



UNIVERSIDAD NACIONAL  
AUTÓNOMA DE  
MÉXICO



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES ARAGÓN

# “SISTEMA DE ADMINISTRACIÓN PARA LA INSCRIPCIÓN DE LOS CURSOS EXTRACURRICULARES DE LA FES ARAGÓN”

BAJO LA MODALIDAD DE:

DESARROLLO DE UN CASO PRÁCTICO

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO EN COMPUTACIÓN

**PRESENTA:**

**JUAN CARLOS VILLEGAS GONZÁLEZ**

ASESORA: DRA. NELLY RIGAUD TÉLLEZ

MÉXICO

2010



Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

***“Todo hombre al venir a la tierra tiene derecho a ser educado, en retribución contribuir a la educación de su pueblo”***  
*José Martí*

## Agradecimientos

*A mi esposa*

*por su comprensión, confianza, apoyo, respeto y amor en cada paso de nuestras vidas. Por su sentido del humor y esa gracia que adorna con una sonrisa los buenos y malos momentos. Por haber compartido conmigo las circunstancias más difíciles. Pero sobretodo por no permitir que me dejara vencer.*

*A la UNAM*

*por haberme dado los conocimientos y la enseñanza necesaria para poder ser un profesionalista de calidad y entrega.*

*A la UAM*

*por darme la oportunidad de obtener la experiencia para poder desarrollarme profesionalmente y así mejorar día a día.*

*A mis suegros por*

*Acogerme, apoyarme y guiarme como a un hijo con respeto, paciencia y comprensión.*

*A la Dra. Nelly Rigaud T.*

*por su tiempo, su interés y la confianza depositada en mi para llevar a cabo este y otros proyectos.*

*Al Mtro. Marcelo Pérez M.*

*por todas la atenciones y oportunidades que me brindó.*

*A mis amigos Daniel y Omar*

*por haberme apoyado desinteresadamente en momentos difíciles.*

*A mis padres*

*por la vida, sus atenciones y cuidados.*

*A toda la gente que fue parte de mi formación.*

## ÍNDICE

INTRODUCCIÓN.....	XIII
JUSTIFICACIÓN.....	XIV
<b>CAPÍTULO 1 ANÁLISIS PARA LA CREACIÓN DEL SISTEMA DE INSCRIPCIÓN.....</b>	<b>1</b>
1.1 OBJETIVO GENERAL.....	2
1.2 OBJETIVOS ESPECÍFICOS.....	2
1.3 CALENDARIO DE ACTIVIDADES.....	3
1.4 OBTENCIÓN DE REQUISITOS.....	4
1.5 DEFINICIÓN DEL SISTEMA.....	4
1.6 ALCANCE.....	4
1.7 AMBIENTE DE TRABAJO.....	4
1.8 PROBLEMÁTICA.....	5
1.9 REQUERIMIENTOS DE LA BASE DE DATOS.....	5
1.10 SELECCIÓN DE SOFTWARE.....	6
1.10.1 Análisis de los Lenguajes de Programación.....	6
1.10.2 Selección del Lenguaje de Programación.....	9
1.10.3 Análisis de los Manejadores de las Bases de Datos.....	9
1.10.4 Selección del Manejador de la Base de Datos.....	11
1.10.5 Análisis y Selección de los Distintos Tipos de Tablas.....	12
1.11 SOLUCIÓN PROPUESTA.....	15
1.12 GLOSARIO.....	22
<b>CAPÍTULO 2 DISEÑO DEL SISTEMA DE INSCRIPCIÓN.....</b>	<b>27</b>
2.1 DISEÑO FUNCIONAL.....	28
2.2 DISEÑO DETALLADO.....	29
2.2.1 Módulos Comunes.....	29
2.2.2 Módulos del Alumno.....	30
2.2.3 Módulos del Ponente.....	31
2.2.4 Módulos del Administrador.....	31
2.3 BASE DE DATOS.....	32
2.3.1 Diseño de la Base de Datos.....	33
2.3.2 Análisis de Requisitos.....	34
2.3.3 Diseño Conceptual.....	35
2.3.4 Diseño Lógico.....	38
2.4 ASPECTOS TÉCNICOS.....	39
2.5 ARQUITECTURA DEL SISTEMA.....	41
2.5.1 Capa de Presentación.....	41
2.5.2 Capa de Lógica de Control.....	41
2.5.3 Capa de Acceso a Datos.....	41
2.6 DISEÑO DE PANTALLAS.....	41
2.7 COMPATIBILIDAD.....	47
2.8 SEGURIDAD.....	49
2.8.1 Identificación de Usuario.....	49
2.8.2 Identificación de Tipo de Usuario.....	49
2.8.3 Detección de Código Mal Intencionado.....	49
2.8.4 Uso de Permisos en la Base de Datos.....	50
2.8.5 Registro de Acciones.....	50



<b>CAPÍTULO 3</b>	<b>DESARROLLO DEL SISTEMA .....</b>	<b>51</b>
3.1	ADECUACIÓN DEL ENTORNO DE GENERACIÓN.....	52
3.2	DESARROLLO DE MÓDULOS.....	52
3.3	DESCRIPCIÓN DE FUNCIONES (FUNCTIONS) .....	55
3.3.1	<i>Functions de la Capa de Presentación</i> .....	55
3.3.2	<i>Functions de la Capa de Lógica de Control</i> .....	58
3.3.3	<i>Functions de la Capa de Acceso a Datos</i> .....	62
<b>CAPÍTULO 4</b>	<b>IMPLANTACIÓN Y CONTROL.....</b>	<b>65</b>
4.1	IMPLANTACIÓN.....	66
4.1.1	<i>Instalación del Sistema</i> .....	66
4.1.2	<i>Carga inicial de Datos</i> .....	67
4.1.3	<i>Pruebas de Implantación</i> .....	67
4.1.4	<i>Preparación del Mantenimiento</i> .....	67
4.2	PRUEBAS .....	67
4.2.1	<i>Pruebas Unitarias</i> .....	68
4.2.2	<i>Pruebas de Integración</i> .....	68
4.2.3	<i>Pruebas en Exploradores</i> .....	68
4.2.4	<i>Pruebas de Saturación</i> .....	69
4.3	PRESENTACIÓN Y APROBACIÓN DEL SISTEMA .....	69
4.4	FORMACIÓN DE USUARIOS.....	69
4.4.1	<i>Elaboración del Manual de Usuario</i> .....	69
4.4.2	<i>Formación de Usuarios Finales</i> .....	69
4.5	MANTENIMIENTO .....	70
4.5.1	<i>Registro de Peticiones</i> .....	70
4.5.2	<i>Análisis de Peticiones</i> .....	70
4.5.3	<i>Identificación de los Elementos Afectados</i> .....	70
4.5.4	<i>Pruebas de Regresión</i> .....	70
4.5.5	<i>Implementación de las Modificaciones</i> .....	71
4.5.6	<i>Seguimiento y Evaluación de Cambios</i> .....	71
4.6	CRECIMIENTO DEL SISTEMA.....	71
4.6.1	<i>Modificaciones a el modelo de la Base de Datos</i> .....	71
4.6.2	<i>Módulos Agregados</i> .....	73
4.7	MANUAL DE USUARIO.....	74
4.7.1	<i>Opciones del Ponente</i> .....	74
4.7.2	<i>Opciones del Alumno</i> .....	86
4.7.3	<i>Opciones del Administrador</i> .....	93
<b>CAPÍTULO 5</b>	<b>CONCLUSIONES Y RECOMENDACIONES .....</b>	<b>105</b>
	BIBLIOGRAFÍA .....	109

## ÍNDICE DE TABLAS

TABLA 1-1	CALENDARIO DE ACTIVIDADES .....	3
TABLA 1-2	COMPARATIVA DE LOS LENGUAJES DE PROGRAMACIÓN .....	9
TABLA 1-3	COMPARATIVA DE MYSQL Y ORACLE .....	11
TABLA 2-1	DATOS NECESARIOS PARA LA BASE DE DATOS.....	34
TABLA 2-2	DICCIONARIO DE DATOS (1).....	36
TABLA 2-3	DICCIONARIO DE DATOS (2).....	37
TABLA 4-1	ELEMENTOS AGREGADOS AL DICCIONARIO DE DATOS .....	73



## ÍNDICE DE FIGURAS

FIGURA 1-1 DIAGRAMA DE FLUJO DEL REGISTRO DE CURSO .....	15
FIGURA 1-2 DIAGRAMA DE FLUJO PARA INICIAR SESIÓN .....	16
FIGURA 1-3 DIAGRAMA DE FLUJO PARA VALIDACIÓN DE TIPO DE USUARIO .....	17
FIGURA 1-4 DIAGRAMA DE FLUJO PARA REGISTRO DE CURSOS .....	18
FIGURA 1-5 DIAGRAMA DE FLUJO PARA MODIFICACIÓN DE CURSOS .....	19
FIGURA 1-6 DIAGRAMA DE FLUJO PARA ACTIVACIÓN DE CURSOS .....	20
FIGURA 1-7 DIAGRAMA DE FLUJO PARA INSCRIPCIÓN DE CURSOS .....	21
FIGURA 2-1 DISEÑO FUNCIONAL DEL SISTEMA .....	28
FIGURA 2-2 MODELO COMPLETO DE LA BASE DE DATOS (ELABORACIÓN PROPIA) .....	35
FIGURA 2-3 MODELO ENTIDAD-RELACIÓN (ELABORACIÓN PROPIA) .....	38
FIGURA 2-4 XAMPP .....	39
FIGURA 2-5 MYSQL WORKBENCH .....	40
FIGURA 2-6 ADOBE DREAMWEAVER .....	40
FIGURA 2-7 ARQUITECTURA DE TRES CAPAS .....	41
FIGURA 2-8 FORMULARIO PARA INGRESO AL SISTEMA .....	42
FIGURA 2-9 FORMULARIO PARA REGISTRO .....	42
FIGURA 2-10 MENÚ DE USUARIO .....	43
FIGURA 2-11 MENÚ DE CURSOS INSCRITOS .....	43
FIGURA 2-12 FORMULARIO PARA CAMBIO DE CONTRASEÑA .....	43
FIGURA 2-13 FORMULARIO PARA CAMBIO DE DATOS .....	43
FIGURA 2-14 LISTA DE CURSOS .....	44
FIGURA 2-15 INSCRIPCIÓN DEL CURSO .....	44
FIGURA 2-16 FORMULARIO PARA REGISTRO DE CURSOS .....	45
FIGURA 2-17 FORMULARIO PARA CALIFICAR ALUMNOS .....	45
FIGURA 2-18 LISTA DE CALIFICACIONES .....	45
FIGURA 2-19 FORMULARIO PARA CAMBIO DE DATOS DE SEMESTRE .....	46
FIGURA 2-20 FORMULARIO PARA ACTIVACIÓN DE CURSOS .....	46
FIGURA 2-21 DATOS DEL PONENTE .....	46
FIGURA 2-22 FORMULARIO PARA DESCARGAR LISTAS .....	47
FIGURA 2-23 FORMULARIO PARA BÚSQUEDA DE USUARIOS .....	47
FIGURA 2-24 FORMULARIO PARA RESTABLECER CONTRASEÑAS DE USUARIOS .....	47
FIGURA 3-1 FUNCIONES PARA TODOS LOS MÓDULOS .....	52
FIGURA 3-2 FUNCIONES DE LOS MÓDULOS COMUNES .....	53
FIGURA 3-3 FUNCIONES DE LOS MÓDULOS DEL PONENTE .....	53
FIGURA 3-4 FUNCIONES DE LOS MÓDULOS DEL ADMINISTRADOR .....	54
FIGURA 3-5 FUNCIONES DE LOS MÓDULOS DEL ALUMNO .....	55
FIGURA 3-6 SIMBOLOGÍA DE LOS MAPAS DE LAS FUNCIONES .....	55
FIGURA 4-1 MODELO DE LA BASE DE DATOS (FINAL) .....	72
FIGURA 4-2 DISEÑO FUNCIONAL DEL SISTEMA (FINAL) .....	73
FIGURA 4-3 INGRESO AL SISTEMA .....	74
FIGURA 4-4 FORMULARIO DE INGRESO .....	74
FIGURA 4-5 FORMULARIO DE REGISTRO .....	75
FIGURA 4-6 ERROR EN REGISTRO. FUERA DEL PERIODO DE REGISTRO .....	75
FIGURA 4-7 ERROR EN REGISTRO. DIRECCIÓN DE CORREO .....	75
FIGURA 4-8 ERROR EN REGISTRO. CONTRASEÑAS DIFERENTES .....	75
FIGURA 4-9 ERROR EN REGISTRO. LONGITUD DE CONTRASEÑA .....	75
FIGURA 4-10 ERROR EN REGISTRO. FALTAN DATOS .....	76



FIGURA 4-11 MENSAJE DE REGISTRO EXITOSO .....	76
FIGURA 4-12 ERROR EN REGISTRO. REGISTRO FALLIDO .....	76
FIGURA 4-13 SOLICITUD DE REENVIO DE DATOS .....	76
FIGURA 4-14 ERROR EN REGISTRO. REGISTRO EXISTENTE .....	76
FIGURA 4-15 PÁGINA DE INICIO PARA EL PONENTE.....	77
FIGURA 4-16 MENÚ DEL PONENTE DESPUES DE LA FECHA DE REGISTRO .....	77
FIGURA 4-17 MENSAJE DE SESIÓN CERRADA EXITOSAMENTE .....	77
FIGURA 4-18 ERROR EN CIERRE DE SESIÓN .....	77
FIGURA 4-19 ERROR EN CIERRE DE SESIÓN. NO EXISTE UNA SESIÓN .....	77
FIGURA 4-20 FORMULARIO PARA REGISTRO DE CURSOS.....	78
FIGURA 4-21 ERROR EN REGISTRO DE CURSO. REGISTRO FUERA DE FECHAS .....	78
FIGURA 4-22 ERROR EN REGISTRO DE CURSO. FALTA DE DATOS .....	78
FIGURA 4-23 ERROR EN REGISTRO DE CURSO.....	79
FIGURA 4-24 REGISTRO EXITOSO DE UN CURSO.....	79
FIGURA 4-25 ERROR EN REGISTRO DE CURSO. REGISTRADO ANTERIORMENTE.....	79
FIGURA 4-26 PÁGINA DE INICIO CON LISTA DE CURSOS .....	80
FIGURA 4-27 FORMULARIO PARA LA MODIFICACIÓN DE CURSOS .....	80
FIGURA 4-28 CURSO ACTIVO .....	81
FIGURA 4-29 PÁGINA PARA DESCARGAR LAS LISTAS DE LOS CURSOS.....	81
FIGURA 4-30 DESCARGA DE LA LISTA DE ALUMNOS.....	82
FIGURA 4-31 LISTA EN FORMATO PDF .....	82
FIGURA 4-32 LISTA DE CURSOS POR CALIFICAR.....	83
FIGURA 4-33 FORMULARIO PARA CALIFICAR .....	83
FIGURA 4-34 MENSAJE DE ACTUALIZACIÓN DE CALIFICACIONES EXITOSA.....	84
FIGURA 4-35 ERROR EN ACTUALIZACIÓN DE CALIFICACIONES .....	84
FIGURA 4-36 FORMULARIOS PARA ACTUALIZACIÓN DE DATOS.....	84
FIGURA 4-37 ERROR EN LA ACTUALIZACIÓN. REGISTRO EXISTENTE .....	84
FIGURA 4-38 ERROR EN LA ACTUALIZACIÓN. FALTAN DATOS.....	85
FIGURA 4-39 ERROR EN LA ACTUALIZACIÓN. DIRECCIÓN DE CORREO NO VÁLIDA .....	85
FIGURA 4-40 ERROR EN EL CAMBIO DE CONTRASEÑA. CONTRASEÑAS DIFERENTES .....	85
FIGURA 4-41 ERROR EN EL CAMBIO DE CONTRASEÑA. LONGITUD DE CONTRASEÑA .....	85
FIGURA 4-42 ACTUALIZACIÓN DE DATOS EXITOSA.....	85
FIGURA 4-43 CAMBIO DE CONTRASEÑA EXITOSO.....	85
FIGURA 4-44 ERROR EN LA ACTUALIZACIÓN DE DATOS.....	85
FIGURA 4-45 ERROR EN CAMBIO DE CONTRASEÑA.....	85
FIGURA 4-46 INGRESO AL SISTEMA .....	86
FIGURA 4-47 FORMULARIO DE INGRESO .....	86
FIGURA 4-48 FORMULARIO DE REGISTRO .....	87
FIGURA 4-49 ERROR EN REGISTRO. FUERA DEL PERIODO DE REGISTRO .....	87
FIGURA 4-50 ERROR EN REGISTRO. DIRECCIÓN DE CORREO .....	87
FIGURA 4-51 ERROR EN REGISTRO. CONTRASEÑAS DIFERENTES.....	87
FIGURA 4-52 ERROR EN REGISTRO. LONGITUD DE CONTRASEÑA .....	87
FIGURA 4-53 ERROR EN REGISTRO. FALTAN DATOS.....	88
FIGURA 4-54 MENSAJE DE REGISTRO EXITOSO .....	88
FIGURA 4-55 ERROR EN REGISTRO. REGISTRO FALLIDO .....	88
FIGURA 4-56 SOLICITUD DE REENVIO DE DATOS .....	88
FIGURA 4-57 ERROR EN REGISTRO. REGISTRO EXISTENTE .....	88
FIGURA 4-58 PÁGINA DE INICIO PARA EL ALUMNO .....	89
FIGURA 4-59 MENÚ DEL ALUMNO DURANTE EL PROCESO DE INSCRIPCIÓN.....	89
FIGURA 4-60 MENÚ DEL ALUMNO TERMINADO EL PERIODO DE CURSOS.....	89
FIGURA 4-61 MENSAJE DE SESIÓN CERRADA EXITOSAMENTE .....	89
FIGURA 4-62 ERROR EN CIERRE DE SESIÓN .....	89
FIGURA 4-63 ERROR EN CIERRE DE SESIÓN. NO EXISTE UNA SESIÓN.....	90



FIGURA 4-64 CURSOS INSCRIBIBLES .....	90
FIGURA 4-65 DATOS DEL CURSO .....	91
FIGURA 4-66 MENSAJE DE ERROR EN INSCRIPCIÓN. EL CURSO YA ESTÁ INSCRITO.....	91
FIGURA 4-67 MENSAJE DE ERROR EN INSCRIPCIÓN. NÚMERO MÁXIMO DE CURSOS ALCANZADO .....	91
FIGURA 4-68 MENSAJE DE ERROR EN INSCRIPCIÓN. CURSO LLENO .....	92
FIGURA 4-69 MENSAJE DE ERROR EN INSCRIPCIÓN. FALLA EN EL SISTEMA .....	92
FIGURA 4-70 CURSO INSCRITO.....	92
FIGURA 4-71 LISTA DE CURSOS INSCRITOS CON MATERIAL ASIGNADO .....	93
FIGURA 4-72 FORMULARIO DE INGRESO. ....	93
FIGURA 4-73 PÁGINA DE INICIO PARA EL ADMINISTRADOR .....	94
FIGURA 4-74 MENSAJE DE SESIÓN CERRADA EXITOSAMENTE.....	94
FIGURA 4-75 ERROR EN CIERRE DE SESIÓN.....	94
FIGURA 4-76 ERROR EN CIERRE DE SESIÓN. NO EXISTE UNA SESIÓN.....	94
FIGURA 4-77 LISTA DE SEMESTRES.....	95
FIGURA 4-78 CAMBIO DE ESTADO DE UN SEMESTRE .....	95
FIGURA 4-79 PÁGINA PRINCIPAL CON SEMESTRE ACTIVO CAMBIADO .....	96
FIGURA 4-80 FORMULARIO PARA AGREGAR UN SEMESTRE .....	96
FIGURA 4-81 ERROR AL AGREGAR SEMETRE. FALTAN DATOS.....	96
FIGURA 4-82 FORMULARIO PARA LA EDICIÓN DE LOS DATOS DE UN SEMESTRE.....	97
FIGURA 4-83 ERROR AL MODIFICAR SEMETRE. FALTAN DATOS. ....	97
FIGURA 4-84 FORMULARIO PARA ACTIVACIÓN DE CURSOS .....	97
FIGURA 4-85 ERROR EN ACTIVACIÓN. FALTAN DATOS. ....	98
FIGURA 4-86 FORMULARIO DE REGISTRO.....	98
FIGURA 4-87 ERROR EN REGISTRO. DIRECCIÓN DE CORREO.....	98
FIGURA 4-88 ERROR EN REGISTRO. CONTRASEÑAS DIFERENTES. ....	98
FIGURA 4-89 ERROR EN REGISTRO. LONGITUD DE CONTRASEÑA.....	98
FIGURA 4-90 ERROR EN REGISTRO. FALTAN DATOS. ....	99
FIGURA 4-91 MENSAJE DE REGISTRO EXITOSO.....	99
FIGURA 4-92 ERROR EN REGISTRO. REGISTRO FALLIDO.....	99
FIGURA 4-93 SOLICITUD DE REENVIO DE DATOS.....	99
FIGURA 4-94 ERROR EN REGISTRO. REGISTRO EXISTENTE.....	99
FIGURA 4-95 FORMULARIO PARA LA BUSQUEDA DE UN ALUMNO .....	100
FIGURA 4-96 FORMULARIO PARA LA BUSQUEDA DE UN PONENTE .....	100
FIGURA 4-97 LISTA DE PONENTES ENCONTRADOS .....	101
FIGURA 4-98 DESCARGA DE REPORTE .....	101
FIGURA 4-99 FORMULARIO PARA DESCARGAR LISTAS DE ALUMNOS.....	102
FIGURA 4-100 DESCARGA DE LISTA DE ALUMNOS .....	102
FIGURA 4-101 LISTA EN FORMATO PDF .....	103
FIGURA 4-102 LISTA DE DONACIONES.....	103
FIGURA 4-103 LISTA DE ALUMNOS CON 1 CURSO INSCRITO .....	104
FIGURA 4-104 ASIGNACIÓN DE DONACIONES .....	104
FIGURA 4-105 ERROR EN LA ASIGNACION DE DONACIONES.....	104
FIGURA 4-106 ASIGNACIÓN DE DONACIONES EXITOSA .....	104



## INTRODUCCIÓN

Desde hace varios años, la FES (Facultad de Estudios Superiores) Aragón por medio del Laboratorio de docencia CAE (Centro de Apoyo Extracurricular) ha realizado la impartición de cursos en los períodos inter semestrales. Dichos cursos son impartidos por alumnos, ex alumnos y profesores de la carrera de Ingeniería en Computación.

Los cursos pretenden iniciar, orientar y actualizar a la comunidad universitaria en las nuevas tecnologías del área de la computación, tales como, el desarrollo de programas en los diversos lenguajes (C, C++, C#, Java, PHP, SQL etc.), mantenimiento y corrección de fallas en los equipos de cómputo, desarrollo de hardware, diversos temas relacionados con las redes computacionales, etc.

Sin embargo, durante la inscripción a los cursos se habían tenido dificultades para llevar a cabo esta tarea. Con el fin de dar solución a este problema se ha desarrollado un sistema de administración para la inscripción de los cursos extracurriculares de la FES Aragón.

Para lograr el proyecto, ha sido necesario realizar diversas etapas que se describen a continuación:

El capítulo uno “Análisis para la creación del sistema” tiene por objetivo enlistar los requerimientos necesarios para la generación del sistema, tales como, la problemática actual, tanto el lenguaje de programación como el manejador de base de datos que se debe utilizar, los objetivos del sistema y la solución propuesta.

En el capítulo dos “Diseño del sistema de inscripción” mediante el enfoque de la ingeniería de sistemas, se crea el modelo conceptual y, por ende, funcional. En consecuencia, se genera el modelo de la base de datos y se diseñan los módulos que permiten la eficiencia del sistema en consideración.

Con base en lo anterior, en el capítulo tres “Desarrollo del sistema” se ha creado visualmente el entorno en donde ha de interactuar el usuario y donde se realiza la manipulación de los datos. Es importante mencionar que la programación se ha dividido en las capas de presentación, lógica de control y acceso a datos, conforme se establecen los criterios básicos de la programación.

La última fase se expone en el capítulo cuatro que corresponde a las etapas de implantación, prueba y actualización. Con las que se ha permitido a los usuarios la realización de operaciones de lectura y escritura en las tablas de la base de datos, por medio de consultas sobre los registros de dichas tablas.

Finalmente se mencionan los resultados obtenidos durante el periodo de Junio, Julio y Agosto de 2010, y se manifiestan las recomendaciones dedicadas a estudiantes, investigadores y profesionales dedicados al ámbito del desarrollo de sistemas.



## *Justificación*

El Centro de Apoyo Extracurricular 504 (CAE) es un Laboratorio de Docencia perteneciente a la Carrera de Ingeniería en Computación, creado para satisfacer las necesidades de actividad práctica que requiere la carrera, asimismo, apoyar el proceso de enseñanza- aprendizaje de los alumnos y académicos.

Además, el mencionado centro apoya a los alumnos con cursos que complementan su formación profesional.

Para lograr con sus propósitos, el CAE realiza las siguientes funciones:

- Préstamo de salas de computadoras a alumnos del área de ingeniería para la realización de tareas, consulta en internet, cursos, clases y exposiciones.
- Préstamo de salas a profesores para impartir sus clases.
- Servicio de impresión, con clave.
- Préstamo de cañón para clases o exposiciones.
- Cursos extracurriculares para las áreas de software, hardware y telecomunicaciones.
- Alojamiento de páginas web.
- Cuentas para Linux y Unix.
- Servicio de cuentas de correo.
- Mantenimiento preventivo y correctivo a equipo de cómputo.

La importancia que tiene este laboratorio por sus contribuciones propias a la carrera de Ingeniería en Computación fue la razón por la cual se analizó la forma en que se estaba trabajando, donde se identificaron diversas problemáticas, como es el desconocimiento de actividades operativas en la organización de cursos extracurriculares, es decir, que no se sabe de antemano la cantidad y los temas de los cursos que se impartirán en el próximo periodo, esto debido a que los temas y sus contenidos dependen directamente del instructor.

Lo anterior ha creado incertidumbre en aquellos que pretenden inscribirse, pues no se sabe a ciencia cierta cuántos alumnos más estarán interesados en inscribirse al mismo curso, ya que si es muy grande la demanda hacia un tema en específico, y sólo hay un curso que abarque dicho tema, muchos quedarán fuera. Este es un problema que no tiene una fácil solución, pues como ya fue mencionado, no es posible determinar cuántos cursos de cada tema se tendrán.

Algunas de las graves consecuencias del proceso de inscripción, se manifiestan en que los alumnos han tenido que formarse para la inscripción desde muy temprano, con la finalidad de asegurarse un lugar en algún curso de su elección, de igual forma se han generado pérdidas de tiempo, daños a las instalaciones, inconformidades y conflictos que demeritan el papel de un centro de servicio como es el CAE.



En este sentido en la jefatura de la carrera de Ingeniería en Computación se previó que el problema en concreto por resolver era la ineficiencia que había estado operando en el proceso de inscripción.

Se justifica, entonces, que el presente proyecto de investigación, expresado como un caso práctico, tenga el objetivo de presentar un sistema de información (software) que permita automatizar procesos iterativos, asimismo, administrar la comunicación, representación y seguridad de información para responder a la solución de la problemática en cuestión.

Con base en lo anterior se despliegan las siguientes características del sistema:

- Debe ser capaz de administrar de forma correcta la inscripción de cursos.
- Debe permitir que aquellos que administran estos cursos tengan la posibilidad de realizar los cambios necesarios.
- Debe generar reportes.
- Debe permitir que al término de los cursos se pueda tener de forma eficiente las calificaciones obtenidas por los alumnos, y también servirá para poder generar las constancias de aquellos alumnos que hubiesen aprobado satisfactoriamente los cursos.





# *Capítulo 1*

## *ANÁLISIS PARA LA CREACIÓN DEL SISTEMA DE INSCRIPCIÓN*

*“El único autógrafo digno de un hombre es el que deja  
escrito con sus obras”*

*José Martí*

Este capítulo tiene por objetivo enlistar los requerimientos solicitados por los administradores del sistema, tales como, la problemática actual, la solución propuesta, el lenguaje que se debe utilizar para la programación y el manejador de base de datos. Además, se determina su factibilidad bajo la metodología de la investigación, ha sido necesario definir la problemática, establecer los requerimientos, realizar los diagramas de flujo y elegir una alternativa de software.

En la mayoría de los proyectos se demerita el valor real de esta etapa, ya que aquellos que generan dichos proyectos están siempre urgidos por llegar a la etapa de desarrollo, tomando la premisa “si falta algo, se lo agregamos luego”.

Esta forma de trabajo solo conlleva a problemas como: el mal funcionamiento del sistema, errores continuos, a la generación de una gran cantidad de versiones, a tener pérdida de recursos como: el tiempo, recursos humanos e incluso capital. Pero lo que es aún peor, se puede generar un sistema que no es capaz de resolver de forma eficiente la necesidad básica para lo que ha sido creado.

Es así, que de acuerdo a los principios básicos de ingeniería en computación, se debe establecer el objetivo del sistema, plantear cuál será el alcance y finalmente poder determinar los límites, sin embargo, lo más importante a tomar en consideración son las necesidades y tareas que el sistema debe cumplir, y las decisiones que este debe tomar en concreto ante cada caso al momento de estar en funcionamiento.

De esta forma, se expone parte del listado de elementos que ayudan a generar el análisis del sistema.

## ***1.1 OBJETIVO GENERAL***

El “SISTEMA DE ADMINISTRACIÓN PARA LA INSCRIPCIÓN DE LOS CURSOS EXTRACURRICULARES DE LA FES ARAGÓN” tiene como propósito fundamental eficientar el proceso de inscripción a los cursos inter semestrales del CAE, a través de la creación de un sistema de información (software) que considera la automatización, la administración de la comunicación, la representación y la seguridad de información.

## ***1.2 OBJETIVOS ESPECÍFICOS***

En primera instancia el sistema debe permitir que aquellos que desean ser instructores de algún curso (quienes en adelante serán llamados ponente o ponentes), se puedan registrar en el sistema con una clave y una contraseña, para después poder registrar los cursos que desean impartir en el período inter semestral próximo.

Una vez realizado lo anterior, el personal que labora en el Centro de Apoyo Extracurricular (CAE) y la jefatura de la carrera de Ingeniería en Computación



(quienes en adelante serán referidos como administrador o administradores), podrán tomar la decisión de aprobar o no cada uno de los cursos registrados.

En el caso de que un curso sea aprobado, los administradores deben tener la posibilidad de activarlo de manera que se vea reflejado automáticamente en el sistema, asimismo, en cuanto un curso se active los alumnos puedan visualizar su descripción desde el entorno al que ellos pueden acceder. Esto les ayudará en la toma de decisiones con respecto a los cursos que pretenden inscribir. Ya que la información que se proporciona es días, hora y temario del curso.

La siguiente fase es la de inscripción, por un lado, el sistema debe ser capaz de proporcionarle al alumno una interfaz sencilla y amigable y, por el otro, debe facilitar que el alumno se pueda inscribir en los cursos de su elección, además, ayudará a visualizar la disponibilidad de cada curso en tiempo real.

Finalmente, se necesita que el sistema otorgue a los administradores una interfaz donde puedan realizar todas las modificaciones que se necesitan para llevar a cabo la correcta administración del sistema.

### 1.3 CALENDARIO DE ACTIVIDADES

Los ciclos de vida son una parte esencial para los sistemas, para ello se necesita un calendario de actividades, cuyo propósito es determinar los tiempos para cada etapa, de esta forma se asegura que el sistema concluya en el tiempo definido inicialmente.

El sistema fue creado en el lapso comprendido entre los meses de mayo, junio y julio del 2010, es decir, 13 semanas, como se muestra en la tabla 1-1.

	1	2	3	4	5	6	7	8	9	10	11	12	13
<b>Análisis</b>	x	x											
<b>Diseño</b>		x	x	x									
<b>Desarrollo</b>				x	x	x	x	x	x				
<b>Pruebas</b>								x	x	x	x		
<b>Implementación</b>										x			
<b>Mantenimiento</b>												x	x

Tabla 1-1 Calendario de Actividades



## ***1.4 OBTENCIÓN DE REQUISITOS***

Con la determinación de los productos y resultados esperados que debe cumplir el sistema, ha sido necesario definir las necesidades de los administradores del sistema.

Para tal fin, fue necesario entender cómo se ha realizado el proceso de inscripción en el CAE, y hacer un análisis acerca del tipo y forma de seguridad que debe cumplir el sistema para, finalmente, tratar de determinar las posibles situaciones que se podrían presentar durante el día de la inscripción. Un claro ejemplo de esto último sería la pérdida de contraseñas por parte de los alumnos.

## ***1.5 DEFINICIÓN DEL SISTEMA***

El sistema debe ser capaz de permitir que los alumnos se inscriban en los cursos inter semestrales de forma eficiente, debe mostrar al usuario alumno únicamente aquellos cursos a los que se puede inscribir de acuerdo al subtipo de usuario que este sea, también debe ser capaz de determinar si el cupo de un curso está lleno y por tanto no permitir la inscripción de más alumnos en este, así mismo debe administrar la inscripción de los cursos por parte de los ponentes, la validación de dichos cursos por parte de los administradores, el manejo de calificaciones para los tres tipos de usuario y finalmente una administración general y con acceso total al usuario administrador.

