



T
004.6
ALV



ESCUELA SUPERIOR POLITECNICA DEL LITORAL

FACULTAD DE INGENIERIA EN ELECTRICIDAD Y COMPUTACION

“SISTEMA DE ADMINISTRACION DE CURSOS”

TRABAJO DE GRADUACION
PREVIO A LA OBTENCION DEL TITULO DE:

INGENIERO EN COMPUTACION

PRESENTADO POR:

**PATRICIO ALVAREZ ORTIZ
MARCO INCA QUINCHUELA
LETICIA SANDOYA MOSQUERA
CARLOS VASCONEZ CLAUDETT
MARLIST VILLEGAS ESPINOZA
ALFREDO ZAMBRANO DAVILA**

GUAYAQUIL - ECUADOR
1999



D-19748



BIBLIOTECA
CENTRAL

AGRADECIMIENTO

A Dios, a nuestros profesores, formadores de nuestro conocimiento a lo largo de nuestra vida estudiantil, a todas las personas que de una u otra manera colaboraron con la realización de este proyecto, y de manera muy especial a los Ing. Guido Caicedo y Servio Lima por toda su dedicación a la hora de compartir todos sus conocimientos y experiencias con nosotros.

DEDICATORIA

Dedicamos la obtención de nuestros títulos a Dios, por ser siempre nuestra guía espiritual porque sin El no habiéramos alcanzado esta meta.

También a nuestros padres, esposos, esposas e hijos porque sirvieron de inspiración y nos dieron su apoyo incondicional, a pesar de todos los sacrificios hechos.

Gracias!!





BIBLIOTECA
CENTRAL

MIEMBROS DEL TRIBUNAL

ING. CARLOS MONSALVE

Presidente del Tribunal

ING GUIDO CAICÉDO

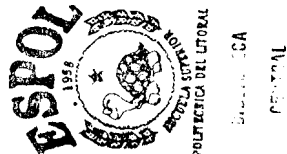
Director del Tópico

ING. SERGIO FLORES

Miembro Principal

ING. CARLOS VALERO

Miembro Principal



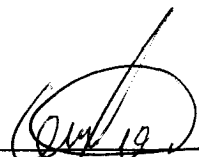
Declaración Expresa

“La responsabilidad por los hechos, ideas y doctrinas expuestas en éste documento de tópico, nos corresponden exclusivamente; y, el patrimonio intelectual de la misma, a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL”

(Reglamento de Exámenes y Títulos Profesionales de la ESPOL)



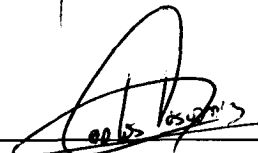
Patricio Alvarez Ortiz



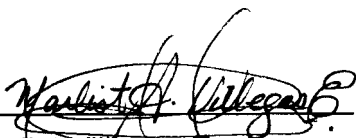
Marco Inca Quincuela



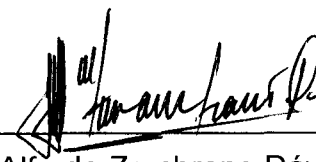
Leticia Sandoya Mosquera



Carlos Vásconez Claudett



Marlist Villegas Espinoza



Alfredo Zambrano Dávila

RESUMEN

El presente proyecto consiste en la realización del Sistema de **Administración de** Cursos, para facilitar la comunicación entre estudiantes y profesores a través del uso de un browser.

En realidad podríamos hablar de un sistema que permite administrar cursos de manera virtual. El sistema permite al profesor enviar tareas, las cuales pueden ser publicadas como un texto (por ejemplo una serie de preguntas), o como un archivo (por ejemplo una forma que debe ser llenada por el estudiante). Así mismo puede verificar las respuestas que han sido enviadas por sus alumnos, puede controlar si las tareas fueron contestadas en el tiempo o fecha establecidos, etc. , lo que le permite al profesor verificar el cumplimiento de las obligaciones de sus alumnos.

Así mismo el sistema le permite publicar el programa de la materia, las políticas con las que se va a manejar el curso, anuncios que tenga que realizar a sus alumnos, los cuales también pueden ser un texto o un archivo, como también las notas de todas las evaluaciones que los alumnos hayan realizado y de todos los trabajos entregados. Dentro de esta opción, también se ha creado una aplicación que permite al profesor evaluar el rendimiento de su curso en

términos estadísticos y así determinar que posibles ajustes se deben realizar para mejorar el rendimiento del mismo.

El alumno sólo puede ingresar las respuestas a las tareas enviadas por el profesor. Las respuestas también pueden ser tipo texto o archivo. En las demás opciones el alumno sólo puede consultar e informarse de lo que ha publicado el profesor. En el caso de que el alumno tenga alguna duda sobre alguna publicación, tarea o nota, el sistema cuenta con sistema de correo interno, lo que le permite interactuar directamente con su profesor.

Por la seguridad del sistema, es necesario un usuario administrador. Este será el encargado de crear los cursos, usuarios, asignar permisos, etc., en otras palabras es el que va a regular la interacción entre los alumnos y los profesores de las diferentes materias.

Por ser un sitio “WEB”, el sistema no podía limitarse a un acceso limitado, sino que tenía que poder ser visitado por el público, por lo que se creó opciones que puedan ser consultadas por ellos y así informarse sobre las materias que se dictan en la institución.

INDICE GENERAL

AGRADECIMIENTO.....	II
DEDICATORIA.....	III
RESPONSABILIDAD DEL DIRECTOR DE TESIS.....	IV
RESPONSABILIDAD DE LOS AUTORES.....	V
RESUMEN.....	VI
INDICE GENERAL.....	VIII
INTRODUCCION	1
1. ESPECIFICACIONES	3
1.1. Objetivos del Proyecto	3
1.2. Descripción Detallada	4
1.3. Requerimientos funcionales.....	6
1.3.1. Requerimientos Generales del Sistema	6
1.3.2. Requerimientos para el Administrador	7
1.3.3. Requerimientos para el Profesor	8
1.3.4. Requerimientos para el Alumno.	9
1.4. Justificación del Proyecto	9
1.5. Limitaciones.....	12
1.6. Restricciones del sistema	13
1.7. Alcance	14

2. ARQUITECTURA DEL SISTEMA DE ADMINISTRACIÓN DE CURSOS	15
2.1. Arquitectura Cliente-Servidor	15
2.1.1. Esquema de los Protocolos usados en esta Arquitectura	17
2.1.2. Componentes	18
2.2. Cliente Browser o Front-End Web	20
2.3. Servidor Web	21
2.4. CGI's y Perl.....	22
2.5. Servidor y Cliente de Notas.	23
2.5.1. Interacción Servidor-Cliente.....	26
3. SERVIDOR WEB	30
3.1. Esquema de la aplicación Servidor de Web	30
3.2. Arquitectura de CGI's y Módulos de Perl usados.....	30
3.3. API Html.....	36
3.4. Base de Datos	37
3.4.1. Justificación de su uso.	37
3.4.2. Modelo Entidad-Relación	40
3.4.3. Estructura de sus entidades (tablas)	41
3.4.4. Acoplamiento con el sistema de asignación de recursos.	55
3.5. Organización de Directorios.....	60

3.6. Manejo de Autenticación y Seguridad.....	63
4. SERVIDOR Y CLIENTE JAVA	65
4.1. Servidor y Clientes original	65
4.2. Requisitos para la integración del nuevo sistema.....	66
4.3. Modificaciones al Servidor Original.....	66
4.4. Modificaciones al cliente original.....	67
5. DISEÑO DE LA INTERFACE CON EL USUARIO	71
5.1. Modos de interaccion.....	71
5.2. Plantilla del sitio	72
5.3. Facilidades de navegacion	74
5.4. Estructura de las pantallas.....	75
5.4.1. Cabecera Horizontal.-.....	75
5.4.2. Cabecera Central	75
5.4.3. Barra de botones.....	75
6. CONCLUSIONES.	80
BIBLIOGRAFÍA.....	82

INTRODUCCION

El mundo de las computadoras y del procesamiento de la información está creciendo rápidamente. En el centro de este sorprendente crecimiento está el Word Wide Web, también conocido como el Web o WWW. Solo hasta hace unos pocos meses atrás, el Web fue un medio para integrar documentos entre gente y computadoras vía el Internet. Hoy en día, rápidamente se ha vuelto un entorno de desarrollo distribuido, capaz de proporcionar información y otros recursos de aplicación a millones de personas alrededor del mundo.

Diversas instituciones y empresas están implementando aplicaciones basadas en Web, para facilitar la comunicación e integrar la información entre sus empleados y proveedores.

Este proyecto tiene como objetivo hacer uso de la tecnología antes mencionada para proveer de un medio eficiente y seguro para administrar información académica y contribuir a la nueva era de interacción virtual entre alumno y profesor.

Entre sus principales ventajas, tenemos el hacer más eficiente el envío y recepción de tareas, consulta de notas, anuncios o políticas de un curso, debido a que el alumno o profesor no necesita estar físicamente presente en la Universidad, sino que puede hacerlo desde cualquier punto conectado a la Internet.

Todas las personas que utilicen este sistema, se verán beneficiados por la rapidez de consultar la información actualizada y el costo es relativamente bajo, debido a que no se necesitan muchos recursos de hardware o software.

1. ESPECIFICACIONES

1.1. Objetivos del Proyecto

- Proveer a los alumnos de una interacción virtual con su profesor, de manera que pueda: Consultar, Responder Tareas, Consultar calificaciones, etc., sin necesidad de presencia física ante el profesor.
- Proveer al profesor de un medio virtual para que pueda: Asignar, Revisar tareas; Publicar calificaciones, sin necesidad de una presencia física del alumno.
- Dar facilidad a los profesores de crear sus propios cursos con los alumnos que él desee tener y asegurarse de que las tareas y las calificaciones que ingresen al sistema no podrán ser alteradas ni extraviadas.
- Brindar un uso fácil y rápido del sistema desde cualquier punto geográfico, desde donde se tenga acceso a Internet .

- Dar seguridad, confiabilidad, basado en esquema de autenticación, que permite acceder a los recursos de acuerdo al nivel de seguridad y premisos de la clave.

1.2. Descripción Detallada

El Sistema de Administración de Cursos se ha desarrollado como una aplicación Cliente - Servidor bajo la arquitectura TCP/IP. El programa servidor será instalado en un computador que provea este servicio, el mismo que tendrá que responder a los requerimientos de los clientes.

A continuación se muestra un diagrama que describe la interacción del servidor con cada cliente. (Ver fig. 1.1.)

