, que sea eficiente significa que el software no debe desperdiciar el uso de recursos del sistema como la memoria y el procesador.
- *Fácil de Usar*: La mayoría de los usuarios no utilizan el potencial completo del software, porque la interfaz en la cual se ofrece, es difícil de utilizar. En el diseño de la interfaz se debe de tomar en cuenta las capacidades y los antecedentes de los usuarios a los que va dirigido el software, es decir se debe estar hecho a la medida de los usuarios.

En este aspecto, la ingeniería de software a través de las metodologías de desarrollo de software provee una guía que sirve como base para el desarrollo de software de manera exitosa.

2.5.1 Metodología de trabajo

La función primaria de una metodología de proceso de desarrollo de software es el determinar el orden de las etapas envueltas en el desarrollo, evolucionar y establecer el criterio de transición para procesar de una etapa a la siguiente.

2.5.1.1 El Modelo de Cascada

El modelo de cascada es atribuido a Walter Royce, (Van 2000) el cual es descrito por Pressman a continuación:

La Figura 3 ilustra el ciclo de vida clásico para la ingeniería del software. Algunas veces llamado 'modelo en cascada', el ciclo de vida exige un enfoque sistemático y secuencial del desarrollo de software que comienza en el nivel del sistema y progresa a través del análisis, diseño, codificación, prueba y mantenimiento.

Conceptualizado a partir del ciclo convencional de una ingeniería, el ciclo de vida abarca las siguientes actividades:

- *Ingeniería y análisis del sistema.* Debido a que el software es siempre parte de un sistema mayor, el trabajo comienza estableciendo los requisitos de todos los elementos del sistema y luego asignando algún subconjunto de estos requisitos del software. Este planteamiento del sistema es esencial cuando el software debe interrelacionarse con otros elementos, tales como hardware, personas y bases de datos. La ingeniería y el análisis del sistema abarcan los requisitos globales a nivel del sistema con una pequeña cantidad de análisis y diseño a nivel superior.
- *Análisis de los requisitos del software.* El proceso de recopilación de requisitos se centra e intensifica especialmente para el software. Para comprender la naturaleza de los programas que hay que construir, el ingeniero de software (analista) debe comprender el ámbito de la información del software, así como la función, el rendimiento y las interfaces requeridas. Los requisitos, tanto del sistema como del software, se documentan y se revisan con el cliente.
- *Diseño.* El diseño del software es realmente un proceso múltiple que se enfoca sobre cuatro atributos distintos del programa: la estructura de los datos, la arquitectura del software, el detalle procedimental y la caracterización de la interfaz. El proceso de diseño traduce los requisitos en una representación del software que pueda ser establecida de forma que obtenga la calidad requerida antes de que comience la codificación. Al igual que los requisitos, el diseño se documenta y forma parte de la configuración del software.
- *Codificación.* El diseño debe traducirse en forma legible para la máquina. El paso de codificación realiza esta tarea. Si el diseño se realiza de una manera detallada, la codificación puede realizarse mecánicamente.
- *Prueba.* Una vez que se ha generado el código, comienza la prueba del programa. La prueba se centra en la lógica interna del software, asegurando que todas las sentencias se han probado, se realizan pruebas que aseguren que la entrada definida produce los resultados que realmente se quieren.
- *Mantenimiento.* El software, indudablemente sufrirá cambios después de que se entregue al cliente una posible excepción es el software empotrado. Los cambios ocurrirán debido a que se hayan encontrado errores, a que el

software deba adaptarse a cambios del entorno externo por ejemplo, un cambio solicitado debido a que se tiene un nuevo sistema operativo o dispositivo periférico, o debido a que el cliente requiera ampliaciones funcionales o de rendimiento. El mantenimiento del software aplica cada uno de los pasos procedentes del ciclo de vida a un programa existente en vez de a uno nuevo.

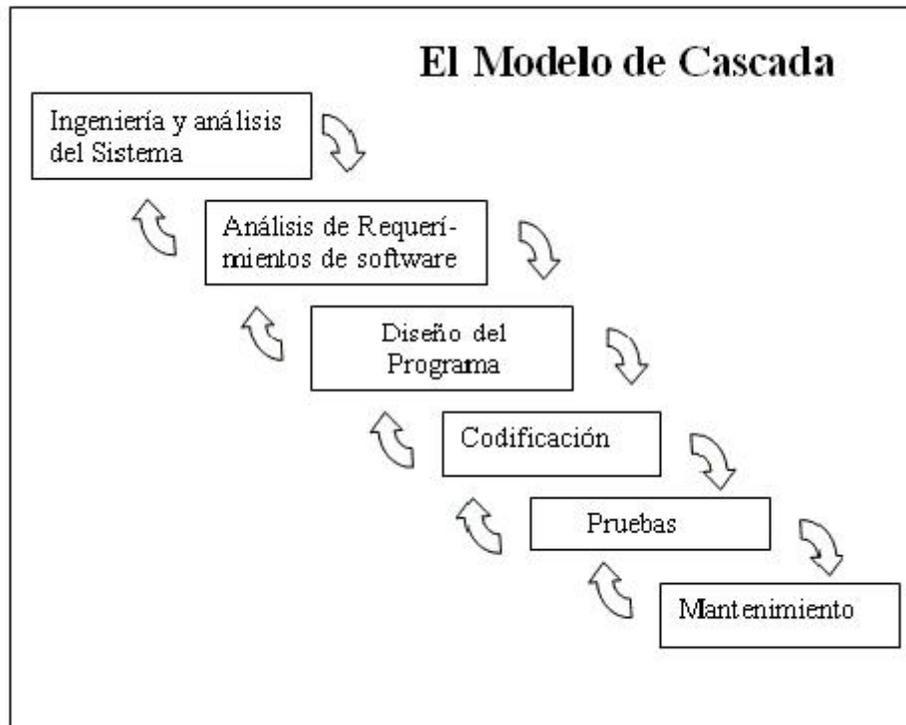


Fig. 3 Esquema de Fases del Modelo de Cascada

2.5.1.2 El Modelo Incremental

En el modelo incremental el software es construido, no escrito, ésto es, el software va siendo construido paso a paso, de la misma manera en que se construye un edificio. Mientras un producto de software está en el proceso de ser desarrollado, cada paso adiciona algo sobre lo que se ha hecho anteriormente. Día a día los módulos del sistema son añadidos, el desarrollo del producto se procesa incrementalmente hasta la conclusión del producto.

Por supuesto no es muy cierto que el progreso se hará diariamente, así como un constructor que tiene que tirar una pared mal posicionada o remplazar una ventana que un pintor ha quebrado, algunas veces es necesario re-especificar, re-diseñar, re-codificar o en el peor de los casos, eliminar lo que ya se había completado y empezar de nuevo.

El hecho de que este software es construido incrementalmente, ha llevado a los desarrolladores del modelo que explotan ese aspecto del desarrollo de software, a llamarlo modelo incremental. El producto es diseñado, implementado, integrado y probado como una serie de construcciones incrementales, donde la construcción consiste en piezas de código de varios módulos interactuando para proveer de una capacidad funcional específica.

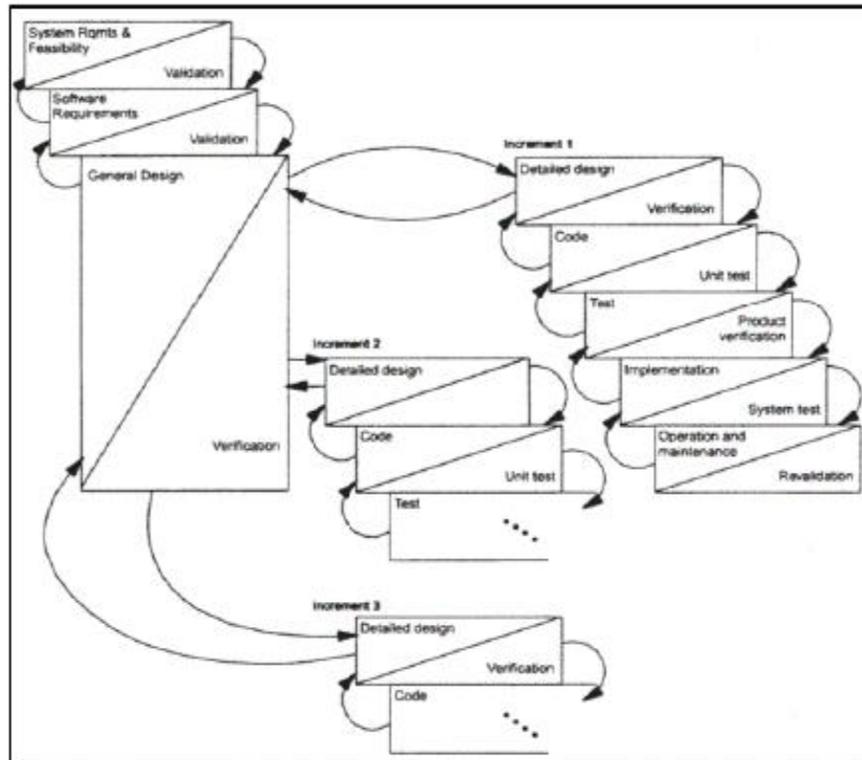


Fig. 4 Fases del Modelo Incremental

2.5.1.3 Modelo de Espiral

Por lo general un elemento de riesgo está envuelto en el desarrollo de software, por ejemplo, personal clave puede renunciar antes que el producto haya sido adecuadamente documentado. También que el proveedor de hardware en el cual el producto es críticamente dependiente puede irse a la bancarrota; o la compañía puede investigar y desarrollar un sistema de administración de base de datos, pero antes que el producto pueda ser vendido, un paquete funcionalmente equivalente y a precio más bajo es anunciado por la competencia.

El modelo de espiral completo se muestra en la figura 5. La dimensión radial representa el costo acumulativo al día. La dimensión angular representa el progreso a través de la espiral. Cada ciclo de la espiral representa una fase, una fase empieza (en la parte izquierda superior del cuadrante) definiendo los

objetivos de esa fase, alternativas para alcanzar esos objetivos, y obligaciones impuestos a esas alternativas. Este proceso resulta en una estrategia para alcanzar esos objetivos, esta estrategia es analizada desde el punto de vista de riesgo. Muchos intentos son hechos para resolver cada riesgo potencial, en algunos casos construyendo un prototipo. Si el riesgo no puede ser resuelto, el proyecto deberá terminar inmediatamente, bajo algunas circunstancias, una decisión podría ser continuar el proyecto pero a una escala menor, si todos los riesgos son resueltos exitosamente, el próximo paso en el desarrollo debe ser iniciado (cuadrante superior derecho). Este cuadrante del modelo de espiral es similar al modelo de cascada. Finalmente, los resultados de esta fase son evaluados y se planea la siguiente fase.

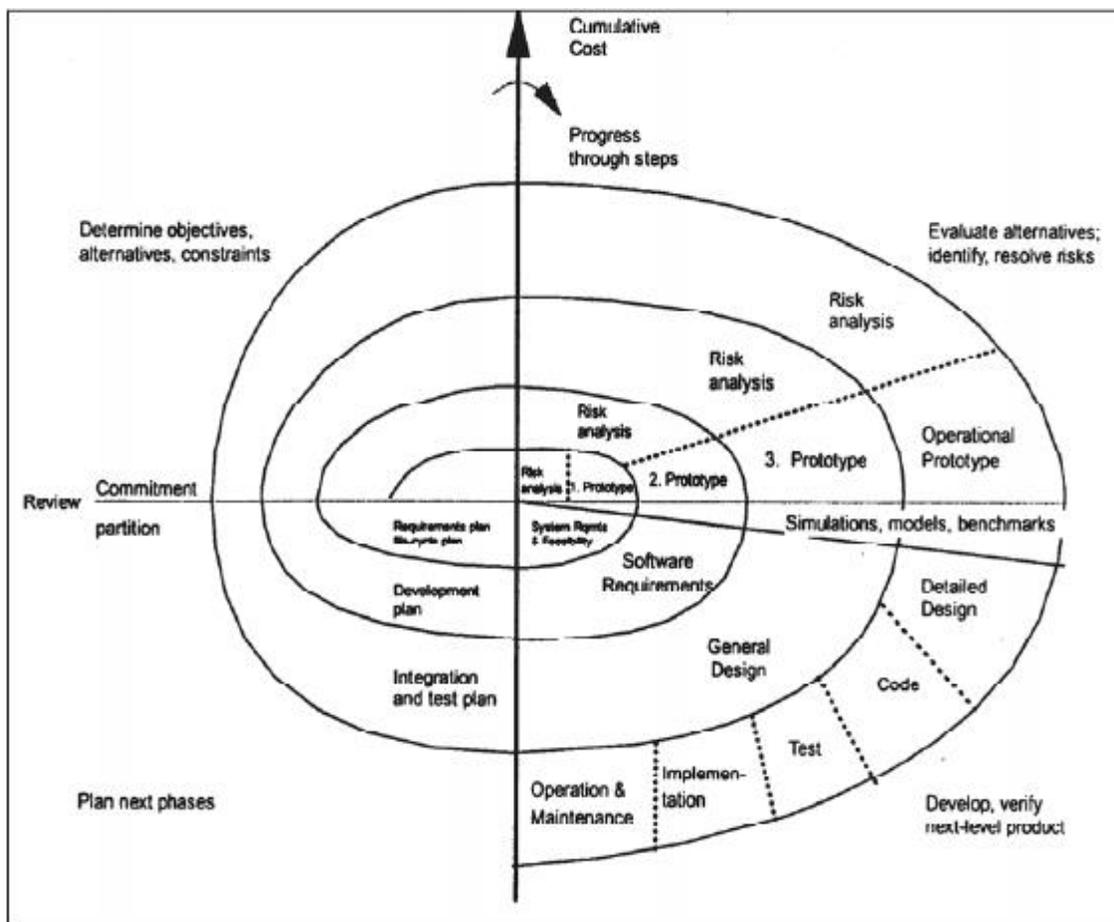


Fig. 5 Esquema de las fases del Modelo Incremental

Pero, al igual que otros paradigmas, el modelo en espiral no es la panacea. Puede ser difícil convencer a grandes clientes (particularmente en situaciones bajo control) de que el enfoque evolutivo es controlable. Si no se descubre un riesgo importante, indudablemente surgirán problemas. Por último, el modelo en sí mismo es relativamente nuevo y no se ha usado tanto como el ciclo de vida o la

creación de prototipos. Pasarán unos cuantos años antes de que se pueda determinar con absoluta certeza la eficacia de este importante nuevo paradigma.

2.5.2 Proceso Unificado (RUP)

La metodología del Proceso unificado de Racional (RUP) es un compendio de ideas y experiencias de los líderes de la industria, socios y literalmente cientos de proyectos reales cuidadosamente sintetizados dentro de un conjunto de mejores prácticas, flujos de trabajo y objetos para un desarrollo de software iterativo. Cuando son usadas en combinación las mejores prácticas promovidas por la metodología del RUP, las cuales incluyen desarrollo iterativo, administración de requerimientos, uso de arquitecturas de componentes, modelado visual, administración de cambios y continua verificación de calidad, golpea a la raíz de los problemas de desarrollo de software, ayudando a evitar caídas comunes con el apalancamiento de nuevas tecnologías y herramientas. Usando esta metodología probada y compartiendo un proceso comprensivo, el equipo será capaz de comunicarse más efectivamente y trabajar más eficientemente.

El Proceso Unificado guía a los equipos de proyecto en la administración del desarrollo iterativo de un modo controlado, mientras se balancean los requerimientos del negocio, el tiempo del mercado y los riesgos del proyecto. El proceso describe los diversos pasos involucrados en la captura de los requerimientos y el establecimiento de una guía arquitectónica para diseñar y probar el sistema hecho de acuerdo a los requerimientos y a la arquitectura. El proceso describe qué entregables producir, cómo desarrollarlos y también provee patrones.

El Proceso Unificado es un proceso de desarrollo de software configurable que se adapta a proyectos que varían en tamaño y complejidad. Se basa en muchos años de experiencia en el uso de la tecnología de objetos, en el desarrollo de software de misión crítica en una variedad de industrias. Uno de los componentes clave es el UML.

El Proceso Unificado ha adoptado un enfoque que se caracteriza por:

- Interacción continua con el usuario desde un inicio.
- Mitigación de riesgos antes de que ocurran.
- Liberaciones frecuentes de ejecutables.
- Aseguramiento de la calidad.
- Participación del equipo en todas las decisiones del proyecto.
- Anticiparse al cambio de requerimientos

Siendo las características primordiales del Proceso Unificado las siguientes:

- Proceso Iterativo e incremental.

- Proceso Centrado en la arquitectura.
- Proceso Guiado por casos de uso.
- Confrontación de riesgos.

La estructura del RUP se muestra en dos dimensiones:

El eje horizontal representa el tiempo y muestra los aspectos del ciclo de vida de procesos como su desarrollo.

El eje vertical representa disciplina, la cual agrupa actividades lógicamente por naturaleza. La primera dimensión representa el aspecto dinámico del proceso como es promulgado, y es expresado en términos de fases, iteraciones y pequeñas metas (Milestones). La segunda dimensión representa el aspecto estático del proceso: Cómo está descrita en términos de componentes de proceso, disciplina, actividades, flujos de trabajo, artefactos, y documentos.

El ciclo de vida del software está compuesto de ciclos. Cada ciclo trabaja en una nueva generación de un producto. El RUP divide un ciclo de desarrolló en 4 fases consecutivas como se muestra en la figura 6:

- La fase Inicial.
- La fase de Elaboración.
- La fase de Construcción.
- La fase de Transición

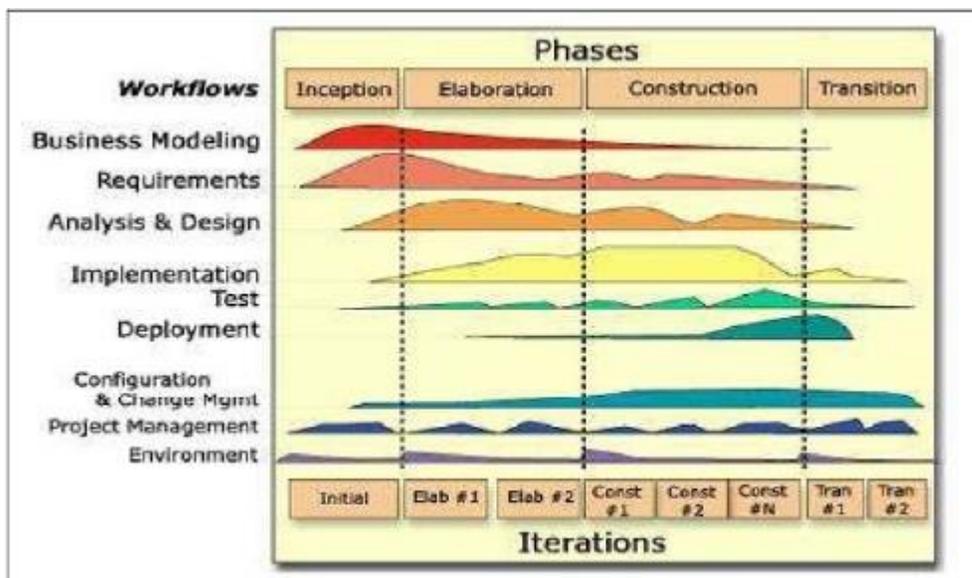


Fig. 6 Esquema de las fases del Proceso unificado

Cada fase es concluida con una meta bien definida (milestone) – la cual es un punto en el tiempo y se centra en las decisiones críticas que se deben tomar y qué metas clave deben ser alcanzadas.

