
# UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

## FACULTAD DE INGENIERÍA

**“Portal COPADI”**

# T E S I S

**Q U E P A R A O B T E N E R E L T Í T U L O D E**

**I N G E N I E R O EN COMPUTACIÓN**

**P R E S E N T A :**

**E N R I Q U E F E L I P E A N A S T A C I O**


**DIRECTORA DE TESIS:**

**M. EN A. MARÍA DE LOURDES CAMPOS LUNA**

**MÉXICO, D.F.** **2012**

## Dedicatorias y agradecimientos

*Son muchas las personas especiales a las que me gustaría agradecer su amistad, apoyo, animo y compañía en las diferentes etapas de mi vida. Algunas están aquí conmigo y otras en mis recuerdos y en el corazón. Sin importar en donde estén o si alguna vez llegan a leer estas dedicatorias quiero darles las gracias por formar parte de mí, por todo lo que me han brindado y por todas sus bendiciones.*

Mamá, no me equivoco si digo que eres la mejor mamá del mundo, gracias por todo tu esfuerzo, tu apoyo y por la confianza que depositas en mi. Gracias porque siempre estas a mi lado, por la paciencia que has tenido para enseñarme, por el amor que me das, por tus cuidados en todos estos años que hemos vivido juntos, por los regaños que me merecía y que no entendía.
***¡Gracias por darme la vida!***
***¡Te quiero mucho!***

Papá, éste es un logro que quiero compartir contigo, gracias por ser mi papá, por confiar, apoyarme y creer en mi. Gracias por enseñarme todo el valor y toda la fuerza en un sólo abrazo. Por los ejemplos de perseverancia y constancia que te caracterizan y que me has infundado siempre, por el valor mostrado para salir adelante y por tu amor.

A mis hermanos, Lalo, Lumi, Humbe y Emi, que con su cariño me han enseñado a salir adelante. Gracias por su paciencia, gracias por preocuparse por su hermano mayor, gracias por compartir sus vidas, pero sobre todo, gracias por estar en otro momento tan importante en mi vida.

A mis abuelitos, por ser un ejemplo para salir adelante y por los consejos que han sido de gran ayuda para mi vida y crecimiento. Gracias por llevarme en sus oraciones porque se que siempre lo hacen.

A mi niña hermosa Kary, gracias por permitirme formar parte de tu vida, gracias por tu amor, gracias por ser como eres, gracias por ser la mujer con los mejores sentimientos que he conocido, gracias por ayudarme a terminar este trabajo, gracias por aguantarme, pero sobre todo gracias por motivarme a hacer las cosas de la mejor manera.

A todos mis amigos, sin excluir a ninguno, mil gracias por todos los momentos que hemos pasado juntos y porque han estado conmigo aunque sea sólo para dar lata y molestar ☺.

A todos mis *profes,* no sólo de la carrera sino de toda la vida, mil gracias porque de alguna manera forman parte de lo que ahora soy. Especialmente a Lulú quién no sólo fue mi directora de tesis, sino también mi jefa y tutora (aunque sólo fuera 1 sesión de tutoría ☺).

A mis sinodales, gracias por darme la oportunidad y por el tiempo que me han dedicado para leer este trabajo.

A la Universidad Nacional Autónoma de México y en especial a la Facultad de Ingeniería, por permitirme ser parte de una generación de triunfadores y gente productiva para el país.

*Enrique*

# ÍNDICE

# CAPÍTULO I

## "Introducción"

### 1.1 Planteamiento del Problema.

La Facultad de Ingeniería como impulsora en el uso e implementación de Sistemas de Apoyo Académico-Docente, con el propósito de seguir a la vanguardia, ofrece atender de forma diferenciada la problemática grupal e individual de los alumnos.

Dado que actualmente, la facultad tiene un ingreso anual aproximado de 2500 alumnos, y 120 tutores contemplados, tomado en consideración que el Sistema de Tutoría pretende llevar el seguimiento de las generaciones de alumnos, nos da un panorama concreto de la gran cantidad de información que debe ser manejada.

Con el actual sistema (TutorNet) con que cuenta la coordinación, se presenta una gran limitación en cuanto a su uso, esto es que el sistema solo está disponible de manera local, lo que implica que los usuarios tengan que acceder a él únicamente desde la sala de computo de la Coordinación de Atención Diferenciada para Alumnos (COPADI). Así mismo, la única información que se puede brindar al tutor es la de la generación en curso, lo cual impide consultar generaciones pasadas. Así como la carencia de información de los estudiantes, y una forma poco práctica de obtener los datos.

Tomando en cuenta lo descrito anteriormente, el sistema se muestra bastante limitado y presenta un problema para que estudiantes como tutores hagan uso de él.

Para facilitar tal manejo, se pretende sistematizar dicha información con un Sistema Web (TutorFI), que sea capaz de albergar la información ingresada año con año.

### 1.2 Objetivo

Diseñar e implementar un portal que será una herramienta que fortalecerá las ventajas, aportaciones y servicios que ofrece la Coordinación de Programas de Atención Diferenciada para Alumnos (COPADI), así como fortalecer el Sistema de tutorías de la Facultad de Ingeniería con un Sistema Web (TutorFI), con el objetivo de brindar apoyo a los tutores y ofrecer información relevante al estudiante. Todo con una amplia disponibilidad al portal desde cualquier ubicación, además de

contar con información actualizada de los cursos, eventos e información de los estudiantes.

### 1.3 Justificación

Para poder llevar a cabo el seguimiento de tutoría a los estudiantes (que actualmente no se puede llevar a cabo), es necesario que el sistema esté disponible en cualquier momento y que se pueda acceder a él desde cualquier lugar. Así mismo proporcionar al tutor mayor información actualizada de cada uno de los estudiantes y que dicha información sea de fácil acceso y manejo.

### 1.4 Delimitaciones

Para el nuevo sistema (TutorFI), además de incluir las funcionalidades del TutorNet, se presentara al tutor información detallada de las calificaciones del primer semestre, resultados del examen diagnóstico, avance y promedio actual del alumno. La información podrá ser consultada en línea, así mismo tendrá la opción de imprimir o descargar la información requerida.

El tutor podrá ubicar más fácilmente al alumno, ya que este nuevo sistema incluirá una foto que se presentara al inicio de la información de cada uno de los estudiantes.

Se realizaran las evaluaciones grupales, semestrales e individuales, así como la consulta a tal información. Se podrá consultar información de asesorías, talleres y comunicados que tenga la coordinación para la comunidad de la Facultad.

Estará disponible material que podrá ser descarga por los tutores como ayuda para la realización de las tutorías para/con los alumnos.

Los coordinadores tendrán la posibilidad de consultar la información de los tutores que corresponden a su área así como compartir documentos y visualizar la foto de cada tutor.

La programación del sistema será modulada para reutilización del código y permita el crecimiento del mismo para nuevos apartados que pueda llegar a necesitar la coordinación, así como llevar el control de registro de los cursos extracurriculares intersemestrales. Así mismo permitirá de una manera más sencilla alguna modificación o actualización tanto al TutorFI como a la página principal.

# CAPÍTULO ll

# "MARCO TEÓRICO"

## 2.1 Antecedentes.

Como todo proyecto a realizar, es necesario conocer las diferentes estrategias y herramientas necesarias para llevar a cabo tal desarrollo.

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

## 2.2 La red mundial.

La Red Mundial o como se le conoce normalmente WWW (World Wide Web), se define simplemente como el universo de redes de información de acceso global. Se trata de un espacio abstracto en el que las personas pueden interactuar y está poblado de páginas con texto, imágenes y animaciones vinculadas entre sí, ocasionalmente con sonidos y videos (Berners-Lee, 1996).

Fue creada por Tim Berners-Lee, del Centro Europeo de Física Nuclear (CERN), con el objetivo de servir como herramienta para la búsqueda y transmisión de información entre los científicos.

El hipertexto (un hipertexto es un documento digital que se puede leer de manera no secuencia; tiene los siguientes elementos: secciones, enlaces y anclajes. Las secciones o nodos son los componentes del hipertexto o hiperdocumento. Los enlaces son las uniones entre nodos que facilitan la lectura secuencial del documento. Los anclajes son los puntos de unión entre nodos), es, la base funcional y estructural de la World Wide Web. La Web es un sistema hipertextual preparado para recorrer diferentes páginas web dispuestas en servidores accesibles desde cualquier ordenador conectado a internet y enlazadas unas con otras, conformando una estructura similar a la tela de araña. Las páginas web se enlazan unas a otras dentro de cada hiperdocumento o sitio Web y pueden conectarse a otros sitios Web

llevando al usuario de un servidor a otro sin necesidad de teclear ninguna ruta (Lamarca, 2005).

En la Web, cada nodo es una página, y cada palabra remarcada (puede ser con determinado color o subrayada) representa la entrada de un enlace. La Web también permite relacionar documentos multimedia (imágenes, sonido, video, animaciones) y recursos residentes en múltiples servidores mundiales, y ofrece un nuevo y más extenso medio para estudiar las consecuencias del hipertexto (Lamarca, 2005).

La Web se ha convertido en uno de los servicios principales de Internet. En pocos años, casi toda la información disponible en la red se ha volcado a la Web y se han ido abandonando otros métodos.

La WWW (World Wide Web) ha triunfado y millones de documentos se encuentran accesibles mediante este sistema de almacenamiento y acceso a la información.

La Web marcó el fin de una era en la que la incompatibilidad entre computadoras debilitaba los sistemas. Esto creó una explosión de accesibilidad, con muchos impactos potenciales en lo económico, lo social y en lo político. La Web ha sido designada como el espacio ideal en el que las personas pueden trabajar en un proyecto determinado. La potencia de este concepto radica en que:

- Las personas pueden unir proyectos de equipo y pueden tener acceso al historial de las actividades del equipo para la toma de decisiones,
- El trabajo de las personas que deja un equipo puede ser capturado para futuras referencias y,
- Las operaciones de equipo, si se encuentran en la Web, pueden ser analizadas de otra manera por otro equipo.

La Web, originalmente se pensó para sistemas de información personales y como herramienta para grupos de todos los tamaños o para el mundo entero. Las personas rápidamente han desarrollado nuevas características para la Web lo que ha construido un enorme potencial comercial. Esto ha hecho posible el mantenimiento global de la Web y la interoperabilidad se ha convertido en una tarea continua. También se han creado nuevas áreas en aspectos de investigación.

### 2.2.1 Antes de la Web.

El concepto de hipertexto fue acuñado en 194 5 en un trabajo de Vannevar Bush titulado "Cómo nosotros podemos pensar (As We May Think)". En este artículo, Bush propone la máquina Memex (MEmory Extended System), la cual podía usar código binario, fotoceldas y fotografías instantáneas, y permitía a las personas hacer y seguir referencias de documentos como microfichas. Este concepto continúo hasta mediados de 1960, cuando Douglas Englebart desarrollo el NLS (oNLine System), que usaba una computadora digital que proporcionaba hipertexto a los e-mails y algunos documentos. Mientras tanto, en 1965, Ted Nelson acuña la palabra hipertexto (Lamarca, 2005).

Sin embargo, en 1980, el mundo aún padecía de incompatibilidad en las redes, en los formatos de los discos, en el formato de los datos y en los esquemas de codificación de caracteres. Se intentó transferir una cantidad impresionante de información sin ningún éxito. Esto provocaba frustración porque cada vez se incrementaba más el número de personas que usaban las computadoras para manejar información. Además, una gran cantidad de información se encontraba almacenada en las computadoras y muchas de ellas estaban conectadas en red.

### 2.2.2 Objetivos de la Web.

El mayor objetivo que tenía la Web era lograr que las personas y las máquinas pudieran comunicarse y compartir información. Era deseable que la interacción entre las personas y el hipertexto fuera intuitivo y que las máquinas leyeran información y dieran representaciones exactas del estado de la interacción entre las personas y los patrones de trabajo. Las máquinas podrán analizar la información y convertirse en potentes herramientas, capaces de resolver distintos tipos de problemas para las grandes organizaciones. La Web está diseñada bajo los siguientes criterios:

- 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 la arquitectura.

La arquitectura de la Web, mostrada en la figura 1, se propuso en 1989. En este diseño encontramos en la parte superior los criterios y enseguida se establecen los principios del diseño de software adaptados al entorno de red.

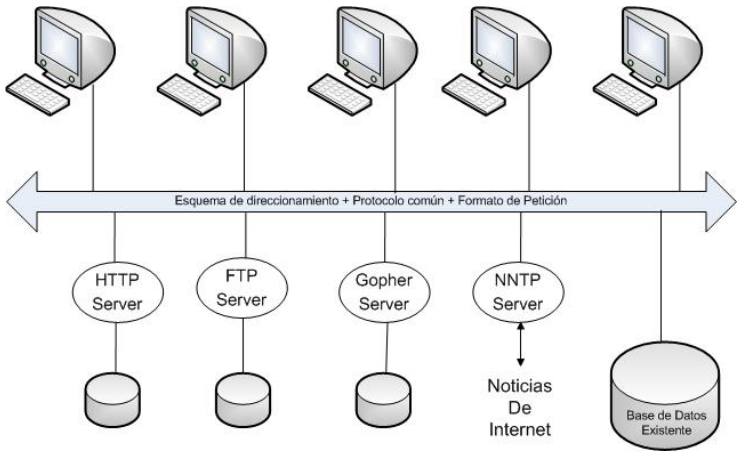



Figura 1.

Figura 1. Diagrama de la arquitectura Web. Esta arquitectura permite a los usuarios leer y manipular información de una gran variedad de clientes. Esta interoperatibilidad requiere estándares comunes, representados en la figura por la flecha: el esquema de direcciones de los identificadores Universales de Recursos, un protocolo como HTTP que negocie la transferencia de acuerdo a un formato, que puede ser HTML.

#### 2.2.3.1 Independencia de las especificaciones.

La flexibilidad es claramente una de las metas principales. Es necesario tener varias especificaciones para poder asegurar la interoperatibilidad y los límites de la implementación de la Web. Por consiguiente, deben existir unas pocas especificaciones posibles (*Principio de mínimas limitaciones*), que son necesarias para lograr independencia (*Principio de modularidad y de información oculta*). Esto permitirá que sean reemplazadas partes del diseño mientras se preserva la

arquitectura básica.

De esta manera, será posible mezclar viejas y nuevas especificaciones. Por ejemplo, el viejo protocolo FTP puede ser mezclado con el nuevo protocolo HTTP en los espacios de direcciones, los textos convencionales pueden ser mezclados con los nuevos documentos de hipertexto. El principio de mínimas limitaciones es uno de los usos más importantes factores que ha adoptado la Web. Las personas frecuentemente incrementan los cambios para adoptar la web, primero como tecnología paralela que existe en los sistemas y después como principal tecnología.

#### 2.2.3.1.1 Identificadores Universales de Recursos (Uniform Resource Identifier - URI).

Típicamente, los sistemas de hipertexto se construyen alrededor de bases de datos vinculadas, esto es, sistemas de información centralizados, poco manejables cuando son escalados. Sin embargo, se garantiza la consistencia, es decir, cuando un vínculo es removido es porque el documento que vinculaba ha sido eliminado. El remover los vínculos garantiza el principal compromiso hecho en la arquitectura de la Web.

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 Universales de Recursos (IUR) 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.

Los IUR son un texto corto que identifica unívocamente cualquier recurso (servicio, página, documento, dirección de correo electrónico, enciclopedia, etc.) accesible en una red (Wikipedia, 2010).

Normalmente un UIR consta de dos partes:

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

La especificación detallada se encuentra en la RFC 2396 (http://www.ietf.org/rfc/rfc2396.txt) - Uniform Resource Identifiers (URI): Generic Syntax.

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.

Afortunadamente, la arquitectura de la Web no depende de que la IUR sea un nombre o una dirección, aunque la palabra Localizador Uniforme de Recursos (LUR, Uniform Resource Locator - URL) haya sido acuñada por Intenet Engineering Task Force e indique que la mayoría de los IUR están vinculados con más direcciones, más que con nombres.

#### 2.2.3.1.2 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 internet. Existe un URL único para cada página de cada uno de los documentos de la WWW (World Wide Web), para todos los elementos del Gopher y todos los grupos de debate UNESET, y así sucesivamente.

El URL de un recurso de información es su dirección de internet, la que permite que el navegador la encuentre y la muestre de forma adecuada. Por ellos el URL combina el nombre del ordenador que proporciona la información, el directorio donde se encuentra, el nombre del fichero y el protocolo a usar para recuperar los datos, y reemplaza la dirección numérica o IP de los servidores; lo que hace más fácil la navegación, si no, de otra forma se tendría que hacer bajo direcciones del tipo http://192.168.1.10 en vez de http://www.pagina.com

El formato general de un URL es:

Protocolo://máquina/directorio/fichero

También pueden añadirse otros datos:

Protocolo://usuario:contraseña@máquina:puerto/directorio/fichero

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

#### 2.2.3.1.3 HTTP (HiperText Transfer Protocol).

HTTP es un protocolo del nivel de aplicación para sistemas de información multimedia distribuidos. Es un protocolo no orientado a estado, que puede ser utilizado para más propósitos que para manejar ficheros HTML. La figura 2 muestra un esquema de la arquitectura cliente-servidor.

Entre las propiedades de HTTP se puede destacar las siguientes (Martínez y Martínez, 2004):

- Posee un esquema de direccionamiento comprensible. Utiliza los Identificadores Universales de Recursos (URI) para localizar sitios (URL) o nombres (URN) sobre los que hay que aplicar un método. La forma general de un URL es servicio://host/fichero.ext

- Arquitectura Cliente-Servidor. HTTP se asienta en el paradigma solicitud/respuesta. La comunicación se asienta sobre TCP/IP. El puerto por defecto es el 80, pero se pueden utilizar otros como el 8080 usado por Tomcat.

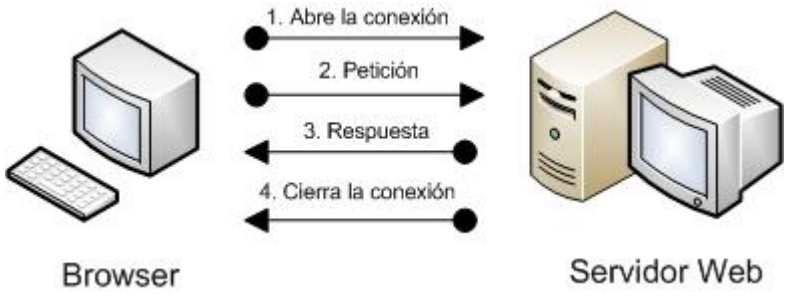



Figura 2. Arquitectura Cliente-Servidor.

- Es un protocolo sin conexión y sin estado. Después de que el servidor ha respondido la petición del cliente, se rompe la conexión entre ambos. Además no se guarda memoria del contexto de la conexión para siguientes conexiones.

- Está abierto a nuevos tipos de datos. HTTP utiliza tipos MIME (Multipart Internet Mail Extension) para la determinación del tipo de los datos que transporta. Cuando un servidor HTTP transmite información de vuelta a un cliente, incluye una cabecera que le indica al cliente sobre los tipos de datos que compone el documento. De la gestión de esos datos se encargan las utilidades que tenga el cliente (visor de imágenes, de video, etc.).

- Una transacción HTTP está compuesta por una cabecera y, opcionalmente, por una línea en blanco seguida de los datos. En la cabecera se especifica tanto la acción solicitada en el servidor, como los tipos de datos devueltos o un código de estado.

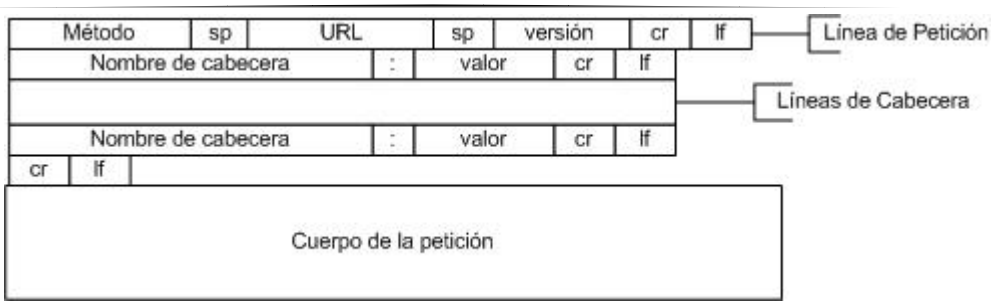
- Mensaje HTTP de petición (Figura 3). Constan de tres partes:

  1ª. Parte. Línea de Petición.

  - Método HTTP Identificador Versión HTTP
    - Método HTTP: acción que el cliente solicita. Puede ser: GET, POST, HEAD, DELETE, PUT.
    - Identificador: URL sin protocolo ni nombre del servidor.
    - Versión HTTP: suele ser: HTTP/1.0 ó HTTP/1.1

  2ª Parte. Líneas de cabecera de petición.
  Informan al servidor sobre el cliente y lo que solicita.

  3ª Parte. Cuerpo de la petición.
  Datos transferidos del cliente al servidor.




```
GET /servicios/prueba.html  HTTP/1.1
Host: ants.dif.um.es
Connection: close
User-agent: Mozilla/4.06
Accept-language: es
Accept: */html
Accept: */text
[Una línea en blanco, contenido solo CRLF]
```

Figura 3. Formato general de una petición.

Campos de cabecera de petición comunes:

- User-Agent: informa del tipo de navegador que emplea el cliente.
- Referer: URL del documento desde el que se accedió al actual.
- If-Modified-Since: sólo se quiere el documento si se ha modificado.
- Accept-Language: lenguajes aceptados por el navegador.
- Host: servidor al que el cliente envía la petición.
- Connection: se emplea para mantener la conexión.
- Accept: tipos de datos aceptados por el cliente.

- Mensaje de respuesta (Figura 4). Consta también de tres partes:

  1ª Parte. Línea de estado

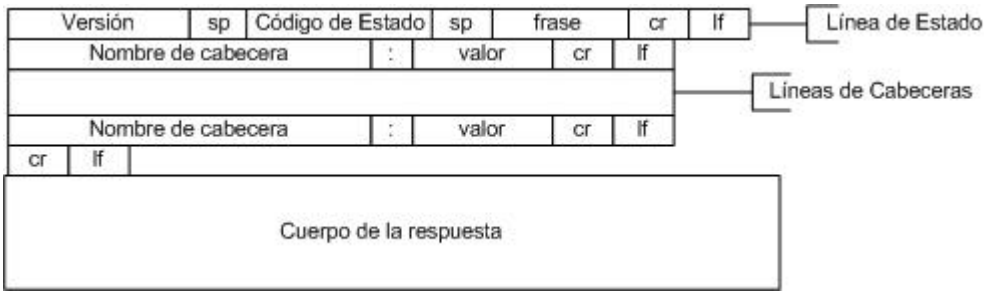
  - Versión HTTP Código Estado Explicación
    - VersiónHTTP: versión del producto: HTTP/1.0 ó HTTP/1.1
    - Código Estado: número entre 200 y 599
      - 2xx Éxito
      - 3xx Redireccionamiento
      - 4xx Error del Cliente
      - 5xx Error del Servidor
    - Explicación: información descriptiva del estado

2ª Parte. Líneas de cabeceras de respuesta:

Informan al cliente sobre el servidor y el recurso consultado.

3ª Parte. Cuerpo de la respuesta

Datos enviados del servidor al cliente.




```
GET /servicios/prueba.html  HTTP/1.1
Host: ants.dif.um.es
Connection: close
User-agent: Mozilla/4.06
Accept-language: es
Accept: */html
Accept: */text
[Una línea en blanco, contenido solo CRLF]
```

Figura 4. Formato general de una respuesta.

Campos de cabecera de respuesta comunes:

- Location: nueva localización del documento solicitado.
- Server: nombre y versión del software servidor.
- Date: día y hora en la que se transmite el documento.
- MIME-version: versión del protocolo MIME usado por el servidor.
- Content-Length: longitud en bytes del cuerpo de la respuesta.
- Content-Type: tipo MIME que identifica el tipo de dato de la respuesta.
- Last-Modified: fecha y hora en la que se modificó por última vez.

#### 2.2.3.1.4 HTML (HyperText Markup Language).

HTML, acrónimo inglés de HyperText Markup Language (lenguaje de marcado de documentos de hipertexto). Es un lenguaje de marcas diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas Web.

El lenguaje de marcado (markup language), también denominado lenguaje de anotaciones o de etiquetas, se define como un conjunto de reglas para estructurar y dar formato a un documento electrónico. Suelen utilizar etiquetas para definir el inicio y el final de un elemento: un párrafo, un título, un elemento subrayado, etc. Los lenguajes de marcas más utilizados son HTML y XML (eXtensible Markup Language – Lenguaje de Marcado Ampliable o Extensible), ambos basados en el metalenguaje SGML (Standard Generalized Markup Language).

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

1. Separa un texto en sus elementos, como por ejemplo un párrafo, un capítulo, etc.
2. Especifica las operaciones tipográficas y funciones que debe ejecutar el programa visualizador sobre dichos elementos. Las operaciones tipográficas de formato que se aplican a cada uno de los elementos de un documento digital, por ejemplo, imprimir un título en itálicas.

Gracias al internet y a los navegadores del tipo Explorer, Mozilla Firefox, Opera, Safari, Google Chrome o Netsacape, el HTML se ha convertido en uno de los formatos más populares que existen para la construcción de documentos.

HTML es hijo de SGML, aunque hay unas versiones de XHTML (eXtensible HyperText Markup Language – Lenguaje Extensible de Marcado de Hipertexto) que son descendientes de XML y exigen que se describa mucho más, para facilitar la vida a los navegadores, que son aquellos programas que nos muestran la información en la pantalla.

#### 2.2.3.1.4.1 Nociones básicas de HTML.

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.

Toda etiqueta se identifica porque está encerrada entre los signos menor que y mayor que (<>), y algunas tienen atributos que pueden tomar algún valor. En general las etiquetas se aplicaran de tres formas especiales:

- Se abren y se cierran, como por ejemplo: negrita que se vería en el navegador "negrita" en negrita.
- No pueden abrirse y cerrarse como <hr> que se vería como una línea horizontal en el navegador.
- Otras que pueden abrirse y cerrarse como por ejemplo <p>

Las etiquetas básicas o mínimas son:

```
<html>
        <head>
                <title>Ejemplo</title>
        </head>
        <body>
                        <title>Ejemplo</title>
        </body>
</html>
```

**2.2.3.1.4.2 Algunas etiquetas básicas de HTML.**

Las etiquetas básicas de HTML, de obligada presencia en todo documento son:

- <html>: Es la etiqueta que define el inicio del documento HTML, le indica al navegador que todo lo que viene a continuación debe tratarlo como una serie de códigos HTML.

- <head>: Define la cabecera del documento HTML; esta cabecera suele contener información sobre el documento que no se muestra directamente en el navegador. Como por ejemplo el título de la ventana de su navegador. Dentro de la cabecera <head> podemos 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 definir algunas características avanzadas, como por ejemplo las stylesheets (hojas de estilo) usadas para el diseño de la página, ejemplo: <link rel="stylesheet" href="/style.css" type="text/css">

- <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> podemos encontrar numerosas etiquetas. A continuación se indican algunas a modo de ejemplo:

- <h1>, <h2>,… <h6>: encabezados o títulos del documento en diferentes tamaños de fuente.
- <p>: párrafo nuevo.
- <br>: salto de línea forzado.
- <table>: comienzo de una tabla (las filas se identifican con <tr> y las celdas dentro de las filas con <td>).
- <a>: indica la existencia de un hipervínculo o enlace, dentro o fuera de la página Web. Debe definirse el parámetro de pasada por medio del atributo href (ejemplo: <a href="www.google.com">Google</a> se representa como Google)
- <img>: indica la existencia de una imagen para mostrarse en el navegador.

**2.2.3.1.4.3 Historia de estándar.**

No hay especificación oficial del HTML 1.0 porque ya existían múltiples estándares informales del HTML cuando se decidió crear un estándar oficial. Los trabajos para crear un sucesor del HTML, posteriormente llamado HTML+ comenzaron a finales de 1993. El HTML+ se diseñó originalmente para ser un superconjunto del HTML que permitiera evolucionar gradualmente desde el formato HTML anterior. A la primera especificación formal de HTML+ se lo dio, por lo tanto, el número de versión 2.0 para distinguirla de esos estándares no oficiales previos. Los trabajos sobre HTML+ continuaron, pero nunca se convirtió en un estándar.

El borrador del estándar HTML 3.0 fue propuesto por el recién formado World Wide Web (W3C) en marzo de 1995. Con él se introdujeron muchas nuevas capacidades, tales como facilidades para crear tablas, hacer que el teto fluyese alrededor de las figuras, mostrar elementos matemáticos complejos. Aunque se diseñó para ser compatibles con HTML 2.0, era demasiado complejo para ser implementado con la tecnología de la época y, cuando el borrador del estándar expiró en septiembre de 1995, se abandonó debido a la carencia de apoyos de los fabricantes de navegadores Web. El HTML 3.1 nunca llego a ser propuesto oficialmente, y el estándar siguiente fue el HTML 3.2, que abandonaba la mayoría

de las nuevas características del HTML 3.0 y, a cambio, adoptaba muchos elementos desarrollados inicialmente por los navegadores Web Netscape y Mosaic.

La posibilidad de trabajar con fórmulas matemáticas que se había propuesto en el HTML 3.0 pasó a quedar integrada en un estándar distinto llamado MathML. El HTML 4.0 también adopto muchos elementos específicos desarrollados inicialmente para un navegador Web concreto, pero al mismo tiempo comenzó a depurar el HTML señalando algunos de ellos como desaprobados.

HTML 5 establece una serie de nuevos elementos y atributos que reflejan el uso típico de los sitios web modernos. Algunos de ellos son técnicamente similares a las etiquetas  y , pero tienen un significado semántico, como por ejemplo <nav> (bloque de navegación del sitio web) y <footer>. Otros elementos proporcionan nuevas funcionalidades a través de una interfaz estandarizada, como los elementos <audio> y <video>. Mejoras en el elemento <canvas>.

Algunos elementos de HTML 4.01 han quedado obsoletos, incluyendo elementos puramente de presentación, como <font> y , cuyos efectos son manejados por el CSS. También hay un renovado énfasis en la importancia del scripting DOM para el comportamiento de la web. (Wikipedia, 2010).

### 2.2.4 Evolución de la WWW

La web nació alrededor de 1989, a partir de un proyecto del Centro Europeo para la Investigación Nuclear (CERN), en el que Tim Berners-Lee construyó el prototipo que dio lugar al núcleo de lo que hoy es la World Wide Web. La intención original era hacer más fácil el compartir textos de investigación entre científicos y permitir al lector revisar las referencias de un artículo mientras lo fuera leyendo. El nombre original del prototipo era Enquire Within Upon Everything (Lamarca, 2005).

La funcionalidad elemental de la Web se basa en tres estándares: El Localizador Uniforme de Recursos (URL), que especifica cómo, a cada página de información, se asocia una dirección única donde encontrarla; el Protocolo de Transferencia de HipeTexto (HTTP), que especifica como el navegador y el servidor intercambian información en forma de peticiones y respuestas, y el Lenguaje de Marcación de

Hipertexto (HTML), un método para codificar la información de los documentos y sus enlaces. Berners-Lee dirige en la actualidad el World Wide Web Consortium, que desarrollo y mantiene éstos y otros estándares que permiten a los ordenadores de la Web almacenar y comunicar todo tipo de información (Lamarca, 2005; W3C, 2010).

El programa inicial de CERN, WWW, solo presentaba texto, pero navegadores Web posteriores, como Viola de Pei Wei (1992), añadieron la capacidad de presentar también gráficos, Marc Andreesen del Centro Nacional de Aplicaciones de Supercómputo (NCSA) presentó un navegador Web llamado Mosaic para X en 1993, que disparó la popularidad de la Web entre principiantes. Andreesen fundó Mosaic Communication Corparation (hoy Netscape Communications), añadiendo características adicionales como contenido dinámico, música y animación, que están incluidas en los modernos navegadores. A menudo la capacidad de los navegadores y servidores avanza mucho más rápido que los estándares, con lo cual es habitual que las características más nuevas no funcionen en todas las máquinas, impidiendo la accesibilidad universal (Lamarca, 2005).

El imparable avance de la WWW permite hoy incluso servicios en tiempo real como Webcasts, radio Web y webcams en directo.

Una de las tecnologías en constante evolución en el mundo de la WWW es Flash, un formato registrado por la compañía Macromedia, que aporta un gran dinamismo a la Web. El lenguaje de scripting que usa, ActionScript, goza de un gran potencial que abarca desde la aplicación visual hasta la interactividad con el servidor.

Otra de las tecnologías implementadas, es la plataforma Java, de Sun Microsystems ahora perteneciente a ORACLE, que permitió a las páginas Web incluir pequeños programas (llamados applets) que se ejecutan en la máquina del cliente y mejoran la presentación y la interactividad.

Entre las más sobresalientes de las tecnologías implementadas, tenemos a **JavaScript.** Es un lenguaje de scripting basado en objetos sin tipo y liviano, utilizado para acceder a objetos en aplicaciones. Principalmente, se utiliza integrado en un navegador web permitiendo el desarrollo de interfaces de usuario

mejoradas y páginas web dinámicas. JavaScript es un dialecto de ECMAScript y se caracteriza por ser un lenguaje basado en prototipos, con entrada dinámica y con funciones de primera clase. JavaScript ha tenido influencia de múltiples lenguajes y se diseñó con una sintaxis similar al lenguaje de programación Java, aunque más fácil de utilizar para personas que no programan.

Todos los navegadores modernos interpretan el código JavaScript integrado dentro de las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM).

El lenguaje fue inventado por Brendan Eich en la empresa Netscape Communications, la que desarrolló los primeros navegadores web comerciales. Apareció por primera vez en el producto de Netscape llamado Netscape Navigator 2.0.

Tradicionalmente, se venía utilizando en páginas web HTML, para realizar operaciones y en el marco de la aplicación cliente, sin acceso a funciones del servidor. JavaScript se ejecuta en el agente de usuario, al mismo tiempo que las sentencias van descargándose junto con el código HTML.