## ***1.6 ALCANCE***

Este sistema está diseñado específicamente para la inscripción en el día señalado, además, se ha considerado el criterio de desempeño de efectividad, ya que entre sus funciones se encuentran la de generar reportes, crear listas de alumnos de cada grupo para los ponentes, restablecer contraseñas, y la obtención eficiente de calificaciones por parte de los administradores, lo cual permitirá que al final de los cursos se conozca exactamente el desempeño de cada alumno y se puedan generar los reconocimientos de los cursos.

## ***1.7 AMBIENTE DE TRABAJO***

Otro aspecto dentro del proceso de desarrollo del sistema, fue la designación del espacio público donde se desarrollaría el sistema. En este sentido, se consideró que en las propias instalaciones de la jefatura de la Licenciatura de Ingeniería en Computación, en conjunto, con las del CAE eran las infraestructuras óptimas, ya que ambas cubren lo relacionado con los aspectos tecnológicos y en ellas se tiene una participación activa para escuchar la voz del cliente.



## 1.8 PROBLEMÁTICA

La manera en la que se realizaba la inscripción de los alumnos a los cursos, era manual. Por tanto, los alumnos debían formarse el día de la inscripción en el exterior del CAE, cuando se iniciaba el proceso de inscripción se pasaba uno a uno a los alumnos, estos elegían los cursos en los que deseaban inscribirse y eran anotados en una lista, hasta que esta se llenaba. Esto solía ser un proceso lento y por tanto tedioso.

Otro problema era; que los alumnos que estaban formados no sabían qué cursos aún tenían cupo, y cuando era su turno ya estaban llenos. El proceso en total duraba horas.

Al terminar el proceso se tenían varios alumnos en descontento, por no inscribirse en algún curso de su elección e incluso por quedarse sin la oportunidad de entrar en algún otro, a pesar de la pérdida de tiempo en formaciones largas y tediosas.

Lo que algunos alumnos hacían era llegar a muy tempranas horas, sin embargo, había alumnos que metían a la fila a otros de sus compañeros, esto también generaba descontento.

El último problema no estaba relacionado con el día de la inscripción. Este proceso era la verificación de calificaciones, ya que al final del período inter semestral, se deben entregar constancias a aquellos que acreditan satisfactoriamente; para este trámite se le solicitaba al ponente una lista de calificaciones, esto de alguna forma generaba papeleo que ahora es innecesario, por tanto, se puede decir que con el sistema también se automatizó este proceso.

## 1.9 REQUERIMIENTOS DE LA BASE DE DATOS

Desde el momento mismo en que se determinó que el problema actual en el proceso de inscripción, sería resuelto por medio de la creación de un sistema, se determinó que este utilizaría una base de datos para el correcto manejo de estos.

Debido a que esta base de datos no existía, la primera tarea sería entonces diseñarla y determinar los roles que esta asumiría.

La base de datos debe almacenar toda la información correspondiente a la inscripción de los alumnos en los cursos, esto indica implícitamente que se debe tener la información de cada alumno, a cuántos cursos está inscrito y su calificación; también debe tener por tanto la información de cada ponente, cuántos cursos está impartiendo y quiénes son los alumnos inscritos en dichos cursos.

La información específica que se necesita es la siguiente:

**Para el alumno:** Nombre, Número de cuenta ó Número de trabajador, Correo electrónico, Contraseña y Subtipo de usuario.



**Para el ponente:** Nombre, Número de cuenta ó Número de trabajador, Correo electrónico, Contraseña y Subtipo de usuario.

**Para el curso:** Temario, Requisitos, Horario, Cupo, Ponente, Tema, Subtipo de usuario al que va dirigido el curso, Grupo, Sala y Lista de alumnos inscritos.

El sistema también necesita otro tipo de información, que es necesaria para el correcto manejo de la inscripción, estos datos ayudarán a manejar las siguientes situaciones:

- Que sólo se muestren los cursos que serán impartidos en el período inter semestral actual.
- Restringir el número máximo de cursos a los que un alumno se puede inscribir en el período actual.
- Fechas de registro.
- Fechas de inscripción.
- Fechas del periodo en el que se impartirán los cursos.
- Activación del curso.

Los datos necesarios para este correcto manejo son: Semestre, Fecha de inicio y término del período de los cursos, Fecha y hora de inicio y término del registro, Fecha y hora de inicio y término del proceso de inscripción, Número máximo de cursos y Estado.

Finalmente también es necesario tener datos con relación a los administradores. Estos datos son: Clave de administrador y Contraseña

## **1.10 SELECCIÓN DE SOFTWARE**

Para poder desarrollar un sistema como este, se necesita un lenguaje de programación del lado del servidor y un manejador de Base de Datos. Antes de tomar la decisión de cuales se utilizarían, se analizaron las posibilidades.

### **1.10.1 Análisis de los Lenguajes de Programación**

Una aplicación para la Web<sup>1</sup>, básicamente, es un programa, en sí muy parecido a cualquiera de los programas que todos nosotros tenemos en nuestra computadora, pero con la salvedad de que la computadora que hace correr ese programa, es el servidor<sup>2</sup> al que nos conectamos a través de un navegador,

---

<sup>1</sup> La **World Wide Web** o **WWW** o **W3** o simplemente **Web**, consiste en ofrecer una interface simple y consistente para acceder a la inmensidad de los recursos de Internet. Es la forma más moderna de ofrecer información en el medio más potente. La información se ofrece en forma de páginas electrónicas.

<sup>2</sup> Un **servidor web** es un programa que se ejecuta continuamente en un computador, manteniéndose a la espera de peticiones de ejecución que le hará un cliente o un usuario de Internet. El servidor web se encarga de contestar a estas peticiones de forma adecuada, entregando como resultado una página web o información de todo tipo de acuerdo a los comandos solicitados. En este punto es necesario aclarar lo siguiente: mientras que comúnmente se utiliza la palabra servidor para referirnos a una computadora con un software servidor instalado, en estricto rigor un servidor es el software que permite la realización de las funciones descritas.



convirtiéndose éste en la "pantalla" o ventana en la que visualizamos las cosas que ocurren al ejecutar las acciones de la aplicación. El servidor nos devuelve la información en formato HTML y así cualquier navegador es capaz de interpretar este código para poder visualizar de forma correcta la información que nos envía la aplicación. Es esta la razón por la cual se llaman lenguajes del lado del servidor.

A continuación se analizan los tres lenguajes más populares que actualmente se utilizan para realizar esta tarea, y se elige entre ellos el que tenga las mejores cualidades para generar el sistema.

**ASP:** Es una tecnología que trabaja sobre IIS y que permite interactuar con el usuario interceptando solicitudes que ingresan y procesan las respuestas que se producen.

Una vez que se crea un archivo ASP y se hospeda sobre IIS, el servidor verifica si se tienen scripts y de ser así se encarga de enrutarlos hacia ASP para procesarlos y enviarlos de vuelta al cliente. Posteriormente ASP procesa los scripts y genera el contenido apropiado para transmitirlo de vuelta al navegador/cliente. Gracias a que ASP tiene la oportunidad de procesar el resultado antes de enviarlo a un usuario se puede elaborar toda la funcionalidad requerida como por ejemplo el acceso a una base de datos.

ASP no es en sí un lenguaje de programación, es más bien un "pegamento" que mantiene unidos scripts, objetos, componentes e interacciones con el servidor Web. Se compone de objetos llamados desde VBScript o JScript para realizar todas las funciones necesarias. Sin embargo, la principal razón por la cual se decidió prescindir de este lenguaje, se debe a su alto costo, ya que para utilizarlo se debe pagar a Microsoft quien es el propietario del lenguaje.

**JSP:** Es una tecnología que permite mezclar HTML regular o estático con un contenido generado dinámicamente a través de los Servlets. Los Servlets son una tecnología de Java, pensada en tener una funcionalidad similar a la programación CGI. Son programas (clases) que corren en un servidor Web, actuando como una capa intermedia entre una solicitud proveniente de un Web Browser<sup>3</sup> u otro cliente HTTP<sup>4</sup> y bases de datos o aplicaciones en el servidor. El trabajo principal de un Servlet es:

- Obtener cualquier dato enviado por el usuario.
- Buscar cualquier otra información acerca de la solicitud que se encuentra incrustado en la solicitud HTTP.
- Generar resultados.

---

<sup>3</sup> Un **navegador** o **navegador web** (del inglés, **web browser**) es un programa que permite ver la información que contiene una página web (ya se encuentre ésta alojada en un servidor dentro de la World Wide Web o en un servidor local). El navegador interpreta el código, HTML generalmente, en el que está escrita la página web y lo presenta en pantalla permitiendo al usuario interactuar con su contenido y navegar hacia otros lugares de la red mediante enlaces o hipervínculos.

<sup>4</sup> Un cliente web (cliente HTTP), es cualquier aplicación que sirve para utilizar la web. Por ejemplo, los navegadores de internet.



- Dar un formato a los resultados dentro de un documento.
- Preparar los parámetros de respuesta HTTP.
- Mandar documentos de vuelta al cliente.

JSP a diferencia de ASP es un lenguaje que permite el desarrollo sin costo, pero por otro lado tiende a requerir gran cantidad de recursos por parte del servidor, esta es la razón por la cual tampoco se utilizó este lenguaje en la programación del sistema.

**PHP:** Es una tecnología que nos permite la generación de un contenido dinámico en un servidor Web. Esta tecnología fue creada originalmente en 1994 por Rasmus Lerdorf como un conjunto de herramientas (conocidas como PHP/FI y posteriormente PHP 2.0) que permitirían tener un dinamismo en la Web.

Dada la naturaleza de ser open-source (código abierto) estas herramientas ganaron gran popularidad y en 1997 Zeev Suraski y Andi Gutmans rediseñaron el parser, bautizándolo con el nombre de PHP3. Poco después se volvió a rediseñar y dio lugar a lo que actualmente conocemos como PHP4. Dentro de PHP existen muchos conceptos de Perl, C y Java por lo que gracias a esta sintaxis es más fácil familiarizarse con él, incluso para programadores principiantes.

Características principales de PHP:

- Tecnología que genera contenidos dinámicos y se encuentra embebido en archivos HTML.
- Dispone de un gran número de librerías de conexión con la mayoría de los manejadores de bases de datos para el manejo de información permanente en un servidor.
- Nos proporciona soporte a múltiples protocolos de comunicación en Internet (HTTP, IMAP, FTP, LDAP, SNMP, NNTP, POP3, etc.).
- Soporte para múltiples servidores Web: Apache, Microsoft Internet Information Server, Personal Web Server, Netscape e iPlanet, Oreilly Website Pro Server, Caudium, Xitami, OmniHTTPd y muchos otros.
- El código del intérprete está accesible para permitir posibles mejoras o sugerencias acerca de su desarrollo (Está escrito en lenguaje C).
- No es necesario adoptar alguna tecnología especial en lo que a Web servers se refiere ya que puede correr sobre cualquier plataforma.
- Consume muy pocos recursos del servidor, no es necesario tener un equipo muy poderoso.
- Es Software libre (gratuito).



### 1.10.2 Selección del Lenguaje de Programación

Con el fin de tomar una mejor decisión se elaboró la siguiente tabla.

	ASP	JSP	PHP
Consumo de recursos	Alto	Alto	Bajo
Portabilidad hacia otros servidores	No	Sí	Sí
Protección de Memoria	No	Sí	Sí
Facilidad de aprendizaje	Alto	Alto	Bajo
Costo	Necesita Licencia	Gratuito	Gratuito

**Tabla 1-2 Comparativa de los Lenguajes de Programación**

Una vez vistas las cualidades de PHP, se llegó a la conclusión de elegirlo como el lenguaje sobre el que se debía desarrollar el sistema.

### 1.10.3 Análisis de los Manejadores de las Bases de Datos

Existen muchos tipos de base de datos en función del modo en que almacenan y acceden a la información que guardan: Relacional, jerárquica, en la red, orientada a objetos, etc. Ejemplos de manejadores de bases de datos relacionales o RDBMS (Relational Database Management System por las siglas en inglés) hay muchos: MySQL, SQLite, Oracle, Informix, SyBase, Microsoft SQL Server, Postgres, Msq, etc.

Básicamente, un manejador de base de datos relacionales almacena los datos en Tablas, cada una de las cuales está formada por filas (o registros), y estas, a su vez, están formadas por columnas (o campos), Antes de definir una tabla, hay que normalizarla, proceso que consiste en evitar redundancia, es decir, que la información éste duplicada.

**MySQL:** Es un sistema de administración de bases de datos relacional (RDBMS). Se trata de un programa capaz de almacenar una enorme cantidad de datos de gran variedad y de distribuirlos para cubrir las necesidades de cualquier tipo de organización, desde pequeños establecimientos comerciales a grandes empresas y organismos administrativos. Mysql compite con sistemas RDBMS propietarios como Oracle, Sql Server y Db2.

Mysql incluye todos los elementos necesarios para instalar el programa, preparar diferentes niveles de acceso de usuario, administrar el sistema y proteger los datos. Puede desarrollar sus propias aplicaciones de bases de datos en la mayor parte de lenguajes de programación utilizados en la actualidad y ejecutarlos en casi todos los sistemas operativos. Mysql utiliza el lenguaje de consulta estructurado (SQL).

Antes Mysql se consideraba como la opción ideal de sitios web; sin embargo, ahora incorpora muchas de las funciones necesarias para otros entornos y



conserva su gran velocidad. Mysql es una base de datos robusta que se la puede comparar con una base de datos comercial, es incluso más veloz en el procesamiento de las transacciones y dispone de un sistema de permisos elegante y potente, y ahora, además, incluye un motor de almacenamiento InnoDB compatible con ACID, además dispone de store procedures, triggers y vistas.

Mysql es rápido, y una solución accesible para administrar correctamente los datos de una empresa. MysqlAB es la compañía responsable del desarrollo de Mysql, dispone de un sistema de asistencia eficiente y a un precio razonable, y, como ocurre con la mayor parte de las comunidades de código abierto, se puede encontrar una gran cantidad de ayuda en la Web.

Son muchas las razones para escoger a Mysql como una solución de misión crítica para la administración de datos:

- Costo: Mysql es gratuito para la mayor parte de los usos y su servicio de asistencia resulta económico.
- Asistencia: MysqlAB ofrece contratos de asistencia a precios razonables y existe una nutrida y activa comunidad Mysql.
- Velocidad: Mysql es mucho más rápido que la mayoría de sus rivales.
- Funcionalidad: Mysql dispone de muchas de las funciones que exigen los desarrolladores profesionales, como compatibilidad completa con ACID, compatibilidad para la mayor parte de SQL ANSI, volcados online, duplicación, funciones SSL e integración con la mayor parte de los entornos de programación.
- Portabilidad: Mysql se ejecuta en la inmensa mayoría de sistemas operativos y, la mayor parte de los casos, los datos se pueden transferir de un sistema a otro sin dificultad.

Facilidad de uso: Mysql resulta fácil de utilizar y de administrar. Las herramientas de Mysql son potentes y flexibles, sin sacrificar su capacidad de uso.

**Oracle:** Es un sistema de administración de base de datos (RDBMS) fabricado por Oracle corporation, básicamente es una herramienta cliente/servidor para la gestión de Bases de Datos. Es un producto vendido a nivel mundial, aunque la gran potencia que tiene y su elevado costo han hecho que su uso sea común únicamente en empresas muy grandes y multinacionales. En el desarrollo de páginas web pasa lo mismo: debido a que es un sistema muy caro, su uso no está tan extendido como otras bases de datos, por ejemplo, Access, Mysql, Sql Server, etc.

Para desarrollar en Oracle utilizamos PL/SQL un lenguaje de 5ª generación, bastante potente para tratar y gestionar bases de datos, también por norma general se suele utilizar SQL.

Oracle es sin duda uno de los mejores manejadores de bases de datos que tenemos en el mercado, es un sistema gestor de base de datos robusto, tiene muchas características que nos garantizan la seguridad e integridad de los datos;



que las transacciones se ejecuten de forma correcta, sin causar inconsistencias; ayuda a administrar y almacenar grandes cantidades de datos; estabilidad, escalabilidad y es multiplataforma.

#### 1.10.4 Selección del Manejador de la Base de Datos

Para poder tomar una mejor decisión se elaboró la siguiente tabla comparativa.

Categoría	Características	Mysql	Oracle
<b>Acceso a Datos</b>	Indices	Sí	Sí
	Diccionario de Datos	Sí	Sí
	Vistas	Sí	Sí
	Vistas Actualizables	Sí	Sí
	Triggers	Sí	Sí
	Cursores	Sí	Sí
	Uso de Alias	Sí	Sí
	Funciones	Sí	Sí
	Procedimientos Almacenados	Sí	Sí
<b>Alta Disponibilidad</b>	Incremental Backup	No	Sí
	FlashBack Table	No	Sí
	Recovery de transacciones erróneas	No	Sí
<b>Funcionalidad</b>	Replica	Sí	Sí
	Cluster	Sí	Sí
	Particionamiento de Tablas	Sí	Sí
	Automatic Storage Managment	Sí	Sí
	DataWareHouse	Sí	Sí
	Grid Computing	No	Sí
	Business Intelligence	Sí	Sí
	Gateways	No	Sí
	Xml	Sí	Sí
	Características de Orientación a Objetos	No	Sí
	Herramientas de Migración de Datos	Sí	Sí
	Identificar Sentencias Sql mal programadas	Sí	Sí
<b>Costo</b>	Necesita Licencia	<b>No</b>	<b>Sí</b>

**Tabla 1-3 Comparativa de MySQL y Oracle**



De entre los gestores anteriormente analizados, la elección de MySQL como gestor de base de datos radica en que es gratuito tanto para usos privados, como comerciales (sólo hay que pagar en el caso de que se desarrolle un producto comercial que éste basado en MySQL).

También se consideró su disponibilidad para distintos sistemas operativos (Unix, Linux, Windows, etc.), en que es capaz de trabajar con millones de registros y porque, además es muy rápido, no necesita grandes recursos del equipo y su uso está íntimamente relacionado con la programación con PHP.

Al revisar la tabla comparativa es posible ver que Oracle ofrece más características que MySQL, sin embargo para las operaciones que se requieren utilizar para el sistema en cuestión no son necesarias.

### *1.10.5 Análisis y Selección de los Distintos Tipos de Tablas*

La última decisión que se tomó en cuanto a la selección de software se refiere, fue con relación al tipo de tabla que se utilizaría.

Debido a que se eligió a MySQL como el manejador que se utilizaría, se necesitó analizar los tipos de tablas que este usa para el almacenamiento de la información.

MySQL soporta varios motores de almacenamiento que tratan con distintos tipos de tabla. Los motores de almacenamiento de MySQL incluyen algunos que tratan con tablas transaccionales y otros que no lo hacen:

Existen dos tipos de tablas de transacción segura (Innodb y BDB). El resto (HEAP, ISAM, MyISAM y MERGE) no son de transacción segura. La elección del tipo de tabla adecuado puede afectar enormemente al rendimiento.

**Tablas Heap:** Son el tipo de tabla más rápido porque se almacena en memoria y utilizan un índice asignado. La contrapartida es que como se almacenan en memoria, en el caso de una falla del sistema, los datos se pierden. No soportan columnas de autoincremento ni que haya valores nulos en los índices. Los datos son almacenados en pequeños bloques.

**Tablas ISAM:** Las tablas de tipo de Método de acceso secuencial indexado (ISAM) era el estándar antiguo de Mysql. Estas fueron sustituidas por las tablas MyISAM en la versión 3.23. Por lo tanto, es probable que solo se tope con este tipo de tablas si está trabajando con bases de datos antiguas.

**Tablas MyISAM:** Son la tecnología de almacenamiento de datos usada por defecto por el sistema administrador de bases de datos relacionales MySQL. Este tipo de tablas están basadas en el formato ISAM pero con nuevas extensiones.

La principal diferencia entre las dos, es el índice de las tablas MyISAM es mucho más pequeño que el de las tablas ISAM, de manera que un SELECT con un índice sobre una tabla MyISAM utilizará mucho menos recursos del sistema.



Se dividen en dos:

**Tablas estáticas:** Tienen longitud fija. Cada registro tiene asignado exactamente 10 Bytes.

Este tipo de tablas se caracterizan por:

- Ser muy rápidas (ya que Mysql sabe que el segundo nombre comienza siempre en el carácter número once)
- Resultan sencillas de almacenar en caché.
- Resultan sencillas para reconstruir tras un fallo.
- Requieren más espacio de disco

**Tablas Dinámicas:** Las columnas de las tablas dinámicas tienen diferentes tamaños. Aunque este tipo de dato ahorra espacio, resulta sin embargo más complejo.

Las tablas de tipo dinámico presentan las siguientes características:

- Todas las columnas de cadena son dinámicas.
- Por regla general, ocupan mucho menos espacio de disco que las tablas fijas.
- Las tablas requieren un mantenimiento regular para evitar su fragmentación.
- No resulta tan sencillo de reconstruir tras un fallo, especialmente si las tablas están muy fragmentadas.

**Tablas comprimidas:** Son tablas de sólo lectura que utilizan mucho menos espacio en disco. Son ideales para su uso con datos comprimidos que no cambien (que sólo se pueden leer y no escribir) y donde no exista mucho espacio disponible.

Las tablas comprimidas presentan las siguientes características:

- Las tablas son mucho más pequeñas.
- Como cada registro se comprime de forma separada, la carga de acceso es reducida.
- Cada columna se podría comprimir de forma diferente, utilizando distintos algoritmos de compresión.
- Se puede comprimir formatos de tabla fija y dinámica.

**Tablas Merge:** Son la fusión de las tablas MyISAM iguales. Por lo general se usa cuando las tablas MyISAM comienzan a resultar demasiado grandes.

Entre las ventajas de estas tablas se pueden mencionar las siguientes:

- Resultan más rápidas en determinadas situaciones.
- El tamaño de la tabla es más pequeño



Desventajas de la tabla Merge:

- Resultan mucho más lentas en búsquedas.
- El comando REPLACE no funciona sobre ellas.

**Tablas InnoDB:** Es un motor de bases de datos muy completo que ha sido integrado dentro de MySQL. Son tablas de transacción segura (lo que significa que dispone de las funciones COMMIT y ROLLBACK). En una tabla MyISAM, la tabla entera se bloquea al realizar funciones de inserción. Durante esa fracción de segundo, no se puede ejecutar ninguna otra instrucción sobre la tabla. El motor InnoDB usa una técnica conocida como multi-versioning que elimina la necesidad de hacer bloqueos en consultas SELECT muy simples. Lo que hacen estas funciones de bloqueo a nivel de fila es que solo se bloquee dicha fila y no toda la tabla, y se puedan seguir aplicando instrucciones sobre otras filas.

Por otro lado, InnoDB es el único formato que tiene MySQL para soportar Integridad referencial. Ahora se pueden definir llaves foráneas entre tablas InnoDB relacionadas para asegurarse de que un registro no pueda ser eliminado de una tabla si aún está siendo referenciado por otra tabla.

Otras de sus características es la recuperación automática ante fallas. Si MySQL se da de baja de una forma anormal, InnoDB automáticamente completará las transacciones que quedaron incompletas.

Además de todo lo comentado, InnoDB ofrece rendimientos superiores a la anterior tecnología de tablas de MySQL (MyISAM).

Una vez que se tomaron en cuenta las posibilidades, se seleccionó a InnoDB para la Base de Datos, debido a que esta última garantiza mejores resultados.



## 1.11 SOLUCIÓN PROPUESTA

Para llevar a cabo la inscripción se debe considerar un registro de los posibles participantes tanto de los alumnos como de los ponentes, por tanto este sería el primer proceso ver figura 1-1.

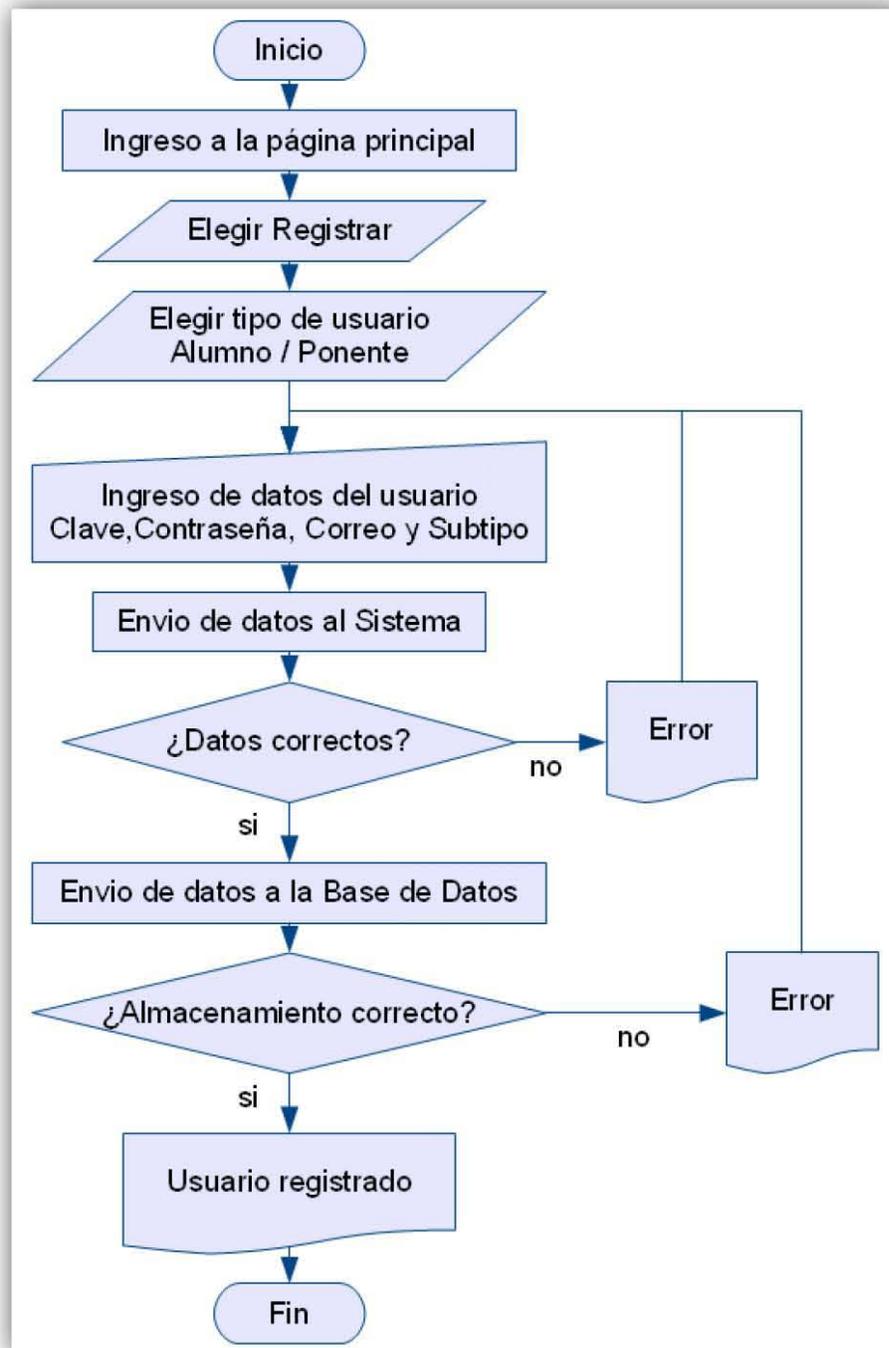
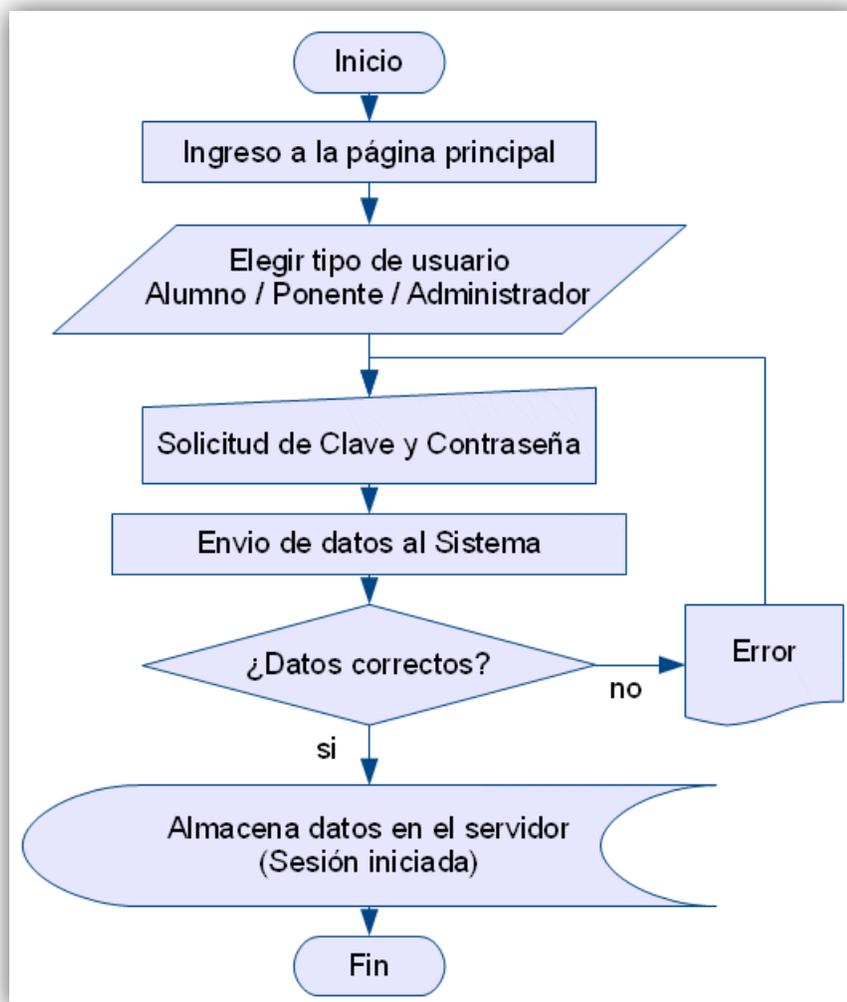


Figura 1-1 Diagrama de Flujo del Registro de Curso



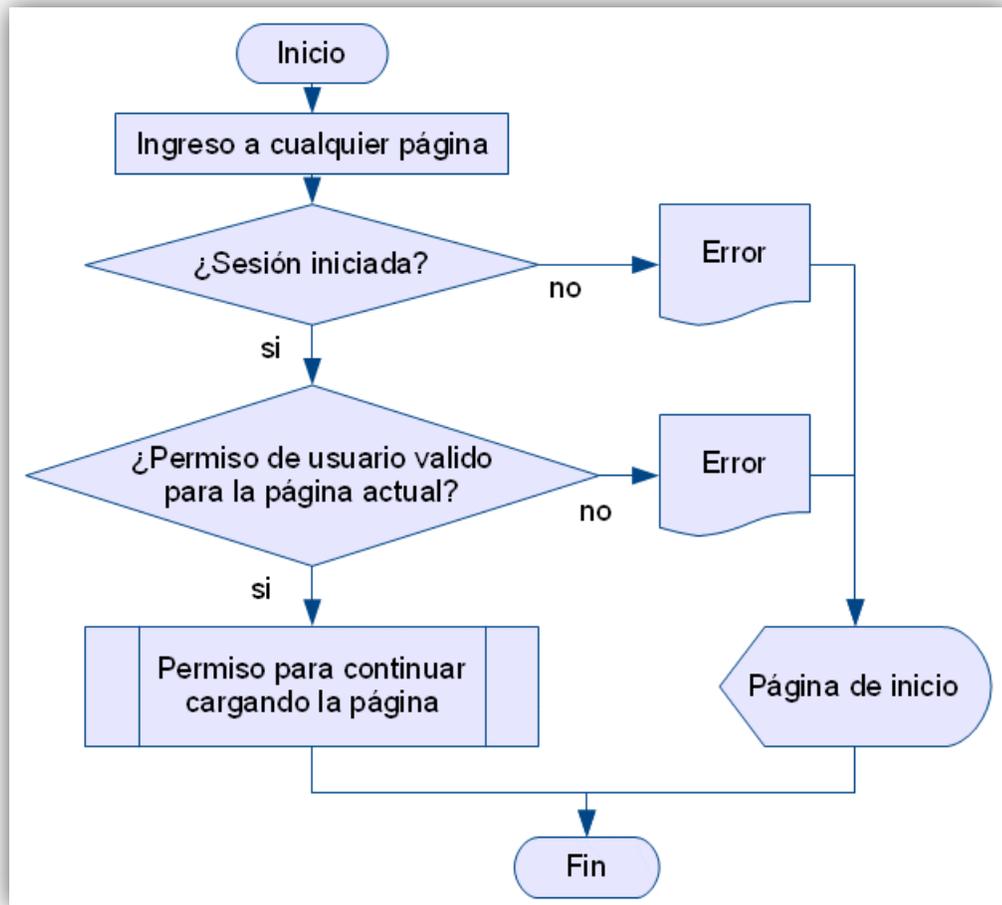
El siguiente proceso será el inicio de sesión, este proceso está ilustrado en la figura 1-2.



