

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN TECNOLÓGICA

PORTAL WEB PARA EL CONSERVATORIO SUPERIOR NACIONAL DE MÚSICA

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO, EN
ANÁLISIS DE SISTEMAS INFORMÁTICOS**

VERÓNICA ELIZABETH VITERI EGAS

DIRECTORA: ING. ROSA NAVARRETE

Quito, junio del 2007

DECLARACIÓN

Yo, Verónica Elizabeth Viteri Egas, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

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

VERÓNICA ELIZABETH VITERI EGAS

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Verónica Elizabeth Viteri Egas.

ING. ROSA NAVARRETE MSc

AGRADECIMIENTOS

A mi familia, mis padres y hermanos por su apoyo y por enseñarme a cruzar los obstáculos sin buscar refugios, ni escondites. Gracias por enseñarme a valerme por mi misma.

A mis maestros, que han sido mi guía de estudio entregando sus conocimientos de ciencia y tecnología; que han construido en nosotros, responsabilidad, capacidad y dedicación en cada tarea y se han convertido en muy buenos amigos en quienes se puede confiar.

A mis amigos y compañeros que me han apoyado en cada momento y con quienes he compartido muchas alegrías, tristezas y desilusiones, a mis amigos con los que he vivido momentos inolvidables y juntos nos hemos esforzado para alcanzar el tan anhelado título. A mis amigos que siempre permanecerán en mi corazón.

Y sobre todo a mi querida Escuela Politécnica Nacional; por abrirme sus puertas, brindarme sus enseñanzas e indicarme el camino para ser una buena profesional.

DEDICATORIA

A mi hijo Gabrielito, tu has sido mi ángel, mi fuerza y voluntad, el deseo de ser mejor, mi felicidad y motivo de orgullo y dicha. Gracias mi chiquito por ser mi compañerito y por haberme enseñado a luchar para conseguir un futuro mejor para nosotros.

A mi querida madre, por su dulzura, su amor y ayuda, por ser mi amiga, confidente y consejera, gracias mamita por tu comprensión y apoyo en los momentos más difíciles, por haberme enseñado a ser una mujer de bien, y por tu presencia en cada etapa de mi vida.

A mi novio, por ser mi amigo, compañero y mi amor, por entregarme su comprensión, solidaridad, paciencia y dedicación, Gracias Mi Amor por tus exigencias, por tu optimismo y tu confianza en mí que me dieron animo y aliento para continuar el camino, gracias por tu apoyo incondicional y desinteresado.

Gracias Gabrielito, Mamita y David por su apoyo y entrega. Sin ustedes a mi lado no hubiese llegado a culminar mi carrera, los amo mucho.

Un especial agradecimiento a Dios que me dio sus bendiciones día a día para alcanzar mis metas.

RESUMEN

La aplicación esta desarrolla con la metodología OOHDM (Object Oriented Hypermedia Design Method), con servidor Apache, lenguaje de programación PHP 5.0 y base de Datos MySQL 5.0.

En este proyecto se presenta un desarrollo de aplicación hipermedia, que permite a los estudiantes del Conservatorio Superior Nacional de Música la visualización de calificaciones correspondientes a cada nivel de estudio.

Además se muestra reportes de horarios de docentes, nómina académica, eventos que se realizan en el Conservatorio.

Capítulo Uno; presenta el problema actual del manejo de información del Conservatorio Superior Nacional de Música, las soluciones planteadas para mejorar y canalizar la información existente.

Capítulo Dos; trata acerca de las herramientas utilizadas en el desarrollo del proyecto tales como: conceptos de HTML, Java Script, PHP y MySQL

Capítulo Tres; nos da una descripción de cada aspecto teórico utilizado para el desarrollo del proyecto como: Paradigma, Metodología y Modelamiento.

Capítulo Cuatro; plantea las conclusiones y recomendaciones que se pueden apreciar tras el desarrollo de la aplicación.

CONTENIDO

CAPITULO I

Aspectos Generales1
1.1 INTRODUCCIÓN1
1.2 PLANTEAMIENTO DEL PROBLEMA1
1.2.1 ÁMBITO1
1.2.2 DESCRIPCIÓN DEL PROBLEMA1
1.3 OBJETIVOS2
1.3.1 OBJETIVO GENERAL2
1.3.2 OBJETIVOS ESPECÍFICOS2

CAPITULO II

Marco Teórico4
2.1 INGENERÍA WEB4
2.1.1 SISTEMAS Y APLICACIONES BASADOS EN WEB (WEBAPPS)4
2.1.2 INGENIERÍA WEB4
2.2 ARQUITECTURA WEB MULTICAPAS5
2.3 HERRAMIENTAS DE DESARROLLO7
2.3.1 EL SERVIDOR WEB7
2.3.2 LENGUAJE DE INTERFAZ9
2.3.2.1 HTML9
2.3.2.2 JAVA SCRIPT10
2.3.3 LENGUAJE DE SERVIDOR11
2.3.3.1 PHP11
2.3.3.1.1 USOS DE PHP12
2.3.3.1.2 VENTAJAS12
2.3.4 BASE DE DATOS13
2.3.4.1 MYSQL13
2.3.4.1.1 MEJORAS FUTURAS14
2.3.4.1.2 CARACTERÍSTICAS14
2.4 HERRAMIENTAS DE APOYO15

2.4.1 HERRAMIENTAS DE ANÁLISIS Y DISEÑO	15
2.4.2 RATIONAL ROSE	15
2.4.2.1 CARACTERÍSTICAS	16
2.4.3 MACROMEDIA FLASH	17
2.4.4 MACROMEDIA FIREWORKS	17
CAPITULO III	
Aspectos Metodológicos	19
3.1 PARADIGMA ESPIRAL PARA LA WEB	19
3.2 METODOLOGÍA DE DESARROLLO	21
3.2.1 INTRODUCCIÓN A LA METODOLOGÍA OOHDM	22
3.2.2 FASES DE LA METODOLOGÍA	24
3.2.2.1 DISEÑO CONCEPTUAL	24
3.2.2.2 DISEÑO NAVEGACIONAL	25
3.2.2.3 DISEÑO DE INTERFAZ ABSTRACTA	26
3.2.2.4 IMPLEMENTACIÓN	26
3.2.3 ANÁLISIS	27
3.2.4 DISEÑO DEL SISTEMA	28
3.3 MODELAMIENTO	28
3.3.1 ANÁLISIS DE REQUERIMIENTOS	30
3.3.1.1 DIAGRAMAS DE CASOS DE USO	30
3.3.1.2 ELEMENTOS	31
3.3.1.2.1 ACTOR	31
3.3.1.2.2 CASOS DE USO	32
3.3.1.3 RELACIONES	32
3.3.2 MODELO ESTÁTICO	33
3.3.2.1 DIAGRAMA DE CLASES	33
3.3.2.1.1 LA CLASE	34
3.3.2.1.2 RELACIONES	34
3.3.2.2 DIAGRAMA DE OBJETOS	35
3.3.3 MODELO DINÁMICO	36
3.3.3.1 DIAGRAMAS DE SECUENCIA	36
3.3.3.1.2 ELEMENTOS	37
3.3.3.2 DIAGRAMAS DE COLABORACIÓN	38

3.3.4 MODELO FUNCIONAL	39
3.3.4.1 DIAGRAMA DE ESTADOS	39
3.3.4.1.1 ELEMENTOS	39
3.3.4.2 DIAGRAMA DE ACTIVIDADES	40
3.3.4.2.1 ELEMENTOS	41
3.3.5 MODELO ARQUITECTÓNICO	41
3.3.5.1 DIAGRAMA ARQUITECTÓNICO	41
3.3.5.1.1 ESTRUCTURA JERÁRQUICA	42
3.3.5.1.2 ESTRUCTURA LINEAL	42
3.3.5.1.3 LINEAL CON JERARQUÍA	43
3.3.5.1.4 ESTRUCTURA EN RED	43
3.3.5.2 DIAGRAMA NAVEGACIONAL	44
CAPITULO IV	
Conclusiones y Recomendaciones	45
4.1 CONCLUSIONES	45
4.2 RECOMENDACIONES	46
4.3 REFERENCIAS BIBLIOGRÁFICAS	47
ANEXOS	
Manual Técnico	50
IDENTIFICACIÓN DE ACTORES	52
DIAGRAMAS DE CASOS DE USO	53
DIAGRAMA DE CASOS DE USO TOTAL	53
DIAGRAMA DE CASOS DE USO MODULAR	54
DIAGRAMA DE CASOS DE USO POR ACTIVIDADES	55
DICCIONARIO DE CASOS DE USO	58
DIAGRAMA DE CLASES	62
DIAGRAMA DE OBJETOS	63
DICCIONARIO DE CLASES	64
DIAGRAMA DE SECUENCIA	68
DIAGRAMA DE COLABORACIÓN	74
DIAGRAMA DE ESTADOS	81
DIAGRAMA DE ACTIVIDADES	83
DIAGRAMA ARQUITECTÓNICO	87

DIAGRAMA NAVEGACIONAL89
CONSTRUCCIÓN91
Manual de Usuario102

CAPITULO I

ASPECTOS GENERALES

1.1 INTRODUCCIÓN

El avance y desarrollo de nuevas tecnologías de información han facilitado y simplificado las actividades del hombre contribuyendo en más tareas útiles para su vida. Los Sitios Web son una fuente muy importante de información que hoy en día todo tipo de instituciones, empresas, centros educativos; utilizan para comunicar sus servicios.

El Conservatorio Superior Nacional de Música no puede quedar aislado de estos beneficios por lo que sus directivos han pensado en el diseño de un portal Web, capaz de mostrar a la sociedad la información referente a la Institución, facilitando a aspirantes, alumnos y maestros su interacción.

1.2 PLANTEAMIENTO DEL PROBLEMA

1.2.1 ÁMBITO

El Conservatorio Superior Nacional de Música es un centro de formación académica en el Arte Musical, al cual acceden estudiantes desde los 6 años de edad en diferentes áreas, para obtener títulos de bachilleres en diversas especialidades de instrumentos.

1.2.2 DESCRIPCIÓN DEL PROBLEMA

La entrega de la información referente al Conservatorio Superior Nacional de Música se realiza de forma directa y personal, es necesario que el interesado

acuda personalmente a la Institución y de esta manera reciba la información los datos de su interés; al mismo tiempo es necesaria la presencia de la persona encargada en brindar la información.

Así mismo la entrega de calificaciones para los estudiantes es un trámite directo; es decir, de forma personal, el estudiante recibe sus calificaciones a través de la Secretaría General.

Es por esta razón y debido a la necesidad de organizar y mostrar información que se ha creído necesario crear un sitio Web que provea de toda la información y servicios de esta Institución a la comunidad y a través de la misma posibilitar que los estudiantes de este centro educativo puedan verificar información sobre notas y comunicaciones de la Institución.

1.3 OBJETIVOS

1.3.1 OBJETIVO GENERAL

Diseñar un sitio web para apoyar la difusión de los servicios que ofrece el Conservatorio Superior Nacional de Música y facilitar la interacción con sus estudiantes a través del diseño y posterior implementación de un sitio web.

1.3.2 OBJETIVOS ESPECIFICOS

Diseñar interfaces web que permita a usuarios (cibernautas) acceder de manera interactiva a información referente al Conservatorio.

Publicar calificaciones semestrales de manera que el estudiante tenga fácil acceso a éstas.

Producir reportes de datos personales de estudiantes, reportes para horarios, calificaciones y maestros.

Publicar información referente a eventos, fechas de matrículas, costos, cursos y actividades generales del Conservatorio.

Permitir la administración del sitio Web para mantener la información actualizada.

CAPITULO II

MARCO TEÓRICO

2.1 INGENIERIA WEB

2.1.1 SISTEMAS Y APLICACIONES BASADOS EN WEB (WEBAPPS)

Son aplicaciones o portales web que abarca todo, desde una página web simple hasta un sitio web completo que proporcione servicio completos, estas aplicaciones residen en internet, en una intranet o en una extranet, que hacen posible que una población extensa de usuarios dispongan de una variedad de contenido y funcionalidad. La función primaria de una wepapp es utilizar hipermedia para presentar al usuario el contenido de textos, gráficos, sonido y video.

A medida que las WebApps se integran cada vez más en grandes y pequeñas empresas, y cada vez es más importante la necesidad de construir sistemas fiables, utilizables y adaptables. Esta es la razón por la que es necesario un enfoque disciplinado para el desarrollo de las webapps.

2.1.2 INGENIERÍA WEB (IWEB)

Al igual que cualquier disciplina de ingeniería la Ingeniería Web aplica estrategias tácticas y métodos especializados. La ingeniería Web se diferencia de la ingeniería de software, pero toma prestados varios conceptos y principios básicos de software convencional, dando importancia a las mismas actividades técnicas y de gestión.

El proceso de ingeniería web comienza con una formulación del problema que pasa a resolverse con las webapps. Se planifica el proyecto y se analizan los

requisitos de la webapp, entonces se lleva a cabo el diseño de interfaces arquitectónico y del navegador. El sistema se implementa utilizando lenguajes y herramientas especializados asociados con la web y entonces comienzan las pruebas.

Dado que las webapps están en constante evolución, deben de establecerse los mecanismos para el control de configuraciones, garantía de calidad y soporte continuado. (Pressman, pag.525)

2.2 ARQUITECTURA WEB MULTICAPAS

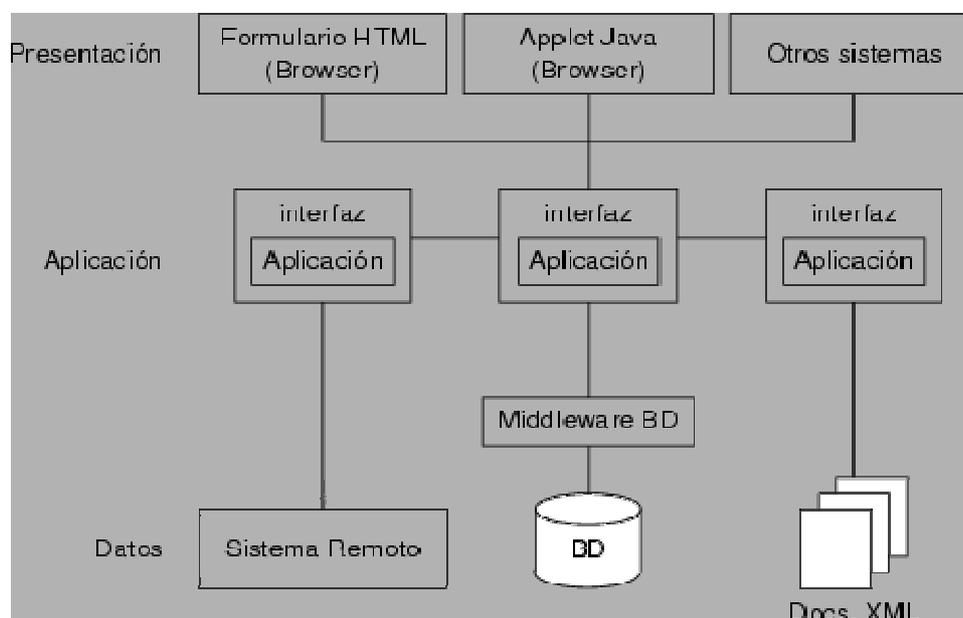
Al hablar del desarrollo de aplicaciones Web resulta adecuado presentarlas dentro de las aplicaciones multinivel. Los sistemas típicos cliente/servidor pertenecen a la categoría de las aplicaciones de dos niveles. La aplicación reside en el cliente mientras que la base de datos se encuentra en el servidor. En este tipo de aplicaciones el peso del cálculo recae en el cliente, mientras que el servidor hace la parte menos pesada, y eso que los clientes suelen ser máquinas menos potentes que los servidores. Además, está el problema de la actualización y el mantenimiento de las aplicaciones, ya que las modificaciones a la misma han de ser trasladada a todos los clientes.

Para solucionar estos problemas se ha desarrollado el concepto de arquitecturas de tres niveles: interfaz de presentación, lógica de la aplicación y los datos.

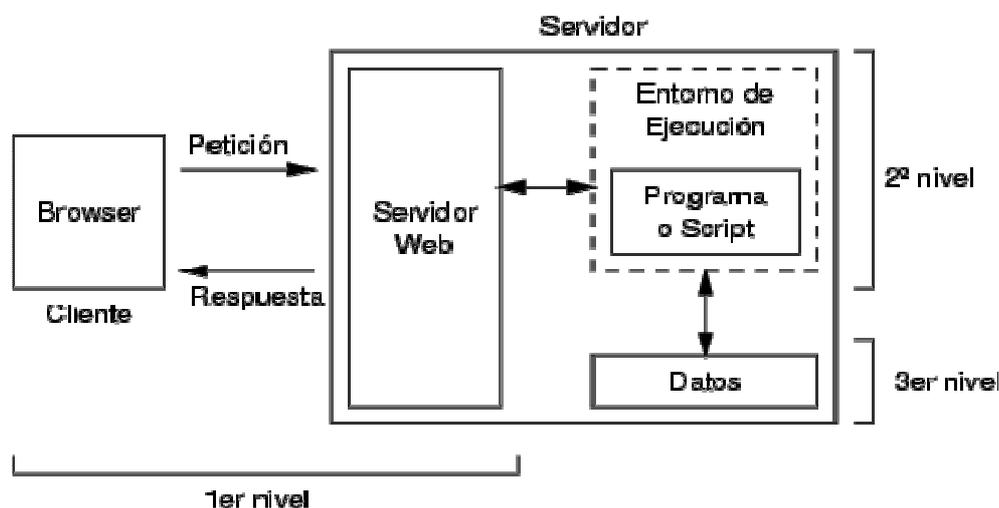
La capa intermedia es el código que el usuario invoca para recuperar los datos deseados. La capa de presentación recibe los datos y los formatea para mostrarlos adecuadamente. Esta división entre la capa de presentación y la de la lógica permite una gran flexibilidad a la hora de construir aplicaciones, ya que se pueden tener múltiples interfaces sin cambiar la lógica de la aplicación.

La tercera capa consiste en los datos que gestiona la aplicación. Estos datos pueden ser cualquier fuente de información como una base de datos.

Convertir un sistema de tres niveles a otro multinivel es fácil ya que consiste en extender la capa intermedia permitiendo que convivan múltiples aplicaciones en lugar de una sola.



La arquitectura de las aplicaciones Web suelen presentar un esquema de tres niveles.



El primer nivel consiste en la capa de presentación que incluye no sólo el navegador, sino también el servidor web que es el responsable de dar a los datos un formato adecuado. El segundo nivel está referido habitualmente a algún tipo de

programa o script. Finalmente, el tercer nivel proporciona al segundo los datos necesarios para su ejecución.

Una aplicación Web típica recogerá datos del usuario (primer nivel), los enviará al servidor, que ejecutará un programa (segundo y tercer nivel) y cuyo resultado será formateado y presentado al usuario en el navegador (primer nivel otra vez).

<http://www.infor.uva.es/~jvegas/cursos/buendia/pordocente/node21.html>

2.3 HERRAMIENTAS DE DESARROLLO

2.3.1 EL SERVIDOR WEB

El servidor Web es un programa que corre sobre el servidor que escucha las peticiones HTTP (hypertext transfer protocol) que le llegan y las satisface. Este protocolo está diseñado para transferir lo que llamamos hipertextos, páginas web o páginas HTML (hypertext markup language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de sonidos. Dependiendo del tipo de la petición, el servidor Web buscará una página Web o bien ejecutará un programa en el servidor. De cualquier modo, siempre devolverá algún tipo de resultado HTML al cliente o navegador que realizó la petición.

Un servidor web se encarga de mantenerse a la espera de peticiones HTTP llevada a cabo por un cliente HTTP que solemos conocer como navegador. El navegador realiza una petición al servidor y éste le responde con el contenido que el cliente solicita. A modo de ejemplo, al teclear <http://www.wikipedia.org/> en nuestro navegador, éste realiza una petición HTTP al servidor de dicha dirección. El servidor responde al cliente enviando el código HTML de la página; el cliente, una vez recibido el código, lo interpreta y lo muestra en pantalla. Como vemos con este ejemplo, el cliente es el encargado de interpretar el código HTML, es

decir, de mostrar las fuentes, los colores y la disposición de los textos y objetos de la página; el servidor tan sólo se limita a transferir el código de la página sin llevar a cabo ninguna interpretación de la misma.

Sobre el servicio web clásico podemos disponer de aplicaciones web. Éstas son fragmentos de código que se ejecutan cuando se realizan ciertas peticiones o respuestas HTTP. Hay que distinguir entre:

- Aplicaciones en el lado del cliente: el cliente web es el encargado de ejecutarlas en la máquina del usuario. Son las aplicaciones tipo Java o Javascript: el servidor proporciona el código de las aplicaciones al cliente y éste, mediante el navegador, las ejecuta. Es necesario, por tanto, que el cliente disponga de un navegador con capacidad para ejecutar aplicaciones (también llamadas scripts). Normalmente, los navegadores permiten ejecutar aplicaciones escritas en lenguaje javascript y java, aunque pueden añadirse más lenguajes mediante el uso de plugins
- Aplicaciones en el lado del servidor: el servidor web ejecuta la aplicación; ésta, una vez ejecutada, genera cierto código HTML; el servidor toma este código recién creado y lo envía al cliente por medio del protocolo HTTP.

Algunos servidores web importantes son:

- Apache
- IIS
- Cherokee

Otros servidores, más simples pero más rápidos, son:

- lighttpd
- thttpd

<http://www.infor.uva.es/~jvegas/cursos/buendia/pordocente/node20.html>

2.3.2 LENGUAJE DE INTERFAZ

2.3.2.1 HTML (Lenguaje de Formato de Documentos de Hipertexto)

El HTML, acrónimo inglés de HyperText Markup Language (lenguaje de marcas hipertextuales), lenguaje de marcación diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web. Gracias a Internet y a los navegadores del tipo Internet Explorer, Opera, Firefox o Netscape, el HTML se ha convertido en uno de los formatos más populares que existen para la construcción de documentos y también de los más fáciles de aprender.

El lenguaje HTML puede ser creado y editado con cualquier editor de textos básico, como puede ser el Bloc de Notas de Windows (o *Notepad*), o cualquier otro editor que admita texto sin formato como GNU Emacs, Microsoft Wordpad, TextPad etc.

Existen además, otros programas para la realización de sitios Web o edición de código HTML, como por ejemplo Microsoft FrontPage, el cual tiene un formato básico parecido al resto de los programas de Office. También existe el famoso software de Adobe (que adquirió la empresa Macromedia) llamado Dreamweaver, siendo uno de los más utilizados en el ámbito de diseño y programación Web. Estos programas se les conoce como editores WYSIWYG o What You See Is What You Get (en español: "lo que ves es lo que obtienes"). Esto significa que son editores los cuales van mostrando el resultado de lo que se está editando en tiempo real a medida que se va desarrollando el documento, además de tener la opción de trabajar con la vista preliminar, tiene su propia sección HTML la cual va generando todo el código a medida que se va trabajando.

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. <http://es.wikipedia.org/wiki/HTML>

2.3.2.2 JAVASCRIPT

JavaScript es un lenguaje interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C.

Al contrario que Java, JavaScript no es un lenguaje orientado a objetos propiamente dicho, ya que no dispone de Herencia, es más bien un lenguaje basado en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad.

El lenguaje fue inventado por Brendan Eich en la empresa Netscape Communications, que es la que fabricó 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 tareas y operaciones en el marco de la aplicación únicamente 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.

Javascript se puede incluir en cualquier documento HTML, o todo aquel que termine traducándose en HTML en el navegador del cliente; ya sea PHP, ASP, SVG. <http://developer.mozilla.org/es/docs/JavaScript>

2.3.3 LENGUAJE DE SERVIDOR

2.3.3.1 PHP

PHP es un lenguaje de programación usado generalmente para la creación de contenido para sitios Web. PHP es un acrónimo recurrente que significa "PHP Hypertext Pre-processor" (inicialmente PHP Tools, o, Personal Home Page Tools), y se trata de un lenguaje interpretado usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios web. Últimamente también para la creación de otro tipo de programas incluyendo aplicaciones con interfaz gráfica usando la biblioteca GTK+.

El fácil uso y la similitud con los lenguajes más comunes de programación estructurada, como C y Perl, permiten a la mayoría de los programadores experimentados crear aplicaciones complejas con una curva de aprendizaje muy suave. También les permite involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones y prácticas.

Su interpretación y ejecución se da en el servidor, en el cual se encuentra almacenado el script, y el cliente sólo recibe el resultado de la ejecución. Cuando el cliente hace una petición al servidor para que le envíe una página web, generada por un script PHP, el servidor ejecuta el intérprete de PHP, el cual procesa el script solicitado que generará el contenido de manera dinámica, pudiendo modificar el contenido a enviar, y regresa el resultado al servidor, el cual se encarga de regresárselo al cliente. Además es posible utilizar PHP para generar archivos PDF, Flash, así como imágenes en diferentes formatos, entre otras cosas.

Permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, Postgres, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite; lo cual permite la creación de Aplicaciones web muy robustas.