La Fase de Inicio.- Durante la fase de inicio se establece un caso de negocios del sistema y se delimita el alcance del proyecto. Para completar esto, se debe de identificar todas las entidades externas con las cuales el sistema interactuara (actores) y se debe definir la naturaleza de estas interacciones a un alto nivel. El caso de negocios incluye un criterio de éxito, valoración de riesgos, estimado de recursos necesarios y un plan de la fase mostrando las fechas principales de las metas.

La Fase de Elaboración.- El propósito de la fase de elaboración es analizar el dominio del problema, establecer una sólida arquitectura, desarrollar el plan del proyecto y eliminar los elementos de mayor riesgo del proyecto. Las decisiones sobre arquitectura deben ser hechas con un entendimiento de todo el sistema, sus alcances, su función principal y los requerimientos no funcionales, así como los requerimientos de desempeño.

Es fácil entender que la fase de elaboración es la fase más crítica de las 4 fases. Al término de esta fase la “ingeniería dura” es considerada completa y el proyecto experimenta su más importante día en los cálculos; la decisión sobre continuar o no con las fases de construcción o transición. Mientras el proceso debe siempre acomodarse a los cambios, las actividades de la fase de elaboración se aseguran que la arquitectura, los requerimientos y el plan sean lo suficientemente estables, y el riesgo sea lo suficientemente mitigado, así que se puede predecir el costo y el calendario para completar el desarrollo. Conceptualmente, este nivel de fidelidad corresponde al nivel necesario para que una organización contemple costo de la fase.

En la fase de elaboración un prototipo ejecutable de la arquitectura es construido en una o más iteraciones, dependiendo del alcance, el tamaño, el riesgo y la novedad del proyecto.

Este esfuerzo por lo menos deberá manejar los usos críticos identificados en la fase de inicio, la cual típicamente expone la mayoría de los riesgos del proyecto. Mientras un prototipo evolucionado de un componente de producción de calidad es la meta, este no lo excluye del desarrollo de uno o más prototipos exploratorios, a través de prototipos que mitiguen los riesgos específicos como los de diseño, requerimientos intercambiados, estudios de factibilidad de componentes, o la demostración a los inversionistas, clientes y usuarios finales.

Al final de la fase de elaboración se encuentra la segunda meta importante. La meta de arquitectura del ciclo de vida. En este punto se examinan a detalle los objetivos y alcance del sistema, la selección de la arquitectura y la resolución de los riesgos mayores.

La Fase de Construcción.- Durante la fase de construcción, todos los componentes restantes y las características de las aplicaciones son desarrollados e integrados al producto y todas las características son ampliamente probadas. La fase de construcción es en un sentido un proceso de manufactura, donde el énfasis está colocado en la administración de recursos y el control de las operaciones para optimizar costos, calendarios y calidad. En este sentido, la administración mental experimenta una transición del desarrollo de la propiedad intelectual durante la fase de inicio y elaboración, al desarrollo de un producto destacable durante las fases de construcción y transición.

Al término de la fase de construcción se encuentra la tercera mayor meta del proyecto (la meta de capacidad operacional inicial) en este punto, se decidirá si el software, el sitio y los usuarios están listos para ser operacionales sin exponer al proyecto a riesgos mayores.

Esta liberación es frecuentemente llamada versión "Beta".

La Fase de Transición.- El propósito de la fase de transición es la transición del producto de software a la comunidad de usuarios. Una vez que el producto ha sido dado al usuario final, usualmente se requiere del desarrollo de nuevos entregables, corrigiendo algunos problemas o el terminar las características que fueron pospuestas.

La fase de transición es introducida cuando el producto es lo suficientemente maduro como para ser entregado al usuario final. Esto típicamente requiere que algunos subconjuntos del sistema hayan sido completados a un nivel aceptable de calidad y que la documentación del usuario esté disponible, así la transición del usuario proveerá de resultados positivos para todas las partes.

Los objetivos primarios de la fase de transición incluyen:

- Alcanzar el soporte de usuario
- Lograr la concurrencia de los accionistas que el desarrollo está completo y es consistente con el criterio de evaluación de la visión.
- Lograr que los lineamientos de producto final tanto en efectividad de tiempo y costo, como práctica.

Al final de la fase de transición se encuentra la cuarta meta en importancia del proyecto. La meta de liberación de producto. En este punto, se decide si los objetivos fueron cumplidos o si se debe de empezar otro ciclo de desarrollo, en algunos casos, estas metas pueden coincidir con el fin de la fase de inicio del siguiente ciclo.

2.5.3 Programación extrema (eXtreme Programming- XP)

Se puede considerar la programación extrema como la adopción de las mejores metodologías de desarrollo de acuerdo a lo que se pretende llevar a cabo con el proyecto y aplicarlo de manera dinámica durante el ciclo de vida del software.

Este método es típicamente atribuido a Kent Beck, Ron Jeffries y Ward Cunningham

XP intenta minimizar el riesgo de fallo del proceso por medio de la disposición permanente de un representante competente del cliente a disposición del equipo de desarrollo. Este representante debería estar en condiciones de contestar rápida y correctamente a cualquier pregunta del equipo de desarrollo de forma que no se retrase la tomar de decisiones, de ahí lo de competente.

XP define UserStories como base del software a desarrollar. Estas historias las escribe el cliente y describen escenarios sobre el funcionamiento del software, que no solo se limitan a la GUI si no también pueden describir el modelo, dominio, etc. A partir de las UserStories y de la arquitectura perseguida se crea un plan de releases entre el equipo de desarrollo y el cliente.

Junto a los UserStories están los escenarios de pruebas que describen el escenario contra el que se comprueba la realización de las UserStories. UserStories y casos de pruebas son la base sobre la que se asienta el trabajo del desarrollador.

La programación extrema proporciona una serie de ventajas a la hora de afrontar ciertos proyectos, que deben ser tenidas muy en cuenta, porque el uso de la XP en ciertos contextos puede ahorrarnos mucho tiempo y muchos recursos.

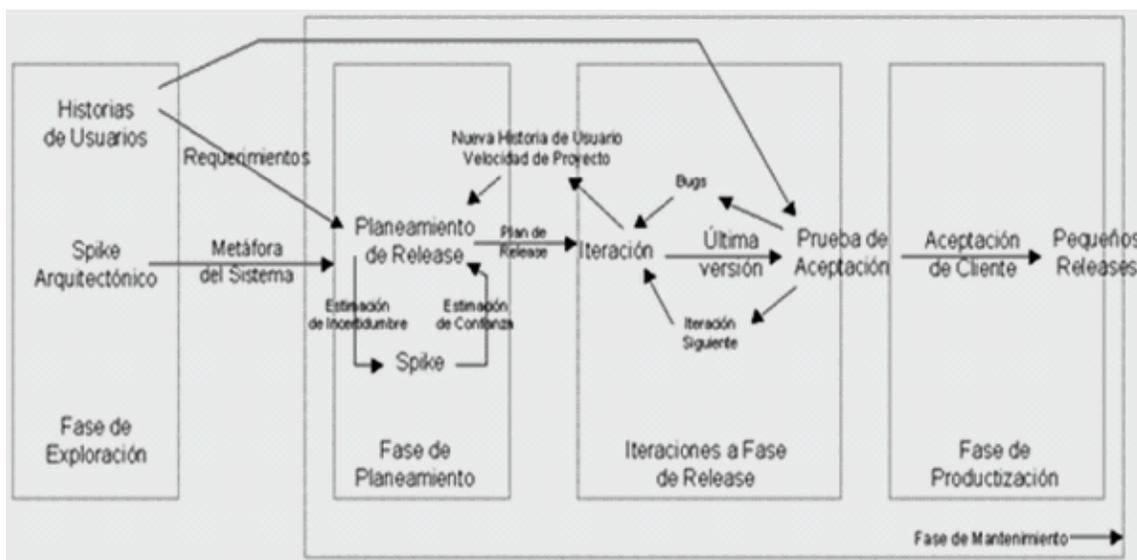


Fig. 7 Ciclo de vida eXtreme Programming

Los principios básicos de la programación extrema son: simplicidad, comunicación, retroalimentación y valor.

Simplicidad: la simplicidad es la base de la programación extrema. Se simplifica el diseño para agilizar el desarrollo y facilitar el mantenimiento. Un diseño complejo del código junto a sucesivas modificaciones por parte de diferentes desarrolladores hace que la complejidad aumente exponencialmente. Para mantener la simplicidad es necesaria la refactorización del código, ésta es la manera de mantener el código simple a medida que crece. También se aplica la simplicidad en la documentación, de esta manera el código debe comentarse en su justa medida, intentando eso sí que el código esté autodocumentado. Para ello se deben elegir adecuadamente los nombres de las variables, métodos y clases. Los nombres largos no disminuyen la eficiencia del código ni el tiempo de desarrollo gracias a las herramientas de autocompletado y refactorización que existen actualmente. Aplicando la simplicidad junto con la autoría colectiva del código y la programación por parejas se asegura que mientras más grande se haga el proyecto, todo el equipo conocerá más y mejor el sistema completo.

Comunicación: la comunicación se realiza de diferentes formas. Para los programadores el código comunica mejor mientras más simple sea. Si el código es complejo hay que esforzarse para hacerlo inteligible. El código autodocumentado es más fiable que los comentarios ya que éstos últimos pronto quedan desfasados con el código a medida que es modificado. Debe comentarse sólo aquello que no va a variar, por ejemplo, el objetivo de una clase o la funcionalidad de un método. Las pruebas unitarias son otra forma de comunicación ya que describen el diseño de las clases y los métodos al mostrar ejemplos concretos de cómo utilizar su funcionalidad. Los programadores se comunican constantemente gracias a la programación por parejas. El cliente decide qué características tienen prioridad y siempre debe estar disponible para solucionar dudas.

Retroalimentación (feedback): al estar el cliente integrado en el proyecto, su opinión sobre el estado del proyecto se conoce en tiempo real. Al realizarse ciclos muy cortos tras los cuales se muestran resultados, se minimiza el tener que rehacer partes que no cumplen con los requisitos y ayuda a los programadores a centrarse en los que es más importante. Considérense los problemas que derivan de tener ciclos muy largos. Meses de trabajo pueden tirarse por la borda debido a cambios en los criterios del cliente o malentendidos por parte del equipo de desarrollo. El código también es una fuente de retroalimentación gracias a las herramientas de desarrollo. Por ejemplo, las pruebas unitarias informan sobre el estado de salud del código. Ejecutar las pruebas unitarias frecuentemente permite descubrir fallos debidos a cambios recientes en el código.

Coraje: se requiere coraje para implementar las características que el cliente quiere ahora sin caer en la tentación de optar por un enfoque más flexible que permite futuras modificaciones. No se debe emprender el desarrollo de grandes marcos de trabajo mientras el cliente espera. La forma de construir marcos de trabajo es mediante la refactorización del código en sucesivas aproximaciones.

Ventajas

Evidentemente, para que algo esté siendo tomado tan en cuenta como la XP, debe ofrecer una serie de ventajas a la hora de ponerlo en práctica que haga que el esfuerzo de entender y aplicar sus prácticas, sea insignificante con respecto a los beneficios obtenidos.

- Se consiguen productos usables con mayor rapidez.
- El proceso de integración es continuo, por lo que el esfuerzo final para la integración es nulo. Se consigue integrar todo el trabajo con mucha mayor facilidad.
- Se atienden las necesidades del usuario con mayor exactitud. Esto se consigue gracias a las continuas versiones que se ofrecen al usuario.
- Se consiguen productos más fiables y robustos contra los fallos gracias al diseño de los test de forma previa a la codificación.
- Obtenemos código más simple y más fácil de entender, reduciendo el número de errores.
- Gracias a la filosofía del “pair programming” (programación en parejas), se consigue que los desarrolladores apliquen las buenas prácticas que se les ofrecen con la XP.
- Gracias al “refactoring” es más fácil el modificar los requerimientos del usuario.
- Conseguimos tener un equipo de desarrollo más contento y motivado. Las razones son, por un lado el que la XP no permite excesos de trabajo (se debe trabajar 40 horas a la semana), y por otro la comunicación entre los miembros del equipo que consigue una mayor integración entre ellos.

Capítulo 3

ANÁLISIS DEL PROBLEMA

3.1 Descripción de la situación actual

La Coordinación de Atención Diferenciada para Alumnos COPADI ha venido impartiendo cursos desde hace ya años. Tales cursos permiten mejorar el desempeño académico de los estudiantes. Pero esto tiene limitantes, los cuales son:

- Primeramente la información se difunde mediante carteles puestos en mamparas y fuera de las oficinas de esta coordinación.
- Segundo, el registro de alumnos se realiza en las oficinas de COPADI en horarios de oficina y de manera manual, es decir, se cuenta con una lista por curso para que se presente el alumno y se registre en el formato impreso.
- Y finalmente la información de cada curso es capturada en archivos digitales para su posterior generación de listas de asistencias y realización de constancias de participación.

Este proceso puede y genera inconsistencias de datos ya sea por no distinguirse la información escrita por cada alumno o al momento de capturar dicha información en los archivos digitales.

Otro de los beneficios con que cuenta dicha coordinación es la posesión de una biblioteca para consulta y préstamo de libros, destinada a los alumnos del Programa de Alto Rendimiento Académico (PARA), en la cual el préstamo y devolución se siguen realizando de manera manual y en hojas con formatos impresos.

La carencia de un sistema que realice las actividades propias de una biblioteca limita un servicio de calidad tanto a alumnos del PARA de la Facultad, por lo que se pretende sistematizar el uso de un sistema informático que permita llevar el control bibliotecario.

3.2. Determinación de requerimientos del sistema

Analizando de manera exhaustiva la situación descrita en el subcapítulo anterior y ante la situación actual de COPADI, se generará la siguiente lista de requerimientos para satisfacer en su totalidad las necesidades de esta área de la coordinación.

De manera general tenemos:

- Para que el sistema de inscripción pueda trabajar de manera global se requiere de un espacio en un servidor al que se pueda acceder desde cualquier ubicación dentro y/o fuera de la Coordinación.
- Para el sistema bibliotecario bastará con contar con un servidor en el cual se alojara dicho sistema
- Se requiere de un servidor con los mejores recursos que permita una disponibilidad del 100% por la gran cantidad de usuarios que tendrán acceso al sistema web.
- Se requiere de una página Web en la cual podremos visualizar tanto el sistema de cursos intersemestrales como el sistema de la biblioteca COPADI.

De manera muy específica para cada caso se pretende:

La página web de los cursos Intersemestrales debe contener:

- Objetivo
- Inscripción
- Programa de cursos
- Asignación de salones
- Requisitos

Para el caso de los Cursos Intersemestrales tenemos:

- En el sistema de cómputo podrá visualizarse desde cualquier computadora con acceso a internet siendo flexible para poder inscribirse. Proporcionando la siguiente información al solicitante de los cursos:
 - Nombre del curso
 - El nombre del expositor que impartirá el curso
 - Objetivo de cada curso
 - Antecedentes del curso
 - Comentarios del curso
 - Fecha de inicio, fecha de fin del curso.
 - El horario del curso
 - El salón en el que se impartirá el curso.

- El alumno podrá:
 - Observar y seleccionar el curso de su preferencia.
 - Observar el número de lugares disponibles.
 - Seleccionar solo un curso del área físico-matemático.
 - Seleccionar varios cursos del área de humanidades.
 - Registrarse directamente por medio de la página web.
 - Los datos ingresados al momento de la inscripción serán responsabilidad del alumno.
 - Estudiantes de otras facultades o externos a la universidad podrán inscribirse.

- El administrador podrá:
 - Actualizar la información de los cursos.
 - Visualizar la lista de alumnos inscritos en cada curso.
 - Ingresar a la base de datos para la extracción, eliminación o modificación de los datos contenidos en dicha base.

- El sistema:
 - Se alojará dentro de la página web principal de COPADI.
 - Contendrá la información de cada curso.
 - Generará reportes con listas de los alumnos inscritos en cada curso.
 - Guarda la información proporcionada por el alumno:
 - Nombre(s)
 - Apellido paterno
 - Apellido materno
 - No. Cuenta (si es alumno de la UNAM)
 - Teléfono
 - Correo electrónico
 - Nombre del curso al que se inscribió
 - Generará comprobante de inscripción por cada curso inscrito.

Para el caso de la biblioteca tenemos:

Bibliotecario:

- Se designará a un usuario que realizará las actividades de bibliotecario.
- Podrá acceder a los módulos de usuario.
- Tendrá acceso al material bibliográfico.
- Realizará los préstamos de ejemplares.
- Registrará la devolución de los libros.
- Asignará las multas pertinentes según el caso.

- Podrá generar reportes y realizar el mantenimiento del material bibliográfico.

- El alumno PARA
 - Podrá consultar el ejemplar dentro de la biblioteca.
 - Podrá pedir el préstamo del libro para consulta en otro lugar fuera de la biblioteca.
 - Al momento del préstamo los datos del alumno serán registrados en el sistema.
 - Al momento de la devolución el sistema lo registrará nuevamente.

- El sistema:
 - El sistema se alojará dentro del apartado PARA de la página principal de COPADI
 - Se podrá realizar la búsqueda del libro por: título del ejemplar, nombre del autor, editorial, país, año de edición, ISBN o género.
 - El resultado mostrará el título del ejemplar, nombre del autor y la editorial.
 - Sera operado por el bibliotecario y realizará las actividades que se requieran.

3.3 Herramientas a utilizar

Para realizar ambos sistemas se decidió hacerlo en el lenguaje de programación Java, esto porque dicho lenguaje es multiplataforma (característica de funcionar de forma similar en distintas plataformas, distintos sistemas operativos por ejemplo), de software libre desde la instalación de sus librerías la instalación de su máquina virtual su entorno de desarrollo o la facilidad de conectarse con bases de datos, desarrollar aplicación de escritorio, aplicaciones web, entre otros; asimismo cuenta con opciones y herramientas más fáciles, más vistosas, permite exportar documentos en PDF, Word, Excel, etc.

Otras de las ventajas que ofrece Java es la gestión de memoria esto ya que existe un proceso independiente denominado "Garbage Collector" que se encarga de liberar automáticamente toda la memoria que ya no se utiliza, de manera que la liberación de memoria se hace de manera transparente al programador. Las librerías estándar es una de las características que más potencia aporta al

lenguaje Java es que viene acompañado de una serie de librerías estándar para realizar multitud de operaciones comunes a la hora de programar. Para los Frameworks de desarrollo web existen multitud de frameworks y librerías que ofrecen capas para el desarrollo de aplicaciones Web de manera más ágil, organizada y escalable. El lenguaje Java es relativamente fácil de aprender comparado con otros. Si se conoce algún lenguaje de programación como Pascal o C, la transición a Java no es muy complicada, ya que la sintaxis es muy parecida.

Lo anterior, entre otras cosas, fue parte de lo que se consideró para utilizar Java.

La utilización de JavaScript fue también implementado en nuestros sistemas, esto porque es un lenguaje de programación que se ejecuta en el navegador web del visitante (o también llamado usuario). JavaScript permite crear funcionalidades específicas en las páginas webs como lo es en la mejora de interfaz de usuario y páginas web dinámicas.

Jquery es una librería de JavaScript que permite simplificar la manera de interactuar con HTML, manejar eventos, desarrollar animaciones y agregar interacciones con páginas webs, se utilizó ya que es software libre y de código abierto lo que permite su uso en proyectos libres y/o privados, asimismo fue tomado en cuenta ya que ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio. Permite manipulación de hojas de estilo CSS, permite desarrollar efectos y animaciones, soporta extensiones y es compatible con los navegadores Mozilla Firefox 2.0+, Internet Explorer 6+, Safari 3+, Opera 10.6+ y Google Chrome 8+. Finalmente podemos decir que existe bastante documentación de esta librería por lo que no sería complicado encontrar información y la utilización sería menos complicada.