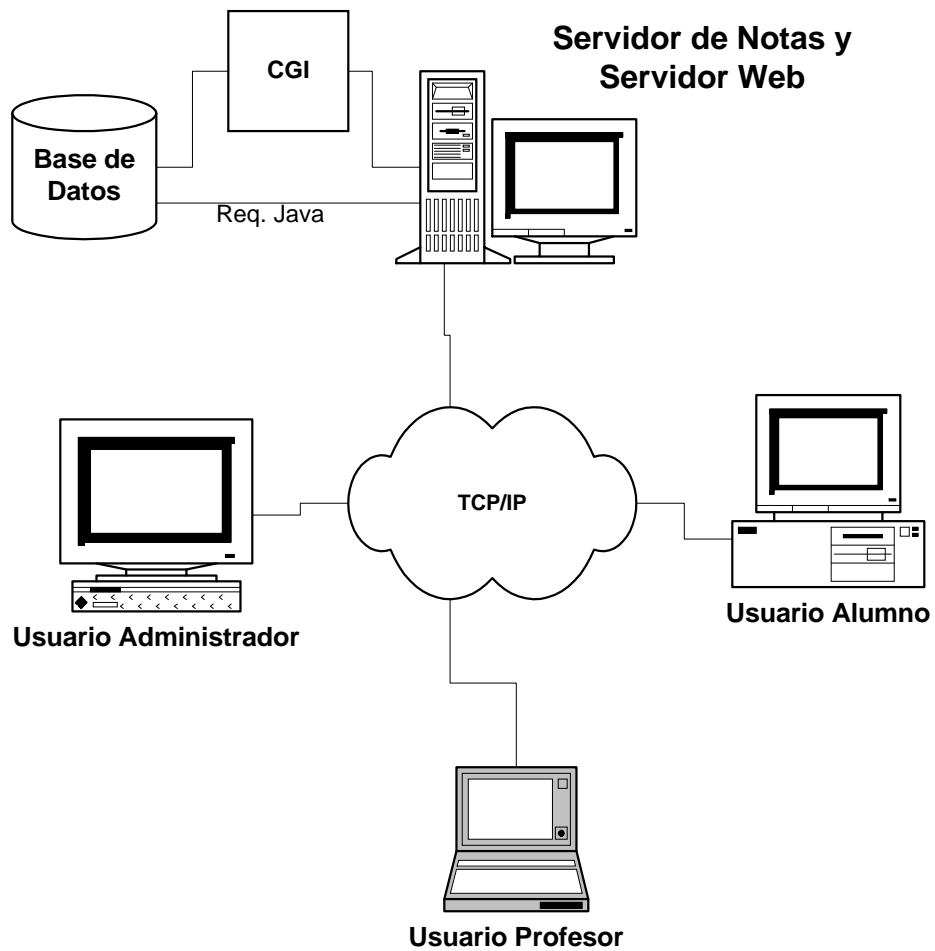


Fig.1.1. Ilustración de la arquitectura Cliente - Servidor del Sistema de Administración de Cursos.

La siguiente sección (1.3) procederá a detallar los requerimientos necesarios para el servidor y cada uno de los clientes.

1.3. Requerimientos funcionales

1.3.1. Requerimientos Generales del Sistema

- Establecer y manejar la comunicación con cada cliente de manera confiable y segura.
- Proveer seguridad bajo un esquema de autenticación para la identificación de los usuarios.
- Interactuar con una base de datos y obtener toda la información necesaria para que el sistema pueda formular las respuestas que satisfagan los requerimientos de los clientes. La obtención de la información se realizará a través de sentencias SQL bajo Perl.
- Llevar un registro de materias, alumnos y cursos.

- El administrador del sistema podrá tener el control sobre las materias, profesores o alumnos que lo podrán utilizar.
- El sistema deberá proveer los servicios de control de tareas, anuncios, notas, políticas y cronograma.
- Llevar un antecedente histórico de las tareas así como de las notas de los alumnos de un determinado año/periodo.
- Prevenir y advertir los errores o situaciones especiales que pueden ocurrir durante la ejecución del sistema. Por Ejemplo, el ingreso valores no permitidos en una nota, el envío de una tarea fuera de la fecha permitida, etc.

1.3.2. Requerimientos para el Administrador

1. Control de claves para la seguridad de la Base de Datos.
2. Crear y denegar el acceso al sistema tanto de alumnos, como de profesores.

3. Determinar el servidor contra que servidor se desea autenticar las claves de acceso.
4. Cambiar la interfaz del sistema a través de un metaformato para hacer más sencillo los cambios en la presentación de la información.

1.3.3. Requerimientos para el Profesor

1. Ingresar al sistema desde cualquier lugar que tenga acceso a Internet.
2. Determinar que alumnos pueden ingresar a consultar datos de su materia.
3. Controlar las tareas, anuncios, cronograma de estudios, políticas y notas de las materias dictadas. Por ejemplo, la eliminación o ingreso de nuevas tareas o anuncios.

1.3.4. Requerimientos para el Alumno.

1. Ingresar al sistema desde cualquier lugar que tenga acceso a Internet.
2. Consultar todas las opciones disponibles para él en el sistema tales como tareas, anuncios, políticas, cronograma y notas .
3. Responder las tareas enviadas ya sea como archivo o como texto.

1.4. Justificación del Proyecto

A continuación se detallarán las justificaciones de la necesidad de implementar el sistema de Administración de Cursos:

- **Agiliza conocimiento de las tareas:** El sistema servirá como un medio de comunicación ágil entre los alumnos y el profesor, esto es, para que los alumnos tengan conocimiento de las tareas inmediatamente después de que el profesor las ingrese al sistema

(incluso lo podrán hacer desde la comodidad de sus hogares), sin tener que depender de la ubicación del profesor.

➤ **Sencillo, intuitivo e interactivo, beneficiándose del World Wide**

Web: Es bastante sencilla la navegación en Internet a través de un "Browser", es tan fácil como hacer un "click" en un botón, escoger una opción de una lista desplegable o llenar una caja de diálogos con datos que el usuario conoce perfectamente tal como su "User", su "Password", mostrando mensajes de errores que los adviertan y guíen siempre hacia el camino correcto. Además, no requiere la instalación de otra aplicación fuera de un browser de World Wide Web en la máquina de los usuarios.

➤ **Información en línea:** El sistema brinda información tal como anuncios, cronograma o políticas del curso, ingresados por el profesor ingresa, de tal forma que los alumnos siempre estén informados de cualquier actividad de su curso.

➤ **Privacidad al enviar las tareas:** Cada alumno tiene su propio directorio en donde "depositan" todas sus tareas. Estos directorios son privados y no pueden ser accedidos por otros alumnos.

- **Seguridad de datos en la red:** En el inmenso mundo de Internet es común el ataque de los "Hackers" sobre las redes, en muchas ocasiones buscan alterar datos, lo que podría ser un problema muy serio. Por ejemplo, si se borra la tarea de un alumno, se podría perjudicar con una baja calificación e incluso si la tarea era importante podría costar la reprobación de la materia.

- **Total Control del Sistema:** Debido al esquema de seguridad del sistema, utilizando un tiempo de expiración, el administrador tiene la facultad de determinar el tiempo de permanencia de los usuarios.

Además, permite manejar el sistema con una interfaz fácil de operar, de manera remota, permite utilizar un archivo de configuración (sistema.conf) que me permite configurar algunas características del sistema como por ejemplo el nombre de la base de datos a utilizar, ubicación y nombres de directorios, dirección y puerto (protocolo POP3) contra el cual los usuarios del sistema se autentificaron.

- **Fácil y eficiente revisión de Tareas por parte del Profesor:** El profesor podrá desde cualquier lugar revisar las tareas sin necesidad de recibirlas exclusivamente en la Universidad. Además podrá poner límite en la fecha de entrega de éstas, para de esta manera conocer los alumnos que las respondieron dentro de la fecha límite o no.

1.5. Limitaciones

- Los profesores no podrán crear cursos que no correspondan al semestre y año actual; no podrán crear un curso futuro o uno que haya finalizado su período de vigencia.
- El sistema no puede funcionar directamente con los datos almacenados en la ESPOL, sino que existirá un proceso de migración de dichos datos a la base de datos usada por este sistema.
- No se podrán publicar políticas, ni el cronograma del curso como archivos. Se publicarán estas dos opciones del sistema únicamente como texto.

1.6. Restricciones del sistema

El sistema debe cumplir los siguientes requerimientos de software:

Para el Servidor:

- Linux Red Hat versión 5.2 como sistema operativo base para el servidor.
- Un "Browser" (Netscape, Internet Explorer, etc.) que permita visualizar JAVA.
- Un compilador de JAVA. (JDK 1.1.7)
- Una base de datos Postgres.

En cuanto al "Hardware" del Servidor necesario para la ejecución del sistema necesita:

Procesador

- Pentium de 100 Mhz. como mínimo.

Memoria

- 32 MB en RAM (recomendable) como mínimo.

Espacio Libre en Disco Duro

- 2 GB en disco duro como estimado.

Para el Cliente:

Software

- Win 3.1 o Superior
- Protocolo TCP/IP
- Browser con soporte para Java

Hardware

- 16 MB en memoria RAM
- Procesador Pentium 100 MHz
- 15 MB de espacio libre en Disco Duro

1.7. Alcance

- El sistema podrá controlar todo lo referente a tareas, anuncios, políticas y cronogramas de estudio (Ingreso, eliminación, consultas, etc.)
- El sistema no podrá proveer un control de calificación de tareas, es decir, el alumno no podrá saber cuando su tarea fue revisada o que calificación obtuvo.

- El sistema no permite enviar los anuncios o políticas como archivos, sólo se lo podrá hacer como texto.

2. ARQUITECTURA DEL SISTEMA DE ADMINISTRACIÓN DE CURSOS

2.1. Arquitectura Cliente-Servidor

Debido a que nuestro sistema funciona bajo la arquitectura cliente-servidor, debemos decir que esta es descrita como una extensión lógica de la programación modular; lo que significa que un software complejo puede ser separado en un conjunto de módulos, cada uno de los cuales esté diseñado para desempeñar un conjunto limitado de funciones. Cada módulo es entonces invocado como parte de un programa principal.

En algunas ocasiones, una aplicación puede requerir que un procesamiento deba ser hecho por otro proceso o programa. En este tipo de entorno de comunicación cliente - servidor, el proceso o programa que requiere un servicio es considerado "*el cliente*", mientras que el proceso o programa que provee un servicio es considerado "*el servidor*".

El modelo cliente-servidor está basado en el concepto de que cada aplicación consiste de dos partes funcionales: Una que inicia la comunicación (Cliente) y otra que la responde (Servidor). El servidor espera por requerimientos de comunicación de cualquier cliente, ejecuta la acción requerida por él, y retorna el resultado al cliente.

Nuestro sistema de Administración de Cursos se basa en esta arquitectura, utilizando un Servidor Web y un cliente (Browsers), por medio del protocolo HTTP (Hyper Text Transfer Protocol).

Una aplicación cliente-servidor maneja todas las transacciones: requerir, recibir y procesar el documento. El usuario final no sabe qué es lo que está ocurriendo durante el procesamiento, en lo único en que se concentra es en obtener la información requerida.

