



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE ESTUDIOS SUPERIORES
ARAGÓN

“Desarrollo del Sistema de Informe PRIDE (SIP)”

BAJO LA MODALIDAD DE:
DESARROLLO DE UN CASO PRÁCTICO

QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN

PRESENTA:
CARMONA SOTO EDWARD ALBERTO

ASESOR:
VILLEGAS GONZÁLEZ JUAN CARLOS

MÉXICO

2014





Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



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.

"La solución a un problema cambia el problema"

John Peers

Agradecimientos

A mis Padres

Por haberme apoyado en todo momento, por sus consejos, sus valores, por la motivación constante que me ha permitido ser una persona de bien, pero más que nada, por su amor.

A mi pareja

Quien me apoyo y alentó para continuar, cuando parecía que me iba a rendir, por todo su amor, y por siempre estar a mi lado. ¡Gracias!

A la UNAM

Por brindarme una formación académica, por permitirme ser parte de esta institución y darme grandes momentos durante los 3 años de CCH y los 4 años de la carrera, por hacer de mí un profesional.

A la UAM

Por darme la oportunidad para desarrollarme profesionalmente, por toda la experiencia adquirida.

Al Ing. Juan Carlos Villegas González

Por su gran apoyo, por su paciencia, por impulsar el desarrollo de este trabajo.

A mis profesores

Quienes nunca desistieron al enseñarme, por los conocimientos ofrecidos semestre a semestre y por enseñarme lo que significa ser profesional.

A todos los que me apoyaron para escribir y concluir esta tesis. Para ellos es esta dedicatoria de tesis, pues a ellos se las debo por su apoyo incondicional.

1 Tabla de contenido

Introducción	IX
Justificación	X
Desarrollo de tesis	XI
1 Capítulo I: Análisis y descripción de la problemática actual	2
1.1 Objetivo general.....	2
1.2 Objetivos específicos.....	3
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.....	5
1.8 Problemática.....	5
1.9 Requerimientos de la base de datos.....	6
1.9.1 Requerimientos por tipo de usuario.....	6
1.10 Selección de software.....	7
1.10.1 Análisis de modelo y manejadores de bases de datos.....	7
1.10.2 Selección de modelo y manejador de bases de Ddatos.....	14
1.10.3 Análisis y selección de tipos de tablas.....	14
1.10.4 Análisis del lenguaje de programación.....	16
1.10.5 Selección de lenguaje de programación.....	19
1.10.6 Análisis de Framework del lado del servidor y de lado del cliente.....	19
1.2 1.11 Solución propuesta.....	28
1.3 1.12 Glosario.....	33
2 Capítulo II: Diseño y arquitectura del sistema	38
2.1 Diseño funcional.....	38
2.2 Diseño detallado.....	39
2.2.1 Módulos comunes.....	39
2.2.2 Módulos del académico.....	39
2.2.3 Módulos del administrador.....	39
2.3 Base de datos.....	40
2.3.1 Diseño de la base de datos.....	40

2.3.2	Análisis de requisitos	40
2.3.3	Diseño conceptual	42
2.3.4	Diseño lógico	46
2.4	Aspectos técnicos.....	47
2.5	Arquitectura del sistema	50
2.6	Diseño de pantallas	52
2.7	Compatibilidad	56
2.8	Seguridad	57
2.8.1	Seguridad interna	57
2.8.2	Ataques externos (SQL Injection y XSS)	57
3	Capítulo III: Desarrollo del Sistema	61
3.1	Adecuación del entorno	61
3.2	Desarrollo de módulos	61
3.3	Descripción de funciones.....	61
3.3.1	Descripción de controladores	61
3.3.2	Descripción de modelos.....	65
3.3.3	Descripción de helpers.....	66
4	Capítulo IV: Implantación del Sistema.....	68
4.1	Implantación	68
4.1.1	Instalación del sistema.....	68
4.1.2	Carga inicial de datos.....	68
4.1.3	Pruebas de implantación	68
4.1.4	Preparación del mantenimiento	68
4.2	Pruebas.....	69
4.2.1	Pruebas unitarias.....	69
4.2.2	Pruebas de integración	69
4.2.3	Pruebas en navegadores.....	69
4.2.4	Pruebas de saturación	70
4.3	Presentación y aprobación del sistema.....	70
4.4	Formación de usuarios	70
4.4.1	Elaboración del manual de usuarios.....	70
4.4.2	Formación de usuarios finales	70

4.5 Mantenimiento	70
4.5.1 Mejoras	70
4.6 Manual del usuario.....	72
4.6.1 Opciones generales	72
4.6.2 Opciones del académico	78
4.6.3 Opciones del administrador	95
5 Conclusiones	105
Bibliografía.....	106

Índice de figuras

Figura 1-1 Ciclo del software	3
Figura 1-2 Proceso actual de informe PRIDE	5
Figura 1-3 Acceder al sistema	28
Figura 1-4 Inicio de sesión	29
Figura 1-5 Validación de seguridad	30
Figura 1-6 Creación de un informe	31
Figura 1-7 Editar un informe	32
Figura 2-1 Casos de uso	38
Figura 2-2 Modelo de la base del sistema	42
Figura 2-3 Tabla actividades y actividad_tipo	43
Figura 2-4 Tabla asignaturas y carreras.....	43
Figura 2-5 Tabla extraordinarios y facultades.....	44
Figura 2-6 Tabla informes y registro.....	44
Figura 2-7 Tabla títulos y títulos_informe.....	45
Figura 2-8 Tabla usuarios y usuarios_carreras.....	45
Figura 2-9 Relaciones de modelo de base del sistema	46
Figura 2-10 Interfaz MySQL Workbench	48
Figura 2-11 Interfaz XAMPP	48
Figura 2-12 Interfaz Netbeans	49
Figura 2-13 Interfaz de Mozilla Firefox.....	49
Figura 2-14 Interfaz de Firebug.....	50
Figura 2-15 Modelo Vista Controlador (MVC)	51
Figura 2-16 Interfaz: Página Principal.....	52
Figura 2-17 Interfaz: Formulario de inicio de sesión	53
Figura 2-18 Interfaz: Formulario de registro.....	53
Figura 2-19 Interfaz: Menú del usuario.....	54
Figura 2-20 Interfaz: Cambio de contraseña	54
Figura 2-21 Interfaz: Cambio de información de perfil	54
Figura 2-22 Interfaz: Menú para crear informe	55
Figura 2-23 Interfaz: Historial de informes.....	55

Figura 2-24 Interfaz: Activar usuarios.....	56
Figura 2-25 Interfaz: Modificar usuarios.....	56
Figura 4-1 Página Inicio del Sistema de Informe PRIDE	72
Figura 4-2 Opción de registro.....	73
Figura 4-3 Formulario de registro	73
Figura 4-4 Recordar Clave	74
Figura 4-5 Inicio de sesión	75
Figura 4-6 Error inicio de sesión	75
Figura 4-7 Inicio de sesión correcto	76
Figura 4-8 Perfil	76
Figura 4-9 Cambiar clave	77
Figura 4-10 Editar perfil	78
Figura 4-11 Cerrar sesión	78
Figura 4-12 Página de inicio de sistema de informe PRIDE.....	79
Figura 4-13 Nivel académico: título o grado.....	80
Figura 4-14 Nivel académico: curso.....	81
Figura 4-15 Trayectoria académica	82
Figura 4-16 Trayectoria profesional	83
Figura 4-17 Labor docente	84
Figura 4-18 Asignaturas	85
Figura 4-19 Exámenes extraordinarios.....	86
Figura 4-20 Documentos de impartición del curso	87
Figura 4-21 Asesorías y tutorías a alumnos.....	88
Figura 4-22 Apoyo a la formación de recursos humanos	89
Figura 4-23 Productividad científica, humanística, artística o tecnológica.....	90
Figura 4-24 Productividad de la docencia	91
Figura 4-25 Difusión, extensión y servicio a la comunidad	92
Figura 4-26 Participación institucional.....	93
Figura 4-27 Terminar informe	94
Figura 4-28 Historial	94
Figura 4-29 Lista de informes	95
Figura 4-30 Menú del administrador	96
Figura 4-31 Notificación no hay informes	96
Figura 4-32 Opciones de control	97
Figura 4-33 Activación de usuarios.....	97
Figura 4-34 Confirmación de activación de usuario	97
Figura 4-35 Mensaje de activación de usuario	98
Figura 4-36 Modificar usuarios	98
Figura 4-37 Formulario de modificación de usuario	99
Figura 4-38 Mensaje de modificación de usuario	99
Figura 4-39 Administración de facultades y carreras.....	100
Figura 4-40 Formulario para agregar facultad.....	100
Figura 4-41 Mensaje de agregar facultad	100
Figura 4-42 Formulario para editar facultad.....	101
Figura 4-43 Mensaje de editar facultad	101
Figura 4-44 Mensaje de eliminar facultad.....	101
Figura 4-45 Lista de carreras	102

Figura 4-46 Formulario para agregar carrera.....	102
Figura 4-47 Mensaje de agregar carrera.....	102
Figura 4-48 Formulario para editar carrera	103
Figura 4-49 Mensaje de editar carrera.....	103
Figura 4-50 Mensaje de eliminar carrera.....	103

Índice de tablas

Tabla 1-1 Calendario de actividades	4
Tabla 1-2 Comparativa de lenguajes de programación del lado del servidor	19
Tabla 1-3 Comparativa de frameworks PHP.....	23
Tabla 1-4 Comparativa de frameworks de lado del cliente	27
Tabla 2-1 Requisitos del usuario	41
Tabla 2-2 Requisitos del informe	41
Tabla 2-3 Requisitos de los títulos	41
Tabla 2-4 Requisitos de las actividades.....	41
Tabla 2-5 Requisitos de las asignaturas.....	41
Tabla 2-6 Requisitos de los extraordinarios	41
Tabla 3-1 Controlador: Principal.....	62
Tabla 3-2 Controlador: Login.....	62
Tabla 3-3 Controlador: Registro.....	62
Tabla 3-4 Controlador: Admin	62
Tabla 3-5 Controlador: Carreras.....	63
Tabla 3-6 Controlador: Facultades	63
Tabla 3-7 Controlador: Historial	63
Tabla 3-8 Controlador: Actividad.....	63
Tabla 3-9 Controlador: Asignatura.....	63
Tabla 3-10 Controlador: Extraordinario	64
Tabla 3-11 Controlador: Informe	64
Tabla 3-12 Controlador: Titulo.....	64
Tabla 3-13 Controlador: Pride_pdf.....	64
Tabla 3-14 Controlador: Usuario.....	64

Introducción

El Programa de Primas al Desempeño del Personal Académico de Tiempo Completo (PRIDE), implementado por la Universidad Nacional Autónoma de México, tiene como finalidad reconocer y estimular al personal académico de tiempo completo, como partícipe del área de recursos humanos, docencia, investigación y vinculación, incrementando los beneficios de la cultura, fomentar su superación y elevar el nivel de productividad y calidad en su desempeño.

El personal académico que podrá participar en el PRIDE deberá tener nombramiento de profesor, investigador o técnico académico ordinario, de tiempo completo, o ser personal de tiempo completo contratado por el procediendo establecido en el artículo 51 del estatuto del personal académico (EPA).

El PRIDE analiza las actividades realizadas por el académico, así como sus logros y habilidades, los puntos a evaluar por este por el programa son:

- A. Formación académica y trayectoria académica y/o profesional
- B. Labores docentes y de formación de recursos humanos
- C. Productividad académica
- D. Difusión, extensión y servicios a la comunidad
- E. Participación institucional

Justificación

En estos momentos el informe cuenta con un formato que cubre todos los puntos necesarios para la evaluación del académico, este formato es elaborado (cada 3 años) por el académico interesado, mediante el llenado de un archivo de procesador de texto, el académico por lo tanto tiene la responsabilidad del manejo, orden y disponibilidad de sus informes.

Las responsabilidades que adquiere el académico pueden resultar complicadas, esto debido a las múltiples formas de pérdida de información. Por lo tanto se deben buscar alternativas, tales como la creación de un sistema, capaz de realizar las tareas necesarias para el correcto manejo del informe.

El Sistema de Informe PRIDE (SIP) servirá como una herramienta para facilitar al académico la elaboración, manejo, orden y disponibilidad del informe PRIDE, así como de los materiales adjuntos para la evaluación del personal.

El SIP deberá contar con las siguientes características:

- Debe ser capaz de permitir el llenado conforme al formato del informe PRIDE.
- Debe ser capaz de permitir la modificación y eliminación del informe creado.
- Debe ser capaz de mostrar el historial de informes creados por el usuario.
- Debe ser capaz de imprimir el formato entregable una vez creado el informe.

Desarrollo de tesis

Para llevar a cabo el proyecto, se ha dividido en etapas, para un mejor desarrollo, siendo estas las siguientes:

En el capítulo uno Análisis y descripción de la problemática actual, se analizarán el objetivo del sistema, los requerimientos necesarios para el sistema, el lenguaje de programación, el manejador de bases de datos.

En el capítulo dos Diseño y arquitectura del sistema, se crearán los modelos conceptual y funcional del sistema, en base a estos se creará el modelo de la base de datos, esto se hará tomando en cuenta el enfoque de la ingeniería de software y aplicando los métodos que nos proporciona.

En el capítulo tres Desarrollo del sistema, se retomarán los análisis del capítulo uno y el diseño del capítulo dos, para desarrollar el sistema, teniendo muy en cuenta el objetivo del sistema. Es en esta etapa donde se creará el código computacional necesario para que el sistema funcione.

En el capítulo cuatro Implantación y control, el sistema se implantará para su uso. Una vez que el sistema esté en marcha, se mostrarán las funciones y forma de uso del sistema, tanto para el usuario de tipo académico, como para el usuario administrador del sistema.

Por último se darán las conclusiones y recomendaciones, sobre el desarrollo y funcionamiento del sistema.



**“Si no se ha hecho una buena definición del problema,
podrías estar resolviendo el problema equivocado durante la
construcción”**

Steve McConnell

1 Capítulo I: Análisis y descripción de la problemática actual

El objetivo de este capítulo es analizar el objetivo del sistema, los requerimientos necesarios para el correcto funcionamiento del sistema, elegir el lenguaje de programación y manejador de bases de datos más adecuado para su desarrollo.

Esta etapa es muy importante porque de ella dependerá el funcionamiento del sistema, ya que si se omite un requerimiento, o se hace una mala elección del manejador de bases de datos, o del lenguaje de programación, o se define mal el objetivo del sistema, no funcionará correctamente y retrasará el desarrollo del mismo.

1.1 Objetivo general

El Sistema de Informe PRIDE debe ayudar al profesional académico a la creación y administración de sus informes y materiales adjuntos, brindándole la posibilidad de consultar los detalles sobre su informe e impresión del formato entregable en cualquier momento.

Para el desarrollo de este sistema y como parte de la ingeniería de software se tomará como referencia el ciclo de vida del software, se puede apreciar el ciclo en la figura 1-1, en él aparecen las etapas a seguir para el desarrollo de software.

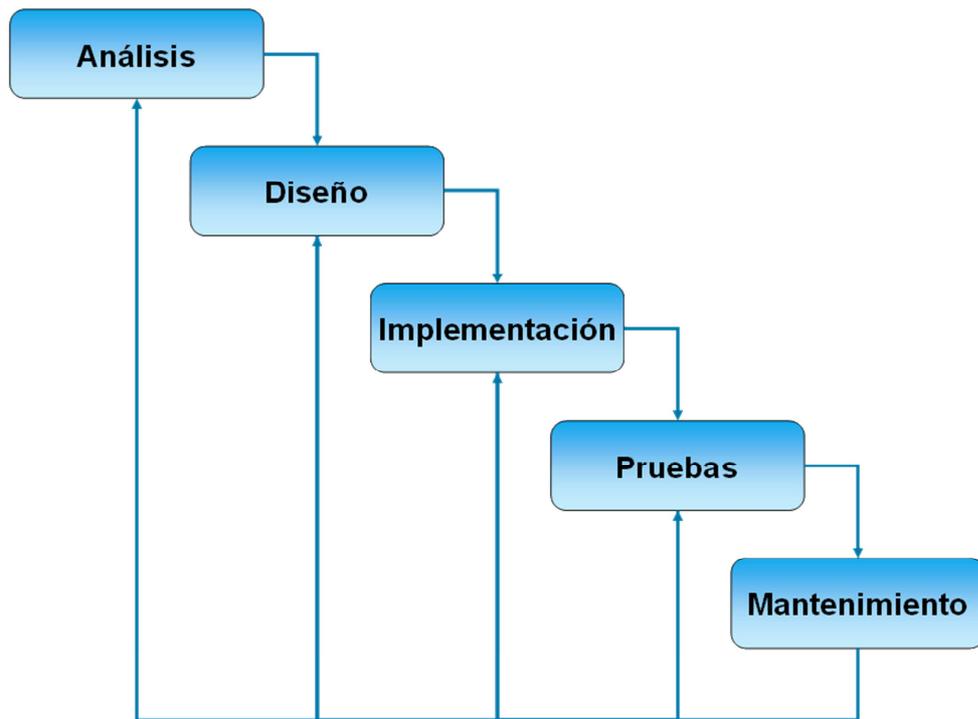


Figura 1-1 Ciclo del software

1.2 Objetivos específicos

El profesional académico de tiempo completo con posibilidad de realizar el informe PRIDE, debe registrarse para después ser dado de alta por el administrador del sistema, teniendo en cuenta los datos que fueron ingresados por el profesional, asegurándose de que este cumpla con los requisitos para participar en el PRIDE.

Después de ser dado de alta por el sistema, el académico podrá realizar informes y adjuntar los materiales necesarios para la realización del informe. Después de finalizado el informe este se agregará al historial del académico y podrá imprimirse el formato.

El académico podrá consultar el informe en cualquier momento, y tendrá las opciones de editar o eliminar tanto el informe como los documentos adjuntos.

1.3 Calendario de actividades

Para el desarrollo de cualquier sistema es necesario un calendario de actividades, para un mejor manejo de las actividades que se deben realizar y al mismo tiempo tener en cuenta el tiempo disponible para cada actividad.

El sistema será desarrollado en 12 semanas que durante los meses de julio, agosto y septiembre mostrados en la tabla 1-1.

ID	Task Name	Start	Finish	Duration	jul 2013		ago 2013					sep 2013				
					21/7	28/7	4/8	11/8	18/8	25/8	1/9	8/9	15/9	22/9	29/9	
1	Análisis	19/07/2013	01/08/2013	2w	█											
2	Diseño	29/07/2013	16/08/2013	3w		█										
3	Desarrollo	14/08/2013	17/09/2013	5w				█								
4	Pruebas	09/09/2013	04/10/2013	4w								█				
5	Implantación	18/09/2013	24/09/2013	1w										█		
6	Mantenimiento	26/09/2013	02/10/2013	1w												█

Tabla 1-1 Calendario de actividades

1.4 Obtención de requisitos

Tomando en cuenta las necesidades del sistema, el formato del informe y la experiencia en la creación por parte de algunos profesores se obtuvieron los requisitos necesarios para que el sistema SIP (Sistema de Informe PRIDE).

Los resultados incluirán la opción del manejo de materiales extras; que son adjuntados con el informe, esto mejorará el orden y disponibilidad brindada por el sistema.

1.5 Definición del sistema

El sistema servirá como herramienta para los académicos que realizan informes PRIDE, ayudándoles con la administración de sus informes y materiales en un solo lugar, simplificando además el proceso de llenado del informe para su creación.

El administrador del sistema podrá dar de alta usuarios, restringirles el acceso, podrá editar y eliminar los informes y materiales de todos los usuarios, así como modificar la información personal de todos los usuarios en el sistema.

1.6 Alcance

El sistema será un historial de los informes PRIDE generados por los usuarios y estará disponible siempre que el académico los necesite. Al estar basado en el formato, el sistema cubrirá los puntos necesarios para la correcta elaboración del informe.

SIP permitirá un mejor manejo en la creación, administración y disponibilidad de los informes, con esto simplificará el proceso que rodea al informe PRIDE.

1.7 Ambiente de trabajo

Un punto importante en el desarrollo de un sistema, es el ambiente de trabajo, por lo tanto, se llegó a la conclusión de que el lugar apropiado para realizar el desarrollo son las instalaciones de la jefatura de Ingeniería en Computación de la Facultad de Estudios Superiores Aragón, las instalaciones cuentan con lo necesario para un buen desarrollo como lo son: equipo de cómputo adecuado, conexión a internet, además de permitir el contacto con los profesores, lo cual traerá beneficios en el desarrollo del sistema.

1.8 Problemática

Actualmente cada académico es el encargado de crear sus informes PRIDE en base al formato, dejándoles la responsabilidad de llevar el orden de los informes que generaban, y aunque esto no resultaba muy difícil por cuestiones ajenas como pérdida de información o de equipo de cómputo, la pérdida de información por fallas de hardware o software, entre otras, los profesores no contaban con toda la información acerca de los informes.

La figura 1-2 muestra las acciones a realizar para generar un informe PRIDE.

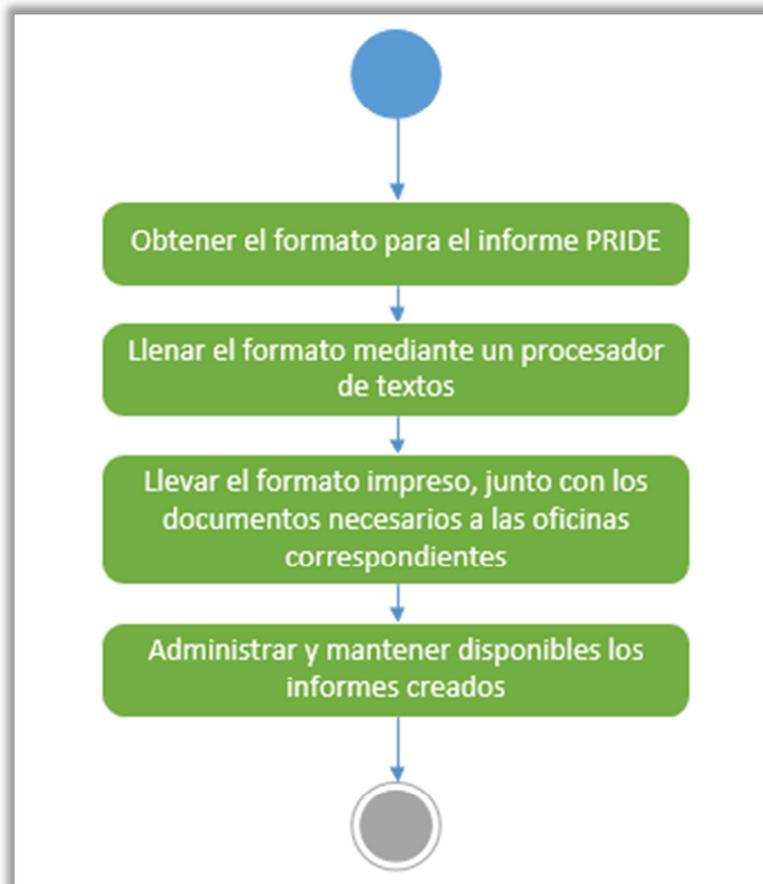


Figura 1-2 Proceso actual de informe PRIDE

El proceso que actualmente realiza el académico implica una inversión en tiempo, tanto para llenar el formato e imprimirlo, como para ir a las oficinas correspondientes y entregar los documentos necesarios.

Además del tiempo dedicado al informe, la disponibilidad y administración de los informes es responsabilidad del académico, estos elementos son suficientes para justificar la creación del sistema, buscando automatizar el proceso y al mismo tiempo quitar el peso al académico en cuanto a la administración de los informes, permitiendo que mediante el sistema él pueda tener a su disposición los informes PRIDE en cualquier momento.

La solución propuesta ante esta problemática, es el desarrollo de un sistema Web¹, al tratarse de un sistema Web se puede tener acceso desde cualquier parte del mundo, siempre y cuando se cuente con una conexión a internet, esto también hace que el usuario no tenga la necesidad de instalar algún software en específico, más que un navegador Web².

1.9 Requerimientos de la base de datos

Desde el inicio del análisis del sistema se tomó en consideración la inclusión de una base de datos³ para el manejo apropiado de la información.

La información que será almacenada en la base de datos incluirá; los datos de los académicos, las facultades y carreras a las cuales están adscritos, así como, toda la información referente al informe; los títulos o grados del académico, las materias y exámenes extraordinarios a su cargo, actividades como: cursos de actualización, vinculación, entre otras.

1.9.1 Requerimientos por tipo de usuario

Académico: Nombre de usuario, Nombre, Apellidos, Correo, Contraseña.

¹ Los "sistemas Web" o también conocido como "aplicaciones Web" son aquellos que están creados e instalados no sobre una plataforma o sistemas operativos (Windows, Linux). Sino que se alojan en un servidor en Internet o sobre una intranet (red local). Su aspecto es muy similar a páginas Web que vemos normalmente, pero en realidad los 'sistemas Web' tienen funcionalidades muy potentes que brindan respuestas a casos particulares. Para utilizar las aplicaciones Web no es necesario instalarlas en cada computadora ya que los usuarios se conectan a un servidor donde se aloja el sistema.

² Un **navegador** o **navegador web** es un software que permite el acceso a Internet, interpretando la información de archivos y sitios web para que éstos puedan ser leídos. La funcionalidad básica de un navegador web es permitir la visualización de documentos de texto, posiblemente con recursos multimedia incrustados. Además, permite visitar páginas web y hacer actividades en ella, es decir, podemos enlazar un sitio con otro, imprimir, enviar y recibir correo, entre otras funcionalidades más.

³ Una **base de datos** o **banco de datos** es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. Actualmente, y debido al desarrollo tecnológico de campos como la informática, la mayoría de las bases de datos están en formato digital (electrónico), y por ende se ha desarrollado y se ofrece un amplio rango de soluciones al problema del almacenamiento de datos.

Informe: Tipo de actividad, Fecha de la actividad, Institución donde se realizó la actividad, Materias impartidas, Extraordinarios realizados, Fecha de creación y modificación del informe, Archivo adjunto al informe, Títulos del académico e instituciones donde obtuvo los títulos al momento de realizar el informe.

1.10 Selección de software

Para el desarrollo del sistema es necesario realizar una evaluación de las tecnologías necesarias en cuanto a manejador y modelo de base de datos, lenguaje de programación⁴ del lado del servidor⁵, uso de Framework⁶ del lado del servidor y de lado del cliente⁷.

1.10.1 Análisis de modelo y manejadores de bases de datos

Los manejadores de Bases de Datos son los encargados de realizar las operaciones en la base de datos, la correcta elección del manejador es de suma importancia, y la elección dependerá de las necesidades tanto del sistema; como del Hardware y Software en que se encontrará instalado el sistema.

En este momento existen 2 modelos de gestión de datos en el mercado, base de datos relacionales y NoSQL, estos modelos se disputan el ser la mejor opción para el manejo de las bases de datos, cada uno tiene características que los vuelven más apropiados para cubrir las necesidades de determinados sistemas.

1.10.1.1 Base de datos relacionales

Es el modelo más usado en la actualidad, fue propuesto en por Edgar Codd, se basa en tablas, que está formada por filas (registros) y columnas (campos); y su idea fundamental es relacionar las tablas creando así un conjunto de datos llamados tuplas.

Los sistemas relacionales son importantes porque ofrecen muchos tipos de procesos de datos, como: simplicidad y generalidad, facilidad de uso para el usuario final, períodos cortos de aprendizaje y las consultas de información se especifican de forma sencilla.

Las tablas son un medio de representar la información de una forma más compacta y es posible acceder a la información contenida en dos o más tablas.

⁴ Un **lenguaje de programación** es un lenguaje formal diseñado para expresar procesos que pueden ser llevados a cabo por máquinas como las computadoras. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana.

⁵ Véase en el glosario.

⁶ Véase en el glosario.

⁷ El **cliente** es una aplicación informática o un computador que consume un servicio remoto en otro computador, conocido como servidor, normalmente a través de una red de telecomunicaciones

Las bases de datos relacionales están constituidas por una o más tablas que contienen la información ordenada de una forma organizada. Cumplen las siguientes leyes básicas:

- Generalmente, contendrán muchas tablas.
- Una tabla sólo contiene un número fijo de campos.
- El nombre de los campos de una tabla es distinto.
- Cada registro de la tabla es único.
- El orden de los registros y de los campos no está determinados.
- Para cada campo existe un conjunto de valores posible.

1.10.1.1.1 Modelo entidad relación

Este modelo se obtiene en tiempo de diseño de la base de datos. Fue propuesto por Peter Chen en 1976 y desde entonces se viene utilizando de una forma muy global.

Se caracteriza por utilizar una serie de símbolos y reglas para representar los datos y sus relaciones. Con este modelo conseguimos representar de manera gráfica la estructura lógica de una base de datos. Los principales elementos del modelo entidad-relación son las entidades con sus atributos y relaciones entre entidades.