Inicialmente los autores lo llamaron Mocha y más tarde LiveScript pero fue rebautizado como JavaScript en un anuncio conjunto entre Sun Microsystems y Netscape, el 4 de diciembre de 1995.

En 1997 los autores propusieron JavaScript para que fuera adoptado como estándar de la European Computer Manufacturers 'Association ECMA, que a pesar de su nombre no es europeo sino internacional, con sede en Ginebra. En junio de 1997 fue adoptado como un estándar ECMA, con el nombre de ECMAScript. Poco después también como un estándar ISO.

JScript es la implementación de ECMAScript de Microsoft, muy similar al JavaScript de Netscape, pero con ciertas diferencias en el modelo de objetos del navegador que hacen ambas versiones sean incompatibles con frecuencia.

Para evitar estas incompatibilidades, el World Wide Web Consortium diseñó el estándar Document Object Model (DOM o Modelo de Objetos del Documento en castellano), que incorporan Konqueror, las versiones 6 de Internet Explorer y

Netscape Navigator, Opera la versión 7, y Mozilla Application Suite, Mozilla desde su primera versión. (Wikipedia, 2010).

La W3C es el organismo que regula los estándares Web, necesario para asegurar así el acceso universal a la información. Aun así, hay quien ignora estos estándares, dando lugar a páginas Web que solo se ven bien con unos o unos pocos navegadores concretos (W3C, 2010).

### 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, 2010):

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

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 interoperatibilidad 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, 2010).

#### 2.2.5.1 Historia del W3C.

En 1989, Tim Berners-Lee creó la World Wide Web. Acuñó el término World Wide Web, desarrolló el primer servidor para la World Wide Web, HTTPd, y el primer programa de cliente (un navegador y un editor), WorldWideWeb, en Octubre de 1990. Creó la primera versión del Lenguaje de Marcado de Hipertexto (HTML), un lenguaje de formato que permite la utilización de enlaces de hipertexto

y que se convirtió en el formato de publicación principal para la Web. Sus especificaciones iniciales para URI, HTTP y HTML, fueron revisadas y discutidas en grandes círculos, según crecía la tecnología Web (Lamarca, 2005; W3C, 2010).

Graduado por la Universidad de Oxford, Inglaterra, Tim-Berners-Lee ha sido el director de W3C desde su creación. En octubre de 1994, fundó el W3C en el Laboratorio de Ciencias Informáticas del Instituto de Tecnología de Massachusetts (MIT/LCS), en colaboración con el CERN, donde la Web tuvo su origen, con la colaboración de la Agencia de Investigación de Proyectos Avanzados de Defensa (DARPA) y de la Comisión Europea (Lamarca, 2005; W3C, 2010; Wikipedia, 2010).

En Abril de 1995, el Instituto Nacional de Investigación en Informática y Automática (INRIA) 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 el año 2003, el Consorcio Europeo para la Investigación en Informática y Matemáticas (ERCIM) sustituyó, en el papel de sede Europea al INRIA. El W3C, con el objetivo de captar una audiencia internacional, abre oficinas en todo el mundo (Lamarca, 2005; W3C, 2010; Wikipedia, 2010).

Diferentes organizaciones procedentes de diversos puntos del mundo y dentro de campos muy diferentes, forman parte del W3C para tomar parte en un foro neutral donde se participa en la creación de estándares Web. Miembros del W3C y un grupo de expertos técnicos, han hecho posible que el W3C sea reconocido a nivel internacional por su contribución en el desarrollo de la Web.

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, 2010).

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, 2010).

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 Consorcio 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, 2010).

#### 2.2.5.2 Estándares del W3C.

Un estándar pasa por los siguientes estados (Wikipedia, 2010):

- Working Draft (Borrador de Trabajo)
- Last Call (Última convocatoria)
- Proposed Recommendation (Propuesta de Recomendación) y
- Candidate Recomendation (Recomendación Candidata)

Finaliza con la aprobación de la Recomendación, lo que equivale a una homologación de la propuesta, es decir, un nuevo estándar público y abierto para la Web. La mayoría de estas recomendaciones son secundadas por los fabricantes de herramientas (navegadores, editores, buscadores) y tecnologías (servicios Web, directorios, registros). Esta competencia en exclusiva del W3C para crear estándares abiertos es crucial, pues de ella depende que ningún fabricante alcance nunca el monopolio de explotación de la Web (Wikipedia, 2010).

## 2.3 Lenguajes de Programación.

Actualmente existen diferentes lenguajes de programación para desarrollar en la web, estos han ido surgiendo debido a las tendencias y necesidades de las plataformas. Más adelante se mostraran las ventajas y desventajas de los lenguajes más conocidos.

Desde los inicios de Internet, fueron surgiendo diferentes demandas por los usuarios y se dieron soluciones mediante lenguajes estáticos. A medida que paso el tiempo, las tecnologías fueron desarrollándose y surgieron nuevos problemas a dar solución. Esto dio lugar a desarrollar lenguajes de programación dinámicos para la

Web, que permitieran interactuar con los usuarios y utilizaran sistemas de Bases de Datos. A continuación se presentará una introducción a los diferentes lenguajes de programación para la web.

### 2.3.1 Lenguaje HTML

Desde el surgimiento de internet se han publicado sitios web gracias al lenguaje HTML. Es un lenguaje estático para el desarrollo de sitios. Los archivos pueden tener las extensiones (htm, html).

**Ventajas:**

- 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 web o WYSIWYG.
- 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 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.

El código Javascript puede ser integrado dentro de nuestras páginas web. Para evitar incompatibilidades el World Wide Web Consortium (W3C) diseño un estándar denominado DOM (en inglés Document Object Model, en su traducción al español Modelo de Objetos del Documento).

**Sintaxis:**

<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 Site Scripting renombrado a XSS por su similitud con las hojas de estilo CSS).

### 2.3.3 Lenguaje PHP

Es un lenguaje de programación utilizado para la creación de sitio web. PHP es un acrónimo recursivo que significa "PHP Hypertext Pre-processor", (inicialmente se llamó Personal Home Page). Surgió en 1995, desarrollado por PHP Group.

PHP es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. PHP no necesita ser compilado para ejecutarse. Para su funcionamiento necesita tener instalado Apache o IIS con las librerías de PHP. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas. Los archivos cuentan con la extensión (php).

**Sintaxis:**

La sintaxis utilizada para incorporar código PHP es la siguiente:

```
<?
$mensaje = “Hola”;
echo $mensaje;
?>
```

También puede usarse:

```
<?php
$mensaje = “Hola”;
echo $mensaje;
?>
```

**Ventajas:**

- Muy fácil de aprender.
- Se caracteriza por ser un lenguaje muy rápido.
- Soporta en cierta medida la orientación a objeto. Clases y herencia.
- Es un lenguaje multiplataforma: Linux, Windows, entre otros.
- Capacidad de conexión con la mayoría de los manejadores de base de datos: MysSQL, PostgreSQL, Oracle, MS SQL Server, entre otras.
- Capacidad de expandir su potencial utilizando módulos.
- Posee documentación en su página oficial la cual incluye descripción y ejemplos de cada una de sus funciones.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Incluye gran cantidad de funciones.
- No requiere definición de tipos de variables ni manejo detallado del bajo nivel.

**Desventajas:**

- Se necesita instalar un servidor web.
- Todo el trabajo lo realiza el servidor y no delega al cliente. Por tanto puede ser más ineficiente a medida que las solicitudes aumenten de número.

- La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP.
- La programación orientada a objetos es aún muy deficiente para aplicaciones grandes.
- Dificulta la modularización.
- Dificulta la organización por capas de la aplicación.

**Seguridad:**

PHP es un poderoso lenguaje e intérprete, ya sea incluido como parte de un servidor web en forma de módulo o ejecutado como un binario CGI separado, es capaz de acceder a archivos, ejecutar comandos y abrir conexiones de red en el servidor. Estas propiedades hacen que cualquier cosa que sea ejecutada en un servidor web sea insegura por naturaleza.

PHP está diseñado específicamente para ser un lenguaje más seguro para escribir programas CGI que Perl o C, y con la selección correcta de opciones de configuración en tiempos de compilación y ejecución, y siguiendo algunas prácticas correctas de programación.

### 2.3.4 Lenguaje ASP

Es una tecnología del lado de servidor desarrollada por Microsoft para el desarrollo de sitio web dinámicos. ASP significa en inglés (Active Server Pages), fue liberado por Microsoft en 1996. Las páginas web desarrolladas bajo este lenguaje es necesario tener instalado Internet Information Server (IIS).

ASP no necesita ser compilado para ejecutarse. Existen varios lenguajes que se pueden utilizar para crear páginas ASP. El más utilizado es VBScript, nativo de Microsoft. ASP se puede hacer también en Perl y Jscript (no JavaScript). El código ASP puede ser insertado junto con el código HTML. Los archivos cuentan con la extensión (asp).

**Sintaxis:**

<% %>

**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

Este es un lenguaje comercializado por Microsoft, y usado por programadores para desarrollar entre otras funciones, sitios web. ASP.NET es el sucesor de la tecnología ASP, fue lanzada al mercado mediante una estrategia de mercado denominada .NET.

El ASP.NET fue desarrollado para resolver las limitantes que brindaba tu antecesor ASP. Creado para desarrollar web sencillas o grandes aplicaciones. Para el desarrollo de ASP.NET se puede utilizar C#, VB.NET o J#. Los archivos cuentan con la extensión (aspx). Para su funcionamiento de las páginas se necesita tener instalado IIS con el Framework .Net. Microsft Windows 2003 incluye este framework, solo se necesitará instalarlo en versiones anteriores.

**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 se servlets.

**Desventajas:**

- Complejidad de aprendizaje.

### 2.3.7 Lenguaje Python

Es un lenguaje de programación creado en el año 1990 por Guido van Rossum, es el sucesor del lenguaje de programación ABC. Python es comparado habitualmente con Perl. Los usuarios lo consideran como un lenguaje más limpio para programar. Permite la creación de todo tipo de programas incluyendo los sitios web.

Su código no necesita ser compilado, por lo que se llama que el código es interpretado. Es un lenguaje de programación multiparadigma, lo cual fuerza a que los programadores adopten por un estilo de programación particular:

- Programación orientada a objetos.
- Programación estructurada.
- Programación funcional.
- Programación orientada a aspectos.

**Sintaxis:**

Ejemplo de una clase en Phyton:

```
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

Es un lenguaje interpretado de muy alto nivel y orientado a objetos. Desarrollado en el 1993 por el programador japonés Yukihiro "Matz" Matsumoto. Su sintaxis está inspirada en Phyton, Perl. Es distribuido bajo licencia de software libre (Opensource).

Ruby es un lenguaje dinámico para una programación orientada a objetos rápida y sencilla. Para los que deseen iniciarse en este lenguaje pueden encontrar un tutorial interactivo de ruby. Se encuentra también a disposición de estos usuarios un sitio con informaciones y cursos en español.

**Sintaxis:**