2.1.1. Esquema de los Protocolos usados en esta Arquitectura

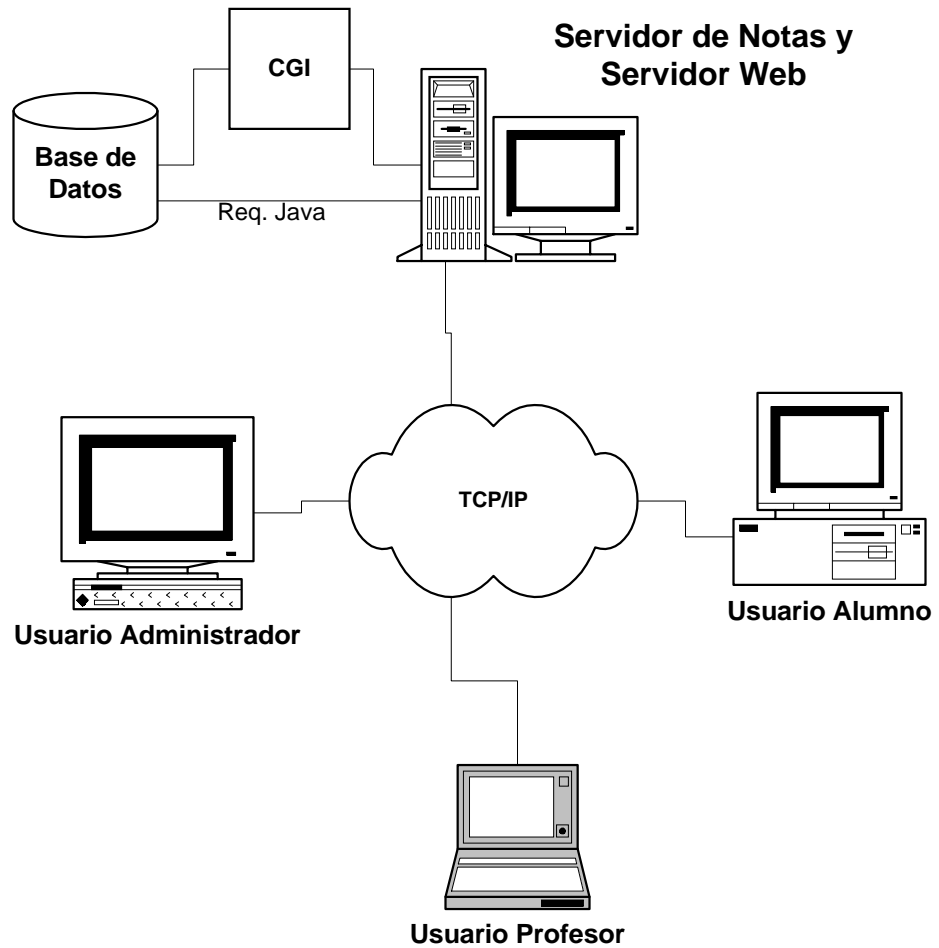
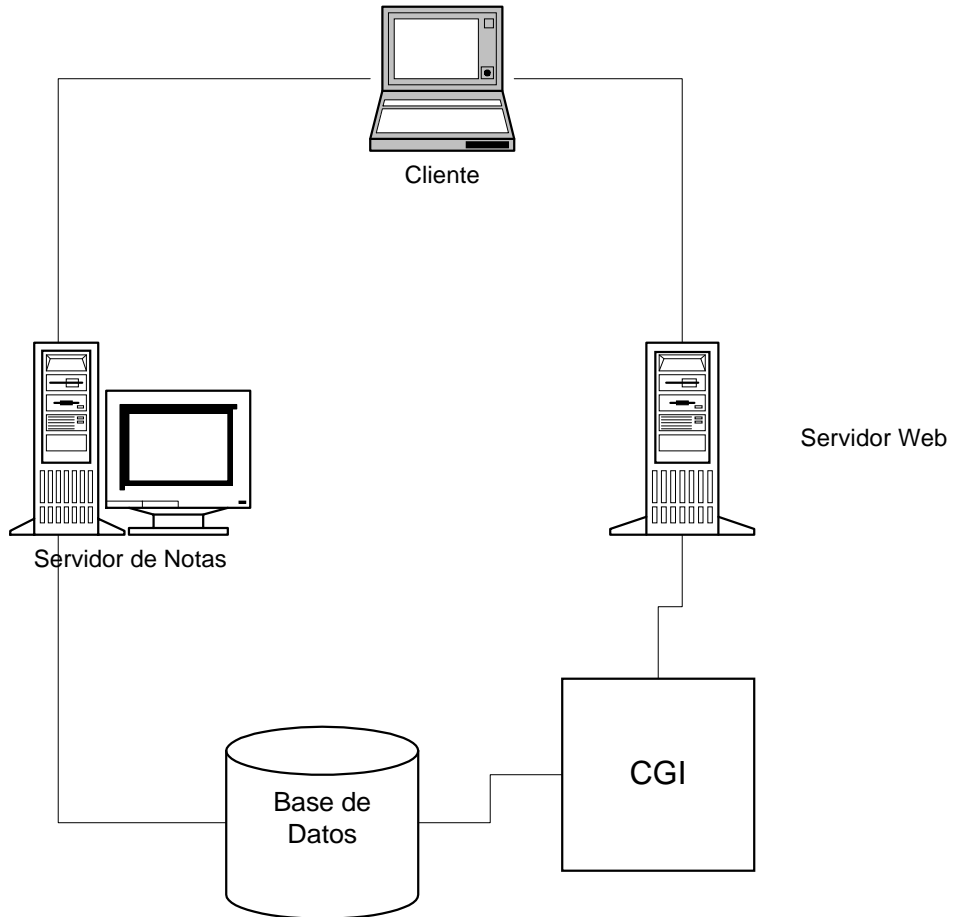


Fig. 2.1.1. Gráfico de los Protocolos Usados

2.1.2. Componentes



Servidor de Web

Los servidores Web (HTTP) proporcionan acceso a documentos distribuidos de hipertexto, aplicaciones y base de datos. HTTP es un protocolo orientado a objetos y "Stateless", es decir que no mantiene información de la conexión durante la transferencia. Debido a esta

limitación para no mantener información de la aplicación durante múltiples sesiones, debemos ser creativos con nuestros documentos HTML para pasar información entre ellos.

Los servidores HTTP operan bajo un paradigma requerimiento/respuesta en donde un cliente establece una conexión con un servidor y le envía un requerimiento. El servidor responde con una línea de status, incluyendo la versión del protocolo de mensaje y un código de error o éxito, seguido por un mensaje de tipo MIME (Multipurpose Internet Mail Extensions).

Servidor de Notas

Este servidor se encuentra desarrollado totalmente en Java; se encarga de atender todos los requerimientos (consultas) que el cliente realice. Utiliza un protocolo que interpreta específicamente la información que el cliente necesita de la base de datos a través de un código (como por ejemplo CMP “ConsultarMateriaProfesor”) que indica el tipo de consulta a realizar.

Cliente Notas

El Cliente Notas fue desarrollado en Java, se encarga de hacer requerimientos al “Servidor de Notas”, utilizando el protocolo de comunicación implementado en éste. El Cliente Notas le envía un código (como por ejemplo CNC “ConsultaNotasCurso”) al servidor indicando el tipo de consulta requerida y recibe de éste el resultado de dicha consulta. Existen dos tipos de clientes: alumno y profesor.

2.2. Cliente Browser

Nuestro sistema de Administración de Cursos utiliza como cliente a los browsers, estas aplicaciones se encargan de proporcionar la interfaz entre el sistema y el usuario.

El browser es responsable de recibir entradas del usuario, iniciando requerimientos de transferencia de documento con el servidor de web, recibiendo el código del documento HTML del servidor y convirtiendo el mismo documento en una salida con formato para desplegarlo en la pantalla. Existen browsers para todos los sistemas operativos y son capaces de comunicarse con todos los servidores de web.

Los documentos HTML pueden contener hiperenlaces. Cuando hacemos "click" en uno de ellos, estos hacen que el browser contacte al apropiado servidor de web para el documento. Los browsers soportan niveles predefinidos de especificaciones HTML. Estas especificaciones determinan cómo los browsers leen los "tags" HTML contenidos dentro del documento; sin embargo, debido a los browsers de diferentes fabricantes, no todos soportan exactamente los mismos niveles de especificación HTML.

Es necesario recalcar que habrá diferencias obvias entre desplegar la salida de un documento entre diferentes browsers, como Internet Explorer o Netscape Communicator.

2.3. Servidor Web

El sistema de Administración de Cursos utiliza como servidor de web el Apache, que es una de los más estables y populares que se utilizan en internet. Este contiene un módulo que se encarga de administrar todos los CGI que muestran la información por medio del browser.

El Apache tiene predefinido los directorios en donde deben funcionar las páginas elaboradas en html o CGI, con sus respectivos atributos o permisos para su ejecución.

De igual manera, Apache, se encarga de velar para que todas las páginas solicitadas por el cliente browser, se ejecuten de una manera segura y confiable.

2.4. CGI's y Perl

Common Gateway Interface. CGI, es un standard de programación de interfaces para servidores Web. Nos ofrece una manera para hacer nuestras sitios Web dinámicos e interactivos.

Perl es un compilador que funciona como interpretador. Compila el código de programa dentro del mismo, antes de ejecutarlo, entonces ocurre una optimización, y el ejecutable funciona fácilmente. Perl no escribe este código en un archivo ejecutable separado, como es normal, se graba el código en la memoria y se ejecuta.

El sistema fue realizado en su gran mayoría utilizando CGI's (implementados en Perl), con un esquema modular de rutinas dentro de cada uno.

El CGI recibe algunos parámetros con los que puede realizar las diferentes consultas a la Base de Datos y devolver la información al cliente por medio del browser.

2.5. Servidor y Cliente de Notas.

El Servidor y Cliente de Notas fueron implementados en JAVA por un tópico anterior, y se lo agregó al presente proyecto para así completar la administración de cursos incluyendo las notas de los estudiantes.

El Servidor se encarga de recibir los requerimientos del cliente, consultar con la Base de Datos, y enviar de vuelta (al browser) la información solicitada.

El programa Servidor está adaptado para que funcione en computadoras que utilicen el sistema operativo Linux. La comunicación entre el servidor y el cliente se realizará utilizando

sockets que, asignados a procesos esclavos, se encargarán de atender los requerimientos de cada cliente. El lenguaje en que han sido implementados tanto el cliente como el servidor, es lenguaje de programación JAVA.

A continuación, se muestra una ilustración de la arquitectura utilizada.

(Fig. 2.5)