1.10.1.1.2 Elementos del modelo entidad relación

Entidad: Se trata de un objeto del que se recoge información de interés de cara a la base de datos. Gráficamente se representan mediante un rectángulo. Un ejemplo sería la entidad banco, donde se recogerían los datos relativos a ese banco, como puede ser el nombre, el número de sucursal, la dirección, etc.

Las entidades pueden ser fuertes o débiles. Las fuertes son las que no dependen de otras entidades para existir, mientras que las entidades débiles siempre dependen de otra entidad sino no tienen sentido por ellas mismas.

Relación: Podemos definir la relación como una asociación de dos o más entidades. A cada relación se le asigna un nombre para poder distinguirla de las demás y saber su función, dentro del modelo entidad-relación. Otra característica es el grado de relación, siendo las de grado 1 relaciones que solo relacionan una entidad consigo misma. Las de grado 2 son relaciones que asocian dos entidades distintas, y las de grado n que se tratan de relaciones que unen más de dos entidades.

Las relaciones pueden representarse gráficamente con rombos, dentro de ellas se coloca el nombre de la relación.

- **1:1:** Uno a uno, a cada ocurrencia de una entidad le corresponde como máximo una ocurrencia de la otra entidad relacionada.
- **1:N:** Uno a Muchos, a cada ocurrencia de la entidad A le pueden corresponder varias de la entidad B.

- **N:M:** Muchos a muchos, cada ocurrencia de una entidad puede contener varias de la otra entidad relacionada y viceversa.

Para finalizar las características de la relación tenemos la cardinalidad que define el número máximo y mínimo de ocurrencias de cada tipo de entidad. Se representa con los valores máximo coma mínimo encerrado entre paréntesis encima de la relación.

Atributo: Se define como cada una de las propiedades de una entidad o relación. Cada atributo tiene un nombre y todos los posibles valores que puede tener. Dentro de una entidad tiene que haber un atributo principal que identifica a la entidad y su valor tiene que ser único. Un ejemplo de atributo principal sería el curp dentro de la entidad persona.

SQL (Structured Query Language): Es un lenguaje declarativo para el manejo de bases de datos mediante consultas, que permiten crear, seleccionar, actualizar y eliminar información dentro de la base de datos. Es el lenguaje principal de la mayoría de los modelos de Base de Datos.

1.10.1.1.3 Sistema de gestión de bases de datos relacionales

Un **Sistema de Gestión de Bases de Datos** (SGBD): es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos, además de proporcionar herramientas para añadir, borrar, modificar y analizar los datos. Los usuarios pueden acceder a la información usando herramientas específicas de interrogación y de generación de informes. Los SGBD también proporcionan métodos para mantener la integridad de los datos, para administrar el acceso de usuarios a los datos y para recuperar la información si el sistema se corrompe. Permite presentar la información de la base de datos en variados formatos.

Para administrar la base de datos necesitaremos un SGBD, a continuación se listan los más populares junto a sus características más importantes.

MySQL: Es un sistema administrativo relacional de bases de datos. Este tipo de bases de datos; puede ejecutar desde acciones tan básicas, como insertar y borrar registros, actualizar información o hacer consultas simples, hasta realizar tareas tan complejas como la aplicación lo requiera.

MySQL es muy conocido y ampliamente usado por su simplicidad y notable rendimiento. Aunque carece de algunas características avanzadas disponibles en otros SGBD del mercado, es una opción atractiva en el desarrollo de aplicaciones web, precisamente por su facilidad de uso y tiempo reducido de puesta en marcha. Esto y su libre distribución en Internet bajo licencia GPL⁸ le otorgan como beneficios adicionales (no menos importantes) contar con un alto grado de estabilidad y un rápido desarrollo.

MySQL es un servidor multi-usuarios muy rápido y robusto de ejecución de instrucciones en paralelo, es decir, que múltiples usuarios distribuidos a lo largo de una red

⁸ Véase el glosario.

local o internet podrán ejecutar distintas tareas sobre las bases de datos localizadas en un mismo servidor.

MySQL está disponible para múltiples plataformas, sin embargo, las diferencias con cualquier otra plataforma son prácticamente nulas, ya que la herramienta utilizada en este caso es el cliente mysql-client, que permite interactuar con un servidor MySQL (local o remoto) en modo texto. De este modo es posible realizar todos los ejercicios sobre un servidor instalado localmente o, a través de Internet, sobre un servidor remoto.

Oracle database: es un sistema de gestión de bases de datos objeto-relacional (ORDBMS por el acrónimo en inglés de Object-Relational Data Base Management System), desarrollado por Oracle Corporation.

Se considera a Oracle como uno de los sistemas de bases de datos más completos, destacando:

- soporte de transacciones,
- estabilidad,
- escalabilidad y
- Soporte multiplataforma.

Su dominio en el mercado de servidores empresariales ha sido casi total hasta hace poco, recientemente sufre la competencia del Microsoft SQL Server de Microsoft y de la oferta de otros RDBMS con licencia libre como PostgreSQL, MySQL o Firebird. Las últimas versiones de Oracle han sido certificadas para poder trabajar bajo GNU/Linux.

Postgres: Es un gestor de bases de datos orientadas a objetos (ORDBMS) muy conocido y usado en entornos de software libre porque cumple los estándares SQL92 y SQL99, y también por el conjunto de funcionalidades avanzadas que soporta, lo que lo sitúa al mismo o a un mejor nivel que muchos SGBD comerciales.

El origen de PostgreSQL se sitúa en el gestor de bases de datos POSTGRES desarrollado en la Universidad de Berkeley y que se abandonó en favor de PostgreSQL a partir de 1994. Ya entonces, contaba con prestaciones que lo hacían único en el mercado y que otros gestores de bases de datos comerciales han ido añadiendo durante este tiempo.

PostgreSQL se distribuye bajo licencia BSD⁹, lo que permite su uso, redistribución, modificación con la única restricción de mantener el copyright del software a sus autores, en concreto el PostgreSQL Global Development Group y la Universidad de California. PostgreSQL puede funcionar en múltiples plataformas (en general, en todas las modernas basadas en Unix).

⁹ Véase el glosario.

SQL Server: Es un sistema de gestión de base de datos relacionales basado en lenguaje Transact-SQL y usa específicamente un motor de base de datos Sybase IQ, la cual brinda grandes cantidades de información a muchos usuarios de una manera simultánea.

Entre sus características destacan:

- Soporte de transacciones
- Escalabilidad, estabilidad y seguridad
- Soporta procedimientos almacenados
- Entorno gráfico de administración
- Permite trabajar en modo cliente-servidor
- Permite administrar información de otros servidores de datos.
- Licencia privada (Microsoft) etc.

Este sistema de gestión de bases incluye una versión reducida, llamada MSDE con el mismo motor de base de datos pero orientado a proyectos más pequeños, que en su versión 2005 pasa a ser el SQL Express Edition, que se distribuye en forma gratuita.

1.10.1.2 NoSQL

Es una amplia clase de sistemas de gestión de bases de datos, que difieren del modelo clásico del sistema de gestión de bases de datos relacionales en aspectos importantes, a continuación se hace un resumen de las características.

La primer característica a notar es el hecho de no usar SQL como principal lenguaje de consultas, otra característica es que los datos no tienen una definición de atributos fija, es decir: cada registro (o documento) puede contener una información con diferente forma cada vez, pudiendo así almacenar sólo los atributos que interesen en cada uno de ellos, facilitando el polimorfismo de datos bajo una misma colección de información. También se pueden almacenar estructuras de datos complejas en un sólo documento, como por ejemplo almacenar la información sobre un informe (título, cuerpo de texto, autor, etc) junto a los comentarios y etiquetas vertidos sobre el mismo, todo en un único registro. Hacerlo así aumenta la claridad (al tener todos los datos relacionados en un mismo bloque de información) y el rendimiento (no hay que hacer un JOIN¹⁰ para obtener los datos relacionados, pues éstos se encuentran directamente en el mismo documento).

Los datos almacenados al no requerir estructuras fijas como tablas, no garantizan completamente ACID¹¹ (atomicidad, coherencia, aislamiento y durabilidad), pero en cambio escalan bien horizontalmente.

¹⁰ Un **join** es una combinación de dos o más tablas de una base de datos relacional, es una instrucción de lo más imprescindible si queremos realizar una aplicación que realice un uso correcto e intensivo de un gestor de Bases de datos como MySQL. Y es que gracias a los JOIN podemos leer datos de diferentes bases de datos en una única consulta. Lo que nos permitirá diseñar fácilmente tablas relacionadas entre ellas.

¹¹ Véase en el glosario.

Con escalabilidad horizontal se refiere a la posibilidad de aumentar el rendimiento del sistema simplemente añadiendo más nodos¹², sin necesidad en muchos casos de realizar ninguna otra operación más que indicar al sistema cuáles son los nodos disponibles. Muchos sistemas NoSQL permiten utilizar consultas del tipo Map-Reduce, las cuales pueden ejecutarse en todos los nodos a la vez (cada uno operando sobre una porción de los datos) y reunir luego los resultados antes de devolverlos al cliente. La gran mayoría permiten también indicar otras cosas como el número de réplicas en que se hará una operación de escritura, para garantizar la disponibilidad.

Por último, muchos de estos sistemas realizan operaciones directamente en memoria, y sólo vuelcan los datos a disco cada cierto tiempo. Esto permite que las operaciones de escritura sean realmente rápidas. Por supuesto, trabajar de este modo puede sacrificar fácilmente la durabilidad de los datos, y en caso de la interrupción en el suministro de energía eléctrica se podrían perder operaciones de escritura. Normalmente, esto lo resuelven permitiendo que una operación de escritura haya de realizarse en más de un nodo antes de darla por válida

Por lo general, los investigadores académicos se refieren a este tipo de bases de datos como *almacenamiento estructurado*, término que abarca también las bases de datos relacionales clásicas. A menudo, las bases de datos NoSQL se clasifican según su forma de almacenar los datos, y comprenden categorías como *clave-valor*¹³, las implementaciones de Big Table, bases de datos documentales, y Bases de datos orientadas a grafos¹⁴.

1.10.1.2.1 Sistema de gestión de bases de datos NoSQL

MongoDB: MongoDB (de la palabra en inglés “humongous” que significa enorme) es un sistema de bases de datos NoSQL orientado a documentos, desarrollado bajo el concepto de código abierto.

MongoDB forma parte de la nueva familia de sistemas de base de datos NoSQL. En vez de guardar los datos en tablas como se hace en las base de datos relacionales, MongoDB guarda estructuras de datos en documentos tipo JSON¹⁵ con un esquema dinámico (MongoDB llama ese formato BSON), haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida.

El desarrollo de MongoDB empezó en octubre de 2007 por la compañía de software 10gen. Ahora MongoDB es una base de datos lista para la producción de uso y con muchas

¹² Un **nodo** es un registro que contiene un dato de interés y al menos un puntero para referenciar (apuntar) a otro nodo. Si la estructura tiene sólo un puntero, la única estructura que se puede construir con él es una lista, si el nodo tiene más de un puntero ya se pueden construir estructuras más complejas como árboles o grafos.

¹³ **Clave-valor** significa que los datos que maneja la base se obtienen a partir de una clave única. Es como si manejaras únicamente una enorme tabla, y en el caso de REDIS y otras bases de datos NoSQL, sin verte obligado a seguir los convencionalismos de una típica base de datos SQL con una definición rígida en su estructura.

¹⁴ El **grafo** es una estructura de datos, en concreto un tipo abstracto de datos (TAD), que consiste en un conjunto de nodos y un conjunto de arcos que establecen relaciones entre los nodos.

¹⁵ Véase el glosario.

características. Esta base de datos es altamente utilizada en las industrias. Craigslist¹⁶ y Foursquare¹⁷ son algunas de las empresas que utilizan este modelo de base de datos.

MongoDB se puede obtener de una forma gratuita bajo la licencia pública general de Affero (AGPL¹⁸). Los controladores de base de datos para los lenguajes de programación están bajo la licencia de Apache¹⁹. Adicional a estas licencias, 10gen ofrece una licencia comercial para MongoDB.

Apache Cassandra: es una base de datos no relacional distribuida y basada en un modelo de almacenamiento de «clave-valor», escrita en Java. Permite grandes volúmenes de datos en forma distribuida. Por ejemplo, lo usa Twitter²⁰ para su plataforma. Su objetivo principal es la escalabilidad lineal y la disponibilidad. La arquitectura distribuida de Cassandra está basada en una serie de nodos iguales que se comunican con un protocolo²¹ P2P²² con lo que la redundancia (almacenamiento de los mismos datos varias veces en diferentes lugares) es máxima.

Cassandra es desarrollada por la Apache Software Foundation. En las versiones iniciales utilizaba un API²³ propia para poder acceder a la base de datos. En los últimos tiempos están apostando por un lenguaje denominado CQL (Cassandra Query Language) que posee una sintaxis similar a SQL aunque con muchas menos funcionalidades. Esto hace que iniciarse en el uso de la misma sea más sencillo. Permite acceder en Java desde JDBC²⁴.

Big Table: es un motor de bases de datos creado por Google con las características de ser distribuido, de alta eficiencia y propietario. Está construido sobre GFS²⁵ (Google File System), y algunos otros servicios y programas de Google, y funciona sobre 'commodity hardware' (sencillos y baratos PCs con procesadores Intel).

BigTable almacena la información en tablas multidimensionales cuyas celdas están, en su mayoría, sin utilizar. Además, estas celdas disponen de versiones temporales de sus

¹⁶ **Craigslist** es una web de comunidades en línea organizadas por ciudades, que ofrece en cada una de ellas anuncios clasificados gratis sobre diversos temas: empleo, vivienda, sentimentales, venta/trueque, servicios diversos, trabajos de corta duración, etc.

¹⁷ **Foursquare** es un servicio basado en localización web aplicada a las redes sociales. La geolocalización permite localizar un dispositivo fijo o móvil en una ubicación geográfica.

¹⁸ Véase el glosario.

¹⁹ Véase el glosario.

²⁰ **Twitter**, un término inglés que puede traducirse como “gorjear” o “trinar”, es el nombre de una red de microblogging que permite escribir y leer mensajes en Internet que no superen los 140 caracteres. Estas entradas son conocidas como **tweets**.

²¹ Véase el glosario.

²² Véase el glosario.

²³ Véase el glosario.

²⁴ Véase el glosario.

²⁵ El **Sistema de Archivos Google**, en inglés **Google File System (GFS, GooFS o GoogleFS)**, es un sistema de archivos distribuido propietario desarrollado por Google Inc, que soporta toda su infraestructura informática de procesamiento de información en nube. Está especialmente diseñado para proveer eficiencia, fiabilidad de acceso a datos usando sistemas masivos de cluster de procesamiento en paralelo. La actual versión de Google File System tiene el nombre clave **Colossus**.

valores, con lo que se puede hacer un seguimiento de los valores que han tomado históricamente.

Para poder manejar la información, las tablas se dividen por columnas, y son almacenadas como 'tabletas' de unos 100-200 Mbytes cada una. Cada máquina almacena 100 tabletas, mediante el sistema 'Google File System'. La disposición permite un sistema de balanceo de carga (si una tableta está recibiendo un montón de peticiones, la máquina puede desprenderse del resto de las tabletas o trasladar la tableta en cuestión a otra máquina) y una rápida recomposición del sistema si una máquina 'se cae'.

1.10.2 Selección de modelo y manejador de bases de Ddatos

Se ha tomo la decisión de usar el modelo de bases de datos relacionadas, porque además de ser el más usado es la opción que más se aproxima a la estructura necesaria para manejar el sistema de una manera eficiente, se eligió MySQL ya que además de ser gratuito, cuenta con los recursos y herramientas necesarias para la buena administración de los datos que se tendrán almacenados.

1.10.3 Análisis y selección de tipos de tablas

Después de haber elegido el modelo de base de datos y el manejador, se elegirá el tipo de tabla a manejar para ser usado en MySQL.

ISAM: es el formato de almacenaje más antiguo, y posiblemente podría desaparecer en futuras versiones. Presenta limitaciones importantes; como la no exportación de ficheros entre máquinas de distintas arquitecturas o que no podía usar mayores de 4 GigaBytes.

En un sistema ISAM, la información se organiza en registros compuestos por campos de tamaño fijo. Los registros se almacenan secuencialmente, inicialmente para acelerar el acceso en sistemas de cinta. Un conjunto secundario de ficheros dispersos (tablas "hash") conocidos como *índices* contienen «punteros» a los registros que permiten acceder a los registros individuales sin tener que buscar en todo el fichero. Este es el punto de partida para todos los modernos sistemas de bases de datos, en los cuales los punteros que dirigen hacia otra información fueron almacenados dentro de los propios registros. El avance clave que posee ISAM es que los índices son pequeños y pueden ser buscados rápidamente, permitiendo a la base de datos acceder sólo a los registros que necesita. Modificaciones adicionales a la información no requieren cambios a otra información, sólo a la tabla y los índices.

MYISAM: es el tipo de tabla por defecto en MySQL desde la versión 3.23. Optimizada para sistemas operativos de 64 bits, permite ficheros de tamaños mayores que las ISAM. Los datos se almacenan en un formato independiente, lo que permite pasar tablas entre distintas plataformas. Los índices se almacenan en un archivo con la extensión ".MYI" y los datos en otro archivo con extensión ".MYD". Ofrece la posibilidad

de indexar campos BLOB²⁶ y TEXT²⁷. Además este tipo de tablas soportan el tipo de dato VARCHAR²⁸.

Un inconveniente es que las tablas pueden llegar a corromperse, almacenando datos incorrectos. Esto puede ser causado por:

- El proceso mysqld haya sido eliminado en el transcurso de una escritura.
- Problemas de hardware.
- Una caída del sistema durante su utilización.
- Un gusano en el código Mysql o MyISAM.

INNODB: InnoDB provee a MySQL con el soporte para trabajar con transacciones, además de hacer un mejor bloqueo de registros para las instrucciones SELECT muy parecido al usado por Oracle, con lo que incrementa el rendimiento y la concurrencia (que muchas transacciones puedan acceder a una misma base de datos a la vez) en ambientes multiusuario, por otro lado, InnoDB es el único formato que tiene MySQL para soportar llaves foráneas (FOREIGN KEY²⁹). Además de todo lo comentado, InnoDB ofrece unos rendimientos superiores a la anterior tecnología de tablas de MySQL (MyISAM).

InnoDB es un motor de bases de datos muy completo que ha sido integrado dentro de MySQL. Otras de sus características son:

- Recuperación automática ante fallas. Si MySQL se da de baja de una forma anormal, InnoDB automáticamente completará las transacciones que quedaron incompletas.
- Integridad referencial. Ahora se pueden definir llaves foráneas entre tablas InnoDB relacionadas para asegurarse de que un registro no puede ser eliminado de una tabla; si aún está siendo referenciado por otra tabla.
- Bloqueo a nivel de filas. Al usar tablas MyISAM, y tener consultas muy grandes que requieren de mucho tiempo, simplemente no se podían ejecutar más consultas hasta que terminarían las consultas que estaban en ejecución. En cambio, las tablas InnoDB usan bloqueo a nivel de filas para mejorar de manera impresionante el rendimiento.
- SELECTs sin bloqueo. El motor InnoDB usa una técnica conocida como multi-versioning (similar a PostgreSQL) que elimina la necesidad de hacer bloqueos en consultas SELECT muy simples. Ya no será necesario molestarse porque una simple consulta de sólo lectura está siendo bloqueada por otra consulta que está haciendo cambios en una misma tabla.

²⁶ Los **BLOB** (**B**inary **L**arge **O**bjects, *objetos binarios grandes*) son elementos utilizados en las bases de datos para almacenar datos de gran tamaño que cambian de forma dinámica. No todos los Sistemas Gestores de Bases de Datos son compatibles con los BLOB.

²⁷ El tipo de dato de SQL **TEXT** guarda en la base datos binarios de longitud variable, puede contener hasta 2000000000 caracteres. No admite argumento para especificar su longitud.

²⁸ El tipo de dato de SQL **VARCHAR** define una cadena de caracteres de longitud variable en la cual determinamos el máximo de caracteres con el argumento "x" que va entre paréntesis.

²⁹ Una **clave foránea** (Foreign Key FK) es una limitación referencial entre dos tablas. La clave foránea identifica una columna o grupo de columnas en una tabla (tabla hija o referendo) que se refiere a una columna o grupo de columnas en otra tabla (tabla maestra o referenciada). Las columnas en la tabla referendo deben ser la clave primaria u otra clave candidata en la tabla referenciada.

HEAP: Tablas en memoria. Son temporales y desaparecen cuando el servidor se cierra, a diferencia de una tabla **TEMPORARY**, que solo puede ser accedida por el usuario que la crea, una tabla **HEAP** puede ser utilizada por diversos usuarios. No soportan columnas de autoincremento ni que haya valores nulos en los índices. Los datos son almacenados en pequeños bloques.

MAX: Este tipo de tablas permite la realización de transacciones (a partir de la versión 3.23.34), por lo que es posible la recuperación de datos (**COMMIT**³⁰ y **ROLLBACK**³¹). Estas tablas necesitan de una clave primaria en cada tabla, que ha de crear el administrador o de lo contrario Mysql creará una oculta. Otra de sus características es que pueden ser bloqueadas con el comando **LOCK**. Estas tablas son almacenadas en archivos “.DB”.

TST: *'Transactions safe tables'*, o tablas para transacciones seguras. Son menos rápidas y ocupan más memoria, pero a cambio ofrecen mayor seguridad frente a fallos durante la consulta. Las tablas **TST** permiten ir introduciendo consultas y finalizar con un **COMMIT** (que las ejecuta) o **ROLLBACK** (que ignora los cambios). Disponibles a partir de la versión 4 de MySQL.

Después de hacer un análisis se tomó la decisión de usar **innoDB** por las características propias de la tabla y ajustarse al sistema que se va a desarrollar.

1.10.4 Análisis del lenguaje de programación

Un lenguaje de programación es un lenguaje artificial que puede ser usado para controlar el comportamiento de una máquina, especialmente una computadora. Estos se componen de un conjunto de reglas sintácticas y semánticas que permiten expresar instrucciones que luego serán interpretadas.

En su uso, un lenguaje de programación puede acercarse a la forma humana de expresarse y, por eso, este tipo de lenguajes es llamado de alto nivel. Esto significa que utilizan palabras y formas en sus estructuras que se asemejan al lenguaje natural (especialmente al inglés). En cambio, aquellos lenguajes que se aproximan más a la forma en la cual la computadora se maneja, son llamados lenguajes de bajo nivel. Esto significa que lo que el programador deba escribir se acercará al lenguaje máquina, que es, en definitiva, lo que las computadoras pueden interpretar.

Los lenguajes para el desarrollo web están clasificados en cliente y servidor, para el lado del cliente debe usarse **HTML**, **CSS** y **Javascript** por defecto, en cambio, para el lado del servidor existen varias opciones, la decisión acerca de que lenguaje de programación de

³⁰ Una sentencia **COMMIT** en SQL finaliza una transacción de base de datos dentro de un sistema gestor de base de datos relacional (RDBMS) y pone visibles todos los cambios a otros usuarios.

³¹ **ROLLBACK** es un comando que causa que todos los cambios de datos desde la última sentencia **BEGIN WORK**, o **START TRANSACTION** sean descartados por el sistema de gestión de base de datos relacional (RDBMS), para que el estado de los datos sea "rolled back"(devuelto) a la forma en que estaba antes de que aquellos cambios tuvieran lugar.

lado del servidor se debe tomar en base a las necesidades del sistema y la infraestructura necesaria para el soporte del lenguaje elegido.

A continuación se muestran los lenguajes más usados de lado del servidor para el desarrollo web.

ASP: Active Server Pages también conocido como ASP, es una tecnología de Microsoft del tipo "lado del servidor" para páginas web generadas dinámicamente, que ha sido comercializada como un anexo a Internet Information Services (IIS³²).

Esencialmente, los ASP son una manera de adquirir externamente algunos o casi todos los aspectos de la tecnología de información que necesitan las compañías. Pueden ser empresas comerciales que abastecen a clientes, o empresas sin ánimo de lucro u organizaciones gubernamentales que proporcionan servicio y ayuda a los usuarios finales.

La mayoría de las páginas ASP son escritas en VBScript³³, pero también suele usarse JScript³⁴ (implementación de ECMAScript³⁵ por Microsoft) y PerlScript (derivado de PERL³⁶). Son utilizados especialmente para la consulta en bases de datos, y generación dinámica de páginas. Tecnologías como InstantASP y ChiliASP permiten ejecutar ASP fuera de un sistema operativo Windows.

JSP: JavaServer Pages (JSP) es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML³⁷ o de otro tipo.

Esta tecnología es un desarrollo de la compañía Sun MycroSystems. La Especificación JSP 1.2 fue la primera que se liberó y en la actualidad está disponible la Especificación JSP 3.

Las JSP's permiten la utilización de código Java mediante scripts. Además, es posible utilizar algunas acciones JSP predefinidas mediante etiquetas. Estas etiquetas pueden ser enriquecidas mediante la utilización de Bibliotecas de Etiquetas (TagLibs o Tag Libraries) externas e incluso personalizadas.

PHP: es un lenguaje de programación de uso general de script del lado del servidor originalmente diseñado para el desarrollo web³⁸ de contenido dinámico. Fue uno de los

³² **Internet Information Services** o **IIS** 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.

³³ Véase en el glosario.

³⁴ Véase en el glosario.

³⁵ Véase en el glosario.

³⁶ Véase en el glosario.

³⁷ Véase en el glosario.

³⁸ **Desarrollo web** es un título algo arbitrario para el conjunto de tecnologías de software del lado del servidor y del cliente que involucran una combinación de procesos de base de datos con el uso de un navegador en Internet a fin de realizar determinadas tareas o mostrar información.

primeros lenguajes de programación del lado del servidor que se podrán incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante.

PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. PHP puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo.

PHP fue creado originalmente por Rasmus Lerdorf en 1995. Actualmente el lenguaje sigue siendo desarrollado con nuevas funciones por el grupo PHP. Este lenguaje forma parte del software libre publicado bajo la licencia PHP que es incompatible con la licencia pública general de GNU debido a las restricciones del uso del término *PHP*.

Python: Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis muy limpia y que favorezca un código legible.

Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa³⁹ y, en menor medida, programación funcional⁴⁰. Es un lenguaje interpretado⁴¹ usa tipado dinámico (una misma variable puede tomar valores de distinto tipo en distintos momentos), es fuerte tipado (una variable de un tipo concreto, no se puede usar como si fuera de otro tipo distinto a menos que se haga una conversión) y multiplataforma.

Es administrado por la Python Software Foundation. Posee una licencia de código abierto, denominada Python Software Foundation License⁴², que es compatible con la licencia pública general de GNU a partir de la versión 2.1.1, e incompatible en ciertas versiones anteriores.

³⁹ La **programación imperativa**, en contraposición a la programación declarativa es un paradigma de programación que describe la programación en términos del estado del programa y sentencias que cambian dicho estado. Los programas imperativos son un conjunto de instrucciones que le indican al computador cómo realizar una tarea.

⁴⁰ La **programación funcional** es un paradigma de programación declarativa basado en la utilización de funciones aritméticas que no maneja datos mutables o de estado. Enfatiza la aplicación de funciones, en contraste con el estilo de programación imperativa, que enfatiza los cambios de estado.

⁴¹ Un **lenguaje interpretado** es un lenguaje de programación que está diseñado para ser ejecutado por medio de un intérprete, en contraste con los lenguajes compilados. Teóricamente, cualquier lenguaje puede ser compilado o ser interpretado, así que esta designación es aplicada puramente debido a la práctica de implementación común y no a alguna característica subyacente de un lenguaje en particular.

⁴² La **Python Software Foundation License**, o **PSFL** por sus siglas en inglés, es una licencia de software libre permisiva, al estilo de la licencia BSD, es decir, que cumple con los requisitos OSI para ser declarada licencia de software libre; además, es compatible con la licencia GPL. A diferencia de la licencia GPL, y como la mayoría de licencias tipo BSD, la licencia PSFL no es una licencia *copyleft*, y permite modificaciones del código fuente, así como la creación de trabajos derivados, sin requerir que ni las modificaciones ni los trabajos derivados tengan que ser a su vez de código abierto.

Ruby: Ruby es un lenguaje de programación interpretado y orientado a objetos, creado por el programador japonés Yukihiro “Matz” Matsumoto quien comenzó a trabajar en Ruby en 1993, y lo presentó públicamente en 1995. Combina una sintaxis inspirada en Python y Perl con características de programación orientada a objetos similares a Smalltalk⁴³.