Para la parte de las bases de datos se utilizó PostgreSQL, este es un manejador que tiene numerosas ventajas por las cuales se decidió utilizarlo, alguna de ellas son las siguientes, su código fuente es libre y de alta calidad, podemos realizar prácticamente lo que queramos con el producto sin restricción alguna, los requerimientos de administración y mantenimiento son relativamente bajos con respecto del resto de bases de datos comerciales, tiene Fiabilidad y estabilidad respaldada por años de trabajo, tiene un rendimiento excelente, está diseñada para entornos con altos volúmenes de tráfico/transacciones, es multiplataforma y finalmente tenemos que cuenta con herramientas gráficas y de línea de comandos para diseñar nuestras bases de datos y administrarlas

Ahora bien, la implementación de Tomcat fue realizado ya que este es un servidor web que soporta la tecnología J2EE de JAVA, la primera ventaja que ofrece es que es gratuito y es muy sencillo de configurar para empezar a funcionar, y dispone de abundante documentación. Otro punto a favor que tiene y por la cual se utilizo es que es un servidor web autónomo utilizado en entornos con alto nivel

de tráfico y cuenta con alta disponibilidad. Finalmente es software y de código abierto.

Capítulo 4

DISEÑO DEL SISTEMA CURSOS INTERSEMESTRALES Y BIBLIOTECA DIGITAL

4.1 Desarrollo del sistema

Como ya se mencionó anteriormente, no se cuenta con un sistema automatizado en el cual los estudiantes puedan inscribirse a los cursos intersemestrales impartidos por la COPADI, ni tampoco cuenta con un sistema bibliotecario administrado por dicha Coordinación.

Solamente se cuenta con un apartado dentro del portal COPADI que hace referencia a los cursos intersemestrales y en la cual se da una pequeña reseña de lo que se pretende al impartir dichos cursos, y hace hincapié a que los alumnos se dirijan a la planta alta del auditorio Sotero Prieto de la División de Ciencias Básicas para realizar su registro a los cursos de su interés.



Imagen 1. Apartado actual del programa de cursos intersemestrales

Actualmente COPADI cuenta con un servidor ya configurado con el sistema operativo Fedora 11, motivo por el cual no será necesario adquirir otro servidor, y en el cual será alojado el sistema de cómputo de Cursos intersemestrales y el sistema de cómputo de la Biblioteca Digital.

- Un servidor de base de datos (Tomcat).
- Una base de datos
- Conocimiento en los lenguajes de programación HTML y JSP con Bases de Datos, para páginas web.
- Servidor servlets (Tomcat)

Y registros ejemplo de bitácoras pasadas para hacer pruebas del sistema.

Las siguientes líneas de código son importantes para nuestra programación:

```
<%@ page import = "java.util.*" %>  
<%@ page import="BeansTutorFI.*" %>  
<%@page contentType="text/html" pageEncoding="UTF-8"%>  
<jsp:useBean id="contador" class="BeansTutorFI.Contador" scope="application"/>
```

Dentro de nuestro archivo HTML integramos las líneas anteriores escritas en JAVA que nos permiten llamar a otros archivos que serán usados dentro del documento para su ejecución

4.2. Diseño de la página web Cursos Intersemestrales

El esquema siguiente nos muestra la organización del portal COPADI, en la cual estará alojado nuestro aplicativo web de Cursos Intersemestrales y la Biblioteca.

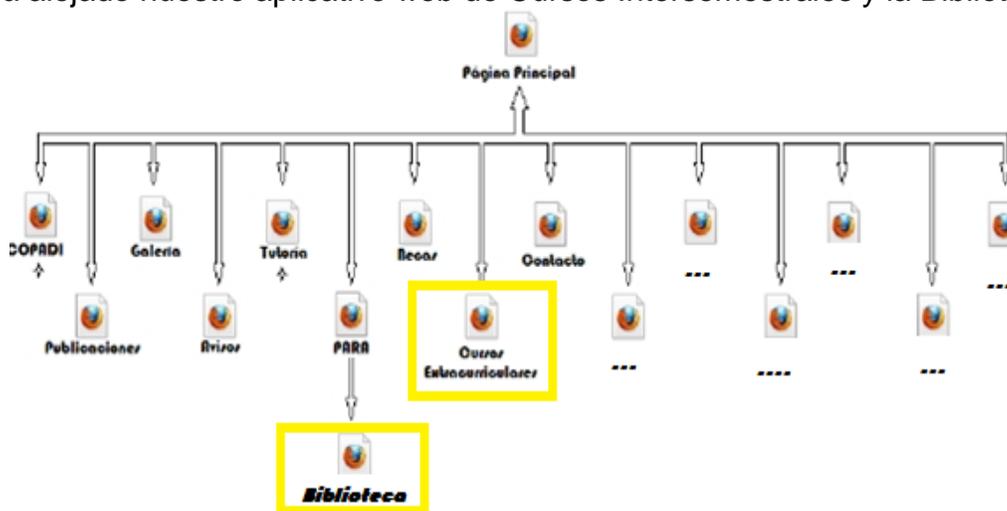


Imagen 2. Ubicación de Cursos Inter-semestrales dentro del organigrama COPADI

Es sabido que DGTIC establece unos lineamientos para la realización y publicación de páginas web institucionales (<http://recursosweb.unam.mx/recursos-web/lineamientos-unam/>), por lo que para la realización de este proyecto se respetó lo establecido por dicha autoridad. El portal COPADI ya cuenta con ciertos lineamientos establecidos y de igual manera se respetaran ya que el Sistema de cursos Intersemestrales y la Biblioteca Digital serán montados dentro del portal antes mencionado.

Primeramente utilizamos hojas de estilo en cascada CSS que nos permitirá realizar tantas modificaciones sean necesarias de una manera rápida y sencilla.

Tanto el encabezado como el pie de página ya fueron establecidos anteriormente por el portal COPADI y se tomó como base para nuestro sistema, dentro de esto albergaremos el cuerpo de nuestra página.



Imagen 1. Encabezado establecido por el portal COPADI



Imagen 3. Pie de página del portal COPADI

La base del portal que se estableció con anterioridad fue definida de la siguiente manera:

```
body {
background-color: transparent;
background-image: url(../Logos/fondo.gif);
background-position: top;
background-repeat: repeat-x;
height: 100%;
text-align: center;
}
```

De manera general está definido el color de fondo, la imagen a utilizar como fondo (degradado de gris), tipo de letra y alineación.

Con lo anterior partiremos para la generación de nuestra hoja de estilo que contendrá nuestro sistema, el cual estará compuesto por el bloque de menús, listado de cursos, descripción del curso, datos de contacto, módulo de registro y pie de página.

Comenzaremos con lo siguiente que son clases que definirán la manera en que quedará nuestra hoja de estilo:

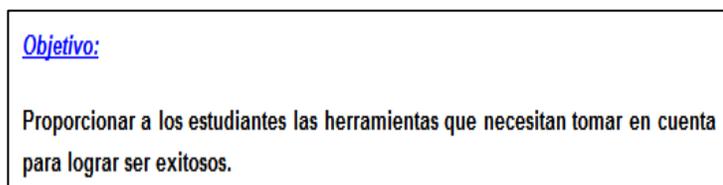
```
.tituloCurso {  
font-weight: bold;  
font-size: 12pt;  
line-height: 16pt;  
font-family: helvetica;  
font-variant: normal;  
font-style: normal;  
text-align: center;  
}
```



```
.expositor {  
font-size: 10pt;  
font-family: helvetica;  
font-variant: normal;  
font-style: normal;  
text-align: left;  
}
```



```
.rotulo {  
font-size: 10pt;  
line-height: 11pt;  
font-family: helvetica;  
font-variant: normal;  
font-style: oblique;  
text-align: left;  
text-decoration: underline;  
color: #0000FF;  
}
```



```
.contenidoCursos {
font-size: 10pt;
font-family: helvetica;
font-variant: normal;
font-style: normal;
text-align: justify;
}
```

<u>Antecedentes:</u>	
<u>Comentarios:</u>	
El 11, 12, 13 y 14 de junio	
<u>Fecha inicio:</u> 2012-06-11	<u>Horario:</u> 10:00 - 13:00 hrs.
<u>Fecha fin:</u> 2012-06-14	

```
.campoInscripcion {
color: #000000;
font-weight: bold;
font-size: 11pt;
font-family: helvetica;
font-variant: normal;
font-style: normal;
text-align: left;
}
```

¡Inscríbete Ya !	
Número de cuenta:	<input type="text"/>
Nombre:	<input type="text"/>
Paterno:	<input type="text"/>
Materno:	<input type="text"/>
Teléfono:	<input type="text"/>
Tel. cel:	<input type="text"/>
Correo electrónico	<input type="text"/>
Semestre	-Seleccione- ▼
Carrera:	<input type="text"/>

```
.nota {
font-size: 8pt;
line-height: 10pt;
font-family: helvetica;
font-variant: normal;
font-style: oblique;
text-align: center;
text-decoration: underline;
color: red;
}
```

Nota: En dado caso que el curso proporcione constancias, tus datos aparecerán tal cual los enviaste.

```
.informesCursos {
position: relative;
left: 586px;
top: 250px;
font-size: 10pt;
font-family: helvetica;
font-variant: normal;
font-style: normal;
}
```

Informes:
Lic. Ma. de la Paz González Anaya
Teléfono:
56 22 81 01 ext. 106

Ahora bien, para los efectos de nuestro menú o listado de cursos el cual se encontrará en la parte izquierda de nuestra hoja, se utilizó JavaScript con el objeto de darle un mejor efecto de deslizamiento y una mejor presentación; las librerías pueden ser descargadas de la página: <http://code.jquery.com>

Las librerías utilizadas para dicho efecto son:

- prototype.js
- scriptaculous.js
- os.js
- carousel.js
- application.js

Para integrar los diferentes archivos JS y CSS se utiliza un archivo general donde se hace referencia a cada código que es empleado en la página.

Para hacer la llamada a cada código o librería se realiza de la siguiente manera:

```
<script language="javascript" type="text/javascript" src="js/clientscript/prototype/prototype.js"></script>  
<script language="javascript" type="text/javascript" src="js/clientscript/prototype/scriptaculous.js?load=effects"></script>
```

Con lo que nos permitirá visualizarlo así:

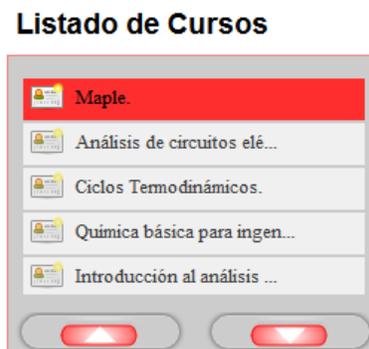


Imagen 4. Listado de cursos

Finalmente tenemos nuestra página que quedará conformada de la siguiente manera:



Imagen 5. Diseño de la página de Cursos Inter-semestrales

Con este diseño se procura mostrar un entorno amigable y de fácil utilización para que el usuario no tenga que abrir y navegar por otras tantas pestañas o páginas que en determinado caso haga tediosa la visualización y que disminuya el interés.

Se pretende que mientras el usuario observe la lista de cursos pueda ir leyendo una breve descripción del contenido del curso y en ese momento poder realizar su registro.

Al ser un aplicativo web se busca fomentar el uso de las tecnologías actuales y dejar a un lado las actividades manuales que permitirán reducir tiempo y reducir errores en la captura de datos.

4.2.1. Estructura de Cursos Inter-semestrales

Dado que este sistema está contenido dentro del portal COPADI, su diseño debe ser homogéneo para brindar una claridad en los procesos.

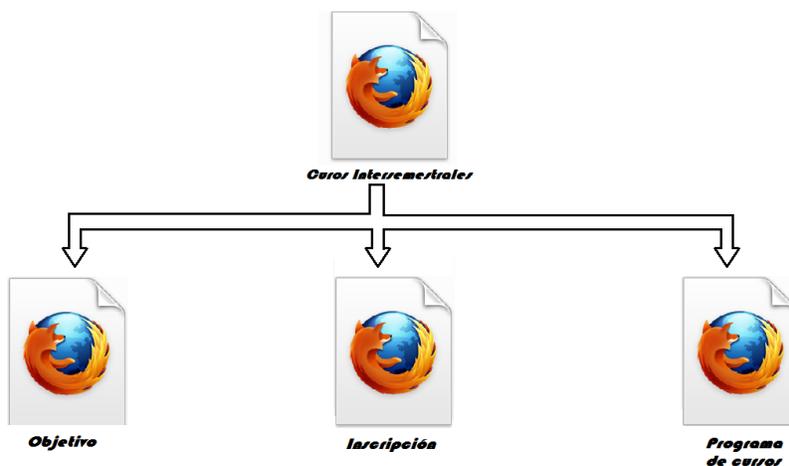


Imagen 6. Esquema de Cursos Intersemestrales

4.2.2. Objetivo del programa de cursos intersemestrales

Con el objetivo de impulsar el desarrollo humano integral de los estudiantes de la Facultad de Ingeniería a través de la Coordinación de Programas de Atención Diferenciada para Alumnos (COPADI) brinda en el intersemestre, diversos cursos con el fin de apoyar la superación personal y formación académica de los estudiantes, mediante la adquisición de conocimientos y habilidades indispensables, que favorecen el desempeño académico y profesional exitoso.

En cada intersemestre, el personal académico de la COPADI y los profesores de la Facultad, ofrecen cursos relacionados con el desarrollo humano y con el área disciplinar, a fin de que el estudiante tenga la oportunidad de fortalecer tanto sus habilidades psicosociales como reforzar sus conocimientos académicos.

En el área de desarrollo humano se brindan cursos que permiten a los alumnos, reconocer las habilidades con las que cuenta, como una fortaleza para su

interacción social, y favoreciendo aquellas habilidades requieren ser desarrolladas, para mejorar su respuesta ante los conflictos cotidianos.

En el área disciplinar se brindan cursos que fortalecen sus conocimientos en Computación, Física y Matemáticas.

A través de los cursos, se inciden en los diversos factores que influyen en el desempeño académico y que constituyen elementos fundamentales para la motivación y actitud positiva en su actividad como estudiante.

La difusión del programa de cursos intersemestrales, se realiza colocando carteles con los datos generales de los cursos, en mamparas y en los espacios destinados para la información para alumnos, en la página de la COPADI y por el circuito cerrado de TV.

4.2.3. Proceso de Inscripción

El estudiante ya sea que pertenezca a la comunicad universitaria inscrito en alguna de sus facultades o siendo estudiante externo a la UNAM, así como al público en general puede inscribirse a algún curso impartido por COPADI, respetando las reglas establecidas por la misma coordinación, reglas que permitirán evitar saturación en los grupos, llevar un control detallado de alumnos por clase, y organizar los cursos de mayor demanda para que toda la comunidad pueda acceder a ellos; dichas reglas son:

- Un estudiante puede inscribirse solamente a un curso del área computacional.
- Un estudiante puede inscribirse a uno o más cursos de área de humanidades.
- Un estudiante puede estar inscrito solo a un curso del área computacional y a la vez a uno o varios cursos de área de humanidades.

The screenshot shows the COPADI website interface. At the top, there is a header with the university logo, the text 'Facultad de Ingeniería COPADI', and the date 'Sábado, 10 de Noviembre de 2012'. Below the header is a navigation bar with links for 'Principal', 'COPADI', 'Publicaciones', 'Galería', and 'Avisos'. On the left, a sidebar lists various programs under the heading 'Programas:'. The main content area is titled 'Cursos Intersemestrales' and displays details for a course named 'El secreto del éxito del estudiante'. The course details include the instructor 'Mtra. Ana G. García y Colomé Dra. Fatima Mouttadi', the objective, and registration information. A registration form is provided with fields for account number, name, paternal name, maternal name, phone, cell phone, email, semester, and career. A 'Enviar' button is located below the form. To the right, there is a 'Listado de Cursos' section showing a list of courses and a login section with fields for 'Usuario:' and 'Contraseña:', and an 'Iniciar Sesión' button.

Imagen 7. Proceso de Inscripción

Los datos a ingresar por el alumno y que son requeridos por el sistema serán:

- Número de cuenta
- Nombre
- Paterno
- Materno
- Teléfono
- Teléfono celular
- Correo electrónico
- Semestre
- Carrera

Una vez ingresado los datos, estos serán insertados en la Base de Datos y el sistema generará un comprobante de inscripción que deberá imprimir el alumno como antecedente de su registro para posibles aclaraciones en caso de tener algún problema.

Para describir de manera general este proceso y tener una visión de cómo se realizan dichas inscripciones, a continuación se mostrará un diagrama de flujo:

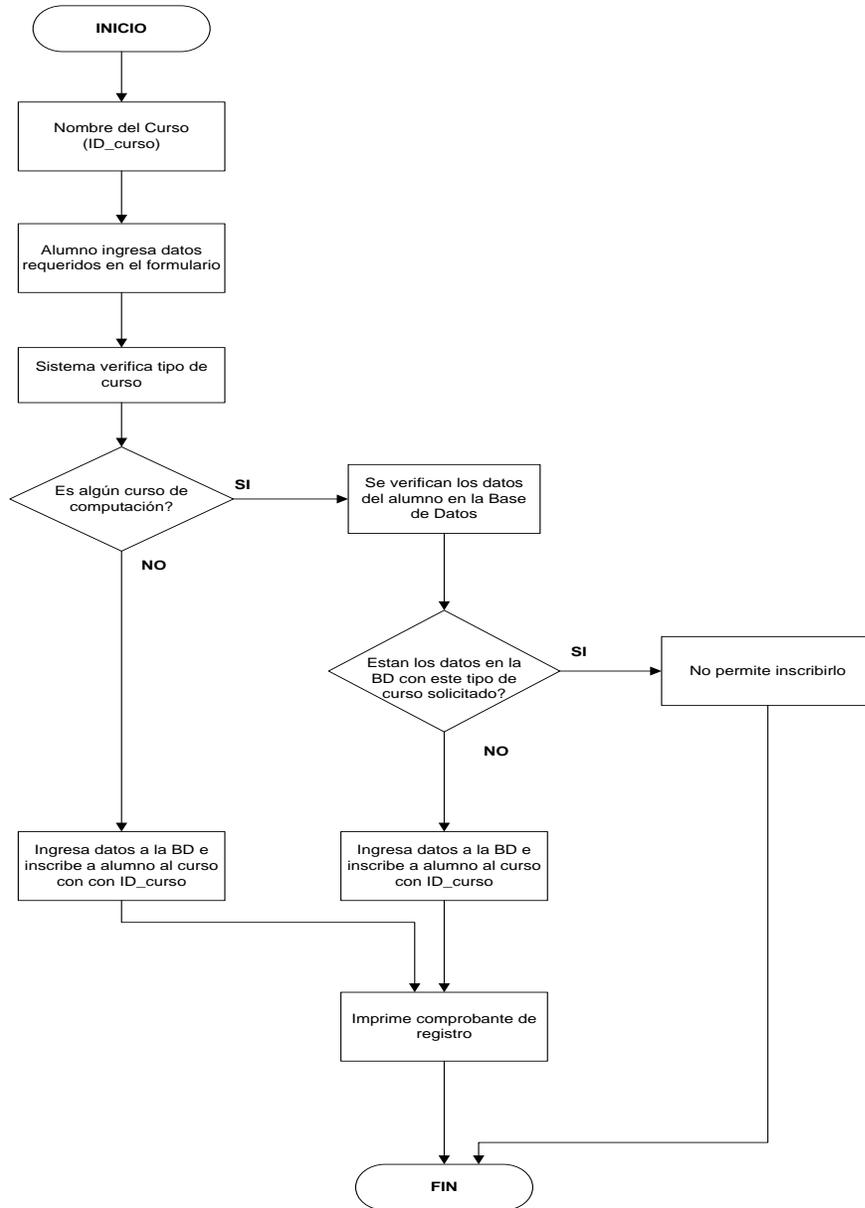


Imagen 8. Diagrama de flujo del proceso de registro a cursos

- En caso de que el alumno al elegir un curso de su agrado (que no sea del área computacional) y este se encuentre sin cupo, entrará a una lista de espera en la cual al inicio del curso deberá asistir para poder obtener un lugar esto en base al criterio del profesor en caso de que alumnos inscritos no asistan a la clase.