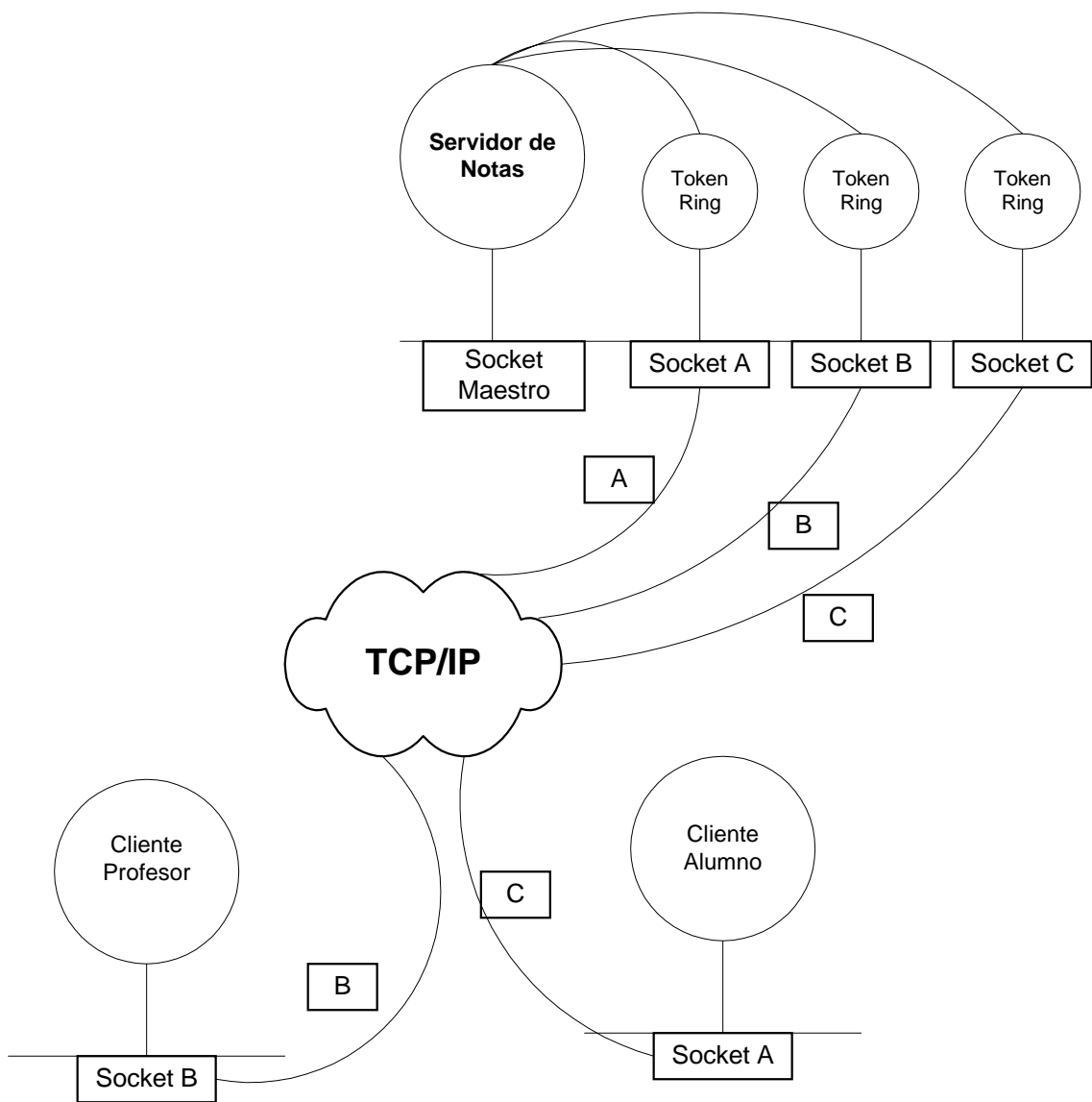


Fig.2.5. Diagrama de procesos que conforman la Arquitectura del Servidor y Cliente JAVA.

A continuación se detallará la interacción del servidor con el cliente JAVA.

2.5.1. Interacción Servidor-Cliente

En el computador donde se encuentre el programa servidor, deberá estar la Base de Datos y el Servidor Web, ya que si no estuvieran en el mismo lugar, habría que implementar otro programa cliente-servidor para acceder a dicha base. Esta facilidad provee velocidad en el procesamiento de la información porque el servidor es el único que tiene la facultad de acceder a la base de datos para realizar las consultas y actualizaciones.

- **¿Cómo se comunica el cliente con el servidor?**

Para lograr una comunicación confiable entre el cliente y el servidor al nivel de la capa de transporte, se utiliza el protocolo TCP.

El cliente envía una clave al servidor indicando el requerimiento (CMP consultar materia profesor), este mantiene la comunicación con el cliente hasta que le envíe la respuesta y se cierra la comunicación.

- **¿Cómo envía el servidor los resultados?**

El cliente envía al servidor el código correspondiente al requerimiento deseado; el servidor lo interpreta y toma la información de la base de datos para luego enviarla encriptada hacia el cliente. El encriptamiento de los datos es transparente para el usuario y únicamente se le presenta la información requerida.

2.6. Seguridades.

Autenticación.

En el presente sistema utilizamos autenticación, que significa saber quién es la persona que ingresa. Y demostrar que *“es quien dice ser”*, es decir, se debe demostrar al sistema que en realidad se es la

persona autorizada para ingresar, todo esto se lo puede hacer utilizando usuarios y claves privadas.

Para esto los usuarios y claves se encuentran almacenados en otro servidor. Se utiliza el esquema de autenticación de POP3 (Post Office Protocol) que es la versión mas reciente de un protocolo standard cliente-servidor para recibir e-mails.

El ejemplo práctico es el User y password:

User: topico

El sistema dice: "Está bien, eres del tópico, demuéstalo."

A continuación pide una clave:

password: xxxxxx

Si la persona que desea ingresar al sistema si está autorizada, el user y password serán los correctos, entonces podrá entrar al sistema; caso contrario, no lo hará.

Encriptamiento.

Para el encriptamiento se utilizó SSL (Security Socket Layers), que nos permite enviar y recibir información de una manera confiable y segura.

SSL se encarga de "encriptar" los datos para poder enviarlos por internet, asi mismo cuando estos llegan los "desencripta" obteniendo la información originalmente enviada.

Un servidor de Web que soporte SSL, generalmente atiende los requerimientos normales en el puerto 80 y los requerimientos encriptados en el puerto 443. Para poder utilizar este encriptamiento, se debe adicionar al URL: **https://**

Cookies.

Otro esquema de seguridad utilizado en el presente proyecto es el de cookies, mediante estos, se puede restringir el acceso tanto a profesores como alumnos a los diferentes directorios. Además, permite al sistema determinar el tiempo de permanencia de un usuario dentro del mismo.

2.7. Mantenimiento de la Base de Datos.

Todos los datos de alumnos, profesores, materias, etc. se encuentran almacenados en una base de datos; en este sistema la base utilizada es Postgres.

La información almacenada en ésta es mantenida usando cualquier browser que accese a diversas pantallas en donde se pueden actualizar, ingresar, consultar o eliminar datos.

3. SERVIDOR WEB

3.1. Esquema de la aplicación Servidor de Web

El comportamiento del sitio web se basa íntegramente en CGI's. Para que estos funcionen correctamente, necesitamos un Servidor de Web, que en este caso será el Apache.

El Apache se encarga completamente de ejecutar correctamente todos los CGI's; si existiera algún error en la ejecución de los mismos, el Apache se encarga de enviar el respectivo mensaje hacia el cliente informando del evento, adicionalmente se generan "logs" de error en el Sistema Operativo, indicando la razón por la que el CGI no se ejecuta correctamente (Ej. Acceso a la Base, tipo de dato, etc.)

3.2. Arquitectura de CGI's y Módulos de Perl usados

Todos los CGI's se encuentran ubicados dentro del directorio CGI-BIN del sistema operativo. Adicionalmente, usan rutinas para darle la respectiva presentación a las páginas que el cliente va a observar en su monitor. El detalle de la organización de los directorios en donde se ubican los CGI se encuentran en la sección 3.5.

Se han dividido los CGI's de la siguiente manera:

- a) Administrador
- b) Profesores
- c) Alumnos
- d) Público

Nota: Los Administradores, profesores y alumnos necesitarán una clave de acceso para ingresar al Sistema.

a) Administrador

Se encuentran dentro del directorio Admin (Default). Estos CGI's son los encargados de la administración del sistema; aquí se asignan las materias y profesores que aparecerán en éste y viceversa.

b) Profesores

Se encuentran dentro del directorio admin_tareas (Default). A esta sección sólo pueden ingresar los profesores a sus respectivas materias y paralelos.

Estos CGI's son los encargados de administrar todo lo referente a las opciones que el profesor puede manipular en el sistema.

Los CGI's usados los podemos clasificar por secciones:

Anuncios:

Existen varios CGI's en donde el profesor puede ingresar, modificar, eliminar anuncios en su paralelo:

Ingreso: Se ingresa el texto del anuncio. Todos los anuncios son creados con un número secuencial que empieza en todos los casos con 1. La fecha de ingreso del anuncio es la actual del sistema.

Modificación: Se puede modificar el texto ingresado del anuncio, la nueva fecha será la de la modificación del mismo.

Eliminación: Al eliminar un anuncio del sistema, automáticamente desaparecerá.

Tareas:

Existen varios CGI's en donde el profesor puede ingresar, modificar, eliminar, revisar tareas en su paralelo.

Ingreso: El profesor ingresa la tarea (texto o archivo), y la fecha en la que debe ser entregada. Todas las tareas son creadas con un número secuencial que empieza en 1. La fecha de ingreso de la tarea es la fecha del sistema.

Modificación: Se puede modificar una tarea, ya sea cambiando la fecha de entrega, o cambiando de archivo a texto o viceversa.

Eliminación: Al eliminar una tarea, ésta desaparece automáticamente del sistema.

Consultas: Todos los alumnos que respondan la tarea saldrán en el sistema. Para la consulta de éstos, el sistema los divide en dos:

- Alumnos que respondieron a tiempo
- Alumnos que no respondieron a tiempo.

En ambos casos, saldrá un listado de todos los alumnos que tienen su tarea respondida y cómo la respondieron (texto o archivo). Si es texto, la información saldrá en la pantalla, y si es un archivo, el browser se encargará de mostrarlo con la respectiva aplicación asociada en su sistema (Word, Excel, notepad, etc.)

Políticas del Curso:

Existen varios CGI's en donde el profesor puede ingresar, modificar, eliminar las políticas del curso.

Ingreso: Se ingresa el texto de la política. Estos aparecerán tal cual son ingresados por pantalla. Todas las políticas son almacenadas con un número secuencial que empieza en 1.

Modificación: Se puede modificar el texto ingresado de la política.

Eliminación: Al eliminar una política del sistema, automáticamente desaparecerá.

Cronograma:

Existen varios CGI's en donde el profesor puede ingresar, modificar, eliminar y marcar el avance del cronograma de la materia:

Ingreso: Se ingresa el texto del cronograma. Este aparecerá tal cual es ingresado por pantalla.

Modificación: Se puede modificar el texto ingresado del cronograma.

Eliminación: Al eliminar un punto del cronograma, automáticamente desaparecerá.

Marcar: A medida que se vaya avanzando en el semestre, el profesor puede marcar los puntos vistos, de tal manera que tanto alumnos como profesores puedan ver el avance de la materia.

c) Alumnos

A diferencia del profesor, la mayoría de los CGI's utilizados son de consulta, a excepción de las tareas. Estos CGI's se encuentran dentro del directorio veri_alumnos (Default). A esta sección sólo podrán

ingresar los alumnos seleccionados en los módulos de Administración previamente. Las secciones de anuncios, políticas y cronogramas utilizan sólo CGI's de consulta.

Tareas.

En esta sección el alumno puede ver todas las tareas enviadas en el semestre, las puede responder e incluso ver las respuestas de las anteriores.

Para responder una tarea, el alumno puede hacerlo como texto o archivo. El CGI verifica si la tarea es respondida en la fecha permitida; según esto, lo ubica en la sección de tareas atrasadas o no.

d) Públicos:

Llamaremos CGI's públicos a aquellos en los que se puede acceder sin necesidad de ingresar alguna clave.

Para este proyecto tenemos CGI's de consulta de datos: Listado de materias, listado de profesores que dictan materias, Programas y cronogramas.

Módulos de PERL usados:

Como vimos en los puntos anteriores, los módulos de Perl usados son básicamente los siguientes:

- Ingreso
- Modificación
- Consulta
- Validación de datos
- Eliminación

3.3. API Html

Debido a la necesidad imperiosa de siempre mejorar la interfaz del sistema con el usuario, se ha provisto al administrador de una herramienta que facilita el mantenimiento de todas las páginas del sistema. Esta consiste de un conjunto de funciones agrupadas dentro de un API (Application Programming Interface) llamado "METAFORMATO" que permite la configuración de la apariencia de las páginas WEB a través un conjunto de archivos (.ftml) que guardan las características de las mismas. Por ejemplo si se desea cambiar el

fondo (Background) sólo habrá que cambiar en el archivo respectivo esta característica, para que el cambio se dé.