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

1.10.5 Selección de lenguaje de programación

Después de un análisis realizado sobre las ventajas y desventajas de los lenguajes de programación, se llegó a la conclusión basada en la tabla 1-2, la cual fue usar PHP como el lenguaje para desarrollar el proyecto.

	ASP	PHP	Phyton	JSP
Consumo de recursos	Alto	Bajo	Bajo	Alto
Portabilidad hacia otros servidores	No	Si	No	Si
Protección de memoria	No	Si	Si	Si
Facilidad de aprendizaje	Alto	Bajo	Bajo	Alto
Costo	Licencia	Gratis	Gratis	Gratis

Tabla 1-2 Comparativa de lenguajes de programación del lado del servidor

Se eligió **PHP** porque funciona sobre prácticamente todas las plataformas imaginables y garantiza una alta velocidad de ejecución, además de una excelente estabilidad. Su seguridad se ve reforzada por el hecho de que el código original permanece oculto al usuario: el navegador lo ejecuta y lo muestra “traducido” a HTML⁴⁴. A diferencia de otros lenguajes (como ASP, de Microsoft) es gratuito y open source. Y gracias a una enorme comunidad de desarrolladores a nivel mundial, cuenta con una amplia documentación.

1.10.6 Análisis de Framework del lado del servidor y de lado del cliente

Para el desarrollo del sistema necesitaremos además la ayuda de frameworks del lado del cliente y de lado del servidor, esto reducirá el tiempo de desarrollo del sistema y también contaremos con una base sólida de programación y una amplia documentación disponible.

⁴³ **Smalltalk** es mucho más que un lenguaje de programación. Permite realizar tareas de computación mediante la interacción con un entorno de objetos virtuales. Metafóricamente, se puede considerar que un Smalltalk es un mundo virtual donde viven objetos que se comunican entre sí, mediante el envío de mensajes.

⁴⁴ Véase el glosario.

1.10.6.1 Framework PHP de lado del servidor

CakePHP: es un framework para el desarrollo aplicaciones web escrito en PHP, creado sobre los conceptos de Ruby on Rails⁴⁵.

CakePHP es un framework o marco de trabajo que facilita el desarrollo de aplicaciones web, utilizando el patrón de diseño MVC⁴⁶ (Modelo Vista Controlador). Es de código abierto y se distribuye bajo licencia MIT.

Al igual que Ruby On Rails, CakePHP facilita al usuario la interacción con la base de datos mediante el uso de Active Record. Además hace uso del patrón Modelo Vista Controlador.

- Compatible con PHP4 y PHP5
- CRUD de la base de datos integrado
- URLs amigables
- Sistema de plantillas rápido y flexible
- Ayudas para AJAX⁴⁷, Javascript, HTML, forms y más
- Trabaja en cualquier subdirectorio del sitio
- Validación integrada
- Scaffolding⁴⁸ de las aplicaciones
- Access Control List⁴⁹
- Componentes de seguridad y sesión

Zend: Zend Framework (ZF) es un framework de código abierto para desarrollar aplicaciones web y servicios web con PHP 5. ZF es una implementación que usa código 100% orientado a objetos. La estructura de los componentes de ZF es algo único; cada componente está construido con una baja dependencia de otros componentes. Esta arquitectura débilmente acoplada permite a los desarrolladores utilizar los componentes por separado. A menudo se refiere a este tipo de diseño como "use-at-will" (uso a voluntad).

Aunque se pueden utilizar de forma individual, los componentes de la biblioteca estándar de Zend Framework conforman un potente y extensible *framework* de aplicaciones web al combinarse. ZF ofrece un gran rendimiento y una robusta implementación MVC,

⁴⁵ **Ruby on Rails**, también conocido como **RoR** o **Rails**, es un framework de aplicaciones web de código abierto escrito en el lenguaje de programación Ruby, siguiendo el paradigma de la arquitectura Modelo Vista Controlador (MVC). Trata de combinar la simplicidad con la posibilidad de desarrollar aplicaciones del mundo real escribiendo menos código que con otros frameworks y con un mínimo de configuración

⁴⁶ Véase en el glosario.

⁴⁷ Véase el glosario.

⁴⁸ El **scaffolding** es un método para contruir aplicaciones basadas en bases de datos, esta técnica está soportada por algunos frameworks del tipo MVC en el cuál el programador escribe una especificación que describe cómo debe ser usada la base de datos. Luego el compilador utiliza esa especificación para generar el código que la aplicación usará para crear, leer, actualizar y eliminar registros de la base de datos, esto es conocido como CRUD (create, read, update, delete).

⁴⁹ Una **lista de control de acceso** o **ACL** (del inglés, *access control list*) es un concepto de seguridad informática usado para fomentar la separación de privilegios. Es una forma de determinar los permisos de acceso apropiados a un determinado objeto, dependiendo de ciertos aspectos del proceso que hace el pedido.

una abstracción de base de datos fácil de usar, y un componente de formularios que implementa la prestación de formularios HTML, validación y filtrado para que los desarrolladores puedan consolidar todas las operaciones usando de una manera sencilla la interfaz orientada a objetos. También existen componentes que implementan bibliotecas de cliente para acceder de forma sencilla a los web services más populares. Cualesquiera que sean las necesidades de su solicitud, usted tiene todas las posibilidades de encontrar un componente de Zend Framework que se pueda utilizar para reducir drásticamente el tiempo de desarrollo, con una base completamente sólida.

El principal patrocinador del proyecto Zend Framework es Zend Technologies, pero muchas empresas han contribuido con componentes o características importantes para el marco. Empresas como Google, Microsoft y StrikeIron se han asociado con Zend para proporcionar interfaces de servicios web y otras tecnologías que desean poner a disposición de los desarrolladores de Zend Framework.

Codeigniter: es un framework para aplicaciones web de código abierto para crear sitios web dinámicos con PHP. Su objetivo es permitir que los desarrolladores puedan realizar proyectos mucho más rápido, pone a disposición del desarrollador una estructura para iniciar un proyecto, brindando un conjunto de bibliotecas para tareas comunes, así como una interfaz simple y una estructura lógica para acceder esas bibliotecas.

También hay que destacar que Codeigniter es más rápido que muchos otros entornos. Incluso en una discusión sobre entornos de desarrollo con PHP, Rasmus Lerdorf, el creador de PHP, expresó que le gustaba Codeigniter “porque es rápido, ligero y parece poco un entorno”.

Como cualquier otro framework, Codeigniter contiene una serie de librerías que sirven para el desarrollo de aplicaciones web y además propone una manera de desarrollarlas que debemos seguir para obtener provecho de la aplicación. Esto es, marca una manera específica de codificar las páginas web y clasificar sus diferentes scripts, que sirve para que el código esté organizado y sea más fácil de crear y mantener. Codeigniter implementa MVC, que es un estándar de programación de aplicaciones, utilizado tanto para hacer sitios web como programas tradicionales. Este sistema tiene sus características, que veremos en el siguiente capítulo.

Algunos de los puntos más interesantes sobre este framework, sobre todo en comparación con otros productos similares, son los siguientes:

Versatilidad: Quizás la característica principal de Codeigniter, en comparación con otros frameworks PHP. Codeigniter es capaz de trabajar la mayoría de los entornos o servidores, incluso en sistemas de alojamiento compartido, donde sólo tenemos un acceso por FTP para enviar los archivos al servidor y donde no tenemos acceso a su configuración.

Actualizado: Desde la versión 2 de Codeigniter ya solo es compatible con la versión 5 de PHP. Para los que todavía usen PHP 4 pueden descargar una versión antigua del framework, como Codeigniter V 1.7.3, que todavía era compatible. Estas versiones están en la página de descargas de Codeigniter.

Facilidad de instalación: No es necesario más que una cuenta de FTP para subir Codeigniter al servidor y su configuración se realiza con apenas la edición de un archivo, donde debemos escribir cosas como el acceso a la base de datos. Durante la configuración no necesitaremos acceso a herramientas como la línea de comandos, que no suelen estar disponibles en todos los alojamientos.

Flexibilidad: Codeigniter es bastante menos rígido que otros frameworks. Define una manera de trabajar específica, pero en muchos de los casos podemos seguirla o no y sus reglas de codificación muchas veces nos las podemos saltar para trabajar como más a gusto encontremos. Algunos módulos como el uso de plantillas son totalmente opcionales. Esto ayuda muchas veces también a que la curva de aprendizaje sea más sencilla al principio.

Ligereza: El núcleo de Codeigniter es bastante ligero, lo que permite que el servidor no se sobrecargue interpretando o ejecutando grandes porciones de código. La mayoría de los módulos o clases que ofrece se pueden cargar de manera opcional, sólo cuando se van a utilizar realmente.

Documentación tutorializada: La documentación de Codeigniter es fácil de seguir y de asimilar, porque está escrita en modo de tutorial. Esto no facilita mucho la referencia rápida, cuando ya sabemos acerca del framework y queremos consultar sobre una función o un método en concreto, pero para iniciarnos sin duda se agradece mucho. Cuenta con foros donde la comunidad puede consultar más información relacionada al framework.

Yii: Yii es un framework orientado a objetos, software libre, de alto rendimiento basado en componentes, PHP y framework de aplicaciones web. Yii se pronuncia en español como se escribe y es un acrónimo para "Yes It Is!" (en español: ¡Sí lo es!).

Algunas características de Yii incluyen:

- Patrón de diseño Modelo Vista Controlador (MVC).
- Database Access Object(DAO), query builder, Active Record y migración de base de datos.
- Integración con JQuery.
- Entradas de Formulario Y validación.
- Widgets⁵⁰ de AJAX, como autocompletado de campos de texto y demás.
- Soporte de Autenticación incorporado. Además soporta autorización vía role-based access control (RBAC) jerárquico.
- Personalización de aspectos y temas.
- Generación compleja automática de WSDL⁵¹, especificaciones y administración de peticiones Web Service.

⁵⁰ Un *widget* es una pequeña aplicación o programa, usualmente presentado en archivos o ficheros pequeños que son ejecutados por un motor de *widgets* o *Widget Engine*. Entre sus objetivos están dar fácil acceso a funciones frecuentemente usadas y proveer de información visual.

⁵¹ **WSDL** son las siglas de *Web Services Description Language*, un formato XML que se utiliza para describir servicios Web . La versión 1.0 fue la primera recomendación por parte del W3C y la versión 1.1 no alcanzó nunca tal estatus. La versión 2.0 se convirtió en la recomendación actual por parte de dicha entidad.

- Internacionalización y locación (I18N and L10N). Soporta traducciones, formato de fecha y hora, formato de números, y localización de la vista.
- El manejo de errores y log⁵². Los errores son manejados y personalizados, y los log de mensajes pueden ser categorizados, filtrados y movidos a diferentes destinos.
- Generación automática de código para el esqueleto de la aplicación, aplicaciones CRUD, etc.
- Generación de código por componentes de Yii y la herramienta por línea de comandos, cumple con los estándares de XHTML.

1.10.6.2 Selección de frameworks de lado del servidor

Para la realización de este proyecto se optó por el uso de Codeigniter, este framework es de un fácil manejo, se apega mucho al lenguaje de programación PHP, cosa que los otros frameworks no hacen tan bien, esto hace que la programación sea más sencilla y no se pierda demasiado tiempo aprendiendo a manejar el framework, las características señaladas también simplificarán la actualización del sistema, incluso cuando se trate de otros usuarios los que realicen los cambios, la decisión fue tomada después de analizar la tabla 1-3 donde se mencionan las características que se tomaron en cuenta de los frameworks.

	CakePHP	Codeigniter	Yii	Zend
Soporta PHP 4	Si	Si	No	No
Soporta PHP 5	Si	Si	Si	Si
Soporta MVC	Si	Si	Si	Si
Módulo de validación	Si	Si	Si	Si
Motor de plantillas	No	Si	Si	Si
Facilidad de aprendizaje	Medio	Bajo	Alto	Alto
Footsprint (líneas de código)	277,751	45,971	211,572	2,241,123

Tabla 1-3 Comparativa de frameworks PHP

1.10.6.3 Framework de lado del cliente

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

jQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la GPL, permitiendo su uso en proyectos libres y privados, jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

⁵² Un **log** es un registro oficial de eventos durante un rango de tiempo en particular. Para los profesionales en seguridad informática es usado para registrar datos o información sobre quién, qué, cuándo, dónde y por qué (who, what, when, where y why) un evento ocurre para un dispositivo en particular o aplicación.

Sus características son las siguientes:

- Selección de elementos DOM.
- Interactividad y modificaciones del árbol DOM, incluyendo soporte para CSS⁵³ 1-3 y un plugin básico de XPath⁵⁴.
- Eventos.
- Manipulación de la hoja de estilos CSS.
- Efectos y animaciones.
- Animaciones personalizadas.
- AJAX.
- Soporta extensiones.
- Utilidades varias como obtener información del navegador, operar con objetos y vectores, funciones para rutinas comunes, etc.
- Compatible con los navegadores Mozilla Firefox 2.0+, Internet Explorer 6+, Safari 3+, Opera 10.6+ y Google Chrome 8+.

jQuery UI es una biblioteca de componentes para el framework jQuery que le añaden un conjunto de plug-ins, widgets y efectos visuales para la creación de aplicaciones web. Cada componente o módulo se desarrolla de acuerdo a la filosofía de jQuery (*find something, manipulate it*: encuentra algo, manipúlalo).

Interacciones: Añade comportamientos complejos a los elementos:

- Draggable: Hace al elemento arrastrable.
- Droppable: Permite que el elemento responda a elementos arrastrables.
- Resizable: Permite redimensionar el elemento.
- Selectable: Permite seleccionar entre una lista de elementos.
- Sortable: Ordena una lista de elementos.

Widgets: Es un conjunto completo de controles UI. Cada control tiene un conjunto de opciones configurables y se les pueden aplicar estilos CSS.

- Accordion: Menú con efecto acordeón.
- Autocomplete: Caja con autocompletado.
- Button: Botón.
- Dialog: Ventanas con contenido.
- Slider: Elemento para elegir en un rango de valores.
- Tabs: Pestañas.
- Datepicker: Calendario gráfico.
- Progressbar: Barra de progreso.

⁵³ Véase en el glosario.

⁵⁴ **XPath** (*XML Path Language*) es un lenguaje que permite construir expresiones que recorren y procesan un documento XML. XPath permite buscar y seleccionar teniendo en cuenta la estructura jerárquica del XML.

Yui: Yahoo User Interface (YUI), una serie de bibliotecas escritas en JavaScript, para la construcción de aplicaciones interactivas (RIA⁵⁵). Liberadas bajo licencia BSD por parte de la compañía Yahoo. Dichas bibliotecas son utilizadas para el desarrollo web específicamente para ser usadas como la programación de aplicaciones de escritorio, con componentes vistosos y personalizables y con una amplia implementación con AJAX.

La biblioteca está completamente documentada en su página web y se compone de seis componentes: Núcleo YUI, utilidades, controles UI, componentes CSS, herramientas de desarrollo y de construcción.

Utilidades:

- Animation: Ayuda a crear efectos animados.
- Browser History Manager: Ayuda complementaria para el botón Atrás y la gestión de Marcadores/Favoritos de los navegadores web.
- Connection Manager: Ayuda para manejar el objeto XMLHttpRequest⁵⁶.
- Cookie: Permite gestionar las cookies⁵⁷.
- DataSource: Proporciona una interfaz común para que otros componentes pueden interactuar con diferentes tipos de datos.
- Drag and Drop: Facilita la creación de eventos y elementos que pueden ser arrastrados.
- Element: Proporciona una capa para los elementos que facilita el añadido de escuchadores, manipulación del árbol DOM y atributos 'get' y 'set'.
- Get: La utilidad Get soporta la carga asíncrona de datos y archivos de estilos CSS externos.
- ImageLoader: Permite aplazar la carga de imágenes que no son visibles durante la carga de la página, proporcionando un aumento del rendimiento.
- JSON: Proporciona métodos para el tratamiento de dato JSON.
- Resize: Permite redimensionar los elementos HTML.
- Selector: Permite referenciar elementos HTML mediante la sintaxis CSS3.
- YUI Loader: Es un cargador del lado cliente que permite la carga de forma dinámica de cualquier componente y dependencia de la biblioteca al vuelo.

Controles

- AutoComplete: Proporciona la función de auto completado (lista de sugeridos) para los usuarios. Soporta varios formatos de datos, tanto del lado cliente como del lado servidor (vía XMLHttpRequest).
- Button: Permite la creación de botones gráficos que funcionan como un botón tradicional en HTML.
- Calendar: Un calendario gráfico y de control dinámico.

⁵⁵ Las *rich Internet applications*, o *RIA* (en español "aplicaciones de Internet enriquecidas"), son aplicaciones web que tienen la mayoría de las características de las aplicaciones de escritorio tradicionales. Estas aplicaciones utilizan un navegador web estandarizado para ejecutarse y por medio de complementos o mediante una máquina virtual se agregan las características adicionales.

⁵⁶ Véase en el glosario.

⁵⁷ Una *cookie* (o **galleta informática**) es una pequeña información enviada por un sitio web y almacenada en el navegador del usuario, de manera que el sitio web puede consultar la actividad previa del usuario.

- Charts: Permite la creación de diferentes tipos de gráficos (lineales, de barras, etc.).
- Color Picker: Proporciona una interfaz gráfica para la selección de colores.
- Container: Proporciona una interfaz gráfica como Tooltip, paneles, cuadros de diálogo, etc.
- DataTable: Una potente herramienta para mostrar tablas tabulares en una página web. Permite la ordenación de columnas tanto en el lado cliente como en el servidor, paginación, scroll, selección de filas, redimensionado de columnas y edición inline.

Dojo: es un framework que contiene APIs y widgets (controles) para facilitar el desarrollo de aplicaciones Web que utilicen tecnología AJAX. Contiene un sistema de empaquetado inteligente, los efectos de UI, drag and drop APIs, widget APIs, abstracción de eventos, almacenamiento de APIs en el cliente, e interacción de APIs con AJAX.

Resuelve asuntos de usabilidad comunes como pueden ser la navegación y detección del navegador, soportar cambios de URL en la barra de URLs para luego regresar a ellas (bookmarking), y la habilidad de degradar cuando AJAX/JavaScript no es completamente soportado en el cliente. Es conocido como "la navaja suiza del ejército de las bibliotecas Javascript". Proporciona una gama más amplia de opciones en una sola biblioteca JavaScript y es compatible con navegadores antiguos.

Los complementos de Dojo son componentes preempaquetados de código JavaScript, HTML y CSS que pueden ser usados para enriquecer aplicaciones web.

- Menús, pestañas y tooltips⁵⁸.
- Tablas ordenables, gráficos dinámicos y dibujado de vectores 2D.
- Efectos de animación y la posibilidad de crear animaciones personalizables.
- Soporte para arrastrar y soltar.
- Formularios y rutinas de validación para los parámetros.
- Calendario, selector de tiempo y reloj.
- Editor online de texto enriquecido.
- Núcleo de componentes (dijit) accesible desde versiones anteriores y lector de pantalla.

Dojo provee de un sistema de paquetes para facilitar el desarrollo modular. El script de inicio inicializa una serie de jerarquías de paquetes de espacios de nombre (*io*, *event*, etc.) bajo el paquete raíz *dojo*. Después de la inicialización del paquete *dojo*, cualquier otro paquete puede ser cargado (vía XMLHttpRequest o cualquier otro transporte similar) usando las utilidades ofrecidas en el arranque. También es posible inicializar paquetes adicionales dentro o al mismo nivel que el paquete *dojo*, permitiendo extensiones o bibliotecas de terceros.

Los paquetes de *Dojo* pueden contener múltiples archivos. Cualquier paquete o archivo puede depender de otro. En este caso, cuando el paquete es cargado, cualquier dependencia será también cargada.

⁵⁸ Un *tooltip* (también llamada *descripción emergente*) es una herramienta de ayuda visual, que funciona al situar el cursor sobre algún elemento gráfico, mostrando una ayuda adicional para informar al usuario de la finalidad del elemento sobre el que se encuentra.

Dojo también brinda una manera de crear perfiles; el sistema ofrece una lista de paquetes y usa Apache Ant⁵⁹ para crear un archivo JavaScript comprimido que contiene dichos paquetes y dependencias. De esta manera se tiene todo el código necesario para ser cargado y es inicializado de una sola vez, permitiendo así el cacheado (la mayoría de los navegadores web no permiten el cacheado de archivos vía AJAX).

1.10.6.4 Selección de frameworks de lado del cliente

En cuanto al lado del cliente se usará JQuery, la elección se hizo en base a su sencillez y fácil uso, además de tomar en cuenta la cantidad de plugins gratuitos que pueden simplificar aún más el desarrollo del sistema, la comparación entre los frameworks se muestra en la tabla 1-4.

	Jquery	YUI	Dojo
Soporta Internet explorer	6 y posteriores	6 y posteriores	6 y posteriores
Soporta Mozilla Firefox	2 y posteriores	3 y posteriores	3 y posteriores
Soporta Google Chrome	1 y posteriores	2 y posteriores	3 y posteriores
Manejo de datos XMLHttpRequest	Si	Si	Si
Recuperación de datos vía JSON	Si	Si	Si
Otras formas de recuperación de datos	XML, HTML		HTML, XML, CSV, ATOM
Herramientas de creación de HTML	Si	No	Si
Manejo de eventos	Si	Si	Si
Facilidad de aprendizaje	Fácil	Medio	Medio

Tabla 1-4 Comparativa de frameworks de lado del cliente

⁵⁹ **Apache Ant** es una herramienta usada en programación para la realización de tareas mecánicas y repetitivas, normalmente durante la fase de compilación y construcción (build). Es, por tanto, un software para procesos de automatización de compilación, similar a Make pero desarrollado en lenguaje Java y requiere la plataforma Java, así que es más apropiado para la construcción de proyectos Java.

1.2 1.11 Solución propuesta

El primer paso que deberá realizar el académico para poder tener acceso al sistema es el de registro, a continuación se muestra la propuesta del proceso en la figura 1-3.

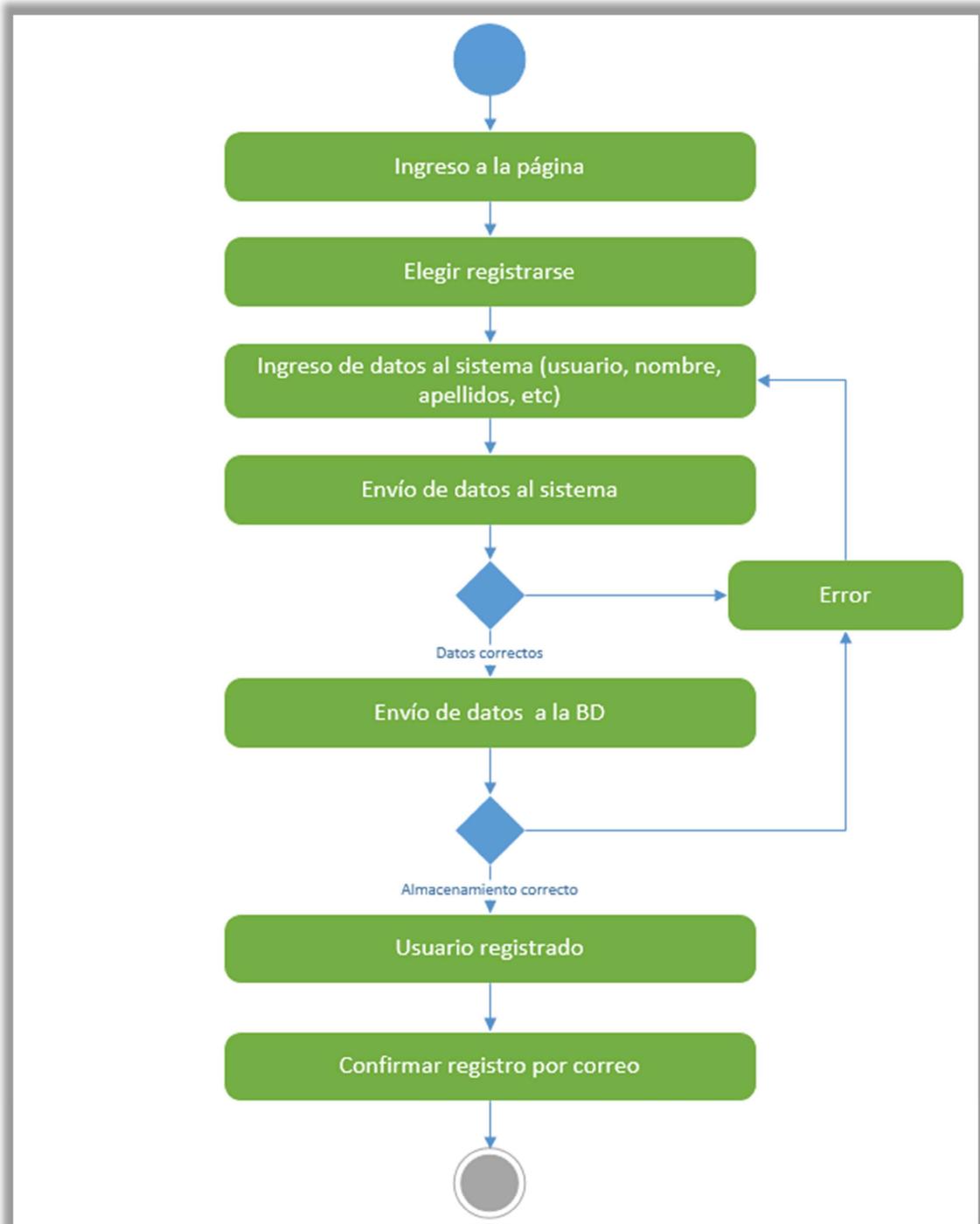


Figura 1-3 Acceder al sistema

El proceso del sistema siguiente es el inicio de sesión, este proceso es mostrado en la figura 1-4.

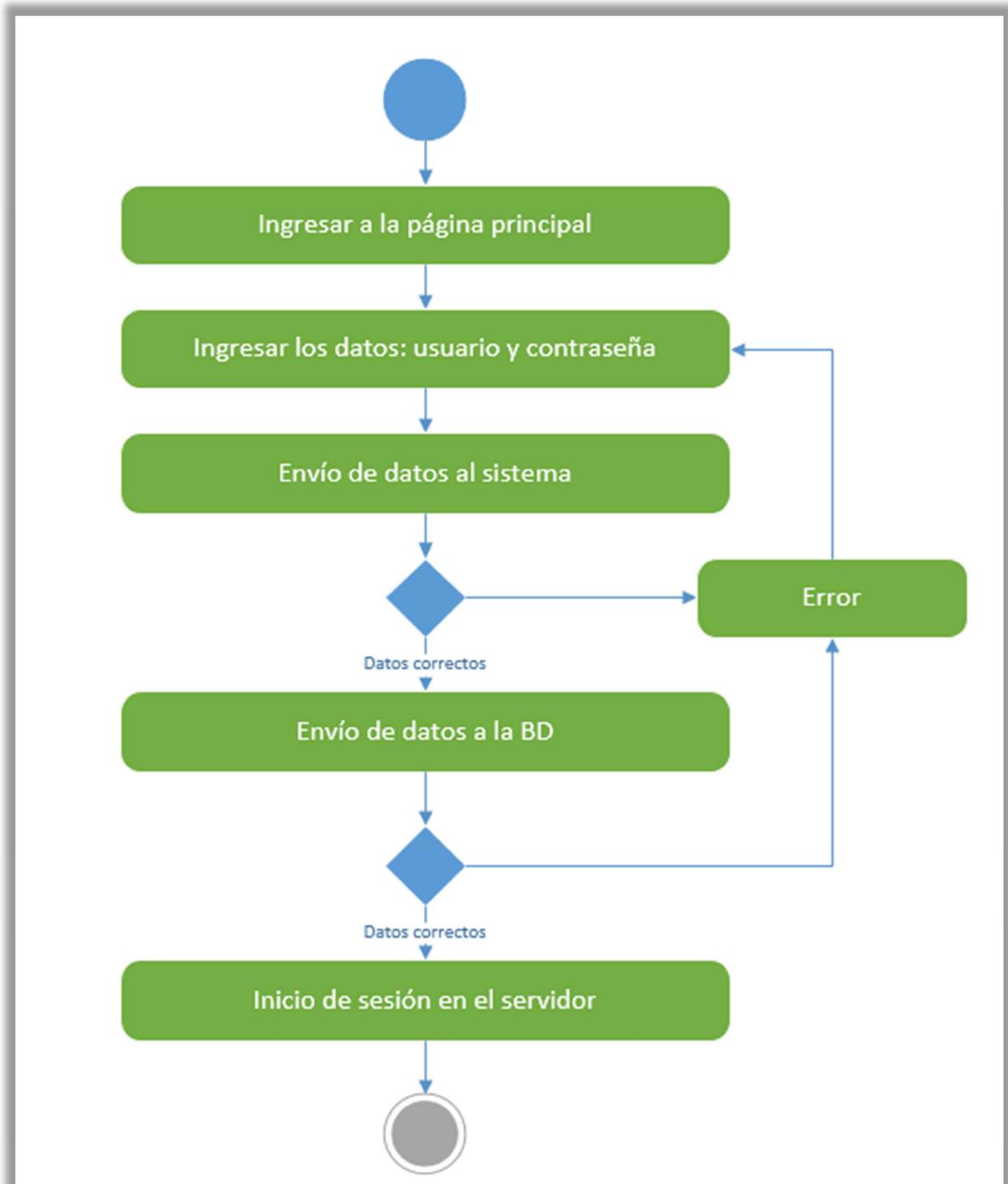


Figura 1-4 Inicio de sesión

El proceso mostrado en la figura 1-5 es la validación de seguridad, esto se hace para mostrar la información solamente a los usuarios autorizados por el sistema.