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

2.3.3.1.1 Usos de PHP

Los principales usos del PHP son los siguientes:

- Programación de páginas web dinámicas, habitualmente en combinación con el motor de base datos MySQL, aunque cuenta con soporte nativo para otros motores, incluyendo el estándar ODBC, lo que amplía en gran medida sus posibilidades de conexión.
- Programación en consola, al estilo de Perl o Shell scripting.
- Creación de aplicaciones gráficas independientes del navegador, por medio de la combinación de PHP y GTK (GIMP Tool Kit), lo que permite desarrollar aplicaciones de escritorio en los sistemas operativos en los que está soportado.

2.3.3.1.2 Ventajas de PHP

- Es un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL
- Leer y manipular datos desde diversas fuentes, incluyendo datos que pueden ingresar los usuarios desde formularios HTML.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- Posee una amplia documentación en su página oficial ([1]), entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.
- Permite crear los formularios para la web.

- Biblioteca nativa de funciones sumamente amplia e incluida
- No requiere definición de tipos de variables ni manejo detallado del bajo nivel. <http://es.php.net/manual/es/>

2.3.4 BASE DE DATOS

2.3.4.1 MYSQL

MySQL es un sistema de gestión de base de datos, multihilo y multiusuario con más de seis millones de instalaciones. MySQL desarrolla MySQL como software libre en un esquema de licenciamiento dual. Por un lado lo ofrece bajo la GNU GPL, pero, empresas que quieran incorporarlo en productos privativos pueden comprar a la empresa una licencia que les permita ese uso.

Está desarrollado en su mayor parte en ANSI C. MySQL está escrito en una mezcla de C y C++.

MySQL es muy utilizado en aplicaciones web como MediaWiki o Drupal, en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python), y por herramientas de seguimiento de errores como Bugzilla. Su popularidad como aplicación web está muy ligada a PHP, que a menudo aparece en combinación con MySQL. MySQL es una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones.

MySQL funciona sobre múltiples plataformas, incluyendo AIX, BSD, FreeBSD, HP-UX, GNU/Linux, Mac OS X, NetBSD, Novell Netware, OpenBSD, OS/2 Warp, QNX, SGI IRIX, Solaris, SunOS, SCO OpenServer, SCO UnixWare, Tru64, Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, Windows Vista y otras versiones de Windows. También existe MySQL para OpenVMS.

Inicialmente, MySQL carecía de elementos considerados esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. Poco a

poco los elementos de los que carecía MySQL están siendo incorporados tanto por desarrollos internos, como por desarrolladores de software libre. Entre las características disponibles en las últimas versiones se puede destacar:

- Amplio subconjunto del lenguaje SQL. Algunas extensiones son incluidas igualmente.
- Disponibilidad en gran cantidad de plataformas y sistemas.
- Diferentes opciones de almacenamiento según si se desea velocidad en las operaciones o el mayor número de operaciones disponibles.
- Transacciones y claves foráneas.
- Conectividad segura.
- Replicación.
- Búsqueda e indexación de campos de texto.

2.3.4.1.1 Mejoras futuras

- Particionado de la base de datos
- Backup en línea para todos los motores de almacenamiento
- Replicación segura
- Restricciones a nivel de columna
- Planificación de eventos
- Funciones XML

2.3.4.1.2 Características distintivas

Las siguientes características son implementadas únicamente por MySQL:

- Múltiples motores de almacenamiento (MyISAM, Merge, InnoDB, BDB, Memory/heap, MySQL Cluster, Federated, Archive, CSV, Blackhole y Example en 5.x), permitiendo al usuario escoger la que sea más adecuada para cada tabla de la base de datos.
- Agrupación de transacciones, reuniendo múltiples transacciones de varias conexiones para incrementar el número de transacciones por segundo.

<http://losinvisibles.net/como/comoMySQL.html>

2.4 HERRAMIENTAS DE APOYO

2.4.1 HERRAMIENTAS DE ANALISIS Y DISEÑO

Las herramientas de análisis y diseño permiten realizar un Modelado Visual el cual es un proceso que representa gráficamente el sistema o software, permitiendo construir la interfaz del sistema.

Características Importantes de las herramientas de análisis

- Identifica requisitos y comunica información
- Se centra en cómo los componentes del sistema interactúan entre ellos, sin entrar en detalles excesivos.
- Permite ver las relaciones entre los componentes del diseño
- Mejora la comunicación entre los miembros del equipo usando un lenguaje gráfico.

http://exa.unne.edu.ar/depar/areas/informatica/anasistem2/public_html/apuntes/maf/

2.4.2 RATIONAL ROSE

Rational Rose es la herramienta CASE que comercializan los desarrolladores de UML y que soporta de forma completa la especificación del UML 1.1. Esta herramienta propone la utilización de cuatro tipos de modelo para realizar un diseño del sistema, utilizando una vista estática y otra dinámica de los modelos del sistema, uno lógico y otro físico. Permite crear y refinar estas vistas creando de esta forma un modelo completo que representa el dominio del problema y el sistema de software.

Rational Rose utiliza un proceso de desarrollo iterativo controlado (controlled iterative process development), donde el desarrollo se lleva a cabo en una secuencia de iteraciones. Cada iteración comienza con una primera aproximación del análisis, diseño e implementación para identificar los riesgos del diseño, los cuales se utilizan para conducir la iteración, primero se identifican los riesgos y después se prueba la aplicación para que éstos se hagan mínimos. Cuando la implementación pasa todas las pruebas que se determinan en el proceso, ésta se revisa y se añaden los elementos modificados al modelo de análisis y diseño. Una vez que la actualización del modelo se ha modificado, se realiza la siguiente iteración.

Rational Rose proporciona mecanismos para realizar la denominada Ingeniería Inversa, es decir, a partir del código de un programa, se puede obtener información sobre su diseño.

2.4.2.1 Características de Rational Rose

- Diseño dirigido por modelos que redundan en una mayor productividad de los desarrolladores, admitiendo UML, COM, OMT y Booch
- Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo
- Soporte OLE
- Disponibilidad en múltiples plataformas
- Da soporte al Proceso Unificado de Rational (RUP)
- Modelado de Negocio
- Captura de Requisitos (parcial)

- Análisis y Diseño (Completo)
- Implementación (como ayuda)

<http://www.rational.com/uml/>

2.4.3 MACROMEDIA FLASH

Flash es una herramienta de edición con la que pueden crearse desde animaciones simples hasta complejas aplicaciones Web interactivas, como una tienda en línea. Las aplicaciones de Flash pueden enriquecerse añadiendo imágenes, sonido y vídeo. Flash incluye muchas funciones que la convierten en una herramienta con muchas prestaciones sin perder por ello la facilidad de uso. Entre dichas funciones destacan: la posibilidad de arrastrar y soltar componentes de la interfaz de usuario, comportamientos incorporados que añaden código ActionScript al documento y varios efectos especiales que pueden añadirse a los objetos.

Es la herramienta perfecta para los diseñadores de páginas Web, profesionales de medios interactivos o personas especializadas que desarrollen contenido multimedia. Pone énfasis en la creación, importación y manipulación de distintos tipos de medios (audio, vídeo, mapas de bits, vectores, texto y datos).

2.4.4 MACROMEDIA FIREWORKS

Macromedia Fireworks es un potente editor gráfico, con una visión al de la optimización web de estos gráficos para vincularlos directamente a Macromedia Dreamweaver o al web.

Fireworks puede utilizarse para crear, editar y animar gráficos Web, añadir interactividad avanzada y optimizar imágenes en entornos profesionales. En Fireworks es posible crear y modificar imágenes vectoriales y de mapa de bits en una sola aplicación. Todo es modificable en todo momento.

Fireworks se integra con otros productos de Macromedia, como Dreamweaver, Flash, FreeHand y Director, y con otros editores HTML y aplicaciones gráficas de uso frecuente para ofrecer una solución Web global. Los elementos gráficos de Fireworks pueden exportarse fácilmente con código HTML y JavaScript adaptado al editor de HTML que se utilice.

Fireworks es una aplicación versátil para crear, editar y optimizar gráficos Web. Permite crear y editar imágenes de mapa de bits y vectoriales, diseñar efectos Web, como rollovers y menús emergentes, recortar y optimizar elementos gráficos para reducir su tamaño de archivo y automatizar tareas repetitivas para ahorrar tiempo.

Cuando un documento está terminado, se puede exportar o guardar en formato de archivo JPEG, GIF u otros -- junto con archivos HTML que contengan tablas HTML y código JavaScript -- para utilizarlo en la Web. También es posible exportar tipos de archivos específicos de otras aplicaciones, como Photoshop o Macromedia Flash, si en un momento dado hace falta seguir trabajando en esa aplicación.

<http://www.macromedia.com/es/software/fireworks/>

CAPITULO III

ASPECTOS METODOLÓGICOS

Para el siguiente Proyecto se ha definido los siguientes aspectos metodológicos:

3.1 PARADIGMA ESPIRAL PARA LA WEB

El paradigma espiral no es sino un modelo de ciclo de vida utilizado en la ingeniería de software; fue desarrollada por Barry Boehm en el año de 1985.

Representa cada actividad realizada en el desarrollo del sistema como un bucle. Las actividades no están fijadas a prioridad sino que se eligen de acuerdo al análisis de riesgos comenzando con un bucle interior.

Este modelo espiral es un esquema de ciclo de vida que representa fases de especificación, diseño, realización, y evaluación y que están marcados en los cuadrantes gráficos de este modelo (Figura 3.1.1).

Las fases de este paradigma son:

1. Planificación: determinación de objetivos, alternativas y restricciones.
2. Análisis de riesgo: análisis de alternativas e identificación/resolución de riesgos.
3. Ingeniería: desarrollo del producto de "siguiente nivel"
4. Evaluación del cliente: valoración de los resultados de la ingeniería

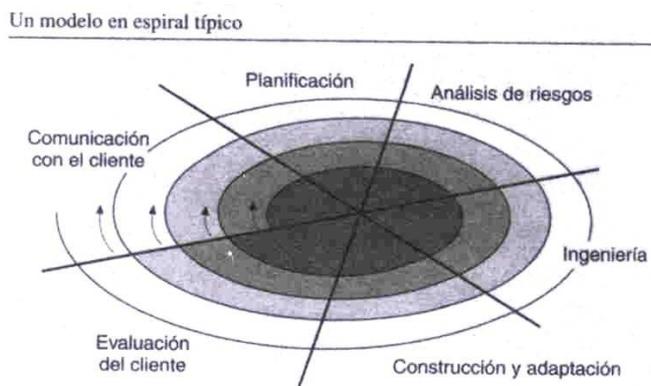


Figura 3.1.1

Fuente: <http://www.umsanet.edu.bo/docentes/gchoque/TextolngSoftw.pdf>

En el desarrollo de Aplicaciones Web debido a que la información ya no es un recurso estático controlado sino más bien un entorno dinámico de información, es importante aplicar principios sólidos de ingeniería del software. Esto se ha conseguido mediante la utilización de un modelo de proceso eficaz como lo es el modelo espiral que varia en dos fases y que proporciona los recursos necesarios a la Aplicación Web.

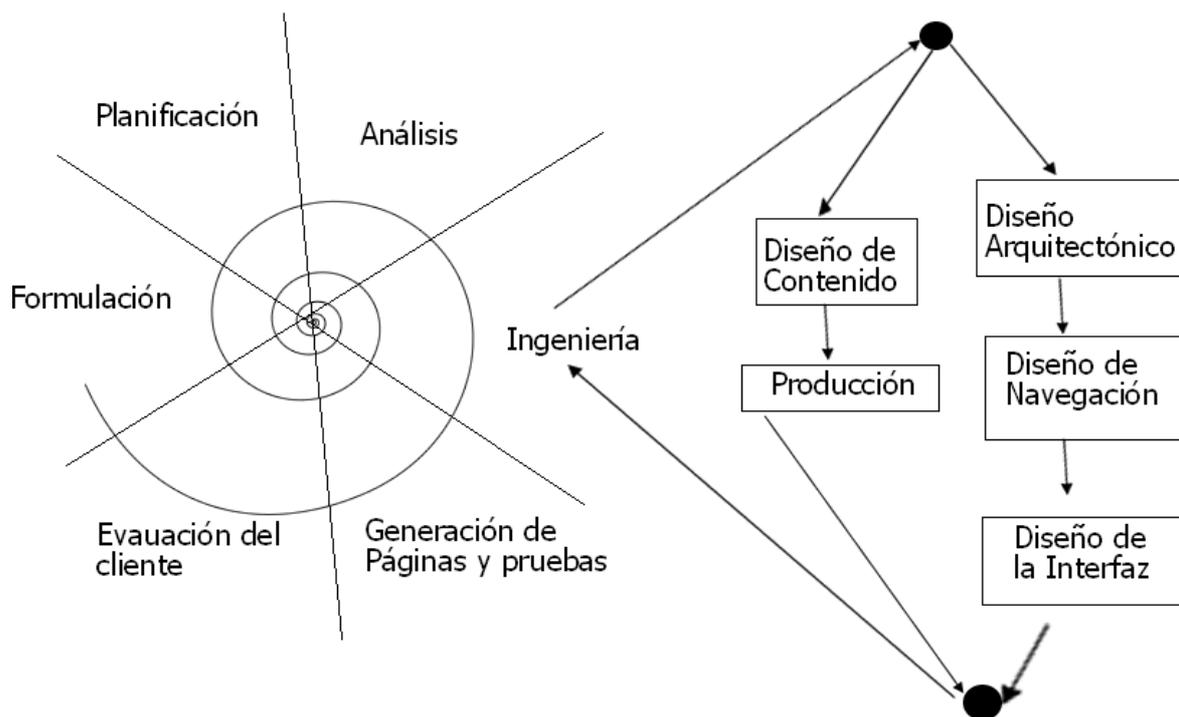


Figura 3.1.2

Fuente:(Pressman, pag.525)

En el caso de las aplicaciones Web las seis fases representadas en la Figura 3.1.2 se define de la siguiente manera:

Formulación: actividad que identifica las metas y los objetivos de la Aplicación Web.

Planificación: Estima el coste global del proyecto, evalúa los riesgos asociados con el esfuerzo del desarrollo.

Análisis: establece los requisitos técnicos para la Aplicación Web e identifica elementos que se van a incorporar al contenido. Se deberá prever los requisitos de diseño gráfico.

Ingeniería: es el desarrollo del producto en la cual se incorpora dos tareas paralelas, como se muestra en la anterior figura.

- Diseño del Contenido y Producción: El objetivo de estas tareas es diseñar el contenido de texto, gráfico y video que se implementarán en la Aplicación Web.

Generación de páginas: es una actividad de construcción que hace uso de las herramientas automatizadas para la creación de Aplicaciones Web.

El contenido definido en la actividad de ingeniería se fusiona con los diseños arquitectónicos, de navegación y de la interfaz para elaborar páginas compatibles en html, xml, y otros lenguajes propios para este tipo de aplicaciones.

Evaluación del cliente: es la actividad en la cual se revisa cada incremento producido por parte del proceso y en la cual las pruebas ejercitan la navegación e intentan descubrir errores de guiones o formularios para asegurar el correcto funcionamiento de la aplicación. (Pressman, pag.525)

3.2 METODOLOGÍA DE DESARROLLO

El creciente desarrollo de la tecnología en el Internet y los avances de las comunicaciones han provocado la aparición de nuevas metodologías que ofrecen un marco teórico como referencia adecuada para el diseño y desarrollo de sistemas de información Web.

Existen varias metodologías que presentan diversas maneras de manejar el desarrollo de una aplicación Web.

3.2.1 INTRODUCCIÓN A LA METODOLOGÍA OOHDM

El número de aplicaciones Web ha crecido notablemente en la actualidad, cada una difiere en complejidad y tamaño, es necesaria la aplicación de una metodología de diseño óptima y sistemática que disminuya la complejidad y admita la reusabilidad. A la vez es muy difícil lograr tareas en una aplicación Web que aproveche potencialidades de paradigma de navegación en sitios Web y fácil manejo para el usuario.

La navegación de una página Web es un factor importante y es la clave del éxito de la aplicación, una navegación clara y de cómodo acceso que permita al usuario entender donde y como llegar a un lugar determinado es la mejor señal de un buen diseño de la aplicación.

Object Oriented Hypermedia Design Methodology (OOHDM, Método de Diseño Hipermedia Orientado a Objetos) es una metodología propuesta por Gustavo Rossi y Daniel Schwabe en el año de 1998, que tiene por objetivo simplificar y a la vez hacer más eficaz el diseño de aplicaciones hipermedia, a su vez es una extensión de HDM con orientación a objetos, que se está convirtiendo en una de las metodologías más utilizadas. Ha sido usada para diseñar diferentes tipos de aplicaciones hipermedia como galerías interactivas, presentaciones multimedia y, sobre todo, numerosos sitios Web.

Dadas las especiales características de las aplicaciones Web en diseño y dinamismo es necesario utilizar una metodología de desarrollo de software. Con esta metodología se pretende desarrollar un producto de calidad y de garantía que satisfaga las necesidades para un buen software.

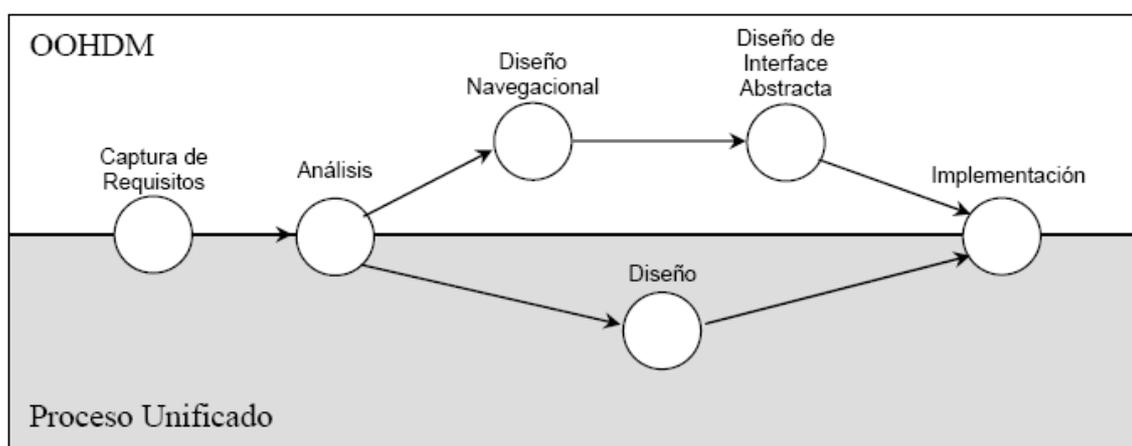
OOHDM propone el desarrollo de aplicaciones hipermedia a través de un proceso compuesto por cuatro etapas: diseño conceptual, diseño navegacional, diseño de interfaces abstractas e implementación.

En sus comienzos OOHDM no contemplaba la fase de captura y definición de requisitos, pero actualmente propone el uso de User Interaction Diagrams (UIDs)

definidos por Vilain, Schwabe & Sieckenius (2000). Esta propuesta parte de los casos de uso, que se considera una técnica muy difundida, ampliamente aceptada y fácilmente entendible por los usuarios y clientes no expertos, igualmente, resalta la necesidad de empezar el diseño del sistema, especialmente en los entornos Web, teniendo un claro y amplio conocimiento de las necesidades de interacción, o lo que es lo mismo de la forma en la que el usuario va a comunicarse con el sistema.

Es por estas razones que OOHDM propone que la comunicación con el usuario se realice utilizando los casos de uso y a partir de ellos se elaboren los diagramas de interacción. Estos diagramas son modelos gráficos que representan la interacción entre el usuario y el sistema, sin considerar aspectos específicos de la interfaz.

OOHDM centra el desarrollo de un sistema de información Web en torno al modelo conceptual de clases. Este diagrama debe surgir de los requisitos que se definan del sistema para lo cual se crearon los casos de uso. Esta metodología también propone la descripción de escenarios en forma textual y gráfica para cada tipo de usuario, como etapa previa al diseño de la navegación. <http://www.hipertexto.info/documentos/oohdm.htm>



3.2.2 FASES DE LA METODOLOGÍA

OOHDM propone un conjunto de tareas que pueden resultar costosas a corto plazo, pero a mediano y largo plazo reducen notablemente los tiempos de desarrollo al tener como objetivo principal la reusabilidad de diseño, y así simplificar el coste de evoluciones y mantenimiento.

Esta metodología plantea el diseño de una aplicación de este tipo a través de cuatro fases que se desarrollan de un modo iterativo.

<http://www.hipertexto.info/documentos/oohdm.htm>

3.2.2.1 FASE 1.- Diseño Conceptual

En la primera fase OOHDM construye un esquema conceptual representado por los objetos o clases y las relaciones entre dichos objetos. Se puede usar un modelo de datos estructural. El modelo OOHDM propone su esquema conceptual basado en clases, relaciones y subsistemas.

Las clases son descritas como en los modelos orientados a objetos tradicionales. Sin embargo, los atributos pueden ser de múltiples tipos para representar perspectivas diferentes de las mismas entidades del mundo real.

Se usa notación similar a UML (Lenguaje de Modelado Unificado) El esquema de las clases consiste en un conjunto de clases conectadas por relaciones. Los objetos son instancias de las clases. Las clases son usadas durante el diseño navegacional para derivar nodos, y las relaciones son usadas para construir enlaces. <http://www.hipertexto.info/documentos/oohdm.htm>

3.2.2.2 FASE 2.- Diseño Navegacional

En primera instancia las aplicaciones Web fueron pensadas para realizar navegación en un espacio de información, utilizando modelos simples de datos de hipermedia. En OOHDM, la navegación se considera un aspecto demasiado importante en el diseño de aplicaciones Web.

El modelo navegacional es construido como una *vista* sobre un diseño conceptual, admitiendo la construcción de modelos diferentes de acuerdo con los diferentes perfiles de usuarios. Cada modelo navegacional provee una vista subjetiva del diseño conceptual. Este diseño está expresado en dos esquemas: el esquema de clases navegacionales y el esquema de contextos navegacionales.

En OOHDM existe un conjunto de clases navegacionales: nodos, enlaces y estructuras de acceso. La estructura de los nodos y los enlaces son las tradicionales de las aplicaciones hipermedia.

La principal estructura ambigua del espacio navegacional es la noción de contexto navegacional. Un contexto navegacional es un conjunto de nodos, enlaces, clases de contextos, y otros contextos navegacionales. Los contextos navegacionales juegan un rol similar a las colecciones y fueron inspirados sobre el concepto de contextos anidados. Organizan el espacio navegacional en conjuntos convenientes que pueden ser recorridos en un orden particular y que deberían ser definidos como caminos para ayudar al usuario a lograr la tarea deseada.

Los nodos son enriquecidos con un conjunto de clases especiales que permiten de un nodo observar y presentar atributos (incluidos las anclas), así como comportamientos cuando se navega en un particular contexto.
<http://www.hipertexto.info/documentos/ooohdm.htm>

3.2.2.3 FASE 3.- Diseño de Interfaz Abstracta

La tercera fase de OOHDM está dedicada a la interfaz abstracta. Así se define la forma como deben aparecer los contextos navegacionales de la fase dos. En esta fase se incluye el modo en que los objetos de interfaz activan la navegación y el resto de funcionalidades de la aplicación, se definen los objetos de interfaz y se asocian con objetos de navegación. La separación entre el diseño navegacional y el de interfaz abstracta permite construir diferentes interfaces para el mismo modelo navegacional.

El aspecto de la interfaz de usuario de aplicaciones Web es un punto crítico en el desarrollo que las modernas metodologías tienden a descuidar. En OOHDM se utiliza el diseño de interfaz abstracta para describir la interfaz del usuario de la aplicación de hipertexto. El modelo de interfaz especifica la organización y comportamiento de la interfaz.

<http://www.hipertexto.info/documentos/oohdm.htm>

3.2.2.4 FASE 4.- Implementación

En esta última fase se debe implementar el diseño.

En la fase de implementación se debe tener en cuenta el entorno particular en el cual se va a correr la aplicación. Al llegar a esta fase, el primer paso que se debe realizar es definir los ítems de información que son parte del dominio del problema. Debe identificarse también, cómo son organizados los ítems de acuerdo con el perfil del usuario y su tarea; decidir qué interfaz debería ver y cómo debería comportarse. A fin de implementar todo en un entorno web, el diseñador debe decidir además qué información debe ser almacenada.

<http://www.hipertexto.info/documentos/oohdm.htm>

3.2.3 ANÁLISIS

El análisis de sistemas y aplicaciones Web representan una sucesión de actividades de ingeniería Web que comienza con la identificación de metas globales para la Aplicación Web y termina con el desarrollo de un modelo de análisis o especificación de los requisitos para el sistema.

Para crear un modelo de análisis completo para una Aplicación Web se elabora el ámbito definido durante la actividad de formulación en el cual se realizan cuatro tipos de análisis diferentes:

Análisis del Contenido: Se trata de la identificación del espectro completo del contenido que se va a proporcionar. En el contenido se incluyen datos de texto, gráficos, imágenes, video y sonido.

Análisis de la Interacción: es la descripción detallada de la interacción del usuario y la Aplicación Web.

Análisis Funcional: Aquí se realiza una descripción detallada de todas las funciones y operaciones. Define las operaciones que se aplicarán en el contenido de la Aplicación Web.