4.2.4. Reportes

En esta sección el administrador podrá revisar los reportes de alumnos inscritos a cada curso, este archivo podrá ser visualizado en Excel y dividido por hojas de cálculo, cada hoja de cálculo contendrá el nombre del curso y los registros de alumnos que decidieron inscribirse a dicho curso, además contará con el registro de los alumnos en la lista de espera.

The screenshot shows an Excel spreadsheet with the following content:

Reporte de Cursos Intersemestrales 2013-1

Introducción al análisis de Fourier.

Ing. Érik Castañeda De Isla Puga

Del 2013-01-07 al 2013-01-18 de 10:00:00 a 13:00:00

No.	Nombre	No. de Cu Correo	Tel. casa	Tel. celul	Semestre	Carrera	
1	Daphne C	309161754	Daph.037@	56029118	55275043	3	Ingeniería Civil
2	Luis Anton	412051722	luis.agmo		55148840	3	Ing. Civil
3	Salvador E	309266598	chavador5	56314827	55476457	3	Ing. electrica electronica
4	Jennifer Ru	309115348	Nena_rock	57017952	55390394	3	Ing. En Telecomunicaciones
5	EDUARDO	309110831	phoenix_@	26224533	55393967	3	Ingeniería en Computación
6	Gerzon G	309100144	ggomezb7	59862082	55641860	3	Ing. en telecomunicaciones
7	Luis Angel	309191887	luisangel.y	26153848	55426891	3	Ing. en telecomunicaciones
8	Mariana G	412003044	marianag	56711691	55414010	3	Telecomunicaciones
9	Adriana Br	412005639	adriana.br	24769351	55106898	3	Telecomunicaciones
10	Edmundo	98313026	plusultra77		55299232	4	Electrica Electronica
11	Vania Del	309010188	vania0793	55797643	55640523	3	Ingeniería Mecatrónica
12	JENNIFER	41106945	jennifer_92	55500829	55406230	5	geofísica
13	Daniel Phe	309716567	pliegodani	55933413	55371144	3	Ingeniería Mecánica
14	Mario Alon	309208673	laxemburg	54861313	55233538	3	Ing. Mecatrónica
15	luis arturo	309305947	yempo_rud	25946261		3	telecomunicaciones
16	Jessica Vil	309339717	villegas_91	58426617	55260988	3	Ingeniería Mecatrónica
17	José Luis	308304521	jlda92@hno	56652154	55181629	4	Telecomunicaciones
18	Luis Ferna	407062122	jlf_eza@g	58413279	55132008	7	Geomática
19	Yoás Saim	308270390	cygrallera	19974627	55158282	5	Ing. en computación
20	Luis Albert	308327542	luigi_avc92		55611282	5	Ingeniería en computación
21	Juan Ange	308193194	angelmata		55284475	5	Ingeniería en computación
22	Jorge Luis	309087968	kdc_707@	58491392	55487977	3	Ingeniería Mecatronica
23	José Rev	411072045	jr moc es		55143773	5	Ingeniería Civil

Imagen 9. Reporte de los Cursos Intersemestrales

The screenshot shows an Excel spreadsheet with the following data:

No.	Nombre
43	SERGIO HERNÁNDEZ SÁNCHEZ
44	OMAR ALEJANDRO HERNANDEZ SUESCUM
45	Eduardo Meza Medina
46	KEVIN DANIEL ABREGO SOLIS
47	Manuel Coral Castro
48	Laura Adriana Casillas Noriega
49	Julio Cesar Hernández Reyes
50	EDGAR ARANDA LUCIO
Lista de Espera	
1	PEDRO JESÚS NÁJERA CHÁVEZ
2	Jorge Chavez Brñseño
3	DANIEL ESQUIVEL TRUJILLO
4	Angelica Ponce Nava Araiza
5	Alberto Ricardo Osorio Cruz
6	Montiel Benitez Raúl
7	Alberto Islas Gatica
8	SAÚL ARTEMIO GARCÍA DUQUE
9	Miguel Barbosa Márquez
10	LINO MATLALCUATZI PATIÑO
11	Jorge Alejandro González Loyola
12	YESICA ITZEL ALVAREZ CAMPERO
13	Héctor Muñoz Rivera
14	Orestes De Brandon Barrera Pastrana
15	Lilia Estefanía Orozco Ramirez

Imagen 10. Lista de espera

4.2.5. Programa de cursos

Este apartado es donde se muestra un listado de los cursos existentes así como una descripción breve que se actualiza cada semestre, se muestra en formato pdf que podrá ser descargado y/o impreso por el interesado. Esta misma información es la que se difunde en las mamparas de toda la Facultad en formato de cartel con carácter informativo. La finalidad de ésto es que se tenga el documento para posteriores consultas y después realizar su registro en la parte correspondiente del aplicativo web.

CURSOS INTERSEMESTRALES COPADI
Programa e inscripciones en
<http://copadi.fi-c.unam.mx>



Curso	Expositor (es)	Fecha	Horario	Salón
Salud Integral.	Mtra. Ana G. García y Colomé	El 10, 11, 13 y 14 de diciembre	De 10 a 14	J102
Modelado Matemático.	Ing. Francisco Barrera García	Del 7 al 11 de enero	De 9 a 12	A-201
Tópicos de Cálculo Diferencial.	Sergio Carlos Crail Corzas	Del 7 al 11 de enero	De 10 a 12	A-202
Liderazgo de alto impacto.	Alejandro Carrasco Bustamante Lic. Leslie Vargas Natividad	Del 7 al 11 de enero	De 10 a 13	A-203
El secreto del éxito del estudiante.	Dra. Fatima Mourtadi Ing. Alejandro Rodríguez Rodríguez	Del 7 al 11 de enero	De 10 a 14	A-204
Análisis Transaccional y comunicación.	Lic. Javier Gómez Rodríguez Lic. Ana María Vieyra Ávila	Del 7 al 11 de enero	De 10 a 14	A-205
Algoritmos numéricos y programación en C.	M. A. Miguel Eduardo González Cárdenas Ing. Jesús Javier Cortés Rosas	Del 7 al 11 de enero	De 13 a 16	SJ203
Maple.	Ing. Ángel Leonardo Bañuelos Saucedo M.I. Nayelli Manzanarez Gómez	Del 7 al 11 de enero	De 16 a 19	J205B
Introducción a la electrodinámica relativista.	Raúl Puente Mancilla	Del 7 al 16 de enero	De 10 a 13	B-301
Introducción al análisis de Fourier.	Ing. Érik Castañeda De Isla Puga	Los días 7, 9, 11, 14, 16 y 18 de enero Lunes, miércoles y viernes	De 10 a 13	B-302
Introducción a la Física Moderna.	Fis. Edgar Raymundo López Téllez Fis. Salvador Enrique Villalobos Pérez	Los días 10, 11, 14, 15, 16, 17 y 18 de enero	De 16 a 19	Lab. Acústica y óptica
Temas de Pelcoanálisis para Ingenieros.	Mtra. Margarita Puebla Cadena	Los días 14, 15, 16 y 17 de enero	De 10 a 13	B-303
Análisis de circuitos eléctricos mediante simuladores.	Mtro. Rigel Gámez Leal	Del 14 al 18 de enero	De 9 a 13	Lab. Acústica y óptica
Ciclos Termodinámicos.	Ing. Genaro Muñoz Hernández	Del 14 al 18 de enero	De 10 a 13	B-304
Hablar en público y liderazgo para universitarios.	Alejandro Carrasco Bustamante Lic. Leslie Vargas Natividad	Del 14 al 18 de enero	De 10 a 13	A-201
Algoritmos numéricos y programación en C.	M. A. Miguel Eduardo González Cárdenas Ing. Jesús Javier Cortés Rosas	Del 14 al 18 de enero	De 10 a 13	A-201
Metodología para la elaboración de la tesis.	Lic. Javier Gómez Rodríguez Ing. Guillermina Pérez Durán	Del 14 al 18 de enero	De 10 a 14	A-202
Temas de Pedagogía para Ingenieros.	Mtra. Ana G. García y Colomé	Del 14 al 18 de enero	De 11 a 15	A-203
Manejo del estrés.	M. A. Miguel Eduardo González Cárdenas M. A. María de Lourdes Campos Luna	Del 14 al 18 de enero	De 16 a 19	A-204
Electrónica básica con Arduino.	Ing. Adriana Yoloxóchil Jiménez Rodríguez Jorge Luis Aguirre Serralde	Del 21 al 25 de enero	De 9 a 14	J203
Los secretos del dinero, (el arte de administrar tus ingresos).	Alejandro Carrasco Bustamante Lic. Karla M. Chavira Romero Oscar Carrasco Bustamante	Del 21 al 25 de enero	De 10 a 13	A-201
Estrategias para mejorar mis hábitos de estudio y rendimiento académico.	Mtra. Claudia Loreto Miranda Lic. Griselda Núñez Núñez	Del 21 a 25 de enero	De 10 a 14	A-202
Taller de creatividad.	Lic. Ma. De la Paz González Anaya Ing. Genaro Muñoz Hernández	Del 21 al 25 de enero	De 10:30 a 12:30	A-203
Química básica para Ingenieros.	Quím. Antonia del Carmen Pérez León	Del 21 al 25 de enero	De 16 a 18	Lab. Química

Revisa los cursos, todavía algunos tienen vacantes



Imagen 11. Programa de cursos

4.2.6. Conformación de la base de datos

La conformación en la base de datos del sistema estará dada de la siguiente manera:

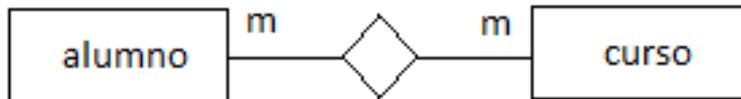
Tabla Alumno:

CAMPO	TIPO DATO	Descripción
Id_alumno	Integer	Identificador de alumno
Id_curso	Integer	Identificador del curso al cual está inscrito el estudiante
numcuenta	Integer	Número de cuenta del estudiante
nombre	varchar(50)	Nombre del estudiante
paterno	varchar(50)	Apellido paterno del estudiante
materno	varchar(50)	Apellido materno del estudiante
telefono	varchar(12)	Teléfono del estudiante
celular	varchar(13)	Numero de celular del estudiante
correoe	varchar(60)	Correo electrónico del estudiante
semestre	integer	Semestre en el cual está inscrito el estudiante
carrera	varchar(40)	Carrera que cursa el estudiante
espera	char(1)	Tipo de inscripción del estudiante, 1 si está en lista de espera, 0 si está inscrito en lista normal

Tabla Curso:

CAMPO	TIPO DATO	Descripción
Id_curso	Integer	Identificador del curso
nombre	varchar(300)	Nombre del curso
objetivo	varchar(600)	Objetivo del curso
temario	varchar(1000)	Temario del curso
expositor	varchar(200)	Expositor del curso
asistentes	integer	Número de asistentes al curso
vacantes	integer	Vacantes que tiene el curso
vacantesespera	integer	Vacantes en espera que tiene el curso
fechainicio	date	Fecha inicial del curso
fechafin	date	Fecha de cierre del curso
horainicio	time	Hora de inicio del curso
horafin	time	Hora de término del curso
duracion	varchar(20)	Duración del curso
recursos	varchar(300)	Recursos que requiere el curso
comentario	varchar(300)	Comentarios para los estudiantes tomen en cuenta
semestre	varchar(6)	Semestre en que se imparte el curso
activo	char(1)	Estado del curso, 1 si está activo, 0 si se dio

		de baja
salon	varchar(50)	Salón donde se imparte el curso
tipo	integer	Tipo de curso
vacantesinicial	integer	Vacantes inicial del curso
vacantesesperainicial	integer	Vacantes espera inicial del curso



Relación muchos-muchos. la entidad alumno hereda la llave primaria de la entidad curso.

Modelo Lógico: Diagrama entidad-relación

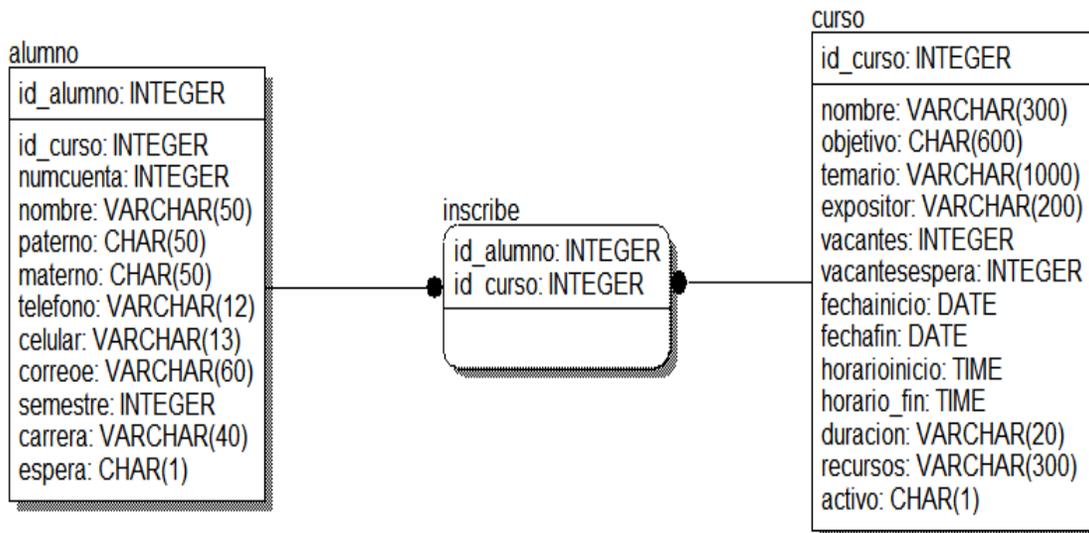


Imagen 12. Esquema de la base de datos, modelado en Erwin

El sistema cuenta con 2 beans:

- BeanAlumno
 - o Donde se encapsulan todos los atributos de esta entidad
- BeanCurso
 - o Donde se encapsulan todos los atributos de esta entidad y donde se definen las funciones para inscribir a un estudiante a un curso. Así mismo se cuenta con las funciones que actualizan los contadores de las vacantes y los reportes de estado de los cursos.

Los archivos con los que cuenta el sistema para realizar las diferentes tareas son las siguientes:

1. inscripcionCursosIntersemestrales.jsp
 - a. que es donde se presenta la lista de los cursos que se ofrecen, así como los formularios para realizar la inscripción
2. inscripcionCursoIntersemestralValida.jsp
 - a. que es donde se validan los datos y se hace el registro de la inscripción así como dar la opción de imprimir el comprobante

4.3. Diseño del sistema de Biblioteca Digital

4.3.1. Menú Inicial

Para el diseño de este aplicativo web de la biblioteca digital se utilizaron hojas de estilo en cascada CSS. Algunas de las hojas de estilo más importantes utilizados fueron:

- **body.css**. Hoja de estilo que en la cual definiremos las dimensiones de nuestra página y que aplicará a todo el contenido.
- **menus.css**. Nos permite crear un menú de navegación dentro de nuestro aplicativo
- **template.css**. Es el estilo en cascada principal que se encuentra dentro de la carpeta 'css'

- **modules.css.** Es un estilo simple que contiene menú con Links que nos permitirá enlazarlos a otras hojas de estilos.

A continuación se mostrarán los menús al que solo podrán acceder los estudiantes para poder realizar la consulta de algún ejemplar dentro de la biblioteca. Cabe mencionar que, para tener acceso a la búsqueda y consulta de algún ejemplar, el alumno deberá ser alumno de alto rendimiento perteneciente al Programa de Alto Rendimiento Académico PARA.

Las siguientes líneas de código corresponden al menú inicial el cual lleva por nombre "index.jsp".

```
<!-- Inicio Bloque de Estilo de Menu -->
<div id="headerbar">
  <div class="floatbox ie_fix_floats">

  </div>
</div>

<div id="menubar">
  <div class="menubar-l"></div>
  <div class="menubar-r"></div>
  <div class="menubar-m"></div>
</div>
<!-- Fin Bloque de Estilo de Menu -->
<!-- Inicio Contenido de Menu -->
<div id="menu">
  <ul class="menu">
    <li class="level1 item2 first active current"><a href="index.jsp" class="level1 item2 first active current"><span>Inicio</span></a></li>
    <li class="level1 item3"><span class="separator level1 item3"><span>Libro</span></span>
      <ul class="level2">
        <li class="level2 item1 first"><a href="consultaLibroInvitado.jsp" class="level2 item1 first" style="background-color: rgb(235, 236, 237); "><span>Consulta</span></a></li>
      </ul>
    </li>
  </ul>
</div>
<!-- Fin Contenido de Menu -->
<!-- Inicio Logo -->
<div id="logo">
  
</div>
<!-- Fin Logo -->
<!-- Fin de header -->
<!-- Inicio Bloque contenido -->

<div id="wrapper">
  <div class="wrapper-container-t">
    <div class="wrapper-tl"></div>
    <div class="wrapper-tr"></div>
```

Imagen 13. Archivo "index.jsp" menú inicial

Con lo anterior visualizaremos lo siguiente:

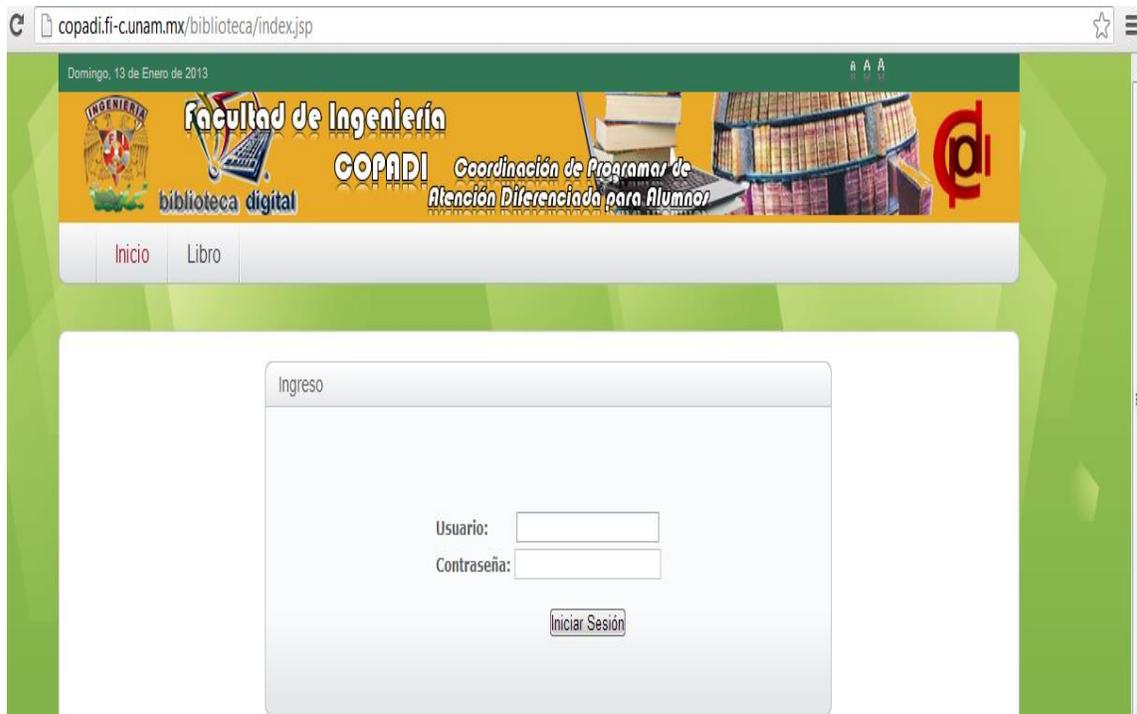


Imagen 14. Menú inicial

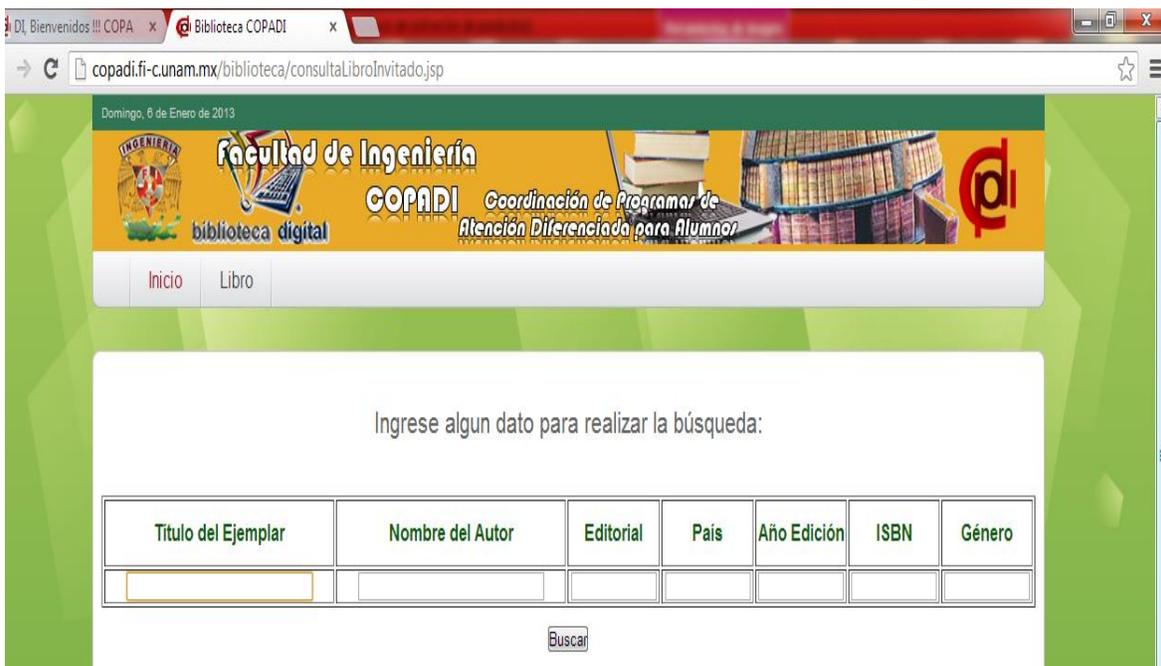


Imagen 15. Menú de búsqueda del ejemplar deseado

```

<form action= consultaLibroInvitadoz.jsp method= post name= form1 style= margin-bottom: -30px onsubmit= return
validarCamposBusquedaInvitado(form1);">
  <div align="center"><p>&nbsp;</p>
  <font face="Arial" style="font-size: 16pt">Ingrese algun dato para realizar la busqueda; </font><p>&nbsp;</p>
  <table width="100%" border="1" style="font-size: 12pt; color: #0B610B; ">
    <tr>
      <td width="25%"><div align="center"><strong>Titulo del Ejemplar</strong></div></td>
      <td width="25%"><div align="center"><strong>Nombre del Autor </strong></div></td>
      <td width="10%"><div align="center"><strong>Editorial </strong></div></td>
      <td width="10%"><div align="center"><strong>País </strong></div></td>
      <td width="10%"><div align="center"><strong>Año Edición </strong></div></td>
      <td width="10%"><div align="center"><strong>ISBN </strong></div></td>
      <td width="10%"><div align="center"><strong>Género </strong></div></td>
    </tr>

    <tr align="center">
      <td width="25%"><strong><input name="titulo" style="font-weight: 700" size="28" maxlength="50" onkeyup="value=value.toUpperCase();"
onfocus="this.input"></strong></td>
      <td width="25%"><strong><input name="autor" style="font-weight: 700" size="28" maxlength="50"
onkeyup="value=value.toUpperCase();"></strong></td>
      <td width="10%"><strong><input name="editorial" style="font-weight: 700" size="10" maxlength="20"
onkeyup="value=value.toUpperCase();"></strong></td>
      <td width="10%"><strong><input name="pais" style="font-weight: 700" size="10" maxlength="20"
onkeyup="value=value.toUpperCase();"></strong></td>
      <td width="10%"><strong><input name="anio" style="font-weight: 700" size="10" maxlength="20"
onkeyup="value=value.toUpperCase();"></strong></td>
      <td width="10%"><strong><input name="isbn" style="font-weight: 700" size="10" maxlength="20"
onkeyup="value=value.toUpperCase();"></strong></td>
      <td width="10%"><strong><input name="genero" style="font-weight: 700" size="10" maxlength="20"
onkeyup="value=value.toUpperCase();"></strong></td>
    </tr>
  </table>
  <div>
  <p align="center"><input type="submit" name="Submit" value="Buscar"></p>
</form>
<br><br>

```

Imagen 16. Parte del código que realiza la búsqueda del libro

En el menú anterior, el alumno podrá realizar la búsqueda del ejemplar ya sea por:

- Título del ejemplar
- Nombre del autor
- Editorial
- País
- Año Edición
- ISBN
- Género

Una vez realizada la búsqueda el alumno podrá solicitar el préstamo del libro, o de lo contrario, consultarlo dentro de la biblioteca.



Imagen 17. Ejemplo de búsqueda

```

69 <%
70 String consulta= "";
71 /*String temp1 = request.getParameter("titulo");
72 String temp2 = request.getParameter("autor");
73 String temp3 = request.getParameter("editorial");
74 String temp4 = request.getParameter("pais");
75 String temp5 = request.getParameter("anio");
76 String temp6 = request.getParameter("isbn");
77 String temp7 = request.getParameter("genero");*/
78 String temp1 = new String(request.getParameter("titulo").getBytes("ISO-8859-1"),"UTF-8");
79 String temp2 = new String(request.getParameter("autor").getBytes("ISO-8859-1"),"UTF-8");
80 String temp3 = new String(request.getParameter("editorial").getBytes("ISO-8859-1"),"UTF-8");
81 String temp4 = new String(request.getParameter("pais").getBytes("ISO-8859-1"),"UTF-8");
82 String temp5 = new String(request.getParameter("anio").getBytes("ISO-8859-1"),"UTF-8");
83 String temp6 = new String(request.getParameter("isbn").getBytes("ISO-8859-1"),"UTF-8");
84 String temp7 = new String(request.getParameter("genero").getBytes("ISO-8859-1"),"UTF-8");
85
86 if(temp1.length() == 0)
87     consulta=consulta;
88 else{
89     if(temp2.length() == 0 && temp3.length() == 0 && temp4.length() == 0 && temp5.length() == 0 && temp6.length() == 0 && temp7.length() == 0)
90         consulta=consulta+"1.titulo LIKE '%"+temp1+"%'";
91     else
92         consulta=consulta+"1.titulo LIKE '%"+temp1+"%' and";
93 }
94
95 if(temp2.length() == 0)
96     consulta=consulta;
97 else{
98     if(temp3.length() == 0 && temp4.length() == 0 && temp5.length() == 0 && temp6.length() == 0 && temp7.length() == 0)
99         consulta=consulta+" 1.autor LIKE '%"+temp2+"%'";
100     else
101         consulta=consulta+" 1.autor LIKE '%"+temp2+"%' and";
102 }
103
104 if(temp3.length() == 0)
105     consulta=consulta;
106 else{
107     if(temp4.length() == 0 && temp5.length() == 0 && temp6.length() == 0 && temp7.length() == 0)

```

Imagen 18. Parte del código que devuelve resultados de la búsqueda

Ahora bien, este sistema de control bibliotecario puede ser administrado por personal designado por la COPADI, al cual se le asignará un nombre de usuario y una contraseña (de aquí en adelante nos referiremos como administrador a la persona encargada de realizar esto), para poder realizar las siguientes actividades:

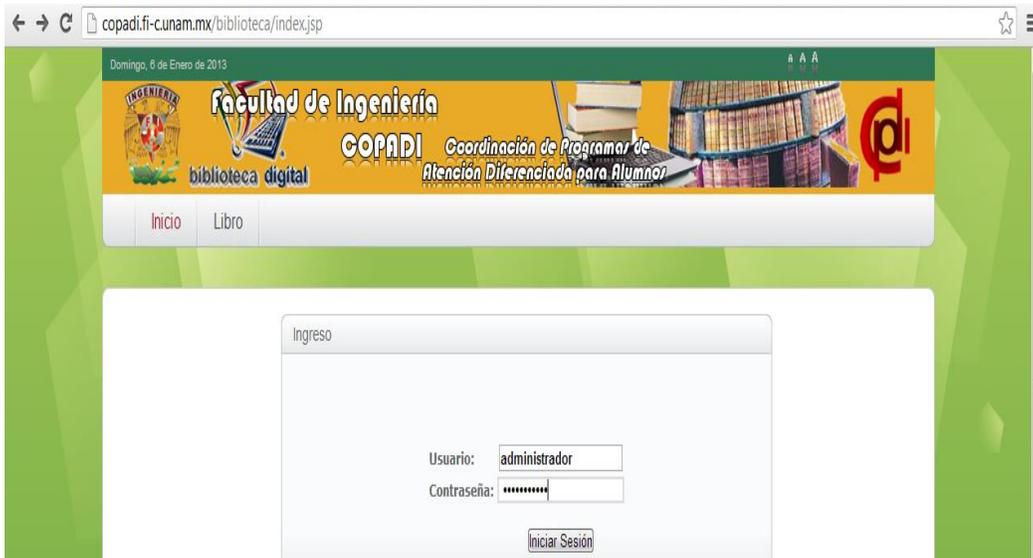


Imagen 19. Inicio de sesión

4.3.2. Sección Libro



Imagen 20. Selección "Libro"

En este apartado el administrador podrá realizar la **consulta de un ejemplar**, también podrá realizar **el alta de un libro**; para capturar e ingresar al sistema el ejemplar se deberá de introducir los siguientes datos:

- Título del ejemplar
- Nombre del autor
- Edición
- Editorial
- País
- Año Edición
- No. de páginas
- ISBN
- No. de ejemplares
- Género

Para realizar la **baja** bastará con introducir el título del libro y nos desplegará una lista con los ejemplares con los que se cuenta, y seleccionaremos el que se desee eliminar del sistema.



Imagen 21. Eliminación de ejemplares

```

<div align="center"><font face="Arial" style="font-size: 16pt;">Ejemplares que coinciden con las caracteristicas indicadas :
</font><p>&nbsp;</p></div>
<table width="100%" border="1" style="font-size: 12pt;">
<tr style="color: #0B610B;">
<td width="40%"><div align="center"><strong>Titulo del Ejemplar</strong></div></td>
<td width="40%"><div align="center"><strong>Nombre del Autor </strong></div></td>
<td width="20%"><div align="center"><strong> Eliminar </strong></div></td>
</tr>
<script>
BeanParaLibro refBeanLibro = new BeanParaLibro();
BeanParaLibro refBeanLibro2 = new BeanParaLibro();
Vector vecLibro = null;
vecLibro = refBeanLibro.obtenFichaLibro(consulta);
for(int i = 0;i < vecLibro.size(); i++){
refBeanLibro2 = (BeanParaLibro)vecLibro.elementAt(i);
}
}
</script>
<form action="bajaLibro3.jsp" method="post" name="form<%=i%>" style="margin-bottom: -30px">
<input type="hidden" name="id_libro" value="<%=refBeanLibro2.getId_libro()%>">
<tr align="center">
<td width="40%"><div align="center"><strong><%=refBeanLibro2.getTitulo()%></strong></div></td>
<td width="40%"><div align="center"><strong><%=refBeanLibro2.getAutor()%></strong></div></td>
<td width="20%"><div align="center"><input type="submit" name="Submit" value="Eliminar" onClick="return confirmSubmit()</div></td>
</tr>
</form>
</script>
</div>
</div>
<br><br>

```

Imagen 22. Parte del código para dar de baja un ejemplar

Para la actualización de uno o varios libros, buscaremos el título del libro y nos desplegará el listado contenido en la base, seleccionaremos el que se desee modificar y actualizaremos los datos necesarios, según sea el caso



Imagen 23. Búsqueda para actualización



Imagen 24. Actualización de datos de un ejemplar

4.3.3. Sección Estudiante

En esta parte, el administrador podrá realizar la:

- **Consulta de un estudiante.** Introduciremos ya sea el número de cuenta, o en su caso introduciremos el nombre y los apellidos del estudiante a buscar.
- **Alta de un estudiante** dentro de la base de datos. En esta parte se podrá ingresar a los datos de un estudiante dentro de la base de datos, la única restricción es que pertenezca al PARA.

Martes, 9 de Enero de 2013

Facultad de Ingeniería
COPADI
Coordinación de Programas de
Atención Diferenciada para Alumnos
biblioteca digital

Salir Libro Estudiante Préstamo

Ingrese los datos para dar de alta un nuevo Estudiante:

302222030	* No. de Cuenta
EDUARDO	* Nombre
BAUTISTA	* Paterno
SALAZAR	Materno
26159297	Teléfono
eduardoba06@gmail.com	Correo Electrónico
5527167223	No. Celular

Enviar

Imagen 25. Alta de estudiante dentro del sistema bibliotecario

- **Baja de un estudiante.** Se podrá realizar la baja de la base de datos ya sea temporal o definitiva de estudiantes que por algún motivo ya no pertenezcan al PARA

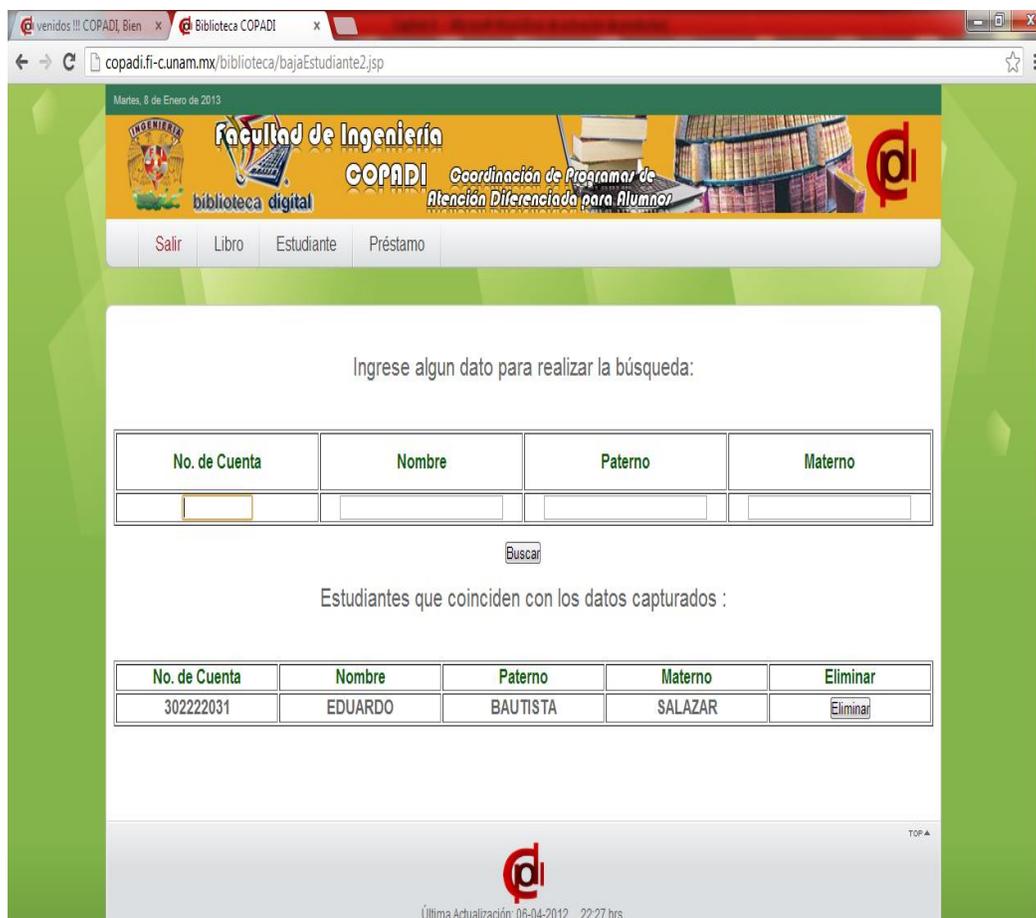


Imagen 26. Baja de estudiante

- **actualización de datos de un estudiante.** Se podrá realizar la actualización de datos de aquellos estudiantes que así lo requieran.

4.3.4. Sección Préstamo

- **Reporte.** En esta parte se podrá visualizar un reporte general con el nombre del ejemplar prestado y el nombre del estudiante que solicitó dicho ejemplar hasta ese momento.
- **Nuevo préstamo.** En esta sección el administrador realizará el registro de préstamo del ejemplar o los ejemplares, para ello necesitará introducir el número de cuenta del alumno y autorizar el préstamo.



Imagen 27. Préstamo de libros

- **Devolución.** En esta parte se realizará la devolución del o los ejemplares que fueron prestados. Bastará con introducir el número de cuenta del alumno que solicitó dicho préstamo y dar clic en el botón “devolución” para que sea registrado la devolución.