**Figura 1-2 Diagrama de Flujo para Iniciar Sesión**



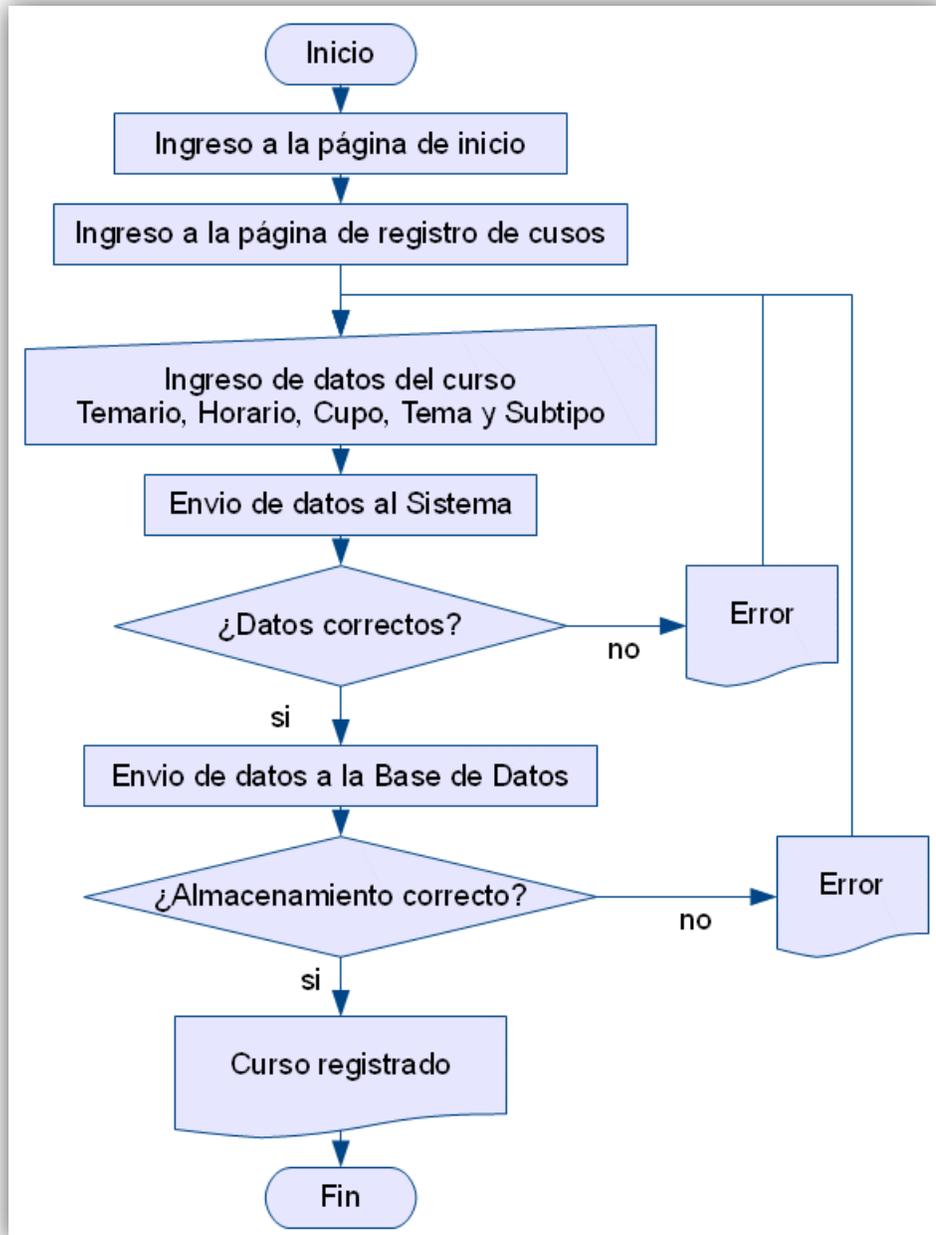
Otro proceso es la validación del tipo de usuario, se puede ver en la figura 1-3.



**Figura 1-3 Diagrama de Flujo para Validación de Tipo de Usuario**



En caso de ser ponente deberán inscribir los cursos que se desean impartir en el período, los cuales serán valorados y en su caso validados por los administradores, para que finalmente puedan ser vistos por los alumnos. Los procesos que realizan las tareas pueden ser vistos en las figuras 1-4, 1-5 y 1-6.



**Figura 1-4 Diagrama de Flujo para Registro de Cursos**



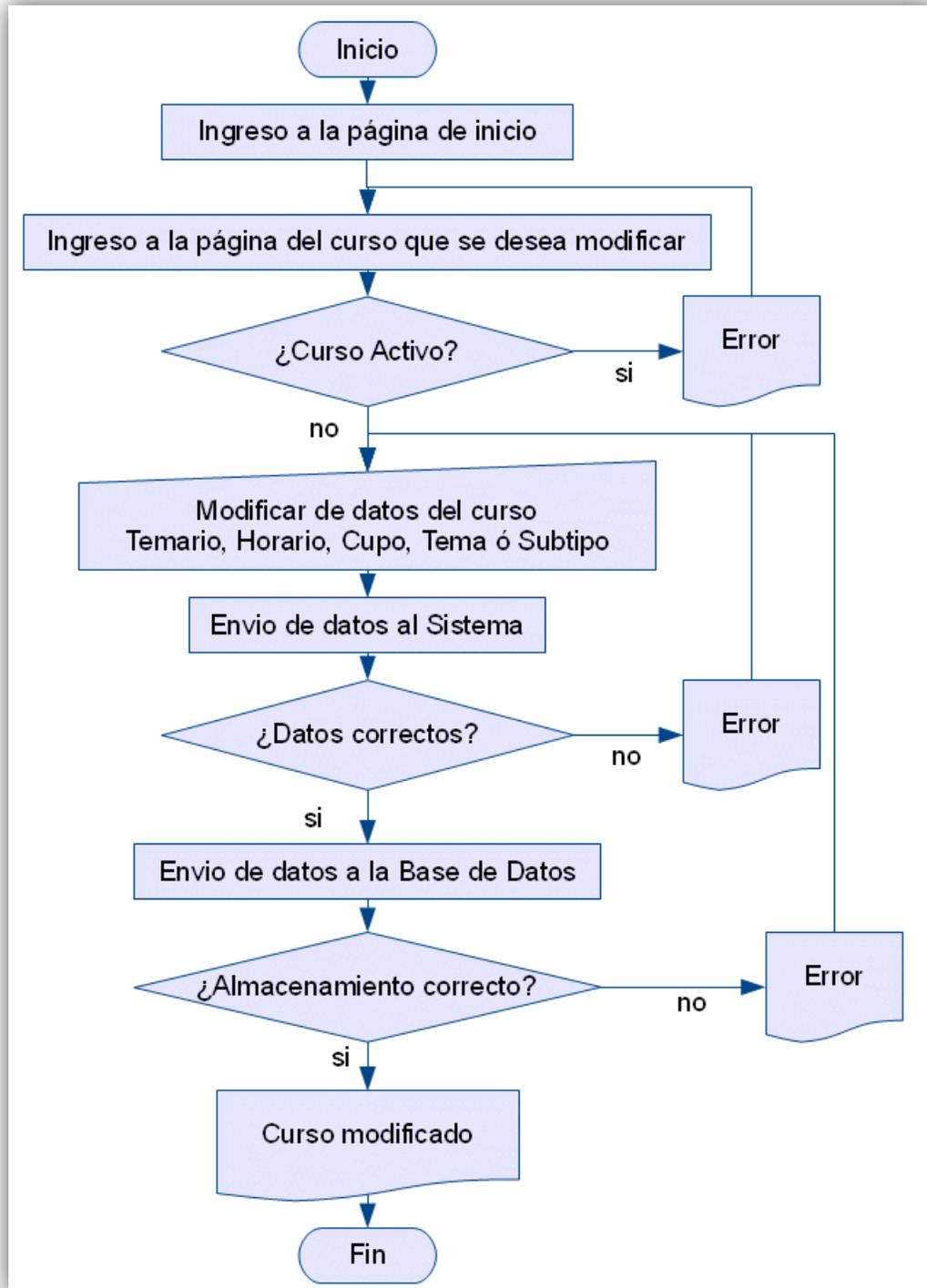


Figura 1-5 Diagrama de Flujo para Modificación de Cursos



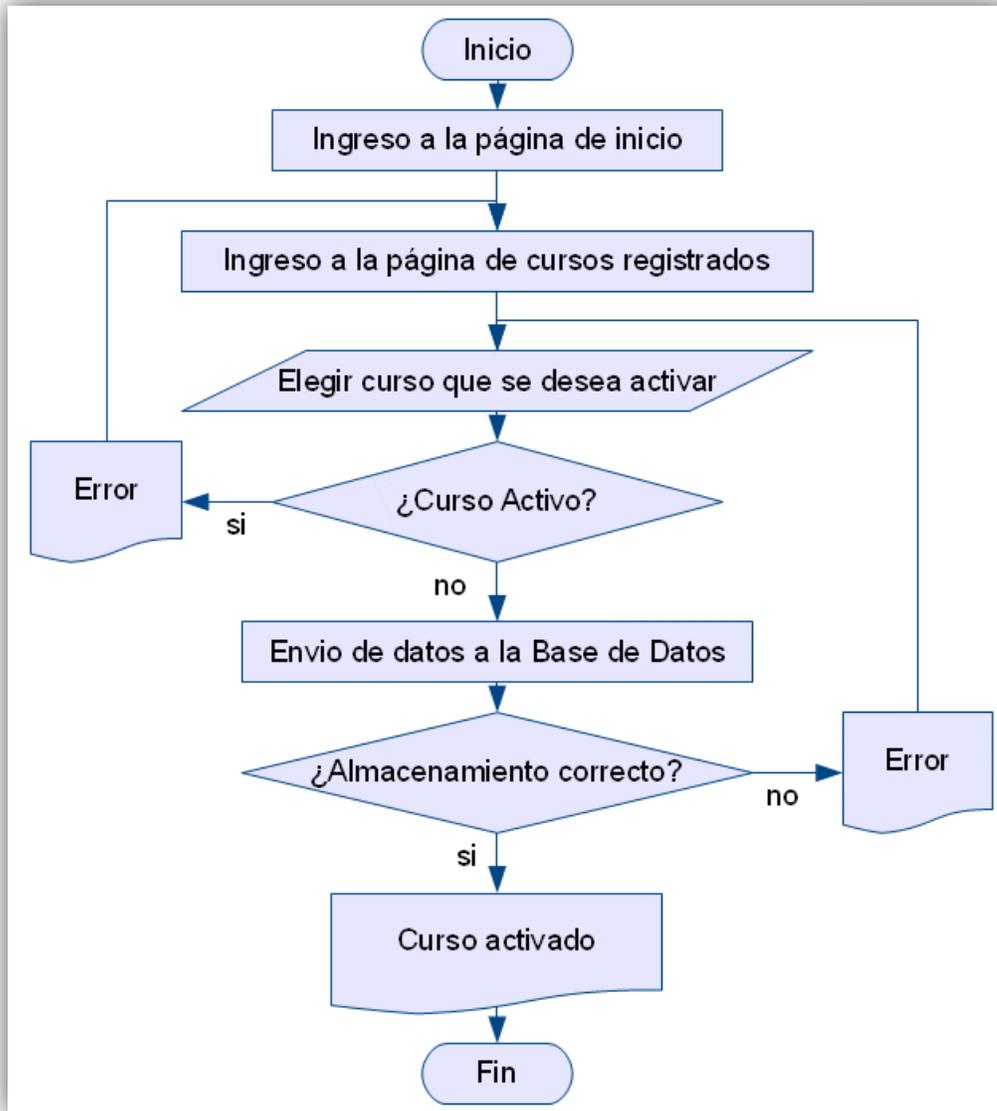


Figura 1-6 Diagrama de Flujo para Activación de Cursos



Otro proceso es el que llevan a cabo los alumnos con la inscripción de los cursos, la cual es muy importante. Ver figura 1-7.

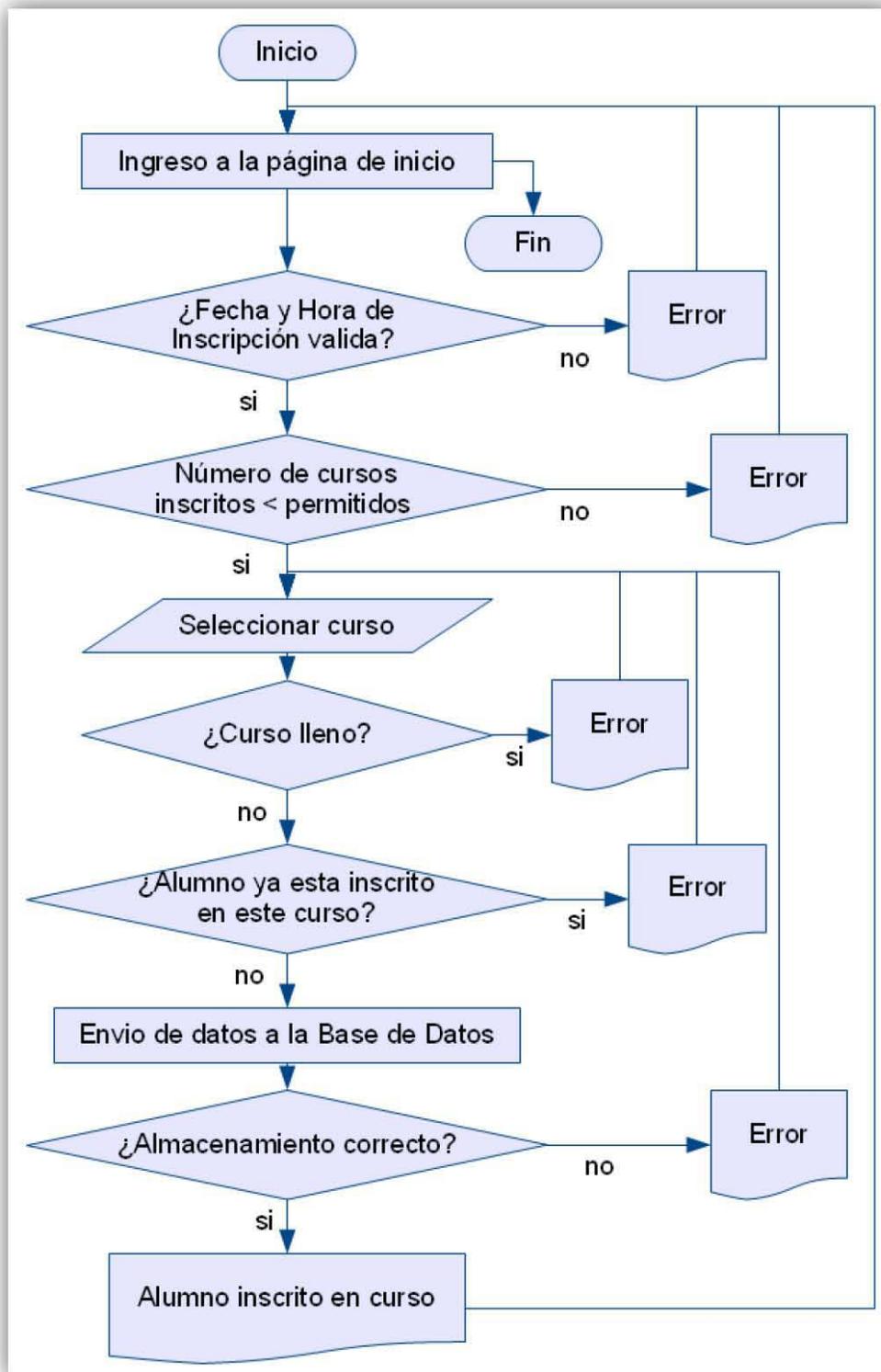


Figura 1-7 Diagrama de Flujo para Inscripción de Cursos



## 1.12 GLOSARIO

**ACID** (*Atomicity, Consistency, Isolation, Durability - Atomicidad, Consistencia, Aislamiento, Durabilidad*): Son las propiedades que una base de datos debe cumplir para que el Sistema administrador de base de datos (DBMS) maneje correctamente la transaccionalidad.

**Apache**: Es un servidor web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo.

**ASP** (*Active Server Pages - Páginas de Servidor Activas*): Tecnología de Microsoft del tipo "lado del servidor" que sirve para generar páginas web dinámicamente.

**Byte**: Es la unidad fundamental de datos en las computadoras, un byte son ocho bits contiguos. El byte es también la unidad de medida básica para memoria, almacenando el equivalente a un carácter.

**C**: Lenguaje de programación estructurado. Lo que quiere decir que el código o algoritmo esta ordenado o estructurado. Es uno de los lenguajes más rápidos y potentes. Tiene una sintaxis sumamente compacta y de alta portabilidad.

**CGI** (*Common Gateway Interface - Interfaz de Entrada Común*): Es de las primeras formas de programación web dinámica. Es un método para la transmisión de información hacia un compilador instalado en el servidor. Su función principal es la de añadir una mayor interacción a los documentos web que por medio del HTML se presentan de forma estática.

**Embeber**: En programación o desarrollo de páginas web, significa insertar (incrustar) código de un lenguaje dentro de otro lenguaje.

**Enrutar**: Redirigir o encaminar una conexión a un equipo en concreto que dispone de un servicio específico o un software que necesita realizar conexiones por un puerto determinado.

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

**HTML** (*Hyper Text Markup Language - Lenguaje de marcación de Hipertexto*): Es el lenguaje de marcas de texto utilizado normalmente en la www (World Wide Web). Fue creado en 1986 por el físico nuclear Tim Berners-Lee; el cual tomo dos herramientas preexistentes: El concepto de Hipertexto (Conocido también como link o ancla) el cual permite conectar dos elementos entre si y el SGML (Lenguaje



Estándar de Marcación General) el cual sirve para colocar etiquetas o marcas en un texto que indique como debe verse.

**HTTP** (*HyperText Transfer Protocol - Protocolo de Transferencia de Hipertexto*): Es el método más común de intercambio de información en la world wide web, el método mediante el cual se transfieren las páginas web a un ordenador. Todas las páginas web están escritas en lenguaje de hipertexto (HTML), por lo que el hipertexto es el contenido de las páginas web. El protocolo de transferencia es el sistema mediante el cual se transfiere información entre los servidores y los clientes (por ejemplo los navegadores).

**InnoDB**: Es un tipo de tabla de Mysql que permite trabajar con transacciones, y definir reglas de integridad referencial.

**IIS** (*Internet Information Services*): Es un servidor web y un conjunto de servicios para el sistema operativo Microsoft Windows. Originalmente era parte del Option Pack para Windows NT. Luego fue integrado en otros sistemas operativos de Microsoft destinados a ofrecer servicios, como Windows 2000 o Windows Server 2003. Windows XP Profesional incluye una versión limitada de IIS. Los servicios que ofrece son: FTP, SMTP, NNTP y HTTP/HTTPS. Este servicio convierte a una PC en un servidor web para Internet o una intranet, es decir que en las computadoras que tienen este servicio instalado se pueden publicar páginas web tanto local como remotamente.

**IMAP** (*Internet Message Access Protocol - Protocolo de Acceso a los Mensajes de Internet*): Es un protocolo de red de acceso a mensajes electrónicos almacenados en un servidor. Mediante IMAP se puede tener acceso al correo electrónico desde cualquier equipo que tenga una conexión a Internet.

**Java**: Lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

**JScript** o **Javascript**: Lenguaje de programación que permite a los desarrolladores crear acciones en sus páginas web. No requiere de compilación ya que el lenguaje funciona del lado del cliente, los navegadores son los encargados de interpretar estos códigos.

**JSP** (*Java Server Pages - Páginas de Servidor Java*): Tecnología orientada a crear páginas web con programación mediante el lenguaje Java.

**LDAP** (*Lightweight Directory Access Protocol - Protocolo Ligero de Acceso a Directorios*): Es un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red. LDAP también es considerado una base de datos (aunque su sistema de almacenamiento puede ser diferente) a la que pueden realizarse consultas.



**Linux:** Se trata de un sistema operativo de 32 bits de libre distribución, desarrollado originalmente por Linus Torvalds, un estudiante de la universidad finlandesa de Helsinki, quien, en 1991, se abocó a la tarea de reemplazar a Minix, un clon de Unix de pequeñas proporciones y finalidad académica desarrollado años antes por Andrew Tannenbaun.

**NNTP** (Network News Transport Protocol - Protocolo para la Transferencia de Noticias en Red): Es un protocolo inicialmente creado para la lectura y publicación de artículos de noticias en Usenet.

**Perl** (*Practical Extraction and Report Language - Lenguaje Práctico de Extracción y de Informes*): Es un lenguaje interpretado que tiene varias utilidades, pero está principalmente orientado a la búsqueda, extracción y formateado de ficheros de tipo texto. También es muy usado para manejo y gestión de procesos (estado de procesos, conteo y extracción de parámetros característicos, etc...).

**PL/SQL** (*Procedural Language extensions to the Structured Query Language – Lenguaje de Procedimiento extendido para el Lenguaje de Consulta Estructurado*): Lenguaje de programación incrustado en Oracle y PostgreSQL. Soporta todas las consultas, ya que la manipulación de datos que se usa es la misma que en SQL, incluyendo nuevas características:

- El manejo de variables.
- Estructuras modulares.
- Estructuras de control de flujo y toma de decisiones.
- Control de excepciones.

**POP3** (*Post Office Protocol - Protocolo de la Oficina de Correo*): En informática se utiliza en clientes locales de correo para obtener los mensajes de correo electrónico almacenados en un servidor remoto. Las versiones del protocolo POP (informalmente conocido como POP1) y POP2 se han hecho obsoletas debido a las últimas versiones de POP3. En general cuando uno se refiere al término POP, nos referimos a POP3 dentro del contexto de protocolos de correo electrónico.

**Script:** Cuya traducción literal es guión o archivo de órdenes o archivo de procesamiento por lotes es un programa usualmente simple, que por lo regular se almacena en un archivo de texto plano. El uso habitual de los scripts es realizar diversas tareas como combinar componentes, interactuar con el sistema operativo o con el usuario.

**SMTP** (*Simple Mail Transfer Protocol - Protocolo Simple de Transferencia de Correo*): Es un protocolo de la capa de aplicación. Basado en texto utilizado para el intercambio de mensajes de correo electrónico entre computadoras u otros dispositivos (PDA's, teléfonos móviles, etc.).

**SNMP** (*Simple Network Management Protocol - Protocolo Simple de Administración de Red*): Es un protocolo de la capa de aplicación que facilita el intercambio de información de administración entre dispositivos de red. Es parte de la familia de protocolos TCP/IP. SNMP permite a los administradores



supervisar el funcionamiento de la red, buscar y resolver sus problemas, y planear su crecimiento.

**SSL** (*Secure Sockets Layer - Protocolo de Capa de Conexión Segura*): Es un protocolo criptográfico que proporciona comunicaciones seguras por una red. Es la misma tecnología que mantiene seguros los datos de los números de tarjetas de crédito cuando se realizan compras a través del internet.

**SQL** (*Structured Query Language –Lenguaje de Consulta Estructurado*): Es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en éstas. Una de sus características es el manejo del álgebra y el cálculo relacional permitiendo efectuar consultas con el fin de recuperar -de una forma sencilla- información de interés de una base de datos, así como también hacer cambios sobre ella. Es un lenguaje informático de cuarta generación (4GL).

**SQL ANSI**: Estándar para el manejo del lenguaje estructurado de consultas. El Instituto Nacional Estadounidense de Estándares (ANSI, por sus siglas en inglés: American National Standards Institute) es una organización sin ánimo de lucro que supervisa el desarrollo de estándares para productos, servicios, procesos y sistemas en los Estados Unidos.

**TCP/IP** (Transmission Control Protocol / Internet Protocol - Protocolo de Control de Transmisión / Protocolo de Internet): Es un conjunto de protocolos. En algunos aspectos, TCP/IP representa todas las reglas de comunicación para Internet y se basa en la noción de dirección IP, es decir, en la idea de brindar una dirección IP a cada equipo de la red para poder enrutar paquetes de datos. Debido a que el conjunto de protocolos TCP/IP originalmente se creó con fines militares, está diseñado para cumplir con una cierta cantidad de criterios, entre ellos:

- dividir mensajes en paquetes.
- usar un sistema de direcciones.
- enrutar datos por la red.
- detectar errores en las transmisiones de datos.

**Trigger** (disparador): En una Base de Datos, es un procedimiento que se ejecuta cuando se cumple una condición establecida al realizar una operación de inserción (INSERT), actualización (UPDATE) o borrado (DELETE). Son usados para mejorar la administración de la Base de datos, sin necesidad de contar con que el usuario ejecute la sentencia de SQL. Además, pueden generar valores de columnas, previene errores de datos, sincroniza tablas, modifica valores de una vista, etc. Permite implementar programas basados en paradigma lógico (sistemas expertos, deducción).

**VBScript**: Subconjunto de Visual Basic for Applications. Lenguaje script cuyo uso se extiende tanto en páginas web de maquinas cliente como en páginas activas de servidor (ASP).



**Unix:** Es un sistema operativo portable, multitarea y multiusuario; desarrollado, en principio, en 1969 por un grupo de empleados de los laboratorios Bell de AT&T, entre los que figuran Ken Thompson, Dennis Ritchie y Douglas McIlroy.

**Windows:** Es el nombre de una serie de sistemas operativos desarrollados por Microsoft desde 1981, año en que el proyecto se denominaba "Interface Manager".





## *Capítulo 2*

# *DISEÑO DEL SISTEMA DE INSCRIPCIÓN*



*“Para ir adelante de los demás, se necesita ver más  
que ellos”*

*José Martí*

Durante la etapa de diseño se generaron los algoritmos que se deben seguir para llevar un sistema al plano físico. Aun cuando ya se había decidido en que lenguaje se realizaría el desarrollo, el diseño de un sistema debe poder ser aplicable a cualquier otro lenguaje, esto lo hace universal, y por tanto de fácil entendimiento para cualquiera.

## 2.1 DISEÑO FUNCIONAL

El sistema debe contar con módulos en donde cada uno de ellos resolverá un problema específico dentro del sistema, estos módulos como ya fue mencionado deben ser concebidos como algo muy independiente al lenguaje en el cual se pretende programar.

Los módulos se dividieron en cuatro grupos, estos son: alumnos, ponentes, administradores y comunes. A continuación se mencionan los módulos; acomodados por grupo.

- Para el alumno: inscripción y calificaciones.
- Para el ponente: registrar cursos, modificar curso y evaluar alumnos.
- Para los administradores: activación de cursos, inscribir alumnos, borrar alumnos, restablecer contraseñas, mostrar calificaciones, generación de reportes y configuración.

Finalmente los módulos comunes: registro, identificación, cambio de datos y descargar listas. La forma en la que los módulos están relacionados entre sí se muestra en la figura 2-1

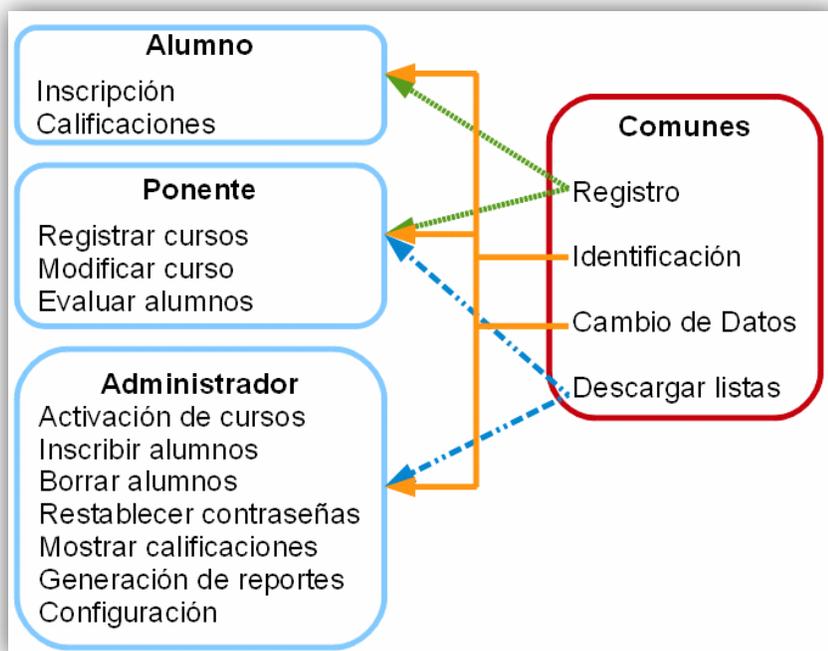


Figura 2-1 Diseño Funcional del Sistema



## 2.2 DISEÑO DETALLADO

A continuación se describirán las características y funcionalidad de cada módulo.

### 2.2.1 Módulos Comunes

A estos módulos se les llama así debido a que son utilizadas por los diversos usuarios.

**Módulo Registro:** Es uno de los módulos más sencillos ya que lo único que valida es que el usuario haya introducido todos sus datos. La finalidad de este módulo es simplemente agregar los datos del usuario al sistema.

El único punto a considerar, será la clasificación en tres subtipos de usuario para el alumno y tres subtipos de usuario para el ponente.

Los subtipos para el usuario alumno son:

- El subtipo alumno, quien a su vez, será alumno de la UNAM.
- El subtipo ex alumno, que será como su nombre lo indica ex alumno de la UNAM, este subtipo será útil para fines estadísticos, ya que permitirá conocer cuántos ex alumnos regresan a tomar cursos a la FES Aragón después de haber egresado. Los cursos a los que se pueden inscribir son los mismos que los del subtipo alumno.
- El subtipo profesor, será un académico de la UNAM. Esta diferenciación servirá para los casos en los que los cursos vayan dirigidos específicamente a profesores de la UNAM, y por tanto que solo ellos puedan inscribirse a los mismos.

Para el usuario ponente también se le solicitará que proporcione el subtipo de usuario al cual pertenece. Este dato también posee solo un valor estadístico. Ayudará a saber quiénes son los grupos que más buscan aportar sus conocimientos a otros miembros de la comunidad. Los tres subtipos también serán: alumno, ex alumno y profesor.

**Módulo Identificación:** Este módulo es sin duda uno de los más importantes de la aplicación, ya que sirve para verificar la autenticidad de las acciones realizadas dentro del sistema, una vez que tanto los alumnos como los ponentes se hayan registrado estos han determinado una contraseña que será necesaria para acceder al sistema.

Así pues, la tarea específica de este módulo es la de solicitar al usuario su número de cuenta (para alumnos y ex alumnos de la UNAM) o bien su número de trabajador (para académicos de la UNAM), estos números serán referenciados en adelante como clave de usuario, y la contraseña antes mencionada para permitir que el usuario pueda acceder al sistema. Una vez que el módulo identifica al usuario como válido, este puede realizar aquellas acciones que le están permitidas



de acuerdo al tipo de usuario que este sea, a esta acción se le conoce como iniciar sesión.

El iniciar sesión también es útil para el sistema, debido a que el sistema almacena los datos del usuario en el servidor por un determinado tiempo, y así cuando el usuario se mueve por diferentes páginas, el sistema sabe de quién se trata. Por ejemplo cuando un alumno presiona el botón inscribir curso, el sistema sabe exactamente a qué alumno debe inscribir a dicho curso.

**Módulo Cambio de Datos:** Este módulo permite que los usuarios puedan cambiar alguno o todos sus datos, ya sea por actualización o por corrección de algún dato erróneo.

**Módulo Descargar Listas:** Este módulo sirve para que los ponentes y administradores tengan las listas de alumnos para cada uno de los cursos.

La generación de las listas se realizará en un archivo PDF. Que es un tipo de archivo universal y útil, por lo que el usuario no tendrá problemas para visualizarlo e imprimirlo desde cualquier computadora.

### *2.2.2 Módulos del Alumno*

A continuación se describirán los módulos que utilizan los usuarios del tipo alumno.

**Módulo Inscripción:** Este módulo debe ser sin lugar a dudas el más complicado, ya que en este cae todo el peso del sistema. En él se realizarán todas las validaciones necesarias para que no se generen errores en el momento de la inscripción, estas validaciones son:

- Verificación de fecha y hora de inscripción, para permitir el acceso.
- Despliega como inscribibles solo aquellos cursos que no están saturados.
- En caso de intentar inscribirse en un grupo saturado no permitirlo.
- No permitir que el alumno se inscriba al mismo curso varias veces (esto puede ocurrir al refrescar la página o a un error del explorador, lo cual haría que se enviara la petición de inscripción al mismo curso nuevamente).
- Limitar la cantidad de cursos a los cuales un alumno puede inscribirse en un mismo período.
- Verificar que se realizó correctamente el envío y almacenamiento de datos a la base de datos.

Como se puede ver, este módulo podría ser determinado como el núcleo del sistema, puesto que además es encargado de resolver el problema principal para lo que fue creado el sistema.

**Módulo Calificaciones:** Este es un módulo sencillo, su tarea es mostrar las calificaciones del alumno por cada curso inscrito.