```
puts "hola"
```

**Características:**

- Existe diferencia entre mayúsculas y minúsculas.
- Múltiples expresiones por líneas, separadas por punto y coma ";".
- Dispone de manejo de excepciones.
- Ruby puede cargar librerías de extensiones dinámicamente si el Sistema Operativo lo permite.
- Portátil.

**Ventajas:**

- Permite desarrollar soluciones a bajo Costo.

- Software libre.
- Multiplataforma.

### 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 uno del internet. Dentro de estos lenguajes tenemos:

#### 2.3.9.1 Lenguaje CSS

Las hojas de estilo en cascada (Cascading Style Sheets), CSS es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El W3C (World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores.

La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación.

Por ejemplo, el elemento de HTML <h1> indica que un bloque de texto es un encabezamiento y que es más importante que un bloque etiquetado como <H2>. Versiones más antiguas de HTML permitían atributos extra dentro de la etiqueta abierta para darle formato (como el color o el tamaño de fuente). No obstante, cada etiqueta <H1> debía disponer de la información si se deseaba un diseño consistente para una página y, además, una persona que lea esa página con un navegador pierde totalmente el control sobre la visualización del texto.

Cuando se utiliza CSS, la etiqueta <H1> no debería proporcionar información sobre cómo va a ser visualizado, solamente marca la estructura del documento. La información de estilo separada en una hoja de estilo, específica cómo se ha de mostrar <H1>: color, fuente, alineación del texto, tamaño y otras características no visuales como definir el volumen de un sintetizador de voz, por ejemplo.

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 Los tres tipos de estilos.

CSS proporciona tres caminos diferentes para aplicar las reglas de estilo a una página Web:

1. 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.
2. 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.
3. 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:

- Control centralizado de la presentación de un sitio web completo con lo que se agiliza de forma considerable la actualización del mismo.

- 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.
- 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.
- 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).

#### 2.3.9.1.3 Diagramado de página en CSS.

Antes de que estuviera disponible CSS, la única forma de componer espacialmente una página era el uso de tablas. Aunque es una técnica cómoda y versátil, se está usando un elemento con una semántica particular, que es la de expresar información tabular, solamente por su efecto en la presentación.

La introducción de CSS ha permitido en muchos casos reemplazar el uso de tablas. Sin embargo CSS todavía no permite la versatilidad que ofrecían las tablas, lograr un diagramado de una página compleja suele ser una tarea difícil en CSS y las diferencias entre navegadores dificultan aún más la tarea. Se espera que futuros desarrollos en CSS3 resuelvan esta deficiencia y hagan de CSS un lenguaje más apto para describir la estructura espacial de una página.

#### 2.3.9.1.4 Recomendaciones del W3C.

Cascading Style Sheets, nivel 1 (CSS1), Diciembre de 1996

- Propiedades de fuentes
- Propiedades de color y fondo
- Propiedades de texto
    - espaciado de palabras
    - alineación
- Propiedades de caja
    - margen

  - borde
  - relleno
- Propiedades de clasificación
  - visualización
  - listas

Ilustración de propiedades de caja:

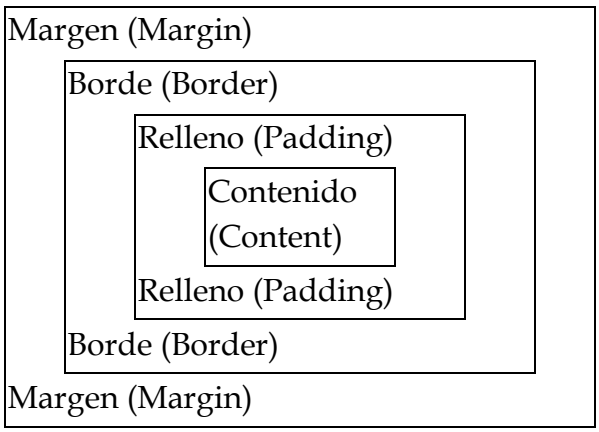



Figura 5. Propiedades de Caja.

Si se define una caja con el atributo width (ancho), se interpreta por el modelo de caja del W3C como la anchura del contenido. La anchura del relleno y del borde se añade a la anchura total del elemento.

En el modelo no puede especificarse el margen, el relleno o el borde, en la misma etiqueta, que el ancho del contenido.

#### 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 requieren 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.

AJAX es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores, dado que está basado en estándares abiertos como JavaScript y Document Object Model (DOM).

#### 2.3.9.2.1 Problemas e Inconvenientes.

- 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.

#### 2.3.9.2.2 Funcionamiento.

El usuario accede a la aplicación que es enviada por el servidor en formato HTML, JavaScript y CSS. Luego el código JavaScript de la aplicación pide al

servidor los datos que quiere mostrar y este, ejecuta un código de lado de servidor que envía al navegador los datos en formato XML.

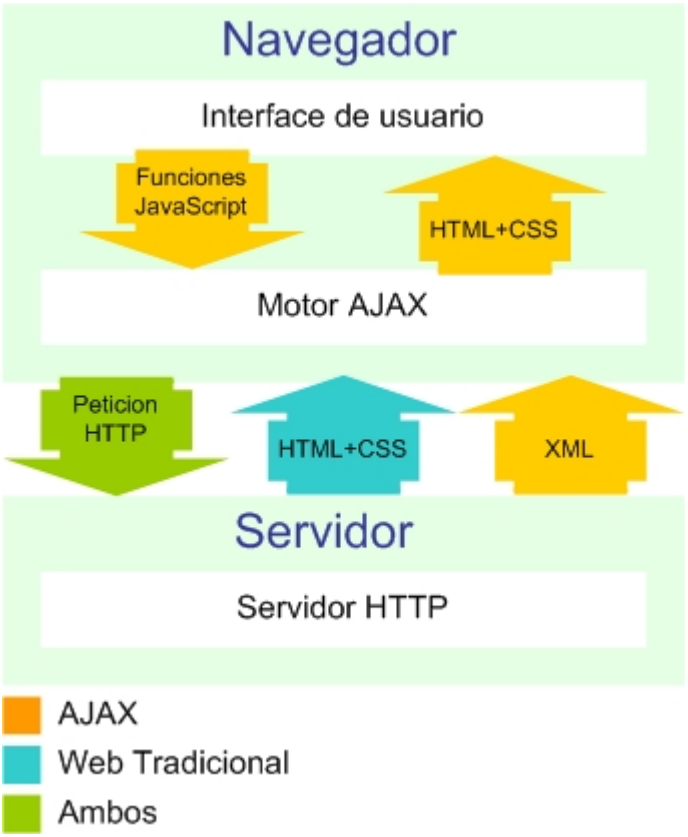



Figura 6. Diagrama de Funcionamiento AJAX.

Cada vez que el usuario realiza una acción que significa mostrar unos datos, la capa JavaScript, repite la acción anterior de manera invisible al usuario y muestra los datos deseados.

#### 2.3.9.3 JQuery.

Actualmente el framework de Javascript jQuery es una herramienta de desarrollo y diseño fundamental para cualquier plataforma web.

jQuery es un conjunto de librerías en un único archivo de varios kilobytes que permite enriquecer estéticamente una página web. Además es compatible con todos los navegadores modernos como Firefox 2.0+, Internet Explorer 6+, Safari

2.0.2+, Opera 9 y Google Chrome**.** Es capaz de crear eventos, efectos y animaciones, obtener información de los navegadores, manipular elementos DOM y AJAX.

## 2.4 Bases de Datos.

Una base de datos es un "almacén" que nos permite guardar grandes cantidades de información de forma organizada para que luego podamos encontrar y utilizar fácilmente.

El término de bases de datos fue escuchado por primera vez en 1963, en un simposio celebrado en California, USA. Una **base de datos** se puede definir como un conjunto de información relacionada que se encuentra agrupada o estructurada.

Desde el punto de vista informático, la base de datos es un sistema formado por un conjunto de datos almacenados en discos que permiten el acceso directo a ellos y un conjunto de programas que manipulen ese conjunto de datos.

Cada base de datos se compone de una o más tablas que guarda un conjunto de datos. Cada tabla tiene una o más **columnas** y **filas**. Las columnas guardan una parte de la información sobre cada elemento que queramos guardar en la tabla, cada fila de la tabla conforma un registro.

### 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.

#### 2.4.1.1 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.

### 2.4.2 Sistema de Gestión de Base de Datos (SGBD).

Los Sistemas de Gestión de Base de Datos (en inglés DataBase Management System - DBMS) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.

### 2.4.3 Ventajas y desventajas de las Bases de Datos.

- Control sobre la redundancia de datos:

Los sistemas de ficheros almacenan varias copias de los mismos datos en ficheros distintos. Esto hace que se desperdicie espacio de almacenamiento, además de provocar la falta de consistencia de datos.

En los sistemas de bases de datos todos estos ficheros están integrados, por lo que no se almacenan varias copias de los mismos datos. Sin embargo, en una base de datos no se puede eliminar la redundancia completamente, ya que en ocasiones es necesaria para modelar las relaciones entre los datos.

- Consistencia de datos:

Eliminando o controlando las redundancias de datos se reduce en gran medida el riesgo de que haya inconsistencias. Si un dato está almacenado una sola vez, cualquier actualización se debe realizar sólo una vez, y está disponible para todos los usuarios inmediatamente. Si un dato está duplicado y el sistema conoce esta redundancia, el propio sistema puede encargarse de garantizar que todas las copias se mantienen consistentes.

- Compartición de datos:

En los sistemas de ficheros, los ficheros pertenecen a las personas o a los departamentos que los utilizan. Pero en los sistemas de bases de datos, la base de

datos pertenece a la empresa y puede ser compartida por todos los usuarios que estén autorizados.

- Mantenimiento de estándares:

Gracias a la integración es más fácil respetar los estándares necesarios, tanto los establecidos a nivel de la empresa como los nacionales e internacionales. Estos estándares pueden establecerse sobre el formato de los datos para facilitar su intercambio, pueden ser estándares de documentación, procedimientos de actualización y también reglas de acceso.

- Mejora en la integridad de datos:

La integridad de la base de datos se refiere a la validez y la consistencia de los datos almacenados. Normalmente, la integridad se expresa mediante restricciones o reglas que no se pueden violar. Estas restricciones se pueden aplicar tanto a los datos, como a sus relaciones, y es el SGBD quien se debe encargar de mantenerlas.

- Mejora en la seguridad:

La seguridad de la base de datos es la protección de la base de datos frente a usuarios no autorizados. Sin unas buenas medidas de seguridad, la integración de datos en los sistemas de bases de datos hace que éstos sean más vulnerables que en los sistemas de ficheros.

- Mejora en la accesibilidad a los datos:

Muchos SGBD proporcionan lenguajes de consultas o generadores de informes que permiten al usuario hacer cualquier tipo de consulta sobre los datos, sin que sea necesario que un programador escriba una aplicación que realice tal tarea.

- Mejora en la productividad:

El SGBD proporciona muchas de las funciones estándar que el programador necesita escribir en un sistema de ficheros. A nivel básico, el SGBD proporciona todas las rutinas de manejo de ficheros típicas de los programas de aplicación.

El hecho de disponer de estas funciones permite al programador centrarse mejor en la función específica requerida por los usuarios, sin tener que preocuparse de los detalles de implementación de bajo nivel.

- Mejora en el mantenimiento:

En los sistemas de ficheros, las descripciones de los datos se encuentran inmersas en los programas de aplicación que los manejan.

Esto hace que los programas sean dependientes de los datos, de modo que un cambio en su estructura, o un cambio en el modo en que se almacena en disco, requiere cambios importantes en los programas cuyos datos se ven afectados.

Sin embargo, los SGBD separan las descripciones de los datos de las aplicaciones. Esto es lo que se conoce como independencia de datos, gracias a la cual se simplifica el mantenimiento de las aplicaciones que acceden a la base de datos.

- Aumento de la concurrencia:

En algunos sistemas de ficheros, si hay varios usuarios que pueden acceder simultáneamente a un mismo fichero, es posible que el acceso interfiera entre ellos de modo que se pierda información o se pierda la integridad. La mayoría de los SGBD gestionan el acceso concurrente a la base de datos y garantizan que no ocurran problemas de este tipo.

- Mejora en los servicios de copias de seguridad:

Muchos sistemas de ficheros dejan que sea el usuario quien proporcione las medidas necesarias para proteger los datos ante fallos en el sistema o en las aplicaciones. Los usuarios tienen que hacer copias de seguridad cada día, y si se produce algún fallo, utilizar estas copias para restaurarlos.

En este caso, todo el trabajo realizado sobre los datos desde que se hizo la última copia de seguridad se pierde y se tiene que volver a realizar. Sin embargo, los SGBD actuales funcionan de modo que se minimiza la cantidad de trabajo perdido cuando se produce un fallo.

Desventajas de las bases de datos

- Complejidad:

Los SGBD son conjuntos de programas que pueden llegar a ser complejos con una gran funcionalidad. Es preciso comprender muy bien esta funcionalidad para poder realizar un buen uso de ellos.

- Costo del equipamiento adicional:

Tanto el SGBD, como la propia base de datos, pueden hacer que sea necesario adquirir más espacio de almacenamiento. Además, para alcanzar las prestaciones deseadas, es posible que sea necesario adquirir una máquina más grande o una máquina que se dedique solamente al SGBD. Todo esto hará que la implantación de un sistema de bases de datos sea más cara.

- Vulnerable a los fallos:

El hecho de que todo esté centralizado en el SGBD hace que el sistema sea más vulnerable ante los fallos que puedan producirse. Es por ello que deben tenerse copias de seguridad (Backup).

### 2.4.4 Modelo de Datos.

Un modelo de datos es un lenguaje orientado a describir una Base de Datos. Típicamente un modelo de datos permite describir:

- Las estructuras de datos de la base: El tipo de los datos que hay en la base y la forma en que se relacionan.
- Las restricciones de integridad: Un conjunto de condiciones que deben cumplir los datos para reflejar correctamente la realidad deseada.
- Operaciones de manipulación de los datos: típicamente, operaciones de agregado, borrado, modificación y recuperación de los datos de la base.

Otro enfoque es pensar que un modelo de datos permite describir los elementos de la realidad que intervienen en un problema dado y la forma en que se relacionan esos elementos entre sí.

Dentro del modelo de datos encontramos 4 tipos:

- Entidad – Relación.
- Relacional.
- De Red.
- Jerárquico

### 2.4.5 Lenguajes de las Bases de Datos.

- DDL -Lenguaje de Definición de Datos.

Un lenguaje de definición de datos (Data Definition Language, DDL) es un lenguaje proporcionado por el sistema de gestión de base de datos que permite a los usuarios de la misma llevar a cabo las tareas de definición de las estructuras que almacenarán los datos así como de los procedimientos o funciones que permitan consultarlos.

El lenguaje de programación SQL, el más difundido entre los gestores de bases de datos, admite las siguientes sentencias de definición: CREATE, DROP y ALTER, cada una de las cuales se puede aplicar a las tablas, vistas, procedimientos almacenados y triggers de la base de datos.

- Diccionario de Datos.

El diccionario de datos es un lugar dónde se deposita información acerca de datos como origen, descripción, relaciones y otros datos, es decir el diccionario de datos es una base de datos misma, la cual deposita datos acerca de los datos, el diccionario de datos es una guía y contiene "mapas guías" para la base de datos en vez de "nuevos datos", es decir es un lugar en dónde se almacena o se mantiene un conjunto de estados (controles), información relacionada con los diferentes tipos de registros (tablas) privilegios de los usuarios y estadísticas (cuantos registros tiene cada tabla, índices, etc.)

Los diccionarios de datos de los Sistemas de Base de datos (DBMS) no son iguales, aunque mantienen los mismos lineamientos o las mismas características.

En otras palabras, es un catálogo, un depósito, de los elementos en un sistema. Contiene las características lógicas de los sitios donde se almacenan los datos del sistema, incluyendo nombre, descripción, alias, contenido y organización. Identifica los procesos donde se emplean los datos y los sitios donde se necesita el

acceso inmediato a la información, se desarrolla durante el análisis de flujo de datos y auxilia a los analistas que participan en la determinación de los requerimientos del sistema, su contenido también se emplea durante el diseño.

En un diccionario de datos se encuentra la lista de todos los elementos que forman parte del flujo de datos en todo el sistema. Los elementos más importantes son flujos de datos, almacenes de datos y procesos. El diccionario guarda los detalles y descripciones de todos estos elementos.

Si los analistas desean conocer cuántos caracteres abarca un determinado dato o qué otros nombres recibe en distintas partes del sistema, o dónde se utiliza, encontrarán las respuestas en un diccionario de datos desarrollado en forma apropiada.

- DML -Lenguaje de Manipulación de Datos.

Un Lenguaje de Manipulación de Datos (Data Manipulation Language, DML) es un lenguaje proporcionado por el sistema de gestión de base de datos que permite a los usuarios de la misma llevar a cabo las tareas de consulta o manipulación de los datos, organizados por el modelo de datos adecuado.

El lenguaje de manipulación de datos más popular hoy día es SQL, usado para recuperar y manipular datos en una base de datos relacional.

- Con procedimiento: Álgebra Relacional.

El álgebra relacional es un conjunto de operaciones que describen paso a paso como computar una respuesta sobre las relaciones, tal y como éstas son definidas en el modelo relacional. Denominada de tipo procedimental, a diferencia del Cálculo relacional que es de tipo declarativo.

Describe el aspecto de la manipulación de datos. Estas operaciones se usan como una representación intermedia de una consulta a una base de datos y, debido a sus propiedades algebraicas, sirven para obtener una versión más optimizada y eficiente de dicha consulta.

- Sin Procedimiento: Cálculo Relacional.

El Cálculo relacional es un lenguaje de consulta que describe la respuesta deseada sobre una Base de datos sin especificar cómo obtenerla, a diferencia del Álgebra relacional que es de tipo procedural, el cálculo relacional es de tipo declarativo; pero siempre ambos métodos logran los mismos resultados.

- DCL -Lenguaje de Control de Datos.

Lenguaje de control de Datos (DCL): Está conformado por sentencias que controlan la integridad, atomicidad y en general la seguridad de los datos. Contiene elementos útiles para trabajar en un entorno multiusuario, en el que es importante la protección de los datos, la seguridad de las tablas y el establecimiento de restricciones en el acceso, así como elementos para coordinar la compartición de datos por parte de usuarios concurrentes, asegurando que no interfieren unos con otros. Se utiliza para cambiar los permisos asociados con un usuario o función de la base de datos. Sus instrucciones son:

  - GRANT: permite a un usuario trabajar con los datos o ejecutar determinadas instrucciones Transact - SQL.
  - DENY: deniega los permisos sobre los objetos de la base de datos. Este es la orden que siempre prevalece.
  - REVOKE: quita un permiso concedido o denegado previamente.

Para el acceso y cambio de los datos Transact - SQL presenta el Lenguaje de tratamiento de datos (DML, Data Management Language), que cuenta con cuatro instrucciones, con las cuales se pueden implementar gran parte de las operaciones de mantenimiento:

  - SELECT recupera los datos existentes.
  - UPDATE se usa para cambiar los datos.
  - INSERT permite agregar nuevos datos.
  - DELETE borra datos de la base de datos.

- TCL –Lenguaje de Control de Transacciones.

### 2.4.6 Abstracción de la Información.

Existen diferentes niveles de abstracción para simplificar la interacción de los usuarios con el sistema; Interno, conceptual y externo, específicamente el de almacenamiento físico, el del usuario y el del programador.

- Nivel físico.

Es la representación del nivel más bajo de abstracción, en éste se describe en detalle la forma en cómo de almacenan los datos en los dispositivos de almacenamiento (por ejemplo, mediante señaladores o índices para el acceso aleatorio a los datos).

- Nivel conceptual.

El siguiente nivel más alto de abstracción, describe que datos son almacenados realmente en la base de datos y las relaciones que existen entre los mismos, describe la base de datos completa en términos de su estructura de diseño. El nivel conceptual de abstracción lo usan los administradores de bases de datos, quienes deben decidir qué información se va a guardar en la base de datos.

Consta de las siguientes definiciones:

1. *Definición de los datos*: Se describen el tipo de datos y la longitud de campo todos los elementos direccionables en la base. Los elementos por definir incluyen artículos elementales (atributos), totales de datos y registros conceptuales (entidades).
2. *Relaciones entre datos*: Se definen las relaciones entre datos para enlazar tipos de registros relacionados para el procesamiento de archivos múltiples.

En el nivel conceptual la base de datos aparece como una colección de registros lógicos, sin descriptores de almacenamiento. En realidad los archivos conceptuales no existen físicamente. La transformación de registros conceptuales a registros físicos para el almacenamiento se lleva a cabo por el sistema y es transparente al usuario.

- Nivel de visión.

Nivel más alto de abstracción, es lo que el usuario final puede visualizar del sistema terminado, describe sólo una parte de la base de datos al usuario

acreditado para verla. El sistema puede proporcionar muchas visiones para la misma base de datos.

La interrelación entre estos tres niveles de abstracción se ilustra en la siguiente figura.

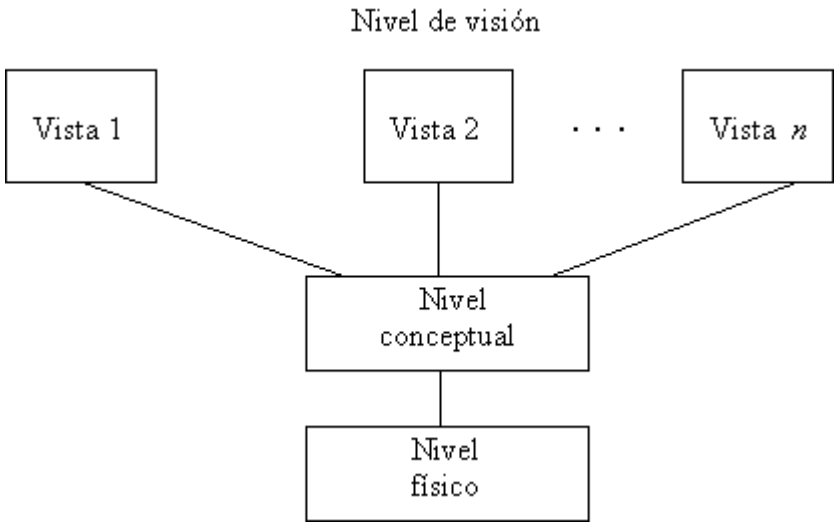



Figura 7. Diagrama del Nivel de Visión.

## 2.5 Metodología del Trabajo

### 2.5.1 Arquitectura del Software

La Arquitectura del Software alude a <<la estructura global del software y la manera en que esa estructura proporciona integridad conceptual a un sistema>>. Si huimos del formalismo, la Arquitectura Software es la parte de la Ingeniería del software que se ocupa de la descripción y el tratamiento de la estructura de un sistema como una serie de componentes, con el fin de organizar adecuadamente los distintos subsistemas y permitir la integración de diferentes grupos de desarrollo en el mismo proyecto.

La arquitectura del software corresponde a la fase de diseño, aunque recientes propuestas incluyen su tratamiento desde etapas tempranas del desarrollo, realizando su especificación de forma concurrente con la especificación de requisitos. Su principal objetivo se centra en la importancia de la descripción estructural de los sistemas software, un aspecto conocido pero poco tratado.

Actualmente, pese al avance de la disciplina, no existe un estándar acerca de qué y cómo tratar la problemática relacionada con la Arquitectura Software, aunque si hay partes comunes.

#### 2.5.1.1 Definición y Arquitectura del Software

Tal y como hemos comentado anteriormente, no hay consenso respecto a la definición de la misma; distintos autores utilizan diferentes definiciones, según el énfasis que quieran hacer en distintos puntos. Por ello es muy habitual, citar de manera simultánea varias definiciones, con el fin de proporcionar una visión de conjunto. Nosotros nos quedaremos con la más citada en la literatura por ser breve y concisa:

*La arquitectura de software está compuesta por la estructura de los componentes de un programa o sistema, sus interrelaciones, y los principios y reglas que gobiernan su diseño y evolución a lo largo del tiempo.*
David Garlan y Dewayne Perry

Esta definición aunque incompleta, poco explícita y en parte ambigua, resulta ser una definición general, abierta, no excluyente, amplia y flexible. También es unificadora, en el sentido de que es válida tanto para la visión descriptiva – representada por Garlan – como para la visión de proceso- representada por Perry. Las siguientes definiciones se apoyan sobre ésta e intentan ser más detalladas.

Cuando hablamos de Arquitectura Software, nos referimos a estructura, refiriéndose tanto a su concepción como a su especificación. Sin embargo, la expresión no es utilizada únicamente en este sentido, sino que a medida que se desarrolla el campo, se han podido identificar al menos cuatro significados diferentes, todos ellos interrelacionados:

· **Arquitectura como Producto:** Éste es el significado por excelencia, y al que se refieren las definiciones proporcionadas. La arquitectura es el conjunto formado por la organización, la estructura de un sistema de software. Desde esta perspectiva, interesa ante todo su descripción, evaluación y análisis.

· **Arquitectura como Proceso:** Habitualmente se considera que el proceso arquitectónico se constituye como el principal interés del campo. En este sentido, se hace referencia tanto al método a seguir para deducir la arquitectura de un sistema existente o no, como a la elaboración de una metodología de desarrollo que considere explícitamente el impacto de la arquitectura de software.

· **Arquitectura como Campo de Estudio:** Los últimos años se está investigando mucho al respecto. La realización de estudios al respecto y numerosas tesis lo avalan. También hay congresos anuales del tema.

· **Arquitectura como profesión:** Aunque este último aspecto es muy discutible, en países como Estados Unidos o Japón ya existe la figura del Arquitecto Software. Éste es el encargado de diseñar y mantener la descripción de la arquitectura, por lo que implícitamente es el que proporciona la visión integral del producto.

#### 2.5.1.2 Procesos de Desarrollo

El mundo de la informática no para de hablar de procesos de desarrollo, el modo de trabajar eficientemente para evitar catástrofes que llevan a que un gran porcentaje de proyectos se terminen sin éxito.

El objetivo de un proceso de desarrollo es subir la calidad del software (en todas las fases por las que pasa) a través de una mayor transparencia y control sobre el proceso. Da igual si es algo casero o para un cliente, hay que producir lo esperado en el tiempo esperado y con el coste esperado. Es labor del proceso de desarrollo hacer que esas medidas para aumentar la calidad sean reproducibles en cada desarrollo.

La implantación de un proceso de desarrollo es una labor más a medio-largo plazo que una labor de resultados inmediatos. Cuesta tiempo que los trabajadores se adapten al proceso, pero una vez superado la inversión se recupera con creces. Es por ello que no tiene sentido ajustarse a un proceso al pie de la letra, sino que hay que adaptarlo a las necesidades y características de cada empresa, equipo de trabajo o casi a cada proyecto.

Es labor del jefe de desarrollo decidir cuál es el que mejor se adapta a la

situación concreta a la que se enfrenta para minimizar la inversión requerida y obtener los resultados esperados. Por ejemplo, ¿tenemos el conocimiento necesario?, ¿es el equipo abierto a posibles cambios? Esta labor es difícil de llevar a cabo si no conocemos las exigencias de los procesos existentes.

Dentro de los procesos de desarrollo más conocidos tenemos:

- Proceso Unificado de Rational (Rational Unified Process - RUP desde ahora)
- Programación Extrema (eXtreme Programming - XP desde ahora) y
- Desarrollo Guiado por la Funcionalidad (Feature Driven Development - FDD desde ahora).

#### 2.5.1.2.1 Proceso Unificado de Rational (Rational Unified Process – RUP)

RUP es uno de los procesos más generales de los existentes actualmente, ya que en realidad está pensado para adaptarse a cualquier proyecto, y no tan solo de software.

Un proyecto realizado siguiendo RUP se divide en cuatro fases:

1. Intercepción (puesta en marcha)
2. Elaboración (definición, análisis, diseño)
3. Construcción (implementación)
4. Transición (fin del proyecto y puesta en producción)

Lista 1: Fases de RUP

En cada fase se ejecutarán una o varias iteraciones (de tamaño variable según el proyecto), y dentro de cada una de ellas seguirá un modelo de cascada o waterfall para los flujos de trabajo que requieren las nuevas actividades anteriormente citadas.

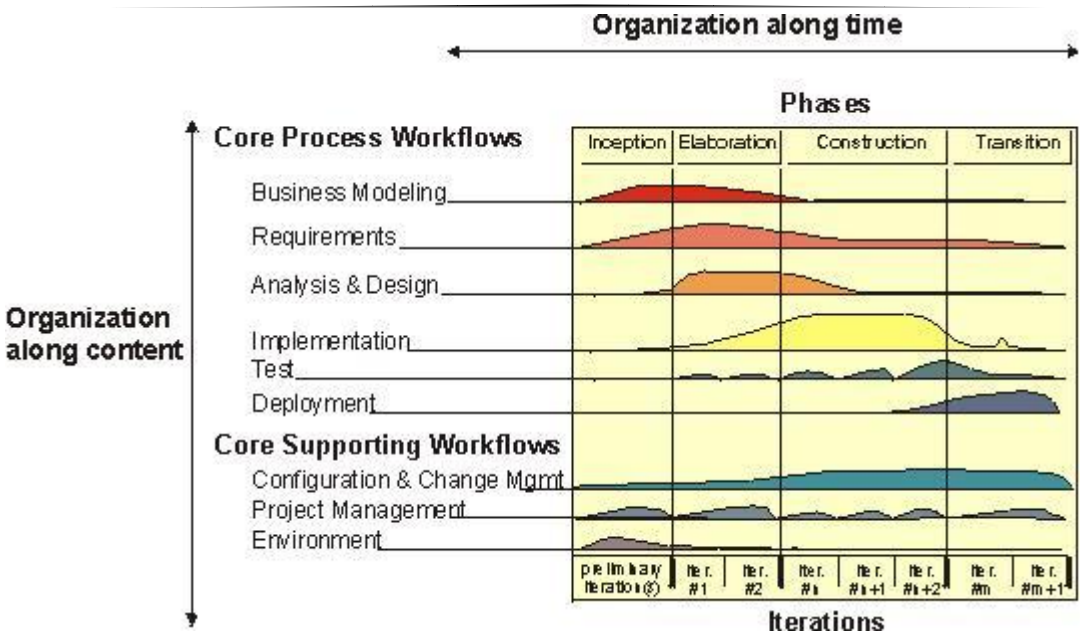



Figura 8: Vista general de RUP

RUP define nueve actividades a realizar en cada fase del proyecto:

1. Modelado del negocio
2. Análisis de requisitos
3. Análisis y diseño
4. Implementación
5. Test
6. Distribución
7. Gestión de configuración y cambios
8. Gestión del proyecto
9. Gestión del entorno

Lista 2: Actividades de RUP

Y el flujo de trabajo (workflow) entre ellas en base a los llamados diagramas de actividad. El proceso define una serie de roles que se distribuyen entre los miembros del proyecto y que definen las tareas de cada uno y el resultado (artefactos en la jerga de RUP) que se espera de ellos.

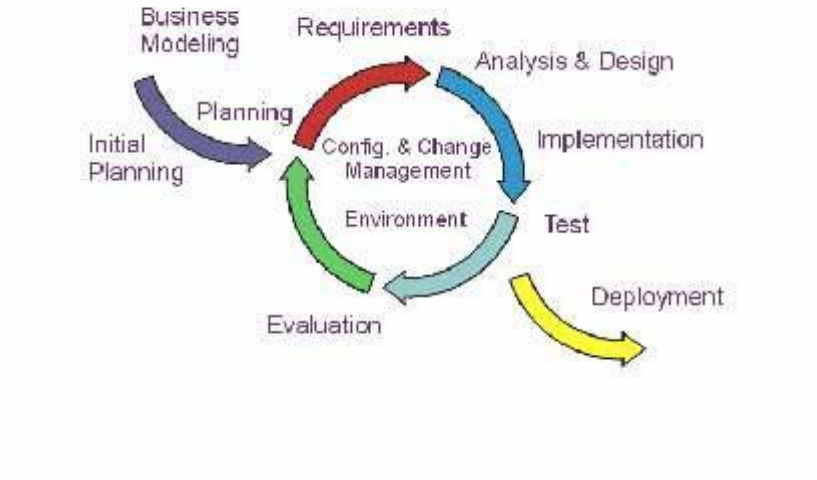



Figura 9: Flujos de trabajo de RUP

RUP se basa en casos de uso para describir lo que se espera del software y está muy orientado a la arquitectura del sistema, documentándose lo mejor posible, basándose en UML (Unified Modeling Language) como herramienta principal.

RUP es un proceso muy general y muy grande, por lo que antes de usarlo habrá que adaptarlo a las características de la empresa. Por suerte ya hay muchos procesos descritos en internet que son versiones reducidas del RUP.

#### 2.5.1.2.2 Programación Extrema (eXtreme Programming - XP)

Mientras que el RUP intenta reducir la complejidad del software por medio de estructura y la preparación de las tareas pendientes en función de los objetivos de la fase y actividad actual, XP, como toda metodología ágil, lo intenta por medio de un trabajo orientado directamente al objetivo, basado en las relaciones interpersonales y la velocidad de reacción.

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 toma 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.

Para cada release se discutirán los objetivos de la misma con el representante del cliente y se definirán las iteraciones (de pocas semanas de duración) necesarias para cumplir con los objetivos de la release. El resultado de cada iteración es un programa que se transmite al cliente para que lo juzgue. En base a su opinión se definen las siguientes iteraciones del proyecto y si el cliente no está contento se adaptará el plan de releases e iteraciones hasta que el cliente de su aprobación y el software este a su gusto.

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.

Como primer paso de cada iteración se escribirán las pruebas, de tal forma que puedan ser ejecutadas automáticamente, de manera que pueda comprobarse la corrección del software antes de cada release. Esto es de vital importancia en XP debido a su apuesta por las iteraciones cortas que generan software que el cliente puede ver y por la refactorización para mejorar el código constantemente, que hacen más que deseable una cantidad considerables de test lo más automatizables posible. Así pues, la funcionalidad concreta del software solo se escribe cuando las pruebas para su corrección estén preparadas.

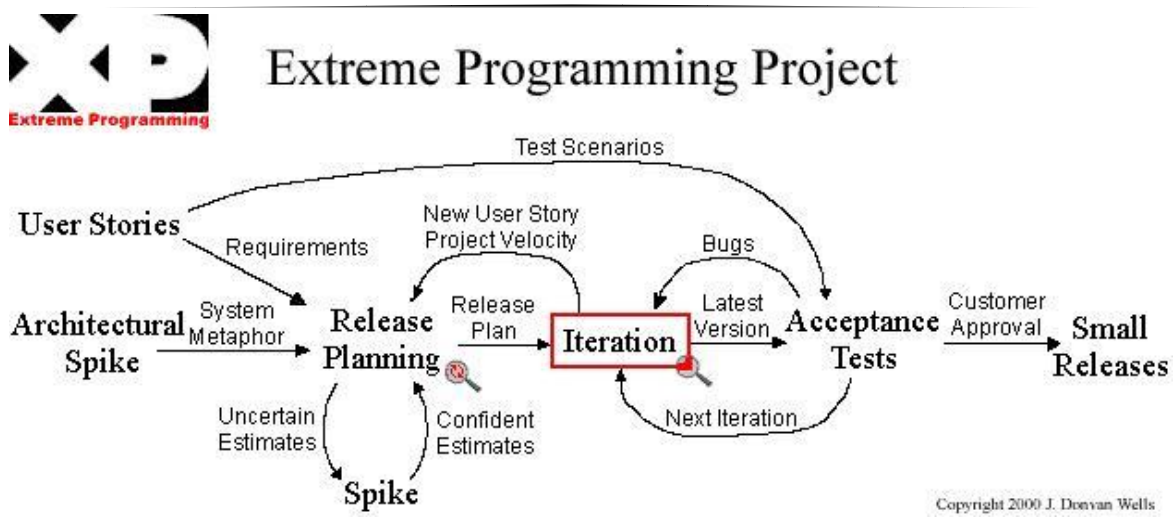



Figura 10: Vista general de XP

La codificación del software en XP se produce siempre en parejas (dos programadores, un ordenador), por lo que se espera que la calidad del mismo suba en el mismo momento de escribirlo. Al contrario que muchos otros métodos, el código pertenece al equipo en completo, no a un programador o pareja, de forma que cada programador puede cambiar cualquier parte del código en cualquier momento si así lo necesita dejándose en todo caso las mejoras orientadas al rendimiento para el final. Las parejas no se mantienen para todo el proyecto si no que rotan cíclicamente a lo largo del mismo, tanto en cuanto a los componentes de la misma como en las partes del software que desarrollan, así cada componente del equipo aprende como trabaja el resto. El objetivo ideal sería que cada componente del equipo trabaje al menos una vez con cada uno de los demás integrantes y con cada componente software, de forma que el conocimiento de la aplicación completa lo posea el equipo entero y no unos pocos miembros.

En XP se programará solo la funcionalidad que es requerida para la release actual. Es decir, una gran flexibilidad y capacidad de configuración solo será implementada cuando sea necesaria para cumplir los requerimientos de la release. Se sigue un diseño evolutivo con la siguiente premisa: conseguir la funcionalidad deseada de la forma más sencilla posible. De ahí una variación educada del famoso KISS (Keep It Simple Stupid), Keep things as simple as possible (mantén las cosas tan sencillas como sea posible). Este diseño evolutivo hace que no se le dé apenas importancia al análisis como fase independiente, puesto que se trabaja exclusivamente en función de las necesidades del momento.

#### 2.5.1.2.3 Desarrollo Guiado por la Funcionalidad (Feature Driven Development – FDD)

FDD es un proceso diseñado por Peter Coad, Erich Lefebvre y Jeff De Luca y se podría considerar a medio camino entre RUP y XP, aunque al seguir siendo un proceso ligero (en mi opinión, si tenemos que distinguir entre pesado/ligero) es más similar a este último.

FDD está pensado para proyectos con tiempo de desarrollo relativamente cortos (menos de un año). Se basa en un proceso iterativo con iteraciones cortas (~2 semanas) que producen un software funcional que el cliente y la dirección de la empresa pueden ver y monitorizar.

Las iteraciones se deciden en base a features (de ahí el nombre del proceso) o funcionalidades, que son pequeñas partes del software con significado para el cliente. Así, construir el sistema de ventas es algo que requiere mucho tiempo, y construir el sistema de persistencia no tiene significado para el cliente, pero si lo tiene enviar pedido por email.

Un proyecto que sigue FDD se divide en 5 fases:

1. Desarrollo de un modelo general
2. Construcción de la lista de funcionalidades
3. Plan de releases en base a las funcionalidades a implementar
4. Diseñar en base a las funcionalidades
5. Implementar en base a las funcionalidades

Lista 3: Fases de FDD

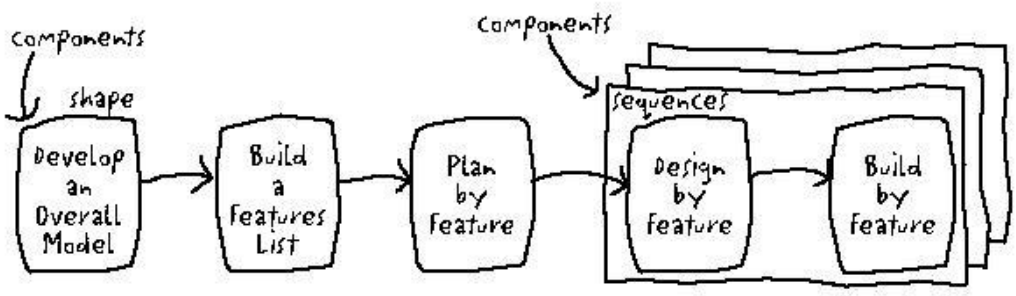



Figura 11: Vista general de FDD

Las primeras tres fases ocupan gran parte del tiempo en las primeras iteraciones, siendo las dos últimas las que absorben la mayor parte del tiempo según va avanzando el proyecto, limitándose las primeras a un proceso de refinamiento.

El trabajo (tanto de modelado como de desarrollo se realiza en grupo, aunque siempre habrá un responsable último (arquitecto jefe o jefe de programadores en función de la fase en que nos encontremos), con mayor experiencia, que tendrá la última palabra en caso de no llegar a un acuerdo. Al hacerlo en grupo se consigue que todos formen parte del proyecto y que los menos inexpertos aprendan de las discusiones de los más experimentados, y al tener un responsable último, se asignan las responsabilidades que todas las empresas exigen.

Las funcionalidades a implementar en una release se dividen entre los distintos subgrupos del equipo, y se procede a implementarlas. Las clases escritas tienen propietario (es decir, solo quién las crea puede cambiarlas), es por ello que en el equipo que implementa una funcionalidad dada deberán estar todos los dueños de las clases implicadas, pudiendo encontrarse un programador en varios grupos, implementando distintas funcionalidades. Habrá también un programador jefe (normalmente el más experimentado) que hará las funciones de líder del grupo que implementa esa funcionalidad.

En el proceso de implementar la funcionalidad también se contemplan como partes del mismo (en otros métodos se describen como actividades independientes) la preparación y ejecución de pruebas, así como revisiones del código (para distribuir el conocimiento y aumentar la calidad) e integración de las partes que componen el software.

FDD también define métricas para seguir el proceso de desarrollo de la aplicación, útiles para el cliente y la dirección de la empresa, y que pueden ayudar, además de para conocer el estado actual del desarrollo, a realizar mejores estimaciones en proyectos futuros.

# CAPÍTULO III

# "ANÁLISIS DEL PROBLEMA"

## 3.1 Descripción de la situación actual.

La FI contaba con un sistema local (TutorNet), el cual permitía gestionar la información de estudiantes, tutores de una generación, es decir, por año. Así mismo, este sistema permitía realizar evaluaciones por alumno, así como realizar una breve evaluación al sistema.

Ahora bien, este sistema limitaba en gran medida a que los tutores pudieran consultar de una forma fácil e intuitiva la información de sus alumnos, así como tener disponible información y archivos necesarios para dar la tutoría. Otro aspecto importante es que los tutores no podían enviar ningún tipo de información que le ayudara a la coordinación a llevar un seguimiento de las sesiones que los tutores tenían con sus estudiantes, así mismo, realizar un análisis a los coordinadores de los tutores era prácticamente imposible, lo que impedía realizar un seguimiento para poder mejorar y corregir las problemáticas que pueden ser presentadas en el programa de tutoría.

Para ofrecer los servicios tanto a los estudiantes, tutores, COPADI y coordinadores de carrera se pretende sistematizar dicha información con un Sistema Web (TutorFI), que sea capaz de albergar la información ingresada año con año que cubriera las necesidades de la Facultad de Ingeniería, como son:

Para el estudiante tenemos:

- Número de Cuenta
- Nombre completo
- Carrera
- Correo Electrónico
- Teléfono
- Id de tutores
- Bloque asignado
- Generación
- Seguimiento al Alumno por el tutor
- Fotografía del Estudiante

El tutor puede consultar su información:

- Id del tutor
- Nombre de usuario
- Nombre completo
- Grado
- Bloque asignado
- Carrera de estudiantes asignados
- División a la cual pertenece
- Ubicación
- Correo electrónico y
- Teléfono.

Además, el tutor puede hacer modificaciones a su información como es:

- Grado
- Carrera de estudiantes asignados
- División a la cual pertenece
- Ubicación
- Correo electrónico
- Teléfono y
- Contraseña

Por parte del alumno solo puede visualizar la información del (os) tutor (es) asignado (s):

- Nombre completo
- Grado
- Bloque
- Carrera de estudiantes asignados
- División
- Ubicación
- Correo electrónico y
- Teléfono

Como usuario COPADI, es posible ingresar en 4 rubros para visualizar información:

- Estudiante
  - Consulta de datos
  - Alta de un estudiante
  - Baja de un estudiante
  - Actualización de Datos
  - Visualizar la evaluación del estudiante
- Tutor
  - Consulta de datos
  - Alta de un tutor
  - Baja de un tutor
  - Actualización de Datos
- COPADI
  - Consulta de datos
  - Alta de un Administrador
  - Actualización de Datos

- Programa tutoría
  - Visualizar la evaluación al sistema
- Reportes
  - Descargar información de los registros de sesiones tanto individuales como grupales
  - Resultados de las encuestas COPADI, coordinador y autoevaluación.

Como se puede apreciar, las ventajas y prestaciones que ofrece el sistema son limitadas, además el uso del sistema se ve restringido en el acceso al mismo, debido a que es un sistema local, solo puede ser utilizado en la sala de computo de COPADI. Esta es la mayor desventaja ya que la mayoría de los tutores no tienen un tiempo para asistir a la sala y realizar el registro de su información, así mismo los alumnos tienen el mismo problema. Por otro lado, la sala de cómputo no cuenta con el equipo suficiente como para dar servicio a toda la comunidad.

### 3.2 Determinación de requerimientos del sistema.

Tomando como referencia lo mencionado en el subcapítulo anterior, y analizando exhaustivamente las necesidades y carencias con las que cuenta actualmente la COPADI, se genera la siguiente lista de requerimientos para satisfacer en su totalidad las necesidades que tiene la secretaría:

- Para que este portal pueda trabajar de manera global y eliminar la funcionalidad local, se requiere de un espacio en un servidor o en su defecto, la asignación de una dirección IP homologada para que se pueda acceder al sistema desde cualquier ubicación, sea dentro o fuera de la Coordinación.

- Teniendo pleno conocimiento de la gran cantidad de usuarios que tendrán acceso al sistema, se requiere de un equipo dedicado al sistema, con el propósito de tener una disponibilidad del 100% y con los mejores recursos para brindar un excelente servicio.

- Se requiere de una página Web donde se pueda mostrar la información de las diferentes actividades y programas que se llevan a cabo en la coordinación:

    - COPADI
        - Misión
        - Visión
        - Objetivo
        - Funciones
        - Valores

    - Tutoría
        - Tutoría Nueva Era
        - Definición
        - Coordinadores
        - Encuentro de Tutores
        - TutorFI

- PARA (Programa de Alto Rendimiento Académico)
  - Objetivos del PARA
  - Definición
  - Justificación
  - Objetivos del Programa
  - Descripción del Programa
  - Ingreso al Programa
  - Condiciones de Permanencia
  - Compromisos y Actividades
  - Objetivos de Tutoría en el PARA
  - Consejo Coordinador y Dictaminador del PARA (CCD)
  - Ventajas del Programa
  - Sociedad de Egresados
  - Actividades en el PARA
  - Reseña del PARA

- Asesorías Psicopedagógicas
  - Objetivos
  - Temáticas
  - Integrantes
  - Horarios

- Cursos Extracurriculares
- Becas
- Inglés
- Publicaciones
- Galería de Fotos
- Avisos
- Contacto
- Ligas de Interés
- Mapa del Sitio
- Créditos
- Organización
- Un contador general del Sitio.

Dentro de cada uno de estos rubros se anexará la información necesaria para mostrarla a los visitantes.

Para el apartado de avisos, cursos extracurriculares y asesorías psicopedagógicas se estará actualizando constantemente para tener al día la información tanto para estudiantes como para público en general, sobre todo la parte de avisos que es la que está en constante cambio.

- Para el nuevo sistema de tutorías (TutorFI) se requiere que sea capaz de cubrir todas las prestaciones y facilidades que ofrece el sistema existente (TutorNet), además de anexar una mayor información con la cual los tutores puedan trabajar para dar un seguimiento a los estudiantes.

  Haciendo una en cuentas a los tutores, se obtuvieron las necesidades más destacadas y más importantes que se requieren para un mejor manejo de la información y un mayor conocimiento de la situación de los estudiantes. Esto nos lleva a que el sistema debe cubrir las siguientes necesidades:

  - El sistema TutorFI deberá contar con 3 tipos de usuario:
    - Estudiante
    - Tutor
    - COPADI
    - Coordinador

  - Como usuario Alumno, el estudiante podrá consultar:
    - Información completa de su(s) tutor(es)
    - Asesorías que brinda la coordinación como:
      - Asesorías Psicopedagógicas
      - Asesorías Académicas
      - Taller de Ejercicios
    - Comunicados

  - Como usuario Tutor:
    - Llevar el seguimiento de los estudiantes en el que podrá:
      - Consultar la lista de sus alumnos de las diferentes generaciones.
      - Consultar la información de cada uno de los estudiantes, así como su foto para una mejor ubicación.

- Realizar evaluaciones a cada estudiante que lo requiera. Así mismo, consultar las evaluaciones que ha capturado.
- Realizar un registro de sesiones individuales que haya tenido con algún estudiante.

- Facilidad para obtener la información, de manera impresa o descargar en un archivo de Excel, toda la información deberá ser descargada por generación. La información es:
  - Lista de alumnos
  - Examen diagnostico
  - Calificaciones de primer semestre
  - Avance y promedio escolar

- El tutor podrá consultar su información, así como hacer las actualizaciones a su información.
- Hacer una evaluación al Programa de Tutoría
- Hacer el registro de las sesiones grupales que se llevan anualmente
- Realizar aportaciones al sistema
  - Enviar su foto
  - Capturar su semblanza
  - Enviar y consultar archivos que contribuyan al Programa de tutoría.
- Consultar información de las asesorías brindadas por la coordinación:
  - Asesorías Psicopedagógicas
  - Asesorías Académicas
  - Taller de Ejercicios

- Descargar material adicional para el apoyo a la tutoría
- Consultar comunicados de la Coordinación

- Como usuario COPADI (administrador), se debe tener acceso total a la información como:
  - Datos de tutores
  - Datos de alumnos
  - Consulta de registro de sesiones individuales

- Consulta de registro de sesiones semestrales
- Consulta de registro de sesiones grupales
- Consulta de avance escolar y materias inscritas en el semestre actual
- Consulta de resultados de examen diagnostico
- Consulta de calificaciones del primer semestre
- Consulta de administradores
- Alta, baja y actualización tanto de información de estudiantes, tutores, administradores, bloques, carreras, becas, etc.

- Como usuario Coordinador:
  - Consultar y modificar información del coordinador.
  - Consulta de reportes
    - Lista de tutores
    - Encuestas realizadas por tutor
    - Sesiones Grupales realizadas por tutor
    - Sesiones Individuales realizadas por tutor
    - Enviar y consultar aportaciones
      - Capturar fotografía
      - Capturar semblanza
      - Enviar y consultar archivos
  - Consultar información de las asesorías brindadas por la coordinación:
    - Asesorías Psicopedagógicas
    - Asesorías Académicas
    - Taller de Ejercicios
  - Descargar material adicional para el apoyo a la tutoría
  - Consultar comunicados de la Coordinación

Además, tanto el sistema como la página principal, deberán contener sus contadores de visitas para tener conocimiento de la cantidad de usuarios que se tiene.

Por otro lado, el sistema deberá presentar un pequeño tutorial sobre el uso del mismo.

# CAPÍTULO IV

## "Diseño y desarrollo del Portal COPADI"

El portal de la COPADI está diseñado para que el usuario pueda navegar de una manera sencilla, sin perder en ningún momento la página principal de la misma.

Su antecesora tenía un diseño elaborado en su totalidad en flash, algo que la normativa de la Dirección General de Servicios de Cómputo Académico (DGSCA) para sitios web institucionales establece como fuera de uso. Así mismo, el uso de flash, junto con una enorme cantidad de imágenes, sonidos y animaciones, hace que el sitio sea sumamente pesado para descargar la información al momento de visualizar su contenido. También es difícil la navegación para personas con diferentes capacidades.


Figura 1. Página COPADI anterior


Figura 2. Página COPADI anterior

Para el alojamiento del Portal, se adquirió y configuro el servidor donde se instaló el sistema operativo Fedora 11 donde será alojado el Portal.

Como anteriormente se mencionó, dado al lenguaje de programación que es empleado para el desarrollo del Portal (JAVA), se optó por usar el lenguaje de programación JSP para páginas web (Ver apéndice C). Para tal desarrollo se está utilizando lo que es el servidor de servlets (Tomcat) que usaremos para la página (ver apéndice A).

La diferencia dentro de los archivos de la programación, no encontramos con estas líneas de código:

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

Que es código JAVA dentro de un archivo HTML que hacen la llamada a otros archivos con código que serán usados dentro del documento para su ejecución.

## 4.1 Diseño del Portal

Dentro de los lineamientos que establece la DGSCA (http://recursosweb.unam.mx/recursos-web/lineamientos-unam/) para páginas web institucionales, se crea un diseño usando hojas de estilo en cascada CSS (Cascading Style Sheets), lo que permitirá realizar modificaciones al portal de una manera rápida y sencilla. Así mismo, se hará uso de JavaScript para realizar efectos que sean atractivos al usuario y permitir una mejor navegación dentro del portal.

Dentro de las hojas de estilo definimos el encabezado que tendrá el cuerpo del portal, el cual contiene logotipo de la Facultad de Ingeniería, logotipo de la Coordinación y la fecha actual:


Figura 3. Encabezado de la Página Principal

También se define el color del sitio, en el caso de la FI se establece que es de color rojo. Así mismo se establece el tipo de fuente, estilo de enlaces imágenes y la localización de los bloques de información dentro del Portal.

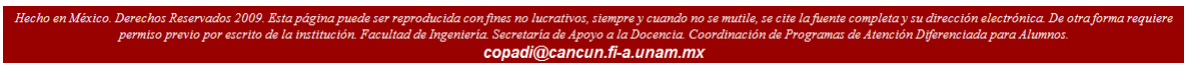



Figura 4. Pie de Página


Figura 5. Diseño del Cuerpo de la Página

Con ello generamos la hoja de estilo que emplearemos en el Portal, el que contendrá el bloque de menús, contenido, pie de página y encabezado, quedando de esta manera:

Para todo el Portal, definimos color de fondo, la imagen a utilizar (degradado de gris) como fondo, tipo de letra y alineación.

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

Definimos estilos para los enlaces:

```
a:link {
text-decoration:none;
}
a:visited {
color:#0073e6;
text-decoration:none;
}
a:hover {
color:#0073e6;
text-decoration:underline;
}
a.mail:link {
FONT-WEIGHT: bold; COLOR: white; TEXT-DECORATION: none
}
a.mail:visited {
FONT-WEIGHT: bold; COLOR: #0073e6; TEXT-DECORATION: none
}
a.mail:active {
FONT-WEIGHT: bold; COLOR: #0073e6; TEXT-DECORATION: none
}
a.mail:hover {
FONT-WEIGHT: bold; COLOR: #0073e6; TEXT-DECORATION: underline
}
a.bannerInf:link {
FONT-WEIGHT: bold; COLOR: white; TEXT-DECORATION: none
}
a.bannerInf:visited {
FONT-WEIGHT: bold; COLOR: #a3a3a3; TEXT-DECORATION: none
}
a.bannerInf:active {
FONT-WEIGHT: bold; COLOR: #a3a3a3; TEXT-DECORATION: none
}
a.bannerInf:hover {
FONT-WEIGHT: bold; COLOR: #a3a3a3; TEXT-DECORATION: underline
}
a img {
border-width:0;
}
```

Así mismo, establecemos el diseño para la cabecera del Portal, contenido, fecha, contador, menús, contenido y pie de página.

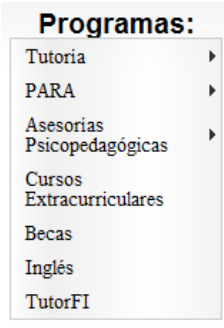
```
#logoFI{
background-image: url(../Logos/logoFI.gif);
float: left;
  width: 105px;
  height:127px;
  font-size: 0px;
  margin: 8px 0px 0px 40px;
}
```

```
#imagenCOPADI{
background-image: url(../Logos/SadCopadi.gif);
float: left;
width: 620px;
  height: 148px;
  font-size: 0px;
  margin: 0px 0px 0px 40px;
}
#logoCOPADI{
background-image: url(../Logos/logoCOPADI.gif);
float: left;
  width: 105px;
  height: 127px;
  font-size: 0px;
  margin: 8px 0px 0px 40px;
}

#fecha{
font-size: 12px;
color: #cc0000;
font-weight:bold;
}

#visitas{
font-size: 16px;
color: #cc0000;
font-weight:bold;
}

#banner{
background-color: transparent;
  clear: both;
  text-align: right;
  color: white;
}
#contenido2{
background-color: transparent;
background-image: url(../diseno/pieBanner.gif);
width: 800px;
height: 7px;
  float: left;
}
/* Fin de Estilo que inserta la linea horizontal bajo el banner Principal*/
#menu-izquierdo{
background-color: transparent;
background-image: url(../diseno/separaBannerIzq.gif);
background-repeat: repeat-y;
width: 180px;
height: 100%;
  float: left;

}
```


Viernes, 13 de Mayo de 2011

34111 Visitas



```
#contenido{
background-color: transparent;
width: 793px;
height: 100%;
  float: left;
}