3.4. Base de Datos

3.4.1. Justificación de su uso.

Al ser diseñado el esquema de cómo funcionaría el sistema de Administración de Cursos, fue evidente la necesidad de utilizar una herramienta de almacenamiento de datos en forma estructurada. Había una herramienta que estaba incluida en el sistema operativo Linux que podía ser utilizada por el proyecto. Como la herramienta se trataba de una base de datos (Postgres) la cual tenía características muy importantes como el manejo de todas las estructuras de consulta (SQL), de disparadores, máscaras, etc., adicionalmente se tenía el código fuente abierto (disponible gratuitamente para el público), por lo que el uso de la misma en el proyecto no involucraría ningún costo adicional, entonces era hasta cierto punto un desperdicio no utilizarla.

Además toda la información sobre la misma se podía obtener gratuitamente desde el sitio web del software (<http://www.postgres.org>),

por lo que eso nos facilitaría mucho nuestra labor investigativa en la obtención de conocimientos para realizar un apropiado uso de la misma y optimizar el funcionamiento de nuestro sistema.

Dentro de lo que era el diseño de nuestro sistema, se había analizado la posibilidad de que toda la información sea almacenada y accesada dentro de archivos planos; pero había un gran problema: uno de los factores más importantes que debían ser tomados en cuenta en nuestro sistema (por tratarse de una intranet) era la seguridad. Por lo tanto, era necesario tener una herramienta que nos ayude a ejercer medidas de control en lo que se refiere al acceso de los futuros usuarios al sistema, y específicamente al manejo de la información por medio de archivos planos, método que no nos proporcionaría un adecuado control sobre todos los datos que iba a manejar nuestro sistema. Además, un archivo de datos era mucho más fácil de destruir que una base de datos protegida con sus debidas seguridades.

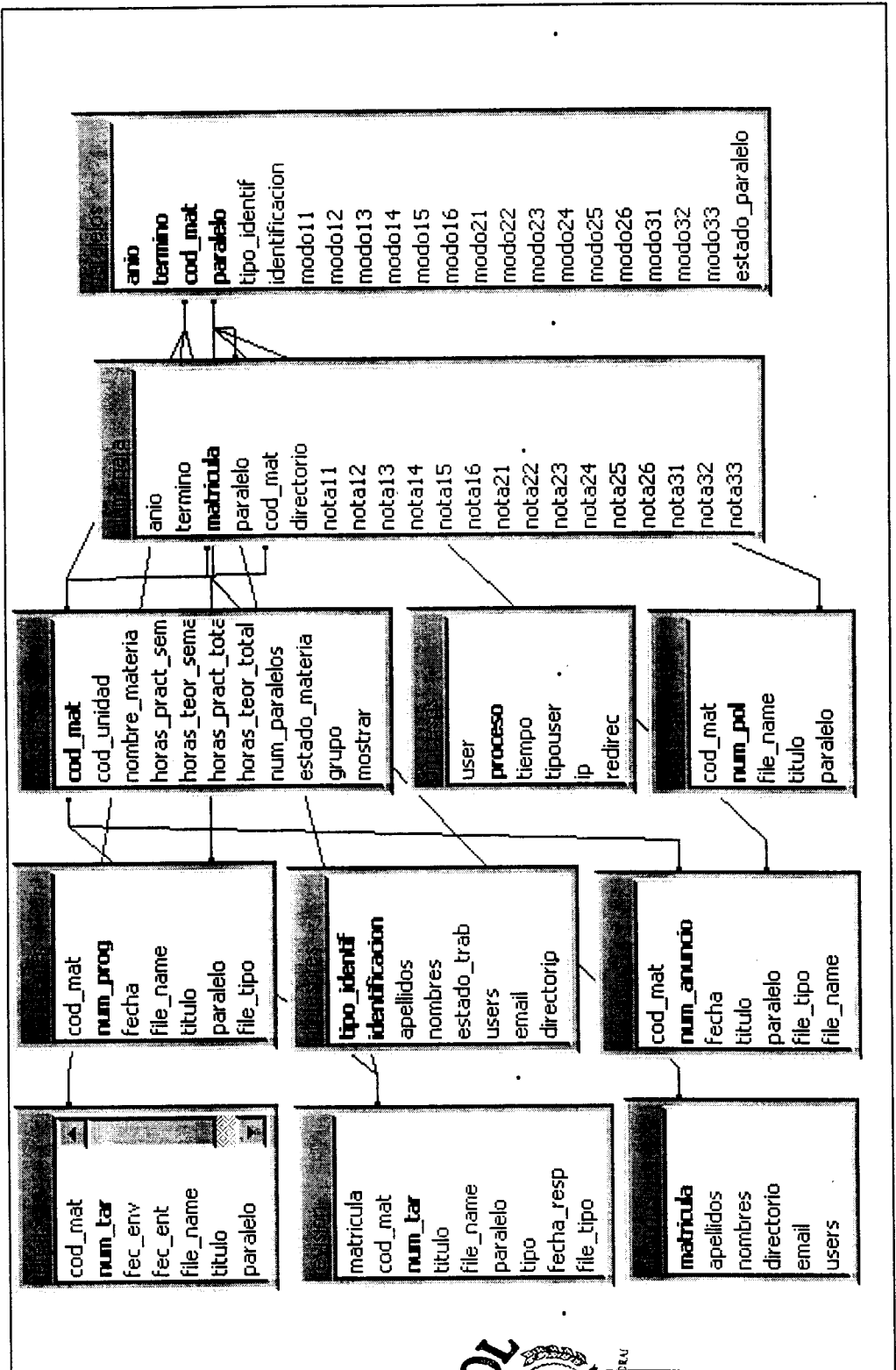
Otro inconveniente que se presentaba, era que el sistema iba a trabajar con archivos de diferente índole que iban a ser creados, manejados y actualizados por los usuarios finales del sistema, lo que involucraba que ya en operaciones de tiempo real, nuestro sistema iba

a ser invadido por diferentes tipos de archivos, lo que a su vez desembocaría en severos problemas con la capacidad de almacenamiento de los discos, y cuando estos problemas se presentaran, iba a ser muy difícil determinar qué archivos o qué información podía ser eliminada y cuál no.

Otro factor muy importante que fue tomado en cuenta, fue que podíamos utilizar todas las ventajas que nos proporciona una base de datos, la cual permite crear y desarrollar métodos de acceso y de consulta de la información de una manera sencilla y estructurada, de esta manera nuestro problema de codificación para la lectura, actualización y escritura de archivos planos se reducía a hallar una interface entre nuestra aplicación y la base de datos, la cual fácilmente se la podía obtener en Internet y cuya implementación era muy sencilla.

Todos los factores detallados anteriormente indican la importancia de utilizar una base de datos dentro del diseño de nuestro proyecto, y que sin la presencia de ésta hubiera sido ardua la tarea de implementación del mismo.

3.4.2. Modelo Entidad-Relación



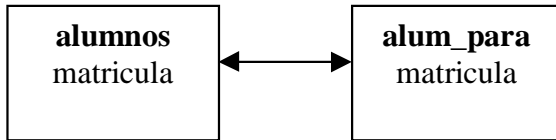
3.4.3. Estructura de sus entidades (tablas)

Tabla: alum_para.

Columnas:

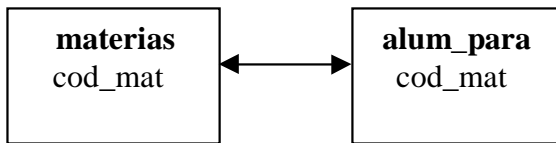
Nombre	Tipo	Tamaño
Anio	Texto	4
Termino	Texto	1
Matricula	Texto	13
Paralelo	Texto	3
cod_mat	Texto	10
Directorio	Texto	1
nota11	Texto	20
nota12	Texto	20
nota13	Texto	20
nota14	Texto	20
nota15	Texto	20
nota16	Texto	20
nota21	Texto	20
nota22	Texto	20
nota23	Texto	20
nota24	Texto	20
nota25	Texto	20
nota26	Texto	20
nota31	Texto	20
nota32	Texto	20
nota33	Texto	20

Relaciones:



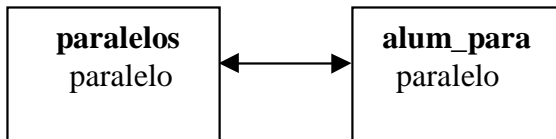
Atributos: Único, No exigir

Atributos: Uno a uno



Atributos: No exigir

Atributos: Uno a varios



Atributos: No exigir

Atributos: Indeterminado

Índices de tabla:

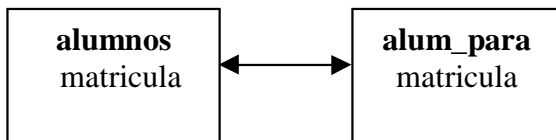
Nombre		Número de campos
alum_para	→ cod_mat	1
Campos:	cod_mat, Ascendente	
alum_para	→ paralelo	1
Campos:	paralelo, Ascendente	
PrimaryKey		1
Campos:	matricula, Ascendente	

Tabla: alumnos.

Columnas:

Nombre	Tipo	Tamaño
Matricula	Texto	13
Apellidos	Texto	40
Nombres	Texto	40
Directorio	Texto	1
Email	Texto	40
Users	Texto	8

Relaciones



Atributos: Único, No exigir

Atributos: Uno a uno

Índices de tabla

Nombre

PrimaryKey

Campos: matricula, Ascendente

Número de campos

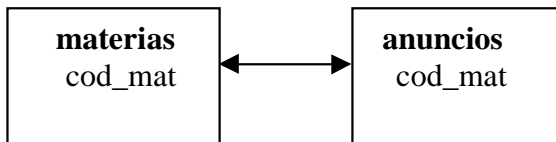
1

Tabla: anuncios.

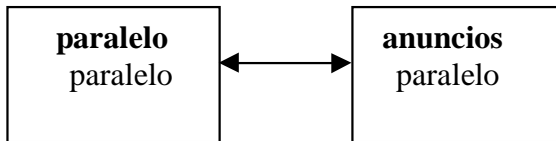
Columnas:

Nombre	Tipo	Tamaño
Cod_mat	Texto	10
Num_anuncio	Número (entero)	2
Fecha	Fecha/Hora	8
Titulo	Texto	40
Paralelo	Texto	3
File_tipo	Texto	1
File_name	Texto	40

Relaciones



Atributos: Único, No exigir
Atributos: Uno a uno



Atributos: No exigir
Atributos: Indeterminado

Índices de tabla

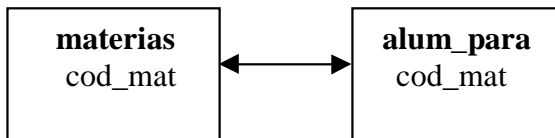
Nombre		Número de campos
Anuncios	→ cod_mat	1
Campos:	cod_mat, Ascendente	
Anuncios	→ paralelo	1
Campos:	paralelo, Ascendente	
PrimaryKey		1
Campos:	num_anuncio, Ascendente	

Tabla: materias.