### 2.2.3 Módulos del Ponente

Para realizar las tareas necesarias para el usuario del tipo ponente se necesitan los módulos que se describen a continuación.

**Módulo Registrar Cursos:** En él se proporciona al ponente de un sencillo formulario donde tiene que proporcionar los datos específicos del curso, una vez terminado, envía la información a la base de datos, la que será aprobada por los administradores.

**Módulo Modificar Curso:** Este módulo permite que el ponente cambie los datos de alguno de los cursos que haya registrado. Un punto importante es el hecho de que el ponente tendrá acceso a la modificación de los datos del curso mientras este no haya sido aprobado (Activado).

**Módulo Evaluar Alumnos:** La función de este módulo es mostrar al ponente un formulario en forma de lista con el nombre de los alumnos y una casilla para cada uno, donde debe suministrar la calificación correspondiente. En el mismo módulo está contemplado el cambio de calificación en caso de algún error por parte del ponente o bien por algún posible error externo.

### 2.2.4 Módulos del Administrador

El usuario administrador posee la mayor cantidad de acciones, para realizarlas se ocuparan los módulos que ahora serán descritos.

**Módulo Activar y Modificar cursos:** La tarea de este módulo es mostrar al usuario administrador aquellos cursos que han sido ingresados por los ponentes y que necesitan ser activados. Bastará pulsar un botón con la leyenda Activar para realizar la acción. Por otro lado hay datos del curso que únicamente deben ser modificados o asignados por el administrador tales como el grupo, la sala y el cupo.

**Módulo Inscribir Alumnos:** Durante el proceso de inscripción cabe la posibilidad de que algún alumno no se pueda inscribir a un curso, ya sea por algún error o bien porque el cupo del curso esté lleno. Es necesario para el administrador tener una forma de inscribir directamente al alumno en dicho curso. Esta es la tarea que realiza este módulo.

**Módulo Borrar Alumnos:** También existe la posibilidad de que algún alumno haya quedado inscrito en un curso y se necesite dar de baja del mismo, este módulo ayuda a realizar la tarea.

**Módulo Restablecer Contraseñas:** Este es un módulo que permite establecer una nueva contraseña para algún usuario que la haya olvidado. La forma de hacerlo será proporcionando un formulario de búsqueda por medio de los campos nombre y número, al realizar la búsqueda se desplegará toda la lista de los usuarios que cumplen con la condición. En dicha lista se podrá seleccionar un elemento que será el usuario al cual se le restablecerá la contraseña.



**Módulo Mostrar Calificaciones:** Es un módulo cuya única tarea es mostrar las calificaciones de los alumnos, las listas son mostradas curso por curso.

**Módulo Generación de Reportes:** Este módulo sirve para que los administradores puedan generar reportes del semestre activo. En él se tendrán los datos de cada curso, su lista de alumnos, los datos del ponente y las calificaciones obtenidas, así como datos específicos de los alumnos. La generación de estos reportes se hará en un archivo csv. Este tipo de archivo es tan solo un formato sencillo donde cada dato es separado por comas. Su utilización es favorable para programas de hoja de cálculo como Excel y de tipo estadístico como el Stata o el Systad.

**Módulo Configuración:** Este módulo es de utilidad para el usuario administrador ya que por medio de él se podrán realizar las modificaciones necesarias para el buen funcionamiento y las restricciones del sistema.

Las principales configuraciones son:

- Fecha y hora del inicio y término del periodo de registro.
- Fecha y hora del inicio y término del proceso de inscripción.
- Número máximo de cursos a los que un alumno se puede inscribir.
- Cambio del estado de un semestre (Actual, Activo o Inactivo).
- Creación de nuevos periodos (Semestres).

## 2.3 BASE DE DATOS

Sin duda el modelado de la base de datos es crucial en el óptimo funcionamiento de cualquier sistema, ya que si este no se realiza adecuadamente puede crear conflictos en la programación. Un buen modelo incluso puede mejorar el tiempo de respuesta del manejador de la base de datos en cada consulta.

En esta tarea, no se debe dejar a un lado el correcto uso de los tipos de datos, por ejemplo: no se debe utilizar un tipo de dato Varchar<sup>5</sup> en vez de Char<sup>6</sup>, cuando el tamaño de los datos es constante.

Dentro del modelo entidad-relación<sup>7</sup> se debe considerar la optimización de las relaciones entre las entidades. Finalmente se debe tratar en la medida de lo posible de utilizar la base de datos para que las configuraciones del sistema se

<sup>5</sup> **Varchar:** Datos de caracteres no Unicode de longitud variable.  $n$  puede ser un valor entre 1 y 8.000. **max** indica que el tamaño de almacenamiento máximo es de  $2^{31}-1$  bytes. El tamaño de almacenamiento es la longitud real de los datos especificados + 2 bytes. Los datos especificados pueden tener una longitud de 0 caracteres. Los sinónimos ISO de **varchar** son **char varying** o **character varying**.

<sup>6</sup> **Char:** Datos de caracteres no Unicode de longitud fija, con una longitud de  $n$  bytes.  $n$  debe ser un valor entre 1 y 8.000. El tamaño de almacenamiento es  $n$  bytes. El sinónimo ISO de **char** es el tipo **character**.

<sup>7</sup> **Entidad-Relación:** Es un diagrama que consiste en objetos relacionados entre sí, que permiten identificar las entidades, atributos y relaciones que van a formar parte de la base de datos.



puedan llevar dinámicamente. Tomando en cuenta todos estos puntos se puede tener un sistema más funcional.

### 2.3.1 Diseño de la Base de Datos

Las bases de datos se pueden clasificar de acuerdo a su modelo de administración de datos. Un modelo de datos es básicamente una "descripción" de algo conocido como contenedor de datos (algo en donde se guarda la información), así como de los métodos para almacenar y recuperar información de esos contenedores. Los modelos de datos no son cosas físicas; son abstracciones que permiten la implementación de un sistema eficiente de base de datos, por lo general se refieren a algoritmos, y conceptos matemáticos.

**Bases de Datos Relacionales:** Su idea fundamental es el uso de "relaciones". Estas relaciones podrían considerarse en forma lógica como conjuntos de datos llamados "tuplas"<sup>8</sup>. Pese a que esta es la teoría de las bases de datos relacionales creadas por Edgar Frank Codd, la mayoría de las veces se conceptualiza de una manera más fácil de imaginar. Esto es pensando en cada relación como si fuese una tabla que está compuesta por registros (las filas de una tabla), que representarían las tuplas, y campos (las columnas de una tabla).

En este modelo, el lugar y la forma en que se almacenen los datos no tienen relevancia (a diferencia de otros modelos como el jerárquico y el de red). Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar para un usuario casual de la base de datos. La información puede ser recuperada o almacenada por medio de "consultas" que ofrecen una amplia flexibilidad y poder para administrar la información.

El lenguaje más común para construir las consultas a bases de datos relacionales es SQL, Structured Query Language o Lenguaje Estructurado de Consultas, un estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales.

**El modelo Entidad Relación:** Este modelo está basado en una percepción del mundo real, que consta de una colección de objetos básicos, llamados entidades, y de relaciones entre estos objetos.

Una entidad es todo aquello que exhibe autonomía, diferenciación y existencia en el mundo real que es distinguible de otros objetos. Por ejemplo cada alumno es una entidad y los cursos a los que está inscrito también pueden ser considerados entidades.

Las entidades se describen en la base de datos mediante atributos. Por ejemplo los atributos alumno\_nombre, alumno\_fecha\_de\_nacimiento pueden describir una entidad alumno.

<sup>8</sup> Tupla: Es una función finita que *mapea* (asocia unívocamente) los nombres con algunos valores.



Adicionalmente `alumno_id`, se usa para identificar unívocamente a los alumnos (dado que puede existir en el mundo real varios alumnos con el mismo nombre y fecha de nacimiento), de esta manera se asigna un identificador único para cada alumno.

Una relación es una asociación entre varias entidades, Por ejemplo una relación puede asociar un alumno con cada curso inscrito. El conjunto de todas las entidades del mismo tipo, y el conjunto de todas las relaciones del mismo tipo, se denominan respectivamente conjunto de entidades y conjunto de relaciones.

La estructura lógica general de una base de datos se puede expresar gráficamente mediante un diagrama E-R. Cada componente se etiqueta con la entidad o relación que representa. Para crear visual y físicamente el modelo (diagrama) se utilizó el MySQL Workbench, que es un software creado específicamente para realizar esta tarea.

### 2.3.2 Análisis de Requisitos

Lo primero que se tuvo que hacer fue listar todos los datos que se necesitaban almacenar. Se dividieron los datos en 5 tipos:

<b>Para el administrador:</b>	Clave Contraseña
<b>Para el alumno:</b>	Nombre Número de cuenta / Número de trabajador Correo electrónico Contraseña Subtipo
<b>Para el ponente:</b>	Nombre Número de cuenta / Número de trabajador Correo electrónico Contraseña Subtipo
<b>Para el curso:</b>	Temario Horario Cupo Mínimo Cupo Máximo Tema Subtipo Ponente Lista de alumnos inscritos Estado Requisitos de ingreso Grupo Sala
<b>Para el sistema:</b>	Semestre Fecha de inicio y termino del período de los cursos Fecha y hora de inscripción Número máximo de cursos Estado del semestre

**Tabla 2-1 Datos Necesarios para la Base de Datos**





Cabe mencionar que en todas las tablas se utilizó la misma nomenclatura a excepción de la tabla del administrador (User), esto es con la finalidad de incrementar la seguridad.

De acuerdo al modelo anterior se muestra ahora el diccionario de datos.

Entidad	Atributo	Llave	Tipo	Tamaño
<b>Alumno</b>	alumno_id	PK	int	11
	usuario_tipo_id	FK	int	11
	alumno_nombre		varchar	60
	alumno_numero		char	10
	alumno_passwd		varchar	50
	alumno_correo		varchar	50
<b>Curso</b>	curso_id	PK	int	11
	estado_id	FK	int	11
	ponente_id	FK	int	11
	semestre_id	FK	int	11
	tema_id	FK	int	11
	usuario_tipo_id	FK	int	11
	curso_nombre		varchar	100
	curso_minimo		int	11
	curso_maximo		int	11
	curso_dias		char	6
	curso_hora_inicio		char	5
	curso_hora_termino		char	5
	curso_temario		text	0
	curso_requisitos		text	0
	curso_hora		datetime	0
	curso_grupo		varchar	10
curso_sala		varchar	10	
<b>Estado</b>	estado_id	PK	int	11
	estado_nombre		varchar	20

**Tabla 2-2 Diccionario de Datos (1)**



Entidad	Atributo	Llave	Tipo	Tamaño
<b>Inscripcion</b>	inscripcion_id	PK	int	11
	curso_id	FK	int	11
	alumno_id	FK	int	11
	inscripcion_calificacion		float	0
	inscripción_hora		datetime	0
<b>Ponente</b>	ponente_id	PK	int	11
	usuario_tipo_id	FK	int	11
	ponente_nombre		varchar	60
	ponente_numero		char	10
	ponente_passwd		varchar	50
	ponente_correo		varchar	50
<b>Semestre</b>	semestre_id	PK	int	11
	semestre_nombre		char	6
	semestre_registro_inicio		datetime	0
	semestre_registro_termino		datetime	0
	semestre_inscripcion_inicio		datetime	0
	semestre_inscripcion_termino		datetime	0
	semestre_inicio		date	0
	semestre_termino		date	0
	semestre_maximo		int	11
	semestre_activo		char	1
<b>Usuario_tipo</b>	usuario_tipo_id	PK	int	11
	usuario_tipo_nombre		varchar	20
<b>Tema</b>	tema_id	PK	int	11
	tema_nombre		varchar	20
<b>User</b>	Username	PK	varchar	16
	Passwd		varchar	50

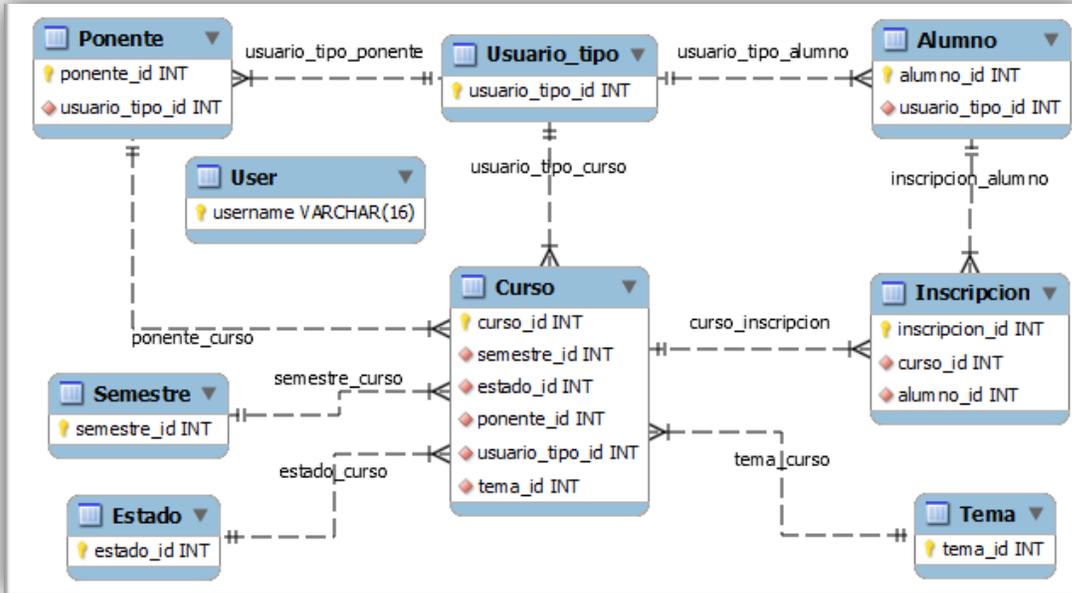
**Tabla 2-3 Diccionario de Datos (2)**

Cabe hacer mención de que durante el modelado se realizó un estudio detallado del tipo y tamaño más conveniente para cada dato, con la finalidad de incrementar la funcionalidad y respuesta de la base de datos, para tal fin se consideraron todos los posibles escenarios.



### 2.3.4 Diseño Lógico

Las relaciones creadas son necesarias para que los datos se comuniquen, ver Figura 2-3.



**Figura 2-3 Modelo Entidad-Relación (Elaboración Propia)**

A continuación se describirán cada una de las relaciones establecidas.

**Relaciones usuario\_tipo\_ponente y usuario\_tipo\_alumno:** Permiten conocer qué subtipo de ponente o subtipo de alumno es el usuario.

**Relación usuario\_tipo\_curso:** Facilita la distinción de los subtipos de usuario a los que va dirigido cada curso.

**Relaciones inscripcion\_alumno y curso\_inscripcion:** Son esenciales, ya que son las que permiten comunicar los elementos de la tabla alumno con los de la tabla curso. Esto quiere decir que la aparición de cada alumno en la tabla inscripción, indica que el alumno está inscrito en el curso con el que aparece.

**Relación ponente\_curso:** Es indispensable para que cada curso sea asignado a su ponente.

**Relación semestre\_curso:** Cuando un usuario inicia sesión en el sistema necesita ver los cursos que actualmente se llevan a cabo, por ejemplo, quizá un ponente desea impartir el mismo curso que impartió hace tres semestres, y si no existiera esta relación se mostrarían los cursos de todos los semestres.

**Relación estado\_curso:** Permite que el curso tenga varios estados, esto es necesario para poder activar los cursos, ya que mientras el estado actual del curso sea registrado, el curso aparecerá en la lista de cursos que necesitan ser activados. Ahora bien cuando el estado del curso sea activo, este ya no aparecerá



en la lista de cursos por activar, en cambio aparecerá en la lista de cursos inscribibles.

**Relación tema\_curso:** Es útil para organizar por tema los cursos, lo cual permite que cuando estos se desplieguen en la lista de inscribibles, se puedan acomodar rápidamente por tema.

## 2.4 ASPECTOS TÉCNICOS

Para desarrollar este sistema se utilizaron las siguientes herramientas:

**XAMPP:** Este es un servidor virtual que permite ejecutar el código PHP, al mismo tiempo que proporciona un manejador de base de datos (MySQL), en un entorno local.

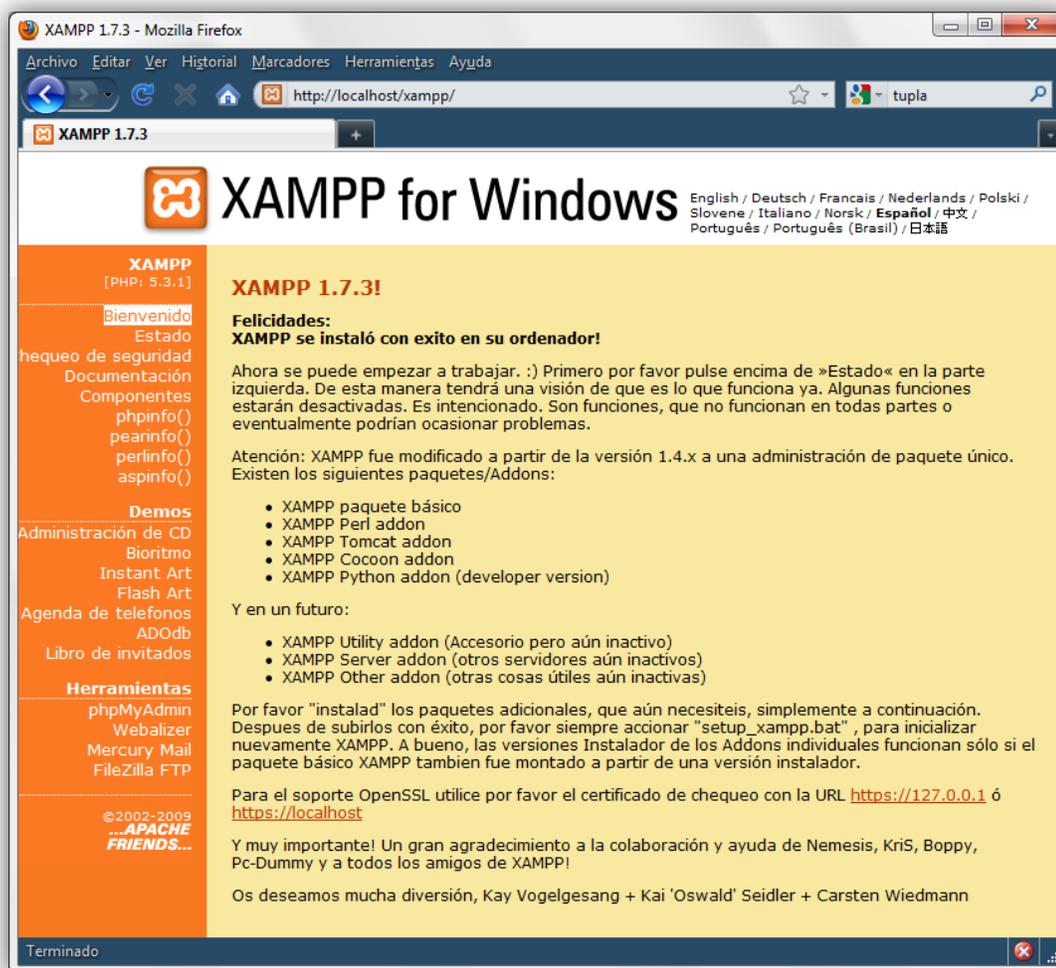
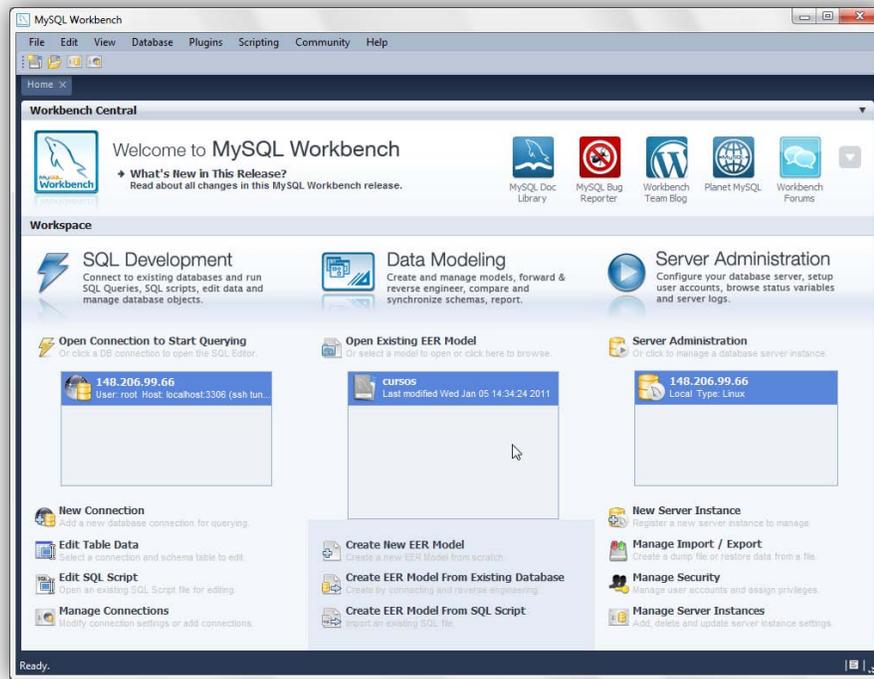


Figura 2-4 XAMPP

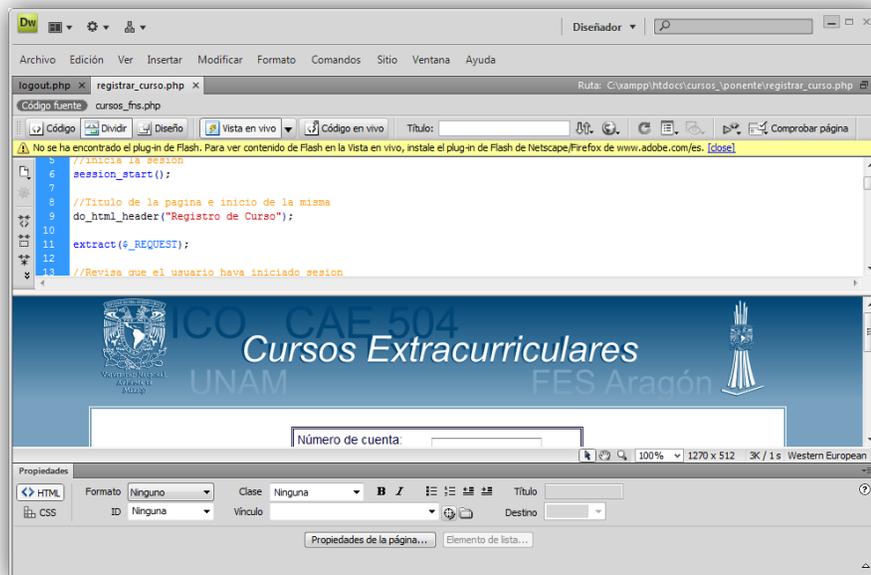


**MySQL Workbench:** Programa que es utilizado para el modelado, la manipulación y administración de bases de datos.



**Figura 2-5 MySQL Workbench**

**Adobe Dreamweaver:** IDE dedicado a la creación de páginas web, este programa soporta eficazmente la programación del código en lenguaje PHP.



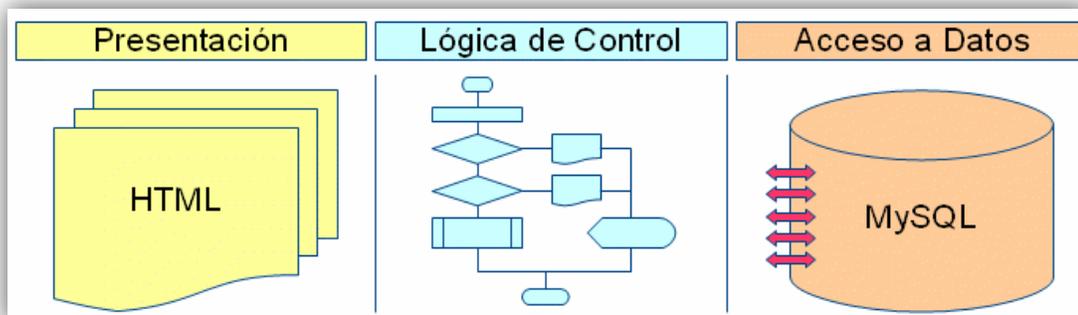
**Figura 2-6 Adobe Dreamweaver**



Las herramientas fueron elegidas debido a su estabilidad y en el caso de las primeras dos también fue considerado el costo ya que estas son software libre.

## 2.5 ARQUITECTURA DEL SISTEMA

La arquitectura del desarrollo del sistema fue de tres capas, lo que hace que el desarrollo de los módulos se vuelva independiente y de rápido acceso, esta arquitectura se muestra en la figura 2-7.



**Figura 2-7 Arquitectura de Tres Capas**

### 2.5.1 Capa de Presentación

Esta es la que genera el entorno gráfico para el usuario, visto de otra forma es la que genera el código HTML, que es desplegado en el explorador, este incluye imágenes, botones, cajas de texto, tablas etc.

### 2.5.2 Capa de Lógica de Control

En ella se llevan a cabo los algoritmos, validaciones y métodos que son necesarios para resolver los problemas. Es decir esta etapa se encarga de manipular los datos, por tanto, es la capa que interactúa con las otras dos. Donde se comunica con la capa de acceso a datos para obtener o enviar la información de la Base de Datos. La procesa y finalmente la envía a la capa de presentación con la finalidad de que esta genere el entorno gráfico.

### 2.5.3 Capa de Acceso a Datos

Esta capa es la encargada de interactuar directamente con el manejador de la base de datos, enviará y recibirá datos a esta, atendiendo a las peticiones generadas en la capa de control.

## 2.6 DISEÑO DE PANTALLAS

Esta parece ser una tarea fácil, sin embargo más allá de que el sistema tenga un buen aspecto, lo más importante, es que al crear el diseño se debe de analizar

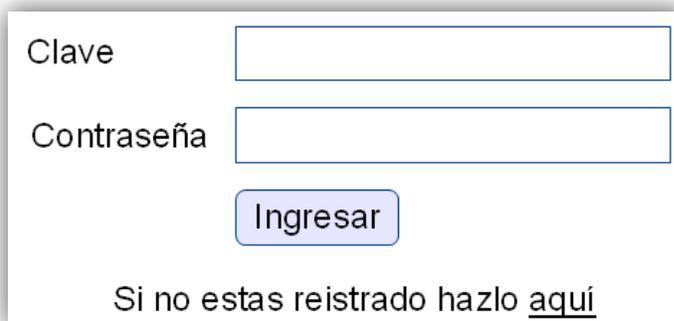


cuidadosamente el lugar y forma apropiada para mostrar los datos en las páginas, ya que de ello dependerá el correcto uso por parte del usuario.

Lo que los usuarios necesitan en una vista es:

- Una apariencia clara y ordenada.
- Indicaciones claras de lo que se debe hacer.
- Indicaciones claras de lo que se puede hacer.
- Indicaciones de lo que se está viendo.
- Que cada uno de los elementos de la pantalla (botones, cajas de texto, mensajes, texto), tengan un significado específico para el usuario
- Que cada elemento deba realizar una tarea.

Una vez tomado en cuenta lo anterior se realizaron los bosquejos de cómo deberían verse las pantallas. Ver figuras de la 2-8 a la 2-24.

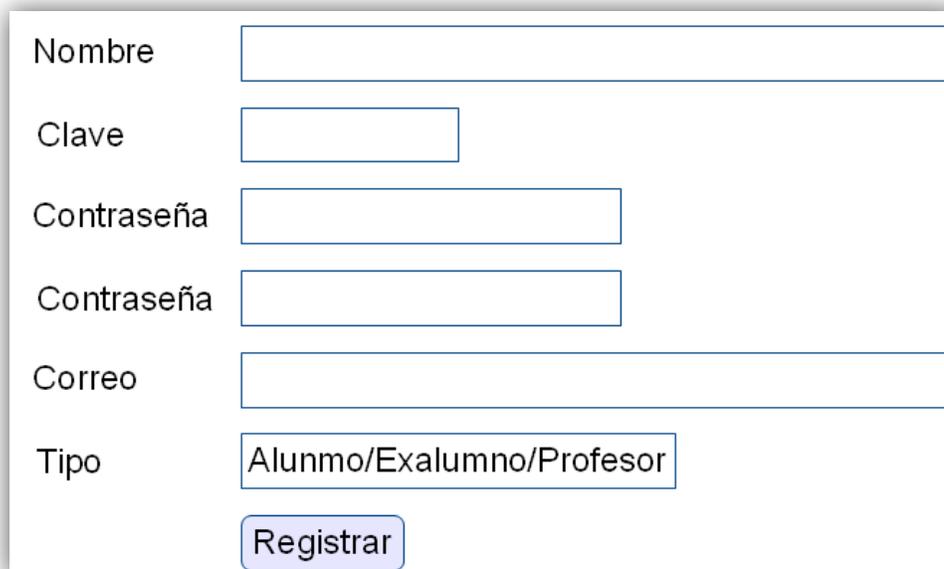


Clave

Contraseña

Si no estas reistrado hazlo [aquí](#)

**Figura 2-8 Formulario para Ingreso al Sistema**



Nombre

Clave

Contraseña

Contraseña

Correo

Tipo

**Figura 2-9 Formulario para Registro**



Semestre	Acciones de Usuario	Datos de Usuario opciones
----------	---------------------	------------------------------

**Figura 2-10 Menú de Usuario**

<b>Cursos Inscritos</b>	No. Máximo de cursos 3
Curso 1	Lun Mier 13:00 a 15:00
Curso 3	Mar Jue 16:00 a 17:30

**Figura 2-11 Menú de Cursos Inscritos**

Vieja Contraseña

Nueva Contraseña

Nueva Contraseña

**Figura 2-12 Formulario para Cambio de Contraseña**

Nombre

Clave

Correo

Tipo

**Figura 2-13 Formulario para Cambio de Datos**



Programación				
Curso	Grupo	Cupo Mínimo	Cupo Máximo	Alumnos Inscritos
Curso a	Grupo 7	5	15	12
Curso a	Grupo 3	5	20	7
Curso e	Grupo 4	Cupo lleno	18	18

Hardware				
Curso	Grupo	Cupo Mínimo	Cupo Máximo	Alumnos Inscritos
Curso b	Grupo 2	5	25	22
Curso c	Grupo 1	5	25	9

Paquetería				
Curso	Grupo	Cupo Mínimo	Cupo Máximo	Alumnos Inscritos
Curso d	Grupo 5	5	20	16

**Figura 2-14 Lista de Cursos**

Curso	Curso b
Grupo	Grupo 1
Sala	Sala 4
Tema	Redes
Horario	Martes y Jueves de 9:00 a 11:00
Cupo	30 personas
Temario	1.- ..... 2.- .....
Requisitos	Ninguno
<input type="button" value="Inscribir"/>	

**Figura 2-15 Inscripción del Curso**



Nombre

Tema

Dirigido a

Horario  a  L M M J V

Temario

Requisitos

**Figura 2-16 Formulario para Registro de Cursos**

Curso d

Nombre	Correo	Calificación
<u>Alumno 1</u>	a@q.com	<input type="text" value="7"/>
<u>Alumno 2</u>	b@w.com	<input type="text" value="9"/>
<u>Alumno 3</u>	c@e.com	<input type="text"/>

**Figura 2-17 Formulario para Calificar Alumnos**

Curso d

Nombre	Correo	Calificación
<u>Alumno 1</u>	a@q.com	7
<u>Alumno 2</u>	b@w.com	9
<u>Alumno 3</u>	c@e.com	9

**Figura 2-18 Lista de Calificaciones**



Nombre	<input type="text"/>
Registro inicio	<input type="text" value="DD-MM-AAAA HH:MM:SS"/>
Registro termino	<input type="text" value="DD-MM-AAAA HH:MM:SS"/>
Inscripción inicio	<input type="text" value="DD-MM-AAAA HH:MM:SS"/>
Inscripción termino	<input type="text" value="DD-MM-AAAA HH:MM:SS"/>
Cursos Inicio	<input type="text" value="DD-MM-AAAA"/>
Cursos Termino	<input type="text" value="DD-MM-AAAA"/>
Máximo de Cursos	<input type="text"/>
<input type="button" value="Cambiar"/>	

**Figura 2-19 Formulario para Cambio de Datos de Semestre**

Curso	Grupo	Sala	Cupo Mínimo	Cupo Máximo	
<u>Curso a</u>	<input type="text" value="Grupo 11"/>	<input type="text"/>	<input type="text" value="5"/>	<input type="text" value="30"/>	<input type="button" value="Activar"/>
<u>Curso a</u>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="Activar"/>
<u>Curso d</u>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="Activar"/>

**Figura 2-20 Formulario para Activación de Cursos**

Nombre	Ponente 3
Clave	123456
Correo	p@unam.mx
Cursos	<u>Curso a - Grupo 2</u> <u>Curso a - Grupo 4</u> <u>Curso e - Grupo 6</u>

**Figura 2-21 Datos del Ponente**



Curso - Grupo	Alumnos Inscritos	
<u>Curso a – Grupo 4</u>	12	Descargar
<u>Curso a – Gurpo 8</u>	23	Descargar
<u>Curso d – Grupo 6</u>	10	Descargar

**Figura 2-22 Formulario para Descargar Listas**

Nombre	<input type="text"/>
Clave	<input type="text"/>
	<input type="button" value="Buscar"/>

**Figura 2-23 Formulario para Búsqueda de Usuarios**

Nombre	Clave	
<u>Alumno 11</u>	1234	Restablecer
<u>Alumno 12</u>	1235	Restablecer
<u>Alumno 21</u>	1245	Restablecer

**Figura 2-24 Formulario para Restablecer Contraseñas de Usuarios**

## 2.7 COMPATIBILIDAD

Actualmente existen varios estándares en cuanto a código HTML se refiere, esto genera incompatibilidades con los diversos exploradores que existen en el mercado, y se ve reflejado directamente en la forma en que los contenidos de una página web son desplegados en la misma.