#loggin{
text-align: center;
width: 100%;
  height: 100%;
}

#bannerInferior{
background-color: #990000;
  clear: both;
  text-align: center;
  font-size: 11px;
  font-style: oblique;
  color: white;
  height: 29px;
  bottom: 50px;
}
#pie{
background-color: #990000;
  clear: both;
  text-align: center;
  font-size: 11px;
  font-style: oblique;
  color: white;
  height: 46px;
  bottom: 50px;
}
```




Tanto para menús como para avisos y galerías de fotos se ha utilizado JavaScript para realizar los efectos mostrados, ayudado de hojas de estilo. Para el menú tenemos sus respectivos Scripts disponibles en la página principal.

## 4.2 Página Principal COPADI

Con el nuevo diseño se pretende mostrar a los usuarios un entorno más dinámico y amigable, que los invite a explorar la página en su totalidad. Se ha buscado presentar imágenes y fotos que reflejen el trabajo que se realiza en la Coordinación; además se busca que el usuario no tenga que navegar en interminables páginas repletas de texto plano, lo que en muchas ocasiones disminuye el interés e invita al abandono. En los casos en que se tiene textos, se ha realizado una estructura sencilla para su consulta. Además en los casos en que se tengan varias secciones, se ha evitado que se desplieguen tantas páginas como vínculos existan en cada uno de los programas de la Coordinación, logrando así que el usuario no se pierda entre varias páginas abiertas.

Así mismo, se busca que esta se convierta en un importante medio de difusión de las actividades que realiza la Coordinación, a la vez de motivar a la comunidad de la Facultad de Ingeniería a visitarla, con el fin de que conozcan horarios de actividades, información relevante sobre algún programa, asesorías, boletines e información de las diversas actividades – cursos que se imparten en la COPADI.

El esquema bajo el cual se ha elaborado la nueva página de la COPADI es el que se muestra a continuación, en el cual se podrán encontrar las diversas secciones y subsecciones que se incluyen en la página.

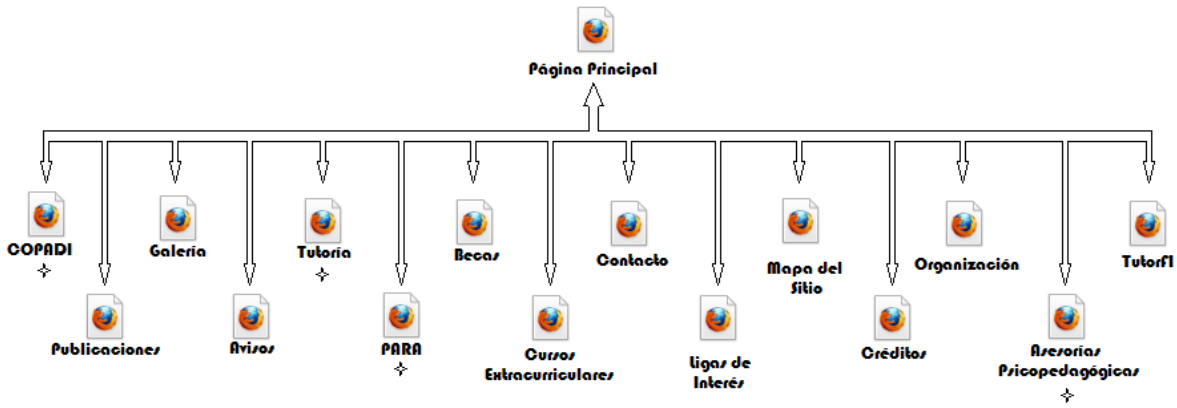



Figura 6. Estructura de la Página Principal COPADI

### 4.2.1 Coordinación de Programas de Atención Diferenciada para Alumnos (COPADI).

Como su nombre lo indica, la Coordinación de Programas de Atención Diferenciada para Alumnos (COPADI) está a cargo de diferentes programas que dan servicio y atienden a toda la comunidad de estudiantes de la Facultad de Ingeniería.

A continuación listamos cada uno de los programas que ofrece la coordinación, así como los servicios que se brindan.

#### 4.2.1.1 COPADI

##### 4.2.1.1.1 Misión de la COPADI

Mejorar el desempeño académico de los estudiantes de la Facultad de Ingeniería, mediante programas y acciones que inciden en su rendimiento escolar, en su formación como futuros ingenieros de alto nivel y comprometidos en una práctica profesional eficaz, eficiente, ética y de calidad, y en su crecimiento y desarrollo como seres humanos capaces de una realización plena, en íntima relación con la naturaleza y con un claro objetivo para trabajar por el mejoramiento de la vida de sus semejantes.

##### 4.2.1.1.2 Visión de la COPADI

La visión que se tiene de la COPADI hacia el año 2010 como una instancia modelo de servicio y apoyo:

- Con **capacidad de liderazgo** al interior de la Facultad en lo que se refiere a programas y acciones de beneficio a los estudiantes.
- Con capacidad de liderazgo en otras instancias universitarias, con respecto a todos los programas que coordina.
- Con una **cultura de la tutoría** consolidada en la que los académicos tutores y los estudiantes en tutoría consideren a este servicio como una práctica cotidiana indispensable en su relación académica y afectiva, así como en su crecimiento personal.

- Con un **Programa de Alto Rendimiento Académico (PARA)** plenamente conocido y eficientemente coordinado, capaz de formar generaciones numerosas de ingenieros de excelencia.
- Con un **Programa CIEC**, Cursos Intersemestrales Extracurriculares COPADI, plenamente conocido y socorrido por la calidad y utilidad de sus talleres y cursos.
- Con un **número suficiente de asesorías psicopedagógicas**, acordes con las problemáticas académicas y existenciales de los estudiantes de la Facultad.
- Con **publicaciones** elaboradas por el personal de la COPADI, así como por estudiantes del PARA, con amplia promoción y difusión en la comunidad académica de la Facultad.
- Con **colaboradores eficientes** y solidarios entre sí, y cooperativos con otras instancias de la Facultad y de la Universidad, convencidos del privilegio de contribuir a formar a los mejores seres humanos y profesionales de la ingeniería del país.

#### 4.2.1.1.3 Objetivos de la COPADI

Planear, realizar y coordinar programas y acciones comunes en grupos diferenciados de estudiantes, cuya finalidad es mejorar su desempeño académico y alentar con ello la excelencia académica en la Facultad de Ingeniería, al abatir la deserción, el rezago estudiantil y los bajos índices de eficiencia terminal.

Lograr que los grupos atendidos estén conformados por estudiantes creativos, innovadores, reflexivos, polifuncionales y emprendedores, en el marco de sistemas de formación avanzada, continua, abierta, crítica, en donde el alumno asuma su calidad de sujeto activo, protagonista de su propio aprendizaje y gestor de su proyecto de vida.

Impulsar la formación integral de los estudiantes de tal forma que tengan habilidades y actitudes para: un aprendizaje para toda la vida, aprender a aprender, aprender a emprender y aprender a ser, reconocer que el proceso educativo puede desarrollarse en diversos lugares formales e informales y participar en el diseño de nuevas modalidades educativas con ellos como actores centrales.

#### 4.2.1.1.4 Funciones de la COPADI

- **Proporcionar servicio de tutoría** para los estudiantes que ingresan a la Facultad y alentar que la relación tutor-estudiante se conserve hasta la culminación de los estudios profesionales.
- **Atender a los estudiantes de alto rendimiento académico**, es decir, proporcionar atención especial a los alumnos que disponen del tiempo y la capacidad para sostener un ritmo de alta exigencia en su formación profesional.
- **Alentar el apoyo académico entre estudiantes** mediante acciones como sesiones de repaso de antecedentes, tratamiento de temas específicos, series de ejercicios para preparar exámenes y asesorías académicas individuales.
- **Atender a estudiantes de escasos recursos económicos** mediante acciones diversas como becas alimenticias, en correspondencia con su rendimiento académico.
- **Atender a estudiantes que trabajan** con tutorías especiales, que propicien una adecuada planeación de sus estudios y den seguimiento a su desempeño académico.
- **Investigar las causas** que propician la deserción y estudiar la instrumentación de acciones para la reincorporación a la vida académica.
- **Publicar el boletín mensual COPADI**, con problemas resueltos de las asignaturas que se imparten en la Facultad.

#### 4.2.1.1.5 Valores de la COPADI

Los valores necesarios en el personal académico que trabaja en la COPADI son:

- La congruencia entre el hablar y el actuar
- La responsabilidad y el compromiso
- La actitud de servicio y un espíritu crítico
- La creatividad y la innovación
- La formalidad y el sacrificio
- La ética y la honestidad
- La pasión y la emoción.

#### 4.2.1.2 TUTORÍA

##### 4.2.1.2.1 Tutoría Nueva Era.

La Tutoría para fines del programa "Nueva Era", se conceptualiza como un proceso de acompañamiento al estudiante por profesionales de la ingeniería y la docencia, para orientarlo y apoyarlo en su formación integral, al inicio, durante y al final de su carrera profesional, y en el desempeño eficiente de funciones y responsabilidades de los diversos momentos y compromisos propios de su aprendizaje.

Este programa es coordinado por la Secretaría de Apoyo a la Docencia (SAD), que orienta y supervisa la labor de los tutores, a través del trabajo conjunto de la Coordinación de Programas de Atención Diferenciada para Alumnos (COPADI), el Centro de Docencia "Ing. Gilberto Borja Navarrete", y la Coordinación de Evaluación Educativa, así como los coordinadores de tutoría por área y carrera.

El programa está organizado en tres etapas, cuyos objetivos son:

- Apoyar a los jóvenes en su integración al ambiente universitario y en el desarrollo de un plan de trabajo para su desempeño como estudiantes.

- Propiciar su formación integral con la adquisición y desarrollo de conocimientos, habilidades y actitudes para el desempeño profesional.

- Apoyar a los estudiantes en la culminación de su licenciatura y en su vinculación y proyección hacia el campo y áreas de desarrollo profesional.

Actualmente el programa tiene 219 profesores tutores en activo y cerca del 20% del estudiantado participa de manera consistente en estas labores. Se tiene el registro de que la tutoría impacta de manera positiva en el proceso de integración del estudiante al régimen escolar de la Facultad de Ingeniería y una cantidad significativa de estudiantes considera que la tutoría es un apoyo importante en su

desarrollo escolar.

#### 4.2.1.2.2 ¿Qué es la Tutoría?

**Tutoría.**

Es una estrategia educativa que se proporciona de manera personal a los estudiantes y que responde sus necesidades e intereses. Se trata de una acción complementaria, cuya importancia radica en orientar a los alumnos a partir del conocimiento de sus problemas y necesidades tanto académicas, como existenciales, a la vez que, saber cuáles son sus inquietudes y aspiraciones individuales y profesionales.

**Misión.**

Favorecer la atención individual del profesor al estudiante y promover un apoyo y una mejor comprensión por parte del tutor, hacia los problemas que enfrenta el alumno en lo que se refiere a su adaptación y al ambiente universitario; las condiciones individuales para un buen desempeño académico; a su formación integral que desarrolla valores, actitudes, habilidades, destrezas y aprendizaje significativo y, a su preparación para un devenir exitoso en su quehacer profesional.

**Objetivos del programa "Tutoría Nueva Era".**

- Establecer y consolidar un sistema de tutoría en el que el tutor considere al estudiante en lo individual.
- Lograr que la tutoría sea un derecho de todos los alumnos y un compromiso del profesor.
- Proporcionar a los estudiantes una orientación adecuada en lo que se refiere a:
    - Elección de carrera (definir o reorientar su vocación).
    - Formación Técnica.
    - Formación Humanística.
    - Crecimiento personal y problemática existencial.
    - Sentido de pertenencia a la UNAM.

- Lograr que los estudiantes sean creativos, reflexivos, innovadores, críticos y emprendedores.

- Inserción en el mercado de trabajo.

**Un Tutor es un Profesor.**

- Que tiene la voluntad y vocación para establecer un vínculo entre el estudiante y las diversas problemáticas académicas y existenciales que enfrenta durante su vida universitaria.
- Que facilita la integración del estudiante con la Universidad.
- Que demuestra al estudiante un genuino interés por su desempeño académico y por su crecimiento personal.
- Al que le interesa que el estudiante sea creativo, innovador y con espíritu libre y crítico.
- Que tiene empatía con el alumno, entendida como la capacidad de sentir y comprender las emociones del alumno como propias, mediante un proceso de plena y genuina identificación.

#### 4.2.1.2.3 Coordinadores.

| Nombre | Carrera |
|---|---|
| AMADOR TERRAZAS EDUARDO HILARIO | ING.INGENIERÍA GEOFÍSICA |
| DE LA TORRE SÁNCHEZ ROCÍO DRA. | INGENIERÍA PETROLERA |
| DEL GALLEGO GARCÍA MARIANO ING. | INGENIERÍA MECÁNICA |
| FLORES MEDERO NAVARRO BILLY ARTURO M. I. | INGENIERÍA MECATRÓNICA |
| GÓMEZ DAZA BENITO ING. | INGENIERÍA GEOMÁTICA |
| GUEVARA ANDRADE JORGE ING. | INGENIERÍA CIVIL |
| HARO RUIZ LUIS ARTURO M. I. | INGENIERÍA ELÉCTRICA ELECTRÓNICA |
| MOCTEZUMA FLORES MIGUEL DR. | INGENIERÍA EN TELECOMUNICACIONES |
| MORENO MAVRIDIS ELIZABETH ING. | INGENIERÍA INDUSTRIAL |
| RAMÍREZ FIGUEROA GABRIEL M. A. | INGENIERÍA DE MINAS Y METALURGIA |
| SALGADO TERÁN VIRGINIO ING. | INGENIERÍA GEOLÓGICA |
| VELÁZQUEZ MENA ALEJANDRO M. C. | INGENIERÍA EN COMPUTACIÓN |
| VÁZQUEZ SEGOVIA LUIS CÉSAR ING. | DIVISIÓN DE CIENCIAS BÁSICAS |

#### 4.2.1.2.4 Encuentros de Tutores.

Los Encuentros de Tutores son reuniones en las que los tutores intercambian puntos de vista sobre sus experiencias en la tutoría, y además se enteran de cómo valoran los estudiantes este quehacer académico. También constituyen magníficas oportunidades de integración, crecimiento y formación. Tienen como objetivo:

Ofrecer a los tutores las herramientas teórico-metodológicas que faciliten su desenvolvimiento en la tutoría, a través de encuentros, conferencias, cursos y otras actividades, entre las que se encuentran, las relacionadas con apoyos didácticos que les permitan comprender y fundamentar todas aquellas acciones que intervienen en el Programa.

### 4.2.1.3 PARA

#### 4.2.1.3.1 Objetivos del PARA.

Proporcionar al tutor de los alumnos que pertenecen al Programa de Alto Rendimiento Académico de la Facultad de Ingeniería, un conjunto de lineamientos que le permitan apoyar al estudiante del Programa para lograr un mejor desempeño académico, profesional y personal**.**

#### 4.2.1.3.2 Definición del PARA

El Programa de Alto Rendimiento Académico (PARA) de la Facultad de Ingeniería de la UNAM es una acción académica de nivel institucional, que reconociendo que algunos estudiantes tienen la posibilidad de contender exitosamente con una carga académica superior al promedio, establece actividades de enseñanza y aprendizaje adicionales para aquellos estudiantes que de manera voluntaria manifiesten su deseo de participar en el mismo.

#### 4.2.1.3.3 Justificación del PARA

El Programa de Alto Rendimiento Académico en la Facultad de Ingeniería comenzó en el año de 1992, desde entonces ha intentado formar a ingenieras e ingenieros de excelente calidad académica y humana. Si bien esto se ha logrado en

su mayoría, estamos en un momento magnífico para hacer un alto y una evaluación de algunos problemas que se han presentado en este camino. El papel del tutor en este proceso se convierte en un promotor importantísimo de solución y logro para optimizar el PARA. El tutor del PARA es un profesor o un investigador de carrera que tiene la voluntad de acompañar al estudiante en su compromiso con el estudio de su carrera, con sus materias adicionales, con su formación integral, con su sociedad y como un profesional consiente de la importancia de ser un digno representante de su Facultad, de su UNAM y de su país.

#### 4.2.1.3.4 Objetivos del Programa

Los objetivos del Programa de Alto Rendimiento Académico de la Facultad de Ingeniería son los siguientes:

- Apoyar a estos estudiantes para que empleen el máximo de su capacidad en beneficio de su formación profesional y desarrollo integral.
- Prepararlos para, en su caso, continuar sus estudios en las mejores universidades del mundo y para desempeñarse con la máxima productividad en ambientes de alta competencia.
- Contar con candidatos que permitan desarrollar innovaciones educativas que contribuyan a mejorar la calidad del proceso enseñanza-aprendizaje.
- Impulsar la superación académica promoviendo la colaboración de los estudiantes del programa para apoyar en lo académico a otros estudiantes que lo requieran.

Y una noción que debe formar parte del ideario del Programa de Alto Rendimiento Académico es:

Formar profesionales conscientes de que el conocimiento conduce a la sabiduría si se adquiere con sencillez, y que la sabiduría los conducirá a enfrentar las diversas problemáticas que enfrenta el país, en la búsqueda de la justicia social y por ende el mejoramiento de la calidad de la vida.

#### 4.2.1.3.5 Descripción del Programa

En su diseño y ejecución el programa incluye diversos componentes, los principales son: asignaturas adicionales, inglés obligatorio (u otro idioma si éste ya se domina), tutoría obligatoria, actividades intersemestrales, proporcionar apoyo académico a sus compañeros a través de asesorías académicas e individuales, participación en proyectos de desarrollo tecnológico y de investigación o de desarrollo empresarial, así como becas de apoyo económico.

#### 4.2.1.3.6 Ingreso al PARA

Para ingresar al programa los estudiantes requieren ser regulares y haber obtenido un promedio igual o mayor a 9.0 durante los primeros dos semestres de la carrera, así como manifestar voluntariamente y por escrito su aceptación y compromiso. El estudiante deberá elaborar una carta de exposición de motivos por escrito que **llevará el visto bueno de su tutor**. Se aceptarán, eventualmente, alumnos de cuarto o quinto semestre, siempre y cuando reúnan con los requisitos establecidos.

#### 4.2.1.3.7 Condiciones de Permanencia en el PARA

El criterio de permanencia es el avance escolar regular y con alto rendimiento, que de manera operativa se entiende como:

Cursar y acreditar como mínimo las asignaturas que para cada semestre señala el plan de estudios de la carrera correspondiente, en examen ordinario, con una calificación de 8.0 (permitiendo tener una calificación menor pero aprobatoria, en tres ocasiones como máximo) y manteniendo un promedio escolar anual mayor o igual a 9.0.

#### 4.2.1.3.8 Compromisos y Actividades dentro del Programa

- Cursar y acreditar una asignatura adicional cada semestre y aprobarla con promedio mínimo de 8.00.
- El alumno deberá demostrar cada semestre que, en cualquier modalidad, estudia el idioma inglés, hasta dominarlo y presentar un certificado que acredite el total conocimiento del idioma.
- Deberá asistir al menos a ocho sesiones de tutoría en el semestre y cumplir con los acuerdos que ahí se tomen.
- Participar en un proyecto dirigido por un académico o profesionista reconocido.

El proyecto puede ser en cualquiera de las siguientes áreas:

- **En el área docente:**
  Conducción de asesorías y clases de ejercicios para estudiantes de los primeros semestres y /o elaboración de ejercicios.

- **En el área de investigación:**
  Participación en proyectos de investigación o desarrollo tecnológico en la facultad o institutos o centros afines.

- **En el área de trabajo.**
  Participación en proyectos de ingeniería en el campo de trabajo profesional.

El alumno se comprometerá a realizar actividades adicionales propias del programa durante los periodos intersemestrales.

#### 4.2.1.3.9 Objetivo de la tutoría del PARA

- Acompañar al estudiante en su formación académica.
- Facilitar a los estudiantes la permanencia en el PARA
- Contribuir en la formación de una conciencia de trabajo en equipo
- Favorecer la integración con su comunidad
- Resolver la formación integral cubriendo todas las necesidades humanas
- Consolidar una conciencia social, innovadora, crítica y creativa

- Formar en la obtención de un criterio propio en la solución de problemas personales y nacionales
- Fomentar un sentido de pertenencia e identidad universitaria, nacional y universal
- Lograr que el estudiante tenga su propio plan de vida.
- Motivar al alumno a que realice su servicio social, orientándolo en los diferentes proyectos que existen en la UNAM.

#### 4.2.1.3.10 Consejo Coordinador y Dictaminador del Programa de Alto Rendimiento Académico (PARA) de la Facultad (CCD)

Constituido el 24 de enero de 2001

INTEGRANTES:

Dr. Jesús Manuel Dorador González.
M. en A. Miguel Eduardo González Cárdenas.
M. en C. Ernesto Riestra Martínez.
M. en I. Yukihiro Minami Koyama.
M. en I. Rodrigo Gutiérrez Arenas.
Lic. Pablo Medina Mora Escalante.

Funciones:

- Fungir como órgano de consulta del Director, estudiar y proponer políticas generales para el desarrollo del PARA, así como los lineamientos para su instrumentación, operación y evaluación; coordinar todas las acciones de diseño, instrumentación, operación y evaluación del PARA.
- Estudiar, proponer, coordinar, instrumentar, operar y evaluar los cambios que se realicen en el PARA.
- Admitir a los estudiantes al PARA.
- Dar bajas temporales con objeto de recuperar y obtener el promedio exigido y para acreditar nuevamente alguna asignatura o enmendar alguna falta disciplinaria, para lo cual deberá estudiarse el caso conjuntamente estudiante y tutor y, en el caso de una asignatura, la investigación correspondiente con el profesor y las autoridades de la división a la que pertenezca.

Las decisiones del CCD serán por consenso y, de no ser esto posible, por mayoría simple de votos (En caso de un "empate", quién preside tendrá voto de calidad).

Reuniones:

Se realiza una sesión ordinaria cada seis meses y extraordinarias cuando se considere necesaria. Las reuniones ordinarias se realizan en febrero y septiembre de cada año y la fecha se fija de acuerdo con las agendas de la Dirección y/o la Secretaría General.

#### 4.2.1.3.11 Ventajas de pertenecer al PARA y terminar en éste la Carrera

- Gozar del prestigio de pertenecer al Programa de Alto Rendimiento Académico de la mejor Facultad de Ingeniería de la mejor Universidad del país.
- Recibir una formación académica que le permita ser líder en su área y poder competir profesionalmente, a nivel nacional e internacional.
- Recibir una formación cultural y social que le permita una realización plena como ser humano.
- Contar con un tutor toda la carrera.
- Acceder a los mejores maestros y horarios con la ayuda y asesoría de su tutor.
- Cursar asignaturas adicionales a las establecidas en los planes y programas de estudio de su carrera, lo que enriquecerá su formación académica.
- Trabajar con investigadores de centros e institutos de investigación de la UNAM.
- Realizar actividades académicas extracurriculares como cursos cortos, seminarios y programas informáticos.
- Adquirir un pleno dominio del idioma inglés.
- Recibir cursos y seminarios de preparación para su quehacer profesional.
- Realizar estancias empresariales en México y en el extranjero.
- Visitar institutos de investigación, otras universidades y lugares de interés.

- Conseguir apoyos económicos para congresos y cursos relacionados con su carrera.
- Asistir, en el último semestre de la carrera, a una feria de trabajo conocida como Encuentro de Empresarios con el Talento Universitario en la que recibirá ofertas de trabajo de las mejores empresas y consultorías del país.
- Recibir, al terminar la carrera, un diploma de la UNAM con alto valor curricular.
- Entrar en contacto con otras universidades nacionales y extranjeras para realizar estudios de posgrado.
- Visitar lugares del país donde tomará conciencia de la problemática de marginación y pobreza que se vive en ellos, para así conocer y comprometer su futuro quehacer profesional con la realidad nacional para transformarla.

#### 4.2.1.3.12 Sociedad de Egresados del Programa de Alto Rendimiento Académico (PARA) de la Facultad de Ingeniería

La Sociedad tiene la intención de reunir a los ingenieros e ingenieras que como alumnos pertenecieron al PARA y concluyeron en éste sus estudios de licenciatura, y su objetivo es colaborar moralmente con el Programa, con la Facultad y con la Universidad.

Para alcanzar su objetivo, la sociedad se propondrá: Obtener el apoyo moral y material de sus socios en primer lugar, y después, de las personas e instituciones que deseen dar su colaboración; propiciar el acercamiento entre el Programa de Alto Rendimiento Académico y el sector productivo; propiciar el acercamiento de nuevo alumnos de la Facultad al Programa de Alto Rendimiento Académico; colaborar con el Programa mediante conferencias, mesas redondas, seminarios y otras actividades que motiven a sus integrantes estudiantes a seguir adelante y culminar sus estudios de licenciatura dentro del Programa; asesorar a los egresados para estudios de posgrado y para su inserción en el mercado de trabajo.

#### 4.2.1.3.13 Actividades del PARA

Si bien sabemos que los tutores del PARA ya cuentan con una experiencia de varios años**,** quisimos en esta guía sugerir algunas actividades:

- Guía de lecturas.

- Seguimiento del desempeño académico.
- El conocimiento del idioma más pertinente para su profesión.
- Buscar apoyo académico para algunas materias.
- Invitación a actividades culturales.
- Visitas dentro de la universidad.
- Visitas y prácticas en el conocimiento de la situación de su ciudad y de otros Estados de la República.
- Canalización, en caso necesario a otras instancias de ayuda al estudiante como las asesorías psicopedagógicas que ofrece la COPADI.

**Actividades Académicas.**
**Actividades Socio-Culturales.**
**Asignaturas Adicionales del PARA**.

#### 4.2.1.4 Publicaciones

##### 4.2.1.4.1 Boletín COPADI

El Boletín COPADI es una publicación mensual, su contenido consta de entre tres a seis problemas resueltos de las diferentes asignaturas, sobre todo de ciencias básicas.

#### 4.2.1.5 Asesorías Psicopedagógicas

##### 4.2.1.5.1 Objetivos

La **Asesoría Psicopedagógica** tiene como objetivo brindar atención especializada al estudiante en el ámbito académico y/o interpersonal, buscando incidir en su actitud y disposición hacia el aprendizaje.

El servicio de asesoría brinda, **orientación individual y personalizada** en las áreas afectivo conductual, con el fin de promover habilidades y estrategias que le permitan mejorar su rendimiento académico.

El **Asesor** es un profesional, que con diferentes técnicas y estrategias aborda las diversas situaciones que llegan a presentar los estudiantes, en las áreas emocional y conductual que afectan el desempeño académico.

Para acceder al servicio se requiere que **los estudiantes**, asistan al cubículo de asesorías psicopedagógicas en el interior de la **Biblioteca Enrique Rivero Borrell.** Si el asesor psicopedagógico se encuentra disponible, le atiende de inmediato, en caso contrario puede acudir a la COPADI y se le canaliza con otro asesor.

Entre los **beneficios** para el estudiante, además de la atención individualizada, se puede mencionar mejor conocimiento de sí mismo y mejor manejo de su situación emocional, de forma tal que no interfiera en su desarrollo académico. También se le apoya para elaborar estrategias que le permitan optimizar su forma y tiempo de estudio.
*"Existen varias formas de resolver los problemas personales, pero sólo tú puedes llegar a detectarlos y trabajar en ellos".*

Si requieres del servicio, los horarios de atención y nombres de los asesores, los encuentras en el cubículo de asesorías en la biblioteca o en esta página. Para mayores informes acude a la COPADI, que se ubica en la planta alta del auditorio Sotero Prieto de la División de Ciencias Básicas.

**4.2.1.5.2** Temáticas

| ÁREA | TEMAS |
|---|---|
| **Institucional** | - Contribuir en el desarrollo de tesis de licenciatura.<br>- Proporcionar información sobre las carreras.<br>- Favorecer la integración de los alumnos a la institución. |
| **Vocacional** | - Dificultades de orientación vocacional.<br>- Orientación sobre cambio de carrera. |
| **Escolar** | - Fortalecer estrategias, hábitos y técnicas de estudio.<br>- Se revisan situaciones relacionadas con reprobación o rezago escolar. |
| **Psicosocial** | - Se revisan actitudes, conductas y habilidades que favorecen el autoconocimiento, la autoestima y la comunicación.<br>- Conflictos asociados con relaciones interpersonales y de pareja.<br>- Problemas de tipo familiar. |
| **Otros** | - Problemas diversos como salud y adicciones. |

**4.2.1.5.3** Integrantes

**INTEGRANTES DEL PROGRAMA DE ASESORÍAS PSICOPEDAGÓGICAS**

| ASESOR PSICOPEDAGÓGICO |
|---|
| Ing. Alfredo Arenas G. |
| Lic. Claudia Margarita Pérez Ruiz. |
| Mtra. Ana G. García y C. |
| Lic. Javier Gómez R. |
| Lic. Ma. de la Paz González A. |
| Lic. Claudia Loreto Miranda. |
| Lic. Pablo Medina Mora. |
| Lic. Griselda Núñez Núñez. |
| Mtra. Martha Rosa del Moral Nieto. |
| Lic. Guadalupe Salazar Hernández. |

### 4.2.1.6 Cursos Extracurriculares

Con el **objetivo** de impulsar el desarrollo humano integral de los alumnos de la F.I., la 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 desenvolvimiento psicosocial y favorecer aquellas que 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.

Para acceder **al servicio** se requiere que los estudiantes acudan a la COPADI, durante el mes anterior al inicio del intersemestre, para que se les inscriba en los cursos que les interesen.

Para mayores **informes** acude a la COPADI, que se ubica en la planta alta del auditorio Sotero Prieto de la División de Ciencias Básicas (zona sur).

#### 4.2.1.7 Inglés

El aprendizaje de otro idioma, especialmente el inglés, además de ser en muchos casos un requisito de titulación, se ha convertido en un factor básico de la formación profesional de cualquier estudiante.

Con el fin de colaborar contigo en este aspecto de tu formación académica, la Secretaría de Servicios a la Comunidad, a través de la DGOSE, cuenta con el programa de apoyo para el estudio de idioma. Mediante este programa es posible obtener descuentos en la inscripción y/o colegiatura de cursos de idiomas ofrecidos por instituciones privadas.

**Requisitos**

- Ser alumno de la UNAM.
- Tener promedio mínimo de 8 o 9 según el Instituto otorgante del descuento.

- Ser alumno regular, es decir, haber cubierto un número de créditos equivalente al previsto en el plan de estudios, de acuerdo con el número de semestres o años cursados.

**Solicitudes**

Para solicitar este apoyo deberás de acudir al Centro de Orientación Educativa de la Dirección General de Orientación y Servicios Educativos con los siguientes documentos:

- Credencial de la UNAM
- Historia Académica
- Comprobante de inscripción

Si cumples con los requisitos, te invitamos a que consultes las diversas opciones que ofrecen y elijas el Instituto que más se adapte a tus necesidades.

### 4.2.2 Desarrollo del Portal COPADI

Como anteriormente se mostró en el diagrama, la estructura que se ha contemplado para la página principal de la coordinación queda de la siguiente manera:

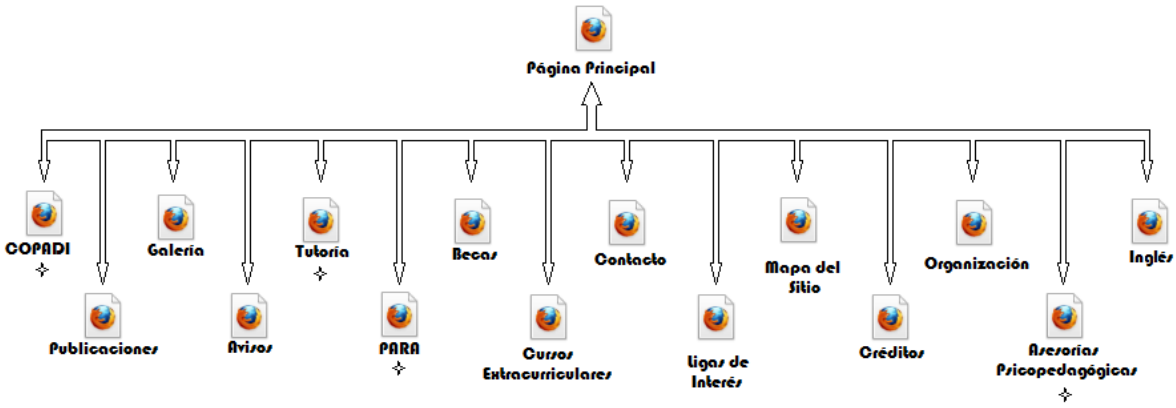



Figura 7. Estructura general de la Página Principal COPADI

Donde se agrupa la diferente información que será presentada al usuario.

Apartado COPADI:

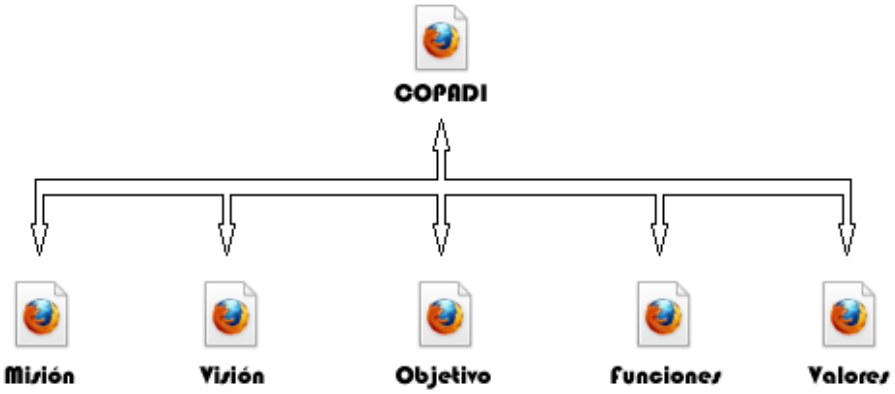



Figura 8. Estructura COPADI

Apartado Tutoría:

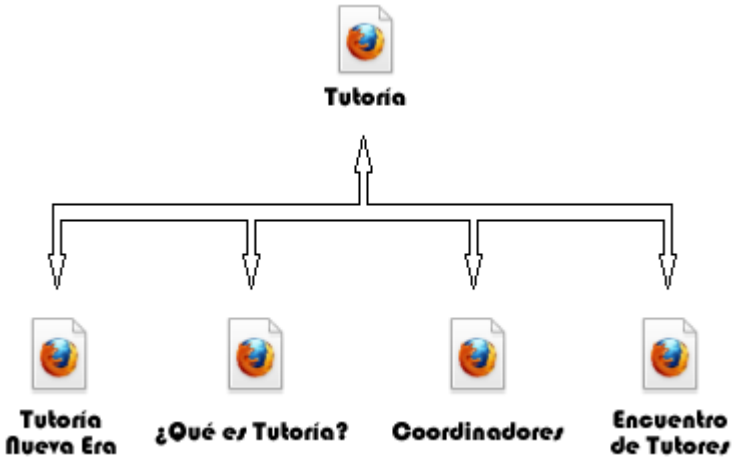



Figura 9. Estructura Tutoría

Apartado PARA:

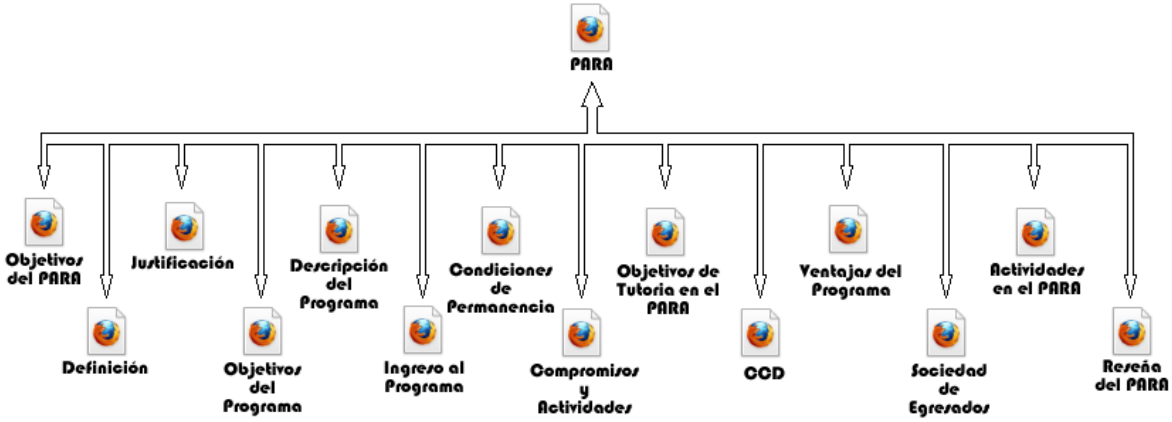



Figura 10. Estructura PARA

Apartado Asesorías Psicopedagógicas:

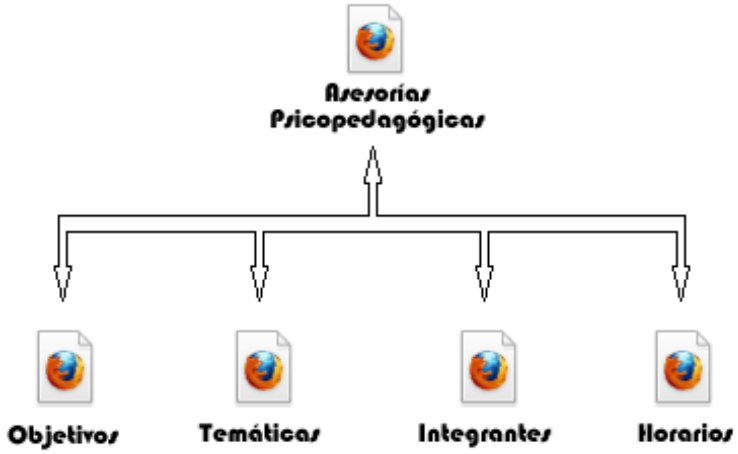



Figura 11. Estructura Asesorías

**4.2.2.1** Efecto de Menú Izquierdo

Este menú está diseñado para implementarse dinámicamente en todo el sitio, es decir, con un solo cambio en un archivo automáticamente cambiará en todos los documentos HTML o JSP según sea el caso.

Para su desarrollo se utilizó la librería de JQuery que podemos descargarla directamente de la página principal (http://code.jquery.com/jquery-1.6.1.min.js), así mismo utiliza el código JavaScript para el efecto transparente y desvanecimiento, el código se encuentra en el archivo http://copadi.fi-c.unam.mx/js/efectoMenuVertical.js, así mismo el archivo http://copadi.fi-c.unam.mx/js/menuVertical.js que es el que contiene los rótulos de los menús así como los links a donde redirige cada uno de los enlaces. Así mismo, tenemos los estilos que se encuentran en la archivo http://copadi.fi-c.unam.mx/css/plantillaWeb.css en donde se define al color del menú, el tipo de letra, así como ubicación de los submenús.

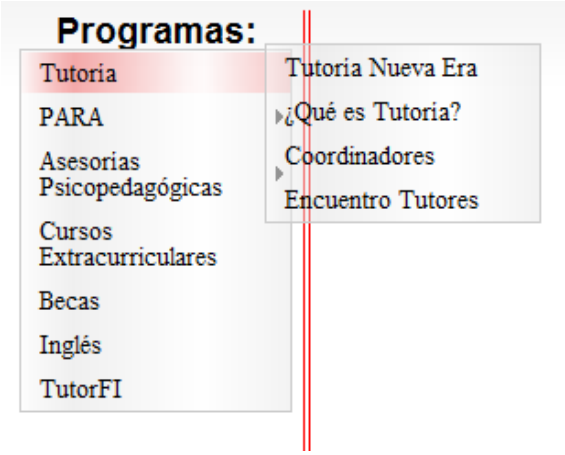



Figura 12. Menú Principal

Como anteriormente se explicó, la programación modulada permite reutilizar el código de una forma sencilla y rápida, por lo que solo es necesario usar esos 3 archivos para su implementación dentro de cualquier documento WEB, y para ello solo es necesario integrar estas líneas de código al documento:

```
<div id="menu-izquierdo"><h2>Programas:</h2>
<script src="js/menuVertical.js" type="text/javascript"></script>
</div>
```

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

Teniendo que cada documento HTML o JSP en nuestro caso se realiza de la siguiente forma:

```
<script type="text/javascript" src="js/scriptPrincipal.js"></script>
```

Donde esta línea de código se coloca dentro de la etiqueta <head></head>. Dicho archivo contiene cada uno de los demás archivos de código JavaScript.

#### **4.2.2.2** Efecto de Menú Horizontal

De la misma forma que el menú Izquierdo, este también cuenta con sus respectivos archivos de programación de JavaScript y CSS. Vuelvo a mencionar todos estos archivos están disponibles en el Portal COPADI.

Para este menú tenemos los siguientes archivos:

*http://copadi.fi-c.unam.mx/js/bannerPrincipal.js*
*http://copadi.fi-c.unam.mx/css/plantillaWeb.js*

y su uso es de la siguiente manera:

```
<div id="menuH">
    <li id="current"><a href="principal.jsp"><span>Principal</span></a></li>
    <li><a href="copadi.jsp"><span>COPADI</span></a></li>
    <li><a href="publicaciones.jsp"><span>Publicaciones</span></a></li>
    <li><a href="#"><span>Galer&iacute;a</span></a></li>
    <li><a href="avisos.jsp"><span>Avisos</span></a></li>
</div>
```

Cabe mencionar, que dentro del archivo *bannerPrincipal.js* se define cada uno de los botones que aparecerán en este menú, sencillamente ingresando cada nombre que se requiera sin la necesidad de realizar alguna otra modificación.

### 4.2.2.3 Efectos de Contenido en los apartados

Dentro de cada apartado, se aplica un efecto que mostrara toda la información rotando cada una de ella de acuerdo a un intervalo de tiempo, es decir, dentro de cada página HTML o JSP se mostrara la demás información que se encuentra dentro de cada subapartado, quedando de la siguiente manera para cada uno de los apartados antes listados:

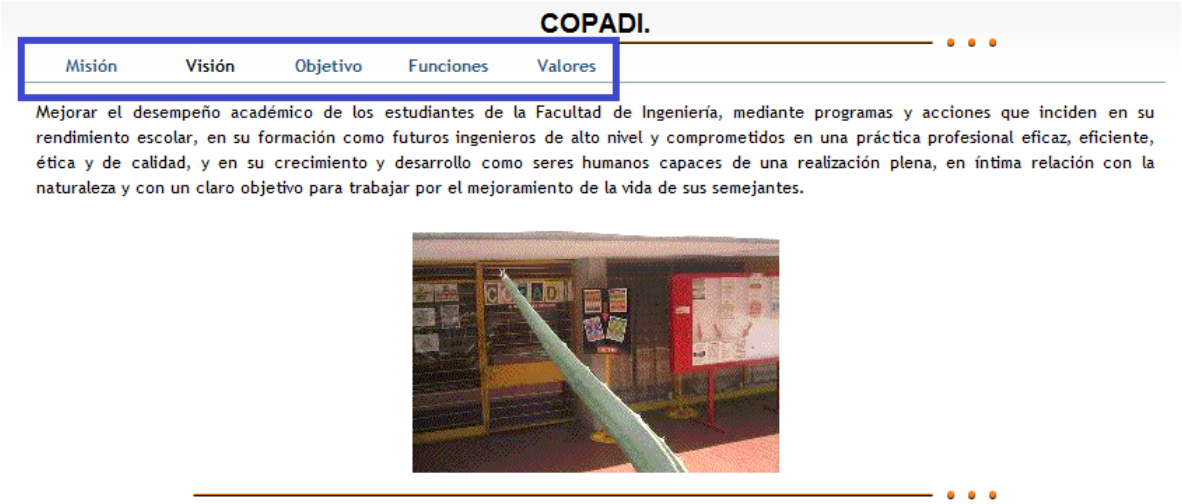



Figura 13. Apartado COPADI

**4.2.2.4** Efecto de Transición en fotografías y anuncios

Para la presentación de las fotografías y algunos avisos dentro del portal se ha implementado un efecto de transición que hace más elegante la presentación de fotografías dentro del Portal.

Para este efecto son necesarios estos archivos:

```
<script src="js/jquery.js" type="text/javascript"></script>
<script src="js/jquery.easing.js" type="text/javascript"></script>
<script src="js/galeriaFotos1.js" type="text/javascript"></script>
<script src="js/jquery.cycle.all.min.js" type="text/javascript"></script>
<link rel="stylesheet" href="css/galeriaFotos1.css" type="text/css" />
```

Los cuales definen el estilo de color y el tipo de transición.

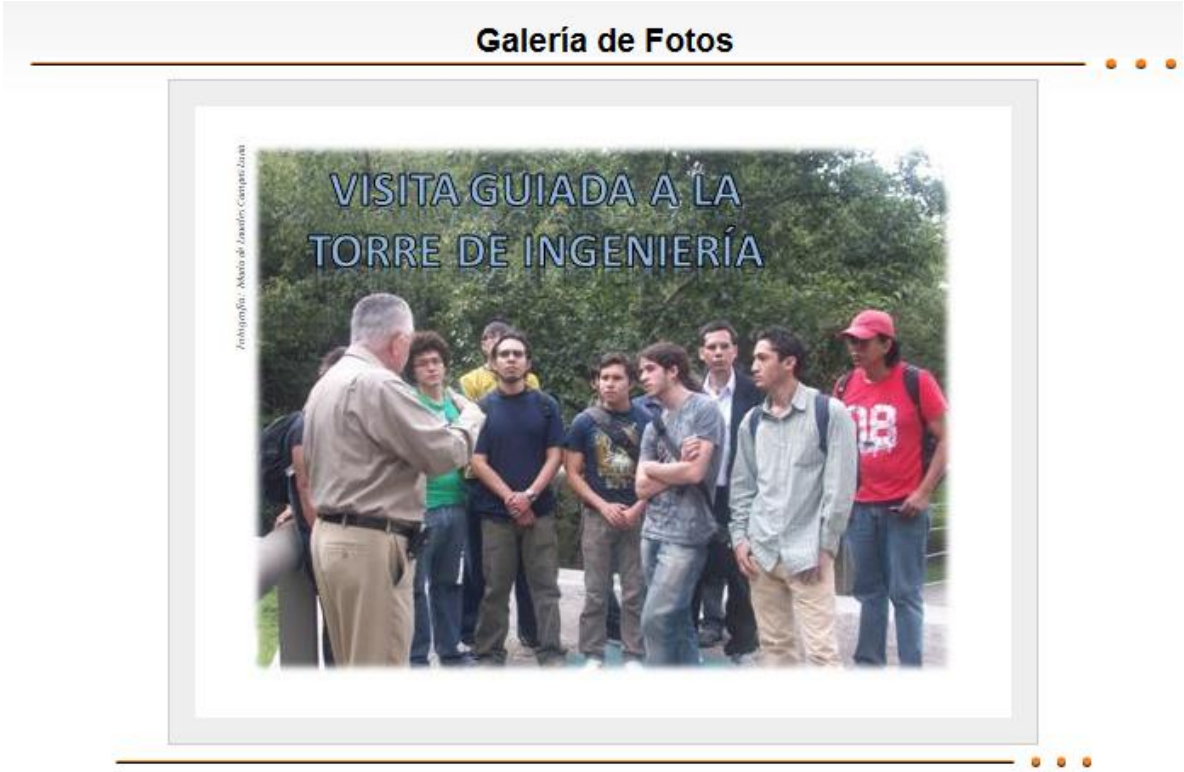



Figura 14. Galería de Fotos

**4.2.2.5** Efecto de Organización

Este efecto nos muestra un portafolio de imágenes que al pasar el puntero del ratón, nos despliega la imagen de cada uno de los integrantes del personal de la Coordinación. Este efecto es llamado KWiks obtenido de *http://www.jeremymartin.name/examples/kwicks.php?example=6* dentro de la suite de JQuery y la implementación del desarrollo Mootools. Para realizar este efecto se ayuda de los siguientes archivos:

```
<link rel="stylesheet" type="text/css" href="kwicks/kwicks.css">
<script src="kwicks/jquery-1.js" type="text/javascript"></script>
<script src="kwicks/jquery_002.js" type="text/javascript"></script>
<script src="kwicks/jquery.js" type="text/javascript"></script>
<script type="text/javascript">
      $().ready(function() {
      $('#organizacion .kwicks').kwicks({
           max: 390,
           spacing: 2,
          duration: 1500,
          easing: 'easeOutBounce'
      });
      });
</script>
```


Figura 15. Organización

Y para cada una de las fotografías se emplea de la siguiente forma:

```
<ul class="kwicks">
    <li id="kwick1"></li>
    <li id="kwick2"></li>
    <li id="kwick3"></li>
    <li id="kwick4"></li>
    <li id="kwick5"></li>
    <li id="kwick6"></li>
    <li id="kwick7"></li>
    <li id="kwick9"></li>
    <li id="kwick8"></li>
</ul>
```

**4.2.2.6** Contador de visitas

Este contador está hecho con JAVA, para su implementación se creó un bean, que es un componente de programación que tiene la particularidad de ser reutilizable y así evitar la tediosa tarea de programar los distintos componentes uno a uno, tales beans serán utilizados en la programación del sistema TutorFI. Para realizar el conteo se realizó el siguiente código:

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package BeansTutorFI;
import java.io.*;
/**
 *
 * @author kike
 */
public class Contador {
   private int contador=6984;

   public Contador() throws IOException{
      try {
         //Bloque que lee el Archivo, debe de existir
         File f = new File( "../contador.txt" );
```

```
        BufferedReader entrada = new BufferedReader( new FileReader( f ) );
        if (f.exists()){
           String texto = entrada.readLine();
           contador = Integer.parseInt(texto);
        }
     }
     catch(java.io.FileNotFoundException fnfex) {
      System.out.println("se presentó el error: "+fnfex.toString());
     }
  }
  public int getValue() throws FileNotFoundException, IOException {
     return(contador);
  }

  public void setValue(int value) throws IOException {
     this.contador = value;
     FileWriter fw = new FileWriter("../webapps/ROOT/contadorPrincipalCOPADI.txt");
        BufferedWriter bw = new BufferedWriter(fw);
        PrintWriter salida = new PrintWriter(bw);
        salida.println(contador);
        salida.close();
  }

  public void incValue() throws IOException
        {
     contador++;
     //Bloque que actualiza
        FileWriter fw = new FileWriter("../webapps/ROOT/contadorPrincipalCOPADI.txt");
        BufferedWriter bw = new BufferedWriter(fw);
        PrintWriter salida = new PrintWriter(bw);
        salida.println(contador);
        salida.close();
        }
}
```

En donde se va haciendo el conteo de las visitas, para ello se genera un archivo en el que se ira almacenando el número, y este ira incrementando cada vez que un usuario visite la página.


Figura 16. Contador de Visitas

#### 4.2.2.7 Pie de página

De acuerdo a los lineamientos que marca la DGTIC para sitios web institucionales, es definido el pie de página de la siguiente forma:

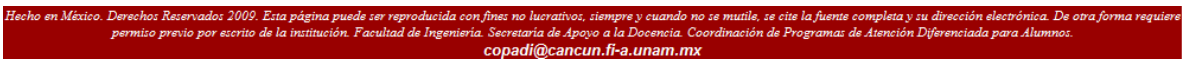



Figura 17. Pie de Página

Con la leyenda:

*" Hecho en México. Derechos Reservados 2009. Esta página puede ser reproducida con fines no lucrativos, siempre y cuando no se mutile, se cite la fuente completa y su dirección electrónica. De otra forma requiere permiso previo por escrito de la institución. Facultad de Ingeniería. Secretaría de Apoyo a la Docencia. Coordinación de Programas de Atención Diferenciada para Alumnos.*
*copadi@cancun.fi-a.unam.mx"*

## 4.3 Sistema TutorFI

Para el desarrollo del sistema, se ha utilizado la plantilla del diseño de la página para que sea más uniforme el diseño de cada una de los módulos que sean desarrollados para el portal, en este caso se describirá cada componente del sistema TutorFI.

Así mismo, la base de datos será almacenada en el manejador PostgreSQL (Ver apéndice D).

### 4.3.1 Diseño de la Base de Datos

El sistema TutorFI requiere de una base de datos sólida, confiable, integra y con una amplia disponibilidad para la cantidad de usuarios que requiere atender. Así mismo, requiere de una alta seguridad para que los datos sean íntegros.

Como anteriormente se mencionó, el esquema de la base de datos con la cual se contaba, no permitía realizar el almacenamiento de varias generaciones de estudiantes, así mismo estaba limitada a trabajar con la generación actual.

Con el nuevo esquema de la base de datos, se tiene la capacidad de almacenar la información de los estudiantes de cada generación (a partir de la generación 2008). También se almacena la información de los diferentes tutores que imparten las sesiones de tutoría, los coordinadores de las diferentes divisiones de la Facultad de Ingeniería y los distintos bloques generados generación tras generación.

Para el estudiante debe ser almacenada su información personal como lo es el número de cuenta, nombre completo, teléfono, etc. Además existe información de estudiantes con becas, programas, exámenes diagnósticos de cada uno de los estudiantes, calificaciones del primer semestre, calificaciones de las materias del semestre anterior al cual estén cursando, así como el número de materias y laboratorios inscritos actualmente. También se cuenta con evaluaciones realizadas a los estudiantes por sus tutores. Cabe mencionar que cada estudiante es asignado a un bloque.

Para los tutores se tiene su información personal que es necesaria para los alumnos en caso de necesitar tener contacto y poder localizarlos sin ningún problema, estos son nombre completo, división, ubicación, teléfono y una breve semblanza, así como su fotografía. A cada tutor se le asignan estudiantes de su misma carrera, realiza evaluaciones al alumno, almacena información de sesiones grupales que se tienen al inicio de cada semestre para la nueva generación. También se almacena información de sesiones individuales durante el transcurso de la carrera de los estudiantes. Cada tutor realiza una evaluación semestral para evaluar a su coordinador, a las COPADI y una autoevaluación. Así mismo, los tutores pueden evaluar al sistema para hacer una retroalimentación y poder mejorar el sistema en cuanto a los servicios y herramientas que brinda el sistema. Uno o dos tutores son asignados a cada bloque cada semestre.

#### **4.3.1.1** Modelo de Datos

Para realizar el almacenamiento de dicha información, que elimine la

redundancia en los datos y que cuenten con consistencia e integridad, se definen 19 tablas que almacenaran la información necesaria para un óptimo y buen funcionamiento del sistema.

#### **4.3.1.1.1** Modelo Lógico: Diagrama entidad – relación

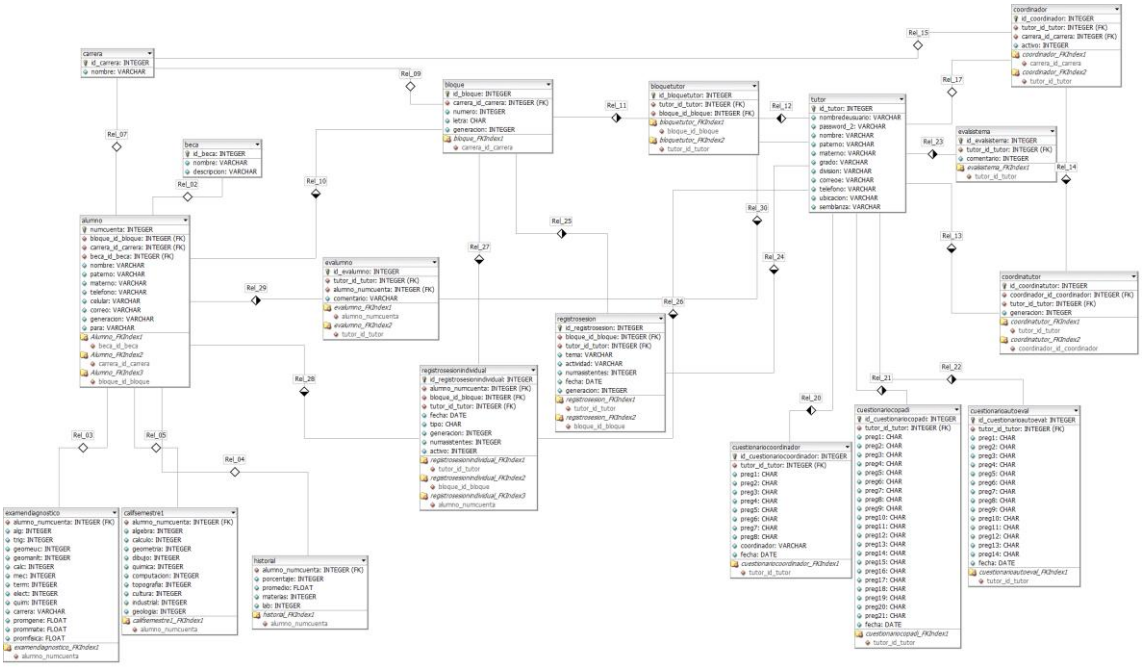


Figura 18. Esquema de la Base de Datos

#### **4.3.1.1.2** Diccionario de Datos.

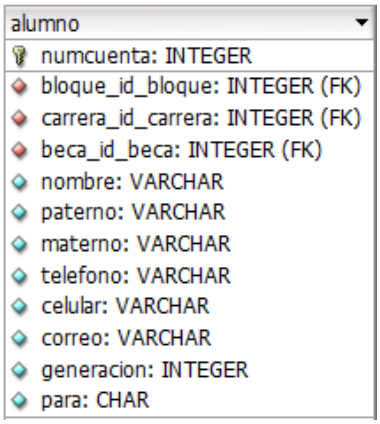
Para el modelo de la base de datos, contamos con 19 tablas que contienes la información precisa y necesaria para el sistema, cada uno de sus atributos con diferente tipo de variable para su uso, entre estos tenemos:

- Varchar (n) que define un atributo de caracteres variable
- Integer que define un tipo de dato entero
- Date que permite estableces la fecha junto con la hora
- Char define un atributo de un solo caracter
- Float

- Define un tipo de dato real

A continuación aparecen listadas las entidades que se encuentran en el diagrama anterior:

- Alumno



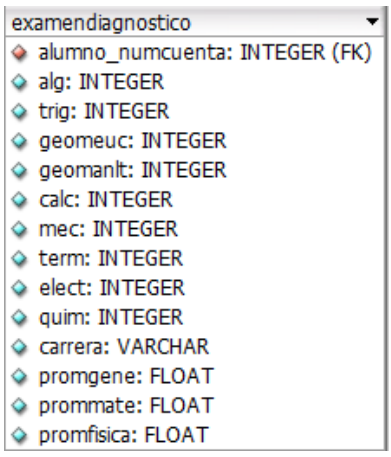

La entidad alumno cuenta con el identificador,llave primaria o PK "numcuenta" que identifica a cada uno de los estudiantes. Las llaves foraneas o FK son las que hacen referencia a las otras entidades donde son PK.

El atributo "para" es de tipo char que solo permitira indicar si es o no es estudiante perteneciente al Programa PARA.

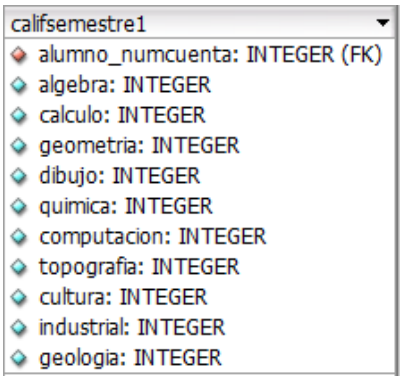
- Examendiagnostico

Esta entidad contiene las calificaciones del Examen Diagnóstico, sus atributos corresponden a las siguientes materias:

- alg – Álgebra
- trig – Trigonometría
- geomeuc – Geometría Euclidiana
- geomanlt – Geometría Analítica
- calc – Cálculo
- mec - Mecánica
- term – Temodinamica
- elect – Electricidad
- quim – Química
- promgene – Promedio General
- prommate – Promedio de Matemáticas
- promfisica – Promedio de Física



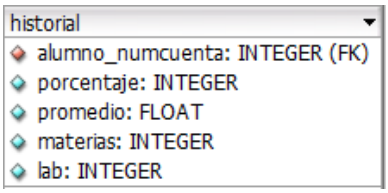
- Califsemestre1



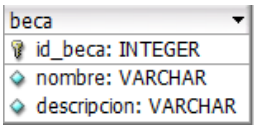

Esta entidad contiene las calificaciones del primer semestre que curso el estudiante dentro de la facultad.

- Historial

La entidad historial almacena el porcentaje de créditos y el promedio que lleva el estudiante hasta el semestre anterior al que cursa, así como las materias y los laboratorios inscritos en su semestre actual.



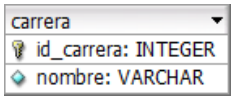

- Beca



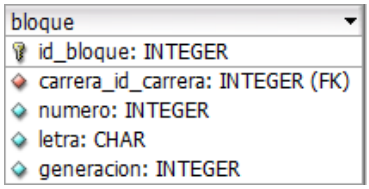

La entidad beca es un catálogo que almacena los diferentes tipos de becas que se otorgan en la Facultad de Ingenieria.

- Carrera

La entidad carrera también es un catálogo que almacena las 12 carreras que ofrece la Facultad.



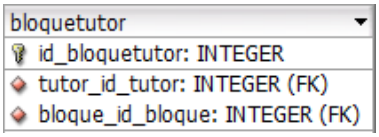

- Bloque



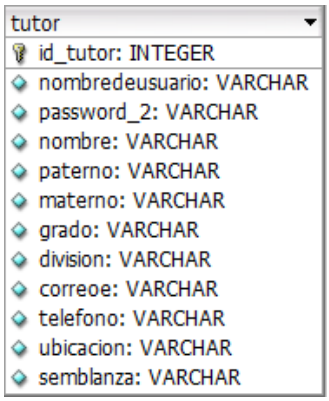

Esta entidad almacena cada uno de los bloques generados cada generación, así como la carrera asignada y la generación correspondiente.

- Bloquetutor

Esta entidad permite hacer la asociación de varios tutores a un bloque, así como permitir que se haga el histórico de los tutores asignados a bloques de todas las generaciones.



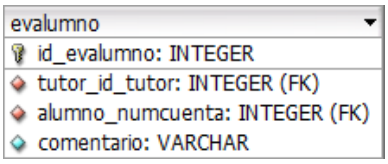

- Tutor



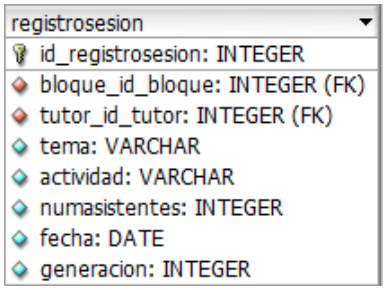

Esta entidad almacena la información de todos los tutores que son miembros del programa de tutoría, así como los coordinadores de cada una de las carreras que ofrece la Facultad.

- Evalumno

Esta entidad almacena las evaluaciones realizadas por los tutores a sus estudiantes de tutoría.



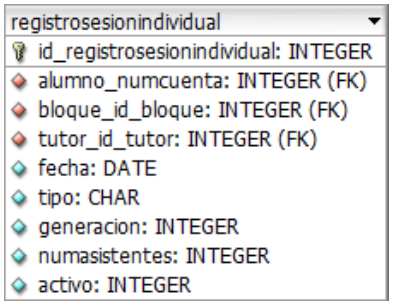

- Registrosesion

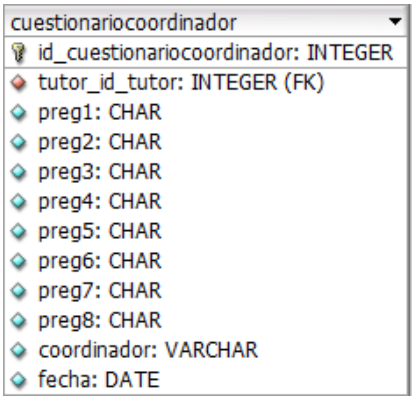



Esta entidad almacena la información ingresada por los tutores de las sesiones grupales que tienen con cada generación al inicio del primer semestre dentro de la facultad.

- Registrosesionindividual

Esta entidad almacena el Registro de las Sesiones Individuales que tienen los tutores durante el transcurso de la carrera de los estudiantes asignados.



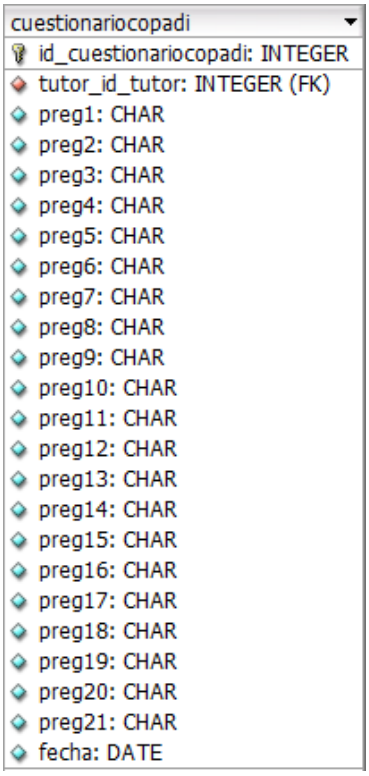




- Cuestionariocoordinador

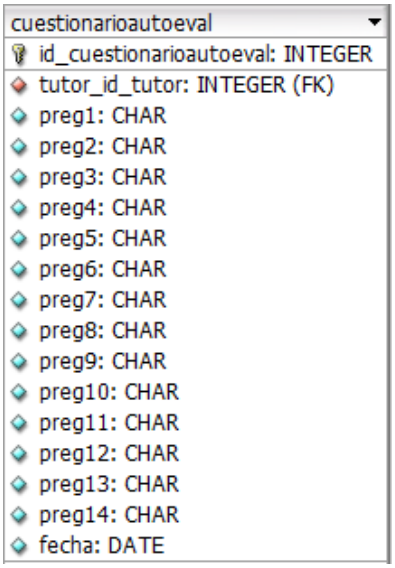
Esta entidad almacena la información de la encuesta llenada por los tutores que evalúan a su coordinador de la carrera.

- Cuestionariocopadi

Esta entidad almacena la información de la encuesta realizada por los tutores para evaluar al programa de tutoría Nueva Era.



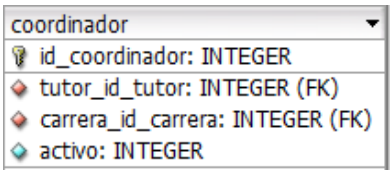
- Cuestionarioautoeval



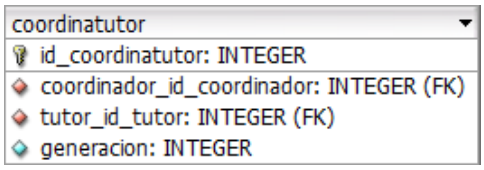

Esta entidad almacena la información de la encuesta llenada por los tutores para realizar una autoevaluación en el desempeño como tutor.

- Coordinador

Esta entidad almacena la relación de la información de los tutores que son coordinadores.



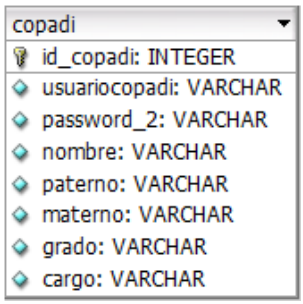

- Coordinatutor



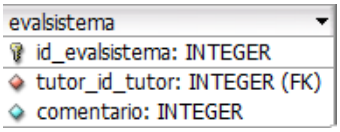

Esta entidad almacena la información de la asociación de tutores a los coordinadores.

- Copadi

Esta entidad almacena la información de los administradores del sistema.



- Evalsistema




Esta entidad almacena la información enviada por los tutores y coordinadores que realizan, comentarios, quejas y sugerencias al sistema TutorFI.