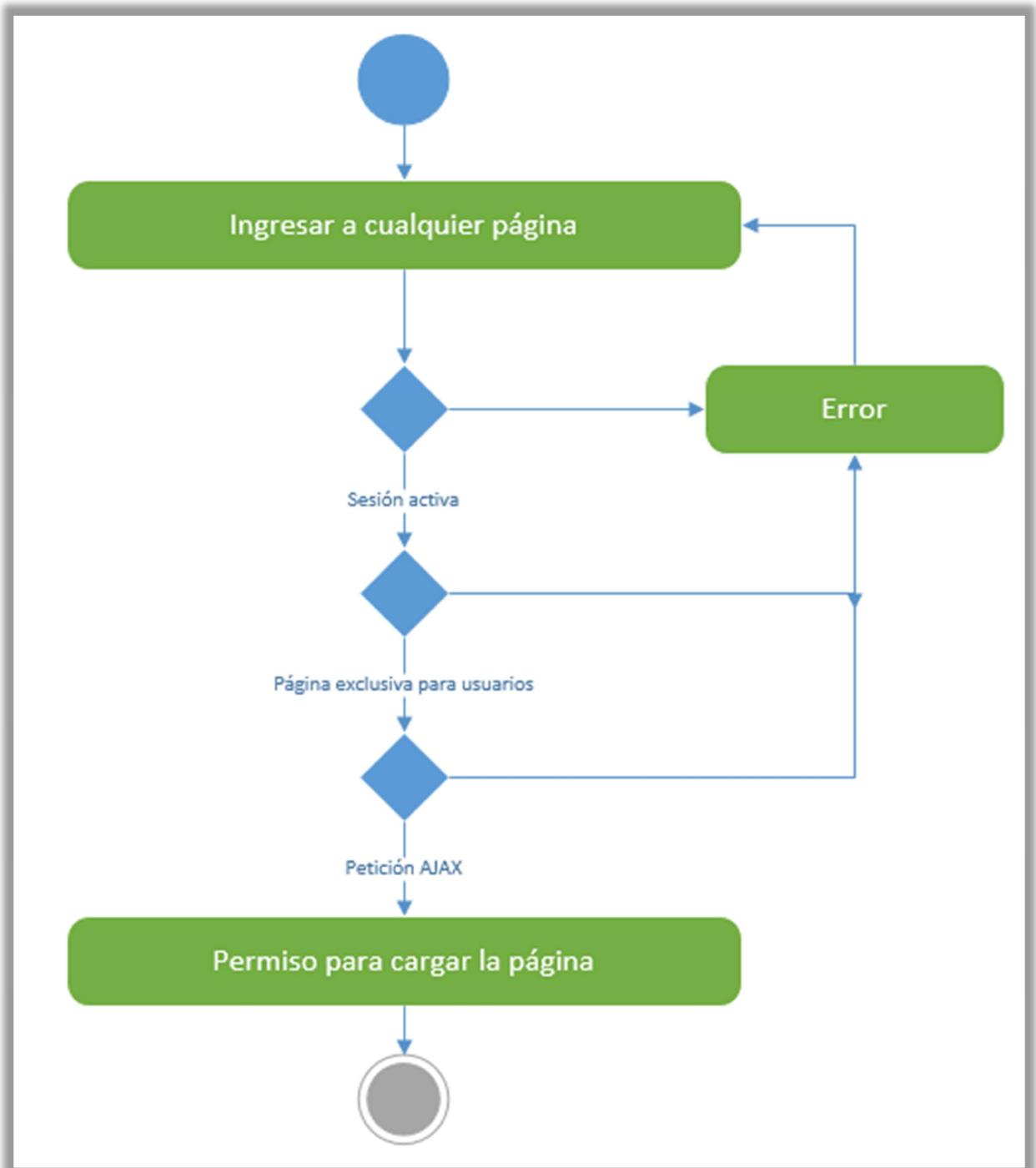


Figura 1-5 Validación de seguridad

La propuesta de la figura 1-6 pertenece al proceso para crear un informe, este proceso incluye verificación de un informe incompleto o creación de uno totalmente nuevo.

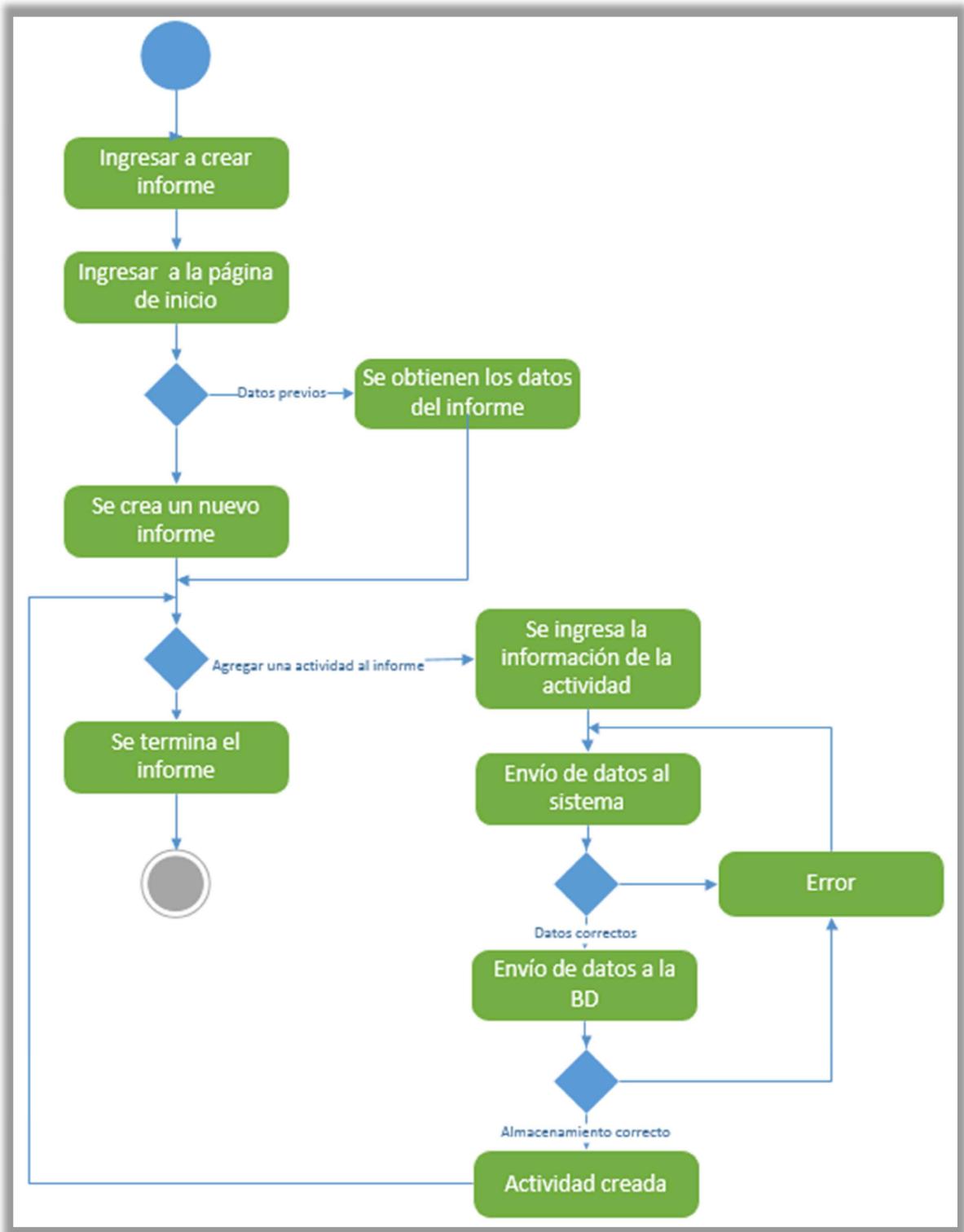


Figura 1-6 Creación de un informe

La figura 1-7 ilustra el proceso de edición de un informe, este proceso incluye el proceso para cargar la información previamente creada con relación al informe.

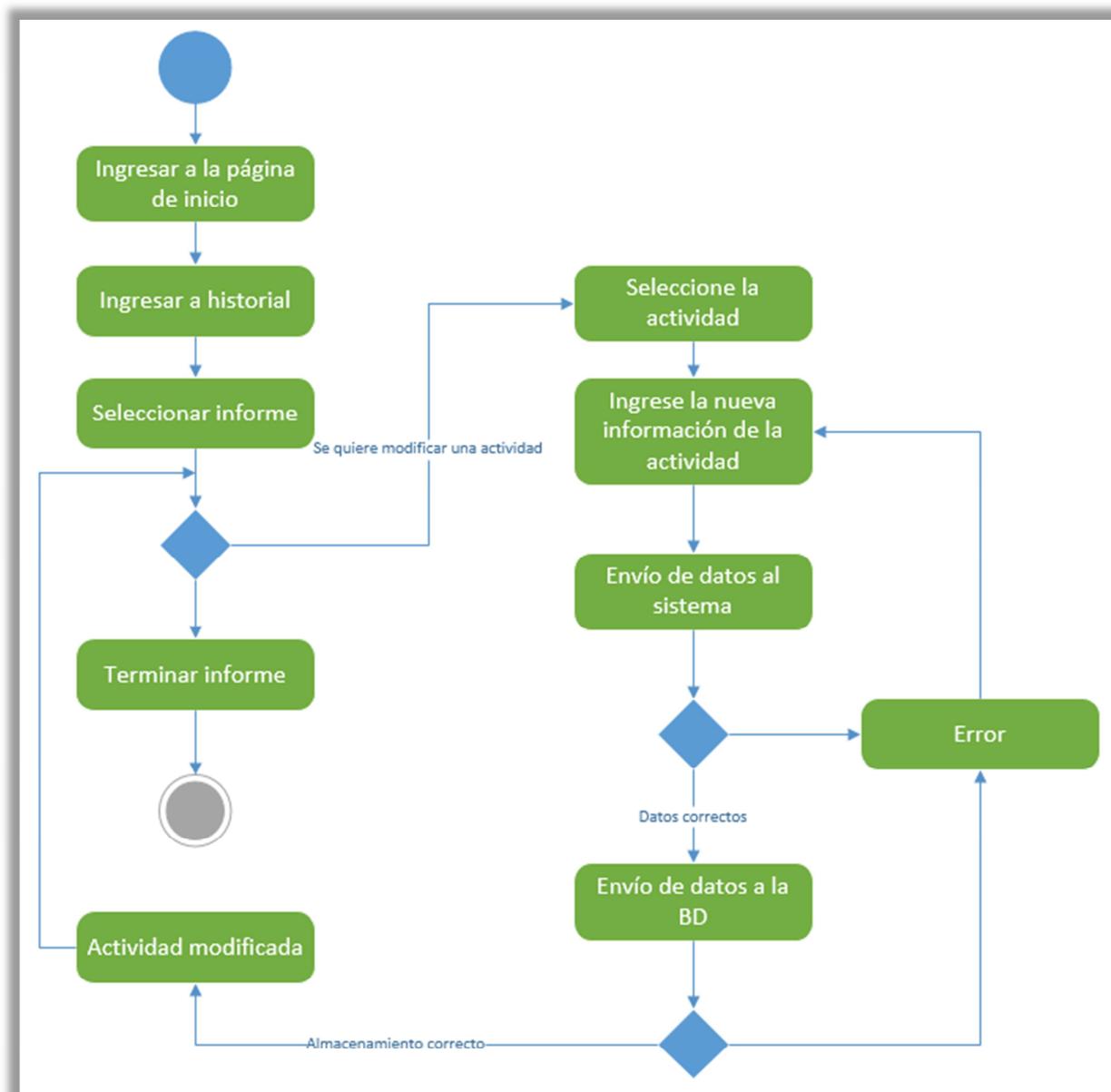


Figura 1-7 Editar un informe

1.3 1.12 Glosario

ACID: características necesarias para que una serie de instrucciones puedan ser consideradas como una transacción. Así pues, si un sistema de gestión de bases de datos es *ACID compliant* quiere decir que el mismo cuenta con las funcionalidades necesarias para que sus transacciones tengan las características ACID. En concreto ACID es un acrónimo de Atomicity, Consistency, Isolation and Durability: Atomicidad, Consistencia, Aislamiento y Durabilidad en español.

Active Record: patrón de acceso a datos que permite obtener, insertar y actualizar información en la base de datos con mínima codificación. En algunos casos, sólo una o dos líneas de código son necesarias para realizar una acción de base de datos.

AGPL: la licencia pública general de Affero (en inglés, Affero General Public License, también Affero GPL o AGPL) es una licencia copyleft derivada de la Licencia Pública General de GNU diseñada específicamente para asegurar la cooperación con la comunidad en el caso de software que corra en servidores de red.

AJAX: es un acrónimo de *Asynchronous JavaScript + XML*, que se puede traducir como "JavaScript asíncrono + XML". Ajax no es una tecnología en sí mismo. En realidad, se trata de varias tecnologías independientes que se unen de formas nuevas y sorprendentes.

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.

API: Interfaz de programación de aplicaciones (IPA) o API (del inglés *Application Programming Interface*) es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

BSD: La licencia BSD es la licencia de software otorgada principalmente para los sistemas BSD (*Berkeley Software Distribution*). Es una licencia de software libre permisiva como la licencia de OpenSSL o la MIT License. Esta licencia tiene menos restricciones en comparación con otras como la GPL estando muy cercana al dominio público. La licencia BSD al contrario que la GPL permite el uso del código fuente en software no libre.

CRUD: es el acrónimo de Crear, Obtener, Actualizar y Borrar (del original en inglés: Create, Read, Update and Delete). Es usado para referirse a las funciones básicas en bases de datos o la capa de persistencia en un software.

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

DAO: Objeto de Acceso a Datos es un componente de software que suministra una interfaz común entre la aplicación y uno o más dispositivos de almacenamiento de datos, tales como una Base de datos o un archivo. El término se aplica frecuentemente al Patrón de diseño Object.

DOM: El Modelo de Objetos del Documento (DOM) es un API para documentos HTML y XML. Proporciona una representación estructural del documento, permitiendo la modificación de su contenido o su presentación visual. Esencialmente, comunica las páginas web con los scripts o los lenguajes de programación.

ECMAScript: es una especificación de lenguaje de programación publicada por ECMA International. El desarrollo empezó en 1996 y estuvo basado en el popular lenguaje JavaScript propuesto como estándar por Netscape Communications Corporation.

Framework: Un framework o infraestructura digital, es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de *software* concretos, que puede servir de base para la organización y desarrollo de *software*. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.

GPL: Licencia Pública General de GNU o más conocida por su nombre en inglés *GNU General Public License* o simplemente sus siglas del inglés GNU GPL, es una licencia creada por la Free Software Foundation en 1989 (la primera versión, escrita por Richard Stallman), y está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

HTML: *HyperText Markup Language* es un lenguaje muy sencillo que permite describir hipertexto, es decir, texto presentado de forma estructurada y agradable, con *enlaces (hyperlinks)* que conducen a otros documentos o fuentes de información relacionadas, y con *inserciones multimedia* (gráficos, sonido...).

IIS: 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.

Java: es un lenguaje de programación de alto nivel orientado a objetos, desarrollado por James Gosling en 1995. 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. La memoria es gestionada mediante un recolector de basura.

Javascript: es un lenguaje orientado a objetos ligero, más conocido como el lenguaje de script para páginas web, pero también usado en muchos entornos sin navegador

JDBC: es una API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java, independientemente del sistema operativo donde se ejecute o de la base de datos a la cual se accede, utilizando el dialecto SQL del modelo de base de datos que se utilice

JSON: (JavaScript Object Notation - Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo. Está basado en un subconjunto del Lenguaje de Programación JavaScript, Standard ECMA-262 3rd Edition - Diciembre 1999.

Linux: GNU/Linux es uno de los términos empleados para referirse a la combinación del núcleo o *kernel* libre similar a Unix denominado Linux con el sistema operativo GNU. Su desarrollo es uno de los ejemplos más prominentes de software libre; todo su código fuente puede ser utilizado, modificado y redistribuido libremente por cualquiera bajo los términos de la GPL (Licencia Pública General de GNU, *en inglés: General Public License*) y otra serie de licencias libres.

MVC: El Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario. Este patrón de diseño se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

Protocolo: Un protocolo de internet es un protocolo de comunicación de datos digitales clasificado funcionalmente en la Capa de Red según el modelo internacional OSI. Su función principal es el uso bidireccional en origen o destino de comunicación para transmitir datos mediante un protocolo no orientado a conexión que transfiere paquetes conmutados a través de distintas redes físicas previamente enlazadas según la norma OSI de enlace de datos.

P2P: Una red *peer-to-peer*, red de pares, red entre iguales, red entre pares o red punto a punto (*P2P*, por sus siglas en inglés) es una red de computadoras en la que todos o algunos aspectos funcionan sin clientes ni servidores fijos, sino una serie de nodos que se comportan como iguales entre sí. Es decir, actúan simultáneamente como clientes y servidores respecto a los demás nodos de la red. Las redes P2P permiten el intercambio directo de información, en cualquier formato, entre los ordenadores interconectados.

RIA: *Rich Internet applications*, o *RIA* (en español "aplicaciones de Internet enriquecidas"), son aplicaciones web que tienen la mayoría de las características de las

aplicaciones de escritorio tradicionales. Estas aplicaciones utilizan un navegador web estandarizado para ejecutarse y por medio de complementos o mediante una máquina virtual se agregan las características adicionales.

Servidor: Un servidor es un ordenador remoto que provee los datos solicitados por parte de los navegadores de otras computadoras. Almacena documentos HTML, imágenes, archivos de texto, escrituras, y demás material Web compuesto por datos (conocidos colectivamente como contenido), y distribuye este contenido a clientes que la piden en la red.

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

VBScript: es un lenguaje interpretado por el Windows Scripting Host de Microsoft. Su sintaxis refleja su origen como variación del lenguaje de programación Visual Basic. Ha logrado un apoyo significativo por parte de los administradores de Windows como herramienta de automatización, ya que, conjunta y paralelamente a las mejoras introducidas en los sistemas operativos windows donde opera fundamentalmente, permite más margen de actuación y flexibilidad que el lenguaje *batch* (o de proceso por lotes) desarrollado a finales de los años 1970 para el MS-DOS.

XMLHttpRequest : es una interfaz empleada para realizar peticiones HTTP y HTTPS a servidores Web. Para los datos transferidos se usa cualquier codificación basada en texto, incluyendo: texto plano, XML, JSON, HTML y codificaciones particulares específicas. La interfaz se implementa como una clase de la que una aplicación cliente puede generar tantas instancias como necesite para manejar el diálogo con el servidor.

El uso más popular, si bien no el único, de esta interfaz es proporcionar contenido dinámico y actualizaciones asíncronas en páginas WEB mediante tecnologías construidas sobre ella como por ejemplo AJAX.



“Hay dos formas de realizar el diseño de una aplicación: La primera es el hacerlo tan sencillo que sea obvio para todos que no tiene deficiencias y la segunda es el hacerlo tan complicado que no queden deficiencias obvias”.

Tony Hoare

2 Capítulo II: Diseño y arquitectura del sistema

En esta etapa se utilizará la información obtenida del análisis previo, tomando en cuenta las necesidades del sistema, así como también las funcionalidades con las que contará el mismo. El diseño planteado en esta etapa deberá ser independiente al lenguaje de programación seleccionado anteriormente, siendo así universal y de fácil entendimiento.

2.1 Diseño funcional

Para el diseño adecuado del sistema se ha decidido dividir el sistema en módulos, en el primer módulo estarán las acciones que pueden realizar los académicos y en el segundo las acciones correspondientes el administrador del sistema.

Las funciones permitidas para cada módulo serían las siguientes:

- Académicos: Registro, creación, modificación y eliminación de Informes, imprimir informes, búsqueda de sus informes creados.
- Administrador: Activar y desactivar usuarios, creación, modificación y eliminación de informes, imprimir informes, búsqueda de todos los informes creados.

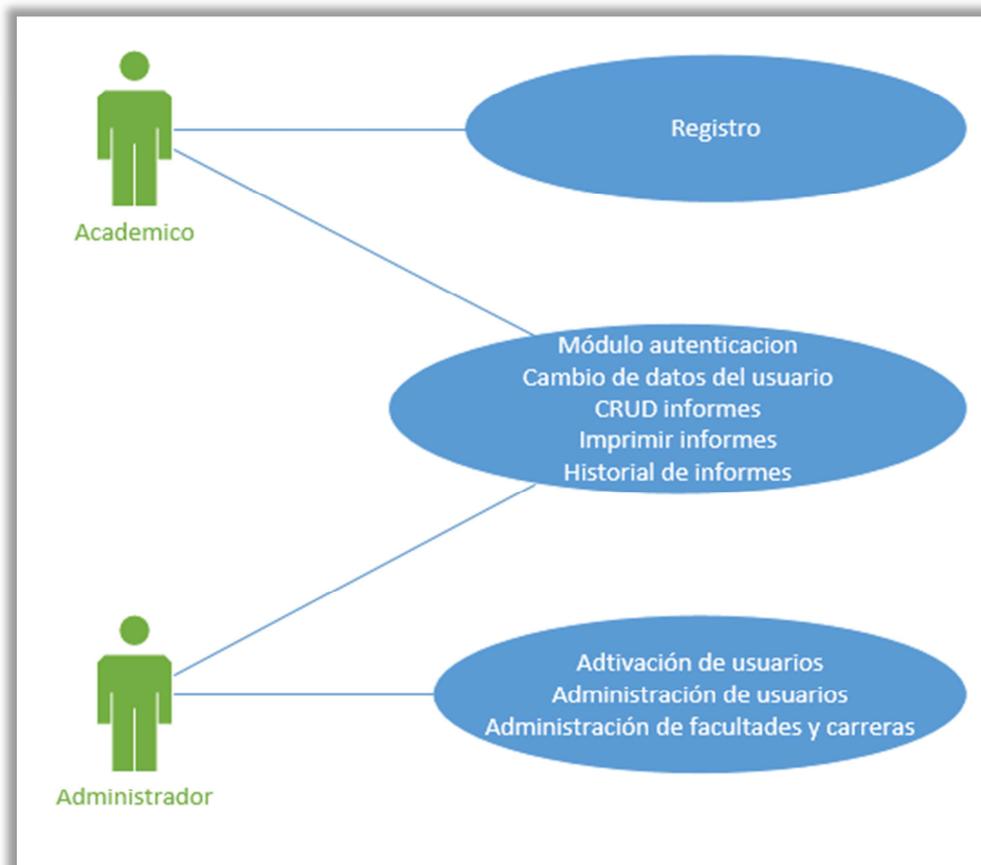


Figura 2-1 Casos de uso

2.2 Diseño detallado

A continuación se detallarán tanto los módulos comunes como los exclusivos para los académicos y para el administrador.

2.2.1 Módulos comunes

Estos módulos son comunes, es decir, los puede usar tanto el académico como el administrador.

Módulo de autenticación: Este módulo es el encargado de validar la autenticidad de un usuario, esto se hace mediante la verificación de un usuario valido activo y su respectiva contraseña, datos que fueron previamente obtenidos en el módulo del registro.

Módulo de cambio de datos: Este módulo permite al usuario cambiar sus datos personales, incluyendo la contraseña con la que se ingresa al sistema, este módulo resulta importante en el caso de que se haya introducido mal la información o exista la necesidad de actualizarla, un cambio de correo por ejemplo.

Módulo creación, modificación y eliminación de informes: Este módulo permitirá al usuario crear, modificar y eliminar un informe. Se deben considerar algunas restricciones en caso de que el usuario sea académico ya que él solamente podrá crear, modificar y eliminar sus informes, mientras que el administrador podrá tener acceso a estas mismas acciones de todos los informes del sistema.

Modelo imprimir informes: Este módulo permitirá al usuario tener una copia en formato pdf del informe creado, este módulo también cuenta con restricciones para el académico el cual solamente podrá tener acceso a sus informes creados.

Módulo historial de informes: Este módulo mostrará los informes creados por el usuario, una vez más el administrador tendrá el acceso a visualizar los informes de todos los académicos. Otra de las capacidades que tendrá el administrador en este módulo será el realizar búsquedas (por nombre, por carrera, por fecha de creación).

2.2.2 Módulos del académico

Módulo de registro: Este módulo permitirá al académico registrarse en el sistema, en este momento se le pedirán sus datos como nombre, apellidos, correo, contraseña, etc., después del registro el módulo se encargará de enviar un correo para la confirmación del registro.

2.2.3 Módulos del administrador

Módulo de activación: Este módulo presentará al administrador la opción de activar a los usuarios que realizaron la petición de registro al sistema.

Módulo de administración de usuarios: Este módulo permitirá modificar el estado de los usuarios (activo, inactivo), así como eliminarlos del sistema.

Módulo de administración de facultades y carreras: En este módulo el administrador podrá agregar, editar y eliminar elementos de los catálogos de facultades y carreras, según las necesidades del sistema y su crecimiento.

2.3 Base de datos

El modelado de la base de datos es muy importante en el desarrollo de un sistema, ya que de esto dependerá en gran medida el buen funcionamiento del sistema y el tiempo de respuesta del servidor ante una petición.

También se debe considerar que ante un buen modelado de la base de datos se evitaban errores en la programación del sistema.

2.3.1 Diseño de la base de datos

Una base de datos, o mejor dicho, un sistema gestor de bases de datos, es un software que gestiona una o más bases de datos y nos permite explotar los datos almacenados en ellas de forma relativamente simple mediante SQL.

Estructura mínima de almacenamiento:

- Tabla:

Objeto de almacenamiento perteneciente a una BD. Es una estructura en forma de cuadrante donde se almacenan registros o filas de datos. Cada tabla tiene un nombre único en la BD.

- Registro:

Cada una de las filas de una tabla, está compuesto por campos o atributos.

- Campo:

Cada uno de los “cajoncitos” de un registro donde se guardan los datos. Cada campo tiene un nombre único para la tabla de la cual forma parte, además es de un tipo (naturaleza) determinado, por tanto no podemos guardar limones en el cajón de las naranjas, en términos informáticos y a modo de ejemplo, no encontraremos un dato alfanumérico (letras y números) en un campo diseñado para guardar datos numéricos.

2.3.2 Análisis de requisitos

Después de analizar los requisitos en base al formato del informe PRIDE, se obtuvo la siguiente lista de datos, esta se encuentra dividida en 6 tablas y cubre los aspectos principales del informe.

Para los usuarios
Usuario
Contraseña
Nombre
Apellidos
Correo
Tipo
Estatus

Tabla 2-1 Requisitos del usuario

Para el informe
Fecha
Usuario creador
Estatus

Tabla 2-2 Requisitos del informe

Para los títulos
Título
Institución
Fecha

Tabla 2-3 Requisitos de los títulos

Para las actividades
Nombre de actividad
Institución
Fecha

Tabla 2-4 Requisitos de las actividades

Para las asignaturas
Nombre de asignatura
Semestre
Grupo
Número de alumnos
Horas de clase
Porcentaje de asistencias
Entrega puntual de actas

Tabla 2-5 Requisitos de las asignaturas

Para los extraordinarios
Nombre del examen
Semestre
Número de alumnos
Entrega puntual de actas

Tabla 2-6 Requisitos de los extraordinarios

2.3.3 Diseño conceptual

Con la información obtenida del análisis de requerimientos se planteó un primer modelo, este primer modelo cumplía con las relaciones necesarias, aun así, era necesario someterlo al proceso de normalización para obtener un modelo de datos más aproximado al estándar relacional.