Hasta hace algunos años se utilizaba un estándar conocido como HTML4.0 el cual permitía que fueran introducidos dentro del código todas las configuraciones que darían el formato grafico a la página web.



Con la llegada de CSS (Cascade Style Sheet – Hoja de Estilo en Cascada) y XHTML esto cambio, ahora se debe tener un archivo CSS donde se declaran todas las configuraciones gráficas para cada elemento que ha de constituir la página web (Color, Tamaño, Tipografía, Ubicación, Grosor, Fondo, Margen, Espacio, etc.).

Una vez que se tiene este archivo CSS se crea un archivo HTML que contendrá el código necesario para crear los elementos de la página, para ello se deberá utilizar código HTML puro y sin formato visual. Este código deberá adaptarse a las especificaciones del lenguaje de marcado XML<sup>9</sup>, de ahí que su nombre cambie a XHTML.

Cuando se despliega la página en un navegador se obtiene como resultado final una página que podrá ser visualizada de la misma forma en un navegador que en otro. Además, de ayudar en la construcción rápida de todo un sitio, esto debido a que el archivo CSS será utilizado para todas la páginas del mismo.

Cuando los programadores diseñan marcado de presentación se duplica el ancho de banda necesario para ver u ofrecer una página. También reduce su accesibilidad para motores de búsqueda, navegadores y dispositivos no tradicionales.

Uno de los principales problemas que se presentan en cuanto a compatibilidad se refiere en diversos navegadores, es el uso de Scripts (Java script, VBscript) y la tecnología de Adobe conocida como Flash. Las cuales no son soportadas por todos, por lo que son desplegadas de formas muy diferentes ó simplemente no son visibles por el usuario. Para evitar esto se decidió no utilizar estas tecnologías en el sistema.

Las tecnologías creadas por el World Wide Web Consortium (W3C)<sup>10</sup> permiten que se puedan diseñar y crear sitios con un funcionamiento duradero, aunque cambien los estándares y navegadores.

Los estándares permiten:

- Tener mayor control sobre el diseño visual.
- Desarrollar comportamientos sofisticados que funcionen de la misma forma en diferentes navegadores.
- Cumplir las normas y directrices de accesibilidad sin sacrificar el aspecto visual o el rendimiento.
- Admitir varios navegadores sin preocuparse de tener que crear diferentes versiones.

---

<sup>9</sup> XML (eXtensible Markup Language - Lenguaje de Marcas Extensible): Es un metalenguaje extensible de etiquetas. Es una simplificación y adaptación del SGML y permite definir la gramática de lenguajes específicos.

<sup>10</sup> W3C. Creado en 1994, el W3C trabaja con especificaciones y directrices con la intención de promover la evolución de la web. Entre las especificaciones desarrolladas por el W3C destacan el HTML, CSS, XHTML y el modelo de objetos de documento DOM entre otras muchas.



- Admitir dispositivos no tradicionales, como teléfonos móviles compatibles, tablets y consolas de videojuego portátiles.
- Conseguir sofisticadas versiones impresas de cualquier página sin crear versiones "aptas para impresión".
- Separar estilo de estructura y comportamiento.
- Realizar la transición entre el lenguaje del pasado HTML, al marcado basado en XML.
- Garantizar que los sitios diseñados de esta forma funcionarán en los futuros navegadores y dispositivos. Como es el caso de HTML5.

## 2.8 SEGURIDAD

Un aspecto fundamental para que un sistema de información funcione correctamente es la seguridad. Esta tiene como fin principal evitar que algún usuario modifique la información, ya sea para beneficio propio o solo con la finalidad de crear errores.

Para conseguir que un sistema no sufra este tipo de percances, se deben implementar las siguientes medidas:

### 2.8.1 Identificación de Usuario

Antes de permitir que un usuario acceda al sistema se debe solicitar que se identifique, esto con la finalidad de saber qué acciones puede realizar, pero sobre todo quién las realiza.

Si el usuario no se identifica, no puede acceder a ningún tipo de información y solamente recibirá un mensaje con la petición de que lo haga.

### 2.8.2 Identificación de Tipo de Usuario

Debido a que cada tipo de usuario tiene permiso a diferentes acciones y a cierto tipo de información, es necesario para el sistema conocer el tipo de usuario que se ha identificado.

En caso de que el usuario se haya identificado como alumno está claro que este no debe realizar aquellas operaciones que sólo le están permitidas al usuario del tipo ponente, y mucho menos a las que tiene acceso el usuario del tipo administrador.

Si el usuario ingresa a una página a la cual no debe tener acceso, el sistema negará el acceso y notificará con un mensaje de error.

### 2.8.3 Detección de Código Mal Intencionado

A pesar de que el código del sistema se compila y se ejecuta en el servidor, existen formas en las que un usuario puede ingresar cierto código para que este se ejecute en el momento de solicitar una acción.



La forma de evitar esto es realizando un análisis de la información recibida antes de realizar acciones con esta. El análisis se encargará de discriminar posibles pedazos de código, que en un determinado momento pudieran ejecutarse.

#### *2.8.4 Uso de Permisos en la Base de Datos*

Las bases de datos actualmente no se restringen únicamente al almacenamiento de información, estas cuentan con un sinnúmero de configuraciones posibles. Una de las utilidades que nos proporcionan es el uso de niveles de acceso de usuario, estos son creados con la finalidad de permitir o restringir las acciones que se pueden realizar.

Los permisos pueden ser tan restrictivos como sean necesarios. Por ejemplo, se puede restringir a un usuario para que solo ejecute peticiones para la obtención de datos y le sea negada la actualización y eliminación de los mismos en una base, en una o varias tablas e incluso solamente en una columna. Se pueden hacer todas las combinaciones imaginables y con ello aumentar la seguridad de la forma deseada.

#### *2.8.5 Registro de Acciones*

Otro método que se ocupa para aumentar la seguridad es el realizar un registro de las acciones que cada usuario realiza, eso se consigue por medio del almacenamiento de estos datos en el momento de que cada usuario realice operaciones.

En el sistema que aquí se describe, se usó este método para almacenar fecha y hora de inscripción a cada curso, esto podría ayudar a verificar que un alumno se inscribió dentro de los horarios preestablecidos.





# *Capítulo 3*

## *DESARROLLO DEL SISTEMA*

*“La educación empieza con la vida, y no acaba sino con la muerte”*

*José Martí*



Es aquí donde se creará físicamente el sistema, aquí finalmente se tendrá que utilizar el lenguaje de programación para crear los archivos que tendrán el código necesario para que los modelos que fueron descritos anteriormente interactúen entre ellos, con la base de datos y con el usuario, para permitir que el proceso de inscripción y demás funciones del sistema se lleven a cabo.

### 3.1 ADECUACIÓN DEL ENTORNO DE GENERACIÓN

Es necesario que se tengan preparados los requerimientos físicos necesarios para poder generar el código. En otras palabras se necesita tener la computadora o computadoras necesarias y que estas a su vez tengan instalado el software necesario para el desarrollo.

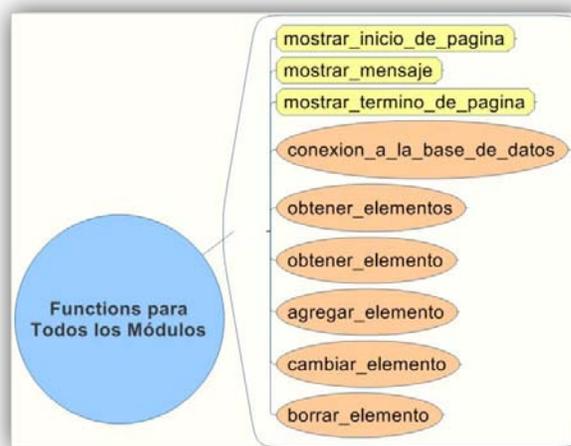
Para la tarea específica se ocupó una computadora genérica, que tenía instalados los siguientes programas: Dreamweaver, XAMMP, MySQL Workbench, Mozilla Firefox, Internet Explorer, Google Chrome, Opera, Safari y Office.

### 3.2 DESARROLLO DE MÓDULOS

Con el afán de evitar confusión entre la palabra función que determina el funcionamiento y el término función utilizado para referirse a un proceso computacional, se decide utilizar el término **function** para este último.

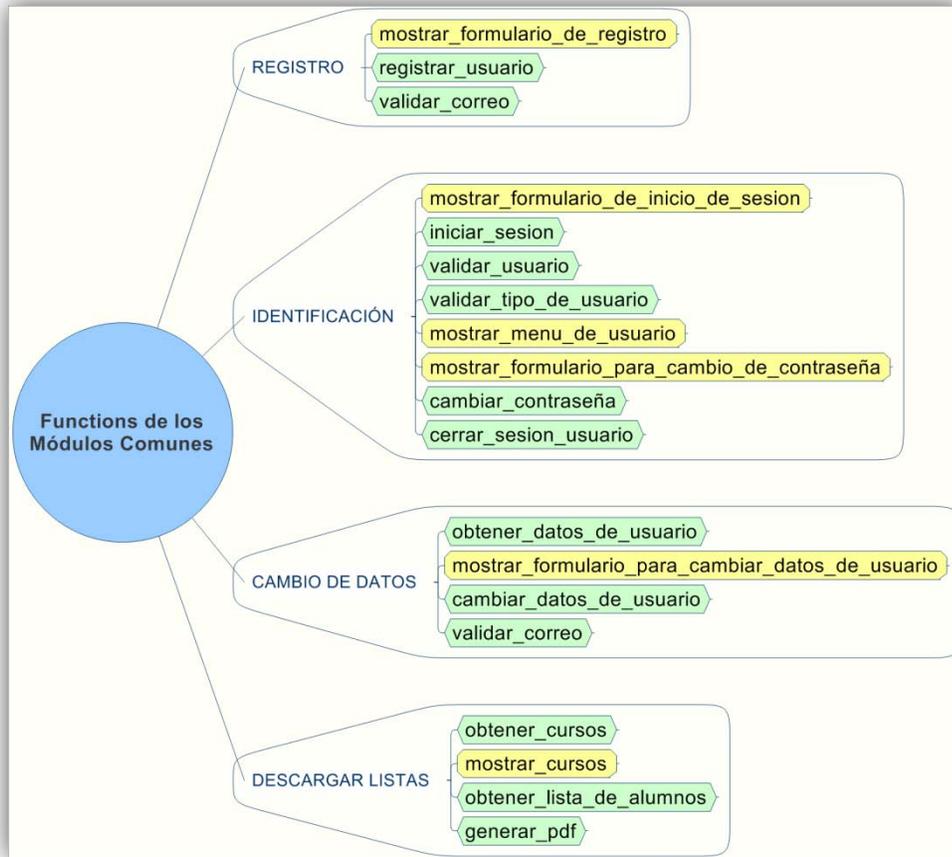
Las functions contendrán pedazos de código que resolverán partes esenciales de cada módulo. Cada function recibe datos de entrada, los manipula y nos devuelve un conjunto de datos, un valor booleano o simplemente realiza acciones como generar el código HTML de una página.

Gracias a la reutilización de código, una misma function puede ser utilizada por varios módulos. A continuación se listarán las funciones que cada módulo ocupa para realizar todas sus tareas. Ver figuras 3-1 y 3-6.

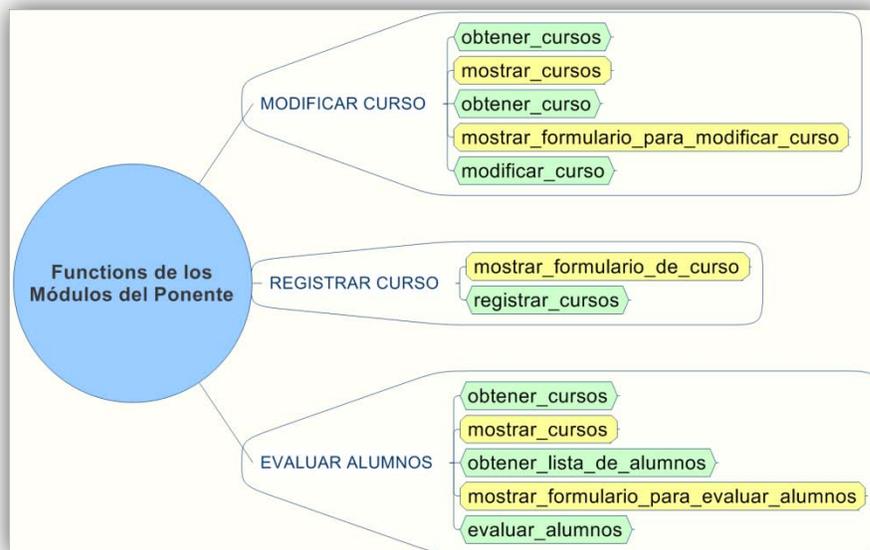


**Figura 3-1** Functions para Todos los Módulos



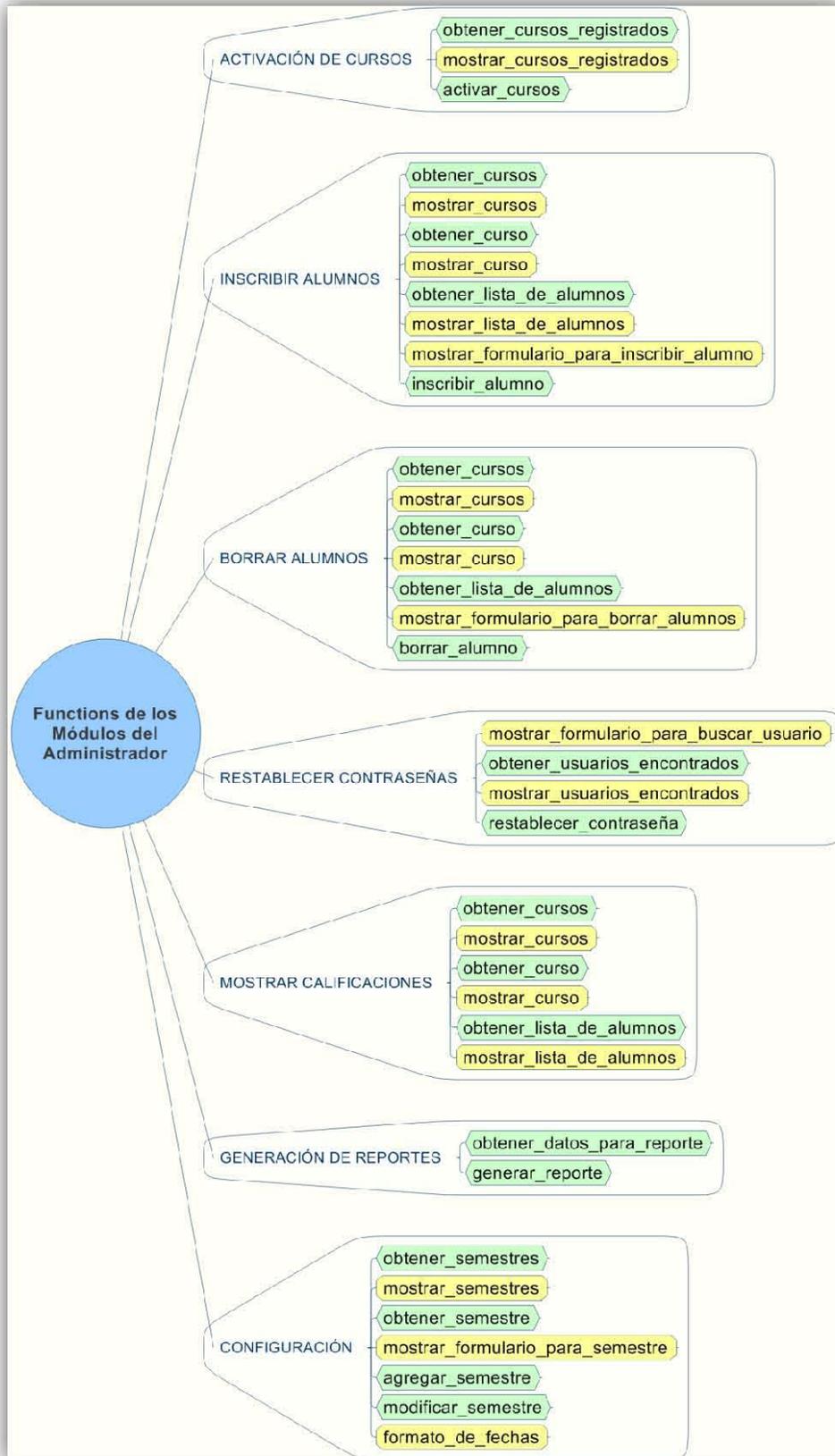


**Figura 3-2** *Funcions de los Módulos Comunes*



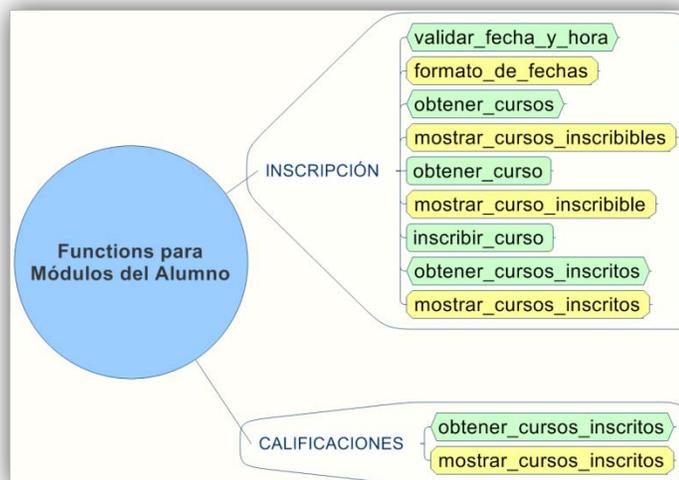
**Figura 3-3** *Funcions de los Módulos del Ponente*



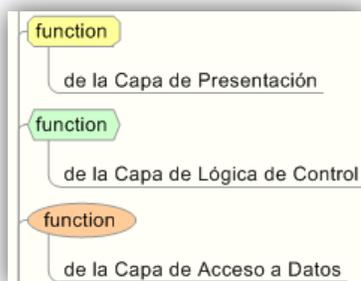


**Figura 3-4 Funcions de los Módulos del Administrador**





**Figura 3-5 Functions de los Módulos del Alumno**



**Figura 3-6 Simbología de los Mapas de las Functions**

## 3.3 DESCRIPCIÓN DE FUNCIONES (FUNCTIONS)

Debido a que se manejó la programación por capas, se describirán todas las functions ocupadas agrupándolas por la capa a la que cada una pertenece.

### 3.3.1 Functions de la Capa de Presentación

Estas functions como ya fue mencionado reciben los datos desde la capa de lógica de control, su tarea será crear en código HTML; los formularios, las cajas de texto, los botones, etc.

**Function mostrar\_inicio\_de\_página:** Imprime en el archivo de salida la cabecera de html. Donde se generan las etiquetas `<html><head>...</head><body>`. Con lo que se deja abierto el cuerpo de la página para que se agregue lo necesario por medio de las demás functions de la capa de presentación.

**Function mostrar\_termino\_de\_pagina:** Genera el final de la página, donde se generan las etiquetas `</body></html>`.



**Function mostrar\_mensaje:** Es una function creada para enviarle un mensaje al usuario, este mensaje puede ser: una recomendación, un mensaje de error, un mensaje de éxito, etc. La function recibe el mensaje como un parámetro.

**Function mostrar\_formulario\_de\_registro:** Generará un sencillo formulario con cuatro campos para que los usuarios se puedan registrar. En estos campos se deberán suministrar nombre, clave, correo y subtipo de usuario.

**Function mostrar\_formulario\_de\_inicio\_de\_sesión:** Se utiliza para generar un formulario básico con dos campos; uno para la clave y otro para la contraseña. Finalmente se genera un botón que servirá para enviar los datos y solicitar el acceso al sistema.

**Function mostrar\_menu\_de\_usuario:** Es utilizado para generar el menú de opciones para el usuario, se debe tomar en cuenta que cada tipo de usuario tiene acceso a diferentes funciones del sistema, por tanto los menús deben ser generados de acuerdo al usuario que haya iniciado la sesión.

Además, las opciones a las que el usuario va teniendo acceso se cambian de acuerdo a la etapa en la que se encuentre (Registro, Inscripción, Terminó, etc.).

**Function mostrar\_formulario\_para\_cambio\_de\_contraseña:** Es encargada de generar un formulario simple con tres campos y un botón. En estos campos el usuario deberá de ingresar la contraseña actual y la nueva contraseña dos veces.

**Function mostrar\_formulario\_para\_cambiar\_datos\_de\_usuario:** Recibe los datos actuales del usuario y los vierte sobre un formulario, donde el usuario podrá cambiar sus datos, por último generará un botón que servirá para solicitar que se ejecute la actualización.

**Function mostrar\_cursos:** Recibe un arreglo de cursos y su tarea es generar la lista de estos con sus datos correspondientes.

**Function mostrar\_curso:** Recibirá los datos de un curso y desplegará al usuario dicha información en forma ordenada.

**Function mostrar\_cursos\_inscribibles:** Recibirá un arreglo de los cursos que aún tienen cupo, con ellos generará una lista con las características de los mismos.

**Function mostrar\_curso\_inscribible:** Recibirá un arreglo con los datos de un determinado curso, al final generará un botón con el que se podrá solicitar la inscripción en el mismo.

**Function mostrar\_cursos\_inscritos:** Recibe un arreglo de datos de los cursos en los cuales el alumno se encuentra actualmente inscrito. Su tarea será generar una lista con estos datos.



**Function mostrar\_formulario\_de\_curso:** Su tarea es generar un formulario sencillo con los campos necesarios para que el ponente ingrese los datos correspondientes al curso que desea ingresar.

**Function mostrar\_formulario\_para\_modificar\_curso:** Recibe los datos correspondientes a un curso específico y su tarea es verterlos sobre un formulario por medio del cual el usuario podrá modificar estos datos, por último generará un botón que servirá para solicitar el cambio de los datos.

Un elemento que se debe contemplar es el tipo de usuario que utiliza esta función ya que si el tipo de usuario es administrador este solo tendrá acceso a la modificación de los datos: sala, grupo y cupo. En cambio para el usuario ponente todos los demás excepto los anteriores.

**Function mostrar\_formulario\_para\_evaluar\_alumnos:** Recibirá un arreglo de datos pertenecientes a los alumnos inscritos en un curso específico. Su tarea será la de generar un formulario con estos datos para que genere una lista con los alumnos inscritos, agregando un campo delante de cada alumno, donde el ponente ingresará la calificación obtenida por cada alumno. Finalmente generará un botón que servirá para ejecutar la actualización de los datos.

**Function mostrar\_cursos\_registrados:** Recibe un arreglo de datos de los cursos que fueron agregados por los ponentes y esperan ser validados. Su función radica en proporcionar la lista de los cursos que cumplen esta característica y para cada uno de ellos un botón con la leyenda activar, que servirá para activar cada curso.

**Function mostrar\_formulario\_para\_inscribir\_alumno:** Creará un formulario con un campo y un botón con la leyenda "Inscribir" al final de la lista de un curso. De tal forma que si se necesita agregar un alumno a la lista de alumnos inscritos en el curso, bastará con proporcionar la clave de usuario en el campo.

**Function mostrar\_lista\_de\_alumnos:** Recibirá un arreglo con los datos de los alumnos inscritos de un curso específico y los utilizará para generar una lista de alumnos. Delante de cada uno se generará un botón con la leyenda borrar, de tal forma que si se desea dar de baja del curso a un alumno, bastará con presionar el botón para el alumno requerido.

**Function mostrar\_formulario\_para\_borrar\_alumnos:** Recibe un arreglo de alumnos de un determinado curso, con él genera una lista de los alumnos inscritos en el curso, delante de cada alumno se genera un botón con la leyenda Borrar, por medio del que se solicita borrar al alumno de la lista.

**Function mostrar\_formulario\_para\_buscar\_usuario:** Generará un formulario con dos campos, uno para el nombre y otro para la clave, al final generará un botón para que se pueda ejecutar la acción.

**Function mostrar\_usuarios\_encontrados:** Recibe un arreglo de datos con los usuarios que fueron encontrados en la búsqueda. Su tarea será generar la lista de



los usuarios y delante de cada uno generará un botón para realizar la acción de restablecer la contraseña para el elemento.

**Function mostrar\_semestres:** Recibe un arreglo de datos de semestres con los cuales genera una lista.

**Function mostrar\_formulario\_para\_semestre:** Recibe los datos de un semestre específico y los agrega a un formulario que le permite al usuario realizar los cambios que se necesiten. Por último genera un botón necesario para solicitar la acción de cambio de datos.

**Function formato\_de\_fechas:** Es un conjunto de funciones utilizadas para realizar todas las conversiones de formato para las fechas utilizadas.

### *3.3.2 Functions de la Capa de Lógica de Control*

Las functions que ahora se describirán son las encargadas de realizar las acciones y validaciones necesarias para que el sistema realice todas las tareas. Se comunican con otras de la capa de acceso a datos, por medio de las que envía y recibe datos desde la base de datos. Y para poder interactuar con los usuarios envía la información a otras functions de la capa de presentación.

**Function registrar\_usuario:** Aquí se reciben los datos de los usuarios que desean registrarse, valida que los datos sean correctos, crea una consulta con la que solicita que se agregue el elemento, esta consulta se envía a la capa de acceso a datos, de la que se espera una respuesta de confirmación.

La capa de acceso a datos devuelve uno de los dos posibles valores:

- Verdadero: En caso de que se haya registrado efectivamente.
- Falso: En caso de haber ocurrido un error.

**Function validar\_correo:** Revisará las cadenas recibidas, para validar una posible cuenta de correo, finalmente devolverá un valor booleano a la function que la llamó, este valor será el resultado de la validación efectuada.

**Function iniciar\_sesion:** Recibe los datos clave y contraseña, su tarea es verificar que estos concuerden con los que existen en la base, para tal fin envía una consulta a la base por medio de la capa de acceso a datos.

Si la respuesta recibida es un valor verdadero, esta function almacena los datos del usuario temporalmente en el servidor, con ello el usuario inicia una sesión, por otro lado si la respuesta es un valor falso, se enviará un mensaje al usuario para indicarle que no se ha podido iniciar una sesión.

**Function validar\_usuario:** Se ejecutará cada vez que se genere una nueva página o se refresque la actual, esto se hace con la finalidad de que no se pueda acceder a la información del sistema sin antes haber iniciado sesión en el mismo.



Para conseguir tal fin deberá verificar que el usuario haya iniciado sesión, en el caso de que esto sea cierto le permitirá continuar con la carga de la página, de lo contrario mandará un mensaje de error y solicitará que el usuario inicie una sesión.

**Function validar\_tipo\_de\_usuario:** Esta validación también se realizará en cada página. Sirve para conocer el tipo de usuario que ha iniciado una sesión. Esta información ayudará al sistema a saber las opciones de las que dispone este usuario. En otras palabras, se utilizará para permitir o denegar el acceso a las funciones del sistema, dependiendo de los permisos que el usuario tenga.

**Function cambiar\_contraseña:** Es la encargada de verificar que la contraseña proporcionada sea correcta y también que la nueva sea válida. Una vez hecho esto realizará el cambio en la base de datos, para ello enviará una consulta a la base por medio de la capa de acceso. Y esperará un valor booleano, para confirmar que el cambio se realizó efectivamente.

**Function cerrar\_sesión\_usuario:** Se utilizará para que el usuario pueda cerrar su sesión en el sistema, con ello se asegura que nadie más pueda realizar modificaciones en su lugar.

**Function obtener\_datos\_de\_usuario:** Solicita por medio de la capa de acceso a datos, los datos de un determinado usuario.

**Function cambiar\_datos\_de\_usuario:** Recibe los datos del usuario que se desean cambiar. Verifica que se suministren todos y que sean válidos. Una vez que fueron validados se intenta actualizarlos en la base de datos.

**Function generar\_pdf:** Se ayudará de la librería fpdf creada especialmente para este fin. Su tarea será generar una lista en el archivo pdf con la lista de alumnos que recibe del curso determinado. Una vez que esto ocurra proporcionará un hipervínculo para la descarga del archivo

**Function validar\_fecha\_y\_hora:** Se necesita que todos los alumnos puedan comenzar a inscribirse a los cursos al mismo tiempo, para conseguir que esto funcione, se determinan la fecha y hora de inscripción en donde deberá iniciar este proceso. La tarea de esta function será verificar esta fecha y hora para que ningún alumno pueda inscribir algún curso antes del tiempo predefinido.

**Function obtener\_cursos:** Recibirá diversos parámetros, su tarea será verificar cuáles son los filtros proporcionados, y con ellos solicitar por medio de una consulta a la capa de acceso a datos los cursos que cumplan con las condiciones solicitadas.

Por ejemplo si no se recibe parámetro alguno, esta function solicita todos los cursos actuales, en cambio si recibe como parámetro un ponente, esta debe entender que se solicitan los cursos que actualmente está impartiendo el ponente específico.



Una vez que la function tiene el arreglo de datos, se encargará de enviarlos a la capa de presentación, para que con ellos se genere una lista.

**Function obtener\_curso:** Recibirá la clave del curso. Con esta clave solicitará a la base de datos que envíe todos los datos correspondientes a este curso.

**Function obtener\_lista\_de\_alumnos:** Se encarga de solicitar por medio de la capa de acceso a datos la lista de alumnos de un curso determinado. El curso será proporcionado como un parámetro.

**Function inscribir\_curso:** Recibirá un elemento „alumno’ y un elemento „curso’. Con estos datos se realizarán varias verificaciones antes de crear un elemento „inscripción’, que servirá para unir los dos elementos anteriores y con ello dar por hecho la inscripción del alumno en el curso.

Las verificaciones son las siguientes:

- La primera verificación será solicitar la fecha y hora de inscripción para saber si está activa.
- La segunda verificación será para asegurar que el alumno no se intenta inscribir en más cursos de los que están permitidos.
- La tercera verificación será para saber si el curso suministrado no está lleno. Esto lo realizará comparando el cupo máximo del curso contra el número de alumnos actualmente inscritos.
- La cuarta verificación será para saber si el alumno no está inscrito actualmente a este curso. Este es un error que puede ocurrir cuando se refresca la página.

Finalmente si la petición ha pasado todas las verificaciones, la function intentará crear un registro en la tabla de inscripción con la clave del alumno y del curso. Esta tarea se realizará por medio de la capa de acceso, que enviará un valor verdadero en caso de que la inscripción haya sido registrada, enviará un valor falso para comunicarle a esta function que ha ocurrido un error y que por tanto se solicitará al usuario que vuelva a intentar realizar la operación.

**Function obtener\_cursos\_inscritos:** Esta es una function parecida a la function obtener\_cursos, la diferencia radica, en que esta recibe un elemento „alumno’ y los cursos que se solicitarán son únicamente aquellos que cumplan con la condición de tener inscrito al alumno proporcionado.

**Function registrar\_curso:** Recibirá los datos del curso que se desea registrar, su tarea será verificar que se hayan ingresado todos los datos necesarios acerca del curso. En caso de que esta condición no se cumpla, la function enviará un mensaje de error al usuario, indicándole que faltó proporcionar todos los datos.

Una vez hecha la verificación, se intentará enviar los datos a la base de datos por medio de la capa de acceso y esperará una de las dos posibles respuestas:

- Verdadero: En caso de que se haya agregado correctamente el curso.



- Falso: En caso de que haya ocurrido un error.

**Function modificar\_curso:** Recibirá los datos que se desean modificar de un curso determinado. Su tarea es parecida a la de la función *registrar\_curso*, la diferencia es que en lugar de agregar un nuevo curso, en esta sólo se modifican los datos del curso determinado.

**Function evaluar\_alumnos:** Recibirá un arreglo de datos con las claves de los alumnos y su calificación correspondiente. Su tarea será generar una consulta de actualización para cada elemento, cada consulta será enviada a la capa de acceso a datos, para que esta actualice las calificaciones.

En caso de que ocurran errores en este proceso se le notificará al usuario para que vuelva a intentarlo.

**Function obtener\_cursos\_registrados:** Solicitará por medio de la capa de acceso a datos los cursos que tienen el estado de registrado. El arreglo obtenido será enviado a la capa de presentación para que el administrador tenga la posibilidad de activarlos.