#### **4.3.1.1.3** Modelo Físico: Esquema de la base de Datos

```
CREATE TABLE copadi (
  id_copadi INTEGER UNSIGNED NOT NULL,
  usuariocopadi VARCHAR NOT NULL,
  password_2 VARCHAR NOT NULL,
  nombre VARCHAR NULL,
  paterno VARCHAR NULL,
  materno VARCHAR NULL,
  grado VARCHAR NULL,
  cargo VARCHAR NULL,
  PRIMARY KEY(id_copadi)
);

CREATE TABLE carrera (
  id_carrera SERIAL,
  nombre VARCHAR NOT NULL,
  PRIMARY KEY(id_carrera)
);

CREATE TABLE tutor (
  id_tutor SERIAL,
  nombredeusuario VARCHAR NULL,
  password_2 VARCHAR NULL,
  nombre VARCHAR NULL,
  paterno VARCHAR NULL,
  materno VARCHAR NULL,
  grado VARCHAR NULL,
  division VARCHAR NULL,
  correoe VARCHAR NULL,
  telefono VARCHAR NULL,
  ubicacion VARCHAR NULL,
  semblanza VARCHAR NULL,
  PRIMARY KEY(id_tutor)
);

CREATE TABLE beca (
  id_beca SERIAL,
  nombre VARCHAR NULL,
  descripcion VARCHAR NULL,
  PRIMARY KEY(id_beca)
);

CREATE TABLE cuestionariocopadi (
  id_cuestionariocopadi SERIAL,
  tutor_id_tutor INTEGER UNSIGNED NOT NULL,
  preg1 CHAR NULL,
  preg2 CHAR NULL,
  preg3 CHAR NULL,
  preg4 CHAR NULL,
  preg5 CHAR NULL,
  preg6 CHAR NULL,
  preg7 CHAR NULL,
  preg8 CHAR NULL,
  preg9 CHAR NULL,
  preg10 CHAR NULL,
  preg11 CHAR NULL,
  preg12 CHAR NULL,
  preg13 CHAR NULL,
  preg14 CHAR NULL,
  preg15 CHAR NULL,
  preg16 CHAR NULL,
  preg17 CHAR NULL,
  preg18 CHAR NULL,
  preg19 CHAR NULL,
  preg20 CHAR NULL,
  preg21 CHAR NULL,
  fecha DATE NULL,
  PRIMARY KEY(id_cuestionariocopadi),
  FOREIGN KEY(tutor_id_tutor)
    REFERENCES tutor(id_tutor)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION
);

CREATE TABLE cuestionariocoordinador (
  id_cuestionariocoordinador SERIAL,
  tutor_id_tutor INTEGER UNSIGNED NOT NULL,
  preg1 CHAR NULL,
  preg2 CHAR NULL,
  preg3 CHAR NULL,
  preg4 CHAR NULL,
  preg5 CHAR NULL,
  preg6 CHAR NULL,
  preg7 CHAR NULL,
  preg8 CHAR NULL,
  coordinador VARCHAR NULL,
  fecha DATE NULL,
  PRIMARY KEY(id_cuestionariocoordinador),
  FOREIGN KEY(tutor_id_tutor)
    REFERENCES tutor(id_tutor)
      ON DELETE NO ACTION
```

```
    ON UPDATE NO ACTION
);

CREATE TABLE evalsistema (
 id_evalsistema SERIAL,
 tutor_id_tutor INTEGER UNSIGNED NOT NULL,
 comentario INTEGER UNSIGNED NULL,
 PRIMARY KEY(id_evalsistema),
 FOREIGN KEY(tutor_id_tutor)
  REFERENCES tutor(id_tutor)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
);

CREATE TABLE cuestionarioautoeval (
 id_cuestionarioautoeval SERIAL,
 tutor_id_tutor INTEGER UNSIGNED NOT NULL,
 preg1 CHAR NULL,
 preg2 CHAR NULL,
 preg3 CHAR NULL,
 preg4 CHAR NULL,
 preg5 CHAR NULL,
 preg6 CHAR NULL,
 preg7 CHAR NULL,
 preg8 CHAR NULL,
 preg9 CHAR NULL,
 preg10 CHAR NULL,
 preg11 CHAR NULL,
 preg12 CHAR NULL,
 preg13 CHAR NULL,
 preg14 CHAR NULL,
 fecha DATE NULL,
 PRIMARY KEY(id_cuestionarioautoeval),
 FOREIGN KEY(tutor_id_tutor)
  REFERENCES tutor(id_tutor)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
);

CREATE TABLE bloque (
 id_bloque SERIAL,
 carrera_id_carrera INTEGER UNSIGNED NOT NULL,
 numero INTEGER UNSIGNED NULL,
 letra CHAR NULL,
 generacion INTEGER UNSIGNED NULL,
 PRIMARY KEY(id_bloque),
 FOREIGN KEY(carrera_id_carrera)
  REFERENCES carrera(id_carrera)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
);

CREATE TABLE bloquetutor (
 id_bloquetutor SERIAL,
 tutor_id_tutor INTEGER UNSIGNED NOT NULL,
 bloque_id_bloque INTEGER UNSIGNED NOT NULL,
 PRIMARY KEY(id_bloquetutor),
 FOREIGN KEY(bloque_id_bloque)
  REFERENCES bloque(id_bloque)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
 FOREIGN KEY(tutor_id_tutor)
  REFERENCES tutor(id_tutor)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
);

CREATE TABLE registrosesion (
 id_registrosesion SERIAL,
 bloque_id_bloque INTEGER UNSIGNED NOT NULL,
 tutor_id_tutor INTEGER UNSIGNED NOT NULL,
 tema VARCHAR NULL,
 actividad VARCHAR NULL,
 numasistentes INTEGER UNSIGNED NULL,
 fecha DATE NULL,
 generacion INTEGER UNSIGNED NULL,
 PRIMARY KEY(id_registrosesion),
 FOREIGN KEY(tutor_id_tutor)
  REFERENCES tutor(id_tutor)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
 FOREIGN KEY(bloque_id_bloque)
  REFERENCES bloque(id_bloque)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
);

CREATE TABLE coordinador (
 id_coordinador SERIAL,
 tutor_id_tutor INTEGER UNSIGNED NOT NULL,
 carrera_id_carrera INTEGER UNSIGNED NOT NULL,
 activo INTEGER UNSIGNED NULL,
 PRIMARY KEY(id_coordinador),
 FOREIGN KEY(carrera_id_carrera)
  REFERENCES carrera(id_carrera)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
 FOREIGN KEY(tutor_id_tutor)
  REFERENCES tutor(id_tutor)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
);

CREATE TABLE alumno (
 numcuenta SERIAL,
 bloque_id_bloque INTEGER UNSIGNED NOT NULL,
 carrera_id_carrera INTEGER UNSIGNED NOT NULL,
 beca_id_beca INTEGER UNSIGNED NOT NULL,
 nombre VARCHAR NULL,
 paterno VARCHAR NULL,
 materno VARCHAR NULL,
 telefono VARCHAR NULL,
 celular VARCHAR NULL,
 correo VARCHAR NULL,
 generacion INTEGER UNSIGNED NULL,
 para CHAR NULL,
 PRIMARY KEY(numcuenta),
 FOREIGN KEY(beca_id_beca)
  REFERENCES beca(id_beca)
    ON DELETE NO ACTION
```

```
    ON UPDATE NO ACTION,
  FOREIGN KEY(carrera_id_carrera)
   REFERENCES carrera(id_carrera)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  FOREIGN KEY(bloque_id_bloque)
   REFERENCES bloque(id_bloque)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
);

CREATE TABLE registrosesionindividual (
  id_registrosesionindividual SERIAL,
  alumno_numcuenta INTEGER UNSIGNED NOT NULL,
  bloque_id_bloque INTEGER UNSIGNED NOT NULL,
  tutor_id_tutor INTEGER UNSIGNED NOT NULL,
  fecha DATE NULL,
  tipo CHAR NULL,
  generacion INTEGER UNSIGNED NULL,
  numasistentes INTEGER UNSIGNED NULL,
  activo INTEGER UNSIGNED NULL,
  PRIMARY KEY(id_registrosesionindividual),
  FOREIGN KEY(tutor_id_tutor)
   REFERENCES tutor(id_tutor)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  FOREIGN KEY(bloque_id_bloque)
   REFERENCES bloque(id_bloque)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  FOREIGN KEY(alumno_numcuenta)
   REFERENCES alumno(numcuenta)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
);

CREATE TABLE historial (
  alumno_numcuenta INTEGER UNSIGNED NOT NULL,
  porcentaje INTEGER UNSIGNED NOT NULL,
  promedio FLOAT NULL,
  materias INTEGER UNSIGNED NULL,
  lab INTEGER UNSIGNED NULL,
  FOREIGN KEY(alumno_numcuenta)
   REFERENCES alumno(numcuenta)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
);

CREATE TABLE examendiagnostico (
  alumno_numcuenta INTEGER UNSIGNED NOT NULL,
  alg INTEGER UNSIGNED NULL,
  trig INTEGER UNSIGNED NULL,
  geomeuc INTEGER UNSIGNED NULL,
  geomanlt INTEGER UNSIGNED NULL,
  calc INTEGER UNSIGNED NULL,
  mec INTEGER UNSIGNED NULL,
  term INTEGER UNSIGNED NULL,
  elect INTEGER UNSIGNED NULL,
  quim INTEGER UNSIGNED NULL,
  carrera VARCHAR NULL,
  promgene FLOAT NULL,
  prommate FLOAT NULL,
  promfisica FLOAT NULL,
  FOREIGN KEY(alumno_numcuenta)
   REFERENCES alumno(numcuenta)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
);

CREATE TABLE califsemestre1 (
  alumno_numcuenta INTEGER UNSIGNED NOT NULL,
  algebra INTEGER UNSIGNED NOT NULL,
  calculo INTEGER UNSIGNED NULL,
  geometria INTEGER UNSIGNED NULL,
  dibujo INTEGER UNSIGNED NULL,
  quimica INTEGER UNSIGNED NULL,
  computacion INTEGER UNSIGNED NULL,
  topografia INTEGER UNSIGNED NULL,
  cultura INTEGER UNSIGNED NULL,
  industrial INTEGER UNSIGNED NULL,
  geologia INTEGER UNSIGNED NULL,
  FOREIGN KEY(alumno_numcuenta)
   REFERENCES alumno(numcuenta)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
);

CREATE TABLE coordinatutor (
  id_coordinatutor SERIAL,
  coordinador_id_coordinador INTEGER UNSIGNED NOT
NULL,
  tutor_id_tutor INTEGER UNSIGNED NOT NULL,
  generacion INTEGER UNSIGNED NULL,
  PRIMARY KEY(id_coordinatutor),
  FOREIGN KEY(tutor_id_tutor)
   REFERENCES tutor(id_tutor)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  FOREIGN KEY(coordinador_id_coordinador)
   REFERENCES coordinador(id_coordinador)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
);

CREATE TABLE evalumno (
  id_evalumno SERIAL,
  tutor_id_tutor INTEGER UNSIGNED NOT NULL,
  alumno_numcuenta INTEGER UNSIGNED NOT NULL,
  comentario VARCHAR NULL,
  PRIMARY KEY(id_evalumno),
  FOREIGN KEY(alumno_numcuenta)
   REFERENCES alumno(numcuenta)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  FOREIGN KEY(tutor_id_tutor)
   REFERENCES tutor(id_tutor)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
);
```

### 4.3.2 Diseño del Sistema TutorFI

Dado que el sistema está contenido dentro del portal, y su diseño debe ser homogéneo, su diseño será tomado de las plantillas de las hojas de estilo de la página principal de COPADI, así como los scripts que usa para generar efectos y aplicaciones como lo son el menú y el contador.

La estructura que sigue el sistema TutorFI se muestra en el diagrama de abajo:

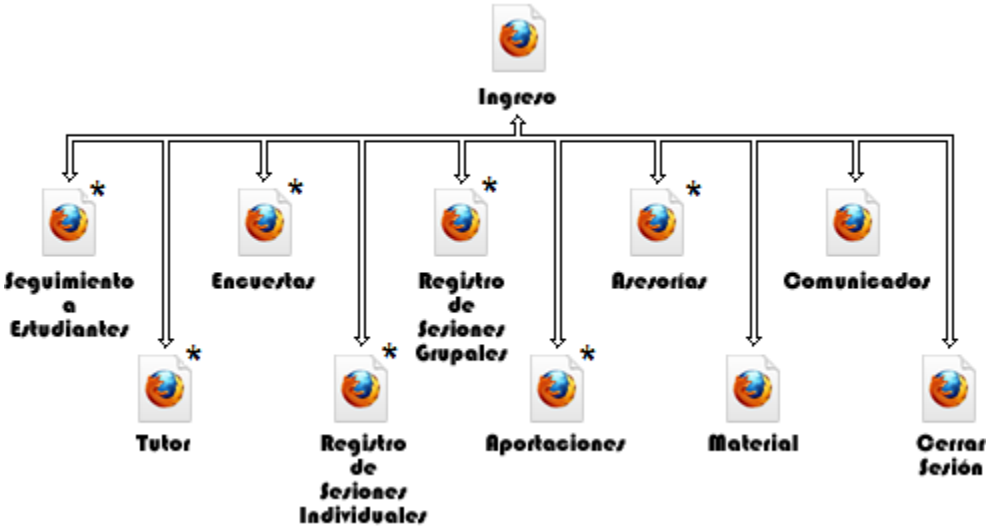



Figura 19. Esquema del TutorFI

Para el seguimiento a estudiantes tenemos:


Figura 20. Esquema de Seguimiento a Estudiante

Tutor:


Figura 21. Esquema Tutor

Encuestas:

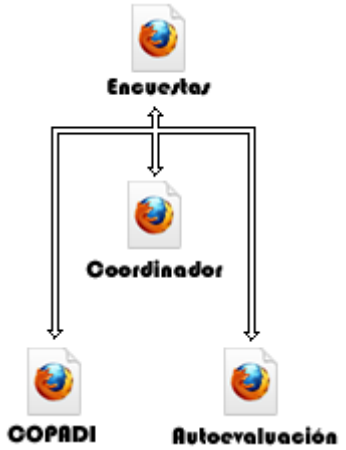



Figura 22. Esquema Encuestas

Registro de Sesiones Individuales:


Figura 23. Esquema Sesiones Individuales

Registro de Sesiones Grupales:


Figura 24. Esquema Sesiones Grupales

Aportaciones:


Figura 25. Esquema Aportaciones

Asesorías:

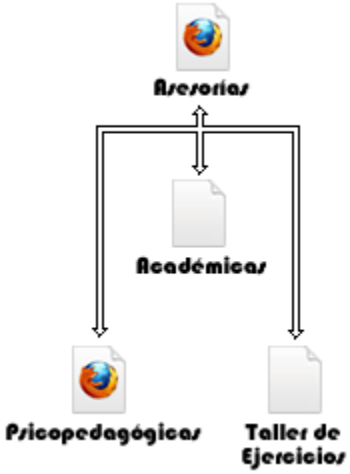



Figura 26. Esquema Asesorías

Para la página de ingreso al sistema queda de la siguiente forma:


Figura 27. Acceso TutorFI

En donde podemos ver que también se cuenta con su propio contador de visitas, la fecha, encabezado, pie de página, acceso al manual de uso del sistema y el menú de los distintos perfiles que maneja el sistema así como el apartado de los datos para ingresar.

Para el menú de los perfiles se utilizó las librerías de JQuery así como los archivos:

- *http://copadi.fi-c.unam.mx/tutorfi/js/efectoLogin.js*
- *http://copadi.fi-c.unam.mx/tutorfi/css/efectoLogin.css*

Este efecto muestra el icono de una forma ampliada del lado derecho para una mejor visualización. El código de la fecha y el contador son los mismos códigos de programación empleados en la página principal. Solo que el registro del contador es independiente al de la página.

Al ingresar al sistema como usuario "Tutor" muestra esta pantalla:


Figura 28. Ingreso Tutor

Como se puede apreciar, solo se cuenta con un menú y la parte del contenido a mostrar.

Al ingresar al sistema como usuario "Coordinador" muestra esta pantalla:

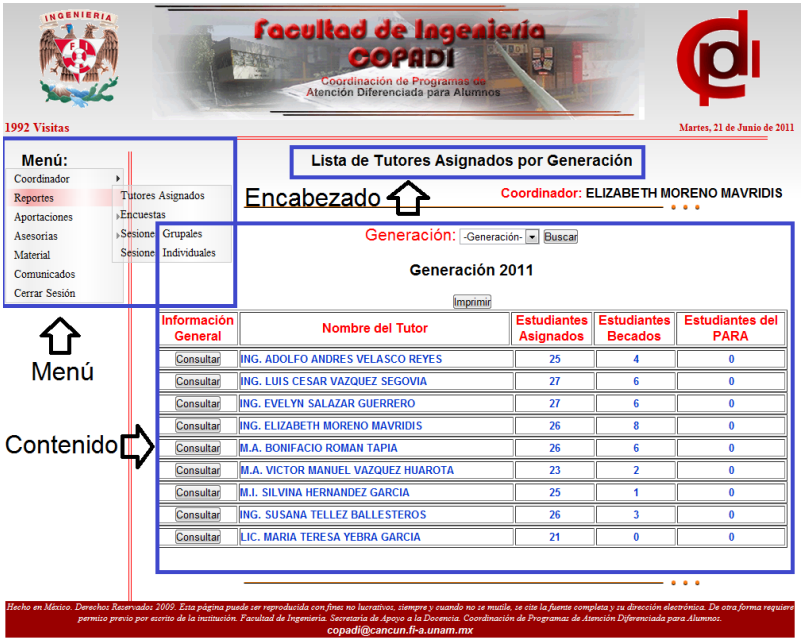


Figura 29. Ingreso Coordinador

Como usuario "Copadi" tenemos:


Figura 30. Ingreso COPADI

Finalmente, como usuario "Estudiante" tenemos:


Figura 31. Ingreso Estudiante

Dentro de los menús de cada uno de los perfiles tenemos:

Estudiante:

- Consulta Tutor
- Asesorías
    - Psicopedagógicas
    - Académicas
    - Taller de Ejercicios
- Comunicados
- Cerrar Sesión

Tutor:

- Seguimiento a Estudiantes
    - Consulte Listas
        - Generación 2011
        - Generación 2010
        - Generación 2009
        - Generación 2008
    - Descargar e Imprimir Archivos
- Tutor
    - Consultar Datos
    - Actualizar Datos
- Encuestas
    - COPADI
    - Coordinador
    - Autoevaluación
- Registro de Sesiones Individuales
    - Registrar Sesión Individual
    - Consultar Sesiones Individuales
- Registro de Sesiones Grupales
    - Registrar Sesión Grupal
    - Consultar Sesiones Grupales
- Aportaciones
    - Enviar Aportaciones

    - Consultar Aportaciones
- Asesorías
    - Psicopedagógicas
    - Académicas
    - Taller de Ejercicios
- Material
- Comunicados
- Cerrar Sesión

Coordinador:

- Coordinador
    - Consultar Datos
    - Actualizar Datos
- Reportes
    - Tutores Asignados
    - Encuestas
    - Sesiones Grupales
    - Sesiones Individuales
- Aportaciones
    - Enviar Aportaciones
    - Consultar Aportaciones
- Asesorías
    - Psicopedagógicas
    - Académicas
    - Taller de Ejercicios
- Material
- Comunicados
- Cerrar Sesión

COPADI:

- Estudiante
    - Consultar
- Reportes
- Cerrar Sesión

### 4.3.3 Estructura y Funcionamiento del TutorFI

Como anteriormente he mencionado, el sistema está programado con JSP, así mismo se han creado 17 beans que son usados en el sistema con determinada tarea, estos son:

- Aportaciones.class
- BeanParaCopadi.class
- BeanCuestionarioAutoevaluacion.class
- BeanParaEvaluarAlumno.class
- BeanCuestionarioCoordinador.class
- BeanParaEvaluarSistema.class
- BeanCuestionarioCopadi.class
- BeanParaExamenDiag.class
- BeanParaAlumno.class
- BeanParaRegistrarSesion.class
- BeanParaAlumnoReporteSemestral.class
- BeanParaTutor.class
- BeanParaAvance.class
- BeanRegistroSesionIndividual.class
- BeanParaCalif.class
- Contador.class
- BeanParaCoordinador.class

Cada uno de estos archivos realiza las acciones necesarias para capturar los datos enviados por los usuarios, así como alta, baja, modificación y consultas. Estos archivos son empleados en el sistema, solo donde son requeridos los datos solicitados por el usuario.

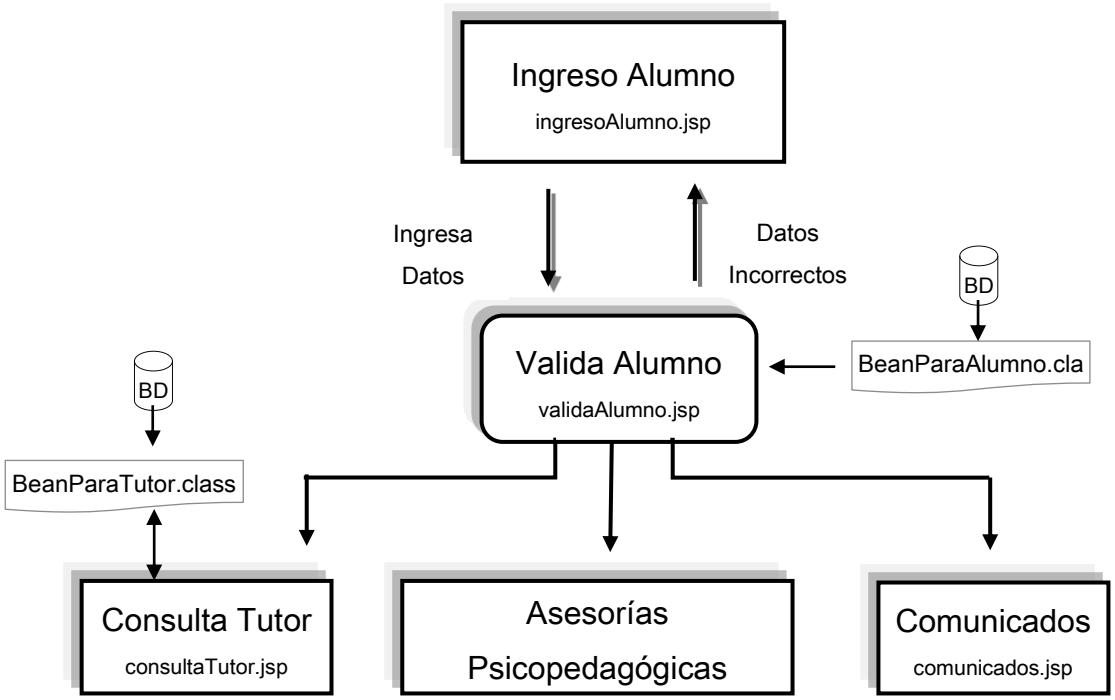
Un ejemplo de estos archivos es el siguiente:

```
package tutorfi;
import java.sql.*;
import java.util.*;
public class BeanParaTutor
{
  private String idTutor = "";
  private String nombreDeUsuario = "";
  private String password = "";
  private String passus = "";
  private String nombre = "";
      .
      .
      .
  public void setIdTutor(String newIdTutor)  {    idTutor = newIdTutor;  }
  public String getIdTutor()  {   return idTutor;  }
  public void setNombreDeUsuario(String newNombreDeUsuario)
  public void setPassword(String newPassword)  {     password = newPassword;  }
  public String getPassword()  {     return password;  }
      .
      .
      .
public String validaDatosDeTutor() throws ClassNotFoundException,SQLException {
      .
      .
      .
  String query = "select nombreDeUsuario from tutor where nombreDeUsuario=? and password=?";
  Class.forName("org.postgresql.Driver");
  con = DriverManager.getConnection("jdbc:postgresql://localhost/tutorfi" , "XXX", "XXX");
  pstm = con.prepareStatement(query);
  pstm.setString(1,nombreDeUsuario);
  pstm.setString(2,password);
  rs = pstm.executeQuery();
  if(rs.next())  {
   respuesta = "Datos correctos";
  }
  else  {
   respuesta = "Datos incorrectos";
  }
  con.close();
  return respuesta;
 }
}
```

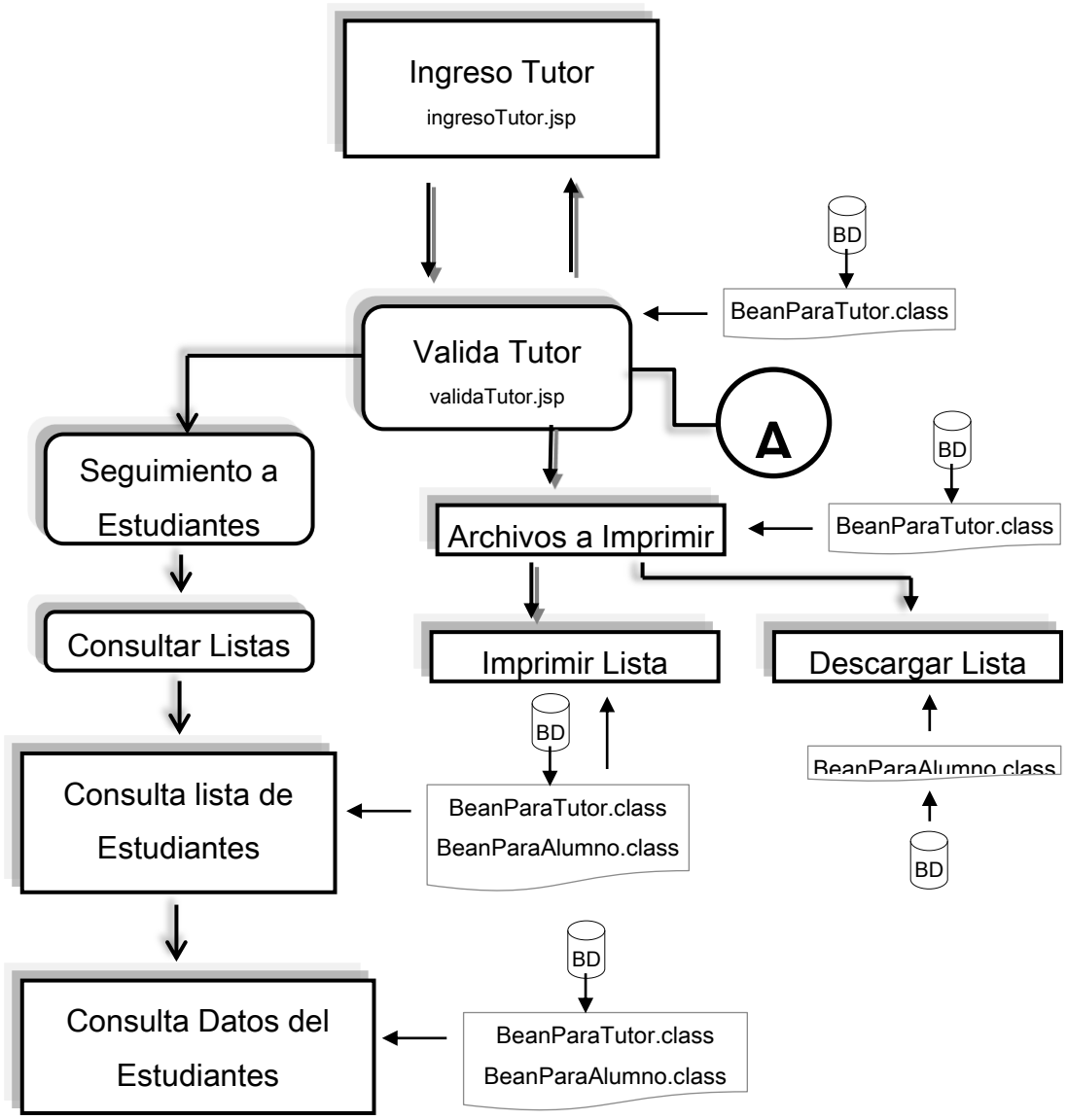
**4.3.3.1** Diagrama de Flujo de Datos

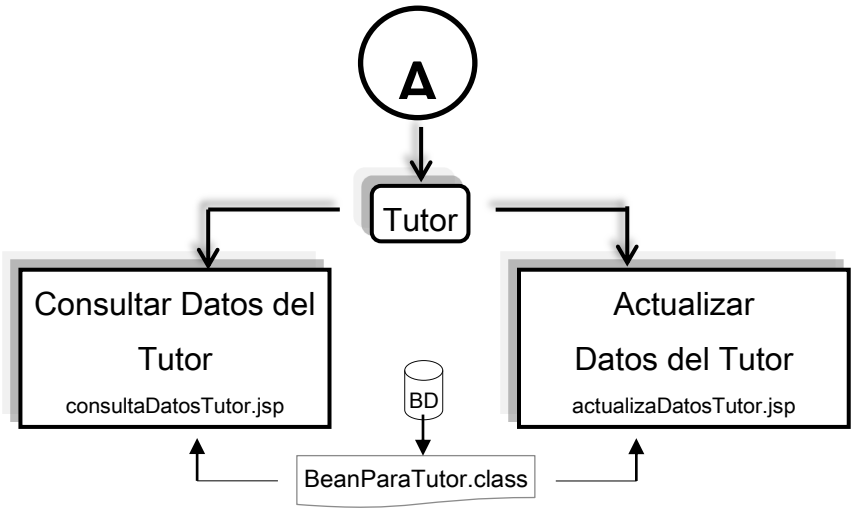
Para dar una explicación más amplia y detallada del funcionamiento y la estructura, así como el uso de cada uno de los archivos del sistema, a continuación presento el diagrama de flujo de datos de cada uno de los perfiles.