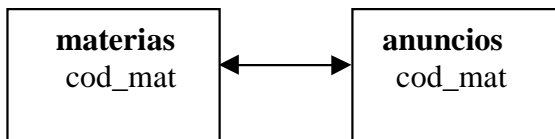
Columnas:

Nombre	Tipo	Tamaño
cod_mat	Texto	10
cod_unidad	Texto	5
nombre_materia	Texto	80
horas_pract_seman	Número (entero)	2
horas_teor_seman	Número (entero)	2
horas_pract_total	Número (entero)	2
horas_teor_total	Número (entero)	2
num_paralelos	Número (entero)	2
estado_materia	Texto	50
Grupo	Número (largo)	4
Mostrar	Texto	1

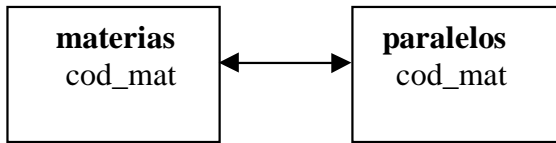
Relaciones:



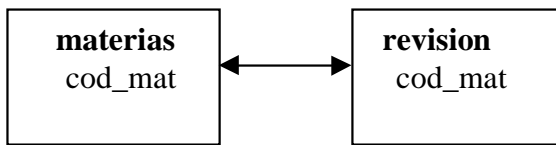
Atributos: No exigir
Atributos: Uno a varios



Atributos: Único, No exigir
Atributos: Uno a uno



Atributos: No exigir
 Atributos: Uno a varios



Atributos: No exigir
 Atributos: Uno a varios

Índices de tabla:

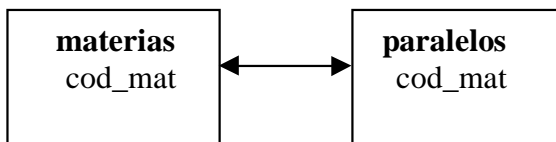
Nombre	Número de campos
PrimaryKey	1
Campos: cod_mat, Ascendente	

Tabla: paralelos.

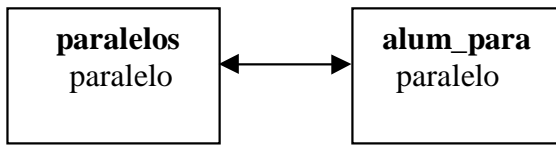
Columnas:

Nombre	Tipo	Tamaño
Anio	Texto	4
Termino	Texto	1
cod_mat	Texto	10
Paralelo	Texto	3
tipo_identif	Texto	3
Identificacion	Texto	13
modo11	Texto	20
modo12	Texto	20
modo13	Texto	20
modo14	Texto	20
modo15	Texto	20
modo16	Texto	20
modo21	Texto	20
modo22	Texto	20
modo23	Texto	20
modo24	Texto	20
modo25	Texto	20
modo26	Texto	20
modo31	Texto	20
modo32	Texto	20
modo33	Texto	20
estado_paralelo	Texto	1

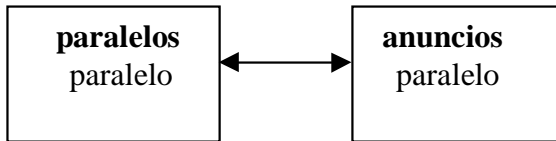
Relaciones:



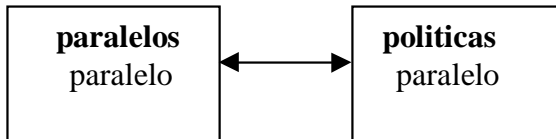
Atributos: No exigir
Atributos: Uno a varios



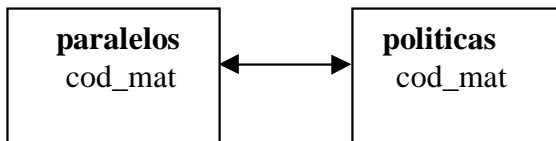
Atributos: Indeterminado
Atributos: No exigir



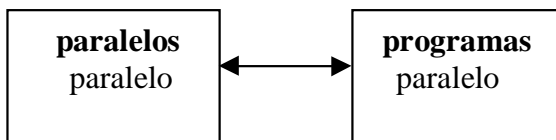
Atributos: Indeterminado
Atributos: No exigir



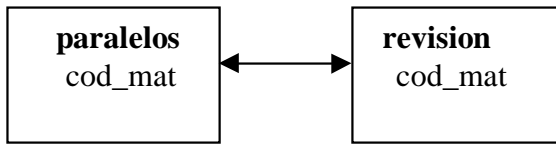
Atributos: No exigir
Atributos: Indeterminado



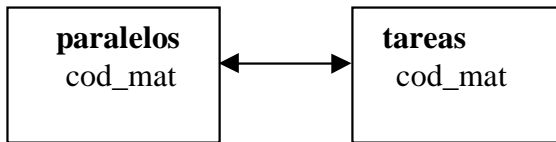
Atributos: No exigir
Atributos: Indeterminado



Atributos: Indeterminado
Atributos: No exigir



Atributos: Indeterminado
 Atributos: No exigir



Atributos: No exigir
 Atributos: Indeterminado

Índices de tabla:

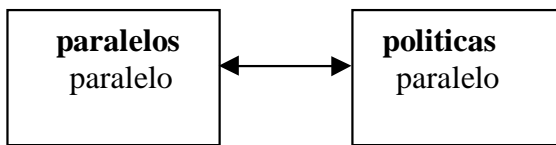
Nombre	Número de campos
identificacion	1
Campos: identificacion, Ascendente	
Paralelos —————> cod_mat	1
Campos: cod_mat, Ascendente	
Paralelos —————> paralelo	1
Campos: paralelo, Ascendente	
PrimaryKey	4
Campos: anio, Ascendente	
termino, Ascendente	
cod_mat, Ascendente	
paralelo, Ascendente	

Tabla: políticas.

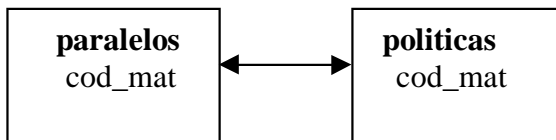
Columnas:

Nombre	Tipo	Tamaño
cod_mat	Texto	10
num_pol	Número (entero)	2
file_name	Texto	40
Título	Texto	40
Paralelo	Texto	3

Relaciones:



Atributos: No exigir
Atributos: Indeterminado



Atributos: No exigir
Atributos: Indeterminado

Índices de tabla:

Nombre		Número de campos
políticas	→ cod_mat	1
Campos:	cod_mat, Ascendente	
políticas	→ paralelo	1
Campos:	paralelo, Ascendente	
PrimaryKey		1
Campos:	num_pol, Ascendente	

Tabla: procesos.

Columnas:

Nombre	Tipo	Tamaño
User	Texto	8
Proceso	Número (doble)	8
Tiempo	Número (doble)	8
Tipouser	Texto	8
Ip	Texto	8
Redirec	Texto	1

Índices de tabla

Nombre	Número de campos
PrimaryKey	1
Campos:	proceso, Ascendente

Tabla: profesores.

Columnas:

Nombre	Tipo	Tamaño
tipo_identif	Texto	3
Identificacion	Texto	13
Apellidos	Texto	40
Nombres	Texto	40
estado_trab	Texto	1
Users	Texto	8
Email	Texto	50
Directorio	Texto	1

Índices de tabla:

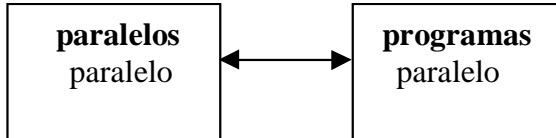
Nombre	Número de campos
identificacion	1
Campos: identificacion, Ascendente	
PrimaryKey	2
Campos: tipo_identif, Ascendente identificacion, Ascendente	
tipo_identif	1
Campos: tipo_identif, Ascendente	

Tabla: programas.

Columnas:

Nombre	Tipo	Tamaño
cod_mat	Texto	10
num_prog	Número (entero)	2
Fecha	Fecha/Hora	8
file_name	Texto	40
Titulo	Texto	40
Paralelo	Texto	3
file_tipo	Texto	1

Relaciones:



Atributos: No exigir
Atributos: Indeterminado

Índices de tabla:

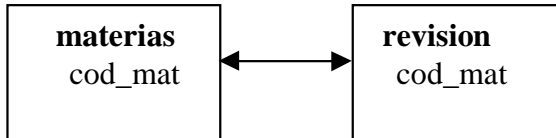
Nombre		Número de campos
PrimaryKey		1
Campos:	num_prog, Ascendente	
programas	→ paralelo	1
Campos:	paralelo, Ascendente	

Tabla: revision.

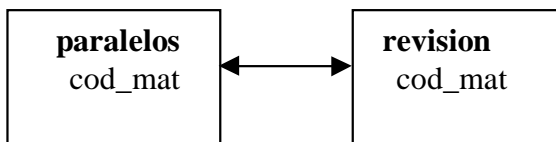
Columnas:

Nombre	Tipo	Tamaño
Matricula	Texto	13
cod_mat	Texto	10
num_tar	Número (entero)	2
Titulo	Texto	20
file_name	Texto	60
Paralelo	Texto	3
Tipo	Texto	1
fecha_resp	Fecha/Hora	8
file_tipo	Texto	1

Relaciones:



Atributos: No exigir
Atributos: Uno a varios



Atributos: No exigir
Atributos: Indeterminado

Índices de tabla:

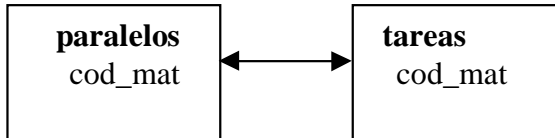
Nombre		Número de campos
PrimaryKey		1
Campos:	num_tar, Ascendente	
Revision	→ cod_mat	1
Campos:	cod_mat, Ascendente	
Revision	→ matricula	1
Campos:	matricula, Ascendente	

Tabla: tareas.

Columnas:

Nombre	Tipo	Tamaño
cod_mat	Texto	10
num_tar	Número (largo)	4
fec_env	Fecha/Hora	8
fec_ent	Fecha/Hora	8
file_name	Texto	40
Título	Texto	40
Paralelo	Texto	3
file_tipo	Texto	1

Relaciones:



Atributos: No exigir
Atributos: Indeterminado

Índices de tabla:

Nombre		Número de campos
PrimaryKey		1
Campos:	num_tar, Ascendente	
Tareas	→ cod_mat	1
Campos:	cod_mat, Ascendente	

3.4.4. Acoplamiento con el sistema de asignación de recursos.

Existe otro proyecto llamado “Asignación de recursos” que utiliza la misma base de datos que el presente sistema. “Asignación de recursos” es el encargado de migrar los datos desde la Base de la ESPOL hacia Postgres que es la utilizada por ambos proyectos. Para ingresar, modificar o eliminar datos se deberá usar el sistema de Asignación de recursos.

Para integrar este sistema con el de “Asignación de recursos”, se tomaron en cuenta varios puntos:

- **Base de Datos:** El levantamiento de la Base se la realizó en conjunto. De esta manera se evita el tener campos o tablas redundantes entre los proyectos.

Al final de este punto se detallan las tablas que no son utilizadas por este sistema.

- **Ingreso de Datos a la Base:** Para el ingreso o modificación de datos de alumnos o profesores, se utilizan las pantallas de mantenimiento del otro proyecto.

Las tablas de las que detallaremos su estructura a continuación no son usadas por nuestro sistema, pero forman parte de la base de datos cursos. Estas tablas fueron creadas para ser usadas por el proyecto de Administración de recursos, por lo que fue necesario introducir esta información.

Tabla: cola.