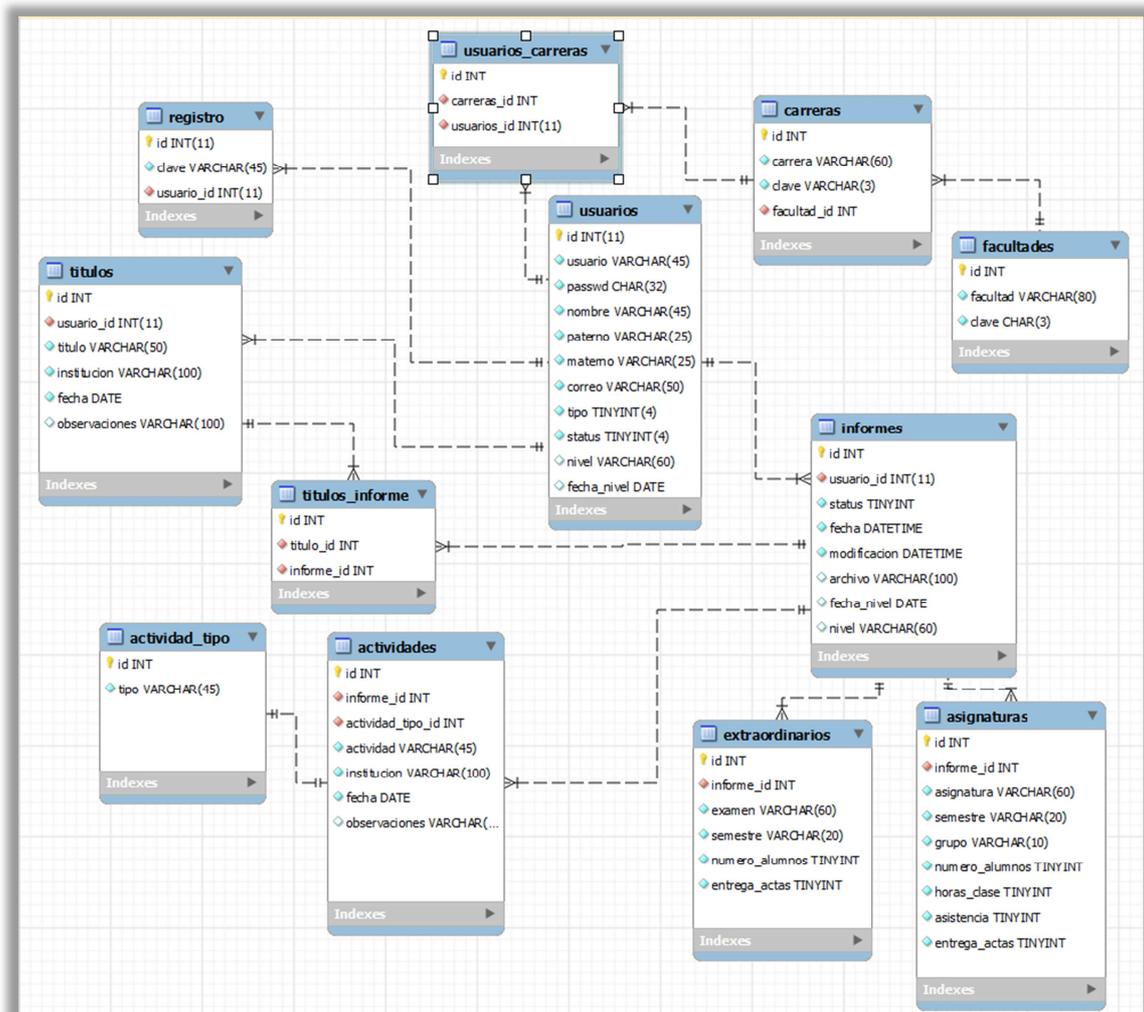


Figura 2-2 Modelo de la base del sistema

A continuación se muestra la información detallada del modelo, esta información es proporcionada por el herramienta navicat.

actividades						
Field Name	Field Type	Default	AllowNull	PriKey	Extra	
id	int(11) FIRST		NO	YES	auto_increment	
informe_id	int(11) AFTER `id`		NO	NO		
actividad_tipo_id	int(11) AFTER `informe_id`		NO	NO		
actividad	varchar(45) AFTER `actividad_tip		NO	NO		
institucion	varchar(100) AFTER `actividad`		NO	NO		
fecha	date AFTER `institucion`		NO	NO		
observaciones	varchar(100) AFTER `fecha`		YES	NO		
Key Name	Field Name	Unique	Collation	Cardinality	Packed	
id_UNIQUE	id	YES	A	0		
fk_actividad_informes1	informe_id	NO	A	0		
fk_actividad_actividad_tipo1	actividad_tipo_id	NO	A	0		

actividad_tipo						
Field Name	Field Type	Default	AllowNull	PriKey	Extra	
id	int(11) FIRST		NO	YES	auto_increment	
tipo	varchar(45) AFTER `id`		NO	NO		
Key Name	Field Name	Unique	Collation	Cardinality	Packed	
id_UNIQUE	id	YES	A	9		

Figura 2-3 Tabla actividades y actividad_tipo

asignaturas						
Field Name	Field Type	Default	AllowNull	PriKey	Extra	
id	int(11) FIRST		NO	YES	auto_increment	
informe_id	int(11) AFTER `id`		NO	NO		
asignatura	varchar(60) AFTER `informe_id`		NO	NO		
semestre	varchar(20) AFTER `asignatura`		NO	NO		
grupo	varchar(10) AFTER `semestre`		NO	NO		
numero_alumnos	tinyint(4) AFTER `grupo`		NO	NO		
horas_clase	tinyint(4) AFTER `numero_alumni		NO	NO		
asistencia	tinyint(4) AFTER `horas_clase`		NO	NO		
entrega_actas	tinyint(4) AFTER `asistencia`		NO	NO		
Key Name	Field Name	Unique	Collation	Cardinality	Packed	
id_UNIQUE	id	YES	A	0		
fk_asignaturas_informes1	informe_id	NO	A	0		

carreras						
Field Name	Field Type	Default	AllowNull	PriKey	Extra	
id	int(11) FIRST		NO	YES		
carrera	varchar(60) AFTER `id`		NO	NO		
clave	varchar(3) AFTER `carrera`		NO	NO		
facultad_id	int(11) AFTER `clave`		NO	NO		
Key Name	Field Name	Unique	Collation	Cardinality	Packed	
fk_carrea_facultad1	facultad_id	NO	A	2		

Figura 2-4 Tabla asignaturas y carreras

extraordinarios						
Field Name	Field Type	Default	AllowNull	PriKey	Extra	
id	int(11) FIRST		NO	YES	auto_increment	
informe_id	int(11) AFTER `id`		NO	NO		
examen	varchar(60) AFTER `informe_id`		NO	NO		
semestre	varchar(20) AFTER `examen`		NO	NO		
numero_alumnos	tinyint(4) AFTER `semestre`		NO	NO		
entrega_actas	tinyint(4) AFTER `numero_alumn		NO	NO		
Key Name	Field Name	Unique	Collation	Cardinality	Packed	
id_UNIQUE	id	YES	A	0		
fk_extraordinarios_informes1	informe_id	NO	A	0		

facultades						
Field Name	Field Type	Default	AllowNull	PriKey	Extra	
id	int(11) FIRST		NO	YES	auto_increment	
facultad	varchar(80) AFTER `id`		NO	NO		
clave	char(3) AFTER `facultad`		NO	NO		

Figura 2-5 Tabla extraordinarios y facultades

informes						
Field Name	Field Type	Default	AllowNull	PriKey	Extra	
id	int(11) FIRST		NO	YES	auto_increment	
usuario_id	int(11) AFTER `id`		NO	NO		
status	tinyint(4) AFTER `usuario_id`		NO	NO		
fecha	datetime AFTER `status`		NO	NO		
modificacion	datetime AFTER `fecha`		NO	NO		
archivo	varchar(100) AFTER `modificacion`		YES	NO		
fecha_nivel	datetime AFTER `archivo`		YES	NO		
nivel	varchar(60) AFTER `fecha_nivel`		YES	NO		
Key Name	Field Name	Unique	Collation	Cardinality	Packed	
id_UNIQUE	id	YES	A	5		
fk_informe_usuarios1	usuario_id	NO	A	2		

registro						
Field Name	Field Type	Default	AllowNull	PriKey	Extra	
id	int(11) FIRST		NO	YES	auto_increment	
clave	varchar(45) AFTER `id`		NO	NO		
usuario_id	int(11) AFTER `clave`		NO	NO		
Key Name	Field Name	Unique	Collation	Cardinality	Packed	
id_UNIQUE	id	YES	A	0		
usuarios_id_UNIQUE	usuario_id	YES	A	0		
fk_registro_usuarios1	usuario_id	NO	A	0		

Figura 2-6 Tabla informes y registro

titulos						
Field Name	Field Type	Default	AllowNull	PriKey	Extra	
id	int(11) FIRST		NO	YES	auto_increment	
usuario_id	int(11) AFTER `id`		NO	NO		
titulo	varchar(50) AFTER `usuario_id`		NO	NO		
institucion	varchar(100) AFTER `titulo`		NO	NO		
fecha	date AFTER `institucion`		NO	NO		
observaciones	varchar(100) AFTER `fecha`		YES	NO		
Key Name	Field Name	Unique	Collation	Cardinality	Packed	
id_UNIQUE	id	YES	A	1		
fk_titulos_usuarios1	usuario_id	NO	A	1		

titulos_informe						
Field Name	Field Type	Default	AllowNull	PriKey	Extra	
id	int(11) FIRST		NO	YES	auto_increment	
titulo_id	int(11) AFTER `id`		NO	NO		
informe_id	int(11) AFTER `titulo_id`		NO	NO		
Key Name	Field Name	Unique	Collation	Cardinality	Packed	
fk_titulos_informe_titulos1	titulo_id	NO	A	1		
fk_titulos_informe_informes1	informe_id	NO	A	1		

Figura 2-7 Tabla titulos y titulos_informe

usuarios						
Field Name	Field Type	Default	AllowNull	PriKey	Extra	
id	int(11) FIRST		NO	YES	auto_increment	
usuario	varchar(45) AFTER `id`		NO	NO		
passwd	char(32) AFTER `usuario`		NO	NO		
nombre	varchar(45) AFTER `passwd`		NO	NO		
paterno	varchar(25) AFTER `nombre`		NO	NO		
materno	varchar(25) AFTER `paterno`		NO	NO		
correo	varchar(50) AFTER `materno`		NO	NO		
tipo	tinyint(4) AFTER `correo`		NO	NO		
status	tinyint(4) AFTER `tipo`		NO	NO		
nivel	varchar(60) AFTER `status`		YES	NO		
fecha_nivel	datetime AFTER `nivel`		YES	NO		
Key Name	Field Name	Unique	Collation	Cardinality	Packed	
id_UNIQUE	id	YES	A	4		
usuario_UNIQUE	usuario	YES	A	4		
correo_UNIQUE	correo	YES	A	4		

usuarios_carreras						
Field Name	Field Type	Default	AllowNull	PriKey	Extra	
id	int(11) FIRST		NO	YES		
carreras_id	int(11) AFTER `id`		NO	NO		
usuarios_id	int(11) AFTER `carreras_id`		NO	NO		
Key Name	Field Name	Unique	Collation	Cardinality	Packed	
fk_usuarios_carreras_carreras1	carreras_id	NO	A	0		
fk_usuarios_carreras_usuarios1	usuarios_id	NO	A	0		

Figura 2-8 Tabla usuarios y usuarios_carreras

2.3.4 Diseño lógico

Las relaciones son necesarias para la comunicación del sistema, a continuación se muestra como se relacionan las tablas que contendrán la información del sistema.

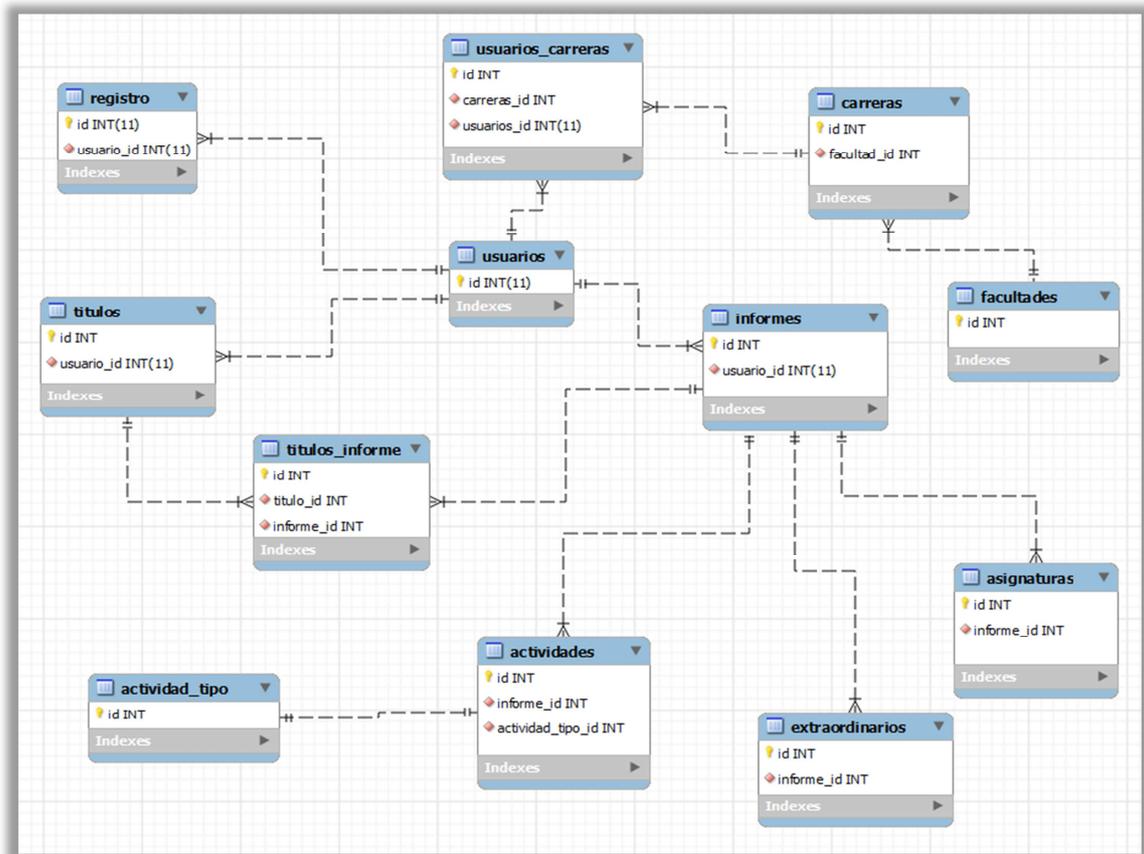


Figura 2-9 Relaciones de modelo de base del sistema

Las relaciones que se generan entre las tablas son las siguientes:

Usuarios – registro: Permite identificar el usuario que se registró, y saber si ya confirmo su registro.

Usuarios- títulos: Asignar uno o varios títulos académicos a un usuario.

Usuarios – informes: Es indispensable para la creación de informes, con esta relación se tiene identificado el usuario que creo el informe.

Usuarios – usuarios_carreras: Identifica la(s) carrera(s) del usuario a las cuales está asignado.

Informes – títulos_informe: Relaciona los títulos con los que cuenta el usuario al momento de crear un informe.

Informes – actividades: Esta relación une los informes junto con las actividades que pudo realizar el usuario en el periodo previo a hacer el informe.

Informes – extraordinarios: Necesaria para relacionar los exámenes extraordinarios realizados por el académico con el informe.

Informes – asignaturas: Relación necesaria entre el informe y las asignaturas impartidas por el académico en el periodo previo a realizar el informe.

Títulos – títulos_informe: Permite relacionar los títulos del académico con los del informe.

Actividades – actividad_tipo: Necesario para relacionar las actividades con el tipo de actividad, de este modo se puede saber si fue un curso de actualización, trayectoria académica, trayectoria profesional, etc.

Carreras – usuarios_carreras: Relaciona los usuarios con las carreras a las que se encuentra asignado.

Facultad – carreras: Identificador que permite agrupar las carreras en facultades.

2.4 Aspectos técnicos

Las herramientas necesarias para desarrollar el sistema serán las siguientes:

MySQL Workbench: Programa para diseño y administración de bases de datos

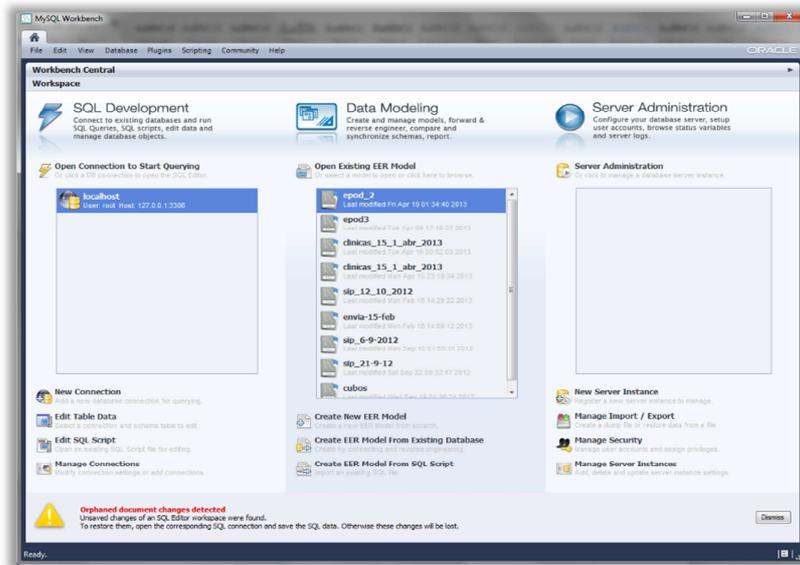


Figura 2-10 Interfaz MySQL Workbench

XAMPP: Servidor virtual, que funciona de forma local, y que cuenta con un manejador de bases de datos (MySQL), y sobre el cual se puede interpretar scripts de código PHP.

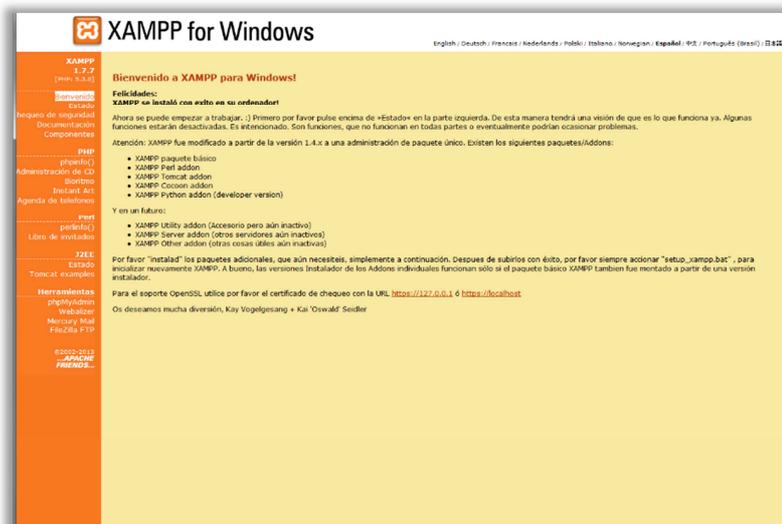


Figura 2-11 Interfaz XAMPP

Netbeans IDE: Ambiente de desarrollo para la creación de aplicaciones web con soporte para el lenguaje PHP, HTML, CSS Y Javascript.

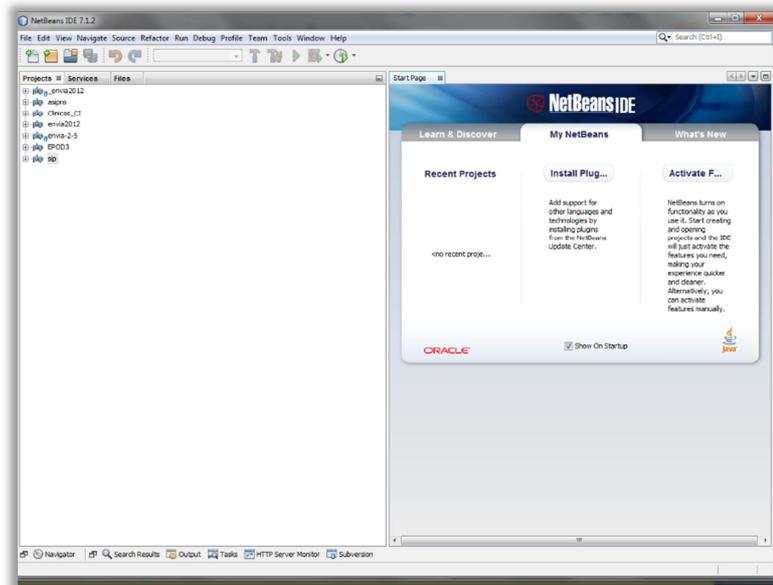


Figura 2-12 Interfaz Netbeans

Mozilla Firefox: Navegador web gratuito encargado de procesar el código HTML, CSS y Javascript.

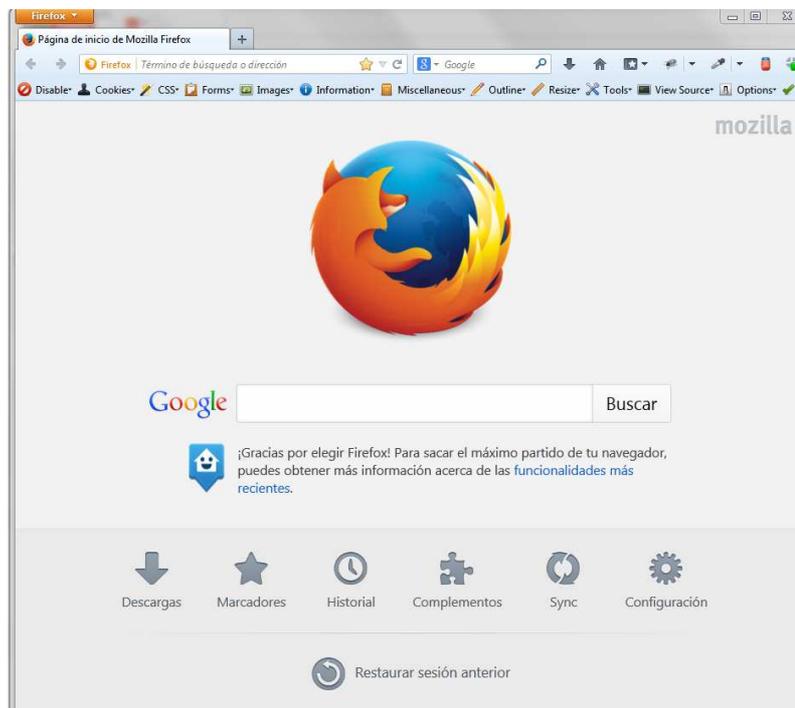


Figura 2-13 Interfaz de Mozilla Firefox

Firebug: Herramienta para el desarrollo web desarrollada para el navegador Mozilla Firefox, cuenta con utilidades con el que se puede analizar (revisar velocidad de carga,

estructura DOM), editar, monitorear y depurar el código fuente, CSS, HTML y Javascript de una página web de manera instantánea.

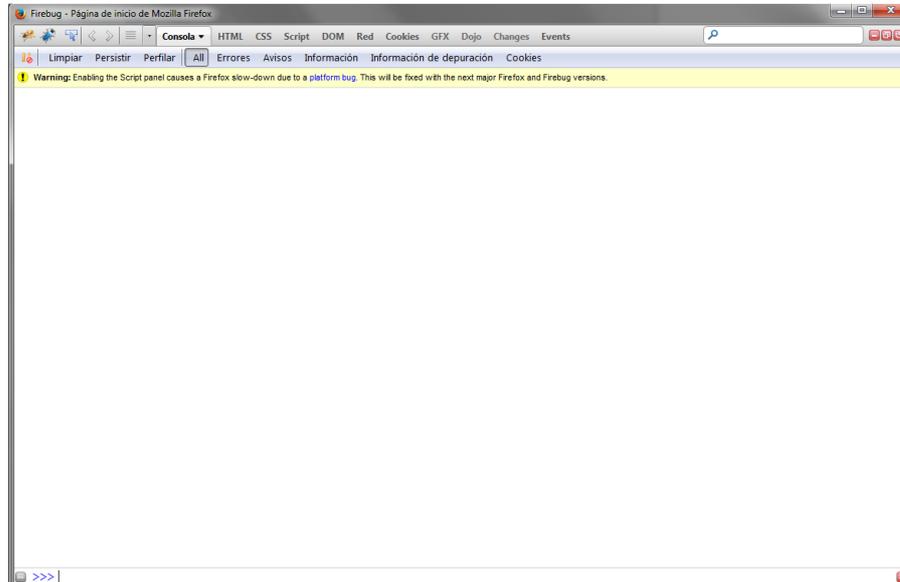


Figura 2-14 Interfaz de Firebug

Las cinco herramientas son software libre, además de ser estables y aceptadas en el mundo del desarrollo web.

2.5 Arquitectura del sistema

Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página, el modelo es el sistema de gestión de base de datos y la lógica de negocio y el controlador es el responsable de recibir los eventos de entrada desde la vista.

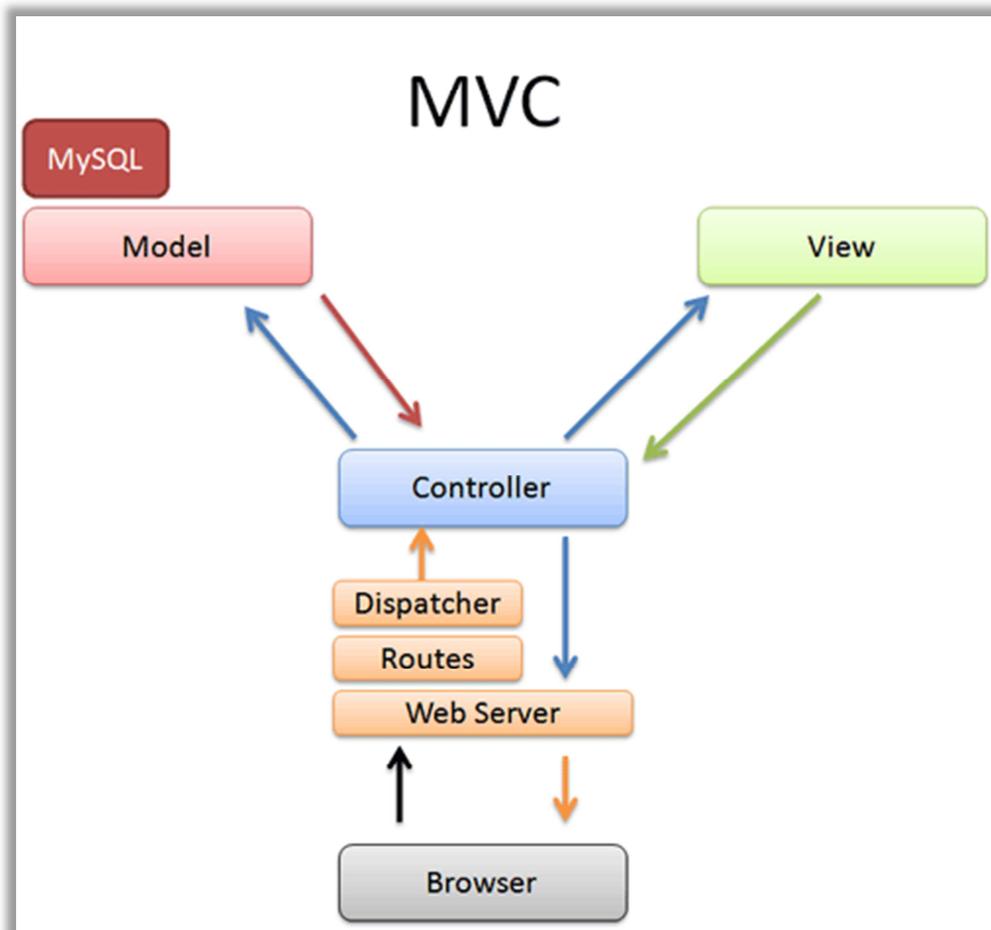


Figura 2-15 Modelo Vista Controlador (MVC)

La finalidad del modelo es mejorar la reusabilidad por medio del desacople entre la vista y el modelo. Los elementos del patrón son los siguientes:

- El **Modelo**: Es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica de negocio). Envía a la 'vista' aquella parte de la información que en cada momento se le solicita para que sea mostrada (típicamente a un usuario). Las peticiones de acceso o manipulación de información llegan al 'modelo' a través del 'controlador'.
- El **Controlador**: Responde a eventos (usualmente acciones del usuario) e invoca peticiones al 'modelo' cuando se hace alguna solicitud sobre la información (por ejemplo, editar un documento o un registro en una base de datos). También puede enviar comandos a su 'vista' asociada si se solicita un cambio en la forma en que se presenta de 'modelo' (por ejemplo, desplazamiento o scroll por un documento o por los diferentes registros de una base de datos), por tanto se podría decir que el 'controlador' hace de intermediario entre la 'vista' y el 'modelo'.