**Function activar\_cursos:** Recibirá un elemento „curso”, con el cual debe crear una consulta para cambiar el estado actual de dicho curso. Esta consulta será enviada a la capa de acceso y esperará una respuesta de confirmación para saber que el cambio fue realizado.

**Function inscribir\_alumno:** Recibirá un elemento „alumno” y un elemento „curso”. Con estos datos deberá crear un nuevo elemento „inscripción”, al hacerlo el alumno quedará inscrito en el curso determinado.

Antes de realizar la creación del elemento „inscripción”, verificará que el elemento „alumno” sea válido. También verificará que actualmente no exista un elemento „inscripción” con los mismos datos, ya que esto implicaría que el alumno ya está inscrito al curso.

**Function borrar\_alumno:** Recibirá un elemento „inscripción”, con el que generará una consulta de eliminación. La consulta será enviada a la capa de acceso a datos, de la que esperará una respuesta para saber si se realizó correctamente la eliminación del elemento.

**Function obtener\_usuarios\_encontrados:** Recibirá clave y/o nombre y tipo de usuario, con los cuales deberá crear una consulta, que servirá para solicitar todos los elementos del tipo específico de usuario, que cumplan con los datos clave y/o nombre.

**Function restablecer\_contraseña:** Su única tarea será tomar el valor del número de usuario, para saber a qué elemento debe de modificar y asignará el mismo número como contraseña. Esta acción la realizará enviando una consulta a la capa de acceso, y esperará una respuesta para notificar al usuario de la acción realizada.



**Function obtener\_datos\_para\_reporte:** Su tarea será generar una consulta donde se soliciten todos los elementos que generarán el reporte del semestre.

**Function generar\_reporte:** Recibirá un arreglo de elementos, su tarea será almacenar estos datos en archivo con formato csv, que se generará para que el usuario lo pueda descargar.

**Function obtener\_semestres:** Su tarea será solicitar por medio de una consulta todos los elementos „semestre’. Esta consulta será enviada a la capa de acceso a datos, que enviará de regreso un arreglo con los semestres o bien un valor falso en caso de que ocurra un error.

**Function obtener\_semestre:** Recibirá un elemento „semestre’ con el cual generará una consulta para la base de datos, por medio de esta solicitará todos los datos del semestre determinado.

**Function agregar\_semestre:** Recibirá los datos de un elemento „semestre’, su tarea será verificar que los datos recibidos estén completos y que sean válidos. Una vez realizada esta verificación creará una consulta para con ella agregar el elemento a la base de datos. La consulta será enviada a la capa de acceso a datos, de la que se esperará un valor verdadero para confirmar que la acción sea exitosa.

**Function modificar\_semestre:** Recibirá los datos de un elemento „semestre’, con los que generará una consulta, misma que enviará a la capa de acceso a datos para solicitar la modificación de los datos del curso. Antes de realizar esta tarea, la function verificará que los datos estén completos y sean válidos.

### *3.3.3 Functions de la Capa de Acceso a Datos*

Aquí se describirán las functions que se encargan de realizar la comunicación entre el sistema y la base de datos, por medio de ellas se recibirán y enviarán los datos necesarios para efectuar las operaciones que se realizan en la capa de lógica de control.

**Function conexion\_a\_base\_de\_datos:** Almacena las configuraciones y los valores necesarios para realizar la conexión a la base de datos.

**Function obtener\_elementos:** Recibe una consulta para la base de datos y su tarea es conectarse con el manejador para poder realizar la consulta. Los datos obtenidos los regresa a la function que los solicitó en forma de arreglo. En caso de que haya ocurrido un error enviará un valor booleano falso a la function que lo llamó, con la finalidad de que esta sepa que no se pudieron obtener los datos.

**Function obtener\_elemento:** Tiene la misma tarea que la anterior su diferencia radica en que en vez de devolver un arreglo esta sólo devuelve un elemento. En caso de error realiza la misma acción.



**Function agregar\_elemento:** Recibe a un elemento en forma de consulta, su tarea será interactuar con la base de datos para que esta consulta se realice por medio del manejador y así el elemento quede agregado a la base de datos.

Esta function devuelve uno de los dos posibles valores:

- Verdadero: En caso de haber conseguido agregar el elemento.
- Falso: En caso de haber ocurrido un error al momento de intentar agregar al elemento.

**Function cambiar\_elemento:** Recibe un elemento en forma de consulta, su tarea será intentar actualizar los datos del elemento.

Esta function devuelve uno de los dos posibles valores:

- Verdadero: En caso de haber conseguido actualizar el elemento.
- Falso: En caso de haber ocurrido un error al momento de intentar actualizarlo.

**Function borrar\_elemento:** Recibirá un elemento en forma de consulta, su tarea será intentar eliminar un elemento de la base de datos, que se comunicará con el manejador.

Esta function devuelve uno de los dos posibles valores:

- Verdadero: En caso de haber conseguido borrar el elemento.
- Falso: En caso de haber ocurrido un error al momento de intentar hacerlo.





# *Capítulo 4*

# *IMPLANTACIÓN Y CONTROL*



*“Hombre recogerá quien siembre escuela”*

*José Martí*

La etapa de pruebas es la encargada de corroborar que el sistema sea funcional, en todos los sentidos para los que fue planeado y para los que fue creado. La implantación, es la etapa donde el sistema se entregó, se aceptó y se instaló en el servidor del CAE. Fue también aquí donde se instruyó y capacitó a los usuarios finales del sistema con el fin de que éstos pudiesen utilizarlo de forma correcta, algunas de las bondades de esto fue tener la opinión de los administradores sobre el sistema, y así poder efectuar las correcciones prudentes para hacer más amigable al sistema, y con ello facilitar algunos procesos.

Esta etapa se lleva en paralelo con la de pruebas sin embargo la etapa de pruebas fue más extensa.

Aun cuando la tarea principal del mantenimiento es la de corregir los errores visualizados en la etapa de pruebas la etapa de mantenimiento también contiene la posible ampliación del sistema.

## 4.1 IMPLANTACIÓN

Aquí se llevarán a cabo las tareas necesarias para poner en marcha al sistema. Se evaluarán aspectos como el lugar específico donde se instalará el sistema tanto en un entorno físico como en el lógico. Se verificarán las versiones de los programas que se encuentran instalados actualmente en el servidor, mismos que son necesarios para el funcionamiento del sistema.

### 4.1.1 Instalación del Sistema

Para poder instalar un sistema como este y por ende ponerlo en funcionamiento. Se necesitan tres cosas, un equipo de cómputo que albergue al sistema, una base de datos y una dirección electrónica para que se pueda acceder al sistema desde internet.

La computadora (servidor) que fue seleccionada para alojar al sistema, es la misma que actualmente se utiliza para alojar la página web de la licenciatura.

Para que el sistema pueda trabajar conjuntamente con la base de datos, por un lado se debe tener instalado un manejador que en este caso es mysql y también es necesario que el compilador PHP esté configurado correctamente para poder realizar la conexión. Una vez que se tiene instalado el manejador se procedió a instalar la base de datos como tal, esto es, creando el schema<sup>11</sup> correspondiente así como sus tablas y relaciones.

---

<sup>11</sup> Schema: El Esquema de una Base de datos (en Inglés Database Schema) describe la estructura de una Base de datos, en un lenguaje formal soportado por un Sistema administrador de Base de datos (DBMS). En una Base de datos Relacional, el Esquema define sus tablas, sus campos en cada tabla y las relaciones entre cada campo y cada tabla.



Por último se tomó la decisión de crear una carpeta dentro de la página de la licenciatura llamada cursos, por tanto la dirección electrónica para acceder al sistema desde internet es: <http://ingenieria.aragon.unam.mx/cursos/>.

### *4.1.2 Carga inicial de Datos*

Existen datos que son necesarios para que el sistema funcione desde un inicio, tales como los posibles valores de estado (Actual, Activo, Inactivo, Registrado etc.), datos para el usuario administrador (clave y contraseña), definición de los temas que se utilizarán para clasificar los cursos (software, hardware, redes, etc.), así como un primer elemento „semestre” en los cuales se puedan agregar los cursos actuales.

### *4.1.3 Pruebas de Implantación*

La primera tarea una vez instalado el sistema fue de hacer pruebas de cada módulo, esto con la finalidad en primera instancia, de que el sistema estuviese configurando correctamente y por otro lado, para mostrarlo a los administradores.

En esta fase se detectaron errores de configuración del servidor, que no fueron contemplados en un inicio. El error principal consistió en que el servidor no tenía instalado un módulo que permitiera comunicarse con el manejador de la base de datos desde el lenguaje PHP. La solución del problema fue sencilla ya que sólo era necesario realizar dicha instalación.

Una vez resueltos todos los errores, se determinó que el sistema estaba listo para pruebas más estrictas.

### *4.1.4 Preparación del Mantenimiento*

Con la finalidad de dar mantenimiento al sistema en el futuro, se instruyó a los administradores en el correcto uso del sistema y también se otorgó el manual de usuario, este último servirá para futuras referencias.

## **4.2 PRUEBAS**

La etapa de pruebas se desarrolló a la par de otras. Comenzando en la etapa de desarrollo, extendiéndose en la de implementación y culminó después de la etapa de mantenimiento.

Durante el desarrollo regularmente se utilizan datos de prueba. Poco antes de dar por terminada esta etapa se utilizaron datos reales, con los que se probaron los módulos por separado y finalmente se realizaron pruebas en conjunto para saber que efectivamente estos módulos se comunican correctamente.

En el momento mismo en que el sistema fue implantado se realizaron diversas pruebas, con la finalidad de asegurar que no existieran incompatibilidades entre el servidor de prueba y el que alojaría realmente al sistema. Este tipo de problemas



pueden ocurrir sobre todo a causa de las versiones de programas instalados en cada uno.

Una vez que el sistema fue implantado se encontraron errores que requirieron ajustes en la programación, una vez hechos los cambios se realizaron nuevamente pruebas para verificar que los errores efectivamente fueron solucionados.

#### *4.2.1 Pruebas Unitarias*

Se comprueba por separado el funcionamiento de cada una de las partes que conforman al sistema.

Primero se debe corroborar el funcionamiento de cada function, para después comprobar que el módulo completo funcione correctamente y que desarrolle su función establecida.

Si se encuentra algún mal funcionamiento en alguna function o bien en algún módulo, se debe proceder a la corrección de éste de forma inmediata, de lo contrario todo el sistema funcionará de forma incorrecta.

#### *4.2.2 Pruebas de Integración*

Debe comprobar la compatibilidad de todos los módulos y functions que componen al sistema. Durante las pruebas unitarias sólo se determinó si los módulos por separado funcionaban correctamente, ahora había que determinar cómo es que trabajaban en conjunto.

Se deben comparar los resultados obtenidos con los esperados, en caso de que éstos no sean los deseados se debe identificar el origen de cada error para poder llevar a cabo su corrección.

#### *4.2.3 Pruebas en Exploradores*

Sin lugar a dudas una parte importante en el desarrollo de sistemas web es la verificación de la compatibilidad del sistema con los diversos exploradores desde los cuales los usuarios pueden ingresar, para llevar esto a cabo se realizaron pruebas en los principales exploradores del mercado entre los cuales están: Internet Explorer (versiones: 5.0, 6.0, 7.0 y 8.0), Mozilla Firefox (versiones: 2.0, 3.0 y 3.5), Google Chrome (versiones: 4.1, 5.0 y 6.0), Opera (versiones: 10.0 y 10.51) y Safari (versión: 4.0). En todos ellos la respuesta fue favorable debido a la utilización del código XHTML para la creación de las páginas y de CSS para su modificación visual, con esto incluso se previó la implementación de nuevas tecnologías como lo es HTML5. Por último se evitó por completo la utilización de tecnologías como JavaScript y Flash, ya que estas suelen causar incompatibilidad y por ende infuncionalidad.



#### 4.2.4 Pruebas de Saturación

Una de las preocupaciones más grandes por parte de los administradores era la posible saturación de peticiones al servidor por parte de los usuarios, esto es debido a que en el momento de la inscripción (los primeros 3 minutos), se espera que todos los alumnos ingresen al mismo tiempo, y con ello comiencen a realizar una gran cantidad de acciones en el servidor y en la base de datos.

Para determinar si el sistema era capaz de soportar este tráfico, se solicitó el apoyo de casi cien alumnos para realizar una simulación de inscripción de cursos, dicha simulación arrojó resultados satisfactorios, con lo que se determinó que el sistema estaba apto para solventar la problemática.

### 4.3 PRESENTACIÓN Y APROBACIÓN DEL SISTEMA

El sistema fue presentado al responsable del CAE y a la Jefa de Carrera de la licenciatura en Ingeniería en Computación, quienes, verificaron que el sistema cumpliera con las funciones para las cuales fue elaborado, una vez que lo verificaron y dieron su aprobación, el sistema fue aceptado para ser empleado en el próximo periodo inter semestral (Agosto 2010).

### 4.4 FORMACIÓN DE USUARIOS

Durante esta tarea, se capacitó a los usuarios finales del sistema para que operaran el sistema de forma eficiente. La información que se le proporcionó, no fue sólo la de cómo opera el sistema, sino también las restricciones en el manejo del mismo y los errores que se pueden producir. Para esto fue necesario desarrollar lo siguiente:

#### 4.4.1 Elaboración del Manual de Usuario

Toda la información sobre el manejo del sistema debe estar contenida en los manuales de usuario, así como los errores que se puedan producir, la forma de tratarlos y de evitarlos.

Se debe especificar de forma detallada qué realiza cada botón, cada menú, cada caja de texto, etc., para que el usuario pueda explotar al máximo todas las funciones del sistema.

#### 4.4.2 Formación de Usuarios Finales

Junto con el manual de usuario, se realizaron reuniones para explicar el funcionamiento del sistema, todos los controles que hay y para qué sirven. La formación de usuarios finales fue rápida. Por un lado, debido a que están familiarizados con los sistemas, pero principalmente, al buen diseño del sistema en cuanto al entorno visual se refiere.



## 4.5 MANTENIMIENTO

El mantenimiento, es la última etapa del desarrollo de un sistema de información, éste puede ser:

- **Correctivo:** Son aquellos cambios realizados para corregir errores en un sistema de información.
- **Evolutivo:** Son las incorporaciones, modificaciones y eliminaciones necesarias en un sistema de información para cubrir las necesidades del usuario.
- **Adaptativo:** Son las modificaciones que afectan a los entornos en los que el sistema opera, por ejemplo, cambios de configuración del hardware, sistema operativo, gestores de base de datos, etc.
- **Perfectivo:** Son las acciones llevadas a cabo para mejorar la calidad interna de los sistemas en cualquiera de sus aspectos: reestructuración del código, definición más clara del sistema y optimización del rendimiento y eficiencia.

### 4.5.1 Registro de Peticiones

Para poder llevar a cabo el mantenimiento del sistema: primero, se necesita realizar una petición, debe llevarse a cabo un registro de peticiones, de modificaciones en los módulos del sistema, ya sea para agregar y quitar funciones o para reparar errores. Al mismo tiempo generará una evaluación sobre ellas.

### 4.5.2 Análisis de Peticiones

Se debe establecer la viabilidad de las peticiones de modificaciones al sistema que se reciban, para establecer si se puede llevar a cabo o no, si es necesaria a corto, mediano o largo plazo.

### 4.5.3 Identificación de los Elementos Afectados

Se identifican los elementos que resultarán afectados a causa de las nuevas funciones que serán incluidas en el sistema, así como las relaciones con los demás módulos y sus funciones.

De acuerdo al tipo de modificación, también deberán tomarse en cuenta los elementos físicos (hardware) debido a que puede requerirse que se adquiera algún elemento adicional.

### 4.5.4 Pruebas de Regresión

Una vez que se establezca qué elementos resultarán afectados y cómo lo harán, se debe corroborar que las modificaciones trabajen de forma correcta, y que éstas no provoquen errores o malas ejecuciones en los módulos y funciones que no fueron modificados, en caso de que se detecte algún problema, deben hacerse las modificaciones necesarias para solucionar dicho problema.



### 4.5.5 Implementación de las Modificaciones

Una vez analizadas las peticiones y realizadas las pruebas de regresión se debe comenzar a implantar el sistema con las nuevas modificaciones por lo que será necesario generar un nuevo manual de usuario e instruir a los usuarios del sistema sobre cómo emplear las modificaciones.

### 4.5.6 Seguimiento y Evaluación de Cambios

Se verifica que sólo los elementos afectados por el cambio hayan sido modificados, los cambios que sean efectuados en el sistema deberán ser revisados para asegurar el correcto funcionamiento, que cumplan con sus especificaciones y que no causen conflicto alguno con el resto de los módulos del sistema.

## 4.6 CRECIMIENTO DEL SISTEMA

En cada realización de cursos se solicita a los alumnos colaboren con la donación de material (teclados, ratones, pendrives, etc.) que será útil para mantener el buen funcionamiento del CAE.

Durante la realización de la primera inscripción en la que se utilizó el sistema la petición de este material se realizó de forma manual, sin embargo se esperaba que pudiese ser agregada.

Apenas antes de terminar el presente trabajo se llevaron a cabo algunas mejoras entre las cuales se agregó esta funcionalidad.

Otra de las cosas que se necesitaba y no había sido contemplada, era la posibilidad de mandar mensajes (avisos) a los usuarios por medio del sistema. Con la finalidad de mantener una comunicación con los mismos. Esta mejora también fue implementada.

Para poder realizar este crecimiento se debió pasar nuevamente por todas las etapas que se han descrito (Análisis, Diseño, Desarrollo, Pruebas e Implantación). Mismas que fueron realizadas en el periodo de Diciembre del 2010 y Enero del 2011.

### 4.6.1 Modificaciones a el modelo de la Base de Datos

Para agregar estas mejoras se hicieron modificaciones en la base de datos. Donde se agregaron tablas para el manejo de los mensajes y para poder asignar la donación que cada alumno debiera realizar.

Las tablas agregadas fueron: **Donacion**, **Articulo** y **Mensaje**.



El modelo E-R final fue el siguiente.

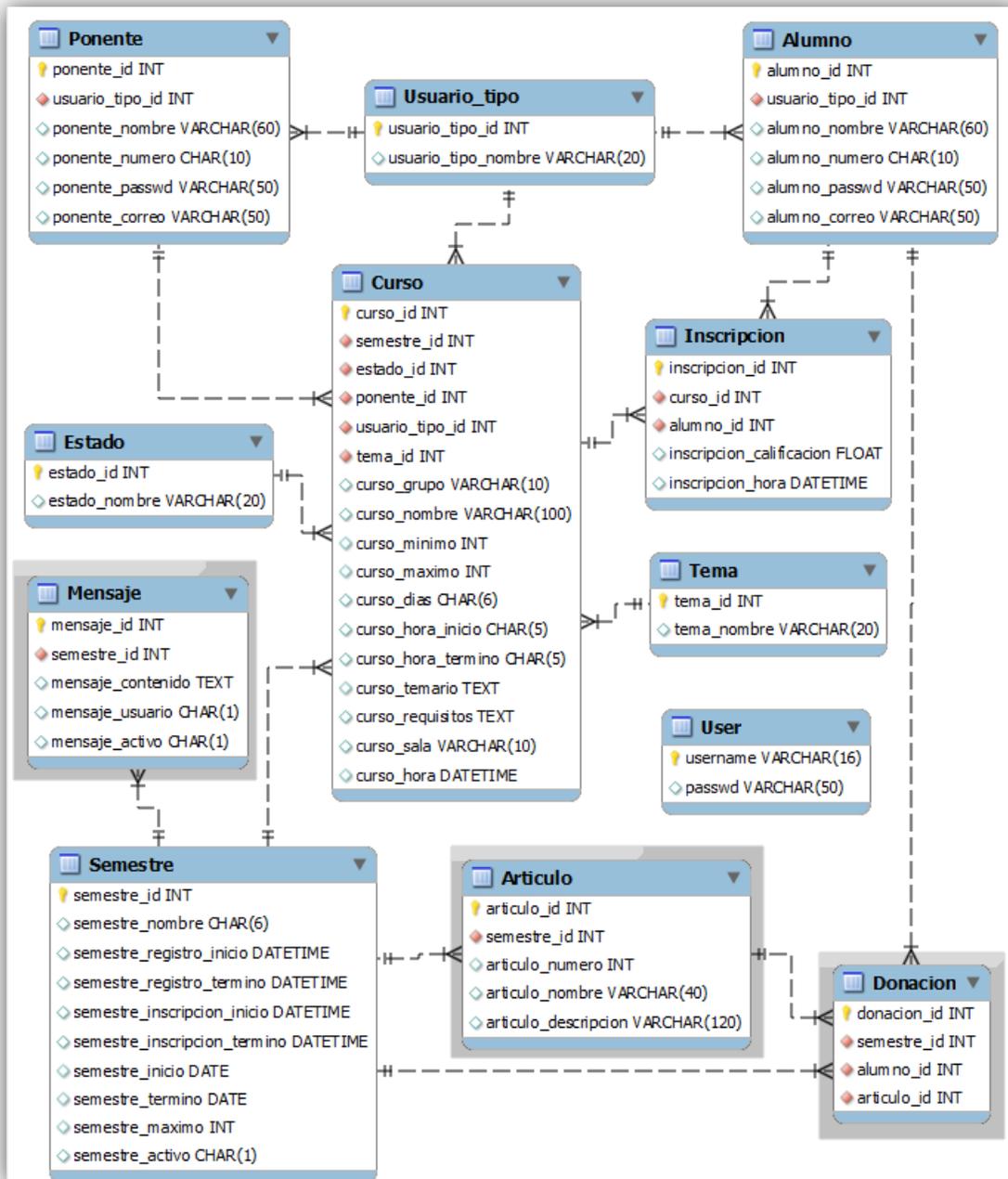


Figura 4-1 Modelo de la Base de Datos (Final)



Con la modificación de la base está implícito el cambio del diccionario de datos.

Entidad	Atributo	Llave	Tipo	Tamaño
<b>Articulo</b>	articulo_id	PK	int	11
	semestre_id	FK	int	11
	articulo_numero		int	11
	articulo_nombre		varchar	40
	articulo_descripcion		varchar	120
<b>Donacion</b>	donacion_id	PK	int	11
	semestre_id	FK	int	11
	alumno_id	FK	int	11
	articulo	FK	int	11
<b>Mensaje</b>	mensaje_id	PK	int	11
	semestre_id	FK	int	11
	mensaje_contenido		text	0
	mensaje_usuario		char	1
	mensaje_activo		char	1

Tabla 4-1 Elementos Agregados al Diccionario de Datos

#### 4.6.2 Módulos Agregados

Para realizar la manipulación de estos nuevos datos se crearon los módulos **Donación** y **Editar Mensajes** para el usuario administrador y uno llamado **Mensajes** el cual es común para los alumnos y ponentes.

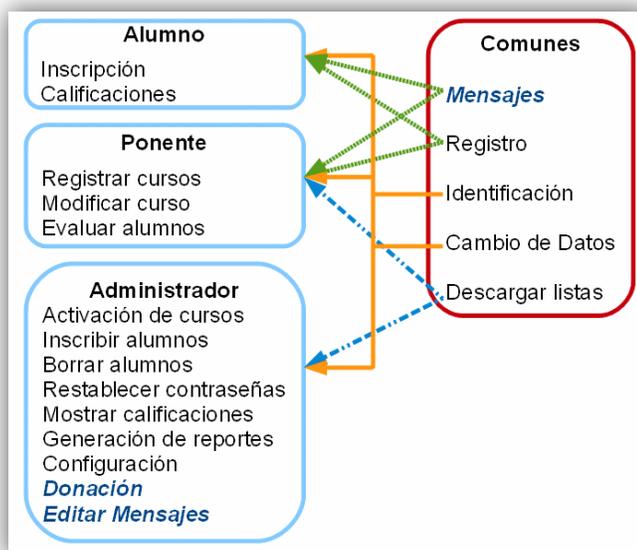


Figura 4-2 Diseño Funcional del Sistema (Final)

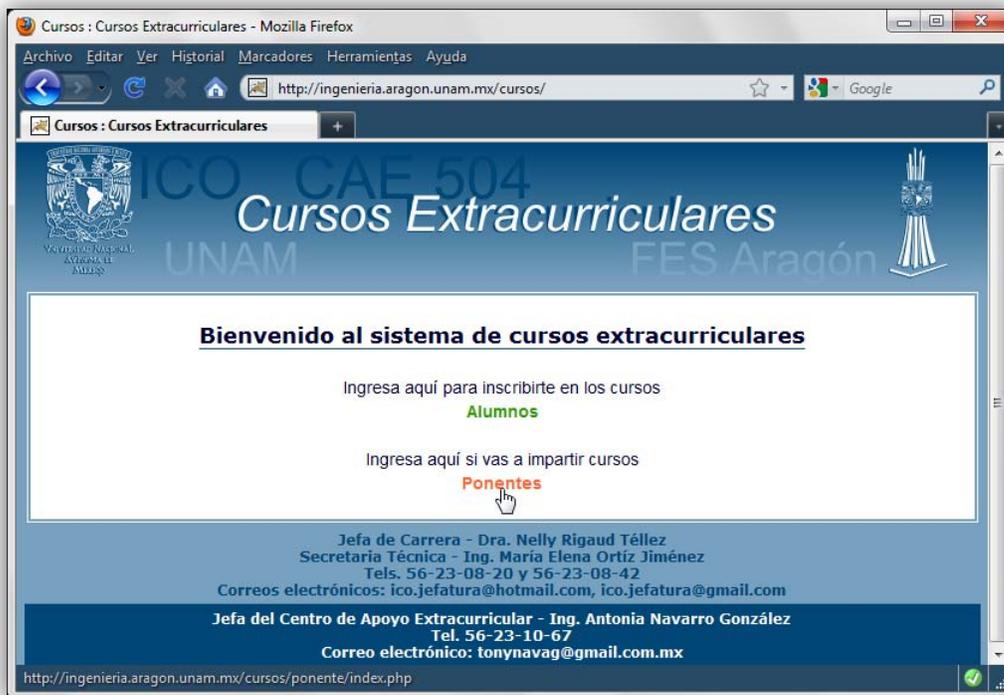


## 4.7 MANUAL DE USUARIO

### 4.7.1 Opciones del Ponente

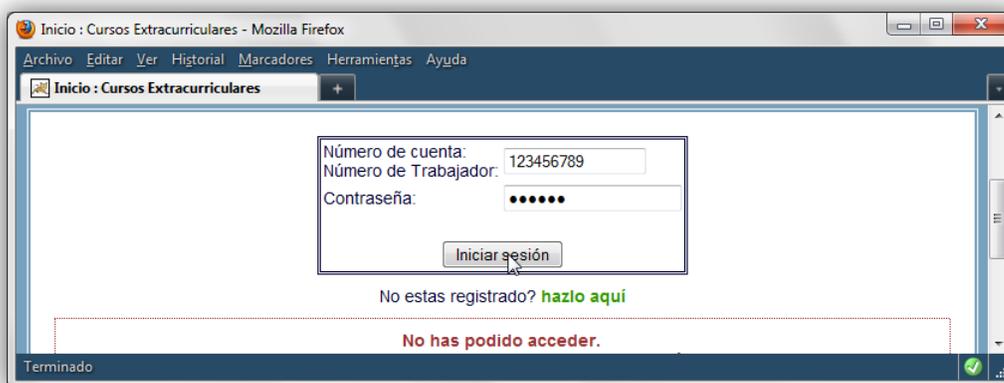
#### INGRESO AL SISTEMA

El sistema para la inscripción de cursos es accesible desde la dirección electrónica <http://ingenieria.aragon.unam.mx/cursos/>



**Figura 4-3 Ingreso al Sistema**

El ingreso al sistema se realizará por medio de la previa autenticación.



**Figura 4-4 Formulario de Ingreso.**



## REGISTRO EN EL SISTEMA

En caso de que no se haya registrado tendrá que hacerlo.

**Figura 4-5 Formulario de Registro.**

El registro se puede hacer únicamente dentro de las fechas determinadas de lo contrario se mostrará el siguiente mensaje.

El periodo para realizar el registro es del 8 de Noviembre de 2010 al 10 de Diciembre de 2010  
Contacte con el CAE para poder realizar su registro despues de estas fechas.

**Figura 4-6 Error en Registro. Fuera del Periodo de Registro.**

Lo primero que se ejecuta es la revisión de los datos, en caso de no cubrir correctamente con alguno de ellos se mostrará alguno de los siguientes mensajes.

No es una dirección de correo válida.

**Figura 4-7 Error en Registro. Dirección de Correo.**

La contraseña que has ingresado no concuerda.

**Figura 4-8 Error en Registro. Contraseñas Diferentes.**

Tu contrasea debe tener entre 6 y 16 caracteres.

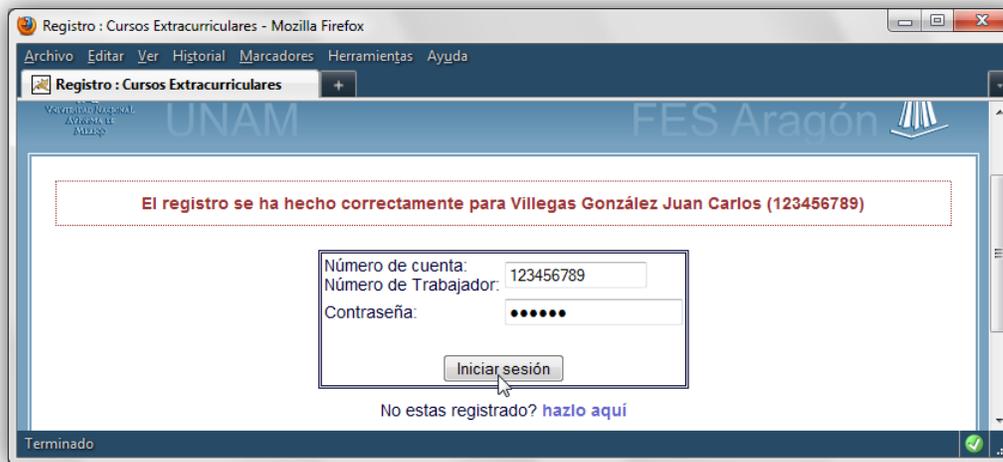
**Figura 4-9 Error en Registro. Longitud de Contraseña.**





**Figura 4-10 Error en Registro. Faltan Datos.**

Si el registro es exitoso, se mostrará el siguiente mensaje y se proporciona el formulario de ingreso.



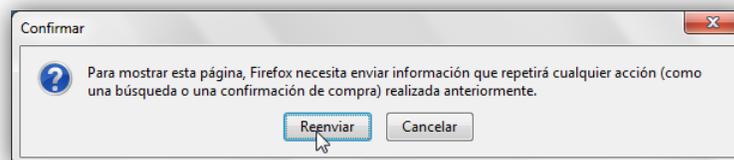
**Figura 4-11 Mensaje de Registro Exitoso.**

De lo contrario enviará un mensaje de error. El cual se puede generar debido a una falla en la base de datos.



**Figura 4-12 Error en Registro. Registro Fallido.**

Y en caso de que el registro ya se haya realizado anteriormente, se mostrará otro error. Este se puede generar si se refresca la página, con lo cual los datos serían enviados nuevamente al servidor.



**Figura 4-13 Solicitud de Reenvío de Datos.**



**Figura 4-14 Error en Registro. Registro Existente.**



## PÁGINA DE INICIO



**Figura 4-15** Página de Inicio para el Ponente.

NOTA: La opción de “Registrar curso” solo aparecerá durante el periodo de registro. Una vez pasado este periodo cambiara el menú.



**Figura 4-16** Menú del Ponente despues de la Fecha de Registro.

## CERRAR SESIÓN

Para cerrar la sesión bastará con presionar “Cerrar sesión” en el Menú de Usuario.



**Figura 4-17** Mensaje de Sesión Cerrada Exitosamente.

En caso de ocurrir algún error se mostrará el siguiente mensaje.



**Figura 4-18** Error en Cierre de Sesión.

Un último error puede ocurrir en caso de que no haya iniciado una sesión o bien que esta ya se haya cerrado.



**Figura 4-19** Error en Cierre de Sesión. No Existe una Sesión.



## REGISTRO DE CURSOS

Para poder registrar un curso se deberá llenar un formulario con los datos correspondientes del curso.

The screenshot shows a web browser window with the title 'Registro de Curso : Cursos Extracurriculares - Mozilla Firefox'. The browser's address bar shows 'Registro de Curso : Cursos Extracurric...'. The main content area displays a form titled 'Registro de Cursos' with the following fields:

- Nombre:** Java Avanzado
- Tema:** Programación
- Dirigido a:** Alumnos
- Operación:** Lun  Mar  Mie  Jue  Vie  Sab
- Horario:** 10:00 a 11:30
- Temario:**
  1. Interfaces Gráficas de Usuario (GUI)
  2. Excepciones
  3. Flujos de Entrada / Salida
  4. Programación Multihilo (Threads)
  5. Eventos
  6. Comunicaciones con Sockets TCP/IP
  7. Paquetes y Distribución
- Requisitos Previos para el Alumno:** Conocimientos básicos de Java (preferentemente haber tomado el curso "Java Básico").