Columnas:

Nombre	Tipo	Tamaño
ts_pedido	Texto	19
Dip	Texto	15
cod_tipo_recu	Texto	4
cod_recurso	Texto	8
cod_mat	Texto	10
Paralelo	Texto	3
Estado	Texto	1
ts_aprobado	Texto	19

Tabla: horario_aprobado.

Columnas:

Nombre	Tipo	Tamaño
cod_tipo_recu	Texto	4
cod_recurso	Texto	8
cod_mat	Texto	10
Paralelo	Texto	3
Dia	Texto	3
Semana	Número (entero)	2
hora_inicial	Fecha/Hora	8
hora_final	Fecha/Hora	8
ts_pedido	Texto	19
ts_aprobado	Texto	19

Tabla: horario_pedido.

Columnas:

Nombre	Tipo	Tamaño
cod_tipo_recu	Texto	4
cod_recurso	Texto	8
cod_mat	Texto	10
Paralelo	Texto	3
Dia	Texto	3
Semana	Número (entero)	2
hora_inicial	Fecha/Hora	8
hora_final	Fecha/Hora	8
ts_pedido	Texto	19

Tabla: parámetros.

Columnas:

Nombre	Tipo	Tamaño
cod_parametro	Texto	7
Descripcion	Texto	40
par_caracteres	Texto	50
par_numero	Número (largo)	4
par_fecha	Fecha/Hora	8
par_hora	Fecha/Hora	8
estado_parametro	Texto	1

Tabla: recursos.

Columnas:

Nombre	Tipo	Tamaño
cod_tipo_recu	Texto	4
cod_recurso	Texto	8
nombre_recurso	Texto	60
estado_recurso	Texto	1

Tabla: secuencia_cod.

Columnas:

Nombre	Tipo	Tamaño
Id	Texto	40
Numero	Texto	8
Estado_numero	Texto	1

Tabla: tipo_recu.

Columnas:

Nombre	Tipo	Tamaño
Cod_tipo_recu	Texto	4
Descripción_tipo	Texto	40
Estado_tipo_recu	Texto	1

Tabla: unidad_académica.

Columnas:

Nombre	Tipo	Tamaño
Cod_unidad	Texto	5
Nombre_unidad	Texto	60
Estado_unidad	Texto	1

3.5. Organización de Directorios

Para garantizar el acceso a las opciones respectivas para los alumnos, profesores o administrador, se ubicaron los CGI's en diferentes directorios, de esta manera sólo las personas autorizadas podrán ingresar a estos.

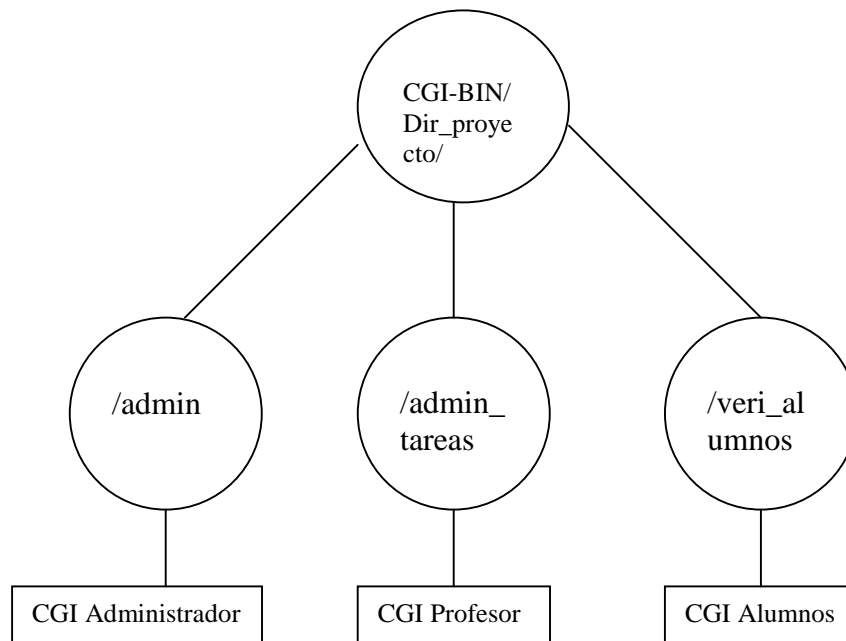
Al especificar cómo están organizados los directorios, podemos dividirlos en dos grandes grupos:

CGI's y HTML's

CGI's:

Se encuentran dentro del directorio CGI-BIN del Sistema Operativo.

CGI-BIN/directorio del proyecto/



Dentro de éste se encuentran todos los CGI's de administración, profesores, alumnos y públicos.

Si no se ha cambiado las rutas predeterminadas en la instalación, el esquema quedaría de la siguiente manera:

Dentro de la ruta anterior:

Administración: /admin/

Profesores: /admin_tareas/

Alumnos: /veri_alumnos/

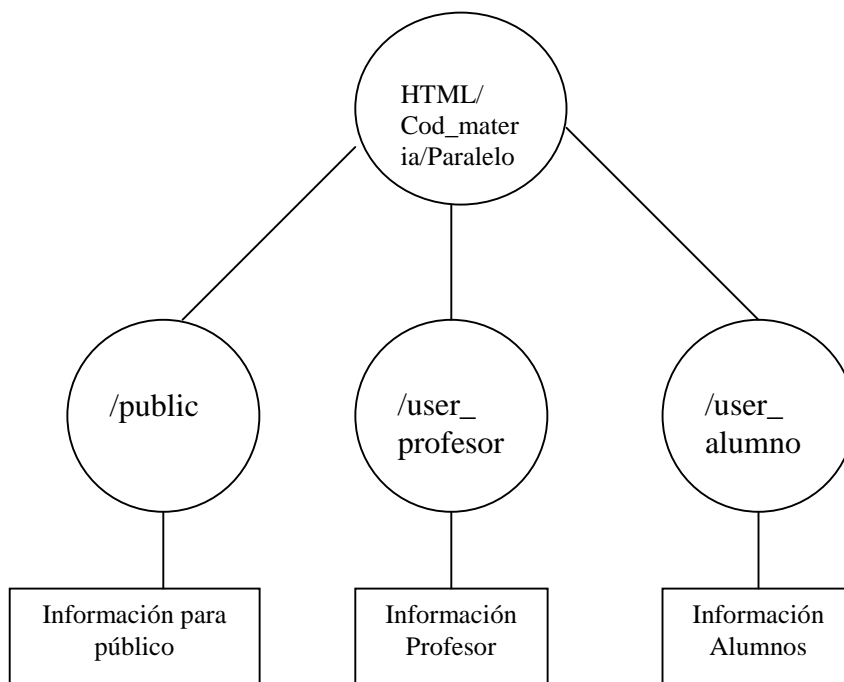
Públicos: /

Nota: Agregar la ruta default de los CGI's

HTML's:

En este directorio se encontrarán todos los archivos o textos enviados tanto por los alumnos como por los profesores en las diferentes opciones del sistema.

Los archivos serán agrupados dentro del directorio default html del sistema operativo/ (Ej. /home/html/www/)



A continuación: /código de materia/paralelo/

Público: En donde se encuentra la información que puede ver todo el mundo

/public/

Profesor: /user_profesor/

Alumnos: /user_alumno/

3.6. Manejo de Autenticación y Seguridad

Autenticación:

Se utiliza POP3 para validar el user y password del usuario, esto se lo hace con otro servidor que no es necesariamente el “Servidor Web”.

Si estos datos no son los correctos, el usuario no podrá ingresar al sistema.

Tiempo de Vida del usuario y cookies:

Se utilizaron los cookies para restringir el acceso a directorios, debido a que el usuario puede ver rutas o parámetros en la dirección del URL.

Para evitar esto, los parámetros que indican alguna información de materia, profesor o alumno, se los graba en cookies.

Se utilizó un esquema de tiempo de vida, es decir, si el visitante pasa mucho tiempo sin navegar, su clave expira. Cuando quiera seguir navegando después del tiempo asignado, le aparecerá en pantalla un mensaje indicándole que *“su tiempo ha terminado”*, de tal manera que debe ingresar nuevamente su usuario y clave.

Para validar el tiempo de vida o de uso de la página, se utilizan cookies, éstas guardan información del número de proceso y el usuario, además se utiliza una tabla en la base de datos que contiene el nombre, la dirección ip de la máquina que esté usando, el tiempo de vida, el número de proceso del usuario.

Al expirar el tiempo de uso o navegación del usuario, se pidió que se regrese a la página en donde se encontraba antes de que suceda la expiración. Esto no se pudo implementar debido a que en muchas páginas se pasan parámetros para que ésta aparezca, y éstos no son almacenados en ninguna parte.

4. SERVIDOR Y CLIENTE DE NOTAS

4.1. Servidor y Clientes original.

El Servidor y Cliente de Notas fueron implementados en JAVA por un tópico anterior, y se lo agregó al presente proyecto para que la administración de cursos sea completa e incluya las notas de los estudiantes.

El Servidor se encarga de recibir los requerimientos del cliente, consultar con la Base de Datos, y enviar de vuelta (al browser) la información solicitada.

Para mayor información tomar como referencia la Documentación del “Sistema integrado de Notas” del tópico anterior(1998) del Ing. Guido Caicedo que detalla el diseño original.

El Servidor y el Cliente Notas se encontraban inicialmente implementados para funcionar bajo la plataforma Windows NT y teniendo como base de datos a Acces de Microsoft; por lo que hubo que realizar varios cambios para que se acople y funcione con el presente proyecto. Estos cambios se verán en la presente sección.

4.2. Requisitos para la integración del nuevo sistema

- JDK: Se debe instalar JDK (Java Development Kit) versión 1.1.7 mínimo para que el servidor funcione.
- JDBC: Se necesita tener instalado el JDBC (Java Database Connection) respectivo para que trabajen las consultas a la base de datos Postgres y que a su vez tenga a linux como plataforma.

4.3. Modificaciones al Servidor Original

Los cambios hechos al servidor original fueron los siguientes:

Cambio de Base y sistema operativo:

Para este nuevo proyecto se utilizó un nuevo sistema operativo "Linux"; además que se cambió la Base de Datos, anteriormente era Access y ahora es Postgres.

Accesos a la Base:

Se cambiaron todos los accesos a la base usando el driver JDBC para poder conectarse. Originalmente se utilizaba el driver de ODBC para Acces debido a que funcionaba en Windows NT.

Parametrización:

El sistema anterior no se encontraba parametrizado, es decir, por cada función que accedía a la base de datos, se efectuaba la conexión; ahora, basta con cambiar en el archivo de configuración los diferentes parámetros para que todas las funciones lo lean. Por ejemplo:

User y Password de la Base de Datos para la conexión, driver JDBC, etc.

SQL:

Por utilizar una nueva base, los campos de las tablas cambiaron, por lo que hubo que modificar los sql originales.

4.4. Modificaciones al cliente original

Los cambios al cliente original fueron los siguientes:

Conexión:

En el sistema anterior una vez que se ingresaba al sistema había que cargar un applet y de allí presionar un botón para conectarse

al servidor. Esto quedó eliminado ya que ahora al hacer un click en la opción de notas del menú, automáticamente se conecta al servidor.

Autenticación y autentificación:

Anteriormente al conectarse al sistema éste pedía user y password. En la actualidad en nuestro sistema se obvió esto ya que para poder entrar a estas opciones, el usuario previamente debió haber sido verificado por las seguridades del nuevo sistema, caso contrario no hubiera podido acceder al applet, ni enviar los parámetros necesarios.