- La **Vista**: Presenta el 'modelo' (información y *lógica de negocio*) en un formato adecuado para interactuar (usualmente la interfaz de usuario) por tanto requiere de dicho 'modelo' la información que debe representar como salida.

2.6 Diseño de pantallas

Esta etapa se presentará la interfaz con la que contará el usuario dentro del sistema, esta interfaz deberá cumplir con las siguientes características.

- Clara y ordenada
- Ser intuitiva para un fácil manejo
- Indicaciones claras sobre lo que se está haciendo
- Significado específico para cada uno de los elementos mostrados

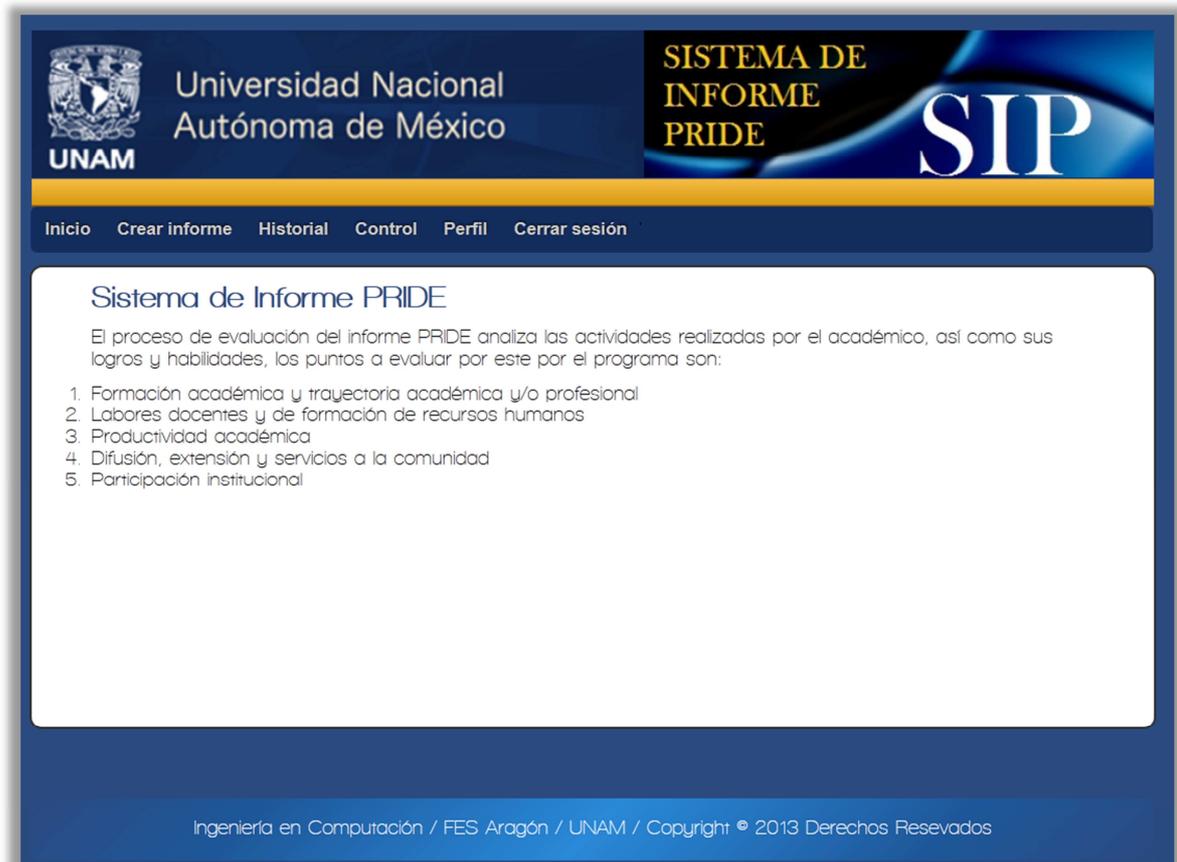


Figura 2-16 Interfaz: Página Principal

The image shows a login interface with a dark blue background. At the top center, the word "Acceso" is written in white. Below it, there are two input fields: "Usuario" and "Contraseña", each with a white border and a light blue placeholder. To the right of the "Contraseña" field is a white button with the text "Entrar". At the bottom of the form, there are two more buttons: "Registrarse" and "Recordar Clave", both with white text on a light blue background.

Figura 2-17 Interfaz: Formulario de inicio de sesión

The image shows a registration interface with a dark blue background. At the top left, the word "Registrar" is written in white. Below it, there are seven input fields: "Usuario", "Nombre", "Apellido Paterno", "Apellido Materno", "Correo", "Contraseña", and "Confirmar Contraseña", each with a white border and a light blue placeholder. To the right of the "Confirmar Contraseña" field is a small image of a CAPTCHA with the text "uagwe" and a label "Texto en la imagen" above a corresponding input field. At the bottom of the form, there is a large black button with the text "Registrar" in white.

Figura 2-18 Interfaz: Formulario de registro

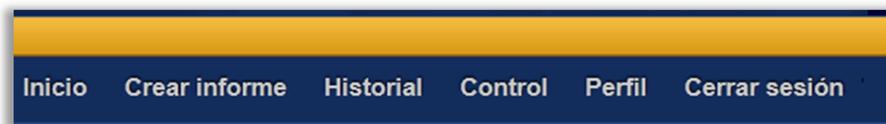


Figura 2-19 Interfaz: Menú del usuario

A form titled 'Cambio de contraseña' with a light gray background. It contains three input fields: 'Contraseña actual', 'Nueva contraseña', and 'Repetir contraseña'. Below the fields is a 'Guardar' button.

Figura 2-20 Interfaz: Cambio de contraseña

A form titled 'Información del usuario' with a light gray background. It contains six input fields: 'Nombre' (Edward), 'Apellido Paterno' (Carmona), 'Apellido Materno' (Soto), 'Correo' (asd@asd.asd), 'Nivel actual' (A), and 'Fecha de nivel actual' (2013-07-31). Below the fields is a 'Guardar' button.

Figura 2-21 Interfaz: Cambio de información de perfil

[Formación y Trayectoria](#) | [Labor Docente](#) | [Productividad Académica](#) | [Difusión](#) | [Participación Institucional](#) |

A. Formación Académica y Trayectoria Académica y/o Profesional

▼ A.1. Nivel Académico

Título / Grado + Agregar
 Ing. en Computacion, Universidad Nacional Autonoma de Mexico
 02/06/2013
 Muy bien
En caso de renovación serán consideradas únicamente las actividades de los últimos 3 años

Curso de Actualización + Agregar

▶ A.2. Trayectoria Académica y/o Profesional

Figura 2-22 Interfaz: Menú para crear informe

Seleccionar filtro de búsqueda
 Últimos informes creados

Mostrar

Ape. Pat.	Ape. Mat.	Nombre	Fecha de Creación	Fecha de Modificación	Liga de infome	Archivo adjunto	Eliminar informe
Carmona	Soto	Edward	2013-06-10 00:25:07	2013-06-10 01:11:11	Editar		Eliminar
Carmona	Soto	Edward	2013-06-10 01:17:07	2013-06-10 01:40:56	Editar		Eliminar
Carmona	Soto	Edward	2013-06-10 01:43:45	2013-06-10 01:43:48	Editar		Eliminar
Carmona	Soto	Edward	2013-06-10 01:44:29	2013-06-10 01:44:58	Editar		Eliminar
Carmona	Soto	Edward	2013-06-10 01:46:38	2013-06-10 01:46:42	Editar		Eliminar

Mostrando 1 a 5 de un total de 5 informes

Figura 2-23 Interfaz: Historial de informes

Mostrar 10		Buscar			
Activar	Usuario	Nombre	Apellido Paterno	Apellido Materno	Correo
Activar	alberto	Alberto	Carmona	Garcia	aswd@aqd.cgt

Mostrando 1 a 1 de un total de 1 usuarios

Figura 2-24 Interfaz: Activar usuarios

Mostrar 10		Buscar			
Nombre	Apellido Paterno	Apellido Materno	Correo	Editar	Eliminar
Arcella	Bernal	Diaz	arcellabernal@hotmail.com	Editar	Eliminar
Juan	Carmona	Lopez	asdwas@dad.cdf	Editar	Eliminar

Mostrando 1 a 2 de un total de 2 usuarios

Figura 2-25 Interfaz: Modificar usuarios

2.7 Compatibilidad

La compatibilidad es un tema demasiado importante en el desarrollo de un sistema, aún más tomando en cuenta que se trata de un desarrollo web, y el usuario final puede decidirse a utilizar un distinto navegador, de un día para otro, para cumplir con este objetivo se pretende realizar el desarrollo del sistema apegado a los estándares internacionales teniendo como resultado un sistema estable en los navegadores más populares como lo son: Mozilla Firefox, Google Chrome, Opera y Safari, suponiendo que el usuario tiene instalado la última versión de uno de ellos.

Los navegadores actúan como traductores, traducen los documentos HTML en páginas web formateadas. Las reglas básicas para traducir estos documentos son establecidas por la World Wide Web Consortium (W3C), que se encarga de publicar los estándares oficiales del HTML en su sitio web (<http://www.w3c.org/>). Ahora, esto es todo un tema, por no decir problema, ya que no todos los navegadores se rigen en base a estas normas, como así tampoco las interpretan de la misma manera. Por ejemplo, W3C establece que el tag TABLE debe soportar el atributo CELLSPACING para que uno pueda definir el espacio entre celdas. Pero los estándares no definen que valor lleva ese atributo por defecto, por lo tanto, si no especificamos un valor, los navegadores lo hacen por nosotros, y este valor por defecto varía entre ellos.

Tenemos que aprender a ver el sistema con los ojos de nuestros usuarios. Esto implica verlo con el mismo navegador, las mismas preferencias, el mismo sistema operativo, las mismas restricciones, el mismo monitor y hasta la misma resolución. Hasta el día de hoy, el método de prueba y error sigue siendo el más efectivo a la hora de combatir la incompatibilidad.

2.8 Seguridad

Un aspecto realmente importante es la seguridad del sistema, este protegerá la información que se pueda obtener de los usuarios y también la que pueda generar el sistema.

La seguridad que el sistema puede proporcionar se divide en dos, los que se pueden manejar mediante la lógica del sistema (restricción a módulos y/o acciones por tipo de usuario) y a los que tiene responder el sistema frente ataques externos.

2.8.1 Seguridad interna

Identificador de usuario: este dato permite en todo momento saber que usuario se encuentra autenticado, y por lo tanto quien realiza las acciones, de esta forma se puede obtener un registro de acciones realizadas por del usuario.

Identificador de tipo de usuario: mediante este identificador delimita las acciones a las cuales tiene permiso el usuario, por ejemplo, en el caso de ser usuario de tipo administrador tendrás acceso a todos los informes.

Permiso a nivel de base de datos: el definir los permisos con los que cuenta un usuario al conectarse a la base de datos disminuye el riesgo de pérdida de información, se debe considerar también que otros sistemas conviven con el nuestro en un mismo servidor y descuidar la base de datos pondría en peligro los demás sistemas, para tal fin se le asignarán permisos bien definidos dentro del gestor de bases de datos.

2.8.2 Ataques externos (SQL Injection y XSS)

Inyección SQL es un método de infiltración de código intruso que se vale de una vulnerabilidad informática presente en una aplicación en el nivel de validación de las entradas para realizar consultas a una base de datos.

El origen de la vulnerabilidad radica en el incorrecto chequeo y/o filtrado de las variables utilizadas en un programa que contiene, o bien genera, código SQL. Es, de hecho, un error de una clase más general de vulnerabilidades que puede ocurrir en cualquier lenguaje de programación o script que esté embebido dentro de otro.

Se conoce como Inyección SQL, indistintamente, al tipo de vulnerabilidad, al método de infiltración, al hecho de incrustar código SQL intruso y a la porción de código incrustado.

Este tipo de intrusión normalmente es de carácter malicioso, dañino o espía, por tanto es un problema de seguridad informática, y debe ser tomado en cuenta por el programador de la aplicación para poder prevenirlo. Un programa elaborado con descuido, displicencia o con ignorancia del problema, podrá resultar ser vulnerable, y la seguridad del sistema (base de datos) podrá quedar eventualmente comprometida.

La intrusión ocurre durante la ejecución del programa vulnerable, ya sea, en computadores de escritorio o bien en sitios Web, en éste último caso obviamente ejecutándose en el servidor que los aloja.

La vulnerabilidad se puede producir automáticamente cuando un programa "arma descuidadamente" una sentencia SQL en tiempo de ejecución, o bien durante la fase de desarrollo, cuando el programador explicita la sentencia SQL a ejecutar en forma desprotegida. En cualquier caso, siempre que el programador necesite y haga uso de parámetros a ingresar por parte del usuario, a efectos de consultar una base de datos; ya que dentro de los parámetros es donde se puede incorporar el código SQL intruso.

Al ejecutarse la consulta en la base de datos, el código SQL inyectado también se ejecutará y podría hacer un sinnúmero de cosas, como insertar registros, modificar o eliminar datos, autorizar accesos o ejecutar otro tipo de código malicioso en el computador.

Por ejemplo, asumiendo que el siguiente código reside en una aplicación web y que existe un parámetro "nombreUsuario" que contiene el nombre de usuario a consultar, una inyección SQL se podría provocar de la siguiente forma:

El código SQL original y *vulnerable* es:

```
$consulta = "SELECT * FROM usuarios WHERE nombre = '".$nombreUsuario."'";
```

En el lenguaje PHP, hay diferentes funciones que pueden servir de ayuda para usar con distintos sistemas de gestión de bases de datos.

Si se usa MySQL, la función a usar es `mysql_real_escape_string`:

```
$query_result = mysql_query("SELECT * FROM usuarios WHERE nombre = \"\" .  
mysql_real_escape_string($nombre_usuario) . \"\");
```

No obstante es más recomendado usar alternativas que ofrecen consultas preparadas como la clase PDO.

```
$statement = $pdo->prepare("SELECT * FROM usuarios WHERE nombre =  
:nombre");  
$statement->bindParam(':nombre', $nombre_usuario);  
$statement->execute();  
$result = $statement->fetch();
```

XSS, del inglés *Cross-site scripting* es un tipo de inseguridad informática o agujero de seguridad típico de las aplicaciones Web, que permite a una tercera parte inyectar en páginas web vistas por el usuario código JavaScript o en otro lenguaje script similar (ej: VBScript), evitando medidas de control como la Política del mismo origen. Este tipo de vulnerabilidad se conoce en español con el nombre de *Secuencias de comandos en sitios cruzados*.

Es posible encontrar una vulnerabilidad XSS en aplicaciones que tenga entre sus funciones presentar la información en un navegador web u otro contenedor de páginas web. Sin embargo, no se limita a sitios web disponibles en Internet, ya que puede haber aplicaciones locales vulnerables a XSS, o incluso el navegador en sí.

XSS es un vector de ataque que puede ser utilizado para robar información delicada, secuestrar sesiones de usuario, y comprometer tanto el navegador como el sistema. Las vulnerabilidades XSS han existido desde los primeros días de la Web.

Esta situación es usualmente causada al no validar correctamente los datos de entrada que son usados en cierta aplicación, o no sanear⁶⁰ la salida adecuadamente para su presentación como página web.

Esta vulnerabilidad puede estar presente de las siguientes formas:

- Directa (también llamada Persistente): este tipo de XSS comúnmente filtrado, y consiste en embeber código HTML peligroso en sitios que lo permitan; incluyendo así etiquetas como `<script>` o `<iframe>`.
- Indirecta (también llamada Reflejada): este tipo de XSS consiste en modificar valores que la aplicación web utiliza para pasar variables entre dos páginas, sin usar sesiones y sucede cuando hay un mensaje o una ruta en la URL del navegador, en una *cookie*, o cualquier otra cabecera HTTP (en algunos navegadores y aplicaciones web, esto podría extenderse al DOM del navegador).

Codeigniter protege las entradas por donde podría entrar tanto código SQL Injection como XSS, esto da más seguridad al sistema. El propio framework se encarga de limpiar las entradas automáticamente, lo que hace que el desarrollador se quite un peso de encima en cuanto a la seguridad.

⁶⁰ **Sanear** o **Sanitizar** es sacar de la entrada al software o página web todo lo que pueda alterar su funcionamiento, ya sea por inyección HTML, inyección JavaScript o inyección SQL. Cada tipo de inyección requiere un saneo diferente porque la naturaleza de cada problema está dada por cada lenguaje en particular.



“El objetivo principal de la educación es crear personas capaces de hacer cosas nuevas, y no simplemente repetir lo que otras generaciones hicieron”

Jean Piaget

3 Capítulo III: Desarrollo del Sistema

En esta etapa es donde el análisis y diseño tomarán forma y se creará el sistema, haciendo uso de los lenguajes de programación, el modelo de datos, el administrador de base de datos, el framework de lado del servidor y el framework de lado del cliente se construirá el sistema en base a las necesidades descritas en los capítulos anteriores, permitiendo así que el usuario pueda disponer de un sistema que le ayude a administrar el formato PRIDE.

3.1 Adecuación del entorno

En este punto del ciclo de la creación del sistema es necesario contar con los recursos básicos que permitirán el desarrollo del mismo como lo son el equipo de cómputo y el software.

El equipo de cómputo que se ocupó tenía instalado el siguiente software: XAMMP, MySQL workbench, Google Chrome, Internet Explorer, Mozilla Firefox, Opera, Safari, Netbeans IDE como ambiente de desarrollo para el lenguaje PHP y PDF Reader.

3.2 Desarrollo de módulos

El uso de un framework para el desarrollo del sistema facilita la programación ya que el framework cuenta con un gran número de herramientas, Codeigniter cuenta además con el modelo de desarrollo MVC que divide el sistema en capas, de esta forma los datos están separados de la lógica del sistema y separados al mismo tiempo de la vista o interfaz que visualizará un usuario, permitiendo trabajar de una manera más limpia.

Las herramientas que Codeigniter provee propician que la programación sea más ordenada y estructurada, al usar las herramientas damos un control al framework a tareas específicas como el acceso a la base de datos, el procesamiento de peticiones, entre otras, esto hace que el sistema esté más ordenado y evita en gran medida la repetición de funciones con el mismo funcionamiento, además de que asegura el buen funcionamiento ya que las herramientas incorporadas el framework han sido probadas con anterioridad y han demostrado estabilidad.

3.3 Descripción de funciones

Debe tenerse en cuenta que el sistema fue desarrollado en un framework, esto hace que las funciones y módulos con los que cuenta hayan sido previamente probadas y sean estables.

3.3.1 Descripción de controladores

Los controladores son el corazón de la aplicación, ya que determinan como se manejan las solicitudes HTTP. Un Controlador es simplemente un archivo de clase que se nombra de una forma en la que se puede asociar con una URI.

A continuación se listan los controladores creados y sus métodos correspondientes:

Principal

Principal	
Ruta	Descripción
Index	Muestra la página principal del sistema
Login	Muestra la vista del login
Registro	Muestra la vista para registro de usuario
Captcha	Muestra la vista del captcha

Tabla 3-1 Controlador: Principal

Acceso

Login	
Ruta	Descripción
Valida	Se encarga de validar los datos para iniciar sesión
Logout	Cierra la sesión
Intentos	Verifica los intentos de acceso al sistema
Captcha	Verifica que la información ingresada coincida con la captcha

Tabla 3-2 Controlador: Login

Registro	
Ruta	Descripción
Checar_usuario	Verifica si ya está registrado el nombre de un usuario
Checar_correo	Verifica si ya está registrado un correo
Registro	Registra un nuevo usuario
Recordar	Manda un correo al usuario con una nueva contraseña

Tabla 3-3 Controlador: Registro

Admin

Admin	
Ruta	Descripción
Index	Muestra la vista principal de administración
Alta	Muestra los usuarios que no han sido activados
Active	Activa un usuario
Edit_user	Muestra la vista para editar un usuario
Edit	Actualiza la información de un usuario
Delete	Elimina un usuario
Facultades	Muestra la lista de las facultades y carreras

Tabla 3-4 Controlador: Admin

Carreras	
Ruta	Descripción
Get_carreras	Muestra la lista de carreras
New_carrera	Crea una nueva carrera
Edit_carrera	Muestra la vista para editar una carrera
Update_carrera	Actualiza una carrera
Delete_carrera	Elimina una carrera

Tabla 3-5 Controlador: Carreras

Facultades	
Ruta	Descripción
Get_facultades	Muestra la lista de facultades
New_facultad	Crea una nueva facultad
Edit_facultad	Muestra la vista para editar una facultad
Update_facultad	Actualiza una facultad
Delete_facultad	Elimina un facultad

Tabla 3-6 Controlador: Facultades

Historial

Historial	
Ruta	Descripción
Index	Muestra la lista de los informes creados
Buscar	Muestra la lista de los informes después de aplicar un filtro

Tabla 3-7 Controlador: Historial

Informe

Actividad	
Ruta	Descripción
Index	Muestra la vista principal de una actividad
New_actividad	Crea una nueva actividad
Edit_actividad	Muestra la vista para editar una actividad
Update_actividad	Actualiza una actividad
Delete_actividad	Elimina una actividad

Tabla 3-8 Controlador: Actividad

Asignatura	
Ruta	Descripción
Index	Muestra la vista principal de una asignatura
New_asignatura	Crea una nueva asignatura
Edit_asignatura	Muestra la vista para editar una asignatura
Update_asignatura	Actualiza una asignatura
Delete_asignatura	Elimina una asignatura

Tabla 3-9 Controlador: Asignatura

Extraordinario	
Ruta	Descripción
Index	Muestra la vista principal de un examen extraordinario
New_extraordinario	Crea un nuevo examen extraordinario
Edit_extraordinario	Muestra la vista para editar un examen extraordinario
Update_extraordinario	Actualiza un examen extraordinario
Delete_extraordinario	Elimina un examen extraordinario

Tabla 3-10 Controlador: Extraordinario

Informe	
Ruta	Descripción
Index	Muestra la vista principal de un informe
Edit_informe	Muestra la vista para editar un informe
Update_informe	Actualiza un informe
Delete_informe	Elimina un informe
Upload_file	Adjunta un archivo al informe
Delete_file	Elimina el archivo adjunto de un informe

Tabla 3-11 Controlador: Informe

Titulo	
Ruta	Descripción
Index	Muestra la vista principal de un titulo
New_titulo	Crea un titulo
Edit_titulo	Muestra la vista para editar un titulo
Update_titulo	Actualiza un titulo
Delete_titulo	Elimina un titulo

Tabla 3-12 Controlador: Titulo

Pdf

Pride pdf	
Ruta	Descripción
Index	Muestra el formato del informe PRIDE en pdf

Tabla 3-13 Controlador: Pride_pdf

Perfil

Usuario	
Ruta	Descripción
Index	Muestra la vista principal del usuario
Cambiar_clave	Muestra la vista para cambiar clase
Cambio_clave	Cambia la clave
Edit_perfil	Muestra la vista para editar el usuario
Update_perfil	Actualiza el usuario

Tabla 3-14 Controlador: Usuario

3.3.2 Descripción de modelos

Los modelos son clases de PHP que se diseñan para trabajar con información en su base de datos. A continuación se listan los modelos usados por el sistema.

- Login
- Registro
- Usuarios
- Carreras
- Facultades
- Informes
- Actividades
- Asignaturas
- Extraordinarios
- Títulos

Los modelos usados cuentan con las funciones necesarias para crear, actualizar, leer y eliminar, esto es llamado CRUD (por su acrónimo en inglés de Create, Read, Update y Delete) y son las funciones básicas con las que cuenta cada modelo.

Codeigniter usa una versión modificada del Patrón de Base de Datos Active Record. Este patrón permite que la información sea obtenida, insertada y actualizada en la base de datos con mínimo código. En algunos casos solamente son necesarias una o dos líneas de código para ejecutar una acción en la base de datos. Codeigniter no necesita que cada tabla de base de datos tenga su propio archivo de clase. En lugar de eso, provee una interfaz más simplificada.

Más allá de la simplicidad, el mayor beneficio de usar las funcionalidades del Active Record, es que le permite crear aplicaciones independientes de las bases de datos, ya que cada adaptador de base de datos genera la sintaxis de la consulta. Esto también permite consultas más seguras, ya que el sistema escapa automáticamente los valores.

Las funciones básicas de active record son:

- Select
- From
- Where
- Or_where
- Where_in
- Or_where_in
- Limit
- Get
- Order_by
- Group_by
- Join
- Like

- Not_like
- Having
- Or_having
- Distinct
- Insert
- Update
- Delete

3.3.3 Descripción de helpers

Los helpers, como su nombre sugiere, ayudan con las tareas. Cada archivo de helper es simplemente una colección de funciones en una categoría particular. Hay Helpers de URL que ayudan en la creación de enlaces, Helpers de Formulario que lo ayudan a crear elementos de formulario, Helpers de Texto que ejecutan varias rutinas de formateo de texto, Helpers de Cookie que escriben y leen cookies, Helpers de Archivo que ayudan a trabajar con archivos, etc.

Uno de los helpers utilizados en el desarrollo del sistema fue el de URL que contiene funciones que ayudan a trabajar con URLs, un ejemplo es la siguiente función.

`Base_url()`: Devuelve la URL base del sitio, según se especifica en el archivo de configuración. En este caso la dirección es `http://ingenieria.aragon.unam.m/sip`

Existen dos archivos helpers especialmente creados para este sistema. El primero, fue creado con el fin de facilitar el manejo de las sesiones dentro del sistema con variables que definen por ejemplo, el tipo de usuario y su identificador. El segundo helper fue creado para facilitar el uso de envío de correos a los usuarios, como puede ser el correo de confirmación de registros, así como en él envío de correo de creación de una nueva contraseña.



“La educación es el pasaporte hacia el futuro, el mañana pertenece a aquellos que se preparan para él en el día de hoy”
Malcolm X

4 Capítulo IV: Implantación del Sistema

4.1 Implantación

Aquí se evaluará el lugar físico y lógico específico para instalar el sistema, y así obtener un óptimo funcionamiento. Al mismo tiempo el servidor será sujeto a un análisis para verificar que el software instalado cumpla los requisitos para el funcionamiento del sistema, como lo pueden ser el manejo de versiones de software recientes, o ajustes de configuraciones del servidor.

4.1.1 Instalación del sistema

Para la instalación del sistema son necesarios los siguientes elementos:

Servidor: computadora necesaria para almacenar el sistema, esta computadora deberá contar con el software necesaria que fue especificado anteriormente, como lo es PHP, MYSQL.

Base de datos: debemos contar con un gestor de base de datos, que ya se encuentre instalado en el servidor.

Dirección web: para que los usuarios puedan entrar al sistema, se necesita que el servidor tenga ya establecida una dirección pública que sea accesible desde internet.

4.1.2 Carga inicial de datos

Existen datos necesarios para que el sistema funcione correctamente, estos datos definen la configuración inicial del sistema. Dentro de los datos necesarios se encuentra la información del administrador del sistema, como debe ser obvio estos datos deberán estar cargados antes de ingresar por primera vez como usuario administrador al sistema.

4.1.3 Pruebas de implantación

Debido a que el servidor en el que se instaló el sistema, ya se había realizado la instalación de sistemas con características similares, se tuvo la expertis necesaria para realizar la implementación sin contratiempos.

4.1.4 Preparación del mantenimiento

Se instruyó a los administradores acerca del correcto uso del sistema, con la finalidad de que estos puedan dar un mantenimiento al sistema en un futuro. Para futuras referencias se entregaron manuales como apoyo a los administradores.

4.2 Pruebas

Se deberán hacer las pruebas necesarias que permitan verificar que el sistema funciona correctamente, de esta manera, en caso de presentar algún problema se deberá corregir de inmediato y así contar con un sistema estable y en buen funcionamiento.

4.2.1 Pruebas unitarias

Para las pruebas unitarias cada parte del sistema debe comprobar su funcionamiento por separado.

Lo primero a hacer es examinar cada función del módulo por separado, posteriormente se evaluará por completo el módulo y se confirmará que su desempeño es el esperado.

En caso de que una función o módulo completo no funcione correctamente se procederá inmediatamente a realizar los ajustes necesarios para el buen funcionamiento, ya que se debe tener en cuenta de que en caso de que una función o módulo trabaje erróneamente puede comprometer a todo el sistema.

4.2.2 Pruebas de integración

Después de haber hecho las pruebas unitarias se realizarán las pruebas de integración, estas pruebas constan en someter a los módulos a trabajar en conjunto, esto dará a su vez un mejor panorama del funcionamiento general del sistema.

Si las pruebas arrojan resultados que no concuerdan con los esperados, se tendría que llevar a cabo una revisión más para identificar el origen del error, y hacer las rectificaciones necesarias.