At the bottom of the form is a 'Registrar' button. The browser's status bar at the bottom left shows 'Terminado'.

**Figura 4-20 Formulario para Registro de Cursos.**

En caso de que se haya terminado el periodo de registro, y se logró ingresar en esta página de alguna forma el sistema enviará el siguiente mensaje.

**Fuera de fechas**

**Figura 4-21 Error en Registro de Curso. Registro Fuera de Fechas.**

Para que el registro sea exitoso se deberán proporcionar todos los datos solicitados.

**Faltan datos, intentalo de nuevo**

**Figura 4-22 Error en Registro de Curso. Falta de Datos.**



Si al momento de intentar registrar el curso ocurriera un error, se mostraría un mensaje.

Ocurrió un error al intentar registrar el curso, intentalo de nuevo

**Figura 4-23 Error en Registro de Curso.**

En caso de que se realice un registro exitoso aparecerán los datos del curso.



**Figura 4-24 Registro Exitoso de un Curso**

NOTA: Los datos faltantes del curso (Grupo, Sala, Cupo Mínimo y Máximo) serán agregados por los administradores en el momento Activar el Curso.

La última posibilidad es que se refresque la página y se envíen los datos nuevamente al servidor. En este caso se informará al usuario con el siguiente mensaje.

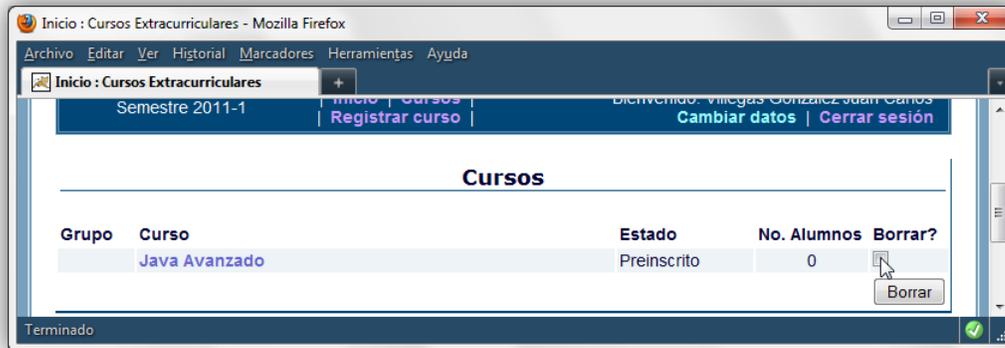
El curso ya está registrado

**Figura 4-25 Error en Registro de Curso. Registrado Anteriormente.**



## MODIFICACIÓN DE UN CURSO

Una vez registrado el curso, se mostrará en la página de los cursos. Donde se pueden realizar acciones como borrar los cursos que no se hayan activado.



**Figura 4-26** Página de Inicio con Lista de Cursos

Desde el mismo se puede ingresar a la página del curso, donde los datos se podrán modificar, siempre y cuando el curso no haya sido activado.

**Modificación de Cursos Registrados**

Nombre:

Tema:

Dirigido a:

Operación:  Lun  Mar  Mie  Jue  Vie  Sab

Horario:  a

Temario:

1. Interfaces Gráficas de Usuario (GUI)
2. Excepciones
3. Flujos de Entrada / Salida
4. Programación Multihilo (Threads)
5. Eventos
6. Comunicaciones con Sockets TCP/IP
7. Paquetes y Distribución

Requisitos Previos para el Alumno:

**Figura 4-27** Formulario para la Modificación de Cursos



Cuando los cursos han sido activados ya no es posible modificarlos, por lo cual en vez de aparecer un formulario solo aparecerá la información correspondiente al curso.

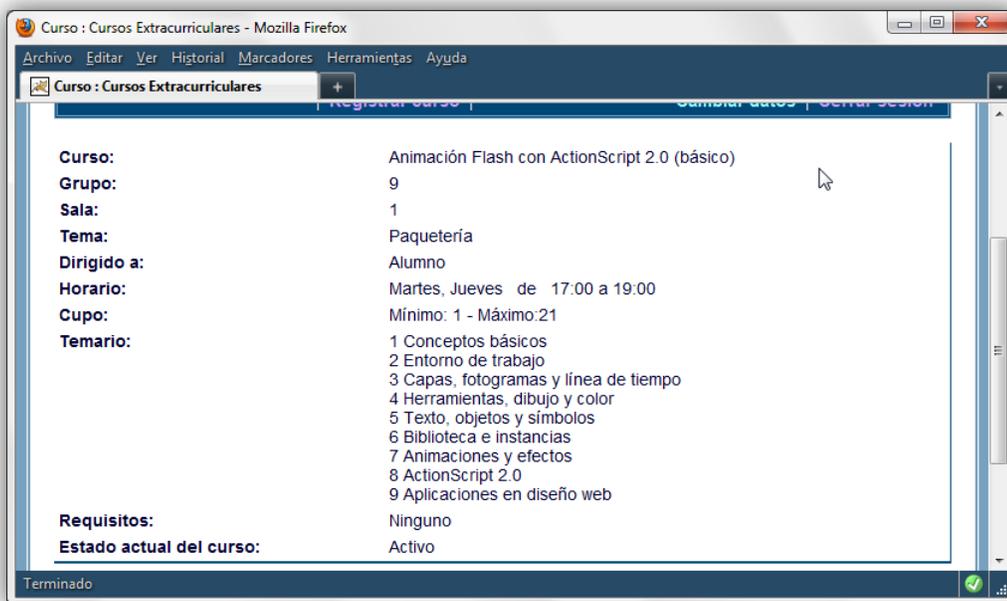


Figura 4-28 Curso Activo

## DESCARGAR LISTAS

Una vez terminado el periodo de inscripción el ponente podrá descargar la lista de los alumnos inscritos.

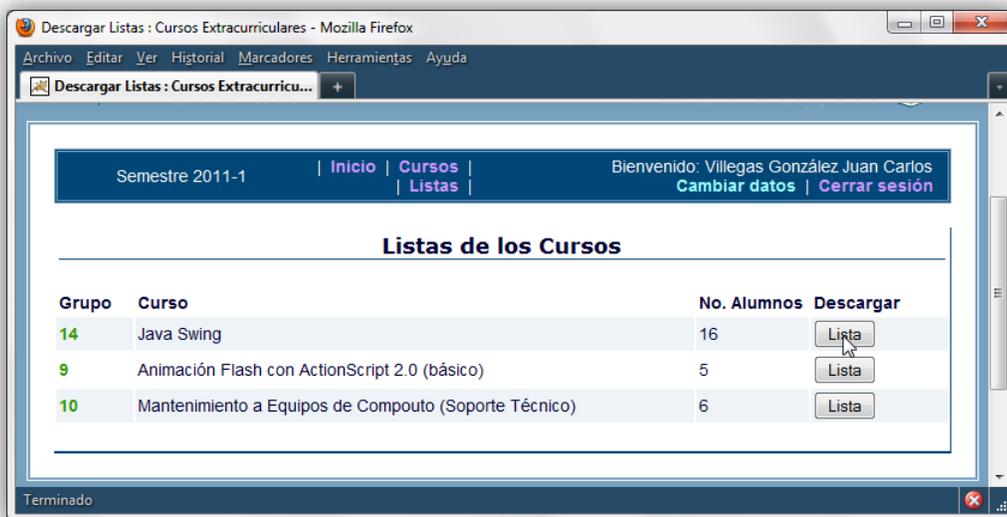
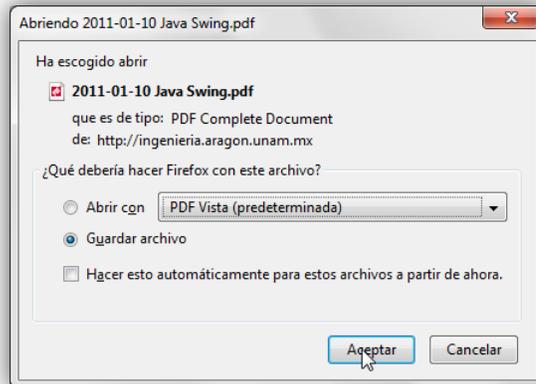


Figura 4-29 Página para Descargar las Listas de los Cursos

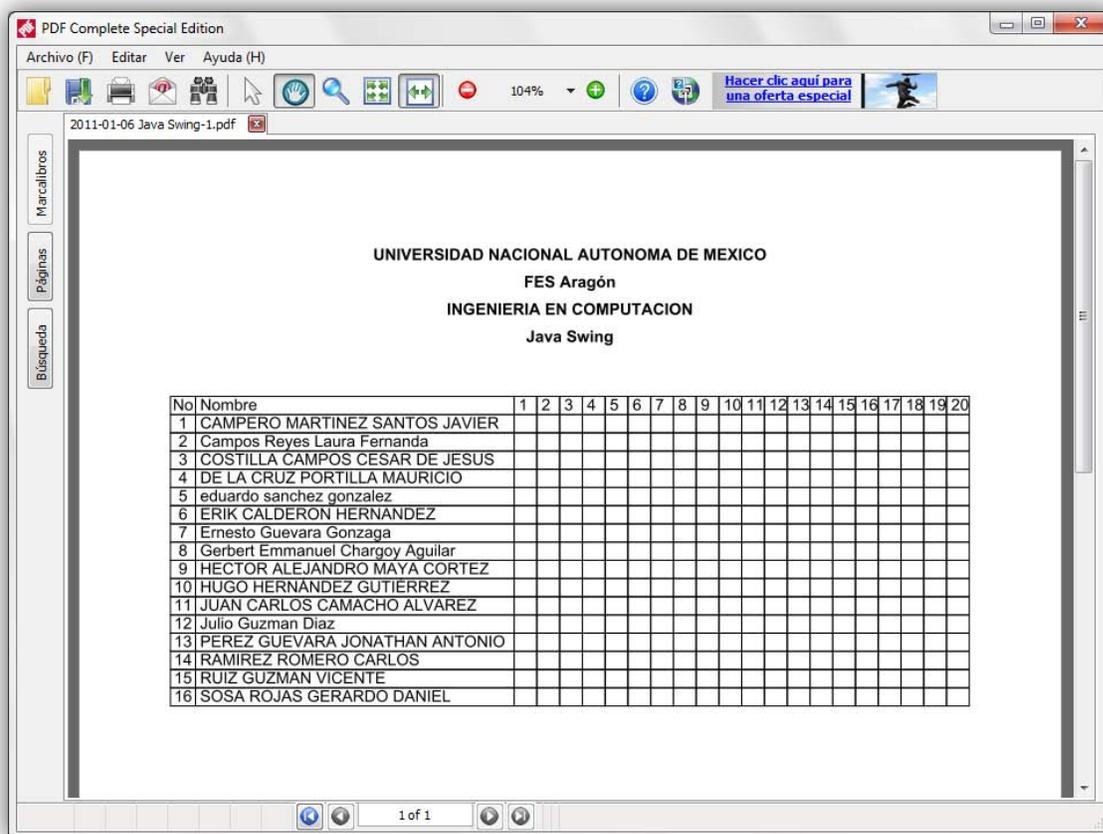


La lista se generará en un archivo pdf.



**Figura 4-30 Descarga de la Lista de Alumnos**

La lista será visible, imprimible y portable una vez descargada.

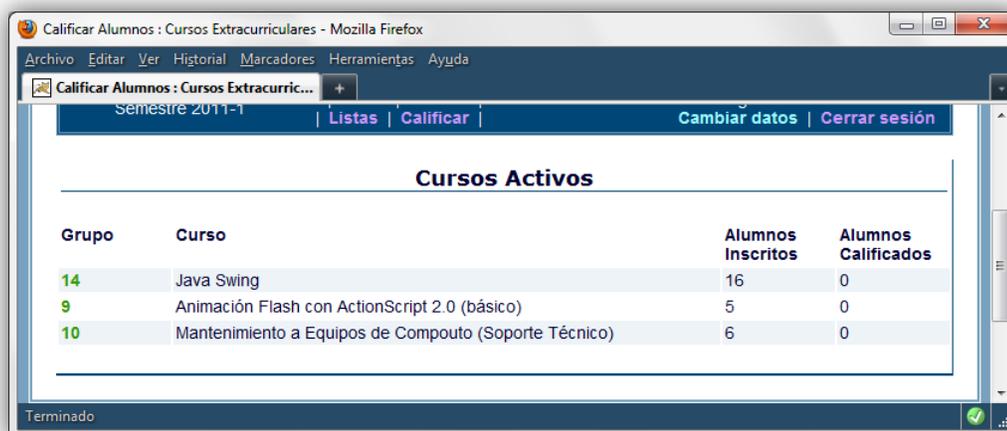


**Figura 4-31 Lista en Formato PDF**



## CALIFICAR ALUMNOS

Al final del periodo de cursos los ponentes deberán ingresar en el sistema las calificaciones obtenidas por los alumnos durante los cursos, para ello el sistema proporcionará un formulario en donde se podrá realizar esta tarea.



Calificar Alumnos : Cursos Extracurriculares - Mozilla Firefox

Semestre 2011-1 | Listas | **Calificar** | Cambiar datos | Cerrar sesión

### Cursos Activos

Grupo	Curso	Alumnos Inscritos	Alumnos Calificados
14	Java Swing	16	0
9	Animación Flash con ActionScript 2.0 (básico)	5	0
10	Mantenimiento a Equipos de Compouto (Soporte Técnico)	6	0

Terminado

**Figura 4-32 Lista de Cursos por Calificar**

Al ingresar en un grupo se muestra la siguiente pantalla, donde se deberán ingresar las calificaciones de cada alumno.



Calificaciones : Cursos Extracurriculares - Mozilla Firefox

Semestre 2011-1 | Inicio | Cursos | Bienvenido: Villegas González Juan Carlos  
| Listas | **Calificar** | Cambiar datos | Cerrar sesión

### Curso: 9 - Animación Flash con ActionScript 2.0 (básico)

#### Lista de Alumnos Inscritos

Alumnos	Calificación
BARRERA PLATA MARCELO DANIEL	10
Ernesto Guevara Gonzaga	<input type="text"/>
MARGARITA ARELLANO JUAREZ	<input type="text"/>
MARIN LOPEZ FELIPE DE JESUS	<input type="text"/>
RUIZ GUZMAN VICENTE	<input type="text"/>

Calificar

Terminado

**Figura 4-33 Formulario para Calificar**



Al enviar los datos se intenta actualizar los datos. En caso de que haya ocurrido un problema se mostrara un mensaje de error.

Datos actualizados correctamente

**Figura 4-34 Mensaje de Actualización de Calificaciones Exitosa**

Ocurrió un error al intentar cambiar los datos de los alumnos, intentalo de nuevo

**Figura 4-35 Error en Actualización de Calificaciones**

## ACTUALIZACIÓN DE DATOS

En este espacio se le proporciona al ponente un sencillo formulario para realizar el cambio de su contraseña y otro para realizar la actualización de sus datos.

**Figura 4-36 Formularios para Actualización de Datos**

Tal como ocurre en el registro no puede haber campos vacíos por tanto si esto ocurre se enviará algún mensaje de error, también se verifican otras situaciones como la no repetibilidad de números (claves) con otros ya registrados.

El número ya está inscrito. Verifica tus datos.

**Figura 4-37 Error en la Actualización. Registro Existente.**



Lo primero que se ejecuta es la revisión de los datos, en caso de no cubrir correctamente con alguno de ellos se mostrará alguno de los siguientes mensajes.

Faltan datos.

**Figura 4-38 Error en la Actualización. Faltan Datos.**

No es una dirección de correo válida.

**Figura 4-39 Error en la Actualización. Dirección de Correo No Válida.**

Contraseña escrita no era la misma.

**Figura 4-40 Error en el Cambio de Contraseña. Contraseñas Diferentes.**

Tu contrasea debe tener entre 6 y 16 caracteres.

**Figura 4-41 Error en el Cambio de Contraseña. Longitud de Contraseña.**

En el caso específico de la contraseña se verificará además que la vieja contraseña sea correcta. Cuando esta etapa es superada se intenta realizar la actualización, al final enviará mensajes de acuerdo al resultado obtenido.

Los Datos han sido cambiados.

**Figura 4-42 Actualización de Datos Exitosa**

Contraseña cambiada.

**Figura 4-43 Cambio de Contraseña Exitoso**

Los datos no han podido cambiarse.

**Figura 4-44 Error en la Actualización de Datos**

La contraseña no ha podido cambiarse.

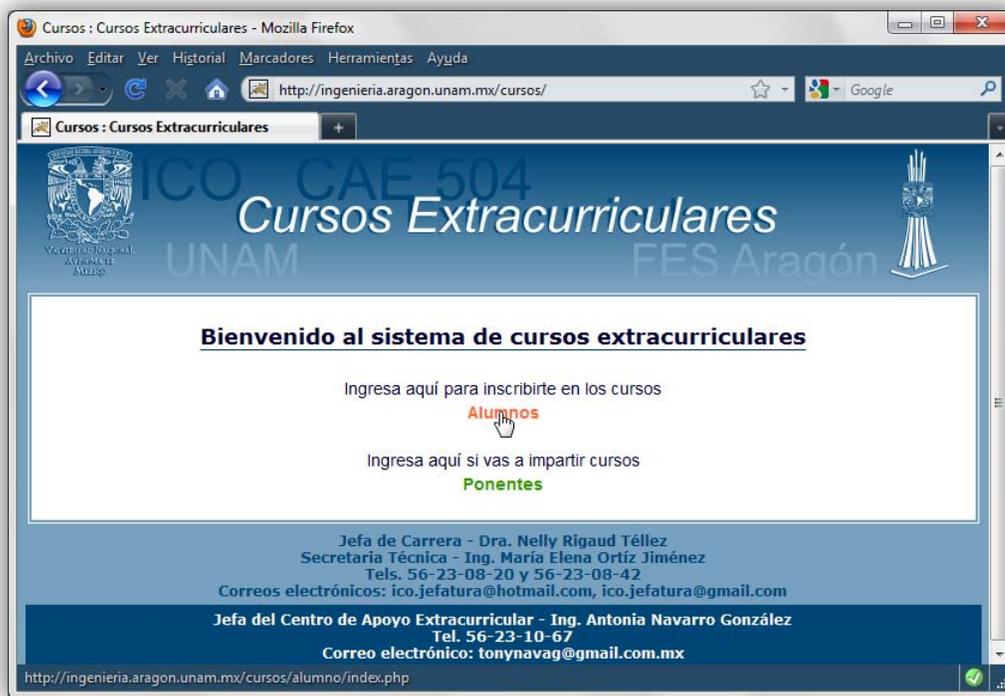
**Figura 4-45 Error en Cambio de Contraseña**



## 4.7.2 Opciones del Alumno

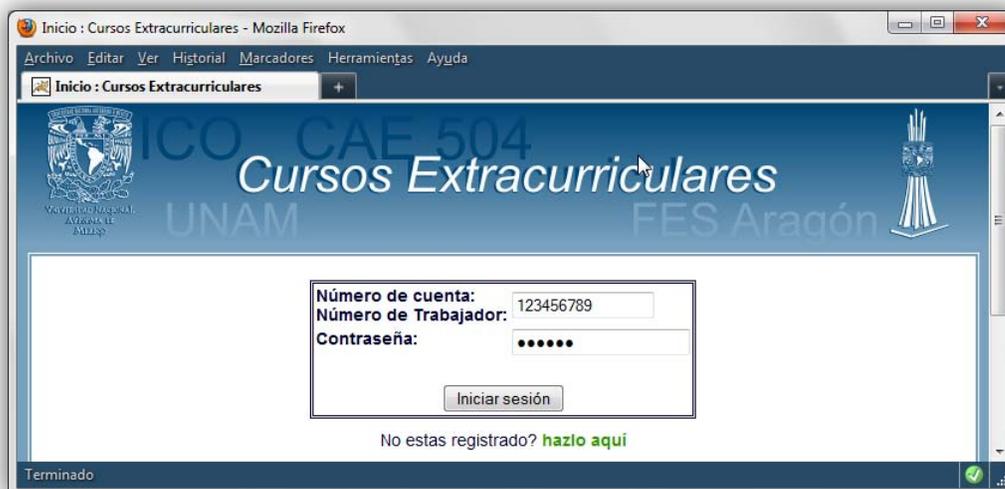
### INGRESO AL SISTEMA

El sistema para la inscripción de cursos es accesible desde la dirección electrónica <http://ingenieria.aragon.unam.mx/cursos/>



**Figura 4-46 Ingreso al Sistema**

El ingreso al sistema se realizará por medio de la previa autenticación.



**Figura 4-47 Formulario de Ingreso.**



## REGISTRO EN EL SISTEMA

En caso de que no se haya registrado tendrá que hacerlo.

**Figura 4-48 Formulario de Registro.**

El registro se puede hacer únicamente dentro de las fechas determinadas de lo contrario se mostrará el siguiente mensaje.

El periodo para realizar el registro es del 8 de Noviembre de 2010 al 10 de Diciembre de 2010  
Contacte con el CAE para poder realizar su registro despues de estas fechas.

**Figura 4-49 Error en Registro. Fuera del Periodo de Registro.**

Lo primero que se ejecuta es la revisión de los datos, en caso de no cubrir correctamente con alguno de ellos se mostrará alguno de los siguientes mensajes.

No es una dirección de correo válida.

**Figura 4-50 Error en Registro. Dirección de Correo.**

La contraseña que has ingresado no concuerda.

**Figura 4-51 Error en Registro. Contraseñas Diferentes.**

Tu contrasea debe tener entre 6 y 16 caracteres.

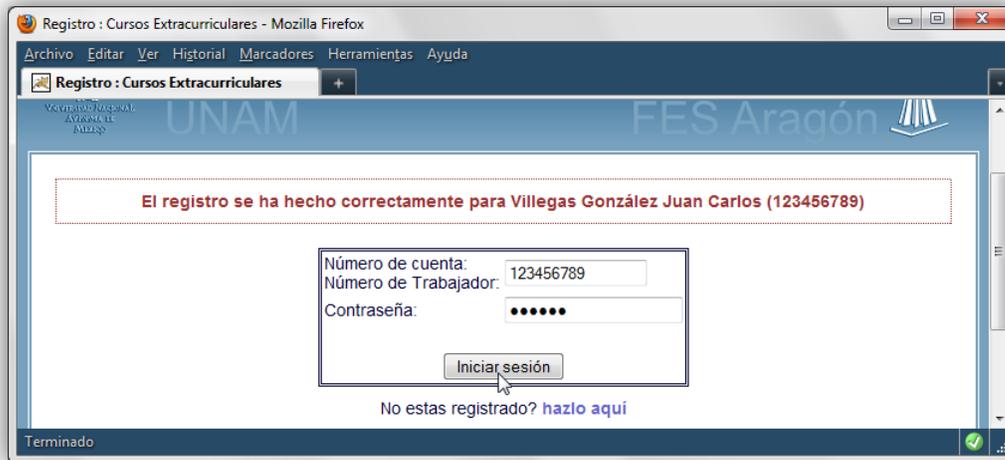
**Figura 4-52 Error en Registro. Longitud de Contraseña.**





**Figura 4-53 Error en Registro. Faltan Datos.**

Si el registro es exitoso, se mostrará el siguiente mensaje y se proporciona el formulario de ingreso.



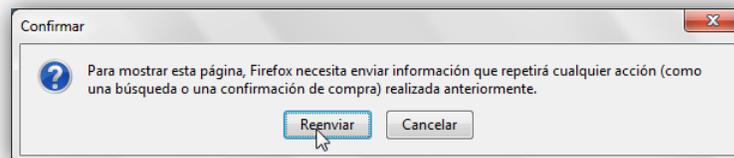
**Figura 4-54 Mensaje de Registro Exitoso.**

De lo contrario enviará un mensaje de error. El cual se puede generar debido a la base de datos.



**Figura 4-55 Error en Registro. Registro Fallido.**

Y en caso de que el registro ya se haya realizado anteriormente, se mostrará otro error. Este se puede generar si se refresca la página, con lo cual los datos serían enviados nuevamente al servidor.



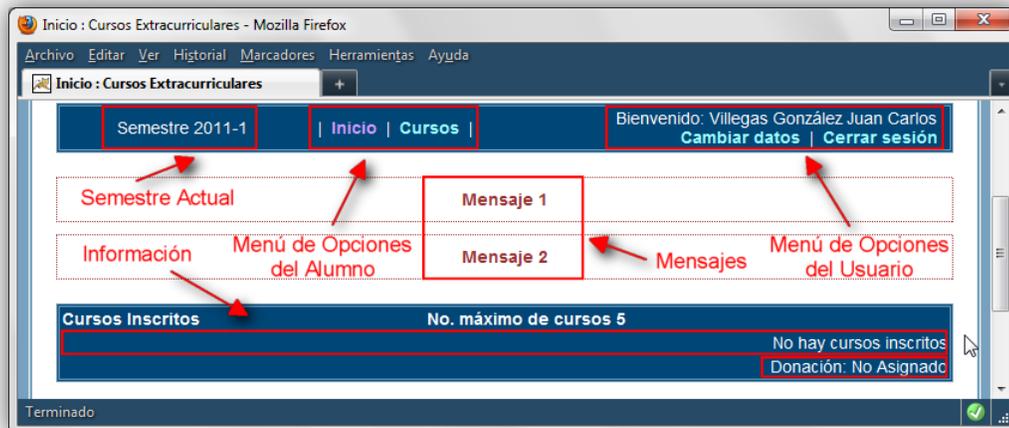
**Figura 4-56 Solicitud de Reenvío de Datos.**



**Figura 4-57 Error en Registro. Registro Existente.**



## PÁGINA DE INICIO

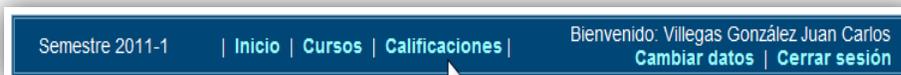


**Figura 4-58** Página de Inicio para el Alumno.

NOTA: La opción de “Inscripción” solo aparecerá durante el proceso de inscripción. Una vez pasado este periodo desaparecerá del menú. Al terminar el periodo de los cursos, los ponentes ingresarán las calificaciones obtenidas por el alumno, se podrán ver en la opción “Calificaciones”.



**Figura 4-59** Menú del Alumno Durante el Proceso de Inscripción.



**Figura 4-60** Menú del Alumno Terminado el Periodo de Cursos

## CERRAR SESIÓN

Para cerrar la sesión bastará con presionar “Cerrar sesión” en el Menú de Usuario. En caso de ocurrir algún error se mostrará un mensaje.



**Figura 4-61** Mensaje de Sesión Cerrada Exitosamente.



**Figura 4-62** Error en Cierre de Sesión.



Un último error puede ocurrir en caso de que no haya iniciado una sesión o bien que esta ya se haya cerrado.

No tienes una sesión abierta, así que no hemos podido cerrarla.

**Figura 4-63 Error en Cierre de Sesión. No Existe una Sesión.**

## INSCRIPCIÓN

Durante el proceso de inscripción el alumno ingresará al apartado “Inscripción” desde donde podrá ver los cursos que actualmente se están impartiendo y a sí mismo podrá ver su disponibilidad.

Semestre 2011-1 | Inicio | Inscripción | Bienvenido: Villegas González Juan Carlos  
Cambiar datos | Cerrar sesión

**Cursos Inscritos** No. máximo de cursos 5  
No hay cursos inscritos  
Donación: No Asignado

### Programación

Grupo	Curso	Cupo Mínimo	Cupo Máximo	No. Alumnos Inscritos
15	Unreal Engine 3 con UDK (Cupo Lleno)	1	7	7
22	Java Básico	1	22	20
16	CREACIÓN DE SITIOS WEB CON PHP(BÁSICO) (Cupo Lleno)	1	20	20
23	Java Avanzado	1	22	9
13	DISEÑO DE SOLUCIONES CON MICROCONTROLADORES MODULO 3	1	8	0
2	Programación en C# para puertos COM y USB	1	7	4
11	Programación Orientada a Objetos con JAVA.	1	22	5
20	Programación Orientada a Objetos con Java	1	18	17
1	Programación Orientada a Objetos con C++ (Cupo Lleno)	1	18	18
29	PROGRAMACIÓN EN C (Cupo Lleno)	1	22	22
14	Java Swing	1	21	16

### Electrónica

Grupo	Curso	Cupo Mínimo	Cupo Máximo	No. Alumnos Inscritos
1	Robots manipuladores	1	9	5
3	Consola de videojuegos con microcontroladores	1	7	3
4	Comunicaciones inalámbricas con microcontroladores (Cupo Lleno)	1	6	6
5	Desarrollo de Soluciones Tecnológicas con microcontroladores PIC. Módulo 1	1	7	4
6	Diseño de un robot móvil autónomo	1	8	6

### Hardware

Grupo	Curso	Cupo Mínimo	Cupo Máximo	No. Alumnos Inscritos
8	Mantenimiento y Reparación en Equipos de Cómputo Básico	1	10	8
21	Mantenimiento Preventivo y Correctivo a Equipo de Cómputo	1	10	8

Terminado

**Figura 4-64 Cursos Inscribibles**



Al seguir el hipervínculo del Curso se desplegarán las características del mismo. Además proporciona el botón para solicitar la acción de inscribir el curso.



**Figura 4-65 Datos del Curso**

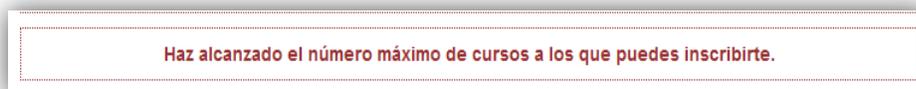
Para efectuar la inscripción del alumno en el curso, se realizan diversas verificaciones, con la finalidad de evitar errores.

Se verifica que el curso no se haya inscrito anteriormente, esto puede ocurrir al refrescar una página, con lo que la petición sería enviada nuevamente al sistema.



**Figura 4-66 Mensaje de Error en Inscripción. El Curso ya está Inscrito**

El alumno solo tendrá la posibilidad de inscribirse a un número máximo de cursos, este será determinado por los Administradores del sistema. En caso de que el alumno intente inscribirse a más cursos de los permitidos el sistema no lo permitirá.



**Figura 4-67 Mensaje de Error en Inscripción. Número Máximo de Cursos Alcanzado**



Es posible que durante el proceso se intente inscribir a un curso cuyo cupo se ha llenado, esto lo verifica el sistema y suprime la posibilidad de que el alumno lo haga.

No te pudiste inscribir, El curso que solicitaste ya está lleno.

**Figura 4-68 Mensaje de Error en Inscripción. Curso LLeno**

Una última posibilidad es que el sistema tenga un problema y por tanto no pueda llevar a cabo la operación.

Ocurrió un error al intentar registrar el curso, intentalo de nuevo

**Figura 4-69 Mensaje de Error en Inscripción. Falla en el Sistema**

En caso de que estas validaciones sean pasadas eficientemente, se mostrará un mensaje de inscripción exitosa y se agregará el curso inscrito al menú de cursos inscritos.

The screenshot shows a web browser window titled 'Cursos : Cursos Extracurriculares - Mozilla Firefox'. The page displays a success message: 'Tu registro se realizo exitosamente en el curso: Programación Orientada a Objetos con Java'. Below this, a section titled 'Cursos Inscritos' shows a maximum of 5 courses. The current course is '20 - Programación Orientada a Objetos con Java' with a schedule of 'Lunes, Miércoles, Viernes de 11:00 a 13:00' and 'Donación: No Asignado'. A table titled 'Programación' lists various courses with their group numbers, names, and enrollment status.

Grupo	Curso	Cupo Mínimo	Cupo Máximo	No. Alumnos Inscritos
15	Unreal Engine 3 con UDK (Cupo Lleno)	1	7	7
22	Java Básico	1	22	20
16	CREACIÓN DE SITIOS WEB CON PHP(BÁSICO) (Cupo Lleno)	1	20	20
23	Java Avanzado	1	22	9
13	DISEÑO DE SOLUCIONES CON MICROCONTROLADORES MODULO 3	1	8	0
2	Programación en C# para puertos COM y USB	1	7	4
11	Programación Orientada a Objetos con JAVA.	1	22	5
20	Programación Orientada a Objetos con Java (Cupo Lleno)	1	18	18
19	Programación Orientada a Objetos con C++ (Cupo Lleno)	1	18	18

**Figura 4-70 Curso Inscrito**

Una vez terminada la inscripción de todos los cursos, bastará con cerrar la sesión para salir.



## ASIGNACIÓN DE MATERIAL

De acuerdo a la cantidad de cursos a los que se inscribió el alumno, le será asignada una donación que deberá presentar en el CAE para poder acceder a los cursos. La fecha y hora se informarán en el área de mensajes en la página principal.