Análisis de la Configuración: Se efectúa una descripción detallada del entorno y de la aplicación detallada del entorno y de la infraestructura en donde reside la Aplicación Web; puede ser una intranet, extranet o Internet. (Pressman, pag.527)

Necesidades y Objetivos del Análisis

La presencia de una Empresa o Institución de cualquier tipo en Internet, debe ser valorada, analizada y programada del mismo modo que la presencia de la empresa en cualquier otro medio, revistas técnicas, prensa diaria, TV, radio, etc. Como resultado de la política comercial de la empresa y en consonancia con su presencia en otros medios, se determinarán los objetivos de esta acción, las

características deseadas para la aplicación y como van a ser mostradas al usuario. Lógicamente, las necesidades manifestadas por los clientes, la existencia en la Red de páginas creadas por competidores, etc. han de tenerse en cuenta al tomar esta decisión.

3.2.4 DISEÑO DEL SISTEMA

Para realizar un diseño eficaz basado en web, se deberá trabajar reutilizando cuatro elementos técnicos:

Principios y métodos de diseño: cuando se crean aplicaciones web se pueden reutilizar los métodos de diseño que se utilizan para los sistemas orientados a objetos.

Reglas de Oro: conjunto de heurísticas de diseño que se podrán volver a aplicar durante el diseño de aplicaciones nuevas.

Configuraciones de Diseño: son un enfoque genérico para resolver problemas. En el contexto de las aplicaciones Web, las configuraciones de diseño se pueden aplicar no solo a elementos funcionales de una aplicación, sino también a los documentos, gráficos y estética general de un Sitio Web.

Plantillas: Las plantillas se pueden utilizar para proporcionar un marco de trabajo estructurado de cualquier configuración de diseño o documento a utilizar dentro de una aplicación Web.(Pressman, pag.527)

3.3 MODELAMIENTO

El modelamiento es un lenguaje utilizado para representar sistemas de software, no es un método de desarrollo de software. En el desarrollo de estos sistemas se

puede utilizar u lenguaje de modelado como lo es UML (Lenguaje Unificado de Modelado).

Es importante desarrollar el modelo para un sistema de software antes de su construcción o renovación. Mientras que la complejidad de los sistemas aumenta se hace más importante buenas técnicas de modelado. Un lenguaje de modelado debe incluir:

Elementos fundamentales que modelan conceptos y la semántica.

Notación para la representación visual de los elementos del modelo.

Guías de consulta para el uso de los desarrolladores, vendedores y la enseñanza.

Un modelo representa a un sistema software desde una perspectiva específica. Al igual que la planta y el alzado de una figura en dibujo técnico nos muestran la misma figura vista desde distintos ángulos, cada modelo nos permite fijarnos en un aspecto distinto del sistema.

UML fue creado por tres de los más importantes metodólogos Booch, Rumbaugh y Jacobson.

UML o Lenguaje Unificado de Modelado prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos y describe la semántica esencial de estos diagramas y los símbolos en ellos utilizados.

El Lenguaje de Modelado Unificado (UML) es una lengua visual para especificar, formular y documentar los sistemas de información.

UML ofrece 9 tipos de diagramas con los cuales se pueden modelar sistemas.

UML no es, por tanto, un método, sino varios. Se trata de una estandarización o consolidación de muchas notaciones y modelos usados anteriormente

UML es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos.

(<http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/multiple-html/c385.html>).

3.3.1 ANALISIS DE REQUERIMIENTOS

En el proceso de desarrollo de un sistema, sea o no para la Web, se debe tener en cuenta la identificación de requisitos.

La definición de las necesidades del sistema es un proceso complejo, pues en él hay que identificar los requisitos que el sistema debe cumplir para satisfacer las necesidades de los usuarios finales y de los clientes.

Para realizar este proceso se utilizará un conjunto de técnicas, cuyo uso también proponen las distintas metodologías para el desarrollo de aplicaciones Web. (Modelando con UML, pag.30)

3.3.1.1 Diagramas de Casos de Uso

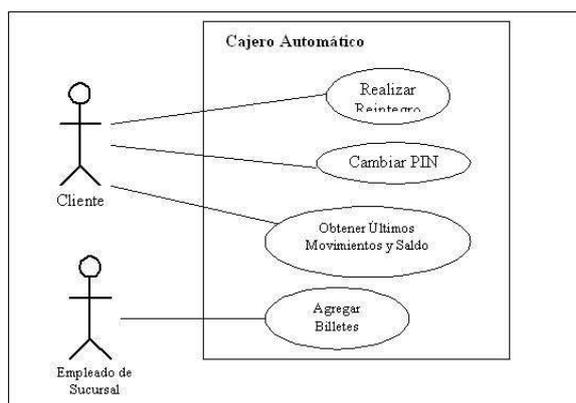
En 1994 Ivar Jacobson propone la utilización de Casos de Uso como una técnica formal para capturar requisitos.

Los Diagramas de Casos de Uso son un estándar de UML que representa lo que hace el sistema y como lo relaciona con su entorno.

Un Diagrama de casos de uso también representa los distintos requerimientos que le hacen los usuarios al sistema, especificando las características de funcionalidad y comportamiento durante la interacción con los usuarios u otros sistemas.

Se basan en la idea de que la mejor manera de comprender un sistema es a través de la descomposición funcional.

En el diagrama de casos de uso se representa también el sistema como una caja rectangular con el nombre en su interior. Los casos de uso están en el interior de la caja del sistema, y los actores fuera, y cada actor está unido a los casos de uso en los que participa mediante una línea. (Modelando con UML, pag.34)



3.3.1.2 Elementos

3.3.1.2.1 Actor

Un actor es algo con comportamiento, como una persona (identificada por un rol), un sistema informatizado u organización, y que realiza algún tipo de interacción con el sistema.. Se representa mediante una figura humana dibujada con palotes. Esta representación sirve tanto para actores que son personas como para otro tipo de actores. (Modelando con UML, pag.37)



Actor

3.3.1.2.2 Caso de Uso

Un caso de uso es una descripción de la secuencia de interacciones que se producen entre un actor y el sistema, cuando el actor usa el sistema para llevar a cabo una tarea específica. Expresa una unidad coherente de funcionalidad, y se representa en el Diagrama de Casos de Uso mediante una elipse con el nombre del caso de uso en su interior. El nombre del caso de uso debe reflejar la tarea específica que el actor desea llevar a cabo usando el sistema. (Modelando con UML, pag.35)



Caso de Uso

3.3.1.3 Relaciones en los diagramas de casos de Uso

Un diagrama de casos de uso muestra las relaciones entre los actores y los casos de uso dentro de un sistema y son:

Asociación: representa la participación la participación de un actor en un caso de uso. Siempre parte desde el actor al caso de uso.

Generalización: representa la relación entre dos objetos del mismo tipo en el cual uno de ellos se comporta igual que otro pero que además contiene características adicionales que lo diferencian. Esto ocurre entre dos a mas actores o dos a mas casos de uso.

Include: este tipo de relación solamente esta entre casos de uso; significa que el caso de uso base incorpora explícitamente el comportamiento de otro caso de uso. El caso de uso base siempre utiliza el caso de uso incluido.

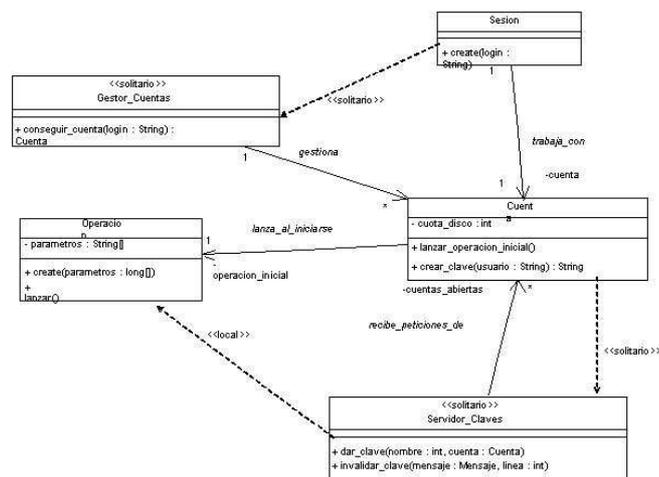
Extend: la elación extend esta dada entre dos casos de uso. Indica que una instancia de caso de uso B puede ser extendida por el comportamiento específico de un caso de uso A. El caso de uso A se ejecutará cuando al ser ejecutado B se den condiciones para que se active. (Modelando con UML, pag.39)

3.3.2 MODELO ESTÁTICO

3.3.2.1 Diagrama de Clases

La clase es un conjunto de cosas que tienen los mismos atributos y comportamientos. El conjunto de estas clases mas sus relaciones forma un diagrama de clases.

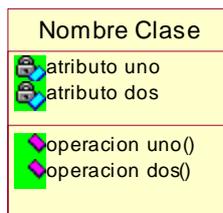
Un diagrama de clases es básicamente un modelo de Entidad/Relación evolucionado, es muy común en el modelado de sistemas orientados a objetos y cubren una vista estática del sistema.



Un diagrama de clases esta compuesto por los siguientes elementos:

3.3.2.1.1 La Clase

Una clase describe un conjunto de objetos con característica y comportamiento idénticos. Corresponden al dominio del problema que deseamos resolver; la clase contiene un nombre único, atributos que la caracterizan y operaciones que son formas como el objeto interactua con su entorno.



3.3.2.1.2 Relaciones

Existen tres relaciones diferentes entre clases, Dependencias, Generalización y Asociación. En las relaciones se habla de una clase destino y de una clase origen. La origen es desde la que se realiza la acción de relacionar. Es decir desde la que parte la flecha, la destino es la que recibe la flecha. Las relaciones se pueden modificar con estereotipos o con restricciones.

Relación de Dependencia: Es una relación de uso, es decir una clase usa a otra, que la necesita para su cometido. Se representa con una flecha discontinua va desde la clase utilizadora a la clase utilizada. Con la dependencia mostramos que un cambio en la clase utilizada puede afectar al funcionamiento de la clase utilizadora, pero no al contrario.



Relación de Generalización: Generalización es la herencia, donde tenemos una o varias clases padre o superclase o madre, y una clase hija o subclase que hereda el comportamiento y atributos de la clase padre y que a su vez significa que el hijo puede sustituir al padre; porque contiene todos sus atributos y operaciones.

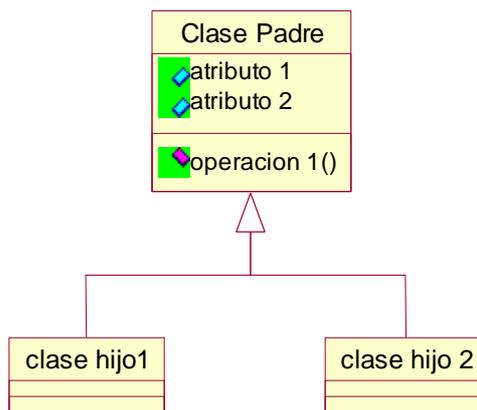


Figura 3.3.2.2

En la figura podemos ver que las clase hijo heredan atributos y operaciones de la clase padre.

Relación de Asociación: Es una relación estructural que describe un conjunto de enlaces o conexiones entre dos o mas clases, permitiendo asociar objetos de las clases que colaboran entre si para llevar a cabo un comportamiento deseado. (Modelando con UML, pag.72)



3.3.2.2 Diagrama de Objetos

Forma parte de la vista estática del sistema. En este diagrama se modelan las instancias de las clases del diagrama de clases. Muestra a los objetos y sus relaciones, pero en un momento concreto del sistema.

Un objeto se define como una instancia de una clase. Estas ocurrencias se muestran en el diagrama de objetos. (<http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/multiple-html/c385.html>).

3.3.3 MODELO DINÁMICO

En el modelo dinámico se utilizan los diagramas de interacción que representan la forma en que grupos de objetos se comunican entre si mediante el envío de mensajes, para realizar un determinado comportamiento.

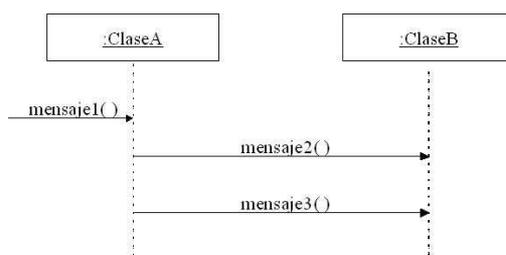
Los Diagramas de Interacción constituyen uno de los artefactos más importantes que se generan en el análisis y diseño orientado a objetos; ya que en ellos se muestra como interaccionan los objetos definiendo quienes cumplen las responsabilidades asignadas.

Hay dos tipos de diagrama de interacción, ambos basados en la misma información, pero cada uno enfatizando un aspecto particular: Diagramas de Secuencia y Diagramas de Colaboración. (Modelando con UML, pag.123)

3.3.3.1 Diagrama de Secuencia

Un diagrama de Secuencia muestra una interacción ordenada de un conjunto de objetos; poniendo énfasis en el orden del envío de mensajes entre objetos..

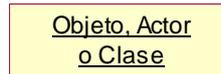
La creación de los Diagramas de Secuencia forma parte de la investigación para conocer el sistema, mediante los diagramas de secuencia se puede dar detalle a los Casos de Uso porque durante la operación del sistema los actores generan eventos que solicita alguna operación.



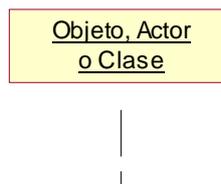
3.3.3.1.2 Elementos

Para la representación de los diagramas de secuencia se hace uso de varios elementos:

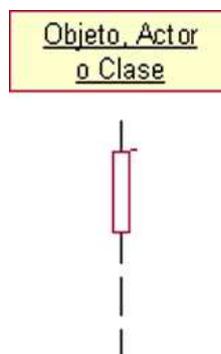
Objeto: representa una clase, actor u objeto que interactúa con el sistema.



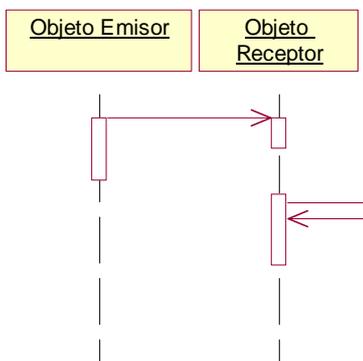
Línea de Vida: indica la vida de un objeto durante la interacción, es una línea punteada debajo del objeto.



Foco de Control: muestra el periodo en el cual el objeto se encuentra desarrollando alguna operación, bien sea por si mismo o por medio de algún procedimiento subordinado, es un rectángulo delgado ubicado en la línea de vida.



Mensaje: se representa mediante una flecha, son los mensajes enviados desde un emisor a un receptor el último ejecutará la acción especificada. Un objeto puede invocarse a si mismo. (Modelando con UML, pag.124)

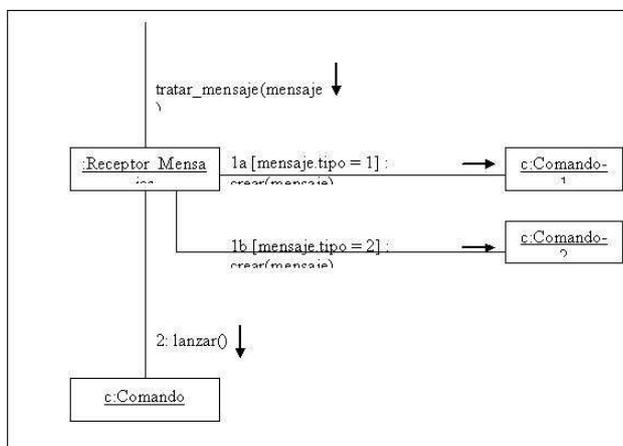


3.3.3.2 Diagrama de Colaboración

Los diagramas de Colaboración permiten mostrar mejor como se vinculan los objetos, a cambio de hacer más difícil observar el orden de ejecución, pues enumeran los mensajes en lugar de mostrar al tiempo como una dimensión.

Muestran la colaboración entre objetos para realizar una tarea mediante el uso de mensajes enviados entre ellos.

La secuencia de los mensajes y los flujos de ejecución concurrentes deben determinarse explícitamente mediante números de secuencia.



Sus elementos son:

Objetos: representa al objeto, actor, o clase que interactúa con el sistema.

Enlace: es la conexión entre dos objetos, es una instancia de la asociación del diagrama de clases.

Flujo de Mensajes: Expresa el envío de un mensaje en una flecha dirigida colocada sobre el enlace.

Numeración de Mensajes: los mensajes ocurren una determinada secuencia y esto se representa mediante un número. (Modelando con UML, pag.139)

3.3.4 MODELO FUNCIONAL

3.3.4.1 Diagrama de Estados

Un Diagrama de Estados muestra la secuencia de estados por los que pasa bien un caso de uso, bien un objeto a lo largo de su vida, o bien todo el sistema. En él se indican qué eventos hacen que se pase de un estado a otro y cuáles son las respuestas y acciones que genera.

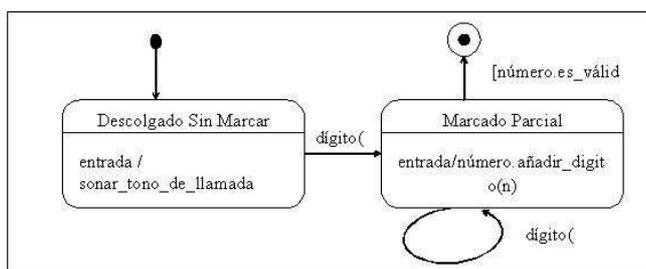
En cuanto a la representación, un diagrama de estados es un grafo cuyos nodos son estados y cuyos arcos dirigidos son transiciones etiquetadas con los nombres de los eventos.

3.3.4.1.1 Elementos

Estado: Un estado se representa como una caja redondeada con el nombre del estado en su interior. Una transición se representa como una flecha desde el estado origen al estado destino. La caja de un estado puede tener 1 o 2 compartimentos. En el primer compartimiento aparece el nombre del estado. El

segundo compartimiento es opcional, y en él pueden aparecer acciones de entrada, de salida y acciones internas.

Acción: es un evento que puede causar una transición del objeto de un estado a otro. Una acción de entrada aparece en la forma entrada/acción asociada donde acción, asociada es el nombre de la acción que se realiza al entrar en ese estado. Cada vez que se entra al estado por medio de una transición la acción de entrada se ejecuta. Una acción de salida aparece en la forma salida/acción asociada. Cada vez que se sale del estado por una transición de salida la acción de salida se ejecuta. Una acción interna es una acción que se ejecuta cuando se recibe un determinado evento en ese estado, pero que no causa una transición a otro estado. Se indica en la forma nombre de evento/acción asociada. (Modelando con UML, pag.157).



3.3.4.2 Diagrama de Actividades

Los diagramas de Actividades muestran el flujo de actividades en cualquier vista de la arquitectura de una aplicación, desde un punto de partida se que complementa con el diagrama de iteración.

Muestra la realización de operaciones para conseguir un objetivo; además presenta una visión simplificada de lo que ocurre en un proceso, mostrando los pasos que se realizan.

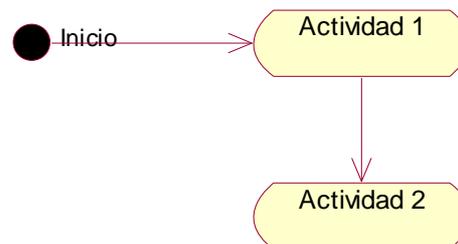
Los Diagramas de Actividad, pueden ser utilizados para modelar y dar detalle a casos de uso en uno o varios diagramas.

3.3.4.2.1 Elementos

Actividades o acciones: Se representa con un rectángulo con bordes redondeados. Son las actividades que realiza el sistema.

Transiciones: cuando se completa la ejecución de una actividad, el flujo de control pasa a la siguiente actividad dando lugar a una transición. Se representa con una flecha.

Decisiones: en estos diagramas se puede incluir caminos alternativos según cumpla alguna condición. (Modelando con UML, pag.187)



3.3.5 MODELO ARQUITECTÓNICO

La estructura de un conjunto de páginas Web es muy importante, ya que una buena estructura permitirá al lector visualizar todos los contenidos de una manera fácil y clara.

3.3.5.1 Diagrama Arquitectónico

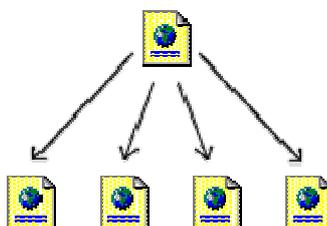
El diseño arquitectónico para Aplicaciones Web se centra en la definición de la estructura hipertexto para la aplicación, y en la utilización de diseño y plantillas en su estructura.

La estructura de una Aplicación Web tiene que ir unida a las metas establecidas, para el diseño arquitectónico de una Aplicación Web se puede elegir entre 4 tipos diferentes:

3.3.5.1.1 Estructura Jerárquica

La estructura jerárquica, es la típica estructura de árbol, en el que la raíz es la hoja de bienvenida, esta hoja se puede también sustituir por la hoja de contenido, en la que se exponen las diferentes secciones que contendrá nuestro sitio. La selección de una sección nos conduce asimismo a una lista de subtemas que pueden o no dividirse.

Este tipo de organización permite al lector conocer en qué lugar de la estructura se encuentra, además de saber que, con forme se adentra en la estructura obtiene información más específica y que la información más general se encuentra en los niveles superiores.

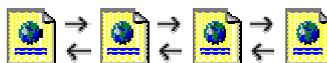


3.3.5.1.2 Estructura Lineal

La estructura lineal es la más simple de todas, la manera de recorrerla es la misma que si estuviésemos leyendo un libro, de manera que estando en una página, podemos ir a la siguiente página o a la anterior.

Esta estructura es muy útil cuando queremos que el lector siga un camino fijo y guiado, además impedimos que se distraiga con enlaces a otras páginas. Por otra parte podemos causar a lector la sensación de estar encerrado si el camino es muy largo o poco interesante.

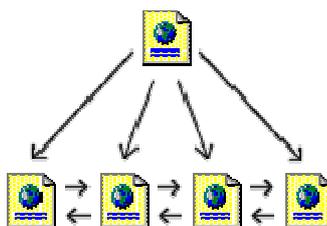
Este tipo de estructura sería válido para tutoriales de aprendizaje o tours de visita guiada.



3.3.5.1.3 Lineal con jerarquía

Este tipo de estructura es una mezcla de la dos anteriores, los temas y subtemas están organizados de una forma jerárquica, pero uno puede leer todo el contenido de una forma lineal si se desea.

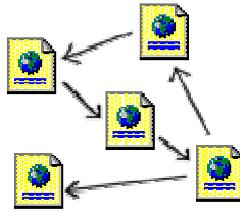
Esto permite tener el contenido organizado jerárquicamente y simultáneamente poder acceder a toda la información de una manera lineal como si estuviésemos leyendo un libro. Esta guía sigue básicamente este tipo de estructura.



3.3.5.1.4 Estructura en Red

La estructura de red es una organización en la que aparentemente no hay ningún orden establecido, las páginas pueden apuntarse unas a otras sin ningún orden aparente.

Este tipo de organización es la más libre, pero también es la más peligrosa ya que si no se informa al lector de en dónde se encuentra, puede perderse o puede no encontrar lo que anda buscando o no llegar a ver lo que le queremos mostrar. Por eso es muy recomendable asociar la estructura de las páginas con alguna estructura conocida, como por ejemplo la de una ciudad. (Pressman, pag.528)



3.3.5.2 Diagrama Navegacional

Unos de los apartados más importantes en el diseño de páginas Web son los elementos de navegación. Ya que son los que nos permiten movernos a través de las diferentes páginas que hemos construido.

El diagrama navegacional permite dar una vista sobre el diagrama de clases de OO-method para cada tipo de usuario que interactúa con el sistema.

Cuando ya se establece la arquitectura de la aplicación se deberá definir las rutas de navegación que permitan al usuario acceder al contenido de la aplicación. Para esto debe identificar la navegación para los diferentes usuarios que tenga el sistema y la mecánica de navegación.

La estructura de una unidad de navegación se compone de un conjunto de subestructuras de navegación o formas de navegación y que representa la ruta para que un usuario con ciertos perfiles logre una meta deseada.

(Pressman, pag.530)

CAPITULO IV

CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

La metodología OOHDM permitió el desarrollo de este proyecto; aplicando esta metodología se obtuvo resultados óptimos en la construcción del sitio web. Cada fase de la metodología OOHDM otorgó una idea clara de cómo crear una aplicación precisa a los requerimientos del usuario.

El uso de herramientas de desarrollo tales como HTML, PHP y MySQL permitieron obtener un resultado satisfactorio al momento de crear la interfaz y cada módulo del sistema.

Este proyecto facilita la visualización de calificaciones de los estudiantes del Conservatorio Superior Nacional de Música además muestra reportes de eventos y horarios de maestros.

El diseño de la interfaz permite la interacción de los usuarios en forma fácil, rápida y precisa.

La futura implementación de este proyecto resolverá problemas de flujo de información del Conservatorio.

4.2 RECOMENDACIONES

Se recomienda que a futuro se considere la inclusión de un módulo para matriculación.

Es importante que se designe un Web master o administrador responsable para que la información que presenta el sitio sea siempre actual y precisa.

Es conveniente hacer el mantenimiento del sitio por lo menos una vez al año innovando el diseño de la interfaz para que sea atractiva al usuario.

4.3 REFERENCIAS BIBLIOGRÁFICAS

LIBROS

Pressman Rogger (2002). "Ingeniería de Software", Madrid (España), editorial Concepción Fernández: 5ta edición.