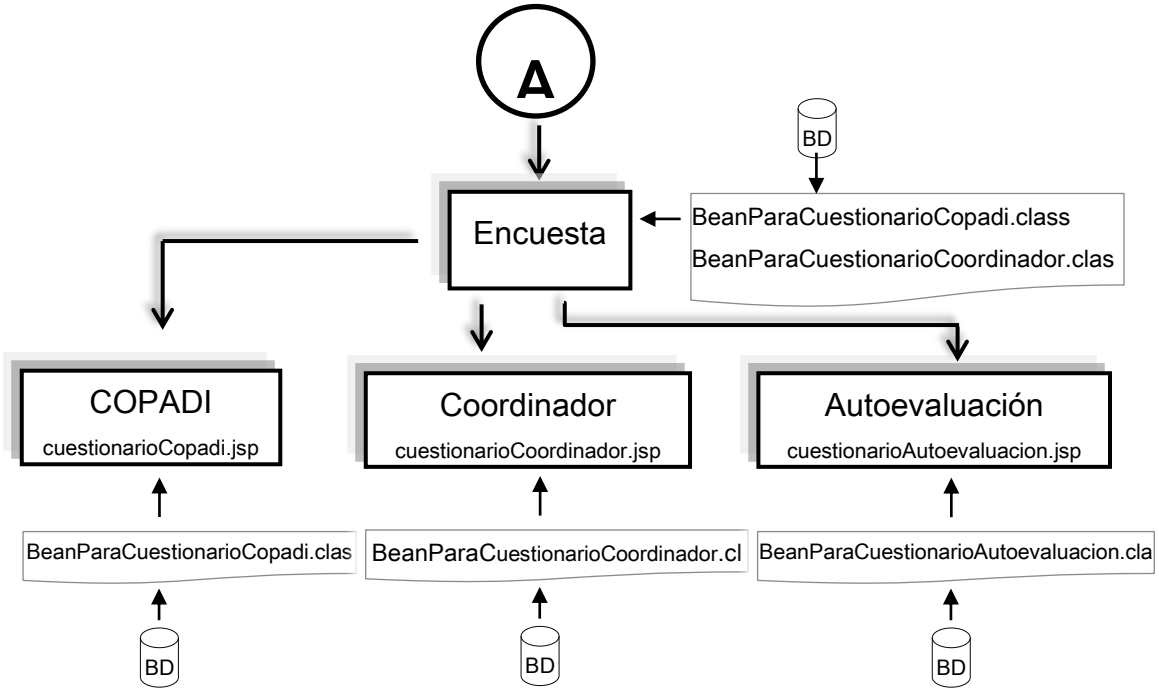
Estudiante:

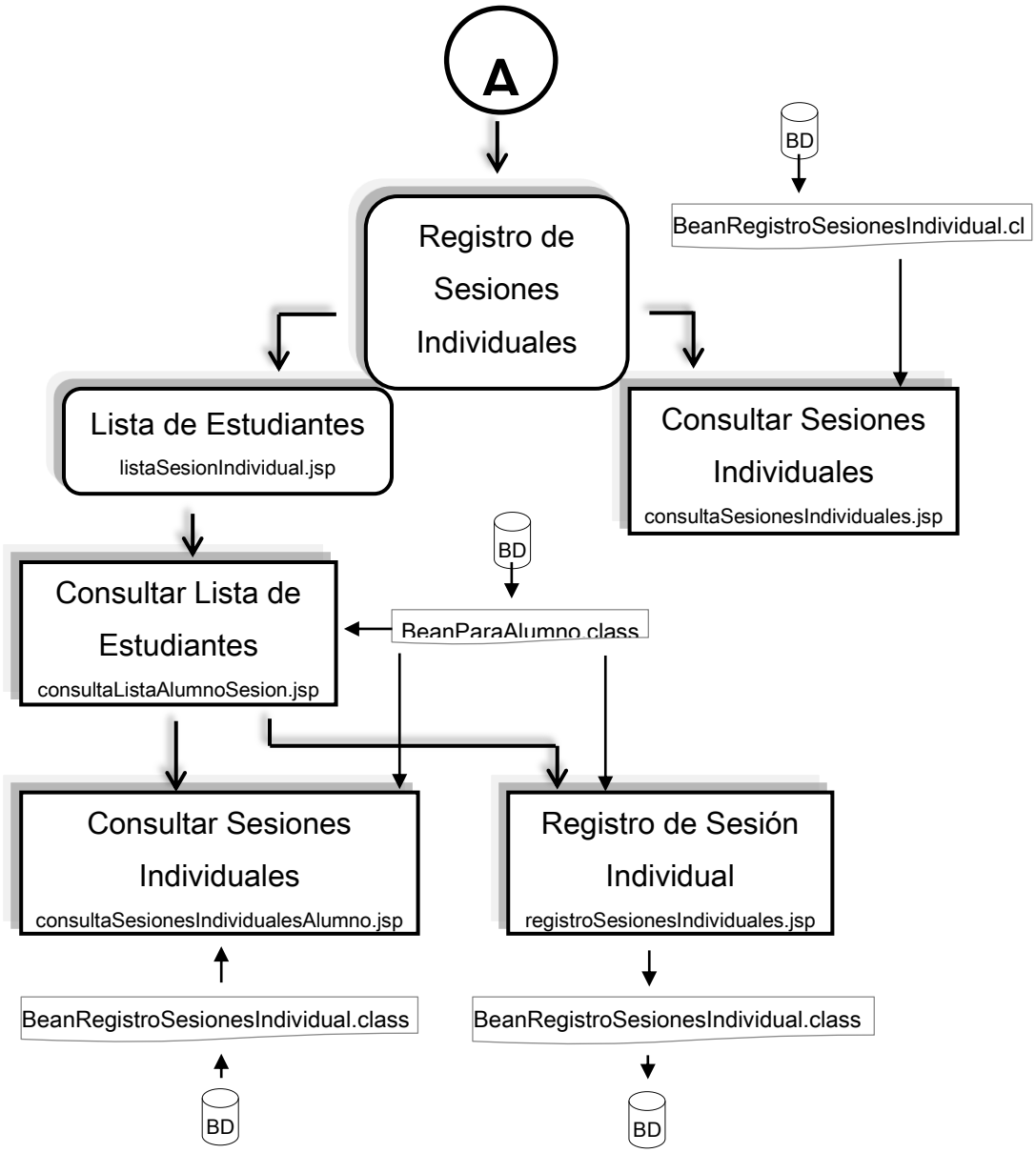


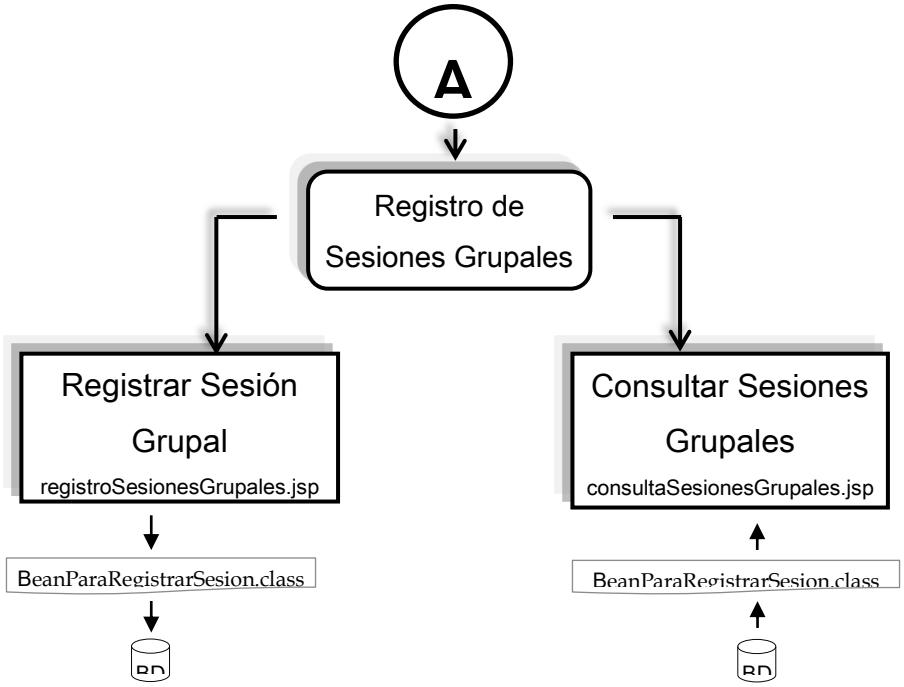
Tutor:

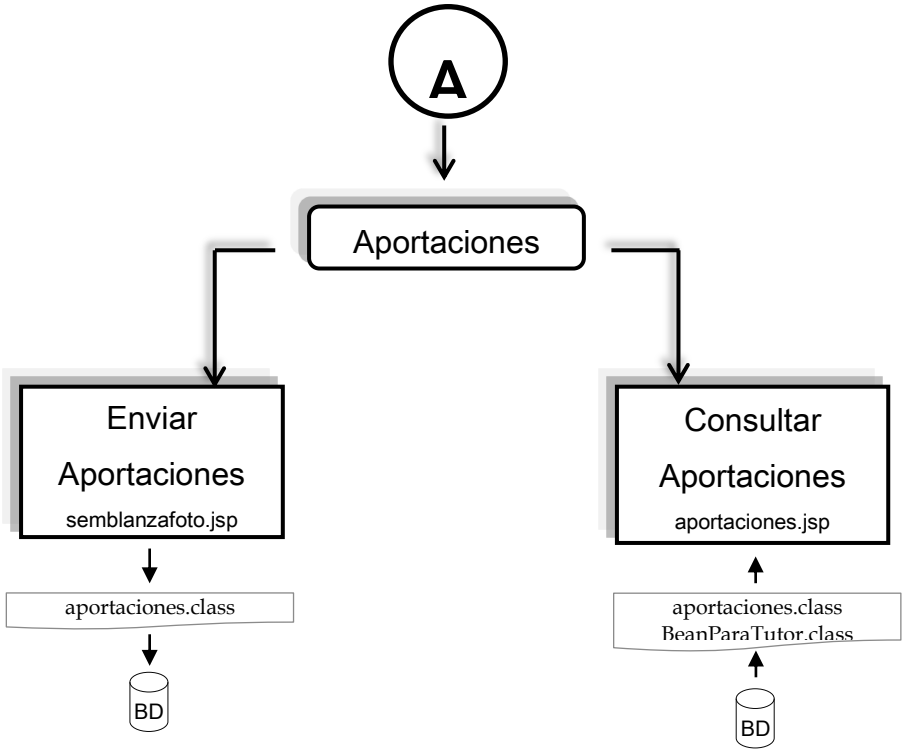

Ingreso Tutor
ingresoTutor.jsp
BD
BeanParaTutor.class
Valida Tutor
validaTutor.jsp
A
Seguimiento a Estudiantes
BD
Archivos a Imprimir
BeanParaTutor.class
Consultar Listas
Imprimir Lista
Descargar Lista
BD
BeanParaAlumno.class
Consulta lista de Estudiantes
BeanParaTutor.class
BeanParaAlumno.class
BD
BD
Consulta Datos del Estudiantes
BeanParaTutor.class
BeanParaAlumno.class

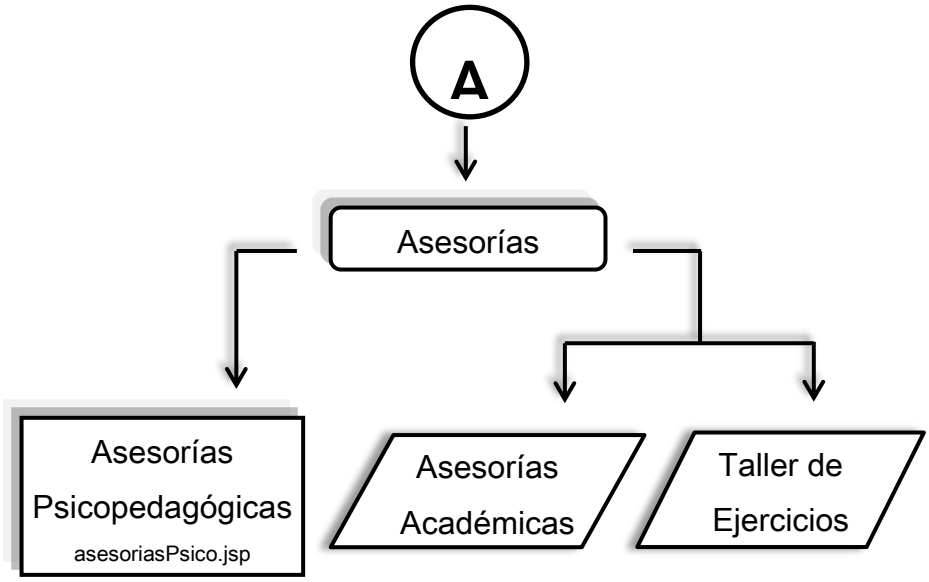

A
Tutor
Consultar Datos del Tutor
consultaDatosTutor.jsp
BD
Actualizar Datos del Tutor
actualizaDatosTutor.jsp
BeanParaTutor.class

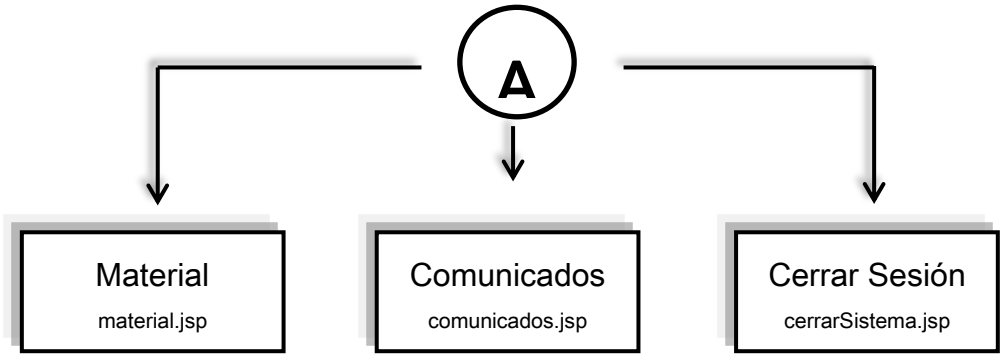

A
BD
Encuesta
BeanParaCuestionarioCopadi.class
BeanParaCuestionarioCoordinador.clas
COPADI
cuestionarioCopadi.jsp
Coordinador
cuestionarioCoordinador.jsp
Autoevaluación
cuestionarioAutoevaluacion.jsp
BeanParaCuestionarioCopadi.clas
BeanParaCuestionarioCoordinador.cl
BeanParaCuestionarioAutoevaluacion.cla
BD
BD
BD


A
BD
BeanRegistroSesionesIndividual.cl
Registro de Sesiones Individuales
Lista de Estudiantes
listaSesionIndividual.jsp
Consultar Sesiones Individuales
consultaSesionesIndividuales.jsp
BD
Consultar Lista de Estudiantes
consultaListaAlumnoSesion.jsp
BeanParaAlumno.class
Consultar Sesiones Individuales
consultaSesionesIndividualesAlumno.jsp
Registro de Sesión Individual
registroSesionesIndividuales.jsp
BeanRegistroSesionesIndividual.class
BeanRegistroSesionesIndividual.class
BD
BD

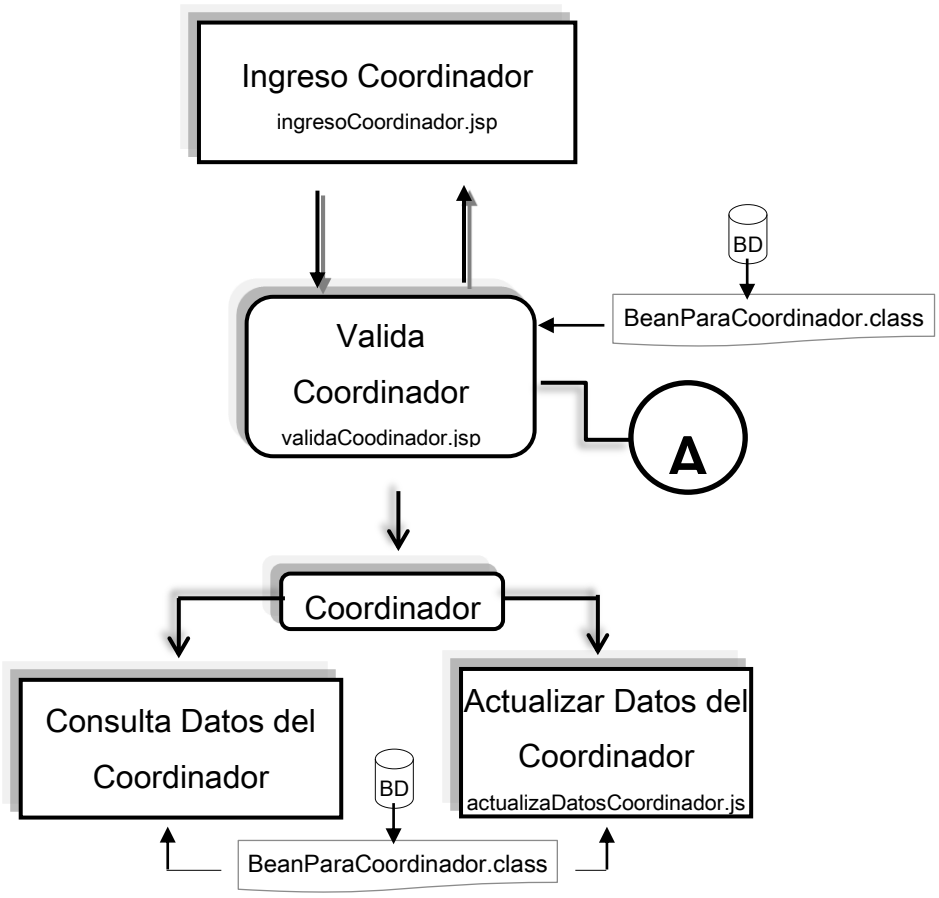

A
Registro de Sesiones Grupales
Registrar Sesión Grupal
registroSesionesGrupales.jsp
Consultar Sesiones Grupales
consultaSesionesGrupales.jsp
BeanParaRegistrarSesion.class
BeanParaRegistrarSesion.class
BD
BD

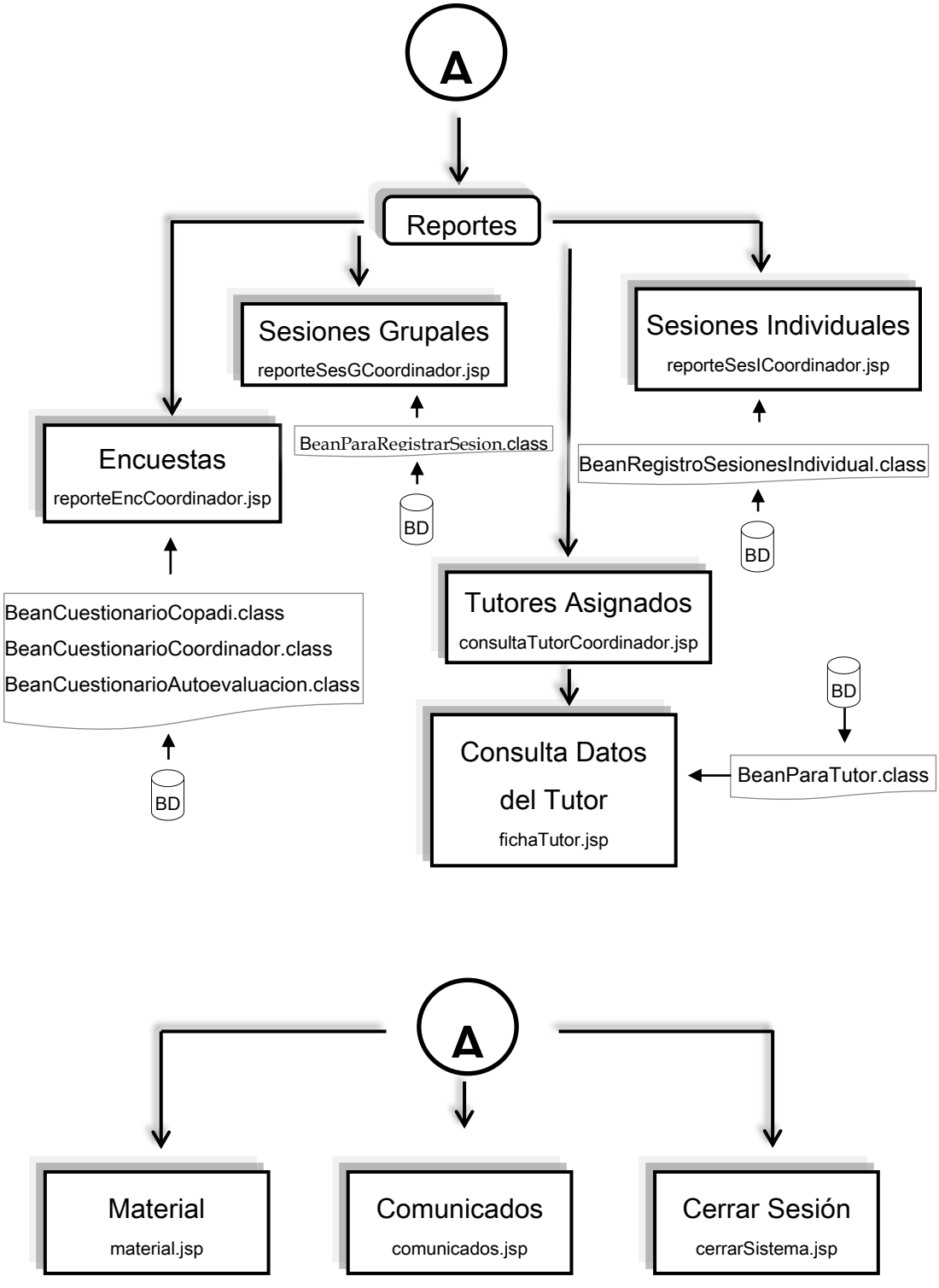

A
Aportaciones
Enviar Aportaciones
semblanzafoto.jsp
Consultar Aportaciones
aportaciones.jsp
aportaciones.class
aportaciones.class
BeanParaTutor.class
BD
BD

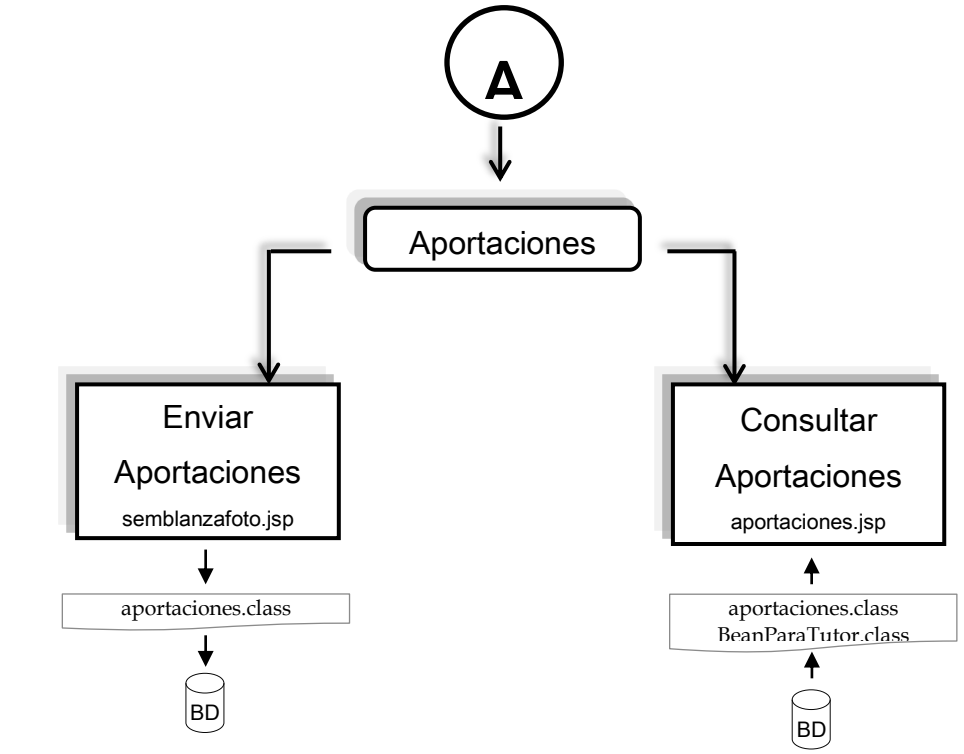

A
Asesorías
Asesorías Psicopedagógicas
asesoriasPsico.jsp
Asesorías Académicas
Taller de Ejercicios

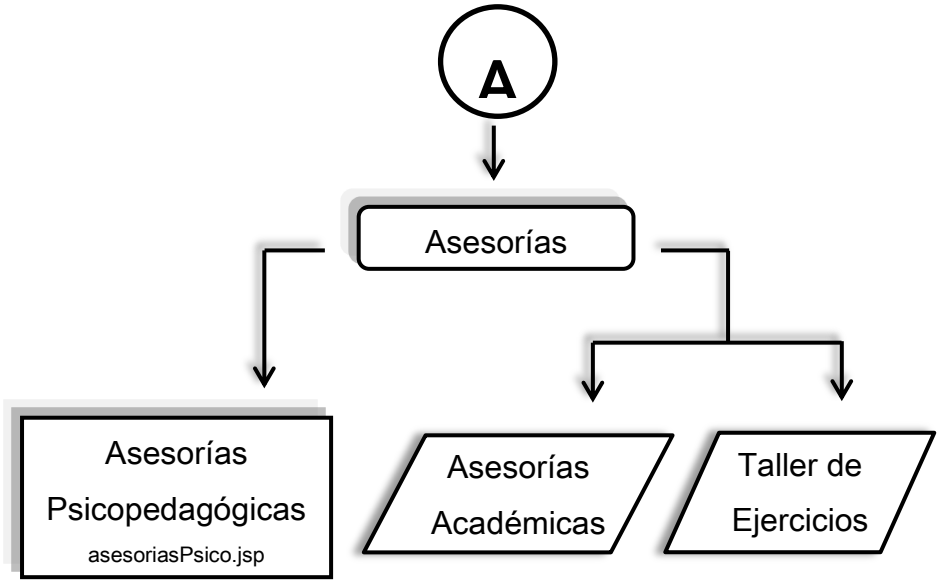

A
Material
material.jsp
Comunicados
comunicados.jsp
Cerrar Sesión
cerrarSistema.jsp

Coordinador:

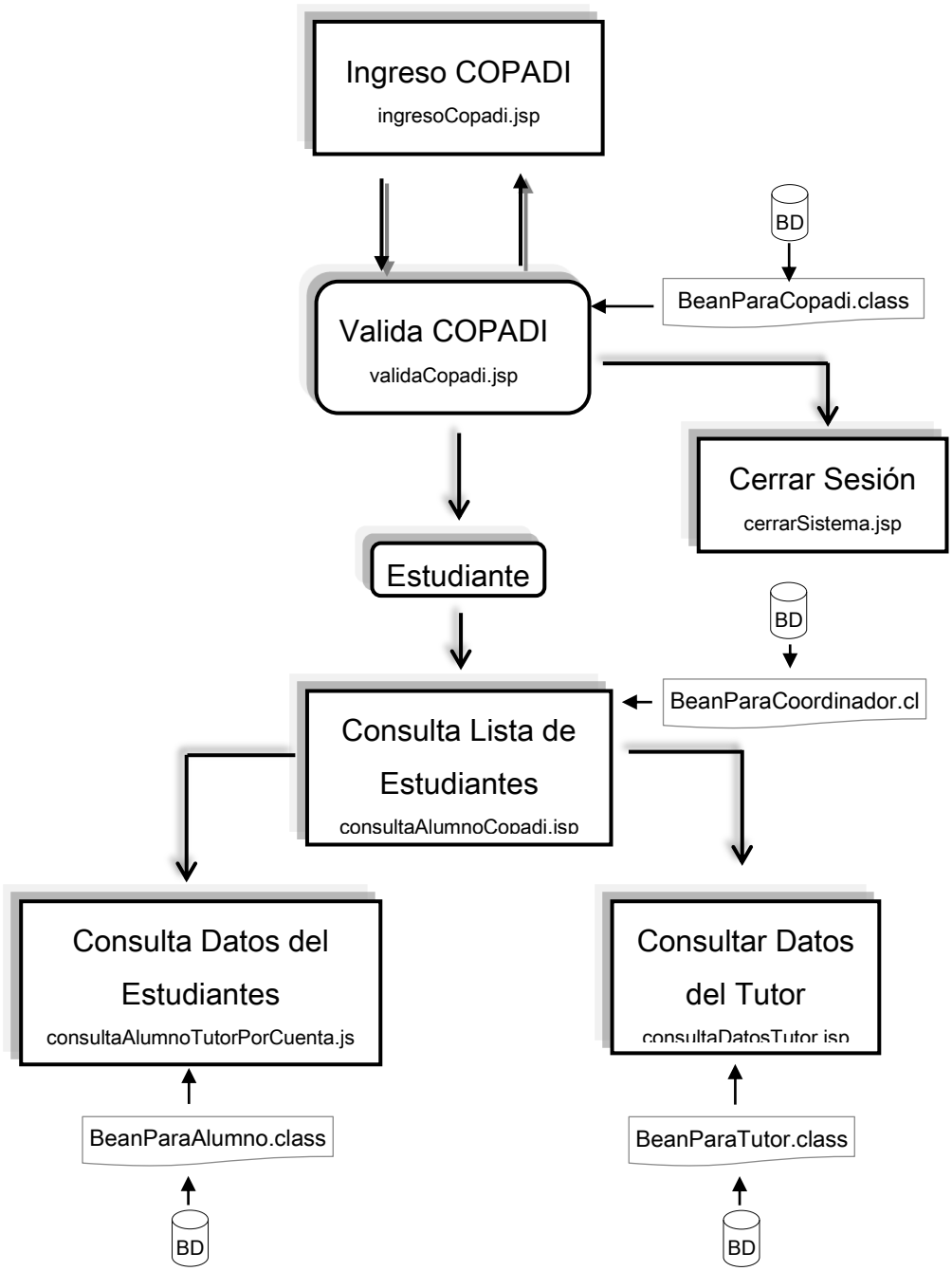



A
Reportes
Sesiones Grupales
reporteSesGCoordinador.jsp
Sesiones Individuales
reporteSesICoordinador.jsp
Encuestas
reporteEncCoordinador.jsp
BeanParaRegistrarSesion.class
BeanRegistroSesionesIndividual.class
BD
BD
BeanCuestionarioCopadi.class
BeanCuestionarioCoordinador.class
BeanCuestionarioAutoevaluacion.class
Tutores Asignados
consultaTutorCoordinador.jsp
BD
Consulta Datos del Tutor
fichaTutor.jsp
BeanParaTutor.class
BD
A
Material
material.jsp
Comunicados
comunicados.jsp
Cerrar Sesión
cerrarSistema.jsp


A
Aportaciones
Enviar
Aportaciones
semblanzafoto.jsp
aportaciones.class
BD
Consultar
Aportaciones
aportaciones.jsp
aportaciones.class
BeanParaTutor.class
BD


A
Asesorías
Asesorías
Psicopedagógicas
asesoriasPsico.jsp
Asesorías
Académicas
Taller de
Ejercicios

COPADI:


Ingreso COPADI
ingresoCopadi.jsp
BD
BeanParaCopadi.class
Valida COPADI
validaCopadi.jsp
Cerrar Sesión
cerrarSistema.jsp
Estudiante
BD
BeanParaCoordinador.cl
Consulta Lista de Estudiantes
consultaAlumnoCopadi.jsp
Consulta Datos del Estudiantes
consultaAlumnoTutorPorCuenta.js
Consultar Datos del Tutor
consultaDatosTutor.jsp
BeanParaAlumno.class
BeanParaTutor.class
BD
BD

## **4.4** Implementación del Sistema

Para la implementación del Sistema TutorFI, previamente se adquirió en la COPADI un servidor para alojar el portal de COPADI, así como almacenar las bases de datos con las que cuenta la coordinación, así mismo alojar los futuros módulos que se integraran al Portal.

Las especificaciones técnicas del Servidor adquirido son:

Servidor DELL PowerEdge T300

Modelo: PowerEdge T300
Memoria: RAM 4 Gb.
Procesador: Intel® Celeron® Dual-Core Intel® Core™2 Duo®
Almacenamiento: 2 discos duros SATA Hot-Plug 500 Gb c/u.
Quemador DVD 16X
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, se le asignó al servidor la

dirección 132.248.139.201 con su respectivo nombre de dominio (copadi.fi.-c.unam.mx) Ver apéndice B.

Con esto el sistema se pone en marcha para realizar las primeras pruebas y se obtienen los primeros resultados.

### 4.5 Pruebas y Resultados

Para el desarrollo del Portal, lo primero que se puso en marcha fue la Página principal de COPADI, en donde solo se muestra información al usuario.

Las primeras pruebas que se hicieron fueron principalmente a que funcionara a la perfección el contador de visitas, así también se verificó el correcto funcionamiento de los efectos implementados.

También se hicieron las correcciones pertinentes a lo que es la información presentada, dentro de esta se revisó la ortografía y se actualizo la información que se tenía para los diferentes programas que ofrece la COPADI.

De acuerdo a la metodología empleada para el desarrollo del portal se van implementando las modificaciones que piden y sugieren los usuarios. Por tanto, se hacen unas pequeñas modificaciones en la manera de presentar la información, así como el diseño en cuanto a la estructura de la información.

En cuanto al Sistema TutorFI, se elige una pequeña población dentro de la comunidad de tutores para realizar las primeras pruebas del sistema. Se hace la entrega de los nombres de usuario y sus respectivas contraseñas para el ingreso al mismo.

Una vez en el sistema, se pide a los usuarios consultar la información que ofrece el sistema, así como realizar las primeras capturas de información para corroborar el funcionamiento del mismo.

Una vez detectados lo primeros errores (corrección en los acentos ortográficos, así como mal funcionamiento en la compatibilidad de navegadores) y obtenidos los primeros comentarios y sugerencias, se pone en marcha su corrección y la nueva revisión al sistema.

Después de la primer prueba, se da a conocer a toda la comunidad de tutores, previamente se imprimió un formato en donde se les otorgaba su respectivo nombre de usuario, sus contraseñas y la dirección electrónica donde se puede acceder al sistema.

Dado que se detectó una muy baja recepción de información en el servidor, se toma la decisión de hacer la visita a los tutores donde se le daba una pequeña explicaciones de los servicios que ofrece el sistema, así como dar de una manera sintetizada la información disponible, así como la información que debe de capturar el tutor para enviarla a la coordinación por medio del sistema.

Con esta nueva implementación para dar a conocer el sistema, se detectan nuevos errores que son generados por la compatibilidad de los navegadores, así como la carga de trabajo en el servidor, por lo que se realizan las nuevas modificaciones en cuanto a generar los documentos para descargar, así como el correcto envío de la información con acentos. En el transcurso de la implementación y de las pruebas, tanto tutores como coordinadores nos retroalimentan con comentarios y sugerencias para el TutorFI.

El correcto funcionamiento del sistema nos ofrece los resultados que se esperaban al inicio del desarrollo del proyecto. El tener una amplia disponibilidad en los datos, así como reducir en gran medida la papelería que se utilizaba para realizar encuestas, entrega de listas de los estudiantes, material para tutores, etc. Nos permite tener una visión amplia del gran apoyo que es el Portal, ya que dentro de él se brinda la información necesaria para los usuarios, como la información que requieren los tutores para impartir su tutoría, obtener información de los tutores para con los estudiantes y los coordinadores pueden llevar un control de sus tutores dentro de su división.

# CAPÍTULO V

# "CONCLUSIONES"

He de agradecer a la Universidad Nacional Autónoma de México la oportunidad de aplicar mis conocimientos del área de la carrera de Ingeniería en Computación de la Facultad de Ingeniería que me gustan, como son las bases de datos, la programación, redes y la seguridad.

Además del gran reto de todo el desarrollo, se realizó en el sistema operativo LINUX, así mismo, la utilización del lenguaje de programación JAVA, por lo que tengo una gran satisfacción de aprender varias cosas.

Resultado de ello, el buen funcionamiento del sistema cubriendo al 100% en todas las herramientas que ofrece, así como cumplir con su misión que es informar, presentar información así como recabarla para su posterior análisis, y también que tanto tutores como coordinadores puedan consultar de una forma rápida y sencilla la información de los estudiantes de donde se puede llevar un control.

De entre los resultados destacan el poder descargar archivos, es decir, las diferentes listas de los estudiantes, así como que la comunidad de tutores puedan compartir archivos entre ellos, como apoyo para el programa de la tutoría.

Con ello se cumplieron las expectativas que se tenían en un inicio.

Adicionalmente la implementación de estos sistemas permite ahorrar tiempo y un ahorro considerable en cuanto a papel, lo que ayuda enormemente al cuidado del medio ambiente.

## 5.1 Contribuciones y mejoras

Como para todo sistema desarrollado, cada usuario tiene diferente opinión y sugerencias en cuanto a sus gustos y necesidades principalmente, a lo que nos lleva tener un enorme campo de contribuciones y mejoras que a algunos les parezcan adecuadas y a otros usuarios no. Sin embargo debemos tener en cuenta los objetivos iniciales para el sistema, que planteaban para cubrir

satisfactoriamente necesidades como brindar información de forma rápida y sencilla de los estudiantes, recabar información de los tutores para retroalimentar a la coordinación por medio de las encuestas, así mismo, obtener la información de las sesiones que se tienen dentro del programa de la tutoría.

Actualmente se ha presentado el gran uso de las librerías que ofrece JQuery, que nos presenta un sinfín de efectos para un mejor diseño y presentación de la información, con el objetivo de hacer más atractivo los sistemas.

Este tipo de desarrollos permite ver de diferente forma la información y por tanto, más atractiva al usuario, así mismo representa un mejoramiento en el acceso y envío de información.

Para el sistema se le pueden implementar muchas mejoras que algunos tutores han sugerido, como son el implementar una herramienta parecida a las que implementan las redes sociales, que es un chat en línea dentro del TutorFI, que es lo que más resalta dentro de las contribuciones.

## 5.2 Limitaciones

Para incorporar nuevas aplicaciones al Portal o directamente al TutorFI, no se presenta ningún inconveniente ni limitaciones que puedan surgir para tal desarrollo, ya que el tipo de programación modular permite la reutilización de código y una mejor integración de diferentes módulos.

Tal vez una limitación que se puede presentar es que se pueda pedir más información para cada uno de los perfiles, es decir, tener más información sobre los estudiantes. Y digo que es una limitación ya que se han tenido problemas al momento de solicitar esa información a las personas correspondientes, ya que se maneja de distinta forma la información. Pero una vez que se tengan estos datos, no hay ningún problema en la incorporación a nuestra base de datos.