Imagen 28. Devolución de ejemplares

4.3.5. Conformación de la base de datos.

Diagrama de flujo del préstamo bibliotecario

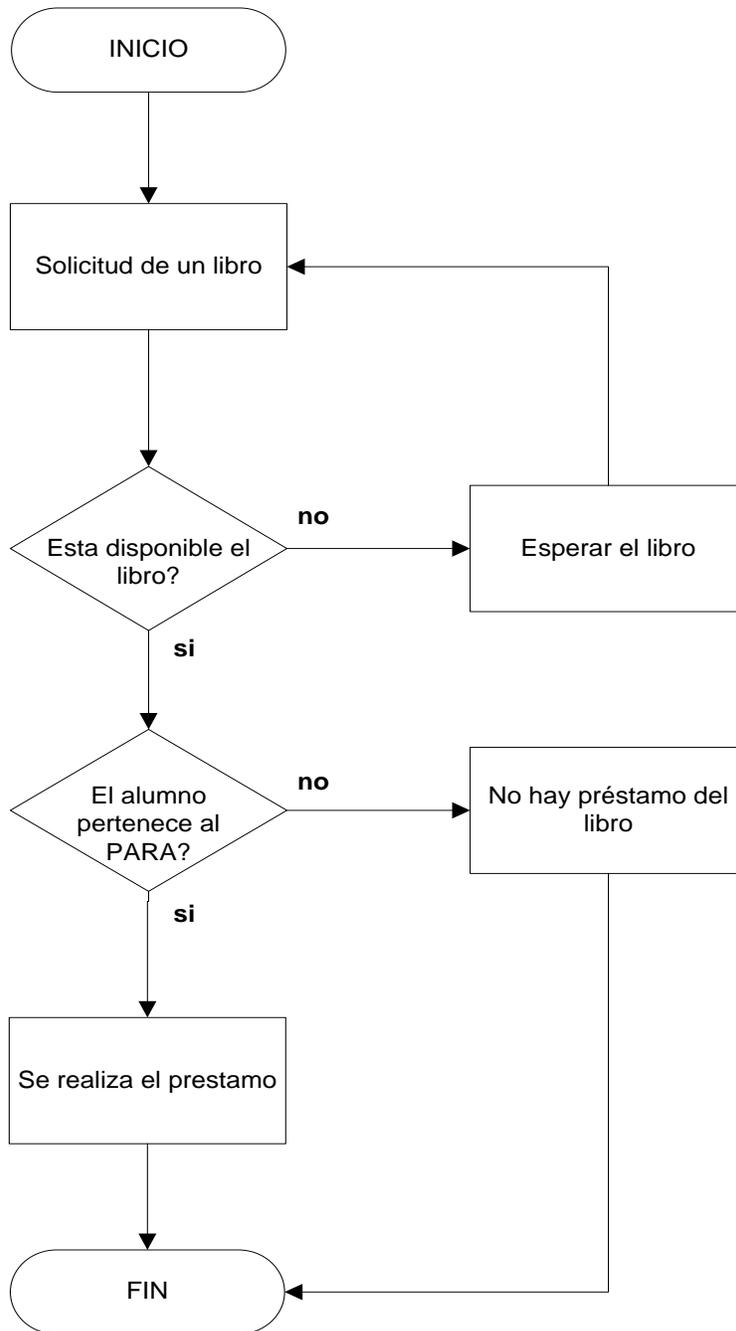


Imagen 29. Diagrama de flujo préstamo bibliotecario

Nuestra base de datos quedará conformada de la siguiente manera:

Base de datos:

Alumno:

CAMPO	TIPO DATO	DESCRIPCIÓN
numcuenta	integer	Número de cuenta del estudiante usado como identificador
nombre	varchar(50)	Nombre del estudiante
paterno	varchar(50)	Apellido paterno del estudiante
materno	varchar(50)	Apellido materno del estudiante
telefono	varchar(12)	Teléfono del estudiante
celular	varchar(13)	Número de celular del estudiante
correoe	varchar(60)	Correo electrónico del estudiante
activo	char(1)	Define si el estudiante aún puede pedir un préstamo o no

Libro:

CAMPO	TIPO DATO	DESCRIPCIÓN
id_libro	integer	Identificador del ejemplar
titulo	varchar(50)	Título del ejemplar

autor	varchar(50)	Autor del ejemplar
edicion	varchar(100)	Edición del ejemplar
editorial	varchar(100)	Editorial del ejemplar
pais	varchar(100)	País del ejemplar
anioedicion	varchar(10)	Año de edición del ejemplar
paginas	Integer	Páginas con las que cuenta el ejemplar
isbn	varchar(15)	Isbn del ejemplar
clasificacion	varchar(15)	Clasificación del ejemplar
ejemplares	Integer	Número de ejemplares con los que se cuenta
genero	varchar(50)	Género del ejemplar

Préstamo:

CAMPO	TIPO DATO	DESCRIPCIÓN
id_prestamo	integer	Identificador del préstamo
id_libro	integer	Identificador del ejemplar
numcuenta	integer	Número de cuenta del estudiante, usado como identificador
fecha_prestamo	date	Fecha en que se realiza el préstamo del ejemplar
fecha_devolucion	date	Fecha de devolución del ejemplar
multa	varchar(300)	Multa para el préstamo
devolucion	varchar(1)	Estado de préstamo del ejemplar, 0 si aún no se ha

		devuelto, 1 si ya se devolvió
--	--	-------------------------------



relación 1-muchos, se crea la entidad prestamo.

Modelo Lógico: Diagrama entidad-relación

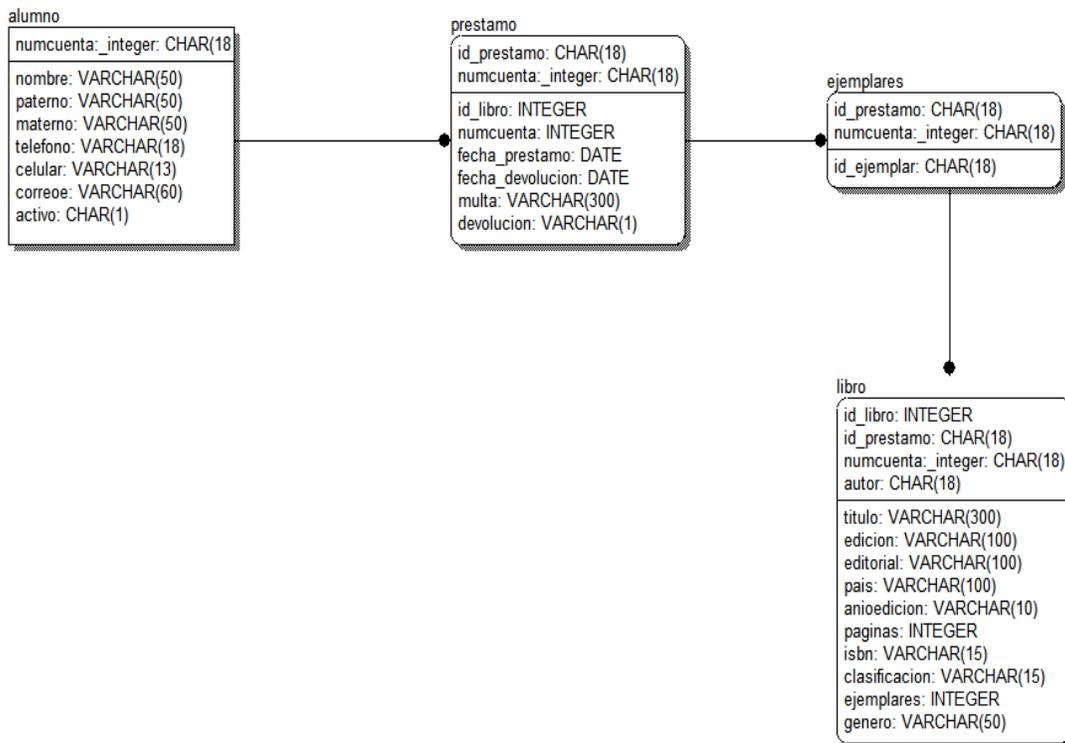


Imagen 30. Esquema de la base de datos, modelado en Erwin

El sistema cuenta con 3 beans:

- BeanAlumno
 - o Donde se encapsulan todos los atributos de esta entidad
- BeanLibro
 - o Donde se encapsulan todos los atributos de esta entidad

- BeanPrestamo
 - o Donde se encapsulan todos los atributos de esta entidad, se definen las funciones para dar de alta un ejemplar, dar de baja un ejemplar, actualizar datos de un ejemplar, realizar el préstamo de un ejemplar, realizar la devolución de un ejemplar, consultar un ejemplar, consultar un estudiante, darlo de alta o de baja y actualizar su información.

Los archivos con los que cuenta el sistema para realizar las diferentes tareas son las siguientes:

1. **index.jsp**
 - a. que es donde se valida si es un usuario válido para entrar al sistema
2. **salir.jsp**
 - a. que es donde se destruye la sesión y cierra el sistema para seguridad
3. **consultaLibro.jsp:**
 - a. que es donde se realiza la consulta de un ejemplar dentro del sistema
4. **consultaLibroInvitado.jsp**
 - a. que es donde se realiza la consulta de un ejemplar para usuarios que no tienen privilegios para usar el sistema
5. **consultaLibro2.jsp**
 - a. que es donde se envía la información y se muestran los resultados de la búsqueda.
6. **altaLibro.jsp**
 - a. donde se muestra el formulario para capturar los datos y enviarlos
7. **altaLibro2.jsp**
 - a. donde se reciben los datos y se envían a la base de datos para realizar la operación
8. **bajaLibro.jsp**
 - a. donde se muestra el formulario para capturar los datos y enviarlos
9. **bajaLibro2.jsp**
 - a. donde se reciben los datos y se envían a la base de datos para realizar la operación
10. **actualizarLibro.jsp**
 - a. donde se muestra el formulario para capturar los datos y enviarlos
11. **actualizarLibro2.jsp**
 - a. donde se reciben los datos y se envían a la base de datos para realizar la operación
12. **altaEstudiante.jsp**
 - a. donde se muestra el formulario para capturar los datos y enviarlos
13. **altaEstudiante2.jsp**
 - a. donde se reciben los datos y se envían a la base de datos para realizar la operación

14. bajaEstudiante.jsp

- a. donde se muestra el formulario para capturar los datos y enviarlos

15. bajaEstudiante2.jsp

- a. donde se reciben los datos y se envían a la base de datos para realizar la operación

16. actualizarEstudiante.jsp

- a. donde se muestra el formulario para capturar los datos y enviarlos

17. actualizarEstudiante2.jsp

- a. donde se reciben los datos y se envían a la base de datos para realizar la operación

18. consultaEstudiante.jsp

- a. donde se muestra el formulario para capturar los datos y enviarlos

19. consultaEstudiante2.jsp

- a. donde se reciben los datos y se envían a la base de datos para realizar la operación

20. prestamo.jsp

- a. donde se muestra el formulario para capturar los datos y enviarlos

21. prestamo2.jsp

- a. donde se reciben los datos y se envían a la base de datos para realizar la operación

22. devolucion.jsp

- a. donde se muestra el formulario para capturar los datos y enviarlos

23. devolucion2.jsp

- a. donde se reciben los datos y se envían a la base de datos para realizar la operación

24. consultaPrestamos.jsp

- a. Muestra los libros en estado de préstamo.

4.4. Implementación del Sistema

Para la implementación del sistema Cursos Intersemestrales y el sistema de la Biblioteca Digital, se utilizó el servidor con el que COPADI ya contaba. Este servidor nos permitió alojar nuestros sistemas dentro del portal de la coordinación <http://copadi.fi-c.unam.mx>. Asimismo nos permito almacenar las bases de datos; tanto las ya establecidas como las que se fueron generando durante el transcurso del desarrollo.

Las especificaciones técnicas del Servidor adquirido son:

Servidor DELL PowerEdge T300

Modelo: PowerEdge T300



Memoria: RAM 24 Gb.

Procesador: Intel® Celeron® Dual-Core Intel® Core™2 Duo®

Almacenamiento: 2 discos duros SATA Hot-Plug 500 Gb c/u.

Quemador: DVD 16X

Otros: 2 puertos NIC Gigabit, 2 puertos USB 2.0, 1 conector de video, 1 conector serial

Al servidor se le ha instalado el sistema operativo Fedora 11 linux que es parte del Proyecto Fedora que tiene como patrocinador a RedHat.

Así mismo para poner en funcionamiento de forma global al servidor, se hizo la solicitud de una dirección IP Homologada con la finalidad de dar el servicio a los usuarios en cualquier parte del mundo. Así pues, el servidor tiene la dirección 132.248.139.201 con su respectivo nombre de dominio (copadi.fi.- c.unam.mx).

4.5. Pruebas y resultados

Las primeras pruebas que se realizaron fue la publicación (dentro del portal COPADI; en la sección destinada específicamente para este sistema) del listado de algunos cursos que se impartirían en el intersemestre, así como su respectiva descripción, en esta etapa no se habilitó la opción de los registros ya que se consideró que esta parte necesitaría una prueba independiente; la intención de esto fue observar el comportamiento de la manera en que se visualizaba la página web en los navegadores de internet, se seleccionaron los navegadores más conocidos y utilizados, algunos de ellos:

- Internet Explorer 7 y 8
- Mozilla Firefox v.18.0
- Google Chrome
- Opera v 12.12
- Safari 5.1.7

Al comprobar el correcto funcionamiento y visualización de los datos en estos navegadores, se publicó la lista completa de los cursos, y se habilitó la opción de registro.

La siguiente etapa fue la de realizar una “prueba de carga”, la cual consistió en la preparación de computadoras (alrededor de 25 ordenadores, entre equipo de escritorio y lap-tops) y la asignación de 25 usuarios para realizar:

- a) El registro de los 25 estudiantes dentro del mismo curso, con una vacante total de 20 lugares.
- b) El registro de aquellos 25 estudiantes, un estudiante por cada curso disponible.

Al estar ingresando cada usuario sus datos para el registro al curso, las primeras 20 solicitudes de registro en ser recibidas fueron aceptadas y procesadas, mientras que las 5 restantes fueron rechazadas, a las cuales se les envió un mensaje de: "NO HAY VACANTES". Para cada inscripción realizada y aceptada se generó e imprimió un COMPROBANTE DE INSCRIPCIÓN, dando por terminado el proceso. Los datos de los estudiantes aceptados fueron guardados en una lista de Excel, generando así un reporte con los alumnos inscritos en este curso.

Para el segundo caso, en el cual cada estudiante se registraría a un curso de los publicados, al realizarse el proceso y enviarse la petición al servidor, este las recibió y proceso, por lo que en la hoja de Excel se generó un registro de estudiante diferente en cada uno de los cursos.

Con lo anterior, comprobamos que el sistema es capaz de procesar varias peticiones de registros a cada curso, si bien no se realizó una prueba con un gran número de usuarios fue porque el cupo a cada curso oscila entre los 20 y 50 vacantes, los de mayor demanda tienen una totalidad de 20 lugares, mientras que los de menor demanda llegan a tener hasta 50 lugares (de los cuales, en promedio llegan a inscribirse en un 80%).

Para la biblioteca digital, se realizó una prueba de rendimiento, en el cual el principal objetivo fue analizar la velocidad de la aplicación.

La herramienta utilizada fue Pingdom, es un analizador web gratuito para medir el tiempo de carga desglosado de nuestra página web.

Como vemos, no hay más que escribir la dirección URL del dominio o página que deseamos testear y comenzará a realizar un análisis detallado de todos los tiempos de carga que registra esa determinada página.

Es muy útil por ejemplo para descubrir que puntos son los que más están sobrecargando una página web, como pueden ser, imágenes pesadas, o como pueden ser también archivos CSS mal estructurados, códigos javascript pesados, etc.

Se eligió esta herramienta por la forma clara y concisa que tiene de mostrar todos los datos estadísticos, también la gran cantidad de datos que nos ofrece. Una vez terminado el chequeo nos muestra la siguiente pantalla, con los siguientes puntos registrados:

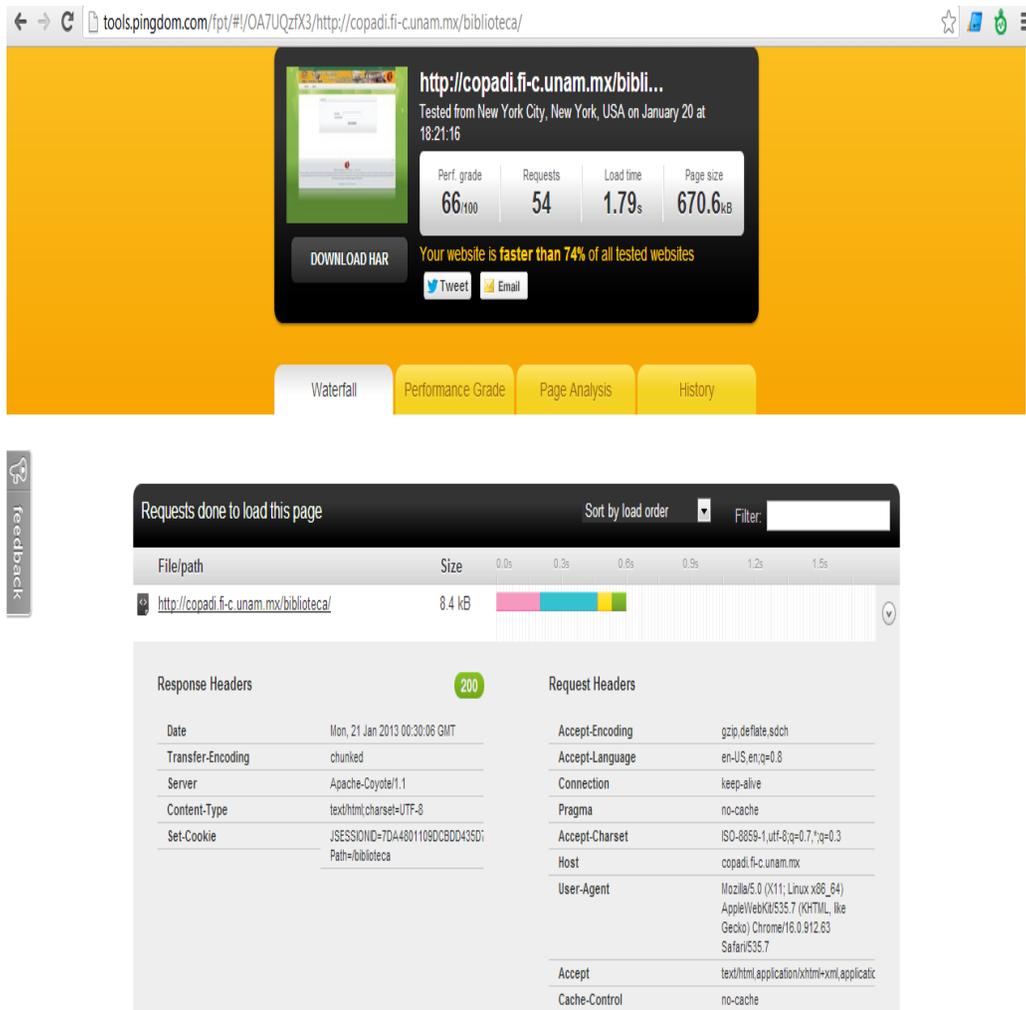


Imagen 31. Waterfall. Resultados de la prueba

Con lo anterior tenemos el siguiente resultado:

Viene el número de request, para los valores del analizador menciona que es preferible tenerlo bajo, menor de 50.

En nuestro caso obtuvo un puntaje de 54.

En Load Time, intentar estar por debajo de los 2 segundos es la recomendación ideal del probador.

En nuestro caso fue de 1.79s, dentro del rango aconsejado.
 En page size, la media de la industria es 600 kb.
 En nuestro caso fue de 670.



Imagen 32. Waterfall. Resultados de la prueba

Waterfall: Nos indica todos y cada uno de los archivos que han sido llamados en la carga de la página, su nombre y tipo y muy importante la franja que están ocupando dentro de los segundos de carga total del portal. También podemos ver que tienen una serie de colores, cada una de las franjas de tiempo de carga, esos colores nos indican lo que opina el sistema sobre esos tiempos, si la franja es azul todo OK, está dentro la normalidad, si es amarilla, también podemos considerar un OK pero prestando atención a que podía no ser amarilla, roja, está excediendo el tiempo normal o que debería tener el archivo específico.