César Liza Avila. "Modelando con UML, Principios y Aplicaciones", Trujillo-Perú, editorial RJ S.R.Ltda. 1ra edición, agosto 2001.

INTERNET

Cuenca L (2005). Backup MySQL con mysqldump.

<http://www.desarrolloweb.com/articulos/1202.php?manual=34>

González P (2003), Creación de gráficas en PHP con JpGraph

<http://www.desarrolloweb.com/articulos/1987.php>

wikipedia (2007), Desarrollo en Espiral

http://es.wikipedia.org/wiki/Desarrollo_en_espiral

El ciclo de Vida del Software Fases, Tipos de modelos....

<http://zoosharp.wordpress.com/2006/12/08/modelo-de-ciclo-de-vida-del-software/>

Moreno (2002) Ergonomía y Usabilidad, Internet transparente

<http://www.uco.es/~el1momua/ergonomia/internetparalaindustria.htm>

El Ciclo de Vida del Software

<http://html.rincondelvago.com/el-ciclo-de-vida-del-software.html>

Lamarca Lapuente(27/12/2006) Modelo OOHDM

<http://www.hipertexto.info/documentos/oohdm.htm>

Wikipedia (3 feb 2007) OOHDM

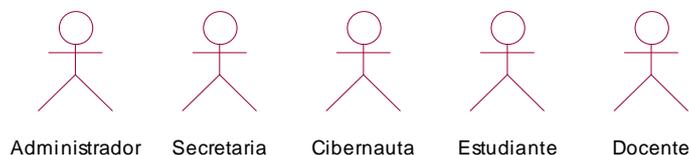
<http://es.wikipedia.org/wiki/OOHDM>

ANEXOS

MANUAL TÉCNICO

ESPECIFICACIÓN DE REQUERIMIENTOS

IDENTIFICACIÓN DE ACTORES



Administrador: Usuario que se encargará de la administración del sitio.

- Administrar Estudiantes
- Administrar Materias
- Administrar Maestros
- Administrar Carga Académica
- Administrar Eventos

Secretaria: Realiza actividades heredadas del administrador.

Estudiante: Usuario que ingresa al sitio en busca de información personal.

- Visualizar Datos
- Consultar Calificaciones

Docente: Usuario que ingresa al sitio en busca de información personal.

- Visualizar Datos
- Consultar Carga Académica y Horarios

Cibernauta: Usuario que ingresa al sitio en busca de información general.

- Visualizar Información Conservatorio

DIAGRAMAS DE CASOS DE USO

DIAGRAMA DE CASOS DE USO TOTAL

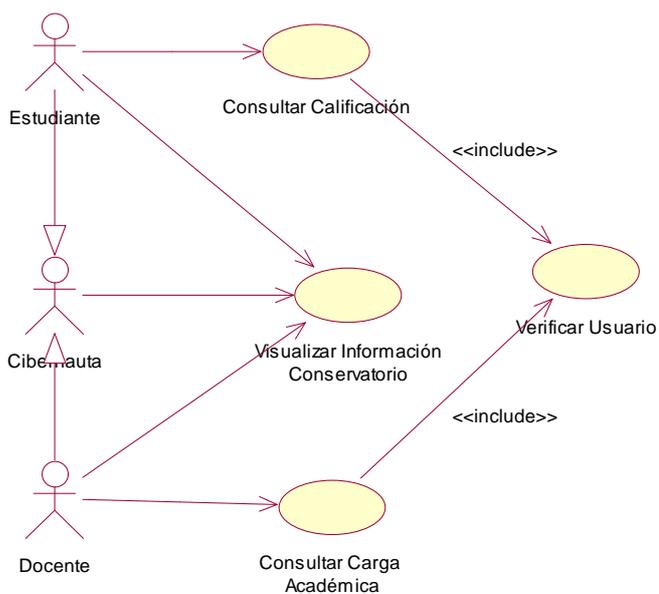
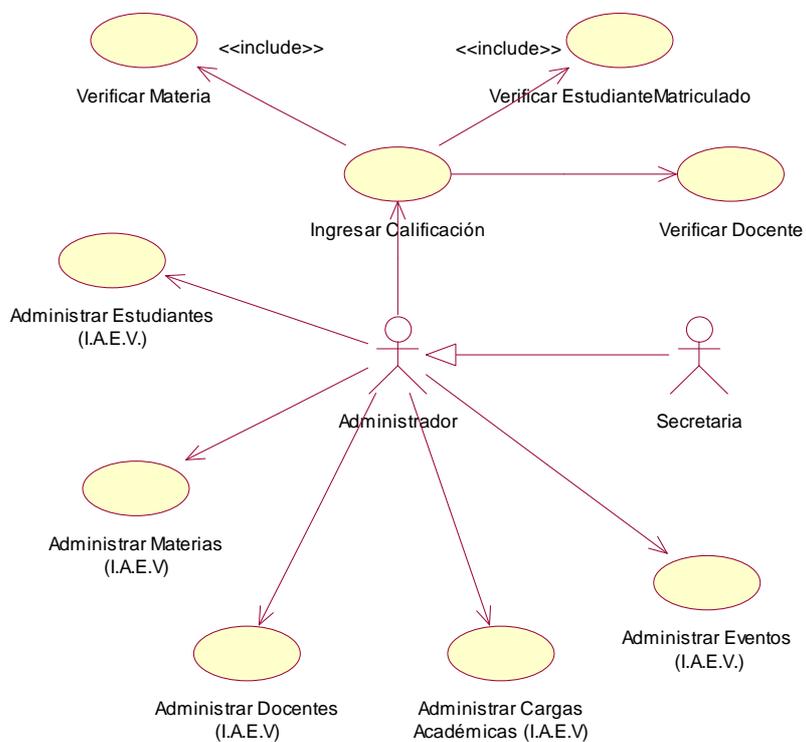
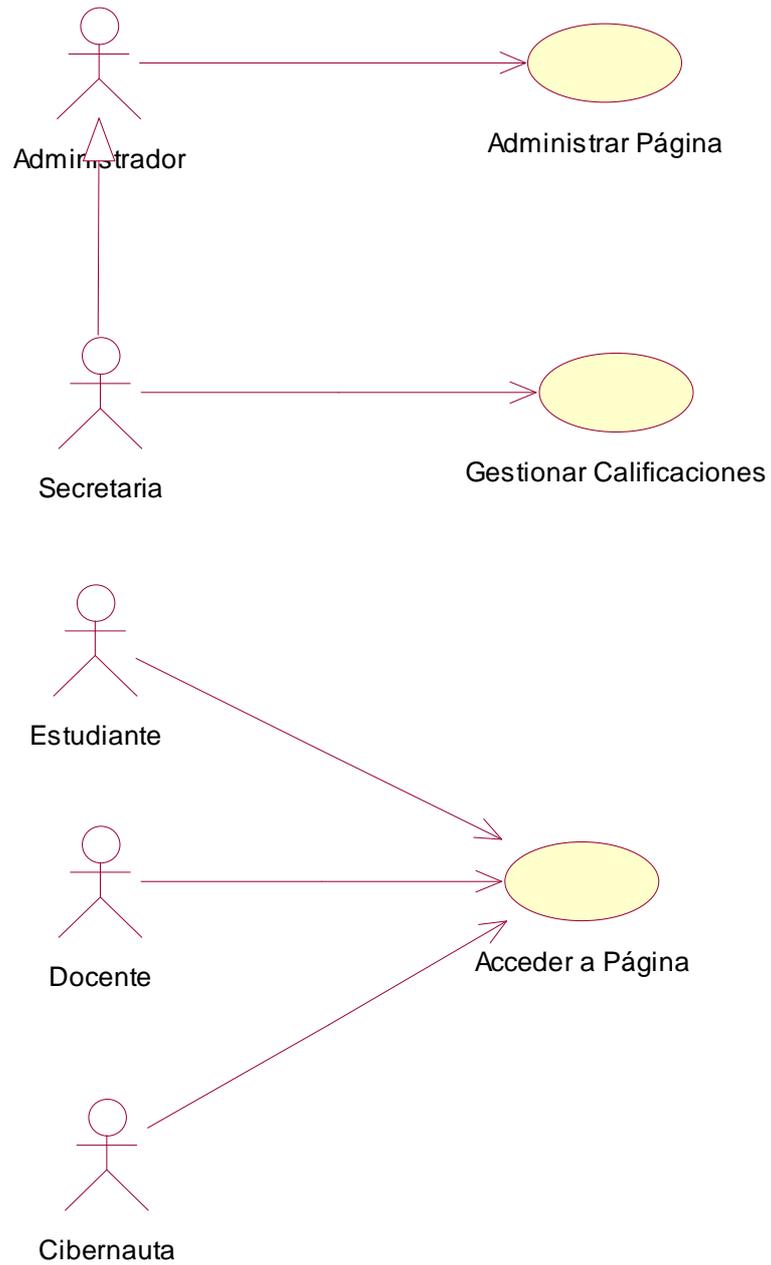
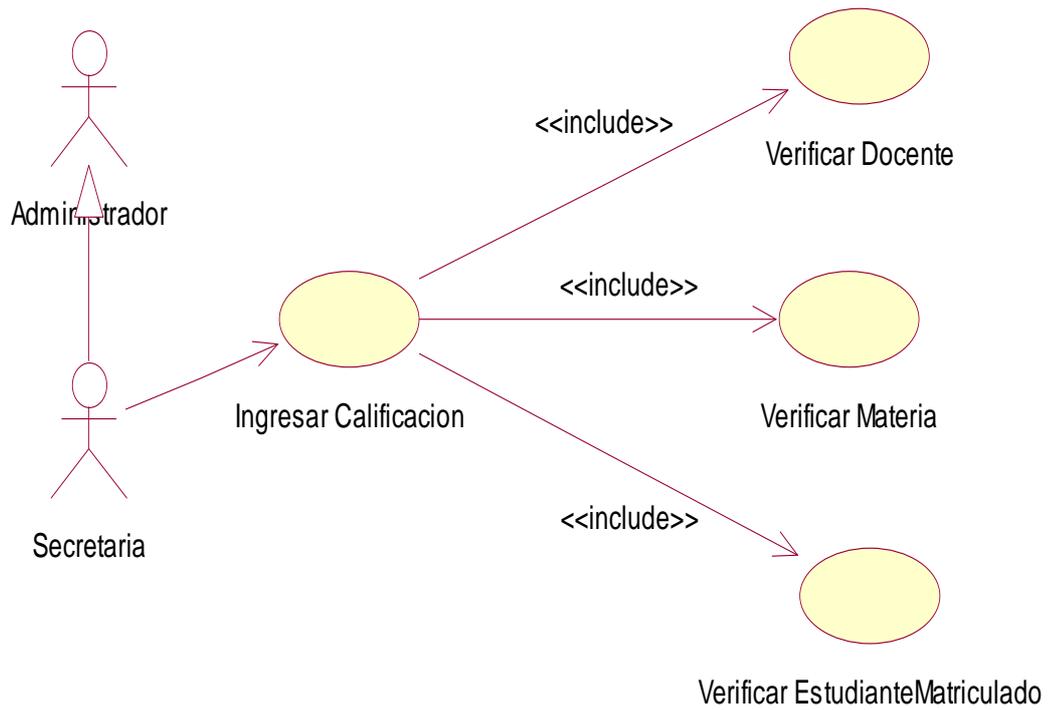


DIAGRAMA DE CASOS DE USO MODULAR

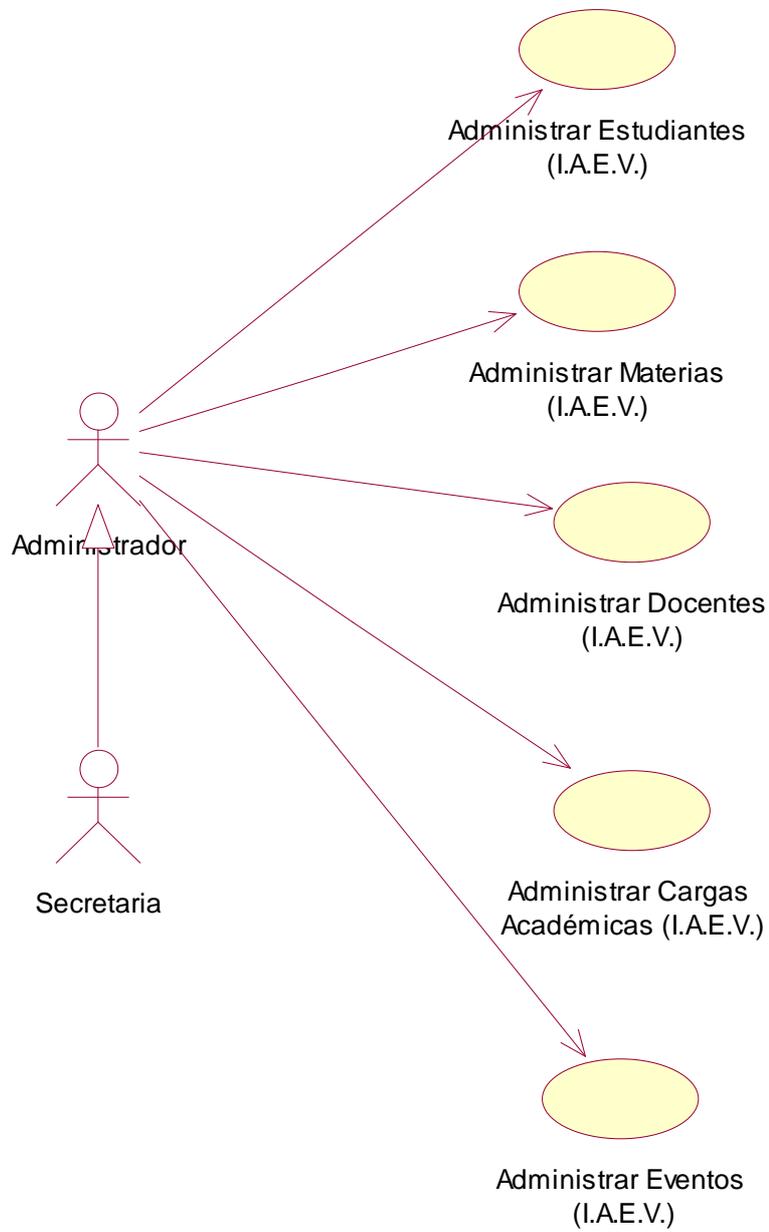


DIAGRAMAS DE CASOS DE USO POR ACTIVIDAD

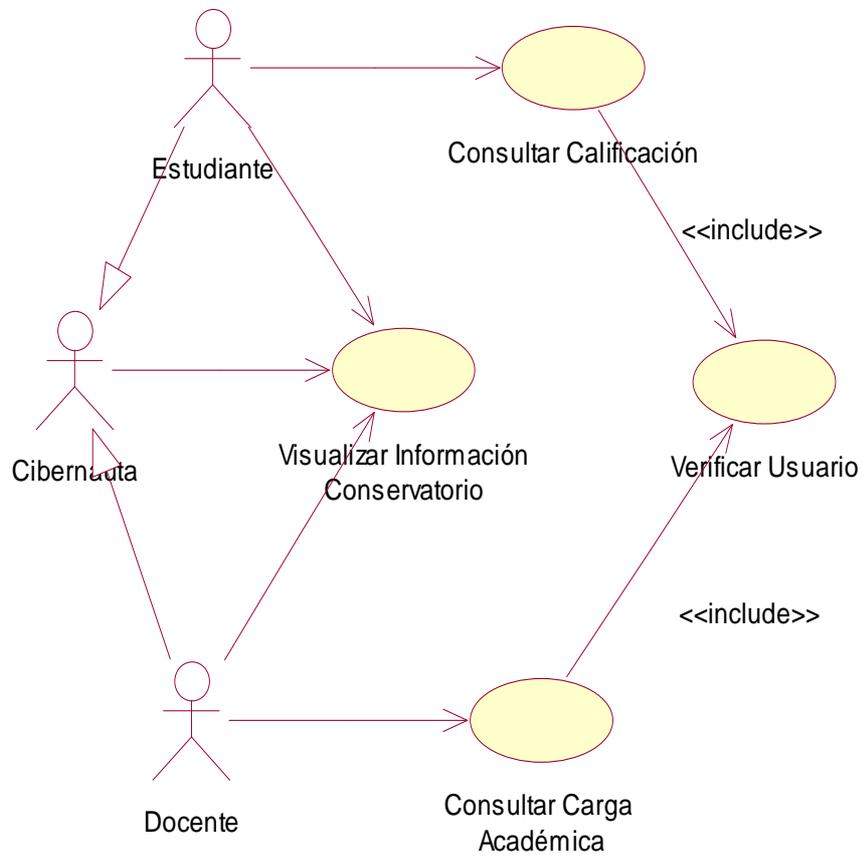
GESTIONAR CALIFICACIONES



ADMINISTRAR PÁGINA



ACCEDER A PÁGINA



DICCIONARIO DE CASOS DE USO

Casos de Uso	Descripción
Ingresar Calificación	Permite ingresar calificaciones de un estudiante.
Verificar Docente	Valida si un maestro se encuentra Registrado o no.
Verificar Materia	Valida si una materia está registrada o no.
Verificar Estudiante Matriculado	Verifica si un estudiante está registrado y matriculado en un período determinado.
Administrar Estudiante	Permite ingresar un nuevo estudiante, modificar datos de un antiguo estudiante, eliminar registros de estudiante y asignar matrícula y materias de un estudiante en un período determinado
Administrar Materia	Permite ingresar una nueva materia, modificar datos de materias existentes y eliminar registros de materias.
Administrar Docente	Permite ingresar un nuevo docente, modificar datos de docentes

	antiguos y eliminar registros de docentes.
Administrar Carga Académica	Permite asignar materias y horarios a un maestro.
Administrar Eventos	Permite crear nuevo evento, modificar y eliminar registros de eventos.
Verificar Usuario	Valida el código del usuario y su clave de acceso si pertenece a algún registro o no.
Consultar Calificaciones	Genera un reporte de las calificaciones de un estudiante.
Consultar Carga Académica	Genera un reporte de materias y horarios asignados a un maestro.
Visualizar Información Conservatorio	Permite a un usuario visualizar la información general del sitio.

ANÁLISIS Y DISEÑO

MODELO ESTÁTICO

DIAGRAMA DE CLASES

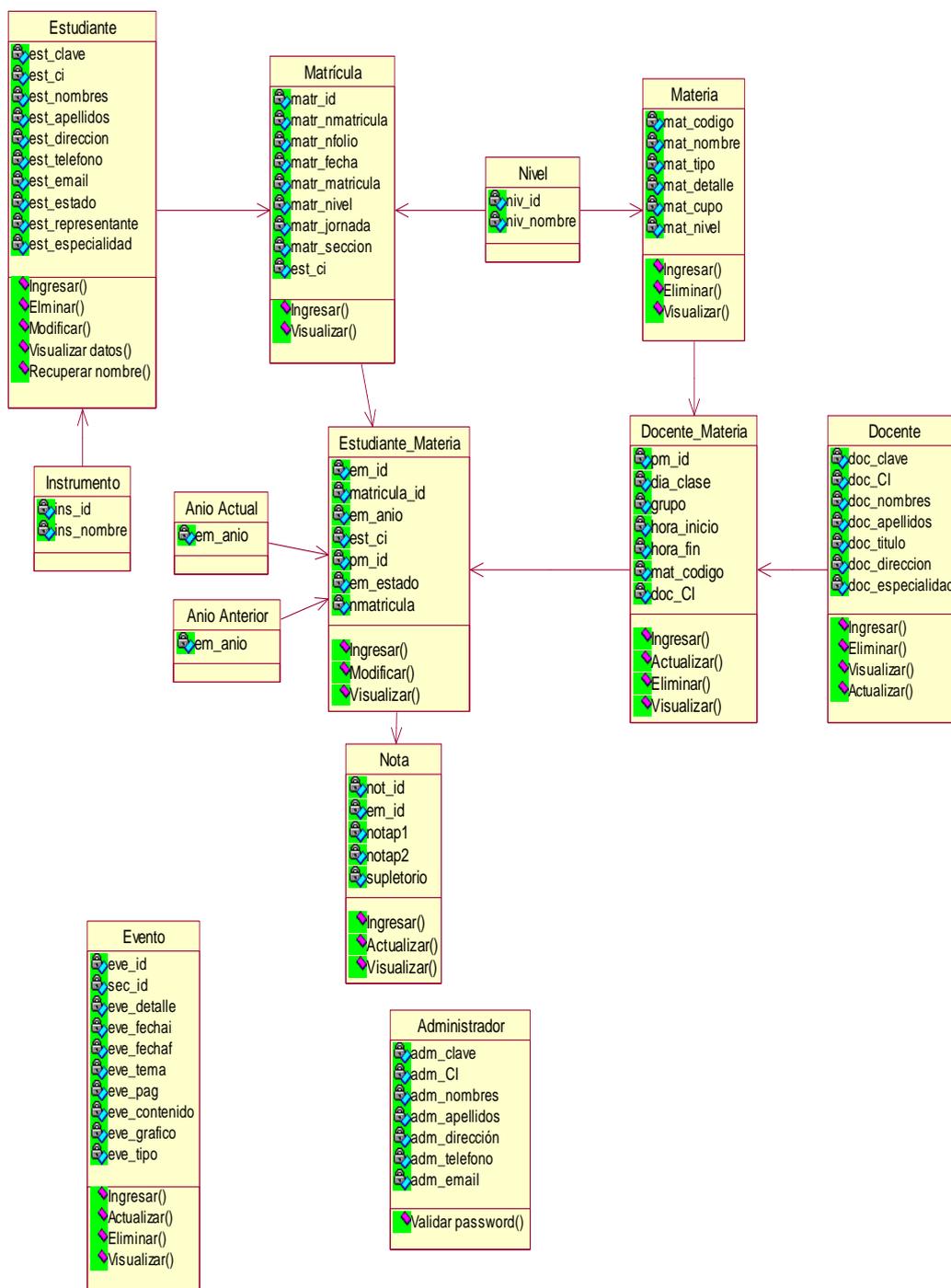
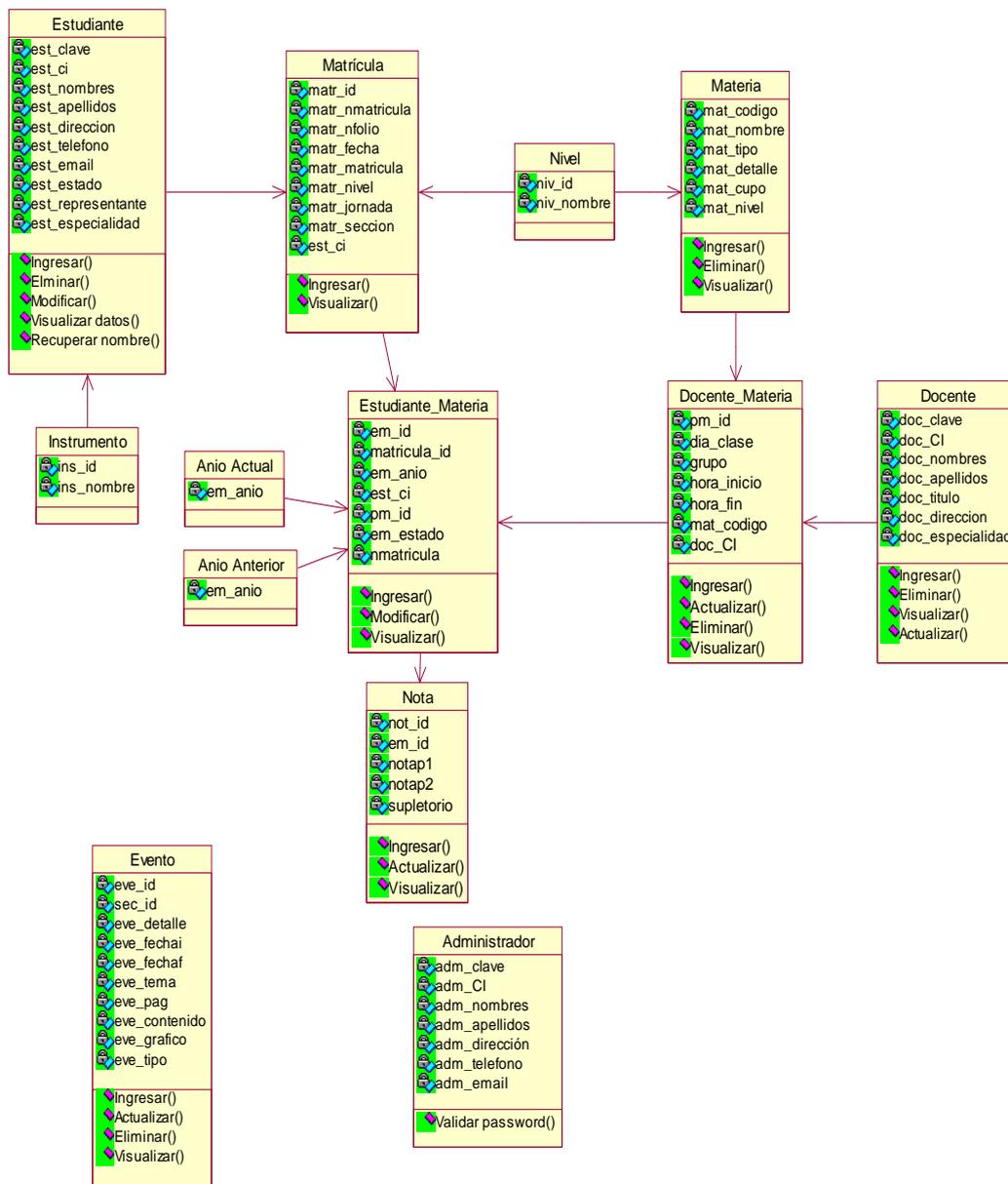


DIAGRAMA DE OBJETOS



DICCIONARIO DE CLASES

Clases	Descripción
Estudiante	Almacena el código del estudiante y sus datos personales.
Matrícula	Almacena el número de matrícula de un estudiante en un período determinado
Anio Actual, Anio Anterior	almacena los años de un período establecido.
Materia	Almacena el código de la materia y datos de la materia por cada nivel de estudios.
Docente	Almacena el código del maestro y sus datos personales.
Materia_Docente	Almacena el código de la materia código del maestro y horarios.
Materia_Estudiante	Almacena el código de materia y código de estudiante en un período.
Notas	Almacena el código de materia, código de estudiante y código de año y calificaciones del estudiante.
Nivel	Almacena los niveles de estudio o cursos.