Ventanas:

Cada vez que se escogía una opción, se abría una nueva ventana, llegando a un punto en que se tenían hasta 5 applets cada uno en una ventana diferente. Ahora todo se hace en la misma ventana del browser.

PRINCIPAL ANUNCIOS TAREAS POLITICAS CRONOGRAMA NOTAS LOGOUT

ANALS.DE REDES ELECT.I Paralelo:001

Profesor: FLORES MACIAS JORGE GUILLERMO ING.
[mail:](#)

NOTAS
Opciones del Profesor

Profesor: JORGE GUILLERMO FLORES MACIAS

Materia: ANALS.DE REDES ELECT.I

Paralelo: 1 Término: 2 Año: 1999

Modo de Calificar Modificar Notas Estadísticas

Interfaces:

Se cambiaron las etiquetas, botones, etc.

Desaparecieron los checkboxes en donde aparecían todas las materias. Ahora sólo se ve la materia en la cual se encuentra el alumno o profesor.

Errores de cálculo:

Para la sección de estadísticas del curso, los cálculos estaban mal hechos para algunos parciales debido a que no se multiplicaba por el porcentaje.

Fallas al cambiar de versión de browser:

El proyecto anterior no funcionaba en IE4, por lo que hubo que realizar modificaciones para cambiar el envío de parámetros desde el cliente al servidor. El signo de interrogación que se utilizaba antes para separarlos, el IE4 lo transformaba en un número aleatorio, al hacer esto, el java del servidor no comprendía que se le estaba enviando y no funcionaba correctamente.

Se eliminó:

- Crear curso: Nuestro sistema no necesita crear cursos, esto lo realiza el sistema de asignación de recursos.
- Cambiar Password: Esta sección es propia del administrador del laboratorio de computación de la ESPOL.
- Ingreso de materias: Las materias son ingresadas por el personal de la ESPOL en su base de datos, al migrar los datos a Postgres, las materias deberán ser las mismas.
- Encuestas: Se la eliminó porque no se le encontró una utilidad relevante en el presente sistema.

Mensajes de error:

Se colocaron nuevos mensajes de error y se hicieron nuevas validaciones.

No se pueden ingresar valores mayores a los permitidos en las notas de los estudiantes.

5. DISEÑO DE LA INTERFACE CON EL USUARIO

5.1. MODOS DE INTERACCION

El sistema interactúa con el usuario a través de los diferentes browsers, los mismos que nos permitirán la captura y entrega de información. Otros modos de interacción son los dispositivos de entrada y salida como la pantalla y el “mouse” o “ratón” del computador. El usuario para poder seleccionar la información necesaria o interactuar con las opciones disponibles deberá utilizar el mouse. La pantalla será el dispositivo que permita visualizar el resultado de los procesos que sean ejecutados en la interacción entre el sistema y el usuario.

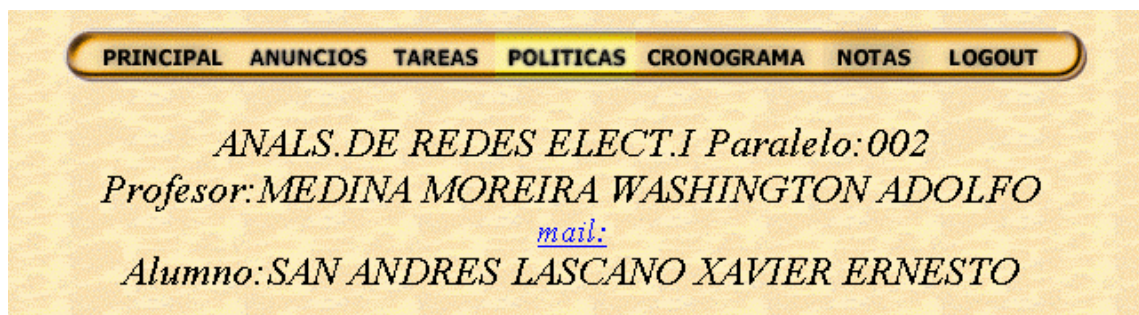
5.2. PLANTILLA DEL SITIO

➤ Cabecera Horizontal



➤ Cabecera Central:

- Barra de Botones
- Nombre de la Materia y número del paralelo
- Nombre del profesor
- Email del profesor
- Nombre del usuario en el caso de ser alumno



➤ Título de la página:

Título que indica de manera muy concreta la página donde el usuario se encuentra ubicado. Por ejemplo:

Políticas del Curso

➤ Contenido de la página:

Herramientas de la página que el usuario va a utilizar. Por ejemplo:



➤ Pie de Página:

Enlaces en Modo texto que cumple las funciones de la barra de botones (Sección 5.4.3).

[\[PRINCIPAL\]](#) [\[ANUNCIOS\]](#) [\[TAREAS\]](#) [\[POLITICAS\]](#) [\[CRONOGRAMA\]](#) [\[NOTAS\]](#) [\[LOGOUT\]](#)

5.3. FACILIDADES DE NAVEGACION

- En todas las páginas existe la barra de botones que con un “click” en uno de ellos puedo pasar a las distintas páginas principales del sistema.
- En la parte final de cada página existe el “*pie de página*”, que realiza las funciones de la barra de botones, de tal forma que si el usuario se encuentra en la parte final de la página no tiene que desplazarse hacia el Inicio para utilizar la barra de botones.
- En todas las páginas hay vínculos que permiten al usuario regresar a los enlaces usualmente requeridos.
- En las páginas principales se encuentran las direcciones de e-mail de profesores y alumnos, de tal manera que el usuario puede enviar mails desde cualquier página del sistema.

5.4. ESTRUCTURA DE LAS PANTALLAS

5.4.1. Cabecera Horizontal.-

En el área superior de la página se encuentra un logotipo de la Escuela Superior Politécnica del Litoral (ESPOL) y otro de la Facultad de Ingeniería en Electricidad y Computación de la ESPOL (FIEC). El usuario podrá posicionarse con el uso del mouse sobre los logotipos y haciendo un click sobre éstos podrá ir a los Web sites respectivos.(ESPOL o la FIEC).

5.4.2. Cabecera Central

- Nombre de la materia y número del Paralelo
- Nombre del profesor
- E-mail del profesor
- En el caso de que el usuario sea el alumno: Nombre del alumno

5.4.3. Barra de botones

El usuario puede utilizar la barra de botones para obtener acceso a los recursos que utiliza con más frecuencia con tan solo hacer click. En la barra se presentan los siguientes botones:

➤ **Botón Anuncios.-**

Mediante un click sobre este botón, el usuario público o el usuario alumno tendrá la opción de ingresar a las páginas que le permiten visualizar los anuncios enviados por un profesor.

En cambio, el usuario profesor, podrá ingresar a las páginas de “administración de anuncios.

➤ **Botón Tareas**

Mediante un click sobre este botón, el profesor podrá tener acceso al área de “administración de tareas”. Allí podrá ingresar el formulario de una tarea a ser realizada por sus alumnos para una fecha determinada, ya sea la tarea de tipo texto o archivo. Una vez publicada la tarea a ser elaborada, podrá modificarla o eliminarla en caso de ser necesario.

Esta opción, le permitirá a todos los alumnos de dicho paralelo, al mismo tiempo, conocer sobre la tarea a ser cumplida, para luego proceder a responder dicha tarea. Sus respuestas podrán ser de

tipo texto o archivo. Todo este proceso de envío y respuestas de tareas logra una mayor comunicación entre el profesor y su alumno, a la vez que facilita el envío, recepción, revisión y control de tareas enviadas, así como también servirá como referencia de temas tratados en clase.

➤ ***Botón Cronograma***

Mediante un click sobre este botón, el usuario profesor podrá tener acceso al área de “administración de programas”. Allí podrá ingresar el programa del curso a ser cubierto en el presente semestre, para publicarlo en su hoja WEB, de esta manera podrá informar a sus alumnos o al público en general, los temas que van a ser revisados durante el curso.

Los alumnos o el público con esta opción, podrán acceder a revisar los cronogramas enviados por el profesor.

➤ **Botón Políticas**

Mediante un click sobre este botón, el profesor tendrá acceso al área de “administración de políticas”. Allí podrá ingresar las políticas del curso, para publicarlos en su hoja WEB, de tal forma que podrá informar a sus alumnos o al público en general, las políticas o normas a ser seguidas durante el curso (como por ejemplo, el número máximo de faltas permitidas, control de asistencia a clases del profesor y ayudantías, requerimientos mínimos en el formato de tareas o proyectos, etc.).

Los alumnos o público en general podrán leer las políticas enviadas por el profesor.

➤ **Botón Notas**

Mediante un click sobre este botón, el profesor tendrá acceso al área de “manejo de notas”. Allí podrá establecer la forma de calificar, estableciendo porcentajes para los aportes, lecciones y tareas, hasta completar el máximo puntaje que es actualmente de

100 puntos. Así también, podrá consultar las notas de cada alumno, y los cuadros estadísticos de los totales de las notas del paralelo, para conocer la mediana y visualizar el nivel de aprovechamiento y deficiencia de su paralelo durante todo el semestre.

Los alumnos podrán consultar sus notas, los cuadros estadísticos de las notas totales del paralelo, y conocer la mediana.

➤ ***Botón Logout***

Mediante un click sobre este botón, el usuario se desconecta del sistema, volviendo a la página principal.

6. Conclusiones y Recomendaciones.

- A diferencia de los sistemas convencionales, el sistema de Administración de Cursos utiliza el Internet como medio de comunicación, lo cual reduce los costos totales de inversión, tanto para la ESPOL como para las personas que van a usar el sistema, debido a que éste no necesita una máquina muy poderosa para funcionar, la gran mayoría del software y herramientas de la plataforma sobre la cual funciona son gratis (servidor de web, manejador de base de datos, etc.) y además lo único que necesita un cliente para usarlo es un browser.

- El Sistema de Administración de Cursos contribuye al desarrollo de la FIEC, modernizando la interacción alumno-profesor y viceversa.

- Además:
 - ❖ Ayuda al profesor en la administración de su curso
 - ❖ Centraliza y organiza la información académica del curso
 - ❖ Proporciona privacidad de la información individual ya sea de alumnos o profesores.
 - ❖ Promueve la creación de páginas Web

- ❖ Como recomendación para futuros proyectos, que como el Sistema de Administración de Cursos, tengan que integrarse con otros sistemas decimos que antes de empezar la implementación , se debe hacer un análisis de todos los recursos que se van a compartir.
- ❖ Otra recomendación fundamental para nuevos sistemas es que estos deben ser altamente configurables , es decir sus parámetros y datos importantes puedan ser cambiados lo más fácil y rápido posible, ya que esto no solo beneficia al usuario final , sino también a los creadores de dichos sistemas.

Bibliografía

- ❖ Perl 5 by Example – David Medinets – Editora QUE – Octubre de 1996.
- ❖ Special Edition Using Perl 5 for Web Programming - David Harlan, Paul Doyle, Micheal O Foghlu, Shelley Powers - Editora QUE - September 1, 1996.
- ❖ Java Developer's Guide - Jamie Jaworski - Editora Sams@net - July 1996
- ❖ Red Hat Linux – <http://www.redhat.com>.
- ❖ PostgreSQL Administrator Guide – <http://www.postgres.org>.
- ❖ PostgreSQL User Guide – <http://www.postgres.org>.
- ❖ PostgreSQL Programmer Guide – <http://www.postgres.org>.
- ❖ PostgreSQL Tutorial – <http://www.postgres.org>.