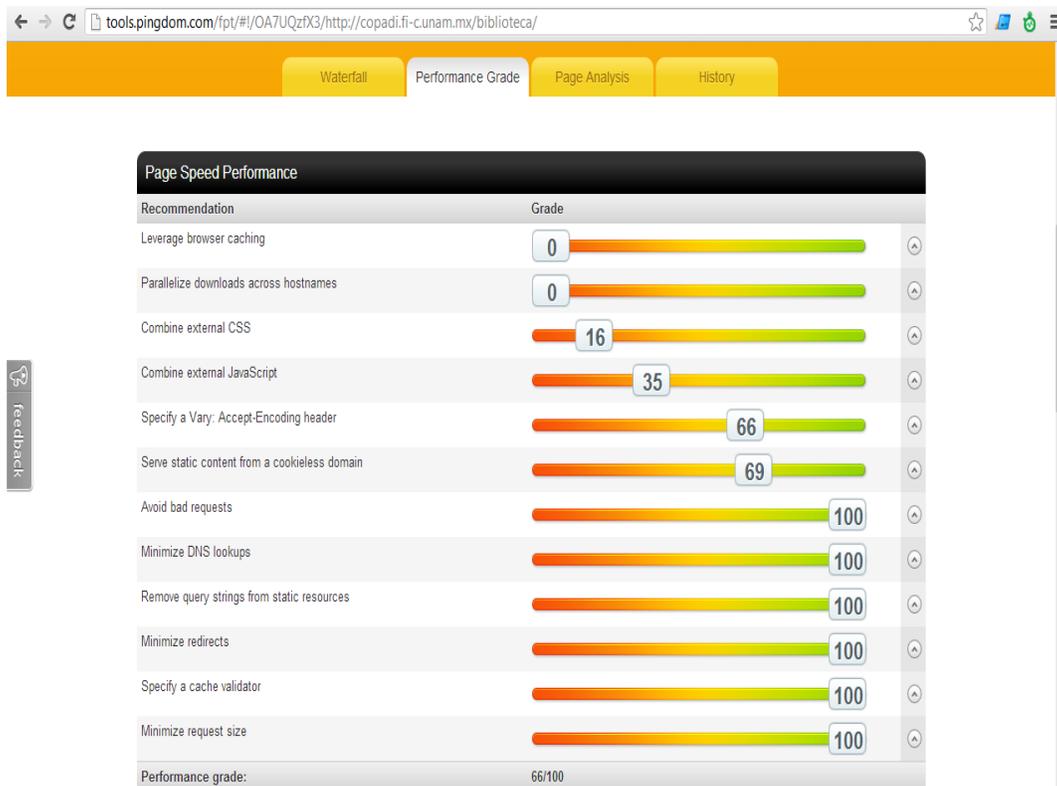


Imagen 33. Performance Grade

PerformanceGrade: Este es el apartado donde nos muestran un resumen del grado de optimización o rendimiento que muestra la página web, desglosado por tipo de componentes. Por ejemplo podemos ver el grado de rendimiento para los archivos del tipo combinación externa de javascript, combinación externa de css, caché navegador, etc. Viendo en qué puntos no recibimos un alto grado y centrando nuestra atención en ellos.

Acorde a la información de la página con que se realizó la prueba:

Server Response Codes

To make it easy for you to differentiate between the HTTP response codes in the waterfall chart, we've color-coded the text and background of each URL.

URL 2xx	The server responded with a successful code
URL 3xx	The request was redirected to another target
URL 4xx	A client error occurred, for example 404 page not found
URL 5xx	A server error occurred, for example 500 internal server error
URL Error	Connection error, no response from the server

Imagen 34. Tabla de contestación del servidor

El resultado general que arroja este analizador es un puntaje de 66/100, equivalente a *El servidor respondió de manera exitosa* (“The server responded with a successful code”).

CONCLUSIONES

Conclusiones generales

El espectacular auge de los sistemas Web, junto con el creciente número de personas que utilizan la Internet para búsquedas y recopilación de información, hacen preciso la creación de aplicaciones que ayuden a satisfacer la búsqueda y recopilación de información de forma más sencilla. Así mismo, contando con las herramientas, los recursos y la tecnología, debe ser posible construir y disponer de aplicaciones que apoyen a la formación académica en las instituciones educativas, como lo es la Facultad de Ingeniería.

Por lo anterior, una de las motivaciones iniciales de este trabajo fue la de ofrecer sistemas que no sólo resultaran útiles a “casi”, sino en realidad a toda la comunidad académica de la Facultad de Ingeniería y externas a esta. Finalmente en éste capítulo se presentan las conclusiones, contribuciones y limitaciones que se encontraron durante el desarrollo de esta tesis.

El sistema de inscripciones y el sistema de la biblioteca digital cubre las necesidades que actualmente necesita COPADI por carecer de las mismas. Con ello se cumplió con su propósito específico, el cual, es proveer de aplicaciones que sirva de apoyo y facilite la labor tanto del alumnado como de los encargados de esta coordinación.

Contribuciones y mejoras

Actualmente el sistema de inscripciones lleva funcionando dos intersemestres, se ha cubierto en su totalidad las necesidades planteadas inicialmente, y principalmente ha ayudado en gran medida a llevar un control y registro detallado en las inscripciones, disminuyendo los errores humanos a la hora de realizar los registros de manera manual.

Algunas mejoras que se podrían realizar a este sistema es incluir un apartado en la cual los alumnos que hayan tomado algún curso puedan insertar un comentario constructivo sobre el mismo, o en su caso describir algunos otros temas que les hubiera gustado haber visto, y esto ser visible para los demás alumnos que visiten la página y que se animen a inscribirse a algún curso, todo esto es pensado en beneficio de los alumnos.

Asimismo podría integrarse un módulo más en la cual se realicen las encuestas de evaluación tanto del alumno como para el instructor que impartió las clases.

Ahora bien, para la biblioteca digital, por ser de uso exclusivo para alumnos pertenecientes al Programa de Alto Rendimiento Académico PARA, y al no carecer de restricciones (tiempo de devolución, cantidad de ejemplares para préstamo, entre otros) podría agregarse un contador de ejemplares por cada título con el que cuente la biblioteca, esto para que se tenga más claro cuántos libros de cada título fueron prestados y cuantos quedan en reserva para futuras consultas.

Adicionalmente se podría incluir el índice de cada libro para una consulta rápida y que el alumno analice si efectivamente el libro a consultar es el que se apega a las necesidades de búsqueda que está realizando.

Podríamos mencionar algunas otras contribuciones, pero cabe señalar que ésta tesis cubre todos sus objetivo y más bien, los puntos muestran posibles trabajos futuros y pueden servir de base para facilitar su formulación de forma más concreta y ayudar a abordarlos como continuación y extensión del presente trabajo.

BIBLIOGRAFÍA

HTML 4 for the World Wide Web
Elizabeth Castro.
Berkeley, California: Peachpit, c2000

Diseño y desarrollo web con Dreamweaver CS5
Osborn, Jeremy
Anaya Multimedia, 2011

Diseño de sitios web
Pipes, Alan
Promoprees, 2011

La biblia de Internet.
Brian Underdahl y Keith Underdahl.
Madrid: Anaya Multimedia, c2000.

Learning jQuery 1.3 : better interaction design and web development with simple
JavaScript techniques
Jonathan Chaffer, Karl Swedberg
Packt Publishing Ltd, February 2009

Foundation website creation with CSS, XHTML, and JavaScript.
Lane Jonathan, Meitar Moscovitz, Joseph R. Lewis.
Berkeley, California: Friends of ED, c2008.

Todo lo básico que debería saber sobre programación orientada a objetos en Java
García Llinás, Luis Fernando
Ediciones Uninorte, Septiembre 2010

Java con programación orientada a objetos y aplicaciones en la WWW.
Paul S. Wang, tr. Eloy Pineda.
México: International Thomson, c2000.

Aprendiendo a convertirse en un webmaster en 14 días.
James L Mohler. Tr Jorge Luis Gutiérrez.
México: Prentice Hall c1998.

Pro Web 2.0 application development with GWT
Dwyer, Jeff
Prentice-Hall, 2008

Beginning PHP and Oracle: from novice to professional.
Gilmore W. J. (W. Jason), Bob Bryla.
Berkeley, California: Apress, c2007

PHP programming solutions
Vikram Vaswani
New York: Mc Graw-Hill, c2007.

<http://www.w3.org/Style/CSS/> [consultado junio 2012, 16:00]

<http://jquery.com/> [consultado junio 2012, 8:00]

http://docs.jquery.com/Downloading_jQuery [consultado junio 2012, 10:00]

<http://recursosweb.unam.mx/recursos-web/lineamientos-unam/> [consultado noviembre 2012]

<http://www.baluart.net/ejemplos/art31/menu-horizontal-desplegable.html> [consultado marzo 2012]

<http://www.adobe.com/support/documentation/en/dreamweaver/> [consultado abril 2012]

<http://tomcat.apache.org/tomcat-5.5-doc/index.html> [consultado abril 2012]

<http://netbeans.org/kb/> [consultado marzo 2012]

http://docs.fedoraproject.org/en-US/Fedora/14/html/Installation_Guide/ [consultado abril 2012]

<http://www.rational.com/products/rup/> [consultado julio 2012]

<http://www.omg.org/uml/> [consultado agosto 2012]

<http://www.extremeprogramming.org> [consultado marzo 2012]

<http://www.xprogramming.com> [consultado marzo 2012]

<http://c2.com/cgi/wiki?ExtremeProgrammingRoadmap> [consultado abril 2012]

<http://thecoadletter.com/download/#fddguide> [consultado junio 2012]

http://www.nebulon.com/arties/fdd/l_estfdd.html [consultado septiembre 2012]

<http://www.globetesting.com/pruebas-de-rendimiento/> [consultado diciembre 2012, 18:00]

http://www.corporacionsybven.com/portal/index.php?option=com_content&view=article&id=246 [consultado noviembre 2012. 16:00]

<http://seo.pcweb.info/pingdom-tools/> [consultado diciembre 2012, 18:00]

<http://tools.pingdom.com/fpt/#!/GzKm1Lghp/http://copadi.fi-c.unam.mx/biblioteca/>
[consultado diciembre 2012, 18:00]

<http://www.globetesting.com/pruebas-de-rendimiento/> [consultado diciembre 2012, 15:00]

Apéndice A

NORMATIVIDAD Y LINEAMIENTOS GENERALES PARA USO DE PÁGINAS WEB EN LA FACULTAD DE INGENIERÍA

NOVIEMBRE DEL 2001

CONTENIDO

1. Generales
2. Normatividad para la publicación de páginas web en la red de la facultad de ingeniería
 - 2.1. Usuarios
 - 2.2. Publicación de trabajos en la web
 - 2.2.1. Publicación de información sobre actividades académicas y culturales
 - 2.2.2. Tesis
 - 2.2.3. Proselitismo académico y estudiantil para elecciones
3. Sugerencias técnicas para la publicación de páginas web en la Facultad de Ingeniería.
- 4.

Anexo1: Consideraciones sobre la página principal web de la Facultad de Ingeniería.

Anexo2: Procedimiento para dar de alta paginas en sitio web principal de la Facultad de Ingeniería

Anexo3: Administradores de la web

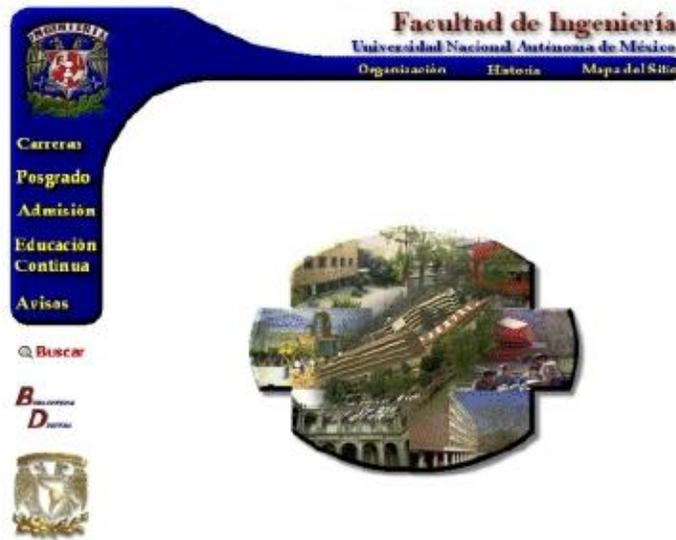
El World Wide Web, o simplemente Web, como generalmente se denomina, es un servicio que se obtiene a través de Internet. Está formado por una colección de documentos (conocidos como páginas) interconectados que se encuentran almacenados en computadoras ubicadas por todas partes del mundo. Dichos documentos pueden contener texto, gráficos y sonidos, y son conocidos como páginas.

La Facultad de Ingeniería cuenta con una página principal de Web para publicar información sobre sus actividades académicas y culturales. A partir de ésta, se tienen ligas a otras páginas relacionadas con la Facultad (Divisiones, Secretarías, y de profesores, entre otras). Debido a la necesidad de mantener una presencia adecuada y una organización de la información en estas páginas, es preciso regular su diseño, contenido, y edición. Por tal motivo, todos los nodos que tengan funciones de servidores de Web de la Facultad de Ingeniería, deberán seguir los lineamientos contenidos en el presente documento.

El área responsable de la página Web de la Facultad de Ingeniería, es la Secretaría General. Como parte integrante de Secretaría General, la Unidad de Servicios de Cómputo Académico (UNICA), será la responsable técnica de la operación, seguridad y respaldos; el Departamento de Información y Estadística será la responsable de los contenidos y la Coordinación de Comunicación de la imagen gráfica institucional (ver detalles en el ANEXO 1).

La página de la Facultad de Ingeniería será el punto de partida para acceder a las demás páginas de Secretarías y Divisiones. Quedando estrictamente prohibido que las páginas de otras Divisiones o Secretarías que no sea la de la Secretaría General den la impresión de representar a la Facultad de Ingeniería en forma oficial. El formato acordado es:

Página principal
<http://www.fi-a.unam.mx>



Página secundaria



Nombramientos



CATEGORÍA	TOTAL
• PROFESORES DE CARRERA	237
• PROFESORES DE ASIGNATURA	1015
• TÉCNICOS ACADÉMICOS	87
• AYUDANTES DE PROFESOR	377
TOTAL	1714

Los diseños de logos, fondos, imágenes etc., deben ser originales.

Cualquier caso no previsto en el presente documento, será resuelto por el Comité Asesor de Cómputo de la Facultad de Ingeniería. Y en caso específico del punto 2.2.3 por el Consejo Técnico y/o en su caso por la Comisión de Vigilancia de la elección correspondiente.

2. NORMATIVIDAD PARA LA PUBLICACIÓN DE PÁGINAS DE WEB EN LA FACULTAD DE INGENIERÍA

2.1 USUARIOS

Para que una persona pueda crear y/o mantener una página en la Web en la facultad de Ingeniería, deberá ser miembro de la comunidad (alumno, personal académico o personal administrativo debidamente autorizado), y cumplir con:

- **PARA ALUMNOS:** Ser usuario de las salas de la red de cómputo de la Facultad de Ingeniería y su trabajo deberá estar avalado por un integrante del personal académico de la Facultad de Ingeniería.
- **PARA EL PERSONAL:** Identificarse como personal laboral activo de la Facultad de Ingeniería.

El solicitante debe llenar un formato en el área correspondiente en el que indique las razones por las cuales desea que su página se publique en algún servidor de Web de la Facultad de Ingeniería. Este formato deberá dirigirse al webmaster de la División o Secretaría en la cual quiera que aparezca su página. La página debe estar terminada por completo. Será responsabilidad del webmaster que reciba esta solicitud, revisar el contenido de la página. No se publicará la página hasta que sea aprobada por el webmaster correspondiente. No se darán de alta páginas que este bajo construcción.

En el caso de páginas dependientes de la página principal ver el procedimiento descrito en el ANEXO 2.

Los puntos que se tendrán en cuenta para su aprobación son los siguientes:

- La publicación de las páginas de los usuarios estará en función de la capacidad que se tenga en cuanto a espacio en los servidores de Web. Si el webmaster o el administrador de red consideran que los recursos del servidor no son suficientes, no se podrá publicar la página aunque ésta cumpla con los demás requisitos. Por lo tanto, se debe recomendar a las personas interesadas en elaborar una página, que pregunte al webmaster correspondiente sobre los recursos de que puede disponer, incluyendo la cantidad máxima en disco duro que puede utilizar.
- El contenido de las páginas deberá ser académico o de aplicación administrativa de interés para la Facultad.
- El tema sobre el que trate la página será libre, con la restricción de no lucrar con el servicio que la Facultad le brinda.
- Nota: Salvo en aquellos proyectos especiales en que la Facultad de Ingeniería participe en convenios con la iniciativa privada.
- Mantener el respeto que nos reconoce como universitarios
- No se publicarán páginas que contengan errores ortográficos. En caso de encontrarlos, el webmaster le indicará al usuario que debe corregirlos.
- Toda página deberá estar escrita con un mínimo del 90% de idioma español. En caso de que se requiera publicar la información en otro idioma, se deberá contar con una referencia de cambio de idioma (icono, imagen, etc.)
- Las páginas no podrán dar a entender que representan en forma oficial a la Facultad de Ingeniería ni alguna de sus Divisiones o Secretarías. Por ello

queda prohibido el uso de escudos y logotipos de la Facultad, que en su apariencia sean similares a los de alguna Secretaría o División

Una vez aprobada la página, se aplicará lo siguiente:

- La permanencia de la página será de un semestre escolar.
- La renovación de la publicación de la página estará sujeta a los recursos disponibles en el servidor Web (ej. espacio en disco duro).
- El usuario podrá promocionar su página en uno o varios exploradores de Internet. Cada usuario deberá realizar el procedimiento necesario para dar de alta su página en cada explorador, posteriormente deberá enviar un correo al administrador de Web correspondiente, notificando estas altas.
- El webmaster de la División o Secretaría en la cual se encuentre la página del usuario revisará periódicamente el contenido de las páginas. Si observa que el usuario realizó modificaciones que no cumplan con lo estipulado, tiene la facultad de desligar la página del usuario. En caso de reincidencia, se dará de baja su página permanentemente.

2.2 PUBLICACIÓN DE TRABAJOS EN LA WEB

2.2.1 PUBLICACIÓN DE INFORMACIÓN SOBRE ACTIVIDADES ACADÉMICAS Y CULTURALES EN LA WEB

Estas páginas contendrán información relacionada con las actividades de una Secretaría o División y aprobada por el representante del Comité Asesor de Cómputo del área en cuestión. El contenido del tema deberá estar avalado por un integrante del personal académico de la Facultad de Ingeniería.

2.2.2 TESIS

Este apartado se refiere a aquellas personas que hayan realizado su trabajo de tesis para publicarse en la Web y esté en alguna de las siguientes modalidades:

El contenido de la tesis esté diseñado para colocarse en la Web. No importando el tema. Que la tesis en sí sea una aplicación en la Web. Aplicación o sistema externo que no tenga que ver con la Facultad de Ingeniería. Aplicación orientada para la Web de la Facultad de Ingeniería y se quiera incorporar de forma oficial.

Para dar de alta las páginas, los tesistas, deberán presentar una carta firmada por su director de tesis que avale su contenido. La carta deberá ser entregada al webmaster de la División o Secretaría donde se desee publicar la página.

En el caso de páginas dependientes de la página principal ver el procedimiento escrito en el ANEXO 2.

La duración de la publicación de la tesis en la Web será de al menos un semestre, condicionada a la normatividad vigente y quedando a consideración del webmaster la permanencia de la página en función de los recursos disponibles.

Para el caso en que la tesis se encuentre en el inciso b2, aplicará lo siguiente:

- Los tesistas tendrán que proporcionar una copia de los archivos de su aplicación.
- El trabajo de tesis será evaluado técnicamente en su funcionamiento por el webmaster y/o el administrador de red, para decidir si es conveniente incorporar su publicación en forma oficial en la página de la Facultad. En caso de que existan dos tesis con temas similares, se seleccionará la más apta funcionalmente, brindando el debido reconocimiento a su(s) autor(es).

- Los tesisistas deben estar conscientes de que están cediendo su aplicación a la Facultad, por lo que ésta puede ser actualizada, conservando sus correspondientes créditos.
- Las páginas de tesis incorporadas podrán ser referenciadas desde cualquier hoja de la Facultad.

2.2.3 PROSELITISMO ACADÉMICO Y ESTUDIANTIL PARA ELECCIONES

Para la publicación de páginas en WEB en los servidores de la Facultad y que tenga como fin realizar proselitismo electoral a favor de alguna fórmula que participe en los diferentes procesos electorales que se llevan a cabo en la dependencia, se deberán tomar en cuenta las siguientes observaciones:

No utilizar logotipos de la Facultad o de la Universidad. b) El periodo de permanencia y el contenido estará sujeta a lo establecido en el capítulo referente a propaganda electoral de la convocatoria respectiva. c) La página deberá sujetarse a lo establecido en el presente reglamento.