4.2.3 Pruebas en navegadores

La compatibilidad de los sistemas entre los distintos navegadores que se encuentran en el mercado, es probablemente uno de los dolores de cabeza más frecuentes en las personas que se dedican a la web, aunque ese dolor de cabeza desde mi punto de vista tiene nombre y apellido, y se hace llamar Internet Explorer. El sistema desarrollado funciona de buena manera en el navegador Internet Explorer 7 y posteriores, aunque no se descarta alguna mala presentación visual en versiones más antiguas.

Por el contrario a lo que sucede en IE, el sistema no tiene cambios visuales en los navegadores Mozilla Firefox y Google Chrome que al momento de hacer las pruebas se encontraban en las versiones 23 y 28 respectivamente.

Formando parte del mercado también se encuentran los navegadores Opera en su versión 15 y Safari la 5.1.7, los cuales poseen un menor número de usuarios, aún así el funcionamiento del sistema en estos navegadores es bueno.

Hay que mencionar que el sistema no incluye objetos creados con la tecnología de flash, lo cual hace que funcione sin la necesidad de instalar un software adicional.

4.2.4 Pruebas de saturación

Es importante que el sistema reaccione bien cuando se sature de peticiones, en este caso esta opción no es descartada y se hicieron las pruebas necesarias, aunque se sabe que la naturaleza del sistema no exige una gran saturación, como en el caso de un sistema de inscripciones.

Las pruebas a las que fue sometido, fue el constante uso del sistema durante determinado tiempo por un grupo de usuarios, la respuesta del sistema fue favorable.

4.3 Presentación y aprobación del sistema

El sistema fue presentado al jefe de carrera de ingeniería en computación, quien verifico que el sistema cumpliera las funciones por las cuales fue elaborado.

4.4 Formación de usuarios

Se capacitó a los usuarios finales del sistema, en base a un manual que fue hecho previamente. Durante la capacitación se mostró al usuario cada una de las opciones con las que cuenta.

4.4.1 Elaboración del manual de usuarios

La información acerca del manejo del sistema se encuentra en el manual de usuario, en él se muestran los flujos normales del sistema y los errores que pueden ocurrir y de esa manera evitarlos.

4.1.1 4.4.2 Formación de usuarios finales

Se realizaron reuniones para explicar el funcionamiento del sistema, esta capacitación a los usuarios finales fue rápida. La claridad del diseño aunado a la familiarización de los usuarios a los sistemas permitió una fácil adaptación a este.

4.5 Mantenimiento

Hasta la creación de este documento el sistema no ha requerido actualización, o algún tipo de mantenimiento.

4.5.1 Mejoras

El alcance del informe PRIDE es bastante amplio, este sistema se centró en la creación del informe, pero no está de más proponer mejoras para hacer de este sistema uno

mucho más completo y facilitar el proceso que conlleva este informe. Lo primero que se debe tomar en cuenta para hacer estas mejorías o actualizaciones el sistema, es contar con la aprobación de los encargados de la evaluación del informe PRIDE, sin su consentimiento no sería posible hacer la mejora ya que ellos exigen un formato específico del informe y un seguimiento al proceso establecido para la evaluación del mismo.

Las mejoras que se identificaron para el sistema son las siguientes:

- El volumen de información que puede alcanzar este informe puede variar mucho, la cantidad va desde unos cuantos documentos, hasta cajas llenas con documentos para validar las actualizaciones o cursos. De esta manera sería bueno contar con espacios para subir los documentos que sean necesarios para la evaluación del informe.
- Otra posible mejora al sistema sería el manejo de otro tipo usuario para el evaluador, de esta manera el sistema podría generar el informe y se evaluaría en el mismo, esto ahorraría tiempo al proceso del informe PRIDE.
- La última mejora que se ha observado después de analizar el proceso del informe es la obtención de información a partir de otros sistemas existentes, la información acerca de las materias impartidas y los exámenes extraordinarios ya se encuentra almacenada por otros sistemas, y por lo tanto se podría obtener esa información de los demás sistemas, dejando así que los datos que se tengan en el informe sean los correctos, y al mismo tiempo ahorrar tiempo en el llenado del informe.

4.6 Manual del usuario

4.6.1 Opciones generales

En la figura 4-1 se encuentra la pantalla principal del Sistema de Informe PRIDE, cuenta con:

- Un mensaje de bienvenida con una breve descripción.
- Un formulario de acceso, en el cual hay que proporcionar Usuario y Contraseña previamente registrados.
- Si no se ha registrado cuenta con la opción de registro.



Figura 4-1 Página Inicio del Sistema de Informe PRIDE

4.6.1.1 Registro al sistema

En caso de no encontrarse registrado en el sistema, deberá dar clic en la opción “Registrarse”, para que se muestre el formulario de registro al sistema.



Figura 4-2 Opción de registro

El formulario para registrarse en sistema aparece en la Figura 4-3, deberá llenarse para posteriormente confirmar el registro mediante el correo personal.

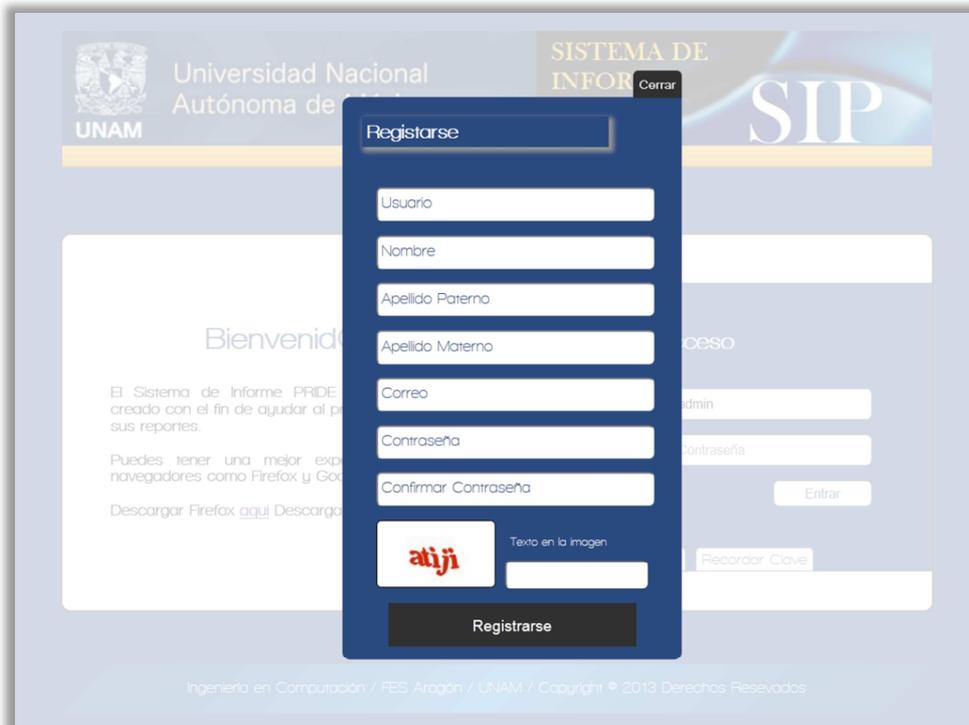


Figura 4-3 Formulario de registro

Después de llenar el formulario, recibirá un correo en la dirección proporcionada al sistema, usted deberá abrir su correo y seguir las instrucciones para confirmar su registro.

Después de ingresar a la dirección que se indicó en el correo, se observará una notificación que indicará que el registro en el sistema se ha hecho correctamente. En caso de que no se haya registrado bien el usuario, mostrará una notificación de error, en este caso se deberá comunicar con los administradores del sistema.

4.6.1.2 Recordar contraseña

En caso de no recordar la contraseña, deberá dar clic en “Recordar Clave”, se mostrará el formulario para crear una nueva contraseña para acceder al sistema.

UNAM Universidad Nacional Autónoma de México

SISTEMA DE INFORME PRIDE SIP

Bienvenid@

El Sistema de Informe PRIDE es un sistema creado con el fin de ayudar al profesor a realizar sus reportes.

Puedes tener una mejor experiencia usando navegadores como Firefox y Google Chrome.

Descargar Firefox [aquí](#) Descargar Chrome [aquí](#)

Acceso

Usuario admin

Contraseña Contraseña

Entrar

Registrarse Recordar Clave

Ingeniería en Computación / FES Aragón / UNAM / Copyright © 2013 Derechos Reservados

Figura 4-4 Recordar Clave

Después de llenar los datos del formulario para recordar contraseña, recibirá en su correo la nueva contraseña.

4.6.1.3 Iniciar sesión

Para iniciar sesión dentro del sistema deberá ingresar el usuario con el que se registró al sistema, además de su contraseña actual.

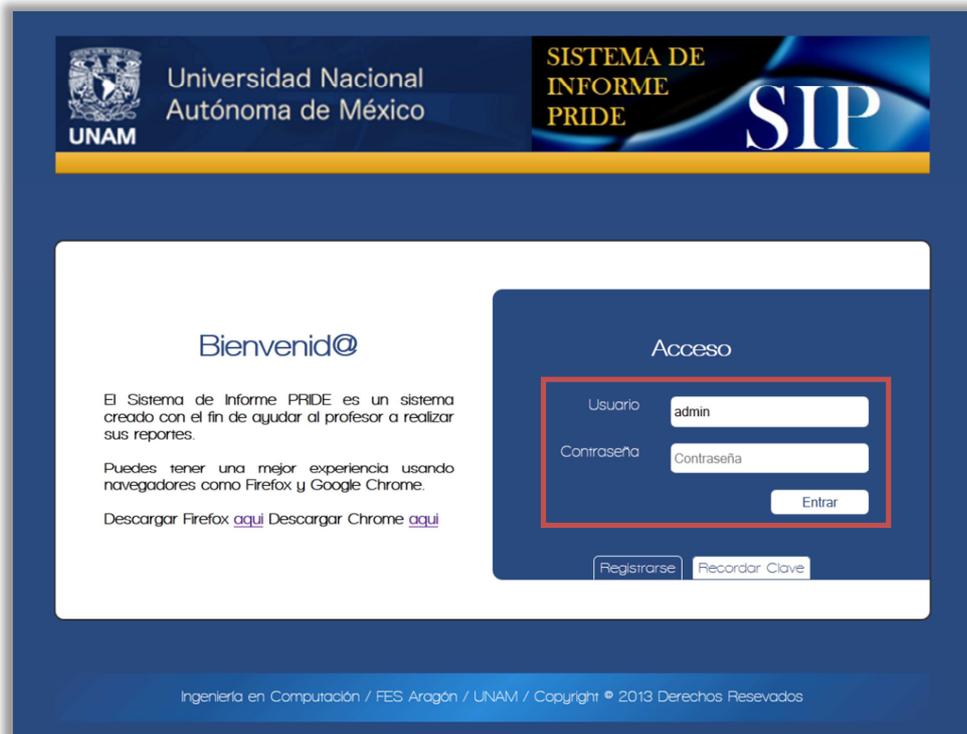


Figura 4-5 Inicio de sesión

En caso de que la información ingresada no concuerde con la almacenada en el sistema se mostrará una notificación de error.



Figura 4-6 Error inicio de sesión

Si la información introducida fue la correcta, se iniciará la sesión en el sistema.



Figura 4-7 Inicio de sesión correcto

4.6.1.4 Perfil

Nos encontramos en la sección de Perfil. Aquí podemos realizar dos acciones, Cambiar clave y Editar perfil.



Figura 4-8 Perfil

4.6.1.4.1 Cambiar clave

En este apartado podemos realizar un cambio de clave de acceso, la cual tiene un formato:

- Contraseña Actual.
- Nueva Contraseña.
- Repetir Contraseña.

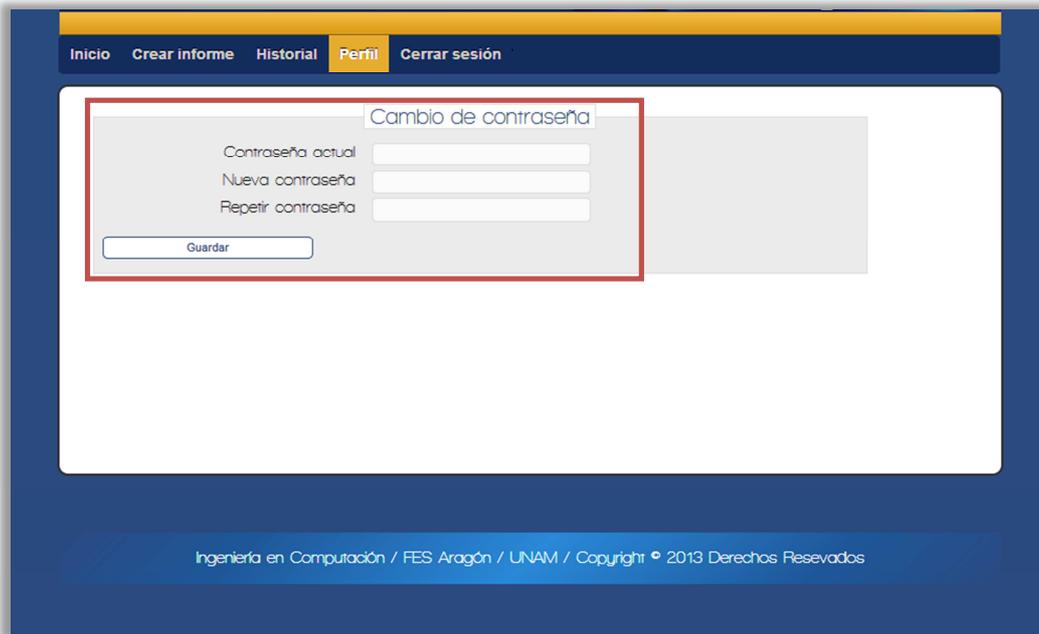


Figura 4-9 Cambiar clave

4.6.1.4.2 Editar perfil

En este apartado podemos realizar cambios nuestro perfil, los datos que podemos modificar son:

- Nombre.
- Apellido Paterno.
- Apellido Materno.
- Correo.
- Nivel Actual.
- Fecha de Nivel Actual.

Figura 4-10 Editar perfil

4.6.1.5 Cerrar sesión

En este apartado al dar clic nos sacara del sistema, es decir cerrará nuestra sesión, y nos regresara a la página donde se realiza la autenticación.

Figura 4-11 Cerrar sesión

4.6.2 Opciones del académico

Una vez que nos hayamos autenticado en el Sistema de Informe PRIDE, nos mostrará esta página con el siguiente mensaje.

El proceso de evaluación del informe PRIDE analiza las actividades realizadas por el académico, así como sus logros y habilidades, los puntos a evaluar por este por el programa son:

1. Formación académica y trayectoria académica y/o profesional
2. Labores docentes y de formación de recursos humanos
3. Productividad académica
4. Difusión, extensión y servicios a la comunidad
5. Participación institucional

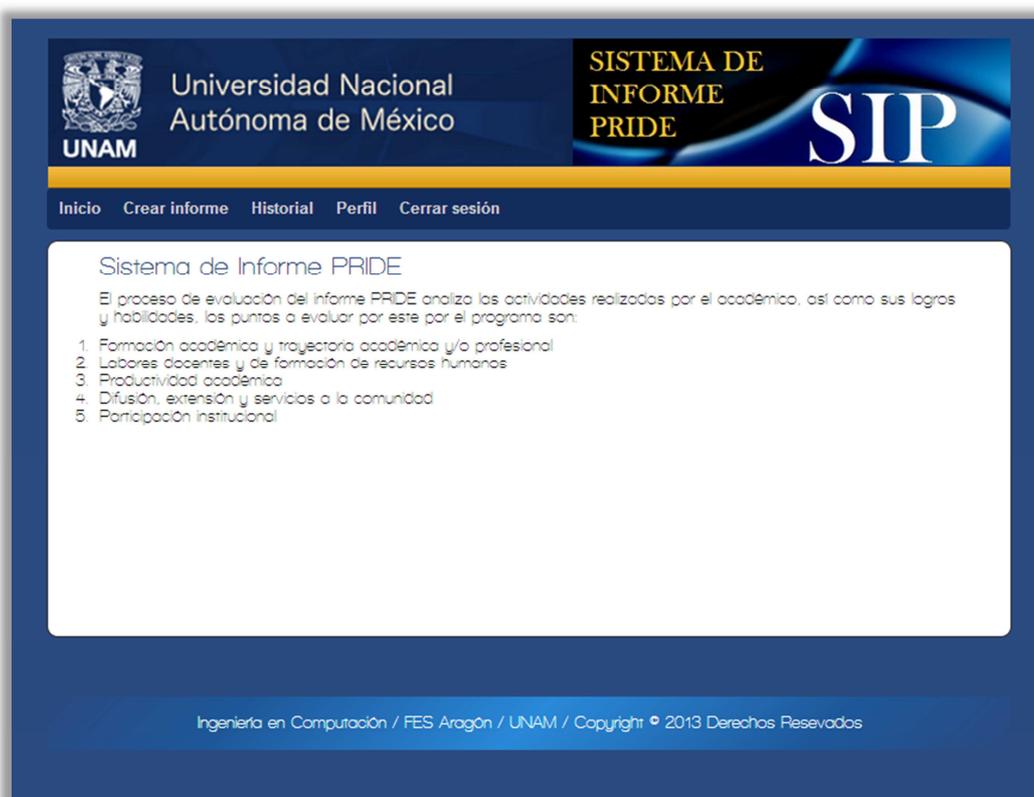


Figura 4-12 Página de inicio de sistema de informe PRIDE

4.6.2.1 Crear informe

Nos encontramos en la sección de Creación de Informes. Esta sección cuenta con varias secciones:

- Formación y Trayectoria.
- Labor Docente.
- Productividad Académica.
- Difusión.

- Participación Institucional.
- Guardar.

A. Formación Académica y Trayectoria Académica y/o Profesional.

A1. Nivel Académico

En este apartado ingresaremos los datos Título/Grado con el siguiente formato:

- Título/Grado.
- Institución.
- Fecha.
- Observaciones.

Figura 4-13 Nivel académico: título o grado

En este mismo apartado, encontraremos la sección Curso de Actualización, donde se muestra el siguiente formato.

- Curso.
- Institución.
- Fecha.
- Observaciones.

Figura 4-14 Nivel académico: curso

A2. Trayectoria Académica y/o Profesional.

En este apartado se registrarán datos como:

- Premios y distinciones académicas (beca, menciones honoríficas, cátedras especiales).
- Invitaciones académicas (conferencias magistrales, cursos especiales, mesas especiales en congresos y simposios, comités editoriales de publicaciones reconocidas nacional e internacionalmente, dictaminación o arbitraje de artículos o libros, exposiciones, obra artística realizada para lugares públicos o museos).
- Participación en comités de evaluación (arbitraje, SIN, SNC, CONACYT, comités para el otorgamiento de premios o distinciones, otros).
- Experiencia profesional (cargos y puestos desempeñados en el medio profesional).
- Reconocimiento a su trayectoria profesional (invitaciones, premios y distinciones).
- Participación en organizaciones profesionales.
- Programas institucionales de servicio- Coordinación de áreas académicas.
- Elaboración de proyectos de investigación, regionales nacionales e internacionales.

En este apartado ingresaremos los datos Actividad (Trayectoria Académica) con el siguiente formato:

- Actividad.
- Institución.
- Fecha.
- Observaciones.

Formación y Trayectoria | Labor Docente | Productividad Académica | Difusión | Participación Institucional

A. Formación Académica y Trayectoria Académica y/o Profesional

▶ A.1. Nivel Académico

▼ A.2. Trayectoria Académica y/o Profesional

- Premios y distinciones académicas (beca, menciones honoríficas, cátedras especiales).
- Invitaciones académicas (conferencias magistrales, cursos especiales, mesas especiales en congresos y simposios, comités editoriales de publicaciones reconocidas nacional e internacionalmente, dictaminación o arbitraje de artículos o libros, exposiciones, obra artística realizada para lugares públicos o museos).
- Participación en comités de evaluación (arbitraje, SIN, SNC, CONACYT, comités para el otorgamiento de premios o distinciones, otros).
- Experiencia profesional (cargos y puestos desempeñados en el medio profesional).
- Reconocimiento a su trayectoria profesional (invitaciones, premios y distinciones).
- Participación en organizaciones profesionales.
- Programas institucionales de servicio- Coordinación de áreas académicas.
- Elaboración de proyectos de investigación, regionales nacionales e internacionales.

En caso de renovación serán considerados únicamente los actividades de los últimos 3 años

Trayectoria Académica

Actividad ✖ Cancelar

Información de la actividad

Actividad

Institución

Fecha

Observaciones

Trayectoria Profesional

Actividad ⊕ Agregar

Figura 4-15 Trayectoria académica

En el mismo apartado ingresaremos los datos Actividad (Trayectoria Profesional) con el siguiente formato:

- Actividad.
- Institución.
- Fecha.
- Observaciones.

A. Formación Académica y Trayectoria Académica y/o Profesional

▶ A.1. Nivel Académico

▼ A.2. Trayectoria Académica y/o Profesional

- Premios y distinciones académicas (beca, menciones honoríficas, cátedras especiales).
- Invitaciones académicas (conferencias magistrales, cursos especiales, mesas especiales en congresos y simposios, comités editoriales de publicaciones reconocidas nacional e internacionalmente, dictaminación o arbitraje de artículos o libros, exposiciones, obra artística realizada para lugares públicos o museos).
- Participación en comités de evaluación (arbitraje, SIN, SNC, CONACYT, comités para el otorgamiento de premios o distinciones, otros).
- Experiencia profesional (cargos y puestos desempeñados en el medio profesional).
- Reconocimiento a su trayectoria profesional (invitaciones, premios y distinciones).
- Participación en organizaciones profesionales.
- Programas institucionales de servicio- Coordinación de áreas académicas.
- Elaboración de proyectos de investigación, regionales nacionales e internacionales.

En caso de renovación serán consideradas únicamente las actividades de los últimos 3 años

Trayectoria Académica

Actividad ➕ Agregar

Trayectoria Profesional

Actividad ✖ Cancelar

Información de la actividad

Actividad

Institución

Fecha

Observaciones

Figura 4-16 Trayectoria profesional

B. Labores Docentes y de Formación de Recursos Humanos.

B1. Datos Generales de la Labor Docente del Académico.

En este apartado ingresaremos datos como:

- Número de asignaturas y grupos.
- Número de alumnos por grupo.
- Número de horas clase por semana.
- Asistencia y puntualidad en la impartición.
- Programa (s) de asignatura (s) (presentación de bibliografías).
- Entrega puntual de actas de examen

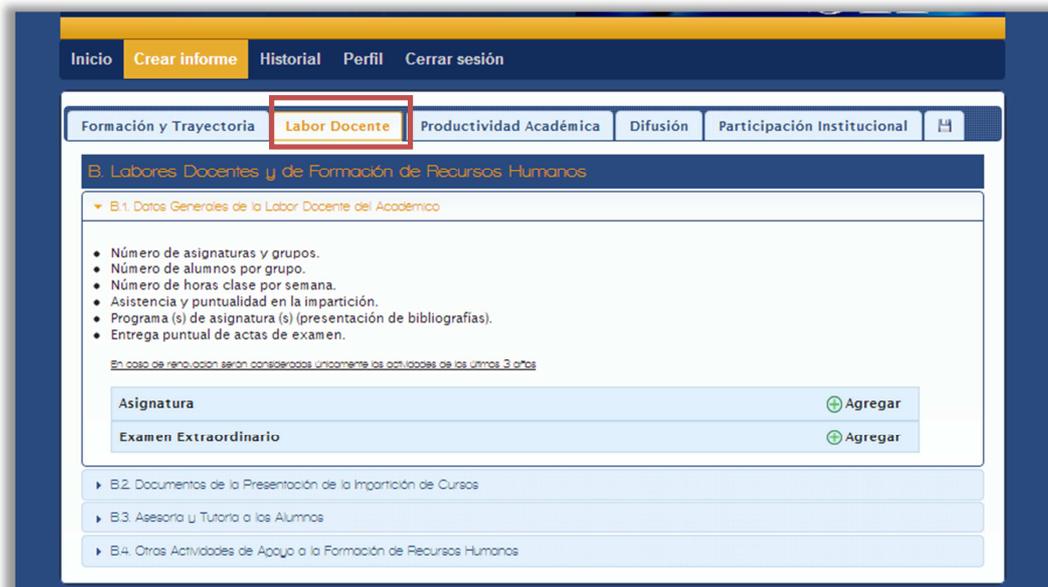


Figura 4-17 Labor docente

En este apartado ingresaremos la Información de la Asignatura con el siguiente formato:

- Asignatura.
- Grupo.
- Número de Alumnos.
- Hrs. De Clase por Semana.
- % Asistencia.
- Entrega Puntual de Actas.

Formación y Trayectoria | **Labor Docente** | Productividad Académica | Difusión | Participación Institucional

B. Labores Docentes y de Formación de Recursos Humanos

▼ B.1. Datos Generales de la Labor Docente del Académico

- Número de asignaturas y grupos.
- Número de alumnos por grupo.
- Número de horas clase por semana.
- Asistencia y puntualidad en la impartición.
- Programa (s) de asignatura (s) (presentación de bibliografías).
- Entrega puntual de actas de examen.

En caso de renovación serán consideradas únicamente las actividades de los últimos 3 años

Asignatura ✖ Cancelar

Información de la asignatura

Asignatura

Grupo

Numero de Alumnos

Hrs de clase por semana

% Asistencia

Entrega Puntual de actas

Examen Extraordinario ⊕ Agregar

► B.2. Documentos de la Presentación de la Impartición de Cursos

► B.3. Asesoría y Tutoría a los Alumnos

► B.4. Otras Actividades de Apoyo a la Formación de Recursos Humanos

Figura 4-18 Asignaturas

En este apartado ingresaremos los datos de Examen Extraordinario (Información de la Asignatura) con el siguiente formato:

- Examen Extraordinario.
- Semestre.
- Número de Alumnos.
- Entrega Puntual de Actas.

Figura 4-19 Exámenes extraordinarios

B2. Documentos de la Presentación de la Impartición del Curso.

En este apartado se puede subir el archivo de impartición del curso, el cual debe cumplir los siguientes puntos:

- Su idea de la enseñanza y sus propósitos (Descritas en un máximo de 3 cuartillas con los siguientes rubros).
- Sus procedimientos o métodos de enseñanza (clase magistral, seminarios, prácticas, tareas, ejercicios, prácticas de campo, visitas, lecturas guiadas, ensayos, investigaciones, conferencias especiales, aprendizaje interactivo en software, otros) y la razones por las que los ha elegido.
- La forma como planea su(s) curso(s) y su enseñanza.
- La forma como su(s) programa(s) de materia se relaciona con el conjunto del plan de estudios.
- El procedimiento que utiliza para evaluar el aprendizaje de sus alumnos y las razones por las cuales dicho procedimiento contribuye al aprendizaje.
- Otras actividades académicas vinculadas a la docencia que el profesor o el investigador considere meritorias y dignas de ser tomadas en cuenta y evaluadas por las comisiones evaluadoras.

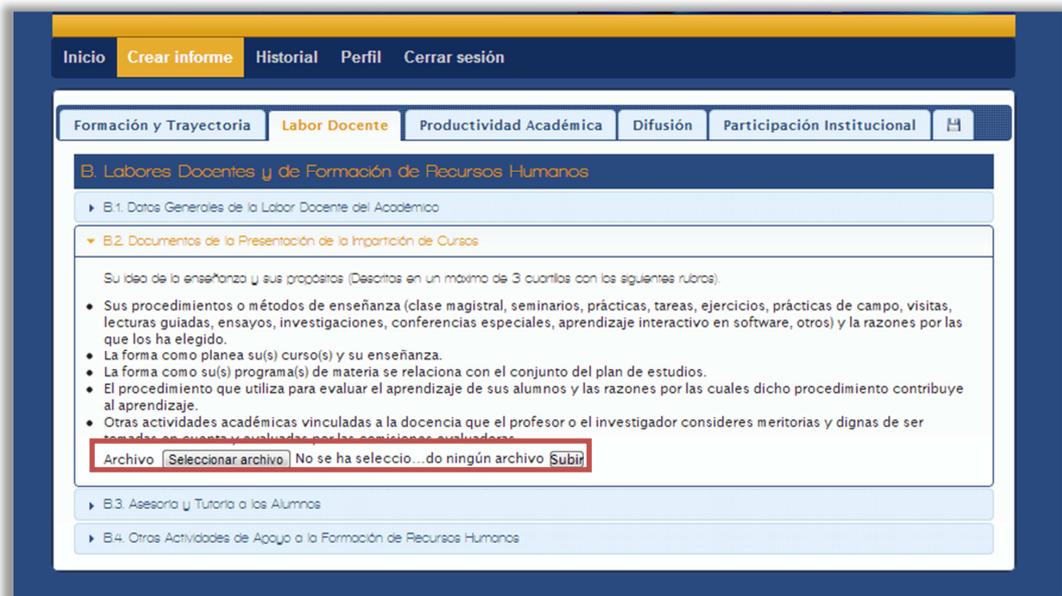


Figura 4-20 Documentos de impartición del curso

B3. Asesoría y Tutoría a los Alumnos.