## 5.3 Líneas Futuras

Ahora bien, se debe de tener especial cuidado en implementar efectos y herramientas, ya que algunas de ellas llegan a tener pequeños problemas en Internet Explorer principalmente, ya que este navegador no sigue al 100 % los

estándares regidos por los demás navegadores. Cabe mencionar que es recomendable utilizar Mozilla Firefox para tener una mejor visualización del Portal.

También hay que tener cuidado en cómo se adquieren los datos, ya que por uno u otro inconveniente, llegan a presentarse erróneos dado que se usan las hojas de Excel para su importación al manejador de la base de datos. También es posible presentar más datos de cada uno de los perfiles, de acuerdo a las necesidades.

# Apéndice A.

## Instalación de Tomcat5.5

Bajar la versión binaria de Tomcat en: http://jakarta.apache.org/tomcat . (La versión de *Código Fuente* (src) solo es necesaria si quiere experimentar y/o Instalar Apache con Tomcat )

1. Descomprimir el archivo *Tar* de Tomcat en /directorio/, esto genera un directorio llamado jakarta-tomcat-<numero_de_version>, para dar mayor uniformidad se recomienda cambiar el nombre de este directorio a tomcat.
2. Posteriormente se debe definir una variable ambiental la cual le indicará al sistema la ubicación de Tomcat , esta variable se llama **CATALINA_HOME** la cual debe ser agregada a /etc/bashrc , si no está familiarizado con ambientes *nix, esto significa agregar la línea:

   ```
   export CATALINA_HOME=/usr/local/tomcat;
   ```

   Para instalaciones Windows esta variable ambiental puede ser definida de la misma manera que CLASSPATH, descrita en las instrucciones del JDK.

### Configuración Local

Solo para ilustrar esta instalación inicial modifiqué el archivo /etc/hosts agregando una línea como la siguiente:

```
132.248.139.201 copadi.fi-c.unam.mx
```

Generalmente el paso anterior se logra a través de DNS, pero por facilidad y demostración se optó por modificar /etc/hosts.

### Ejecución y Pruebas

Una vez efectuados los pasos anteriores es posible realizar pruebas iniciales sobre Tomcat, para esto ejecute lo siguiente:

- Descienda al directorio bin de Tomcat (/directorio/tomcat/bin ) y ejecute el archivo startup.sh :

```
[root@servidor1 bin]# ./catalina.sh run
Using CATALINA_BASE:   /directorio/tomcat
Using CATALINA_HOME:   /directorio/tomcat
Using CATALINA_TMPDIR: /directorio/tomcat/temp
Using JAVA_HOME:   /directorio/jdk
```

**En caso de tener error probar:**

[root@curso09 apache-tomcat-5.5.23]# **vi /root/.bash_profile**

**Insertar las siguientes líneas:**

```
export JAVA_HOME=/directorio/jdk
export CATALINA_HOME=/directorio/tomcat
PATH=$JAVA_HOME/bin:$CATALINA_HOME/bin:$PATH:$HOME/bin
export PATH
unset USERNAME
```

(Esto es debido a que no se definen correctamente las variables del JDK)

Ejecutar el archivo .bash_profile para cargar el nuevo ambiente.

[root@curso09 apache-tomcat-5.5.23]# **./root/.bash_profile**
[root@curso09 apache-tomcat-5.5.23]# **java -version**

**Aquí muestra las versiones de java instaladas**

```
[root@curso09 usr]# cd /directorio/tomcat/bin
[root@curso09 bin]# ./startup.sh
```

**Vemos que levante los servicios**

Verificar en la bitácora /usr/apache-tomcat-5.5.23/logs/catalina.out que el servidor se haya levantado exitósamente.

```
[root@curso09 bin]# tail -f ../logs/catalina.out
```

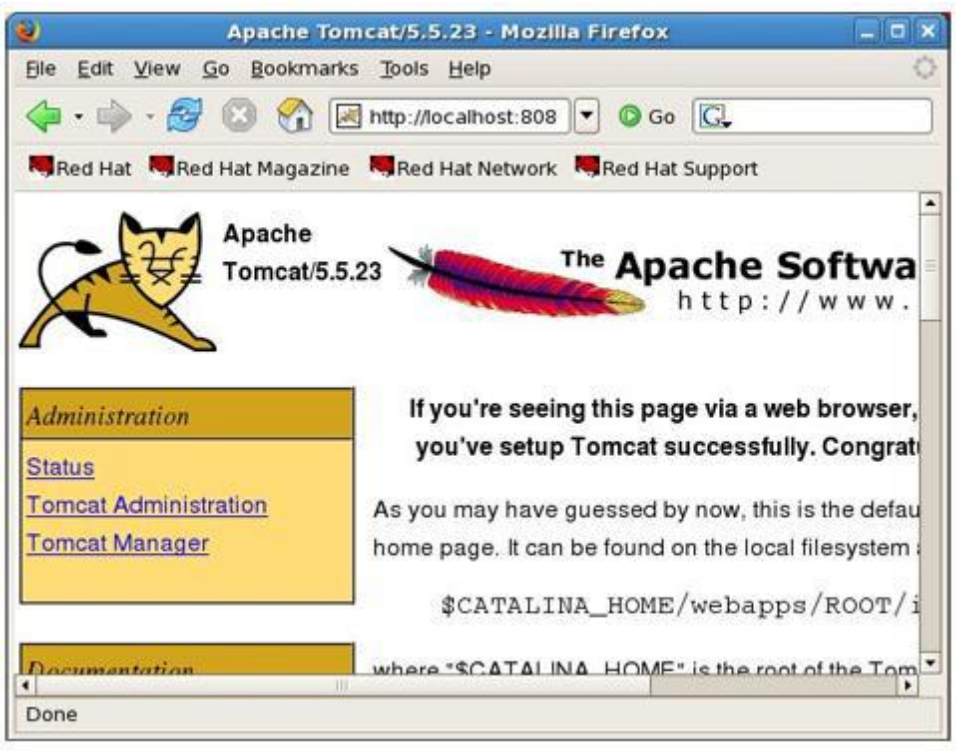
En caso de aparecer un LifecycleException: This server has not yet been started

**Termina los servicios:**

```
[root@curso09 bin]# ./shutdown.sh
```

Pruebas:

**Probar que la página http://localhost:8080 se ejecute satisfactoriamente.**




Configuración de Tomcat5.5

Para comenzar con las configuraciones necesarias, es necesario ubicarnos en el directorio:

```
cd /directorio/tomcat/conf
```

Aquí encontraremos los siguientes archivos:

Web.xml

Al final de este archivo agregamos las líneas correspondientes para que tomcat

abra por default el tipo de páginas web. En nuestro caso:

```
<welcome-file>principal.html</welcome-file>
```

tomcat-users.xml

Este archivo define a los usuarios permitidos para administrar tomcat. Editamos el archivo y agregamos:

```
<role rolename="manager"/>

<role rolename="admin"/>

<user username="root" password="XXXX" roles="manager,admin"/>
```

**server.xml**

*server.xml* es el archivo principal de configuración para Tomcat, al igual que otros archivos de configuración para productos empleados en servidor puede contener una gran variedad de parámetros, sin embargo, esta guia se concentrará en los parámetros principales.

**Validación, <!-- --> y valores "Default"**

El archivo *server.xml* es un archivo en XML , el cual de no contener una estructura conforme a XML, se indicará al arranque de Tomcat; dicho archivo se encuentra bajo el directorio /directorio/tomcat/conf donde /directorio/tomcat es el directorio definido en CATALINA_HOME.

Como cualquier otro documento en XML todo contenido entre <!-- --> es considerado un comentario, y por lo tanto cualquier parámetro que se encuentre entre estos caracteres no es utilizado por "Tomcat"; aquellos parámetros que no sean definidos dentro de *server.xml* son asignados un valor "Default" por Tomcat.

**<Server> - </Server>**

<Server> es el elemento principal del archivo *server.xml* y todas las demás secciones deben encontrarse entre estos nodos; el atributo port indica el puerto TCP donde se espera la señal de cierre (shutdown) de Tomcat, el cual rara vez es modificado.

<Listener>

A través de los elementos <Listener> se configuran las extensiones JMX ("Java Management Extensions") que serán utilizadas por Tomcat, dichos elementos toman dos atributos: className que indica la Clase diseñada para escuchar sobre eventos JMX y debug para especificar el nivel de "debug" generado al tiempo de ejecución.

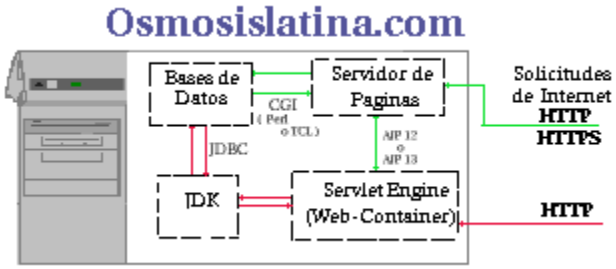
**<GlobalNamingResources> , <Resource> y <ResourceParams>**

Anidado dentro de los elementos <GlobalNamingResources> es posible definir recursos JNDI para ser utilizados globalmente en Tomcat. Lo anterior evita que estos recursos tengan que ser declarados a nivel de WAR de manera individual.

A través de <Resource> es como define el tipo de recurso JNDI que será utilizado y mediante <ResourceParams> se especifican los parámetros específicos que tomará el recurso en dicha instancia de Tomcat.

<Service> - </Service>

Esta parámetro permite configurar Tomcat para diferentes modalidades de ejecución, en el archivo *server.xml* "Default" se definen dos modalidades a través del atributo name; la definición asignada name="Catalina" es empleada para ejecutar Tomcat individualmente, esto es representado con la línea roja de la siguiente gráfica:




**<Connector>**

El elemento Connector representa las conexiones (Puertos TCP) que serán abiertas por Tomcat al arranque, a su vez dentro de cada elemento Connector se definen diversos atributos los cuales dan más detalles acerca de la conexión.

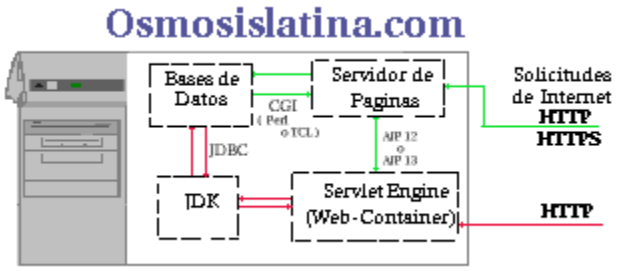
El elemento Connector más importante es aquel que define la clase: HttpConnector.

```
<Connector port="8080"
    maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
    enableLookups="false" redirectPort="8443" acceptCount="100"
    debug="0" connectionTimeout="20000"
    disableUploadTimeout="true" />
```

La declaración anterior indica que Tomcat está dispuesto a dar respuesta a requisiciones que arriben en el puerto 8080 del "Host" donde está instalado Tomcat; si recuerda la ejecución y pruebas de Tomcat a esto se debió el agregar el fragmento :8080.

Para ambientes de producción en los cuales toda requisición será atendida por Tomcat el parámetro port de este elemento debe ser modificado al valor de 80, este puerto es el ampliamente conocido puerto TCP 80 con el que intenta comunicarse cualquier Navegador("Netscape","Explorer","Opera" u otro) en Internet.

Gráficamente esto es representado con la línea roja de la siguiente gráfica:




Esta linea roja indica que Tomcat actúa en efecto como "Servidor de Páginas".

Otras declaraciones para Connectors son aquellas para utilizar encriptación (HTTPS) y los conectores AJP; la configuración de HTTPS es un tanto extensa y no se describirán los detalles, sin embargo, los conectores AJP son aquellos utilizados para la comunicación entre Apache y Tomcat y son los siguientes:

```
<Connector port="8009"
        enableLookups="false" redirectPort="8443" debug="0"
        protocol="AJP/1.3" />
```

Esta declaración indica que el puerto 8009 estará recibiendo conexiones bajo ajp13, lo anterior es *solo de interés si utiliza Apache en conjunción con Tomcat*, lo cual equivale a la linea verde del diagrama anterior.

**<Engine> - </Engine>**

Los elementos <Engine> los cuales deben encontrarse anidados dentro de <Service> representan el mecanismo que atenderá toda solicitud arribada Tomcat, esto es, toda solicitud recibida por las definiciones Connectors será procesada por <Engine>, los atributos de este elemento son los siguientes:

```
<Engine name="Catalina" defaultHost="localhost" debug="0">
```

defaulthost representa el nombre del servidor Tomcat mientras debug indica el nivel de "debug" generado al tiempo de ejecución.

**Logger**

Los elementos Logger le indican a Tomcat hacia donde deben ser enviados los registros "Logs":

```
<Logger className="org.apache.catalina.logger.FileLogger"
        directory="logs"  prefix="localhost_log." suffix=".txt"
       timestamp="true"/>
```

Lo anterior indica que los registros de Tomcat deben ser enviados al archivo localhost_log.txt; la ubicación de este registro ("log") puede ser modificada al nivel de los elementos Host los cuales permiten definir *Virtual Hosts*.

Analice estos registros con Analog .

<Host> - </Host>

Los elementos Host permiten definir varios Hosts "Virtuales" para Tomcat, esto es, a través del elemento <Engine> se define un *sitio* (localhost) para atender solicitudes, a través de Host es posible definir diversos sitios "Virtuales", su configuración es la siguiente:

```
<Host name="desarrollo.servidorprueba.com" debug="0" appBase="webapps"
unpackWARs="true">
```

Lo anterior indica que el sitio desarrollo.servidorprueba.com contiene sus

aplicaciones (Archivos WARS) bajo el directorio $CATALINA_HOME/webapps, donde $CATALINA_HOME es /directorio/tomcat; el atributo unpackWARs le indica a Tomcat que debe descomprimir los archivos WAR's al arranque.

Como ya fue mencionado, dentro de estos elementos <Host> es posible utilizar <Logger> para generar registros ("logs") por cada sitio virtual.

**Context**

Context es un elemento utilizado para indicar la ubicación de las aplicaciones ejecutadas en Tomcat , en su configuración "Default" estas aplicaciones se encuentran dentro del directorio webapps bajo el directorio raíz de Tomcat ( /usr/local/tomcat ). El entrar en los detalles de Context sería un tanto prematuro ya que aún no se ha descrito que es una aplicación en Tomcat.

Una aplicación en Tomcat o cualquier Servlet Engine(Web-Container) es un conjunto de "JSP's (*Java Server Pages*)" y/o "Servlets" agrupados con ciertos parámetros de arranque y seguridad, este conjunto de archivos / aplicación en *todo Servlet Engine* es conocido como un *WAR (Web-Archive)*.

# Apéndice B.

## Configuración de Red.

Una vez instalado el S.O. Es necesario configurar la tarjeta de Red.

Debido a que esta versión de Fedora no tiene el archivo que inicializa el servicio de red, es necesario crearlo.

Para iniciar abrimos una terminal e ingresamos como root:

**su -**

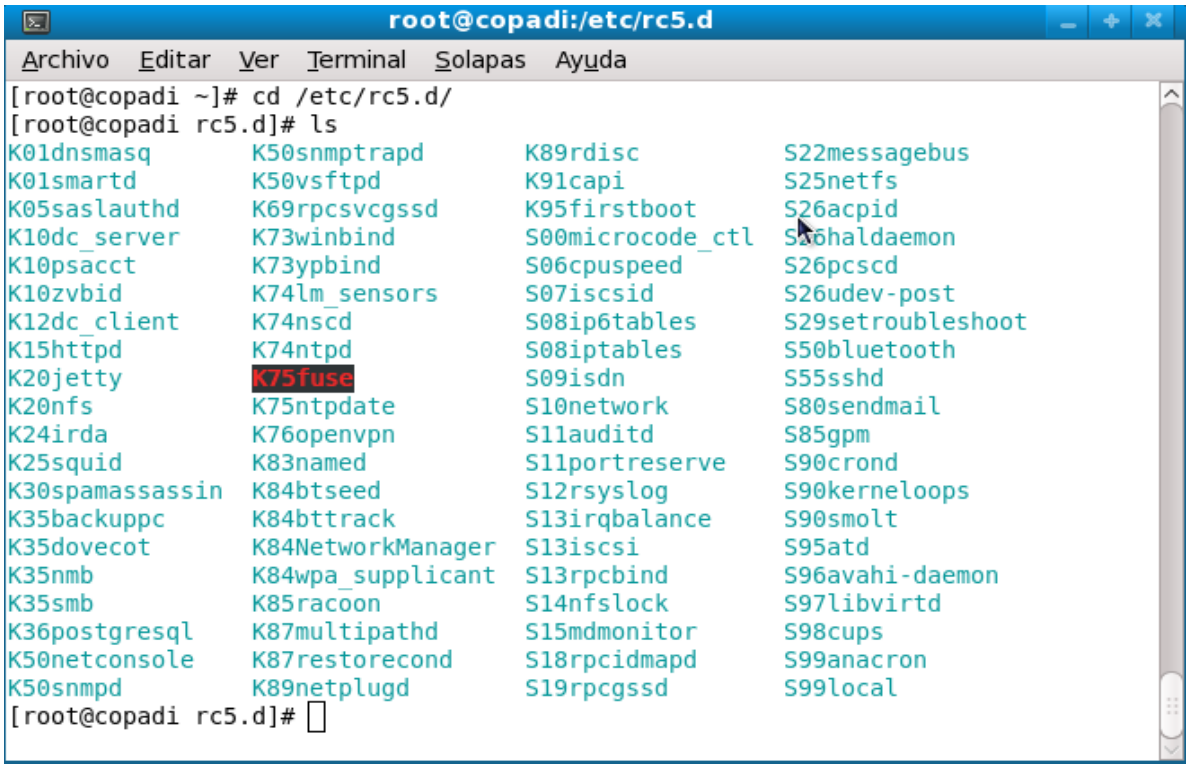
**password:*******

Enseguida nos ubicamos en el directorio:

**/etc/rc5.d/**

Observamos los archivos incluidos en este directorio:

**ls**



Dentro de él podemos ver que no se encuentra el archivo S90network (S = start), pero si el K90network (K = kill), por tanto es necesario crearlo con:

mv K90network S90network

A continuación configuramos los parámetros de red que usaremos. Para ello atamos definitivamente desde la consola el servicio de Red:

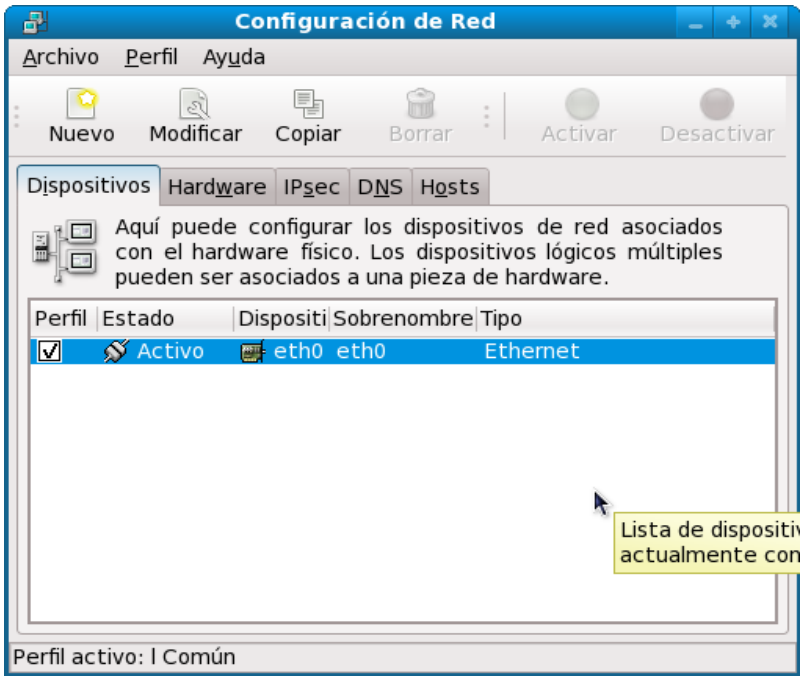
**chkconfig NetworkManager off;**
**service NetworkManager stop**

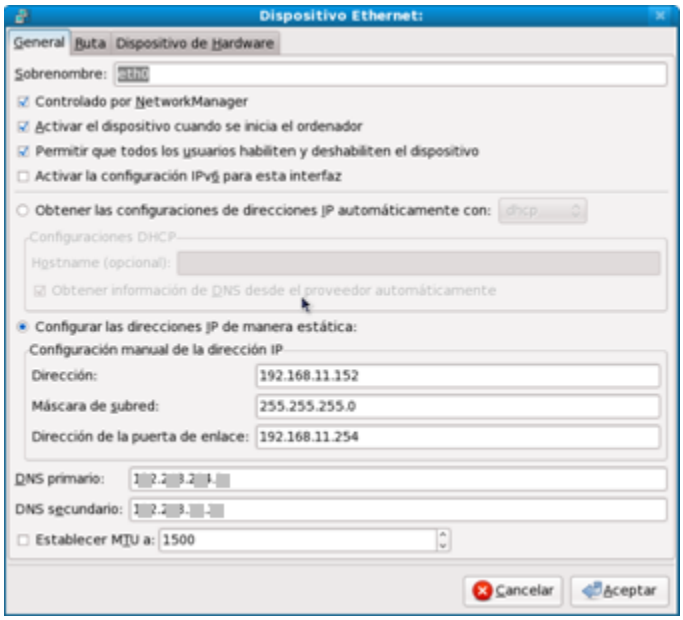
Los parámetros a configurar son:

IP: xxx.xxx.xxx.xxx
Mascara de Red:255,255,255,0
Puerta de Enlace: xxx.xxx.xxx.xxx
DNS1: xxx.xxx.xxx.xxx
DNS2: xxx.xxx.xxx.xxx

Esto lo podemos hacer de manera gráfica:

**sistema – Administración – red.**



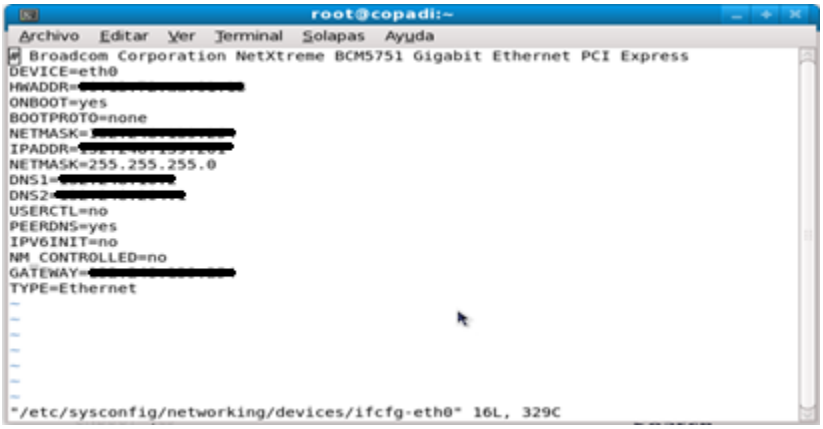



Damos doble click en la tarjeta de red e ingresamos los valores.

La otra opción para configurar los parámetros de Red es editando el archivo de configuración de la interfaz, para eth0 sería:

**gedit /etc/sysconfig/networking/devices/ifcfg-eth0**

y anexamos la información de Red, para que finalmente quede de esta forma:




Para reiniciar el servicio de Red, tecleamos en consola:

**chkconfig NetworkManager on;**
**service NetworkManager start**

# Apéndice C.

## Instalación jdk-6u16-linux-i586.bin

Después de descargar el archivo binario del JDK (página principal de Sun Microsystems), lo copiamos en el directorio donde realizaremos la instalación, se recomienda **/directorio/**, realizamos los siguientes pasos:

- Cambiar los permisos del archivo con el siguiente comando : chmod a+x jdk-6u16-linux-i586.bin
- Ejecutar del directorio local : ./jdk-6u16-linux-i586.bin
- Aceptar la Licencia de Usuario.

Una vez terminada la instalación queda instalado en una ruta absoluta: **/directorio /jdk**

## Configuración jdk-6u16-linux-i586.bin

El JDK requiere configurarse con diversas variables ambientales para su correcta operación:

- JAVA_HOME : Indica el directorio raíz de instalación del JDK, de acuerdo a las instrucciones anteriores esta ruta sería : /directorio/jdk
- PATH : Define la ruta de acceso para los binarios del sistema; la modificación de esta variable permite accesar los ejecutables Java (javac,javadoc,java) proporcionadas con el JDK de cualquier directorio.
- CLASSPATH : Define las rutas de acceso para las diversas librerías empleadas en ambientes Java; su modificación será descrita a lo largo del curso

Las variables anteriores pueden ser definidas de dos maneras:

- **Nivel Global**: Permite que las variables sean accesibles para todo usuario del sistema; estas definiciones son colocadas en el archivo /etc/profile del sistema.
- **Nivel Usuario**: Las variables son definidas para tener efecto únicamente sobre determinado usuario; estas definiciones son colocadas en el archivo ~/.bashrc, donde ~/ es el directorio base del usuario.

Independientemente de los métodos mencionados anteriormente, las declaraciones en estos archivos son idénticos:

```
JAVA_HOME="/directorio/jdk"
CLASSPATH="/directorio/mislibrerias"
```

```
PATH="$PATH:/directorio/jdk/bin"
export JAVA_HOME
export CLASSPATH
export PATH
```

**Pruebas**

Para verificar la correcta instalación del JDK realice la siguiente prueba:

- Colóquese en un directorio arbitrario del sistema.
- Ejecutar el archivo .bash_profile para cargar el nuevo ambiente.

  [root@curso09 apache-tomcat-5.5.23]# **./root/.bash_profile**
  [root@curso09 apache-tomcat-5.5.23]# **java -version**
- Invoque el comando java -version.

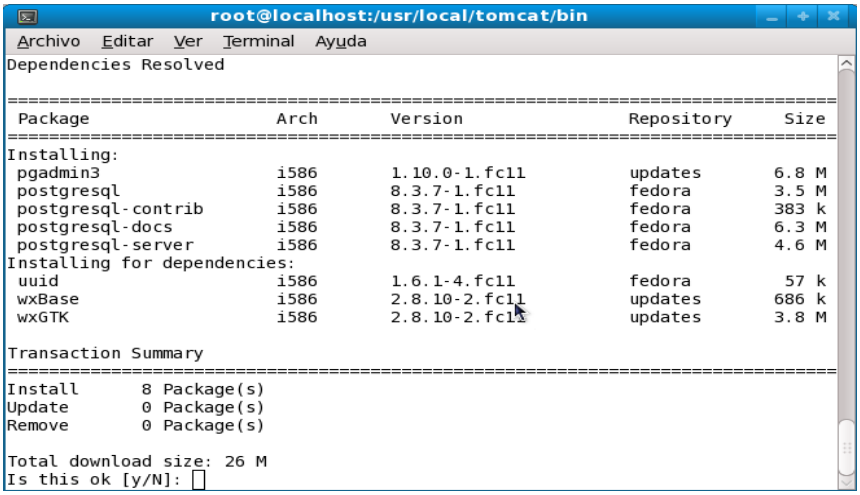
Si observa una respuesta indicando la versión del JDK lo ha instalado correctamente, en caso contrario realice los pasos anteriores hasta que esta prueba sea ejecutada correctamente. (hay ocasiones en que es necesario reiniciar para que se reconozcan las variables).

# Apéndice D.

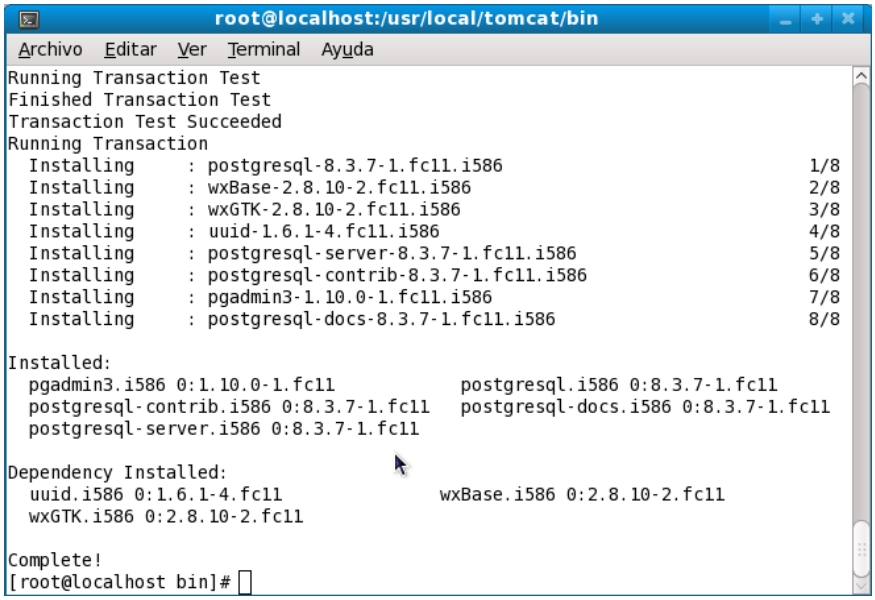
## Instalación de PostgreSQL

Instalaremos el Manejador de la Base de Datos con el siguiente comando:

*yum install postgresql postgresql-libs postgresql-contrib postgresql-server postgresql-docs pgadmin3*




Nos informa de los paquetes a instalar, solo tecleamos "**y**" y ENTER.



Al finalizar obtendremos:

Una vez instalado, se procede a inicializar el clúster de la siguiente manera:

*service postgresql initdb*

```
[root@copadi rc5.d]# service postgresql initdb
Iniciando la base de datos:                    [  OK  ]
```

Luego inicializamos el servicio de postgres:

*service postgresql start*

```
[root@copadi rc5.d]# service postgresql start
Iniciando servicios postgresql:                [  OK  ]
```

La instalación creó un usuario llamado *postgres*, en este momento es el único usuario autorizado para trabajar en el clúster. El usuario *postgres* no tiene asociado password alguna, entonces le crearemos una, para lo cual, nos conectamos de la siguiente manera:

```
su - postgres
[root@copadi rc5.d]# su – postgres
- bash-4.0$
```

Con lo cual, hemos accedido al clúster y la consola queda de la siguiente manera:

*-bash-4.0$*

ahora escribimos lo siguiente:

*psql -d template1 -U postgres*

```
-bash-3.2$ psql -d template1 -U postgres

Bienvenido a psql 8.3.7, la terminal interactiva de PostgreSQL.

Digite:  \copyright para ver los términos de distribución
     \h para ayuda de órdenes SQL
     \? para ayuda de órdenes psql
     \g o punto y coma («;») para ejecutar la consulta
     \q para salir
template1=#
```

Con dicha instrucción nos conectamos a *template1*, la base de datos por defecto (de *template1* se basarán todas las bases de datos que creemos en el futuro).

Ahora ingresamos el password para el usuario *postgres* de la siguiente manera:

```
alter user postgres with password 'XXX';
template1=# alter user postgres with password 'XXX';
ALTER ROLE
template1=#
```

Donde XXX es el password que se va a definir para el usuario *postgres.* Para desconectarnos de *template1*, basta con escribir *\q*

```
template1=# \q

-bash-3.2$
```

Por último, nos queda configurar los accesos a postgres. Para ello, debemos de modificar los archivos **postgresql.conf** y **pg_hba.conf** ubicados en **/var/lib/pgsql/data/**

```
[root@copadi ~]# cd /directorio/data/
[root@copadi data]# vi postgresql.conf
[root@copadi data]# vi pg_hba.conf
```

En el archivo *postgresql.conf* debemos de quitar las almohadillas (#) a las siguientes lineas:

```
listen_addresses = '*'
port = 5432
password_encryption = on
```

En el archivo *pg_hba.conf* buscamos la sección:

```
# "local" is for Unix domain socket connections only
```

Donde originalmente dice:

```
local all all ident sameuser
```

lo cambiamos por:

```
local all all trust
```

con dicha modificación se indica que todas las conexiones locales serán aceptadas. Si se desea impedir la conexión a usuarios que no están explícitamente autorizados para acceder a *template1* se debe hacer lo siguiente:

```
# "local" is for Unix domain socket connections only
local template1 all ident sameuser
local all all trust
```

Con lo cual se deniega el acceso a usuarios no autorizados para conectarse a *template1*.

Finalmente nos queda ubicar la sección:

```
# All IPv4 connections from localhost
```

Y agregar lo siguiente:

```
host all all 192.168.0.0/24 md5
```

En Caso de No funcionar, probar:

```
 "local" is for Unix domain socket connections only
local all all md5
# IPv4 local connections:
host all all 127.0.0.1/32 md5
# IPv6 local connections:
host all all ::1/128 md5
```

# BIBLIOGRAFÍA

The Software Architect Bootcamp.
Raphael Malveau y Thomas Mowbray.
Prentice-Hall, Octubre 2000.

Applied Software Architecture.
Christine Hofmeister, Robert Nord y Dilip Soni.
The Object Technology Series. Addison-Wesley Longman, 2000.

Programacion orientada a objetos
Luis Joyanes Aguilar
Osborne McGraw-Hill, 2da edición. 1998

The Rational Unified Proccess: An introduction
Phili ppe Kruchten
Addison Wesley, 2000

http://jquery.com/
http://docs.jquery.com/Downloading_jQuery
http://recursosweb.unam.mx/recursos-web/lineamientos-unam/
http://www.baluart.net/ejemplos/art31/menu-horizontal-desplegable.html
http://www.w3.org/Style/CSS/
http://www.adobe.com/support/documentation/en/dreamweaver/
http://tomcat.apache.org/tomcat-5.5-doc/index.html
http://netbeans.org/kb/
http://docs.fedoraproject.org/en-US/Fedora/14/html/Installation_Guide/
http://www.rational.com/products/rup/
http://www.omg.org/uml/
Extreme Programming, Kent Beck , Addison Wesley, 2000
http://www.extremeprogramming.org
http://www.xprogramming.com
http:/c2.com/cgi/wiki?ExtremeProgrammingRoadmap
http://thecoadletter.com/download/#fddguide
http://www.nebulon.com/arties/fdd/l estfdd.html