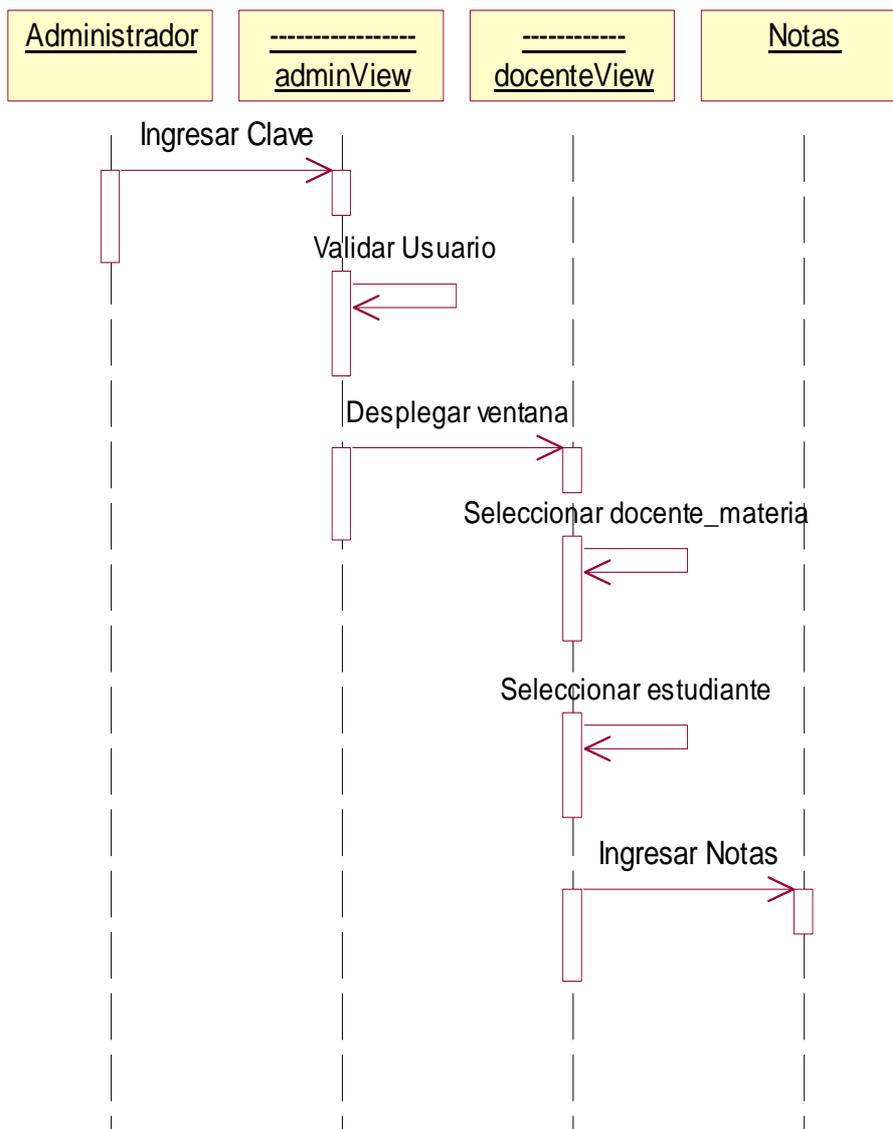
Instrumento	Almacena el código y nombre de instrumentos de especialidad del estudiante.
Eventos	Almacena las actividades a realizarse por la institución.
Administrador	Almacena los datos del administrador quien actualizará los datos del sitio.