Las páginas de la WEB residentes fuera de la universidad también deberán apegarse a los incisos a) y b).

3. SUGERENCIAS TÉCNICAS PARA LA PUBLICACIÓN DE LAS PÁGINAS WEB EN LA FACULTAD DE INGENIERIA

Esta sección presenta sugerencias técnicas para el desarrollo de páginas Web, empleando el lenguaje de programación HTML, con el fin de dar un mejor soporte en la organización, diseño, e integridad de la información que se habrá de presentar al concluir la construcción de una página. Se hace la aclaración de que

estas sugerencias se realizan vigentes a la fecha del presente documento, lo cual se debe de tener en consideración, debido a la dinámica que existe en el área de cómputo.

El lenguaje de programación HTML proporciona diversas herramientas para la creación de una página de Web y el objetivo principal es sugerir el uso óptimo de éstas.

IDEAS DE ORGANIZACIÓN

Elementos para una mejor presentación en línea

- Ortografía y redacción.
- Escribir con claridad y ser breve. Organización óptima para evitar la pérdida de los objetivos.
- Estructura de autosoporte.
- Resaltar sólo lo importante.
- Tomar en cuenta las plataformas en las que se pueda estar visualizando su página.
- Diseño y formato.
 - Imágenes.
 - Encabezados.
 - Agrupación de la información congruente.
 - Formato constante.
 - Vínculos correctos
 - División en las mismas páginas.
 - Generación de documentos vinculados.
 - Estructuras.

ORTOGRAFÍA Y REDACCIÓN

Cuando se consulta una página Web, generalmente la primera impresión que se recibe es básica para continuar revisándola, pero por muy buena presentación que ésta tenga, si contiene errores ortográficos el usuario podría optar por cancelar la exploración. La buena ortografía habla bien del diseñador de la página. Por lo que es recomendable dar a leer el texto a otra persona para que ésta detecte posibles errores de redacción y ortografía.

CLARIDAD Y BREVEDAD EN EL TEXTO

Una página Web debe ser prometedora para aquellos que la consulten, por lo que se recomienda ser concisos y directos, debido a que un mensaje efectivo expresa una idea con las palabras adecuadas y con brevedad. Por otra parte, entre más "cargada" sea una página, mayor será su tiempo de despliegue, lo que puede provocar la desesperación del usuario.

Si una página requiere definiciones densas o largas explicaciones, se puede introducir dicha información dentro de archivos externos, los cuales se pueden llamar con un FTP (empleando el URL apropiado), indicando el tamaño de los archivos, esto ayuda al usuario a calcular el tiempo que requiere la transferencia. El usuario puede entonces decidir si realmente desea el archivo, cuándo programar la carga y cuánto espacio mínimo disponible necesita en su disco duro.

Es recomendable que dentro de la página los encabezados y los títulos den una idea concisa del objetivo de la página, ésta debe seguir la regla de brevedad y claridad. En esta parte se puede hacer uso de las herramientas que nos proporciona HTML para resaltar ideas: tipo y tamaño de letra, imágenes (se aplica la frase de que una imagen dice más que mil palabras), listas de elementos las cuales pueden servir para tener un contexto general del contenido, etc.

También se puede hacer uso de listas para realizar menús de las anclas que se hacen dentro de la página.

Se recomienda colocar la información más importante al inicio de los párrafos y organizar la información, para hacer más ligera la exploración de la página.

ESTRUCTURA DE AUTOSOPORTE

La búsqueda de tópicos en Internet puede dar lugar a acceder a un hipertexto en cualquier punto de su estructura, y no precisamente al inicio de ésta. Por lo tanto, no necesariamente alguien que explore la página lo hará en orden progresivo. Entonces, si una página no se sostiene por sí misma y depende de las anteriores, el objetivo inicial del lector al abrirla estará perdido y decidirá buscar otra que satisfaga su necesidad de conocimiento. Lo anterior se puede evitar otorgando independencia a cada parte del hipertexto:

- Utilizar títulos descriptivos y encabezados con la relación que guarda con las otras páginas.
- Si existe una dependencia inevitable, se deben proporcionar los vínculos básicos.
 - Con la página que depende inmediatamente en forma vertical.
 - Con la página principal de la presentación.
- Evitar frases de dependencia a párrafos o páginas anteriores o posteriores, es decir, que si con frases se hace alusión a información de otras páginas sin explicarlo, el usuario se confundirá.

Tomar en cuenta que no todos emplearán el mismo visualizador. Cada visualizador posee características que lo hacen diferente de los demás. Si dentro de una presentación se hace mención de éstas, como formas de navegación, se

estará individualizando la exploración del hipertexto. Por ejemplo, se puede sustituir el “haga click...” por “seleccione”. No hacer mención de los colores ya que no todos los monitores son multicromáticos. Tomar en cuenta que los visualizadores tienen distintas formas de acceder los vínculos, es decir, no se debe hacer mención de las especificaciones gráficas de algún vínculo. También se debe evitar hacer mención de los comandos que posee un visualizador para ejecutar acciones.

Para el acceso de páginas con imágenes se debe tener en cuenta que no todos los monitores tienen la misma resolución y dimensiones.

DISEÑO Y FORMATO DE LA PÁGINA:

La regla básica para un diseño exitoso en una presentación es hacer el diseño tan sencillo como sea posible, reducir la cantidad de elementos (imágenes, encabezados, plecas) y asegurarse de que sólo las cosas más importantes se enfatizan en el formato.

Imágenes

Igual que la carga de texto, el exceso de imágenes dentro de una página toma mucho tiempo en terminar de acceder; tiempo valioso para el usuario, por lo tanto, se debe reducir el uso de imágenes a la necesidad real de ellas dentro de la página. Evite desviar la atención del usuario con imágenes innecesarias.

Considere el tamaño de las imágenes en dimensiones y en variedad de colores, la transferencia de archivos gráficos pequeños es más rápida. Si se considera que un archivo de 20KB tarda varios segundos en desplegarse con un enlace SLIP de 14.4K baudios, al multiplicar este tiempo por cada una de las imágenes que despliega una página se puede tener una idea del tiempo que tomaría el explorarla. Es por ello que se recomiendan 20KB de tamaño máximo para las

imágenes. La cantidad de imágenes en una presentación sigue la regla del cuestionario: preguntarse si realmente son necesarias, si pueden ser sustituidas por texto y qué tanto mejoran el diseño de la página.

Encabezados

Los encabezados ayudan al usuario a llevar un registro de dónde se encuentra dentro del hipertexto, describen cómo se relaciona dicho archivo con todo el documento y proporcionan un ambiente de trabajo más agradable cuando las páginas son fáciles de usar. Los encabezados claros ayudan al usuario a saber qué es lo que está viendo y si están redactados apropiadamente, describen en forma concisa el contenido o la función del archivo.

Agrupación de la información congruente

Cuando se introduce información en una página Web, se necesita alcanzar ciertos niveles mínimos de calidad si desea llamar la atención de la gente. Agrupar la información que guarda cierta relación es una tarea que se realiza tanto en la redacción del documento como en el diseño del mismo. Al agrupar la información congruente bajo un mismo encabezado hace que la misma sea más fácil de rastrear. Si una página de la Web contiene varias secciones con información distinta, se debe buscar la manera de separar visualmente esas secciones por medio de un encabezado o una línea horizontal.

Formato constante

El principal objetivo de hacer esto es que el usuario sepa que aún se encuentra en su documento y no se ha perdido en algún vínculo que encontró en su página.

Logotipo o imágenes que identifiquen su documento. Intente agregarlas en cada una de sus páginas, de preferencia en el mismo lugar. El mismo fondo para todas

las páginas de su documento es un buen tip para dar un formato constante. Si existen vínculos de relación entre las páginas de su hipertexto, procure ponerlos siempre en el mismo sitio. d) Los títulos de las páginas se sugieren anteceditos de una frase consistente.

Vínculos correctos

Verificar si los vínculos son necesarios o únicamente es información redundante o de más, pues cada vínculo debe cumplir un propósito bien definido. Genere vínculos por razones de importancia. Además, si un vínculo no tiene importancia para el contenido que se está tratando sólo servirá para confundir al lector.

- Vínculos de navegación explícitos: Son los que le muestran al lector hacia donde se debe mover desde cualquier página a otra del mismo documento (Atrás, Adelante, etc.).
- Vínculos de navegación implícitos: Indican que al ser seleccionado se obtendrá mayor información sobre el tema que se está tratando, pero no es necesario que las palabras que conforman el texto del mismo lo digan literalmente.
- Vínculos para definiciones de palabra o concepto: Sirven para incluir glosarios, al vincular la primera aparición de una palabra dada dentro del documento a su definición en algún otro lugar de la red.
- Vínculos tangentes o de información relacionada: son útiles cuando cierta parte del texto se aparta del propósito fundamental del documento, sólo es información adicional que el lector decidirá si se consulta o no.

Características de los vínculos

- Texto breve. Ser descriptivo
- Usar menús de vínculos, es decir, organizarlos según el tema del que se trate en forma de lista o algún otro formato breve.
- No usar vínculos ambiguos, se necesita ser conciso en los nombres o texto de los vínculos.
- Evitar la tendencia a crear vínculos con una sola palabra “aquí” resaltada, para describirlo después en algún lugar incierto.
- Se debe estar consciente de que no todos los usuarios tendrán la posibilidad de oprimir o hacer click con el mouse, ya que algunos usuarios estarán empleando un visualizador en modo texto.
- Son de gran importancia los vínculos de navegación: Atrás (back), Home, Adelante (forward). Éstos nos guiaran dentro de la página Web, el uso correcto de estos vínculos producirá un mejor entendimiento, mayor atracción al usuario y sobre todo el cumplimiento del objetivo individual de la página. Así no se ocasiona la pérdida de algunos objetivos o la del interés por parte del usuario al ocuparse en buscar la página que le antecede a la actual, o por ejemplo si no existe un vínculo que lleve a la página HOME después de haber consultado la información tendrá que recorrer las demás páginas verticalmente hasta llegar a la página principal y poder iniciar una búsqueda horizontal.
- Adicionar un vínculo “mailto” para correo electrónico que permita cumplir con uno de los objetivos de la Web: ser interactivo con el creador de la página.

División en las mismas páginas

Esta parte se refiere a la gran ventaja que trae consigo el reducir el número de páginas dentro de su documento, de tal forma que puedan existir ligas a una misma página para representar información distinta. Así es más fácil manejar un único documento y los vínculos que contengan no se romperán si mueve los elementos o renombra los archivos. Una página presentada de esta forma se asemeja más a la estructura de un documento impreso común; si se debe distribuir documentos tanto de forma impresa como en línea.

También existen desventajas por seguir esta metodología, ya que vuelve a aparecer el tiempo de carga que lógicamente se exagerará, además de que se forma una estructura bastante rígida empleando una estructura lineal.

Tipo de Estructuras recomendadas

Estructura lineal de documentos de varias páginas. El usuario avanza o retrocede por los archivos como si estuviera dando vuelta a las páginas de un libro. Un archivo HTML sigue a otro. Cada archivo HTML, contiene un vínculo a los archivos anterior y siguiente. Usted determina el orden en el que se presenta la información.

Estructura no lineal de documentos de varias páginas. En esta estructura el usuario pasa de una página a otra, también determinará el orden en el que desea que se le presente la información, según las posibilidades que el documento le ofrezca.

Las ventajas de crear una estructura no lineal son que los documentos pequeños se cargan con mayor rapidez, mejor rastreo de información para el usuario.

Pero también existen desventajas y éstas pueden ser por ejemplo que es más complicado manejar vínculos externos que internos, y sobre todo si son una gran cantidad, además que demasiados saltos pueden distraer al usuario de su objetivo inicial, como ya se había mencionado.

Rigidez en la presentación:

El tiempo que tarde en desplegarse la información completa de la página es importante tanto para el creador como para el usuario, debido a la diversidad de páginas que se pueden encontrar en la red.

Existen diseños de páginas que se presentan empleando la herramienta FRAMES, lo que toma mayor tiempo de carga. Para esto se sugiere añadir la posibilidad de que el usuario elija la forma en que quiere que se despliegue la página que visita, es decir, dar la posibilidad de que la misma página se presente en una forma menos rígida NOFRAMES, quizás menos atractiva, pero si con mayor facilidad y rapidez en el manejo.

ANEXO 1

CONSIDERACIONES SOBRE LA PAGINA PRINCIPAL WEB DE LA FACULTAD DE INGENIERIA

El Departamento de Información y Estadística tendrá las siguientes funciones en la Web principal de la Facultad de Ingeniería:

- Planear el contenido de la página principal de la Facultad de Ingeniería, así como establecer la estructura de las páginas.
- Dar de alta las ligas de las páginas que tengan los Vo.Bo. de la Secretaria General correspondientes en las que se incluyen las de las divisiones,

profesores, asociaciones, tesis, alumnos etc.; y cualquier página que se desee depender de la página principal.

- Dar la ubicación de las nuevas páginas dentro de la estructura de la página principal.
- Actualizar y mantener el contenido de las páginas de la Web principal de la Facultad de Ingeniería.
- Administrar las cuentas de correo y la Web de *faiinge* y *webmaster* (faiinge@cancun.fi-a.unam.mx y webmaster@cancun.fi-a.unam.mx)
- Deshabilitar las páginas que rompan con la normatividad de la Web.

La Unidad de Servicios de Cómputo Académico tendrá las siguientes funciones en la Web principal de la Facultad de Ingeniería:

- Mantener la operación Física del servidor de la Web.
- Instalar, mantener y actualizar el sistema operativo del servidor
- Instalar, mantener y actualizar el software de administración de la Web.
- Implementar la Seguridad del Sitio Web y el Sistema Operativo, así como la revisión de bitácoras.
- Instalar las aplicaciones requeridas para la operación del sitio Web.
- Realizar los respaldos del servidor Web.
- Asesorar técnicamente a las áreas de la Facultad en tópicos relacionados con la Web.

- Proponer nuevas innovaciones de tecnología para crear servicios de la Web para el beneficio de la Facultad (como el caso de la Biblioteca Digital, Búsqueda, Bolsa de trabajo etc.)
- Actualizar y mantener el área de la Web correspondiente a UNICA.

La Coordinación de Comunicación tendrá las siguientes funciones en la Web principal de la Facultad de Ingeniería:

- Diseñar la imagen gráfica institucional.
- Crear, diseñar, recabar las imágenes como fotos, backgrounds, íconos etc que se usarán en el sitio de la Web.

ANEXO 2

PROCEDIMIENTO PARA DAR DE ALTA PAGINAS EN SITIO WEB PRINCIPAL DE LA FACULTAD DE INGENIERIA

El contenido de las páginas deberá ser académico o de aplicación administrativa y con objetivos afines a los de la Facultad (docencia, investigación y difusión de la cultura).

En todos los casos, la creación, mantenimiento y diseño de la página serán responsabilidad del solicitante. No se darán de alta páginas que este bajo construcción.

Para dar de alta páginas dependientes de la página principal deberá seguirse el siguiente procedimiento:

El solicitante deberá enviar un oficio dirigido a la Secretaria General donde indicará:

- Objetivos de la página.
- Breve descripción de la página
- Responsable de la página (nombre, teléfono y correo electrónico)

La página debe estar terminada por completo en caso de que ya cuente con un sitio de Web.

En el caso del personal académico:

Si la página de la Web es personal, deberá tener la siguiente leyenda al Inicio:

“El contenido de esta página es responsabilidad del Autor”

Autor:

Cargo- División o área:

Las páginas podrán contener tareas, apuntes personales, ejercicios, ligas de apoyo a la materia o cualquier otra actividad que apoye su actividad de académico.

Las páginas deberán ser alojadas primeramente en los servidores de la división al cual pertenece el académico.

En caso de que la División o Secretaría no contase con la infraestructura necesaria, el solicitante podrá manifestar el requerimiento de una cuenta para el alojamiento de su página.

Las cuentas estarán sujetas de acuerdo a los recursos e infraestructura con que cuente la Facultad y a sus condiciones de operación. La cuenta será intransferible y deberá apegarse a la normatividad de la Web vigente.

El archivo principal de la cuenta deberá llamarse index.html y deberá estar ubicado en el subdirectorío WWW en su directorío casa (home directory).

La página deberá renovarse al inicio de cada año escolar.

ANEXO 3

LOS ADMINISTRADORES DE LA WEB

Este apartado se refiere a aquellas personas que coordinan la publicación de páginas en la Web, conocidas como administradores de la Web o webmasters.

Para una mejor administración de la Web en la Facultad de Ingeniería, se creó el subcomité de webmasters, el cual es un grupo de administradores de la Web, cada elemento integrante representando a una Secretaría o División. Este subcomité del Comité Asesor de Cómputo es el encargado de apoyar en los temas relacionados con la estructuración, presentación y contenido de las páginas de la Facultad, y tiene como objetivos, los siguientes:

- Llevar a cabo la normalización de la administración de la Web en la Facultad de Ingeniería.

- Participar en la normatividad de la Web.
- Administrar de manera óptima los recursos que tiene la Facultad destinados a la Web.
- Control de la información que se publicará en la Web.

El subcomité de webmasters estará presidido por el webmaster de la Secretaría General, quien fungirá como moderador de las juntas y tomará nota de los acuerdos a los que se llegue. Estos acuerdos se tomarán como base para la creación de todas las páginas que dependan de la Facultad de Ingeniería.

Entre algunas de las funciones que tienen los webmaster, se encuentran:

- Creación, mantenimiento y control de las páginas del área a la cual pertenecen.
- Asistir a las juntas de webmasters, las cuales serán al menos una vez por mes.
- Estar pendiente de la información e instrucciones de las áreas correspondientes a efecto de incluir, modificar y borrar oportunamente la página correspondiente.
- Responder los correos que les sean enviados, relacionados con el área a la cual pertenezcan. En caso de que no tuviesen la información que se les solicita, indicarán el nombre de alguna persona que pueda resolver la duda.
- Calificar técnicamente y en su caso dar de alta las páginas de los usuarios.

- Hacer del conocimiento de los usuarios el reglamento que deben cumplir para poder tener derecho a la publicación de sus páginas.
- Revisar periódicamente las páginas de los usuarios en su área, con la facultad de desactivar aquellas que no cumplan con los lineamientos del presente documento.
- El webmaster de la Secretaría General podrá reconvenir sobre la estructura y contenidos de las páginas, siempre que éstas no se apeguen a los lineamientos que se indiquen en el presente documento. Tiene además la facultad de deshabilitar páginas que no cumplan con los lineamientos establecidos.

Índice de figuras

Fig 1. Tecnologías agrupadas bajo el concepto de AJAXs, tomado de:
<http://intellez.wordpress.com/2008/09/01/ajax/> [consultado marzo 2012]

Fig. 2. Los tres niveles de abstracción de datos, tomado de:
<http://tesciinfbasededatos.blogspot.mx/2011/08/niveles-de-abstraccion.html>
[consultado marzo 2013]

Fig. 3. Esquema de Fase del modelo de Cascada, tomado de:
<http://spanishpmo.com/index.php/ciclos-de-vida-modelo-de-cascada/>
[consultado abril 2012]

Fig 4. Fase del Modelo Incremental, tomado de:
<http://heysellopez.blogspot.mx/2010/10/modelos-de-desarrollo-de-software.html>
[consultado abril 2012]

Fig. 5. Esquema de las fases del Modelo en Espiral, tomado de:
<http://srivera334.blogspot.es/1193608080/> [consultado mayo 2012]

Fig. 6. Esquema de las fases del Proceso Unificado, tomado de:
<http://xherrera334.blogspot.es/> [consultado mayo 2012]

Fig. 7. Ciclo de vida eXtreme Programming, tomado de:
http://ingsoftware072301.obolog.com/fotos-articulo_metodologia-xp-2012877
[consultado abril 2012]