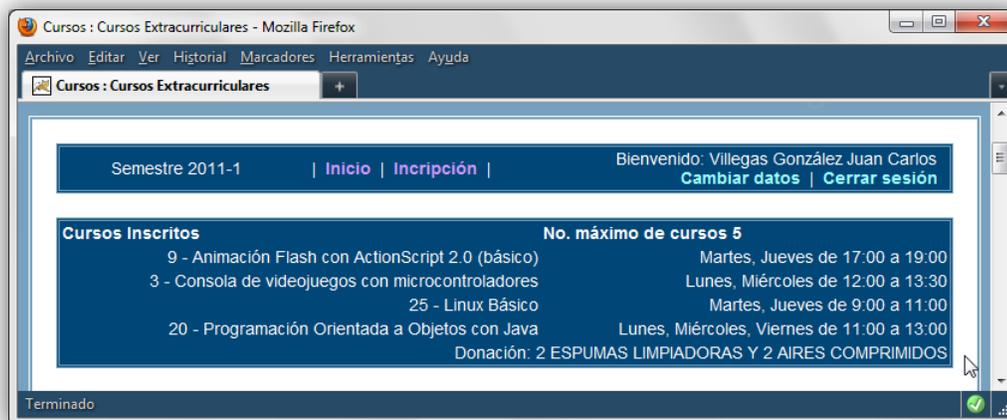


Figura 4-71 Lista de Cursos Inscritos con Material Asignado

### 4.7.3 Opciones del Administrador

## INGRESO AL SISTEMA

El sistema para la inscripción de cursos es accesible desde la dirección electrónica <http://ingenieria.aragon.unam.mx/cursos/>

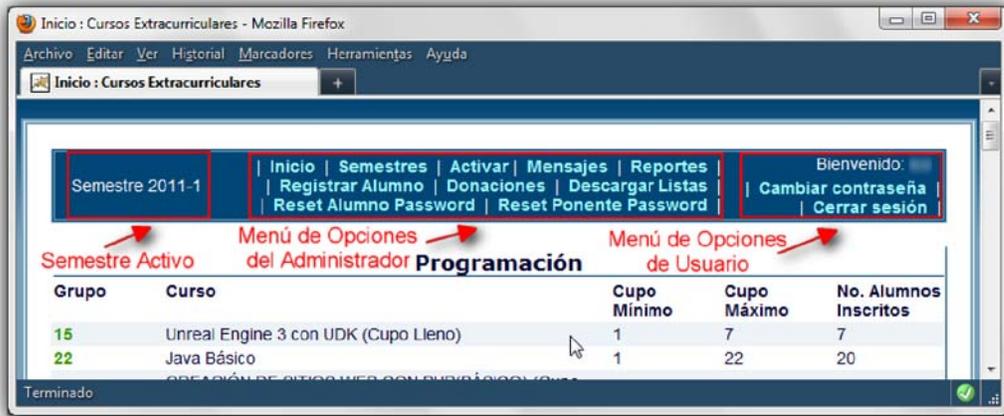
El ingreso al sistema se realizará por medio de la previa autenticación.



Figura 4-72 Formulario de Ingreso.



## PÁGINA DE INICIO



**Figura 4-73** Página de Inicio para el Administrador

## CERRAR SESIÓN

Para cerrar la sesión bastará con presionar “Cerrar sesión” en el Menú de Usuario.



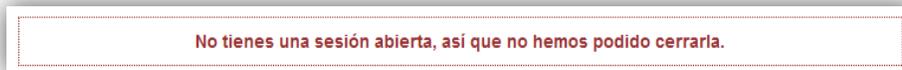
**Figura 4-74** Mensaje de Sesión Cerrada Exitosamente.

En caso de ocurrir algún error se mostrará el siguiente mensaje.



**Figura 4-75** Error en Cierre de Sesión.

Un último error puede ocurrir en caso de que no haya iniciado una sesión o bien que esta ya se haya cerrado.



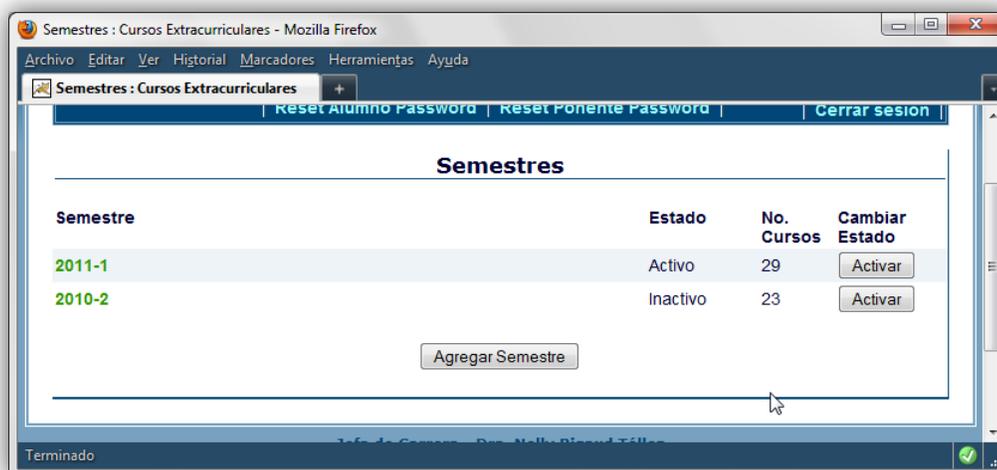
**Figura 4-76** Error en Cierre de Sesión. No Existe una Sesión.



## SEMESTRES

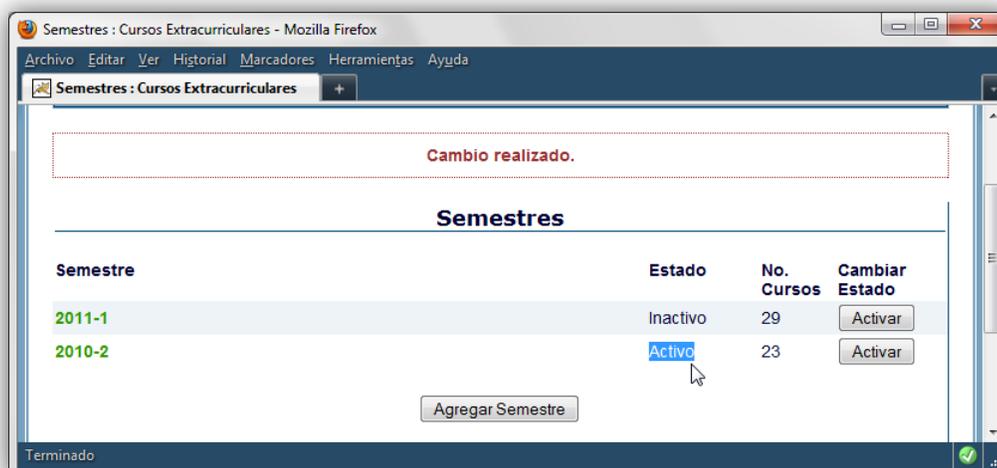
Desde el apartado de “Semestres” se tendrá acceso a las configuraciones propias del sistema, desde este se podrán configurar las fechas y horarios de registro, inscripción y periodo de impartición de los cursos.

En él se muestran los periodos, estos tienen un estado actual (Activo, Inactivo). Este estado servirá para determinar el semestre sobre el cual se desea trabajar.



**Figura 4-77 Lista de Semestres**

Si cambiamos un semestre al estado “Activo” serán los cursos de este semestre los que aparecerán en la página inicial, y por tanto será sobre ellos que se realizarán todas las modificaciones que se realicen.



**Figura 4-78 Cambio de Estado de un Semestre**





**Figura 4-79** Página Principal con Semestre Activo Cambiado

NOTA: El estado del curso solo modifica las acciones del administrador, para los alumnos y ponentes el único semestre visible es el último.

## AGREGAR SEMESTRES

Desde el mismo apartado de “Semestres” se podrá generar uno nuevo por medio del siguiente formulario.

Nombre:

Fecha de Registro Inicio:

Fecha de Registro Termino:

Fecha de Inscripción Inicio:

Fecha de Inscripción Termino:

Fecha de Inicio:

Fecha de Termino:

Máximo No. de cursos:

**Figura 4-80** Formulario para Agregar un Semestre

Para que esta acción se pueda realizar se deberán proporcionar todos los datos requeridos de lo contrario se mostrará el siguiente mensaje.

Faltan datos.

**Figura 4-81** Error al Agregar Semestre. Faltan Datos.



## MODIFICACIÓN DE UN SEMESTRE

Se proporciona la posibilidad de modificar los datos del semestre.

Semestre : Cursos Extracurriculares - Mozilla Firefox

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

Semestre : Cursos Extracurriculares +

### Semestre

Nombre: 2011-1

Fecha de Registro Inicio: 2010-11-22 00:00:00

Fecha de Registro Termino: 2010-12-01 09:00:00

Fecha de Inscripción Inicio: 2011-01-10 09:00:00

Fecha de Inscripción Termino: 2011-01-10 18:00:00

Fecha de Inicio: 2010-12-02

Fecha de Termino: 2011-01-28

Máximo No. de cursos: 5

Cambiar

Terminado

**Figura 4-82** Formulario para la Edición de los Datos de un Semestre

Para que esta modificación se lleve a cabo deberán proporcionarse todos los datos, si esto no se cumple el sistema enviará el siguiente mensaje.



**Figura 4-83** Error al Modificar Semestre. Faltan Datos.

## ACTIVACIÓN DE CURSOS

Una vez que los ponentes han registrado los cursos que desean impartir durante el próximo periodo estos se muestran en el apartado de “Activar Cursos” que como su nombre lo indica servirá para activar los cursos y por tanto hacer visibles para los alumnos toda la información referente a los mismos.

Activar Cursos : Cursos Extracurriculares - Mozilla Firefox

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

Activar Cursos : Cursos Extracurricula... +

### Cursos Preinscritos

Curso	Mínimo	Máximo	Grupo	Sala	Activar?
Introducción a la Programación con Lenguaje C	5	15	23		Activar
Linux Básico					Activar
Programación Orientada a Objetos					Activar

Terminado

**Figura 4-84** Formulario para Activación de Cursos



Para llevar a cabo esta tarea se deberá suministrar todos los datos necesarios, de lo contrario se mostrará el siguiente mensaje y no se activará el curso.

Faltan datos.

**Figura 4-85 Error en Activación. Faltan Datos.**

## REGISTRO DE ALUMNOS

Si el periodo de registro ha terminado y se desea registrar a un alumno, el sistema proporciona esta posibilidad por medio del siguiente formulario.

**Figura 4-86 Formulario de Registro.**

Lo primero que se ejecuta es la revisión de los datos, en caso de no cubrir correctamente con alguno de ellos se mostrará alguno de los siguientes mensajes.

No es una dirección de correo válida.

**Figura 4-87 Error en Registro. Dirección de Correo.**

La contraseña que has ingresado no concuerda.

**Figura 4-88 Error en Registro. Contraseñas Diferentes.**

Tu contrasea debe tener entre 6 y 16 caracteres.

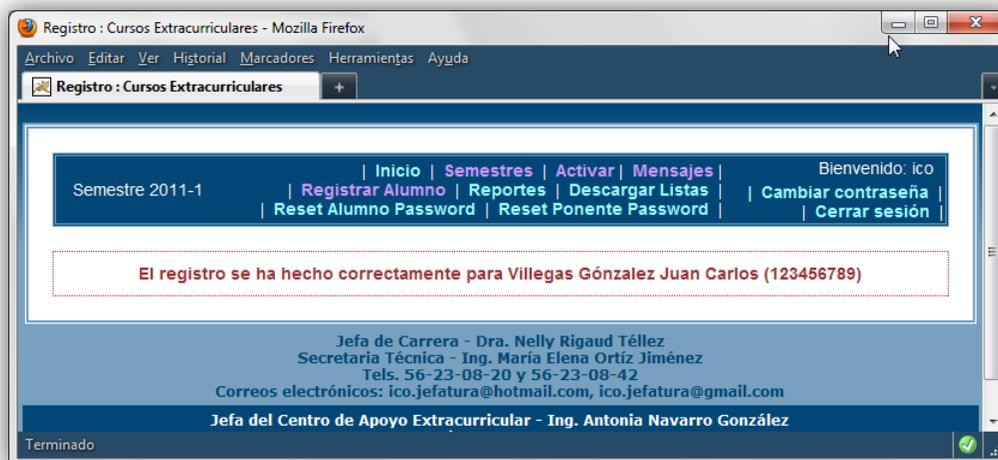
**Figura 4-89 Error en Registro. Longitud de Contraseña.**





**Figura 4-90 Error en Registro. Faltan Datos.**

Si el registro es exitoso, se mostrará el siguiente mensaje y se proporciona el formulario de ingreso.



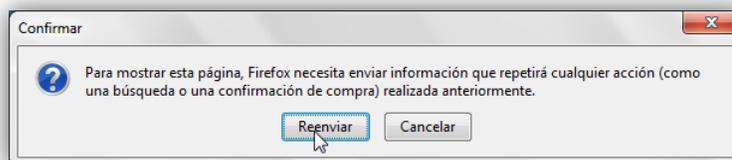
**Figura 4-91 Mensaje de Registro Exitoso.**

De lo contrario enviará un mensaje de error. El cual se puede generar debido a la base de datos.



**Figura 4-92 Error en Registro. Registro Fallido.**

Y en caso de que el registro ya se haya realizado anteriormente, se mostrará otro error. Este se puede generar si se refresca la página, con lo cual los datos serían enviados nuevamente al servidor.



**Figura 4-93 Solicitud de Reenvío de Datos.**



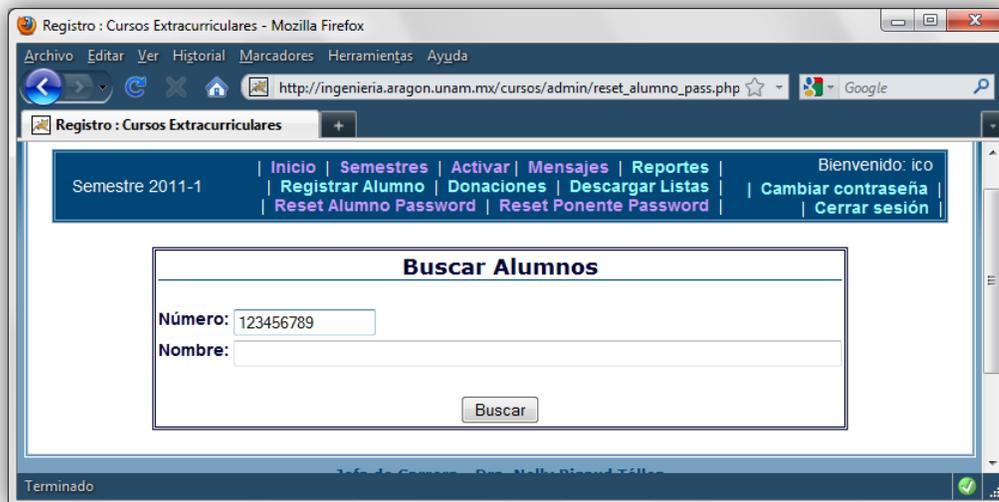
**Figura 4-94 Error en Registro. Registro Existente.**



## RESTABLECER CONTRASEÑAS DE LOS USUARIOS

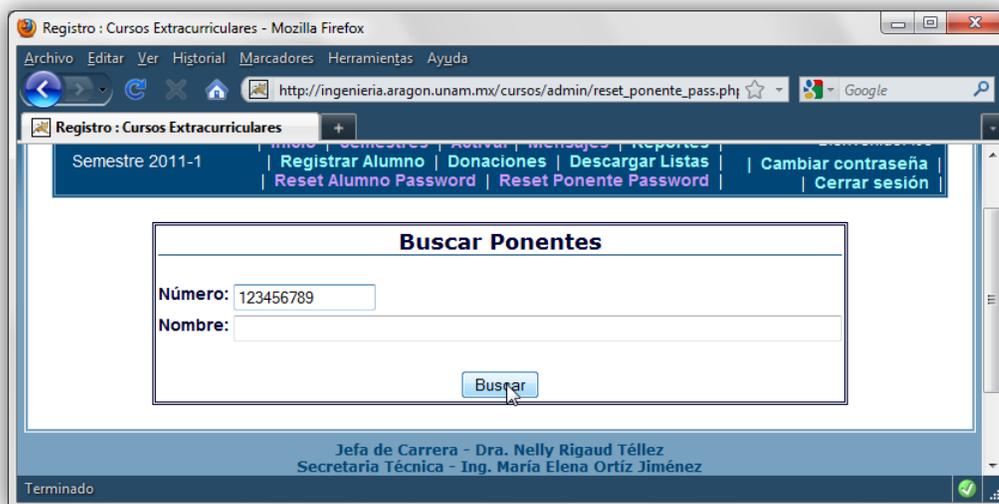
Es común que los usuarios tanto alumnos como ponentes olviden sus contraseñas, para permitir nuevamente que estos usuarios tengan acceso al sistema se deberá cambiar la contraseña del usuario en cuestión.

Para realizar esta tarea se requiere realizar una búsqueda del usuario ya que puede ser que el problema sea también la clave.



The screenshot shows a web browser window titled 'Registro : Cursos Extracurriculares - Mozilla Firefox'. The address bar shows the URL 'http://ingenieria.aragon.unam.mx/cursos/admin/reset\_alumno\_pass.php'. The page header includes a navigation menu with links: Inicio, Semestres, Activar, Mensajes, Reportes, Bienvenido: ico, Registrar Alumno, Donaciones, Descargar Listas, Cambiar contraseña, and Cerrar sesión. Below the header is a sub-menu with links: Reset Alumno Password and Reset Ponente Password. The main content area is titled 'Buscar Alumnos' and contains a form with two input fields: 'Número:' with the value '123456789' and 'Nombre:'. A 'Buscar' button is located below the form. The footer of the page includes the text 'Terminado' and 'Jefa de Carrera - Dra. Nelly Rigaud Téllez'.

**Figura 4-95** Formulario para la Búsqueda de un Alumno

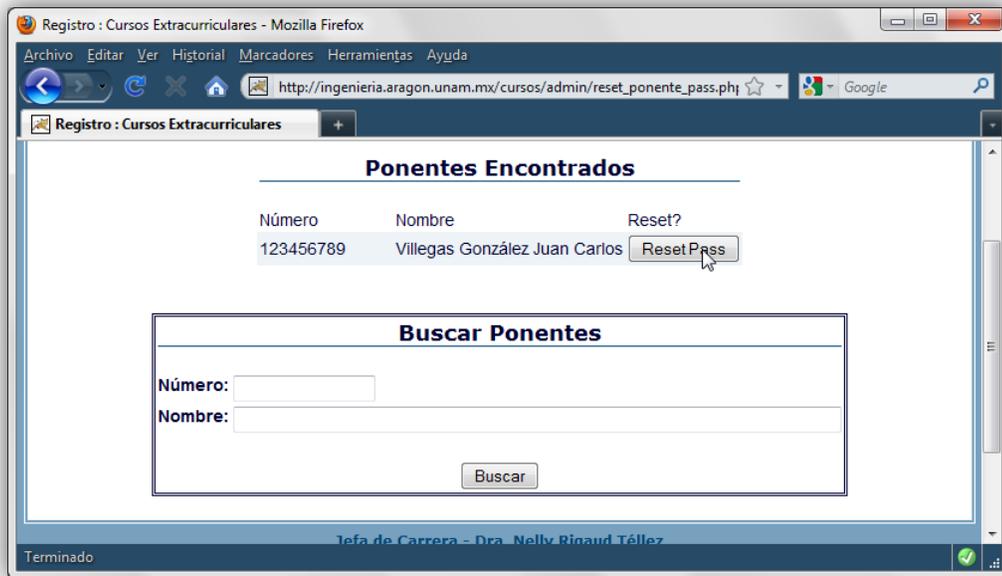


The screenshot shows a web browser window titled 'Registro : Cursos Extracurriculares - Mozilla Firefox'. The address bar shows the URL 'http://ingenieria.aragon.unam.mx/cursos/admin/reset\_ponente\_pass.php'. The page header includes a navigation menu with links: Inicio, Semestres, Activar, Mensajes, Reportes, Bienvenido: ico, Registrar Alumno, Donaciones, Descargar Listas, Cambiar contraseña, and Cerrar sesión. Below the header is a sub-menu with links: Reset Alumno Password and Reset Ponente Password. The main content area is titled 'Buscar Ponentes' and contains a form with two input fields: 'Número:' with the value '123456789' and 'Nombre:'. A 'Buscar' button is located below the form. The footer of the page includes the text 'Terminado' and 'Jefa de Carrera - Dra. Nelly Rigaud Téllez, Secretaria Técnica - Ing. María Elena Ortiz Jiménez'.

**Figura 4-96** Formulario para la Búsqueda de un Ponente



Al realizar la búsqueda se mostrarán aquellos elementos que coincidan con la misma. Y también se proporcionará la posibilidad de restablecer la contraseña.



**Figura 4-97 Lista de Ponentes Encontrados**

Una vez realizada esta operación la contraseña del usuario será igual a la Clave (Número de Cuenta – Número de Trabajador) con el que el usuario este registrado, por ejemplo si la clave es 123456789 deberá ingresar también 123456789 en el lugar de la contraseña.

## REPORTES

Para generar un reporte del estado actual del semestre activo bastará ingresar al apartado “Reportes”, donde se podrá descargar un archivo con formato .csv mismo que será visible y editable desde cualquier aplicación de Hoja de Cálculo.

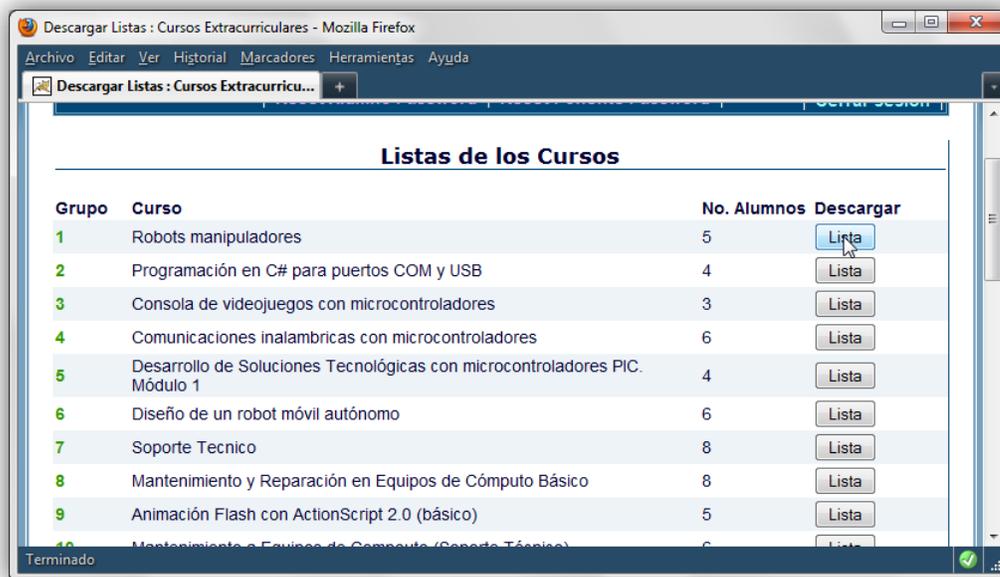


**Figura 4-98 Descarga de Reporte**



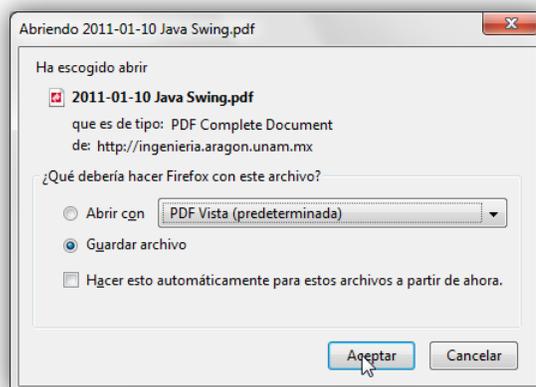
## LISTA DE ALUMNOS

Una vez terminado el proceso de Inscripción de los alumnos a los cursos se hace necesario tener en formato .pdf las listas completas de cada curso.



**Figura 4-99** Formulario para Descargar Listas de Alumnos

La lista se generará en un archivo pdf.



**Figura 4-100** Descarga de Lista de Alumnos



La lista será visible, imprimible y portable una vez descargada.

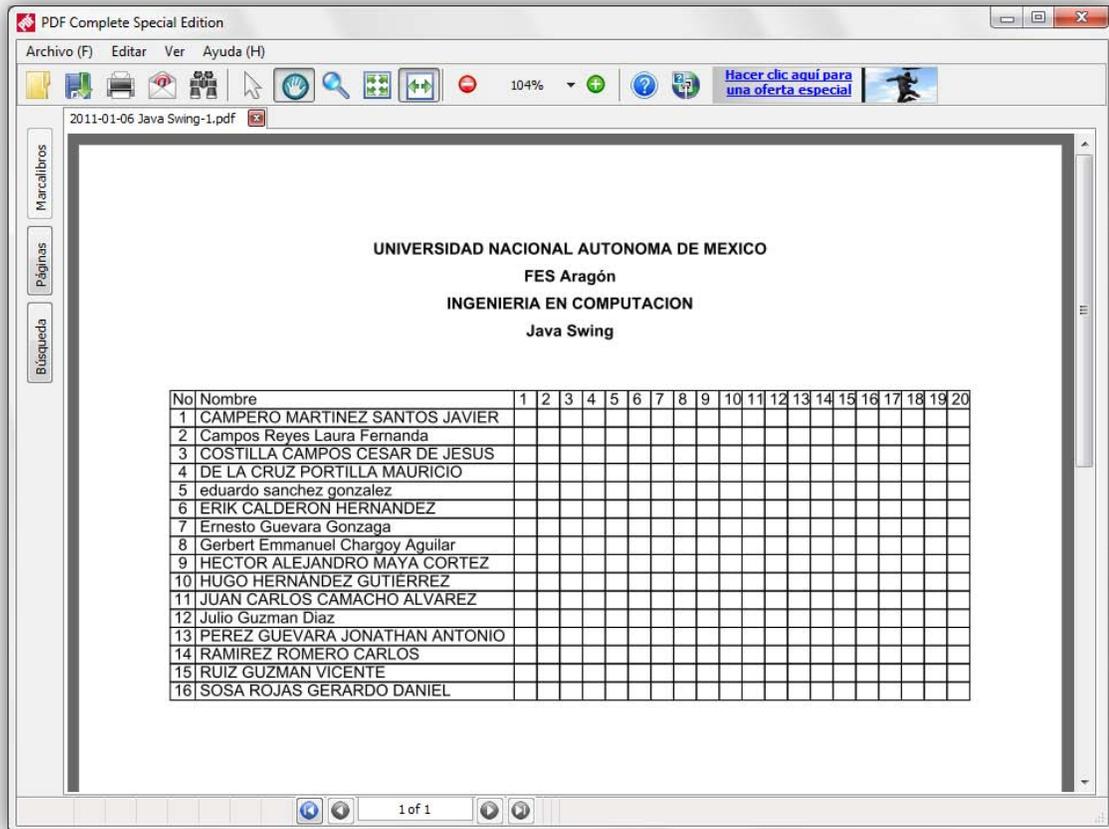


Figura 4-101 Lista en Formato PDF

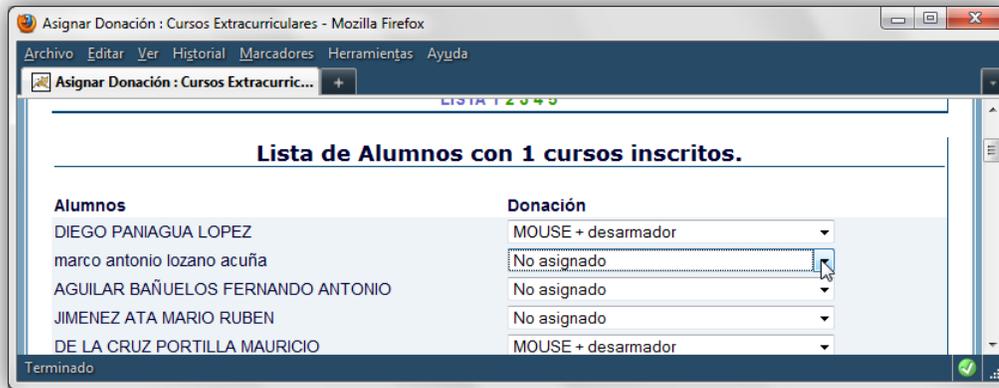
## ASIGNAR DONACIÓN

Para poder asignar la donación que el Alumno deberá entregar en el CAE se ingresará al apartado “Donación” donde se desplegará la lista de los alumnos de acuerdo a la cantidad de cursos que estos hayan inscrito.



Figura 4-102 Lista de Donaciones





**Figura 4-103 Lista de Alumnos con 1 Curso Inscrito**



**Figura 4-104 Asignación de Donaciones**

En caso de ocurrir un error se mostrara el siguiente mensaje.



**Figura 4-105 Error en la Asignacion de Donaciones**

Y en caso de que cada elemento se haya actualizado satisfactoriamente se mostrará el siguiente. Y por tanto se mostrará al alumno la donación que le fue asignada.



**Figura 4-106 Asignación de Donaciones Exitosa**





# *Capítulo 5*

## *CONCLUSIONES Y RECOMENDACIONES*

*“Enseñar, que es lo más bello y honroso del mundo”*  
*José Martí*



Se presentan las conclusiones y recomendaciones que sirvan para continuar con los esfuerzos realizados en programación de sistemas web y, en general, para la ingeniería de la computación, con la finalidad de aportar al conocimiento, por medio de la experiencia obtenida en la creación del presente sistema.

Durante el análisis del sistema se plantearon varios objetivos para la creación del mismo, dichos objetivos se fueron cumpliendo uno a uno, como resultado se obtuvo un sistema de información que permite acceder a una base de datos con una estructura predeterminada, la cual es capaz de permitir que los usuarios realicen operaciones de lectura y escritura en las tablas de la base por medio de consultas.

El sistema solo permite realizar dichas consultas de acuerdo al tipo de usuario que inicia una sesión en el sistema, por tanto, restringe al usuario de acuerdo a las reglas de negocio que fueron planteadas.

Debido a que el sistema ya fue utilizado en un proceso de inscripción, se pueden generar conclusiones con relación al desempeño del sistema en un ambiente real, mismas que arrojan como resultado; la eficiencia del sistema al momento de realizar las operaciones durante el uso del mismo, y esto se concluye debido a la falta de errores durante el proceso.

Es importante no dejar a un lado, la gran satisfacción que causó a los usuarios de todos los niveles la implementación del sistema, ya que con ello se solucionaron los diversos problemas que se presentaban durante el proceso de inscripción de cursos. Sin embargo, esta satisfacción fue mayor, al momento de estar en contacto con el sistema, y observar la funcionalidad, sencillez, utilidad y seguridad del mismo.

A pesar del poco tiempo con el que se contó, se realizaron suficientes pruebas para determinar los posibles fallos y así mismo se logró solucionar cada uno de ellos.

En la etapa de implantación se presentaron algunos percances, los cuales no se previeron en un inicio, estos fueron con respecto a la configuración del servidor donde sería alojado el sistema. Dicha configuración fue modificada en poco tiempo de forma eficiente, y con ello se instaló correctamente el sistema.

Durante la elaboración de este sistema de información, apliqué los conocimientos aprendidos durante mi formación, los que me ayudaron a solucionar un problema real, mismo que diseñé y desarrollé solo.

En caso de que se pretenda realizar un sistema de información con características similares al que se realizó, se recomienda ampliamente; seguir paso a paso cada uno de los lineamientos que la ingeniería de software nos proporciona, ya que con ello se eliminan de antemano muchos problemas que suelen presentarse durante la creación de sistemas.

En el caso específico de los sistemas Web se recomienda hacer pruebas exhaustivas de la compatibilidad en diversos exploradores de las páginas



generadas. Este acto permitirá que nadie tenga problemas al navegar por el sistema. La búsqueda por la compatibilidad debe considerar diversos Sistemas Operativos, navegadores en desuso e incluso las nuevas tecnologías. También se debe contemplar el correcto uso de los estándares definidos por el World Wide Web Consortium (W3C). Por dichas razones se recomienda el uso de XHTML y CSS en vez de HTML, así como la abstinencia en el uso de tecnologías tales como JavaScript y Flash, mismas que continuamente llevan a problemas de compatibilidad.

Una vez terminado el sistema y observado su eficacia, se recomienda ampliamente la creación de sistemas en la automatización de procesos tales como registros, inscripciones y administración de recursos e información en general.





## *Bibliografía*

Griffiths, Patrick. (2006). HTML dog: The best-practice guide to XHTML & CSS. (1era ed.). Berkeley, California, USA: New Riders.

Kofler, Michael. (2005). The definitive guide to MySQL5. (3era ed.). Berkeley, California USA: Apress.

Lea, Chris. (2002). PHP MySQL website programming: problem, design, solution. (1era ed.). Berkley, California, USA: Apress.

Mark W. Maier, Eberhardt Rechtin. (2009). The Art of systems architecting. (3era ed.). Boca Raton, Florida, USA: CRC.

Quigley, Ellie. (2006). PHP and MySQL by example. (1era ed.). Upper Saddle River, New Jersey, USA: Prentice Hall.

Stucky, Matthew. (2001). MySQL: building user interfaces. (1era ed.). Indianapolis, Indiana, USA: New Riders.

Vaswani, Vikram. (2007). PHP programming solutions. (1era ed.). New York, USA: McGraw-Hill.