MODELO DINÁMICO

DIAGRAMAS DE INTERACCIÓN

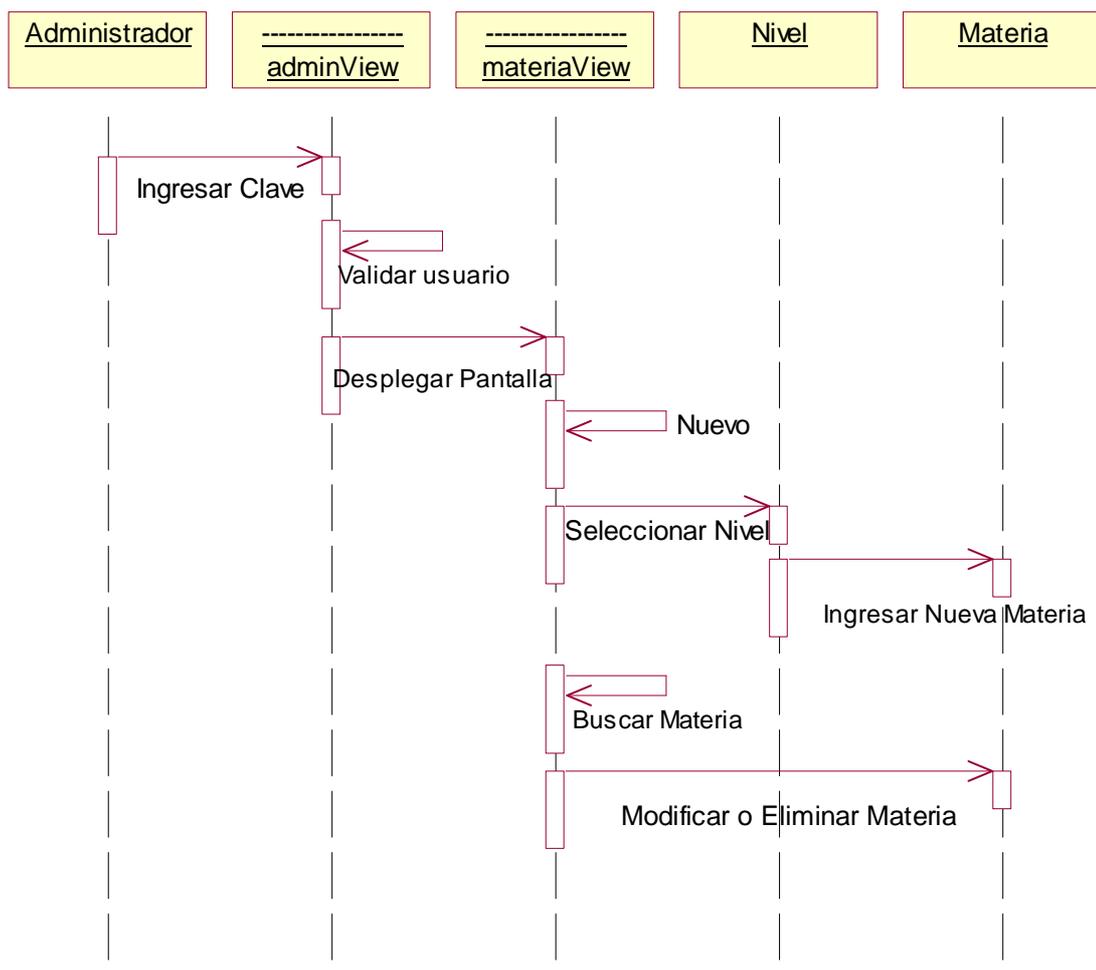
DIAGRAMA DE SECUENCIA

GESTIONAR CALIFICACIONES



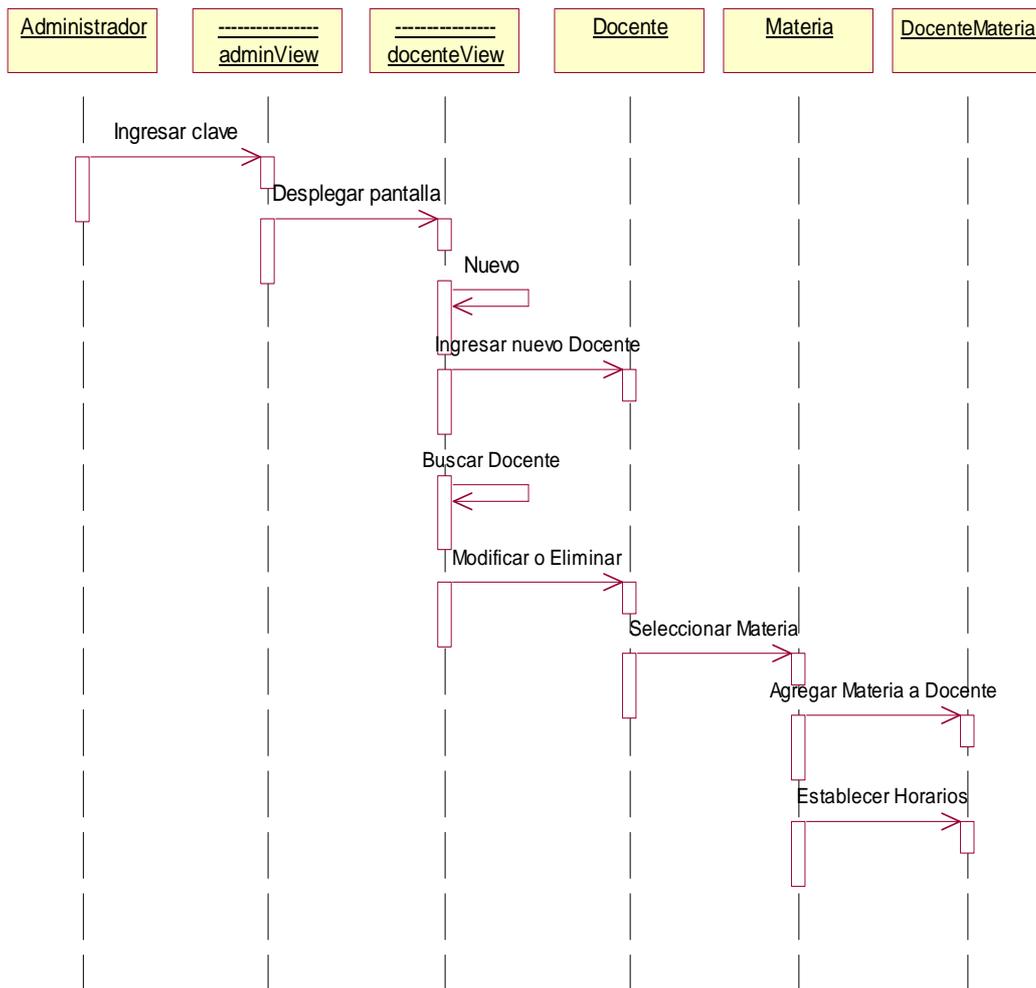
ADMINISTRAR PÁGINA

ADMINISTRAR MATERIAS



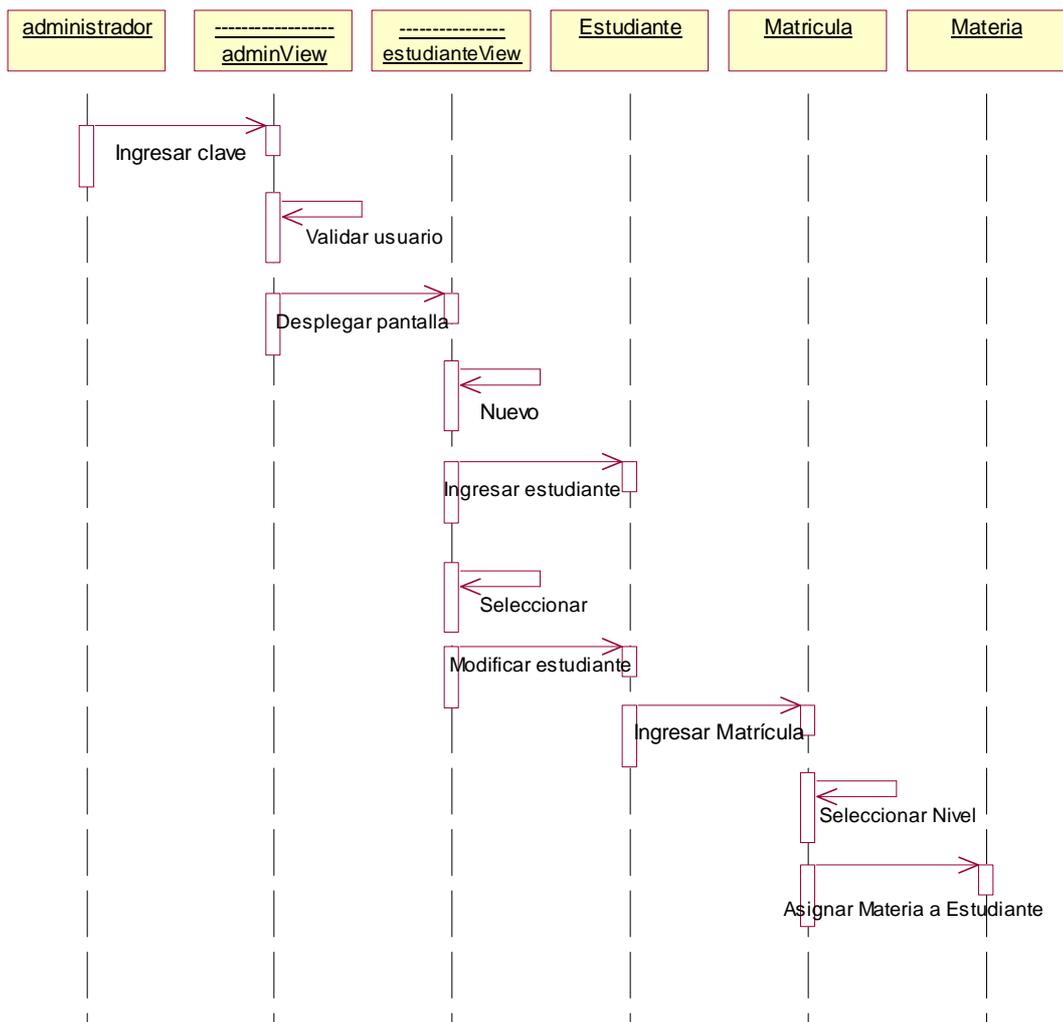
ADMINISTRAR PÁGINA

ADMINISTRAR DOCENTE



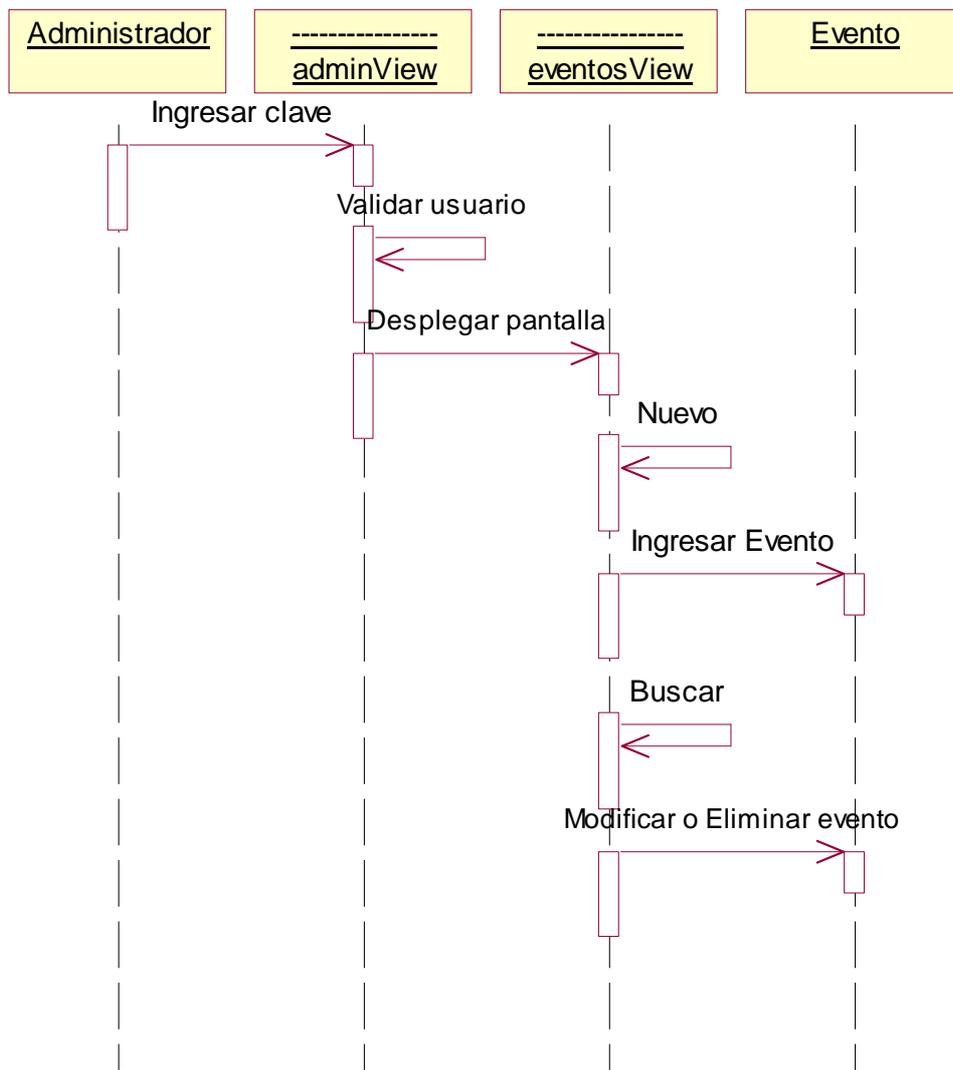
ADMINISTRAR PÁGINA

ADMINISTRAR ESTUDIANTE

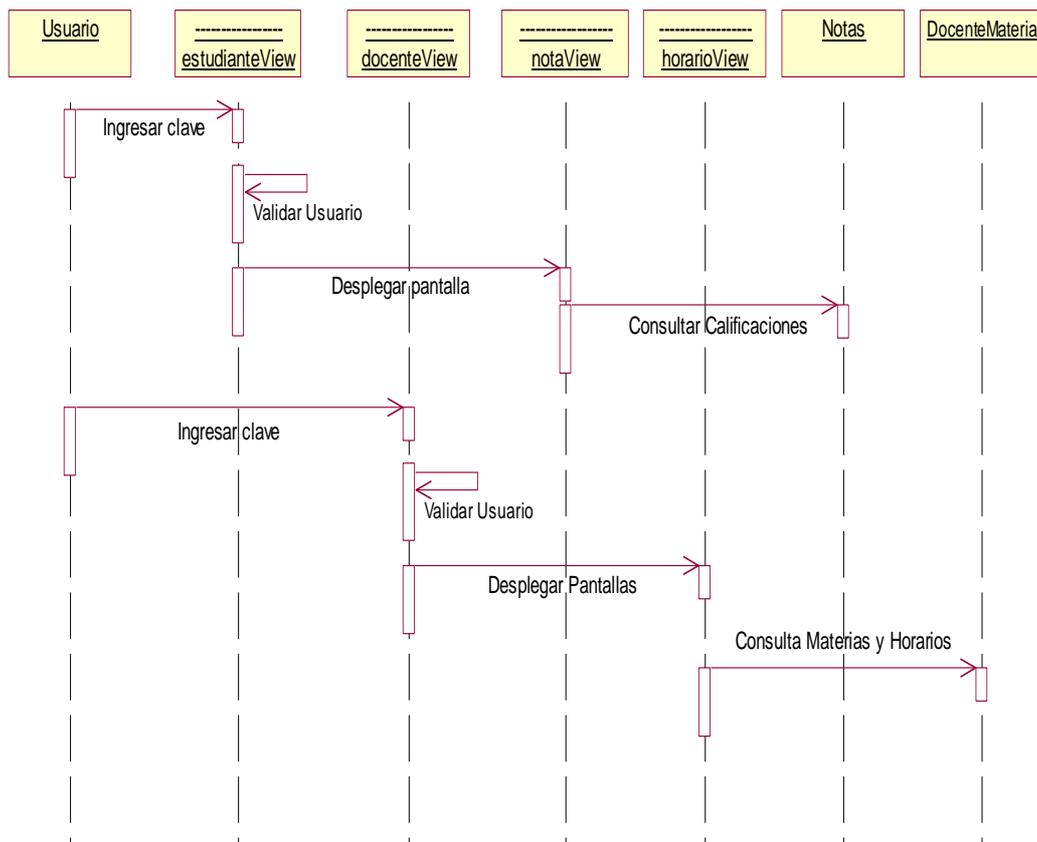


ADMINISTRAR PÁGINA

ADMINISTRAR EVENTO

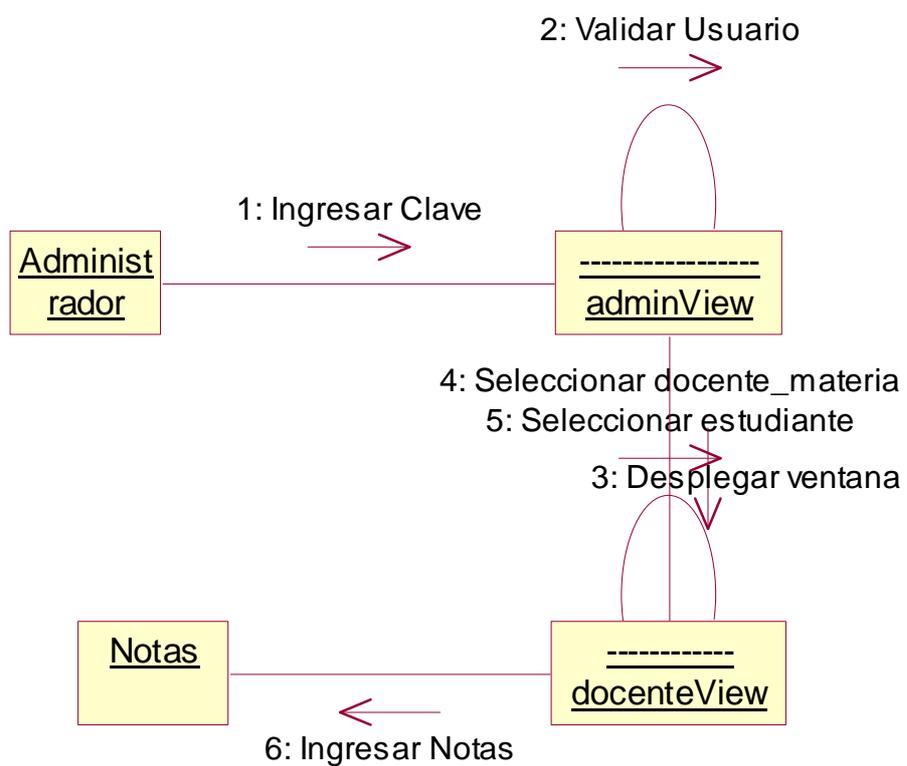


ACCEDER A PÁGINA



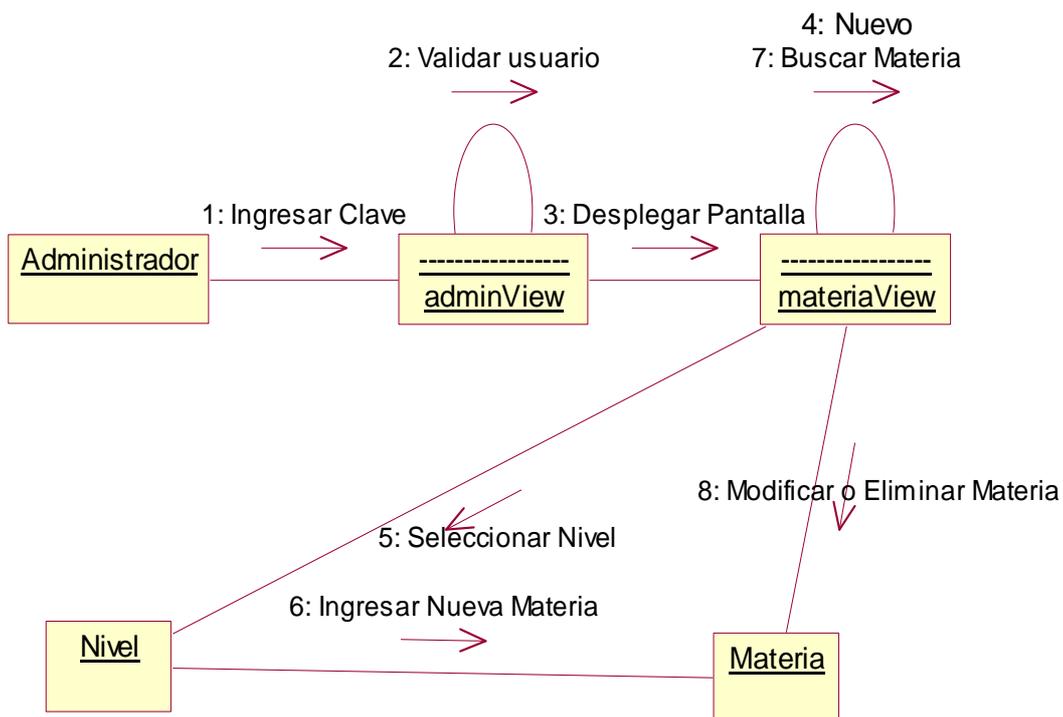
DIAGRAMAS DE COLABORACIÓN

GESTIONAR CALIFICACIONES



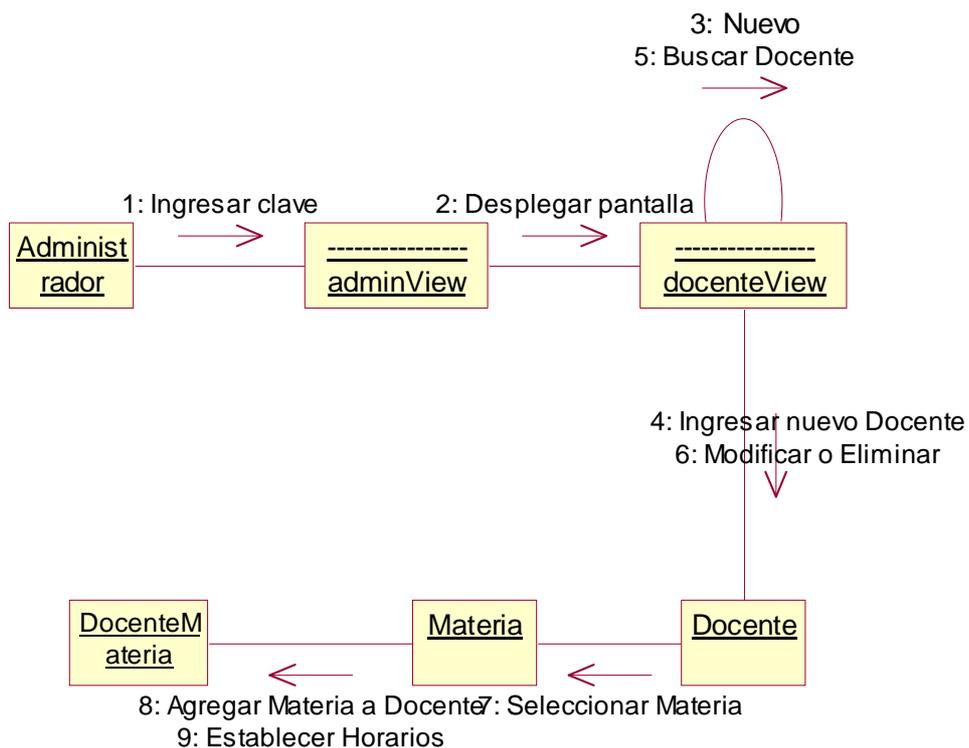
ADMINISTRAR PÁGINA

ADMINISTRAR MATERIA



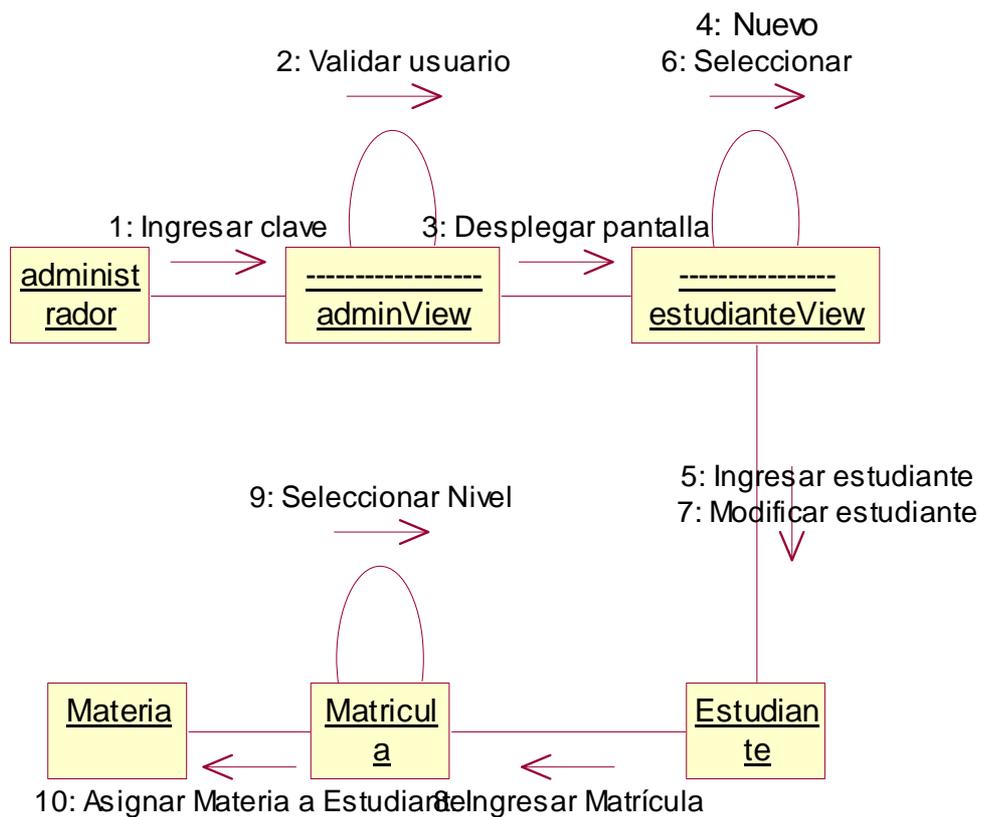
ADMINISTRAR PÁGINA

ADMINISTRAR DOCENTE



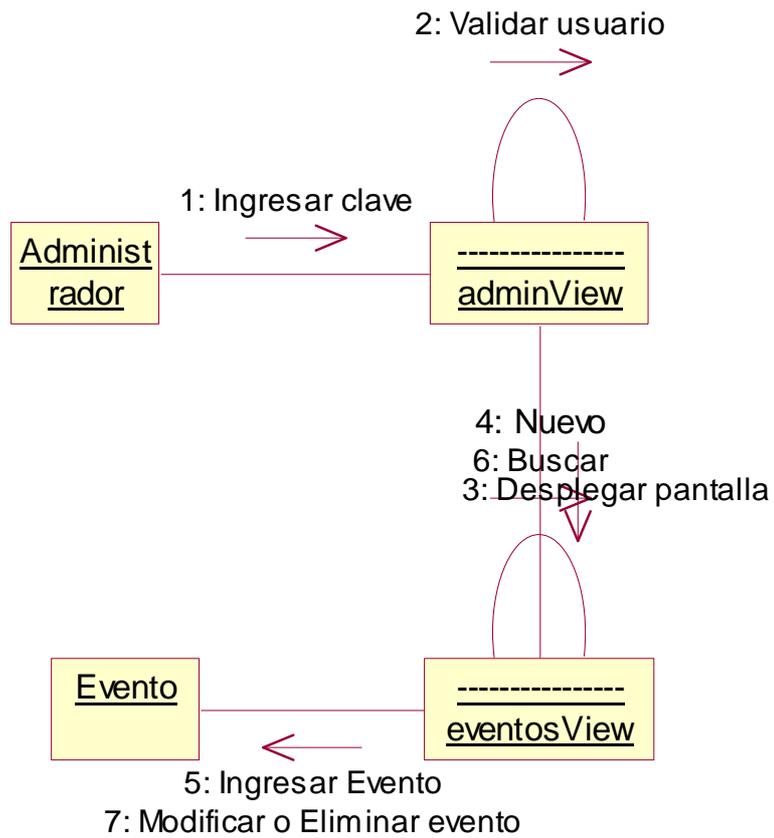
ADMINISTRAR PÁGINA

ADMINISTRAR ESTUDIANTE

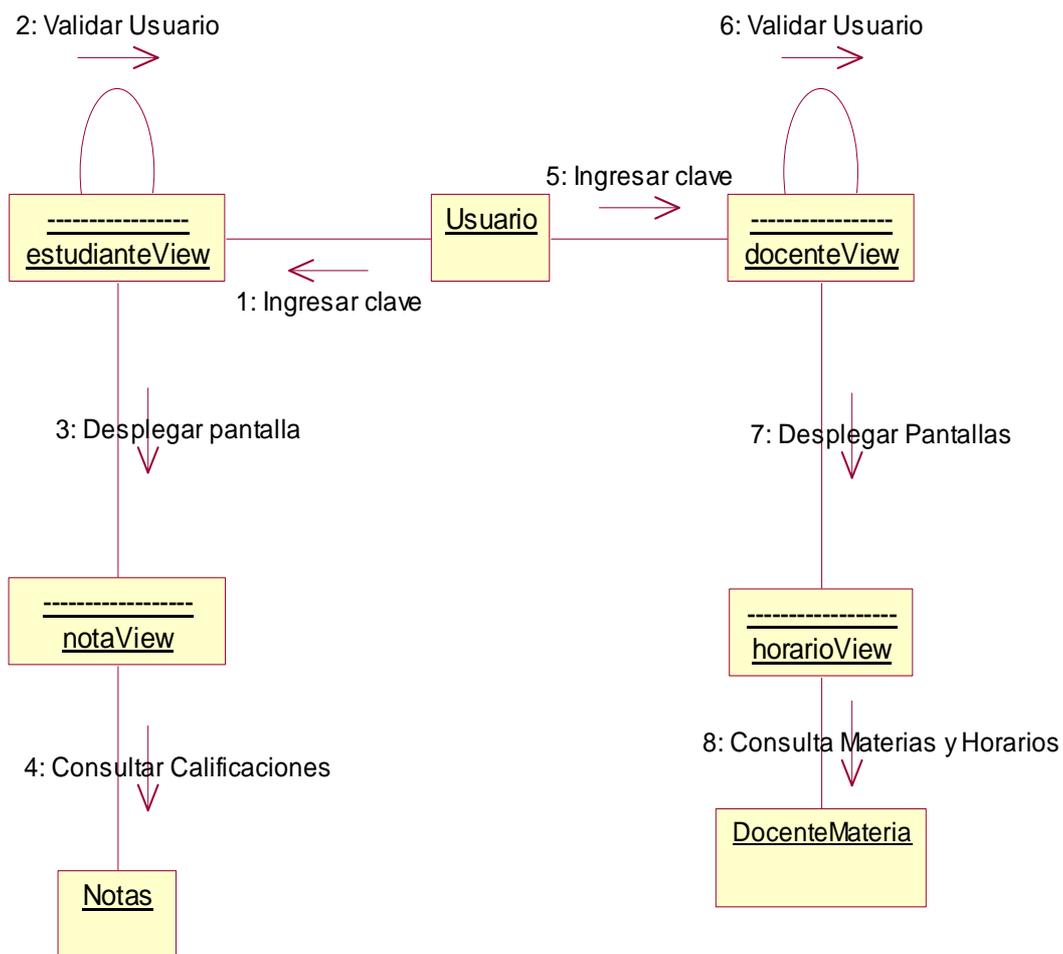


ADMINISTRAR PÁGINA

ADMINISTRAR EVENTO



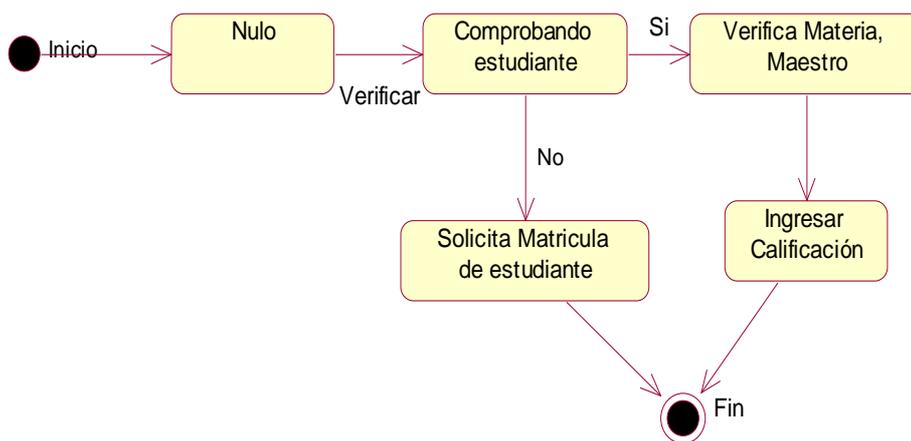
ACCEDER A PÁGINA



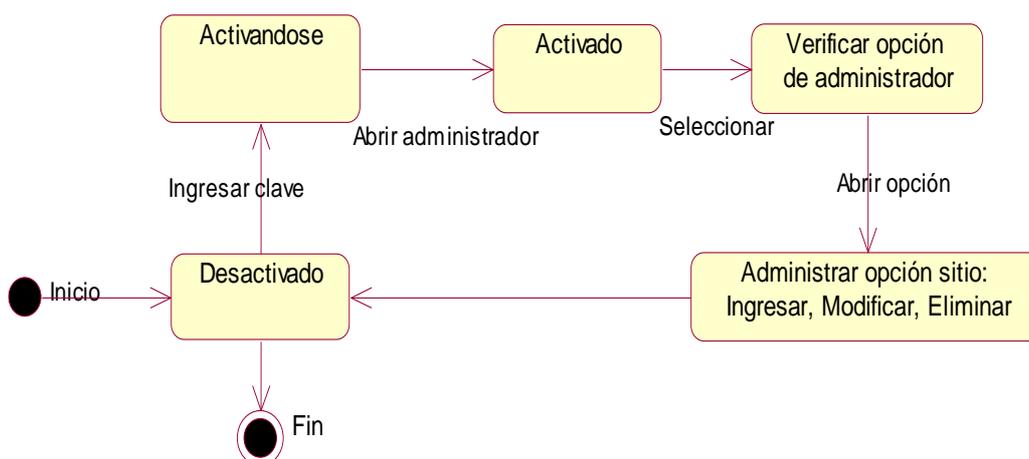
MODELO FUNCIONAL

DIAGRAMAS DE ESTADOS

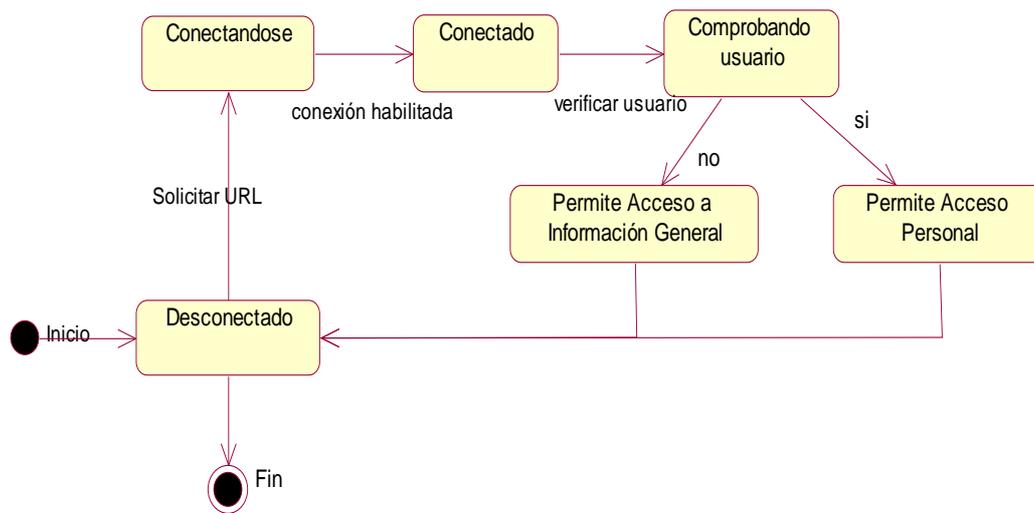
GESTIONAR CALIFICACIONES



ADMINISTRAR PÁGINA

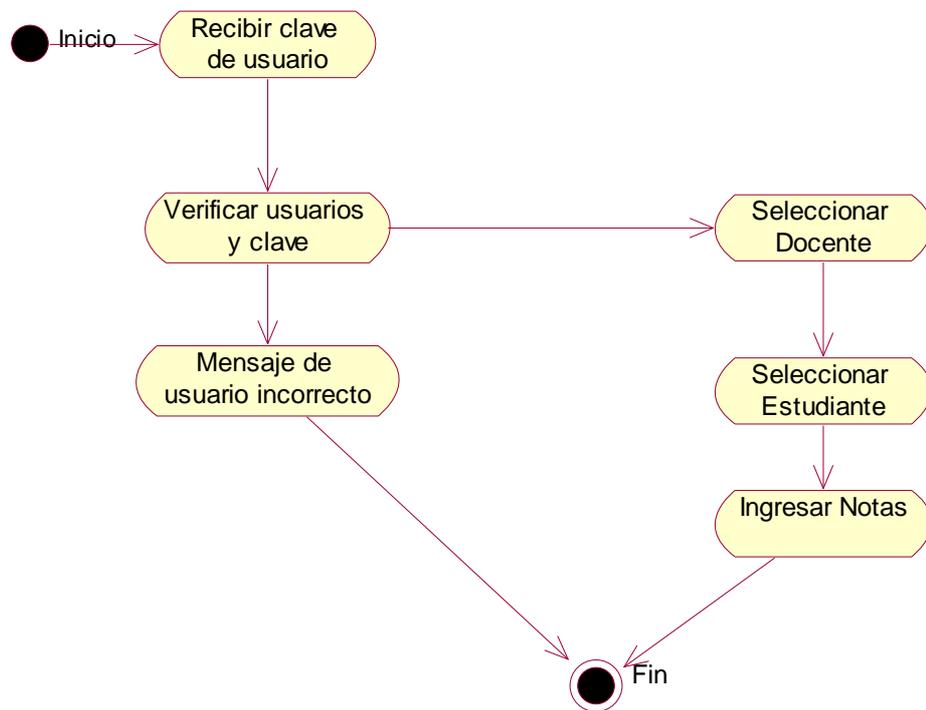


ACCEDER A PÁGINA

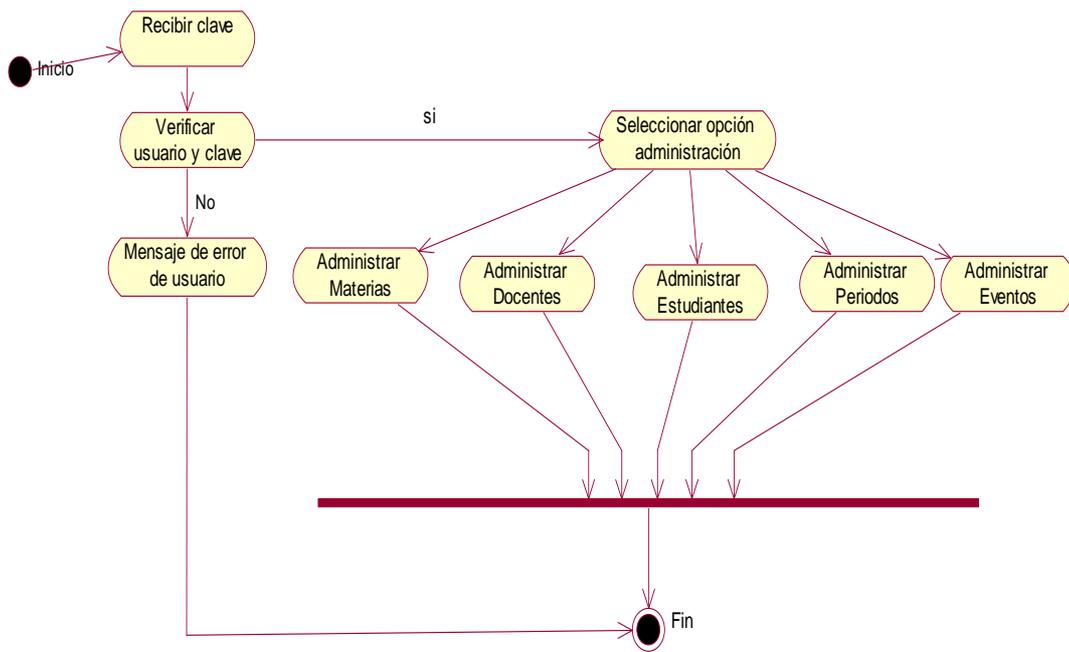


DIAGRAMAS DE ACTIVIDADES

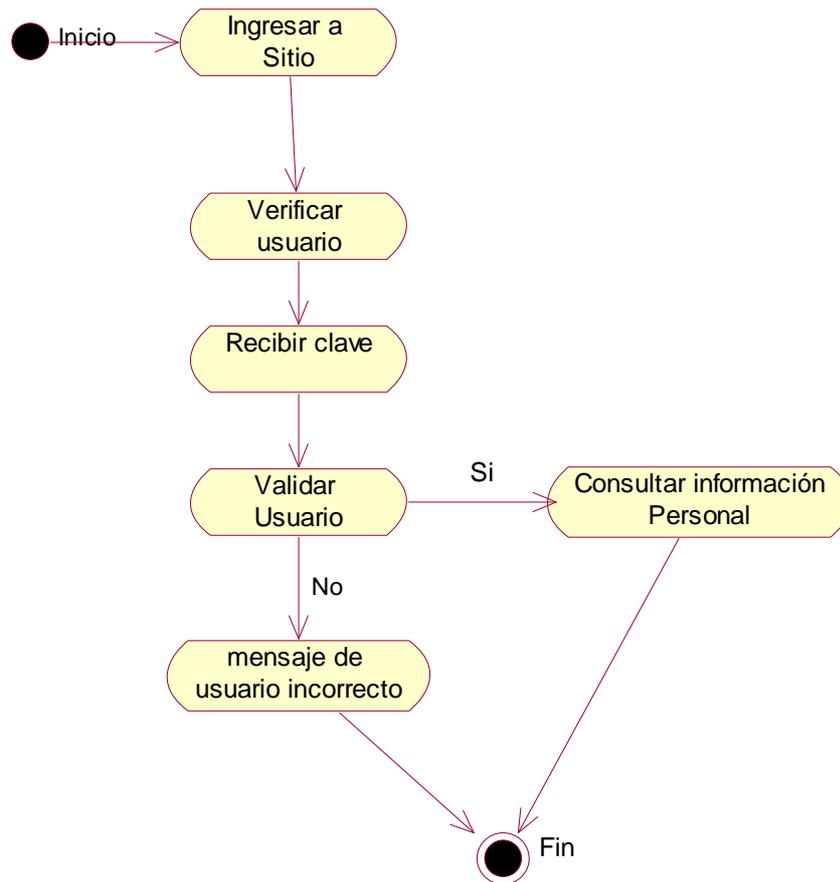
GESTIONAR CALIFICACIONES



ADMINISTRAR PÁGINA

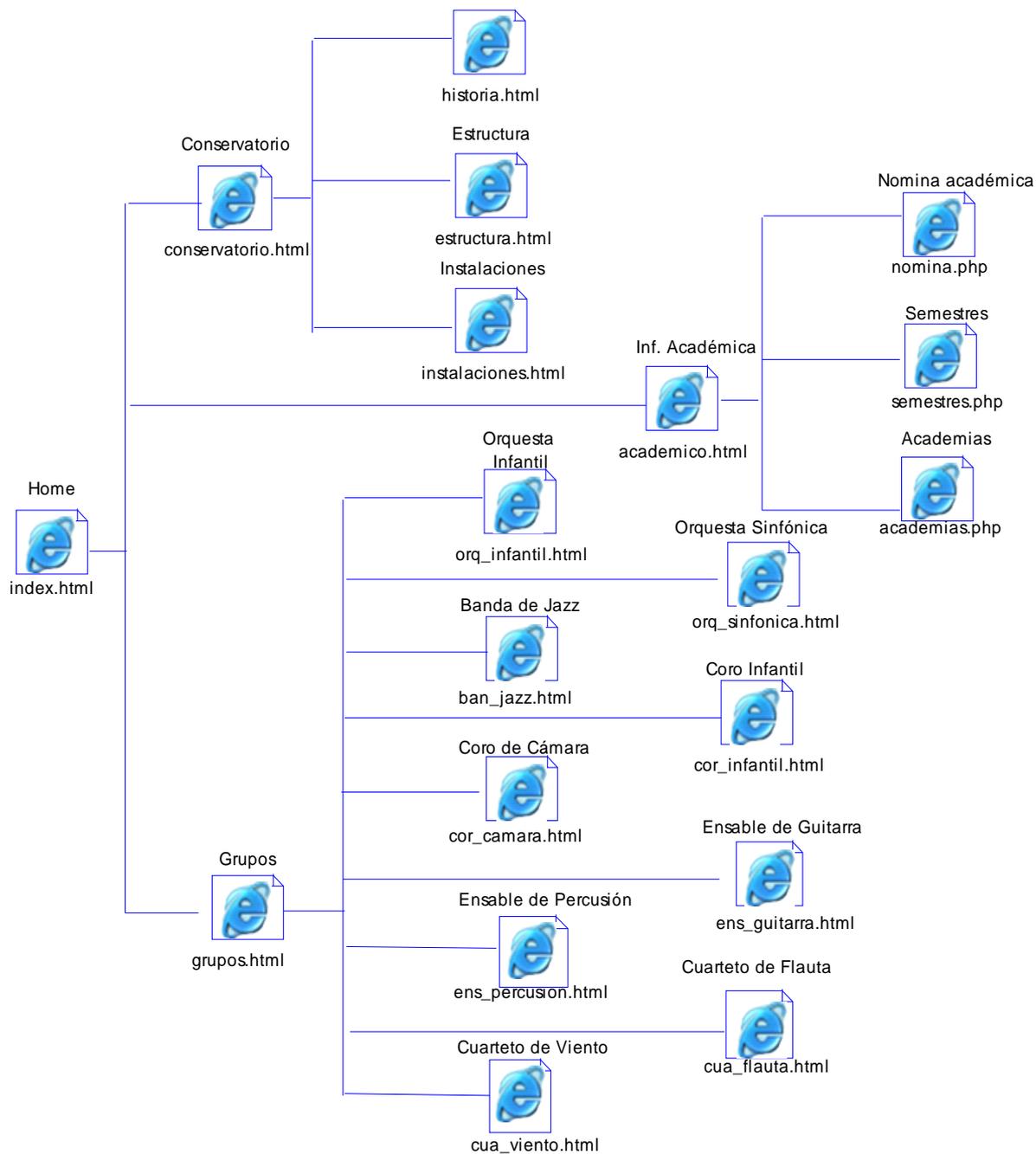


ACCEDER A PÁGINA



MODELO ARQUITECTÓNICO

DIAGRAMA ARQUITECTÓNICO



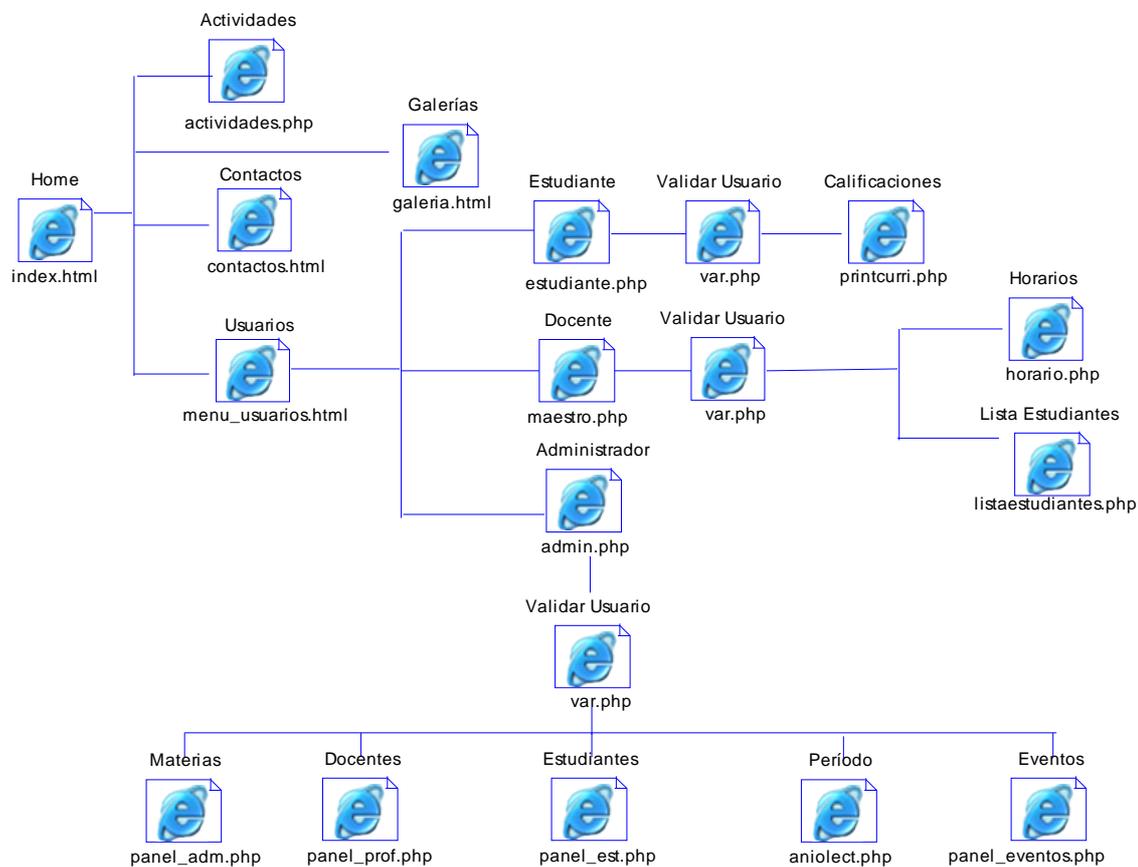
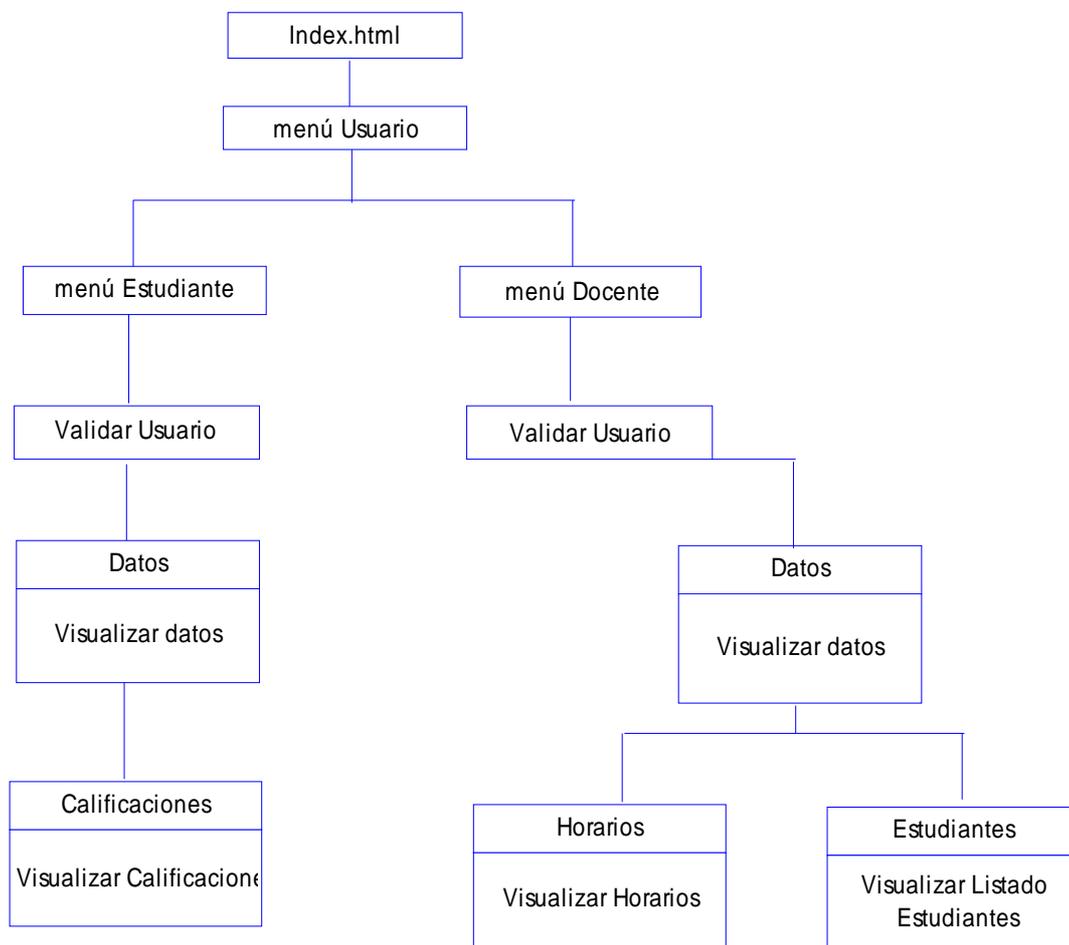
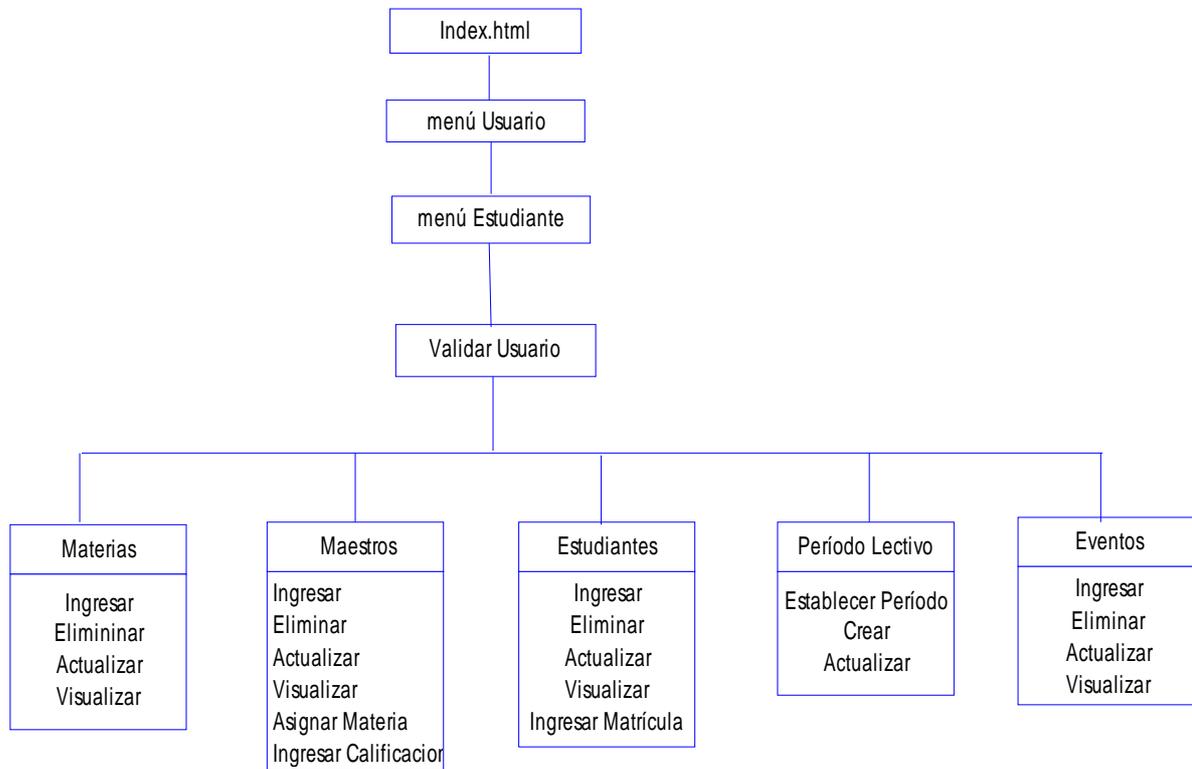


DIAGRAMA NAVEGACIONAL

DISEÑO NAVEGACIONAL PARA EL ESTUDIANTE Y EL DOCENTE



DISEÑO NAVEGACIONAL PARA EL ADMINISTRADOR

CONSTRUCCIÓN

FRONT END EN PHP

Conexión Base de Datos

```
<?php
//Funcion para establecer conexion
function conexion()
{
    if (!($link = mysql_connect("localhost","root")))
    {
        echo "Error de conexion a la base de datos";
        exit();
    }
    if (!mysql_select_db("conservatorio",$link))
    {
        echo "Erro en la base de datos";
        exit();
    }
    return $link;
}
```

Validación de Usuarios

```
function valida_ingreso ($clave,$usuario,$link,$tipo)
{ if ($tipo == "est") {
    $sql = "SELECT est_nombres, est_clave from estudiante where est_CI like
'$usuario'";
    $result = mysql_query($sql,$link);
    if ($row = mysql_fetch_array($result)) {
        if ($row["est_clave"]=="") {
            echo "Usted no tiene una clave...por favor comuniquese con el administrador";

        } else {
```

```

$sql2 = "SELECT * from estudiante where est_clave like '$clave' AND est_CI like
'$usuario'";
$result2 = mysql_query($sql2,$link);
if ($row2 = mysql_fetch_array($result2)) {
    //Codigo seguridad
    include ("estu.php"); }
else {
    //Error de clave incorrecta
    echo "<br><br><center><b>Acceso denegado...</b></center><br><br>";
    }mysql_free_result ($result2);
}
} else
{
    //Error Usuario no existe
    echo "<br><br><center><b>Acceso denegado...</b></center><br><br>";
    }}
else
{
    $sql = "SELECT doc_nombres, doc_clave from docente where doc_CI like
'$usuario'";
    $result = mysql_query($sql,$link);
    if ($row = mysql_fetch_array($result))
        { if ($row["doc_clave"]=="") {
            echo "Usted no tiene una clave...por favor comuniquese con el administrador";
        } else {
            $sql2 = "SELECT * from docente where doc_clave like '$clave' AND doc_CI like
'$usuario'";
            $result2 = mysql_query($sql2,$link);
            if ($row2 = mysql_fetch_array($result2)){
                include ("../admin/prof.php");; }
            else {
                //Error de clave Incorrecta
                echo "<br><br><center><b>Acceso denegado...</b></center><br><br>"; }
            }
        }
    }
}

```

```

mysql_free_result ($result2);    }
} else
{ //Error de docente no existe
echo "<br><br><center><b>Acceso denegado...</b></center><br><br>";
} }
return $nombre;
}
function valida_admin ($clave,$usuario,$link)
{
    $sql = "SELECT adm_nombres, adm_clave from administrador where adm_CI
like '$usuario'";
    $result = mysql_query($sql,$link);
if ($row = mysql_fetch_array($result))
{if ($row["adm_clave"]=="")
    {$acceso= 2;
    }
else
    {
    $sql2 = "SELECT * from administrador where adm_clave like '$clave' AND
adm_CI like '$usuario'";
    $result2 = mysql_query($sql2,$link);
if ($row2 = mysql_fetch_array($result2))
    { //Codigo seguridad
$acceso="1ty6789"; }
else {
$acceso = 3;
} mysql_free_result ($result2); }
}
else
    { $acceso=4;
    }mysql_free_result ($result);
return $acceso;
}

```

```

//Lista Materias
function lista_materia($link)
{
    $sql = "select * from materia";
    $result = mysql_query($sql,$link);
    while($row = mysql_fetch_array($result))
        {
            printf(" <option
value='%s'>%s&nbsp;%s</option>", $row[mat_codigo], $row[mat_codigo], $row[mat
_nombre]);
        }
    mysql_free_result($result);
}
function
guardar_materia($link,$mat_codigo,$mat_nombre,$mat_tipo,$mat_detalle,$mat_c
upo,$mat_nivel)
{
    //ALMACENAMIENTO DE DATOS
    $sql = "INSERT INTO materia VALUES
('$mat_codigo', '$mat_nombre', '$mat_tipo', '$mat_detalle', '$mat_cupo', '$mat_nivel')";
;
    $result = mysql_query($sql);
    //VERIFICACION DE USUARIOS REPETIDOS MEDIANTE $RESULT
    if (!($result))
        {
            echo "<center><p class=titulo>Datos ya existen...</p></cente>";
        }
}

//Borrar registro
function delete_materia ($link,$mat_codigo)
{
    $sql = "select * from docente_materia where mat_codigo like '$mat_codigo'";
}

```

```

$result = mysql_query($sql,$link);
$row = mysql_fetch_array($result);
if ($row[mat_codigo])
{
    echo "<b>La Materia ya esta siendo utilizada por el docente...No
se puede Borrar!!!</b>";
}
else
{
    $sql_b = "delete from materia where mat_codigo like '$mat_codigo'";
$result_b = mysql_query($sql_b);
if (!$result_b)
    {
        echo "Error al quitar los datos";
    }
}
}

//modificar registro
function modificar_materia
($link,$mat_codigo,$mat_nombre,$mat_tipo,$mat_detalle,$mat_cupo,$mat_nivel)
{
    //PROCESO DE ACTUALIZAR DATOS
    $sql = "Update materia set mat_nombre = '$mat_nombre', mat_tipo =
'$mat_tipo',mat_detalle =
'$mat_detalle',mat_cupo='$mat_cupo',mat_nivel=$mat_nivel where mat_codigo
like '$mat_codigo'";
    $result = mysql_query($sql);
    //VERIFICACION DE USUARIOS REPETIDOS MEDIANTE $RESULT
    if ($result)
    {

```

```

        echo "Datos modificados...";
    }
}

//funciones para Estudiante
function lista_estudiante($link)
{
    $sql = "select * from estudiante";