En este apartado se ingresaran datos como:

- Asesorías Académicas (especificar los tipos).
- Dirección de tesis aprobadas en exámenes profesionales o de grado (constancias institucionales).
- Dirección de tesis en proceso. - Tutorías (alumnos de posgrado, becarios, prácticas profesionales, grupos especiales, programas institucionales).
- Participación Activa en jurados de exámenes profesionales y de grado.
- Asesoría a estudiantes que participan en congresos, publicaciones, conferencias, simposios, proyectos tecnológicos, artísticos, de diseño o de comunicación visual

En este apartado ingresaremos los datos Actividad con el siguiente formato:

- Actividad.
- Institución.
- Fecha.
- Observaciones.

Figura 4-21 Asesorías y tutorías a alumnos

B4. Otras Actividades de Apoyo a la Formación de Recursos Humanos.

En este apartado se ingresan datos como:

- Planeación, diseño, coordinación o realización de actividades relacionadas con planes y programas de estudio para la formación o actualización de recursos humanos.

En este apartado ingresaremos los datos Tema/Asignatura con el siguiente formato:

- Actividad.
- Institución.
- Fecha.
- Observaciones.

The screenshot shows a web application interface for 'Labor Docente'. At the top, there is a navigation bar with 'Inicio', 'Crear informe', 'Historial', 'Perfil', and 'Cerrar sesión'. Below this, a sub-menu for 'Labor Docente' is expanded, showing options like 'Formación y Trayectoria', 'Labor Docente', 'Productividad Académica', 'Difusión', and 'Participación Institucional'. The 'Labor Docente' section is selected, displaying a description of the activity and a form titled 'Información de la actividad' with fields for 'Actividad', 'Institución', 'Fecha', and 'Observaciones', and a 'Guardar' button.

Figura 4-22 Apoyo a la formación de recursos humanos

C. Productividad Académica.

C1. Productividad Científica, Humanística, Artística o Tecnológica.

En este apartado ingresaremos datos como:

- Publicaciones (artículos de investigación arbitrados, libros como autor, capítulos en libros, artículos en memorias con arbitraje, publicaciones electrónicas, otros artículos publicados).
- Publicaciones como editor (coordinación, compilación, antologías, boletines). - Obras artísticas, proyectos de diseño y de comunicación visual.
- Productos tecnológicos (patente, estudios para la industria, prototipos terminados y probados, "software", otros).
- Productos de infraestructura académica (promoción, organización y desarrollo de bibliotecas, laboratorios, talleres, plantas piloto, clínicas, granjas, museos, otros).
- Producción de videos y programas de radio y televisión.
- Trabajos presentados en congresos, seminarios y talleres académicos.

En este apartado ingresaremos los datos Actividad/Trabajo con el siguiente formato:

- Actividad.

- Institución.
- Fecha.
- Observaciones.

The screenshot displays a web application interface for recording academic productivity. At the top, there is a navigation bar with options: Inicio, Crear informe, Historial, Perfil, and Cerrar sesión. Below this, a secondary navigation bar contains tabs for Formación y Trayectoria, Labor Docente, Productividad Académica, Difusión, and Participación Institucional. The 'Productividad Académica' section is expanded to show 'C.1. Productividad Científica, Humanística, Artística O Tecnológica'. A list of activity types is provided, including publications, technological products, and academic presentations. A form titled 'Actividad / Trabajo' is highlighted with a red box, containing fields for 'Actividad', 'Institución', 'Fecha', and 'Observaciones', along with 'Guardar' and 'Cancelar' buttons.

Figura 4-23 Productividad científica, humanística, artística o tecnológica

C2. Productividad en la Docencia.

En este apartado ingresaremos datos como:

- Materiales Didácticos.
- Libros de Texto o de apoyo a la enseñanza.
- Capítulos de libros de texto.
- Compilaciones.
- Antologías.
- Ensayos.
- Programas innovadores de enseñanza.
- Presentación de trabajos en reuniones relacionadas con la enseñanza de la disciplina.
- Traducción de materiales especializados que sirvan de apoyo a los programas docentes.

En este apartado ingresaremos los datos Actividad/Trabajo con el siguiente formato:

- Actividad.
- Institución.
- Fecha.
- Observaciones.

Figura 4-24 Productividad de la docencia

D. Difusión, Extensión y Servicio a la Comunidad.

D1. Difusión, Extensión y Servicio a la Comunidad.

En este apartado ingresaremos datos como:

- Libros de divulgación.
- Artículos de divulgación.
- Conferencias.
- Organización de eventos (científicos, tecnológicos o artísticos nacionales e internacionales).
- Exposiciones.

- Servicios a la comunidad (clínicas médicas, odontológicas, psicológicas, construcción de obras, consultorías a instituciones externas, proyectos determinados, servicios a la industria o al sector público).
- Participación en medios de comunicación (prensa, radio, televisión).

En este apartado ingresaremos los datos Actividad/Trabajo con el siguiente formato:

- Actividad.
- Institución.
- Fecha.
- Observaciones.

The screenshot shows a web application interface. At the top, there is a navigation bar with the following items: Inicio, **Crear informe**, Historial, Perfil, and Cerrar sesión. Below this, there is a secondary navigation bar with tabs: Formación y Trayectoria, Labor Docente, Productividad Académica, **Difusión**, and Participación Institucional. The main content area is titled 'D. Difusión, Extensión y Servicios a la Comunidad'. It contains a list of activity categories: Libros de divulgación, Artículos de divulgación, Conferencias, Organización de eventos (científicos, tecnológicos o artísticos nacionales e internacionales), Exposiciones, Servicios a la comunidad (clínicas médicas, odontológicas, psicológicas, construcción de obras, consultorías a instituciones externas, proyectos determinados, servicios a la industria o al sector público), and Participación en medios de comunicación (prensa, radio, televisión). Below the list, there is a note: 'En caso de renovación serán consideradas únicamente las actividades de los últimos 3 años'. A modal window titled 'Actividad / Trabajo' is open, containing a form with the following fields: Actividad, Institución, Fecha, and Observaciones. There are also 'Cancelar' and 'Guardar' buttons in the modal.

Figura 4-25 Difusión, extensión y servicio a la comunidad

E. Participación Institucional.

E1. Participación Institucional.

En este apartado ingresaremos datos como:

- Cuerpos colegiados.
- Comités editoriales.

- Comités de evaluación de programas institucionales de apoyo académico (PRIDE, PAPIIT, PITID, entre otros).
- Programas institucionales de servicio.
- Programas institucionales de desarrollo de infraestructura académica.
- Diseño y revisión de planes y programas de estudio.
- Actividades de dirección, organización o coordinación académica institucional.
- Participación activa como jurado calificador en los concursos de oposición.

En este apartado ingresaremos los datos Actividad con el siguiente formato:

- Actividad.
- Institución.
- Fecha.
- Observaciones.

The screenshot shows the UNAM SIP web interface. At the top, there are logos for UNAM, PRIDE, and SIP. A navigation bar includes 'Inicio', 'Crear informe', 'Historial', 'Perfil', and 'Cerrar sesión'. Below this, a secondary menu has 'Formación y Trayectoria', 'Labor Docente', 'Productividad Académica', 'Difusión', and 'Participación Institucional' (which is highlighted with a red box). The main content area is titled 'E. Participación Institucional' and contains a list of activity types: 'Cuerpos colegiados', 'Comités editoriales', 'Comités de evaluación de programas institucionales de apoyo académico (PRIDE, PAPIIT, PITID, entre otros)', 'Programas institucionales de servicio', 'Programas institucionales de desarrollo de infraestructura académica', 'Diseño y revisión de planes y programas de estudio', 'Actividades de dirección, organización o coordinación académica institucional', and 'Participación activa como jurado calificador en los concursos de oposición'. Below the list is a note: 'En caso de retrocesión serán considerados únicamente los actividades de los últimos 3 años'. A form titled 'Actividad' is displayed, featuring a 'Cancelar' button and a sub-section 'Información de la actividad' (highlighted with a red box) containing input fields for 'Actividad', 'Institución', 'Fecha', and 'Observaciones', along with a 'Guardar' button.

Figura 4-26 Participación institucional

Terminar Informe

En este apartado terminaremos el informe, es decir guardaremos todos los datos ingresados con anterioridad.

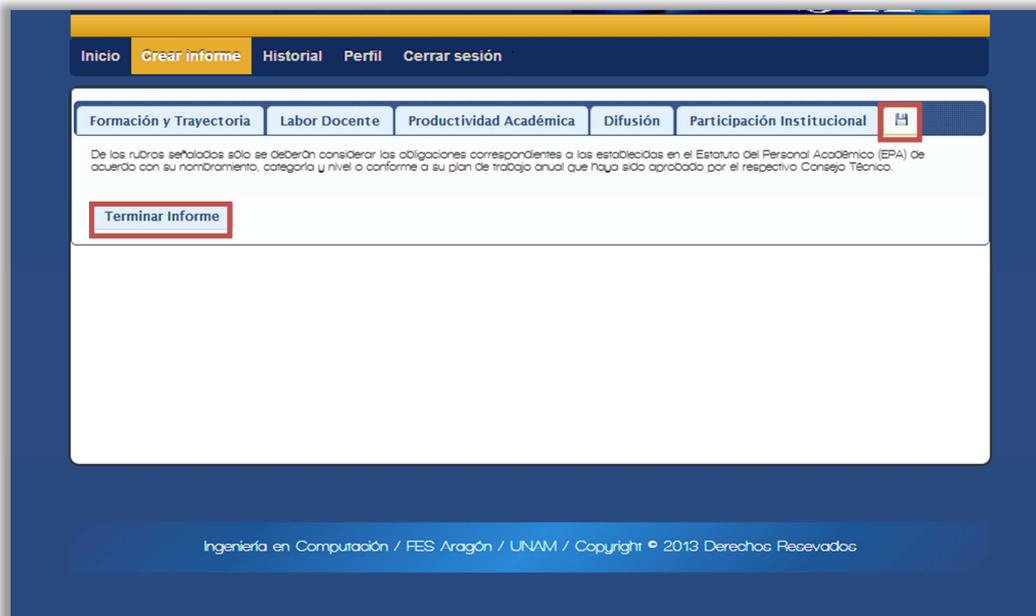


Figura 4-27 Terminar informe

4.6.2.2 Historial

Nos encontramos en la sección de Historial. En esta sección encontraremos nuestros historiales ya sea creado recientemente o con anterioridad.

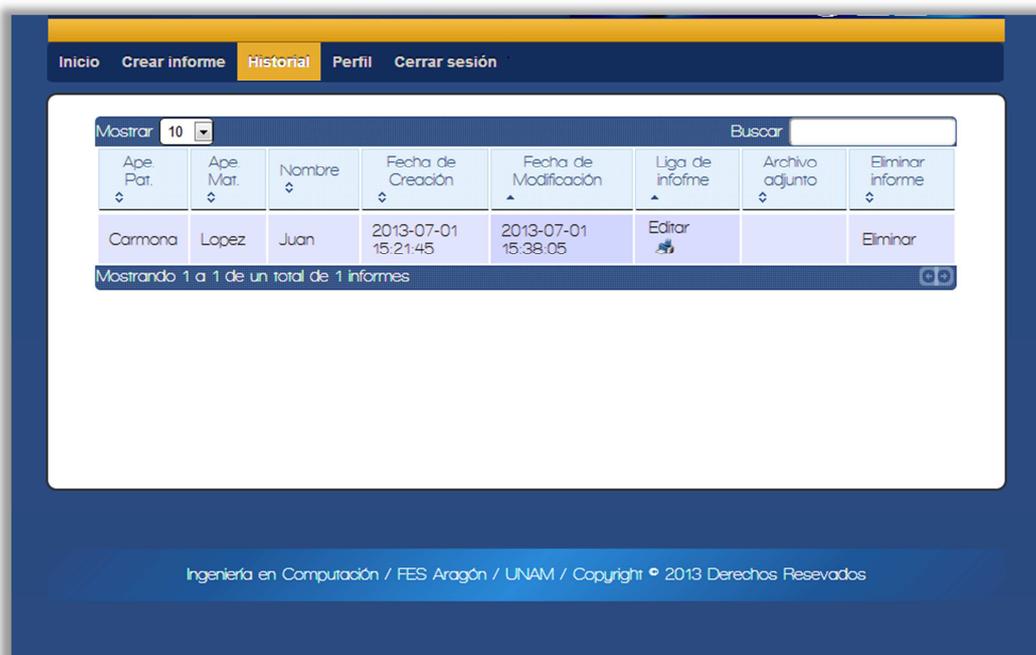


Figura 4-28 Historial

También podemos editar nuestros Informes que encontremos en el historial, o eliminar.

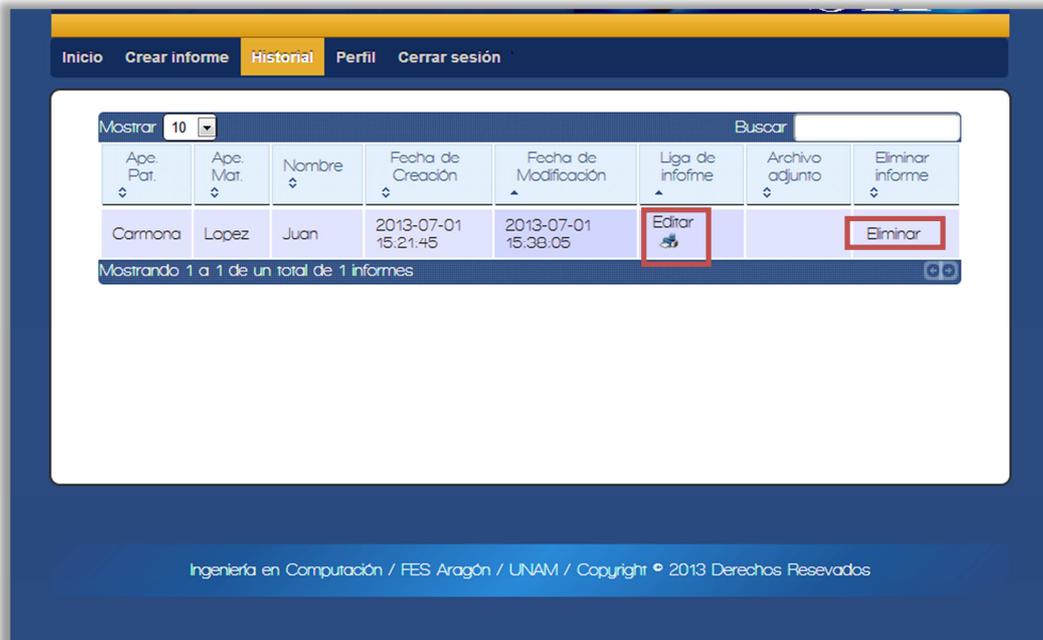


Figura 4-29 Lista de informes

4.6.3 Opciones del administrador

Después de haberse autenticado como administrador se muestra la página de la Figura 4-23, a diferencia del académico no aparecerá la opción de crear un nuevo informe, pero aparecerá una nueva opción “Control” que servirá para administrar los usuarios, facultades y carreras del sistema.

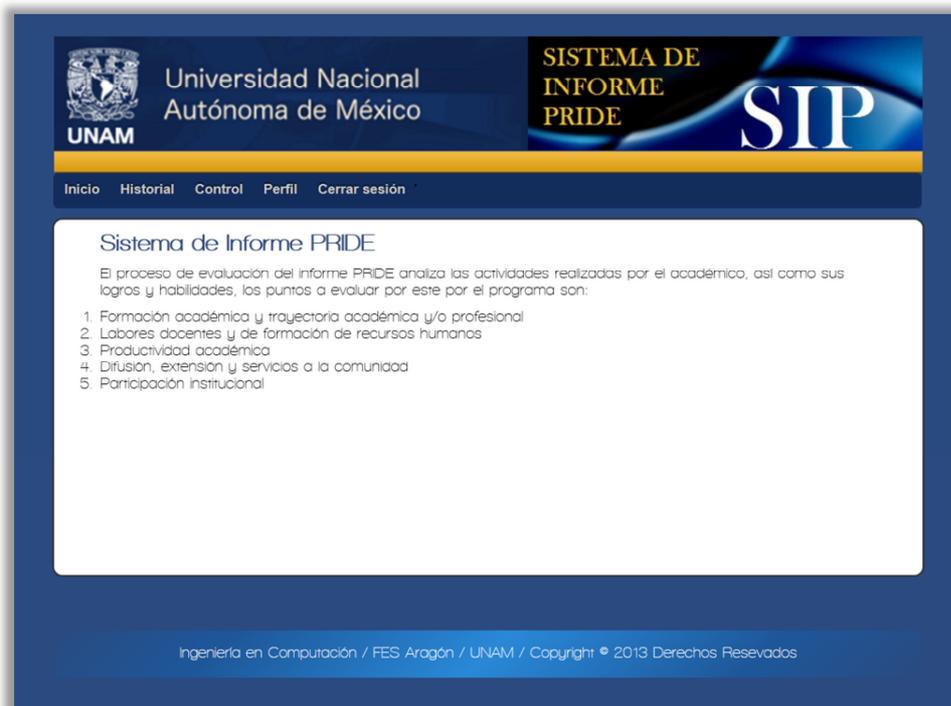


Figura 4-30 Menú del administrador

4.6.3.1 Historial

En esta parte del sistema el administrador podrá ver los informes creados por los académicos y realizar búsquedas. En un principio será mostrada una notificación que informara que no hay informes creados.



Figura 4-31 Notificación no hay informes

4.6.3.2 Control

Al dar clic en “Control” del menú del sistema se mostraran 3 opciones:

- Dar de alta usuarios
- Modificar usuarios
- Administrar facultades y carreras

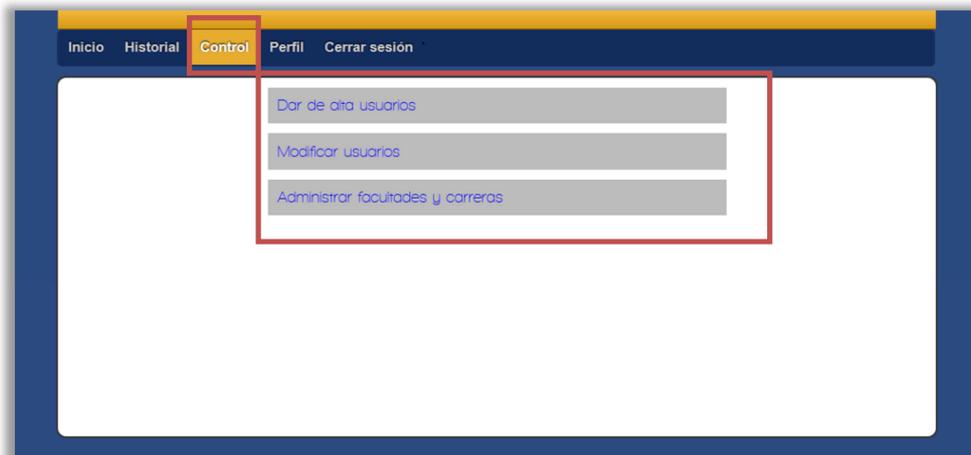


Figura 4-32 Opciones de control

4.6.3.2.1 Activar usuarios

Se muestra la lista de los usuarios que se hayan registrado y no se encuentren activados, el administrador podrá activar la cuenta, para que después de activarlos el usuario podrá iniciar sesión.



Figura 4-33 Activación de usuarios

Para activar un usuario se deberá dar clic en la primera columna del usuario, en ese momento aparecerá una notificación donde se preguntara si se desea activar al usuario.



Figura 4-34 Confirmación de activación de usuario

Si se selecciona activar y la activación fue exitosa aparecerá una notificación como la siguiente.



Figura 4-35 Mensaje de activación de usuario

4.6.3.2.1 Modificar usuarios

Se muestra una lista con todos los usuarios del sistema (a excepción del administrador), donde se podrá editar la información del usuario o eliminarlo (bloquearlo).



Figura 4-36 Modificar usuarios

Si se da clic en "Editar", se muestra un formulario con la información del usuario que podrá ser editada para posteriormente ser guardada en el sistema.

Nombre	Arcelia
Apellido Paterno	Bernal
Apellido Materno	Diaz
Correo	arceliabernal@hotmail.com
Facultad	(411) F.E.S. ARAGON
Carrera	(110) INGENIERIA EN COMPUTACION
Nivel actual	1
Fecha de nivel actual	2013-09-16

Guardar

Figura 4-37 Formulario de modificación de usuario

Para actualizar la información del usuario se deberá dar clic en el botón “Guardar”, si el proceso de actualización fue exitoso se mostrara el siguiente mensaje.

Los datos fueron actualizados

Figura 4-38 Mensaje de modificación de usuario

4.6.3.2.3 Administrar facultades y carreras

Esta sección está dividida en 2 partes, las cuales se muestran visualmente en dos pestañas.

Facultades: Se muestra la lista de facultades disponibles para editarlas y para eliminarlas.

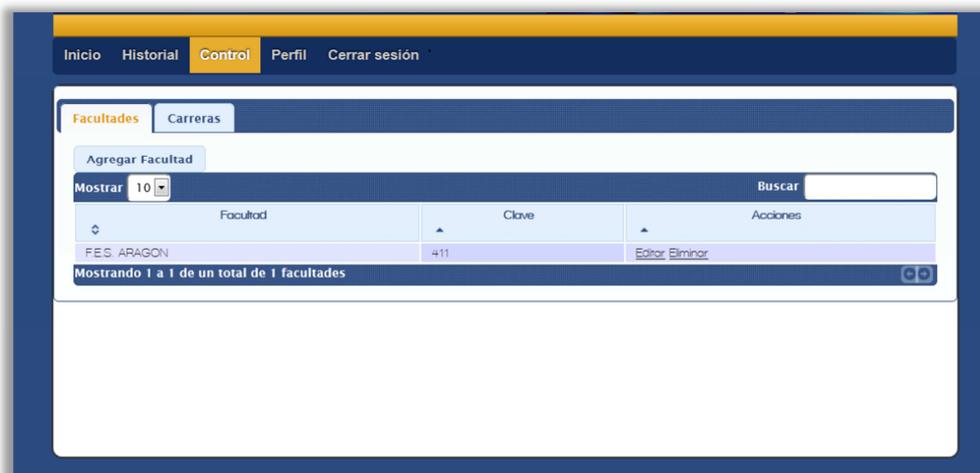


Figura 4-39 Administración de facultades y carreras

En caso de querer agregar una nueva facultad, deberá darse clic e el botón “Agregar facultad”, en ese momento aparecerá una ventana con la información necesaria para crear una facultad.



Figura 4-40 Formulario para agregar facultad

Después de llenar los campos, se deberá dar clic en “Guardar”, si la información ingresada fue almacenada correctamente se mostrara el siguiente mensaje.



Figura 4-41 Mensaje de agregar facultad

Si lo que deseas es editar una facultad, deberás dar clic en “Editar”, aparecerá una ventana donde podrá editarse la información relacionada con la facultad.



Figura 4-42 Formulario para editar facultad

Después de actualizar la información de la facultad se deberá dar clic en “Guardar”, si la información se actualiza correctamente se muestra la siguiente notificación.



Figura 4-43 Mensaje de editar facultad

Si lo que desea hacer es eliminar una facultad, deberá dar clic en “Eliminar”, después de eso le aparecerá una notificación para confirmar que desea eliminar la facultad.

Si se confirma la eliminación de la facultad, y se elimina correctamente del sistema será mostrada una notificación como la siguiente.

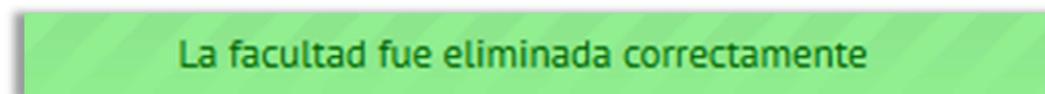


Figura 4-44 Mensaje de eliminar facultad

Carreras: Se listan las carreras registradas en el sistema, donde se podrán agregar, modificar y eliminar.

Carrera	Clave	Facultad	Acciones
INGENIERIA CIVIL	107	(+11) F.E.S. ARAGON	Editar Eliminar
INGENIERIA ELECTRICA Y ELECTRONICA	109	(+11) F.E.S. ARAGON	Editar Eliminar
INGENIERIA EN COMPUTACION	110	(+11) F.E.S. ARAGON	Editar Eliminar
INGENIERIA INDUSTRIAL	114	(+11) F.E.S. ARAGON	Editar Eliminar
INGENIERIA MECANICA	115	(+11) F.E.S. ARAGON	Editar Eliminar
INGENIERIA MECANICA ELECTRICA	116	(+11) F.E.S. ARAGON	Editar Eliminar

Figura 4-45 Lista de carreras

En caso de querer agregar una carrera se deberá dar clic en “Agregar Carrera”, después de eso aparecerá una ventana con el formulario de la información necesaria para crear una nueva carrera.

Figura 4-46 Formulario para agregar carera

Después de ingresar la información se dará clic en “Guardar”, si la carrera fue creada exitosamente se muestra la notificación.

La carrera fue guardada correctamente

Figura 4-47 Mensaje de agregar carrera

Para editar una carrera se deberá dar clic en la opción “Editar”, aparecerá en una ventana el formulario de edición de la carrera.

Carrera

Información de facultad

Carrera: INGENIERIA CIVIL

Clave: 107

Facultad: (411) F.E.S. ARAGON

Guardar

Figura 4-48 Formulario para editar carrera

Después de actualizar la información de la carrera se dará clic en “Guardar”, si la actualización de la información fue exitosa se muestra una notificación como la siguiente.



Figura 4-49 Mensaje de editar carrera

Si lo que desea hacer es eliminar una carrera, deberá dar clic en “Eliminar”, después de eso le aparecerá una notificación para confirmar que desea eliminar la carrera.

Si se confirma la eliminación de la carrera, y se elimina correctamente del sistema será mostrada una notificación como la siguiente.



Figura 4-50 Mensaje de eliminar carrera



“Una vez que salgas de la escuela, sólo lo que hagas por ti mismo dará calidad a tu vida”

Jean Piaget

5 Conclusiones

Durante el tiempo de desarrollo del Sistema de Informe PRIDE (SIP), pude aplicar los conocimientos adquiridos durante mi formación en la Facultad de Estudios Superiores Aragón. Es grato ver terminado un sistema el cual construí con lo aprendido en los 4 años que estude la carrera.

Además de aplicar los conocimientos adquiridos de las materias que curse durante la carrera de ingeniería en computación, aplique conocimientos adquiridos en cursos extracurriculares tomados en el CAE 504 que pertenece a la FES Aragón, cursos tomados en el Telmexhub que es un centro de reunión y esparcimiento tecnológico perteneciente a Telmex, así como conocimientos adquiridos por la curiosidad a través de la lectura e investigación de las tecnologías.

El sistema SIP cumple los objetivos principales por el cual fue desarrollado, los cuales son la creación y administración de los informes PRIDE. El sistema se conecta a una base de datos y garantiza la disponibilidad de la información contenida por el informe PRIDE.

El sistema cuenta con un módulo de seguridad el cual permite que solamente los usuarios poseedores del informe, además del administrador, puedan tener acceso a su información en todo momento, siempre y cuando cuenten con una conexión a internet. El sistema se desarrolló tomando como base la ingeniería de software, lo cual confirma una solides en el diseño del sistema.

El sistema se pone a disposición de los académicos como una herramienta que permita facilitar el manejo de informe PRIDE, se espera que el académico pueda sacar todo el provecho al sistema.

Después de concluir el sistema y observar las ventajas que trae consigo, se recomienda la creación de sistemas para automatizar actividades, como inscripción, tramites y administración de información en general, ya que la creación de estos sistemas conlleva una mejora significativa en la calidad, disponibilidad y facilidad con la que se realiza la actividad.

Bibliografía

Pressman, Roger S. (2006). Ingeniería de software, un enfoque práctico (sexta edición). México: Editorial McGrawHill.

Joyales, Luis. (2013). Big Data - Análisis de grandes volúmenes de datos en organizaciones. México: Editorial Alfaomega.

López, José. (2011). Domine JavaScript (tercera edición). México: Editorial Alfaomega.

Fontela, Carlos. (2010) UML - Modelado de software para profesionales. México: Editorial Alfaomega.

Firtman, Maximiliano. (2010) AJAX - Web 2.0 Con jQuery para profesionales, México: Editorial Alfaomega.

López, José. (2010). Domine PHP Y MYSQL (segunda edición). México: Editorial Alfaomega.

Orós, Juan Carlos. (2012). Guía Práctica De XHTML, Javascript Y CSS. México: Editorial Alfaomega.

Pérez, César. (2008). MySQL - Para Windows Y Linux (segunda edición). México: Editorial Alfaomega.