    $result = mysql_query($sql,$link);
    while($row = mysql_fetch_array($result))
    {
        printf(" <option
value='%s'>%s&nbsp;%s</option>", $row[est_CI], $row[est_nombres], $row[est_ape
lidos]);
    }
    mysql_free_result($result);
}
//Estudiante
function
save_estudiante($link,$est_CI,$est_nombres,$est_apellidos,$est_direccion,$est_t
elefono,$est_email,$mat_nivel,$grupo,$est_representante,$est_especializacion)
{
    $em_anio = anio_ac($link);
    //ALMACENAMIENTO DE DATOS
    $sql = "INSERT INTO estudiante
(est_CI,est_nombres,est_apellidos,est_direccion,est_telefono,est_email,est_repre
sentante,est_especializacion) VALUES
('$est_CI','$est_nombres','$est_apellidos','$est_direccion','$est_telefono','$est_em
ail','$est_representante','$est_especializacion)";
    $result = mysql_query($sql);
    //VERIFICACION DE USUARIOS REPETIDOS MEDIANTE $RESULT
    if (!($result))
    {

```

```

        echo "<center><p class=titulo>Datos ya existen...</p></cente>";
    }
else
    {
        //Creando Nivel
        $sql2 = "select * from materia, docente_materia where materia.mat_codigo
= docente_materia.mat_codigo and materia.mat_nivel like '$mat_nivel' and grupo
like '$grupo'";
        $result2 = mysql_query($sql2,$link);
        while($row2 = mysql_fetch_array($result2))
            {
                // $sql3 = "insert into estudiante_materia
(em_anio,est_CI,pm_ID) values('$em_anio','$est_CI','$row2[pm_ID]')";
                //$result3 = mysql_query($sql3,$link);
            }
    }
}
//Borrar registro
function delete_estudiante ($link,$cod_est)
{
    $sql = "select * from estudiante_materia where est_CI like '$cod_est'";
    $result = mysql_query($sql,$link);
    $row = mysql_fetch_array($result);
    if ($row[est_CI])
        {
            echo "<b>El docente ya tiene materias seleccionadas...No se puede
Borrar!!!</b>";
        }
    else
        {

```

```

        $sql_b = "delete from estudiante where est_CI like '$cod_est'";
$result_b = mysql_query($sql_b);
    if (!($result_b))
        {
            echo "Error al quitar los datos";
        }
    }
}

//modificar registro
function modificar_estudiante
($link,$est_CI,$est_nombres,$est_apellidos,$est_direccion,$est_telefono,$est_email,$est_representante,$est_especializacion)
{
    //PROCESO DE ACTUALIZAR DATOS
    $sql = "Update estudiante set est_nombres = '$est_nombres',est_apellidos = '$est_apellidos',est_direccion='$est_direccion',est_telefono='$est_telefono',est_email='$est_email',est_representante='$est_representante',est_especializacion='$est_especializacion' where est_CI like '$est_CI'";
    //echo $sql;
    $result = mysql_query($sql);
    //VERIFICACION DE USUARIOS REPETIDOS MEDIANTE $RESULT
    if ($result)
        {
            echo "Datos modificados...";
        }
    }

//Agregar Materia
function
agre_mat_prof($link,$diacfase,$grupo,$aula,$hora_inicio,$hora_fin,$mat_codigo,$doc_CI)
{

```

```
if (!($mat_codigo))
{
    $mensaje="Materia-";
    $error = 1;
}
if (!($diacalse))
{
    $mensaje = "$mensaje Día-";
    $error = 1;
}
if (!($aula))
{
    $mensaje="$mensaje Aula-";
    $error = 1;
}
if (!($grupo))
{
    $mensaje="$mensaje Grupo-";
    $error = 1;
}
if (!($hora_inicio))
{
    $mensaje="$mensaje Hora de Inicio-";
    $error = 1;
}
if (!($hora_fin))
{
    $mensaje="$mensaje Hora de Fin-";
    $error = 1;
}

if (!($error))
{
```

```

$doc_CI = trim($doc_CI);
$sql_b = "select * from docente_materia where diaclase like '$diaclase' and
aula like '$aula' and hora_inicio = $hora_inicio and hora_fin = $hora_fin and
mat_codigo like '$mat_codigo' and grupo like '$grupo'";
$result_b = mysql_query($sql_b,$link);
$row_b = mysql_fetch_array($result_b);
if (!$row_b[pm_ID])
{
    $sql_b1 = "Select * from docente_materia Where ((diaclase like
'$diaclase' and aula like '$aula') and ((hora_inicio <= $hora_inicio and hora_fin >
$hora_inicio) or (hora_inicio > $hora_fin and hora_fin <= $hora_fin))) and (doc_CI
like '$doc_CI')";
    $result_b1 = mysql_query($sql_b1,$link);
    $row_b1 = mysql_fetch_array($result_b1);
    if (!$row_b1[pm_ID])
    {
        //almacenar
        //ALMACENAMIENTO DE DATOS
        $sql = "INSERT INTO docente_materia
(diaclase,grupo,aula,hora_inicio,hora_fin,mat_codigo,doc_CI) VALUES
('$diaclase','$grupo','$aula','$hora_inicio','$hora_fin','$mat_codigo','$doc_CI)";
        $result = mysql_query($sql);
        //VERIFICACION DE USUARIOS REPETIDOS MEDIANTE $RESULT
        if (!$result)
        {
            echo "<center><p class=titulo>Datos ya existen...o usted no llenó
algún campo...</p></cente>";
        }
        //fin almacenar
    }
    else
    {
        echo "Materias y Horarios Cruzados...";
    }
}

```

```
    }  
  }  
  else  
  {  
    echo "Materia ya seleccionada....Verifique datos...";  
  }  
}  
else  
{  
  echo "Por favor llene todos los campos...";  
  echo $mensaje;  
  $error = 0;  
}  
}  
?>
```

MANUAL DEL USUARIO

Propósito

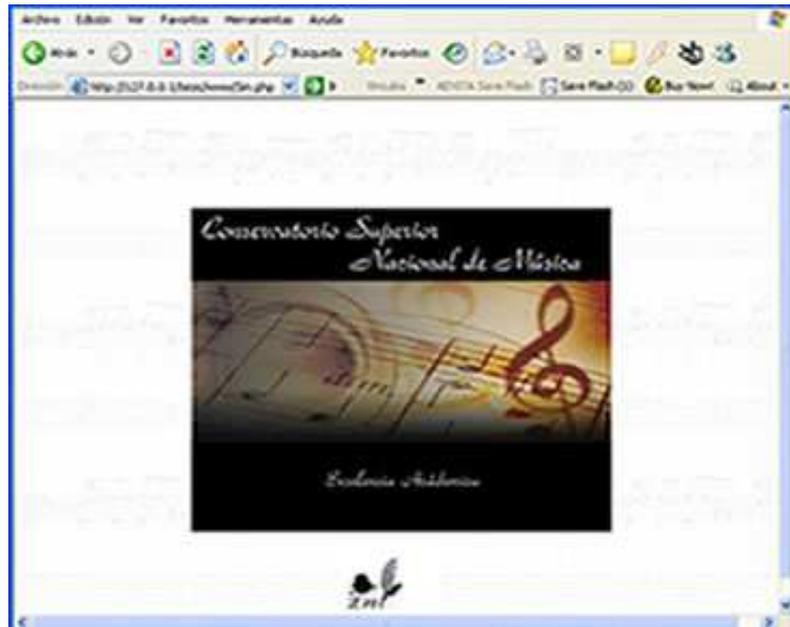
El propósito del manual de usuario es entregar toda la información posible para que el usuario de la página Web del Conservatorio Superior Nacional de Música pueda realizar de manera efectiva todos los procesos que presenta la Página Web.

Descripción De la Página Web

La pagina Web del Conservatorio Superior Nacional de Música a más de ser una página informativa que provee al internauta toda la información sobre esta institución también permite la administración de varios aspectos relacionados con el conservatorio como son las materias , maestros ,estudiantes, periodo lectivo y eventos.

FUNCIONALIDAD DEL SISTEMA

1.- Al ingresar a la pagina del Conservatorio Nacional Superior de Música pueden existir varios usuarios. Usuario Administrador, Usuario Estudiante, Usuario Maestro o simplemente como un internauta que requiere alguna información de la página.



2.- En el dar click en el menú usuarios se presenta la pantalla para escoger el tipo de usuario que interactúa con la página Web.

La pagina Web del Conservatorio Superior Nacional de Música Tiene varios tipos de usuarios específicos los cuales tienen distintos privilegios en la funcionalidad del sistema:

Administrador.

Estudiante.

Maestro.



3.- Una vez que se nos presenta la pantalla de usuarios tendremos que escoger el tipo de usuario en el menú de la nueva página. Para el ingreso de un usuario el sistema hace una verificación en la tabla usuarios de la base de datos, en la cual esta el código y una password. Si el usuario existe en la base de datos entonces iniciara sesión, caso contrario el usuario no podrá entrar al sistema y se despliega un mensaje de error.

ESTUDIANTES

Acceso al sistema estudiantes

Código :

Password :

Al iniciar sesión del ADMINISTRADOR de la página.



USUARIO ADMINISTRADOR

Este usuario tiene todos los privilegios en el sistema. Al iniciar su sesión el usuario Administrador tiene un menú en el cual el puede acceder a la administración de todo lo concerniente al sistema.

ADMINISTRACIÓN DE MATERIAS

Agregar nueva materia

Editar Materia

Borrar Materia

Agregar nueva materia

En Agregar nueva materia se nos presenta un panel en el cual el administrador ingresara los datos de la materia. Una vez que todos los campos estén ingresados correctamente se los puede guardar presionando el botón "Nuevo".

Inicio	
Materias Maestros Estudiantes Período Lectivo Eventos	
Agregar Nueva Materia	
Código :	T005
Materia :	Apreciacion Artistica
Detalle :	Apreciacion Artistica
Curso :	Segundo Infantil
Cupos Disponibles :	17
Nuevo	

La materia nueva se desplegará en el combo list de selección de materia que se encuentra abajo del panel de agregar nueva materia.

Seleccione la Materia para modificar		Datos Materia
I001 CONJUNTO CORAL I002 FLAUTA DULCE I003 HISTORIA DE LA MUSICA T001 prueba T0002 pro2 ESP001 Materia Especialidad T005 Apreciacion Artistica	Ver contenido	Código : T005 Materia : Apreciacion Artistica Detalle : Apreciacion Artistica Curso : Segundo Infantil Cupos Disponibles : 17
Editar		

Editar Materia

Si el Administrador desea editar materia entonces se dirige al panel de Selección Materia para Modificar, en un combo list se encuentran las materias en el cual debe escoger la materia para editar presionando el botón “Editar”

Seleccione la Materia para modificar

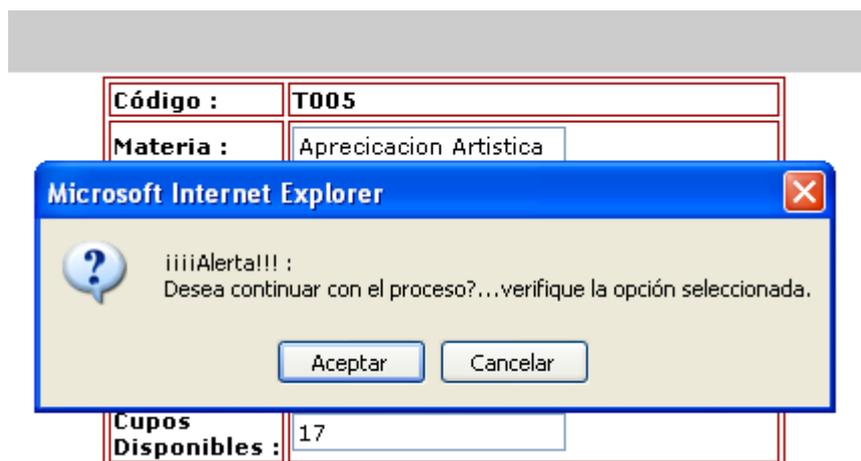
I001 CONJUNTO CORAL
I002 FLAUTA DULCE
I003 HISTORIA DE LA MUSICA
T001 prueba
T0002 pro2
ESP001 Materia Especialidad
T005 Apreciacion Artistica

Editar

Al presionar “Editar” se nos despliega un nuevo panel en el cual se encuentra los campos y sus detalles de la materia seleccionada para editar, en este panel podemos editar cualquier campo, al terminar la edición presionamos el botón “Guardar” y de nos despliega un mensaje alerta que la materia será cambiada, presionamos “Aceptar” y la materia se edito exitosamente.

Código :	T005
Materia :	Apreciacion Artistica
Tipo :	Obligatorio ▼
Detalle :	Texto Editado
Curso :	2 ▼
Cupos Disponibles :	17

Guardar



Borrar Materia

Si el Administrador desea borrar una materia se dirige al panel de selección de materia para borrar, en un combo list se encuentran las materias en el cual debe escoger la materia para borrar presionando el botón “Borrar”.



Al presionar “Borrar” tendremos un mensaje de alerta que la materia será borrada, presionamos “Aceptar” y la materia se borra exitosamente.



ADMINISTRACIÓN DE MAESTROS

Agregar nuevo Maestro

Editar Maestro

Borrar Maestro

Asignar Materia al Maestro

Ingresar Notas

Agregar nuevo Maestro

En Agregar nuevo Maestro se nos presenta un panel en el cual el administrador ingresara los datos del nuevo Maestro. Una vez que todos los campos estén ingresados correctamente se los guarda presionando el botón "Nuevo".

Materias	Maestros	Estudiantes	Período Lectivo	Eventos
Agregar Nuevo Docente				
CI :	<input type="text" value="1712691263"/>			
Nombres :	<input type="text" value="SEBASTIAN"/>			
Apellidos :	<input type="text" value="PACHECO"/>			
Título :	<input type="text" value="Tecnologo"/>			
Dirección :	<input type="text" value="MONJAS ORQUIDEAS"/>			
Teléfono :	<input type="text" value="2458937"/>			
Especialidad:	<input type="text" value="PIANO"/>			
<input type="button" value="Nuevo"/>				

El maestro nuevo se desplegara en el combo list de selección de Maestro que se encuentra abajo del panel de agregar nuevo Maestro.

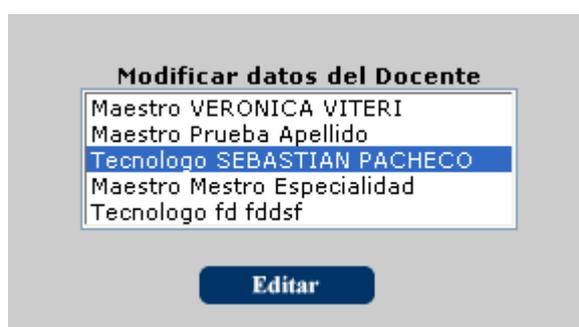


The screenshot shows a window with a button labeled 'Ver contenido' on the left. On the right, there is a table with the following data:

CI :	1712691263
Nombres :	SEBASTIAN
Apellidos :	PACHECO
Título :	Tecnologo
Dirección :	MONJAS ORQUIDEAS
Teléfono :	2458937
Especialidad :	PIANO

Editar Maestro

Para editar Maestro entonces se dirige al panel de Selección Maestro para Modificar, en un combo list se encuentran los todos los maestros en el cual debe escoger el maestro para editar presionando el botón “Editar”.



The screenshot shows a dialog box titled 'Modificar datos del Docente'. It contains a list of teachers with the following entries:

- Maestro VERONICA VITERI
- Maestro Prueba Apellido
- Tecnologo SEBASTIAN PACHECO
- Maestro Maestro Especialidad
- Tecnologo fd fdfs

The entry 'Tecnologo SEBASTIAN PACHECO' is highlighted in blue. Below the list is a button labeled 'Editar'.

Al presionar “Editar” se nos despliega un nuevo panel en el cual se encuentra los campos y sus detalles de el maestro seleccionado para editar, en este panel podemos editar cualquier campo, al terminar la edición presionamos el botón “Guardar” y de nos despliega un mensaje alerta que el maestro será modificado, presionamos “Aceptar” y el maestro se edito exitosamente.

CI :	1712691263
Nombres :	SEBASTIAN
Apellidos:	PACHECO
Titulo :	Tecnologo ▾
Direccion :	TEXTO EDITADO EN MAESTRO
Telefono :	2458937
Especialidad:	PIANO

Microsoft Internet Explorer ✖

 **Alerta!!! :**
Desea continuar con el proceso?...verifique la opción seleccionada.

Direccion :	TEXTO EDITADO EN MAESTRO
Telefono :	2458937
Especialidad:	PIANO

Borrar Maestro

Si el Administrador desea borrar una maestro se dirige al panel de selección de borrar docente, en un combo list se encuentran los maestros, en el cual debe escoger el maestro para borrar presionando el botón "Borrar".

Borrar Docente

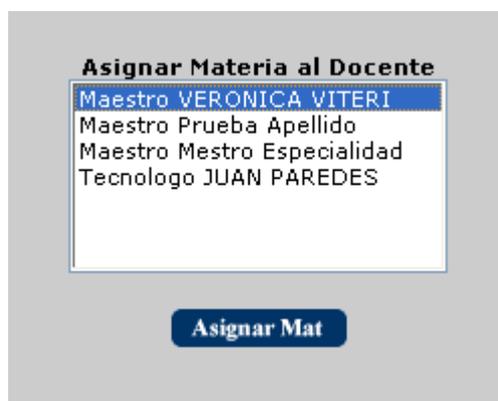
Maestro VERONICA VITERI
Maestro Prueba Apellido
Tecnologo SEBASTIAN PACHECO
Maestro Maestro Especialidad
Tecnologo fd fdfs

Al presionar “Borrar” tendremos un mensaje de alerta que el maestro será borrado del sistema, presionamos “Aceptar” y el maestro se borra exitosamente.



Asignar Materia al Maestro

Para asignar materia a un maestro nos dirigimos al panel de Asignación de materia en el cual se nos muestra los nombres de los maestros que constan en el sistema.



Seleccionamos al maestro al cual le vamos a asignar una materia y presionamos el botón “Asignar Mat”, al presionar “Asignar Mat” se nos despliega una nueva pagina en la cual nos muestra los datos del Maestro seleccionado y un panel en el que están todas las materias a escoger y también un panel de listbox para escoger el día de la clase, la hora inicial y la hora final de la clase.

Maestro VERONICA VITERI

I001 CONJUNTO CORAL
I002 FLAUTA DULCE
I003 HISTORIA DE LA MUSICA
T001 prueba
T0002 pro2

Día de Clase: Lunes

Hora Inicial: 7h00

Hora Final: 9h00

Agregar

Seleccionando la materia y llenando los listbox correctamente presionamos el botón “Agregar” y se nos despliega un panel con la materia seleccionada al maestro y sus correspondientes detalles. Para asignar otra materia al maestro debemos repetir los pasos y su asignación será exitosa.

Materia	Día	Paralelo	Hora Inicio	Hora Fin
CONJUNTO CORAL	Lunes	Paralelo_A	7h00	9h00
HISTORIA DE LA MUSICA	Sabado	Paralelo_A	10h00	11h00

Ingresar Notas

Para ingresar las calificaciones al estudiante se tiene que dirigir al panel de Ingresar Notas que se encuentra en la parte inferior derecha de la página de Administración Maestro .En este panel se encuentra un combo list que tiene una lista de todos los maestros del conservatorio. Tenemos que seleccionar al maestro y presionar “Continuar”.

Ingresar Notas-Seleccione el docente-

Maestro VERONICA VITERI
Maestro Prueba Apellido
Maestro Maestro Especialidad
Tecnologo JUAN PAREDES

Continuar

Posteriormente se nos despliega un informe del maestro seleccionado con sus datos personales y un listado de las materias a su cargo.

Datos Personales	
Maestro. VERONICA VITERI	1716892755
<div style="border: 1px solid black; padding: 5px;"> <p>I001: CONJUNTO CORAL Paralelo_A</p> <p>I003: HISTORIA DE LA MUSICA Paralelo_A</p> </div>	
<input type="button" value="Aceptar"/>	

Para continuar debemos seleccionar la materia en la cual desea poner las calificaciones, al presionar “Aceptar” se despliega una nueva pantalla en la cual están una lista de estudiantes que el maestro tiene a cargo en la materia antes seleccionada. Se selecciona al estudiante para ingresar sus notas y se presiona “Calificaciones”.

Bienvenido	Materia : Primero Infantil-HISTORIA DE LA MUSICA-Paralelo_A
Maestro. VERONICA VITERI	
 Imprimir	
Lista de Estudiantes	
<div style="border: 1px solid black; padding: 5px;"> <p>David Pacheco</p> <p>CARLA PATRICIA TORRES</p> </div>	
<input type="button" value="Calificaciones"/>	

El sistema nos despliega una pagina el la cual el maestro puede ingresar las notas al estudiante seleccionado, se presiona “ingresar notas “y las calificaciones serán guardadas exitosamente.

Materia : HISTORIA DE LA MUSICA-Paralelo_A

Alumno CARLA PATRICIA TORRES

		Ingrese valores solo para modificar e ingresar nuevas notas.
Nota 1:	=0.00	<input type="text" value="10"/>
Nota 2:	=0.00	<input type="text" value="10"/>
Total=		0
Supletorio		=0.00
Suma Final=		0
RESULTADO FINAL Reprobado		

Ingresar Notas

Finalmente en el panel tendremos también un resultado final el cual nos indicara si el estudiante reprueba o aprueba el semestre, de acuerdo con las políticas de calificaciones del conservatorio.

Materia : HISTORIA DE LA MUSICA-Paralelo_A

Alumno CARLA PATRICIA TORRES

		Ingrese valores solo para modificar e ingresar nuevas notas.
Nota 1:	=10.00	<input type="text"/>
Nota 2:	=10.00	<input type="text"/>
Total=		20 Exonerado

Ingresar Notas

ADMINISTRACIÓN DE ESTUDIANTES

Agregar nuevo Estudiante

Editar Estudiante

Borrar Estudiante

Modificar Materias

Agregar nuevo Estudiante

En Agregar nuevo Estudiante se nos presenta un panel en el cual el administrador ingresara los datos del nuevo Estudiante. Una vez que todos los campos estén ingresados correctamente se los guarda presionando el botón “Nuevo”.

Código de Estudiante:	02080
Nombres :	ANDREA
Apellidos :	CARRILLO
Dirección :	FLORESTA
Teléfono :	2846857
Email :	
Representante	ANDRES CARRILLO
Curso actual:	Primero Infantil
Instrumento Especialización:	-Seleccione especialidad- Guitarra Piano Acordeon Arpa Canto Clarinete Contrabajo Corno Fagot

Nuevo

El estudiante nuevo se desplegara en el combo list de selección de Estudiante que se encuentra abajo del panel de agregar nuevo Estudiante.

Estudiante	
Código de Estudiante:	02081
Nombres :	ANDREA
Apellidos :	CARRILLO
Dirección :	FLORESTA
Teléfono :	2846857
E-mail :	
Representante:	ANDRES CARRILLO

Ver contenido

Matrículas
-MARZO 2007-AGOSTO 2007

Matrícula	
N° Matrícula:	02081
Estudiante:	02081
Año lectivo:	MARZO 2007-AGOSTO
Fecha:	2007-02-28
Con Matrícula:	
Curso actual:	Primero Infantil Seleccione el nuevo curso Primero Infantil

Matrículas

[Imprimir](#) [Regresar](#)

Matrícula	
N° Folio: 10	N° Matrícula: 02081
Estudiante: 02081	Año lectivo: MARZO 2007-AGOSTO 2007
Nombres: ANDREA	Apellidos: CARRILLO
Dirección: FLORESTA	Teléfono:
Fecha: 2007-02-28	Con Matrícula:
Curso: Primero Infantil	

Editar Estudiante

Para editar Estudiante se dirige al panel de Selección Estudiante para Modificar, en un combo list se encuentran los todos los estudiantes en el cual debe escoger el estudiante y finalmente presionamos el botón "Editar".

Estudiante A estudiante David Pacheco Prueba Especilidad Vero viteri CARLA PATRICIA TORRES
--

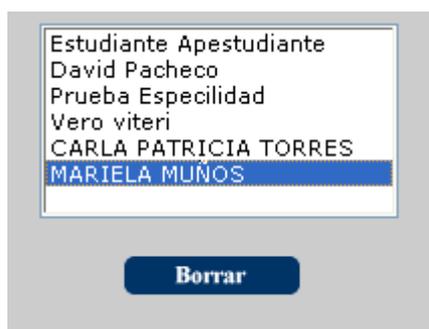
Al presionar “Editar” se nos despliega un nuevo panel en el cual se encuentra los campos y sus detalles del estudiante seleccionado para editar, en este panel podemos editar cualquier campo, al terminar la edición presionamos el botón “Guardar” y de nos aparece un mensaje alerta que el estudiante será modificado, presionamos “Aceptar” y el estudiante se edito exitosamente.

Código de Estudiante:	002079
Nombres :	CARLA PATRICIA
Apellidos:	TORRES
Direccion :	TEXTO A MODIFICAR EN ESTUDIANTE
Telefono :	2741963
E-mail :	
Representante:	CARLOS TORRES
Instrumento Especialización:	-Seleccione especialidad- Guitarra Piano Acordeon Arpa Canto Clarinete Contrabajo Corno Fagot

Ver contenido	Código de Estudiante:	002079
	Nombres :	CARLA PATRICIA
	Apellidos :	TORRES
	Dirección :	TEXTO A MODIFICAR EN ESTUDIANTE
	Teléfono :	2741963
	E-mail :	
	Representante:	CARLOS TORRES

Borrar Estudiante

Si el Administrador requiere borrar una estudiante se dirige al panel de selección de borrar Estudiante, en un combo list se encuentran los estudiantes, en el cual debe escoger el estudiante, para borrar presionamos el botón “Borrar”.



Al presionar “Borrar” tendremos un mensaje de alerta que el estudiante será borrado del sistema, presionamos “Aceptar” y el estudiante se borra exitosamente.



ADMINISTRAR PERIODO LECTIVO

El administrador puede administrar el periodo lectivo haciendo clic en el menú periodo lectivo, se despliega un panel “Sección Periodo Lectivo”, en un listbox están los periodos lectivos y al presionar “Establecer Periodo Lectivo Activo” estamos definiendo el periodo lectivo en el cual el Conservatorio se encuentra.

Sección Período lectivo	
<input type="text" value="MARZO 2007-AGOSTO 2007"/>	
<input type="button" value="Establecer Período Lectivo activo"/>	
<input type="text"/>	<input type="button" value="Crear Período Lectivo"/>
<input type="text"/>	<input type="button" value="Editar Período Lectivo"/>

Para crear un nuevo Período Lectivo ingresamos las fechas del mismo y presionamos el botón “Crear Período Lectivo”, para visualizarlo el administrador tiene que dirigirse al lisbox del panel y escoger el nuevo periodo lectivo que el conservatorio iniciara.

Sección Período lectivo	
<input type="text" value="MARZO 2007-AGOSTO 2007"/>	
<input type="button" value="Establecer Período Lectivo activo"/>	
<input type="text" value="O 2007 - AGOSTO 2007"/>	<input type="button" value="Crear Período Lectivo"/>
<input type="text"/>	<input type="button" value="Editar Período Lectivo"/>

ADMINISTRAR EVENTOS

Para administrar eventos hay que tomar en cuenta que el evento puede ser de dos tipos: el informativo que tiene que ver con todo lo concerniente a información académica y el evento propiamente dicho que son las actividades que el conservatorio organiza como por ejemplo conciertos etc.

Agregar nuevo Evento

Editar Evento

Borrar Evento

Agregar nuevo Evento

En Agregar nuevo Evento se nos presenta un panel en el cual el administrador ingresara los datos de el nuevo Evento .Una vez que todos los campos estén ingresados correctamente se los guarda presionando el botón “Nuevo”.

Tipo contenido:	<input type="text" value="Evento"/>
Título :	<input type="text" value="Concierto Sinfonico"/>
Detalle :	<input type="text" value="Concierto Sinfonico de la Sinfonica del Ecuador"/>
Gráfico/Archivo:	<input type="text"/> <input type="button" value="Examinar..."/>
Contenido:	<input type="text" value="Entadas a la venta en las puertas del teatro de la casa de la Cultura"/>
Fecha inicio:	<input type="text" value="2007/03/20"/> yyyy/mm/dd
Fecha fin:	<input type="text" value="2007/03/20"/> yyyy/mm/dd
Tema :	<input type="text" value="Concierto"/>
Sección :	<input type="text"/> Valores entre 1-3
	<input type="button" value="Nuevo"/>

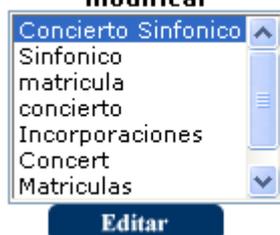
El evento nuevo se desplegara en el combo list de selección de Eventos que se encuentra en la parte inferior derecha de la página de administrar Eventos.

Datos Contenido	
Id :	28
Tipo contenido:	Evento
Título :	Concierto Sinfonico
Detalle :	Concierto Sinfonico de la Sinfonica del Ecuador
Contenido:	Entadas a la venta en las puertas del teatro de la casa

Editar Evento

Para editar Eventos se dirige al panel de “Selección de Eventos para Modificar”, en un combo list se encuentran los todos los Eventos ingresados en el sistema, en el cual se debe escoger el Evento y finalmente presionamos el botón “Editar”.

**Seleccione el contenido para
modificar**



Al presionar “Editar” se nos despliega un nuevo panel en el cual se encuentra los campos y sus detalles con la información del evento seleccionado para editar, en este panel podemos editar cualquier campo, al terminar la edición presionamos el botón “Guardar”.

Id :	28
Tipo contenido:	Evento <input type="button" value="v"/>
Título :	Concierto Sinfonico
Detalle :	Concierto Sinfonico de la Sinfonica del Ecuador
Grafico actual:	Sin gráfico...
Grafico/Archivo:	<input type="text"/> <input type="button" value="Examinar..."/>
Contenido:	Entadas a la venta en las puertas del teatro de la casa de la Cultura
Fecha inicio:	2007-03-20
Fecha fin:	2007-03-20
Tema :	Concierto <input type="button" value="v"/>
Fecha inicio:	2007-03-20
Fecha fin:	2007-03-20
Tema :	Concierto <input type="button" value="v"/>
Sección :	1 Valores entre 1-3

Posteriormente aparece un mensaje de alerta que el evento será modificado, presionamos “Aceptar” y el evento se edita exitosamente.

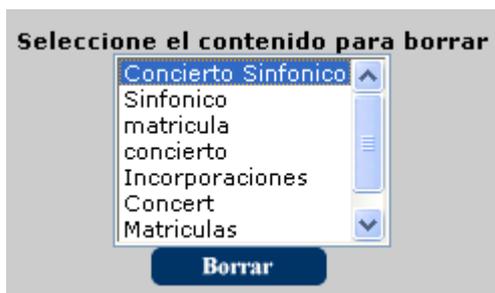
Contenido:	TEXTO A MODIFICAR EN EVENTOS
Fecha inicio:	2007-03-20
Fecha fin:	2007-03-20
Tema :	Concierto <input type="button" value="v"/>

Microsoft Internet Explorer ✖

? **Alerta!!!** :
Desea continuar con el proceso?...verifique la opción seleccionada.

Borrar Evento

Si el Administrador desea borrar una Evento se dirige al panel de selección de borrar Evento, en un combo list se encuentran los eventos existentes en el sistema, de el cual se debe escoger el evento que va a ser borrado. Para borrar presionamos el botón “Borrar”.



Al presionar “Borrar” tendremos un mensaje de alerta que el Evento será borrado del sistema, presionamos “Aceptar” y el evento se borra exitosamente.



USUARIO ESTUDIANTE

Consultar notas

El usuario Estudiante después de haber ingresado su código y su pass Word correspondiente se le despliega una página en la cual tiene acceso a sus datos personales y a sus calificaciones en el semestre.

Datos Personales	
Nombres: Pacheco David	Código: 2222



Consulta Notas

Al hacer click en consultar notas se abre una nueva pagina en la que el estudiante puede ver un informe de las sus materias con sus respectivas notas obtenidas en el semestre.

 [Imprimir](#)

Nombre: Pacheco David
Especialidad: Piano
Código: 2222

Periodo: MARZO 2007-AGOSTO 2007 ----- Curso: Primero Infantil

Código	Materia	NP1	NP2	TOTAL	SUPLE	RESULT. FINAL	ESTADO
I001	CONJUNTO CORAL	10	7	17	0	17	Exonerado

USUARIO MAESTRO

El usuario Maestro después de haber ingresado su código y su pass Word correspondiente se le despliega una página en la cual tiene acceso a sus datos personales, horarios y lista de sus estudiantes.

Datos Personales
Maestro. VITERI VERONICA **Código: 1716892755**


Horarios


Lista estudiantes

Al hacer click en el icono “Horario” se abre una nueva pagina en la que el maestro puede ver un informe de las materias que dicta en el conservatorio con sus respectivos horarios.

HORARIO DEL DOCENTE

 **Imprimir**

Maestro VITERI VERONICA

Materia	Día	Curso	Paralelo	Hora Inicio	Hora Fin
CONJUNTO CORAL	Lunes	Primero Infantil	Paralelo_A	7h00	9h00
HISTORIA DE LA MUSICA	Sabado	Primero Infantil	Paralelo_A	10h00	11h00

Al hacer click en el icono “Lista Estudiantes” se abre una nueva página en la que el maestro puede ver un informe de la lista de estudiantes a los cuales da clases en el conservatorio, mediante un listbox selecciona el paralelo y presiona el botón “Listar” y podrá obtener la lista de estudiantes del paralelo seleccionado.

LISTADO DE ESTUDIANTES

 **Imprimir**

Primero Infantil-Paralelo_A

Curso:Primero Infantil
Paralelo_A

docente :Maestro VITERI VERONICA Periodo :MARZO 2007-AGOSTO 2007

Nombre

CARLA TORRES Especialidad en:Percusion

David Pacheco Especialidad en:Piano