

**DESARROLLO DE UN SITIO WEB
PARA EL CONTROL DE PROYECTOS
DE FIN DE CARRERA**

Tesis presentada para la obtención del título de
Licenciado en Sistemas Computacionales

DIANA JIMÉNEZ ÁNGELES Y NANCY MARTÍNEZ MONTAÑO

3 de marzo de 2010

Índice general

Justificación	9
Planteamiento del Problema	10
Objetivo	11
Objetivos Específicos	11
Propuesta de Solución	12
1. Marco Referencial	13
1.1. Metodologías	13
1.1.1. Metodología Web	13
1.1.2. Metodología de desarrollo de sistemas	16
1.1.3. Método de Cascada Pura	17
1.1.4. Método Espiral	18
1.1.5. Método de Codificar y Corregir (Code-and-fix)	19
1.1.6. Modelo Incremental	20
1.1.7. Modelo DRA (Desarrollo Rápido de Aplicaciones)	20
1.1.8. Modelo de Proceso Incremental	21
1.1.9. UML	22
1.2. Herramientas de desarrollo de Sistemas para Internet	26
1.2.1. CSS	26
1.2.2. ASP	27
1.2.3. XAJAX	27
1.2.4. XAMPP	28
1.2.5. MySql	29
1.2.6. JOOMLA	30
1.2.7. Base de Datos	30
1.2.8. Bases de datos o "bibliotecas"de información Biológica	32
1.2.9. modelado de los datos	32
1.3. Lenguajes de desarrollo de sistemas	32
1.3.1. PHP	33
1.3.2. Programación Java	34
1.3.3. Programación avanzada en Java	34

1.3.4.	Programación Web	34
1.3.5.	HTML	35
1.3.6.	Java Script	36
1.3.7.	Lenguaje ASP	37
1.3.8.	Lenguaje ASP.NET	38
1.3.9.	Lenguaje JSP	39
1.3.10.	Lenguaje PYTHON	40
1.3.11.	Lenguaje Ruby	40
2.	Trabajos Relacionados	42
2.1.	¿Qué es un repositorio de documentos?	42
2.2.	E-LIS Eprints in Library and Information Science (http://eprints.rclis.org) 43	
2.3.	E-PrintsUCM http://www.ucm.es/eprints	44
2.4.	- TDX: Tesis Doctorals en Xarxa	45
3.	Método	47
3.1.	Metodología empleada en el sistema	47
3.2.	Herramientas empleadas en el sistema	48
3.2.1.	UML	48
3.2.2.	DBDesigner	48
3.2.3.	StarUML	49
3.2.4.	NetBeans	50
3.2.5.	Adobe Fireworks	50
3.2.6.	Dreamweaver	50
3.2.7.	Flash	51
3.2.8.	XAMPP	51
3.2.9.	MySQL	51
3.3.	Lenguajes de programación usados en el sistema	52
3.3.1.	Html	52
3.3.2.	Java	52
3.3.3.	JSP	52
4.	Análisis	53
4.1.	Diagramas de Escenarios	56
4.1.1.	Diagrama de escenario de consulta de tesis en proceso	57
4.1.2.	Diagrama de escenario de consulta de tesis terminada	57
4.1.3.	Diagrama de escenario pre-registro	57
4.1.4.	Diagrama de escenario de administrador modificar/eliminar	58
4.1.5.	Diagrama de escenario de administrador	58
4.1.6.	Diagrama de escenario iniciar sesión	58
4.2.	Diagrama de caso de uso	59
4.2.1.	Diagrama de caso de uso sobre las actividades del alumno	59
4.2.2.	Diagrama de caso de uso sobre las actividades del profesor	59
4.2.3.	Diagrama de caso de uso sobre las actividades generales del alumno	60

4.2.4.	Diagrama de caso de uso sobre el proceso de consulta . . .	61
4.2.5.	Diagrama de caso de uso sobre el proceso de eliminación de información	61
4.2.6.	Diagrama de caso de uso sobre el proceso de modificación del registro	62
4.2.7.	Diagrama de caso de uso sobre el proceso de pre-registro .	63
4.2.8.	Diagrama de caso de uso de un pre-registro inválido	63
4.2.9.	Diagrama de caso de uso de un pre-registro valido	64
4.3.	Diagrama de secuencia	65
4.3.1.	Diagrama de secuencia de pre-registro	65
4.3.2.	Diagrama de secuencia del proceso de iniciar sesión	65
4.3.3.	Diagrama de secuencia del proceso de creación nueva cuenta	66
4.3.4.	Diagrama de secuencia del registro	67
4.3.5.	Diagrama de secuencia de eliminación y modificación . . .	68
4.3.6.	Diagrama de secuencia de consulta	69
4.4.	Diagrama de Colaboración	70
4.4.1.	Diagrama de Colaboración de la consulta	70
4.4.2.	Diagrama de Colaboración del proceso de creación de nueva cuenta	71
4.4.3.	Diagrama de Colaboración del proceso de pre-registro . .	71
4.4.4.	Diagrama de Colaboración del proceso de registro	72
4.4.5.	Diagrama de Colaboración de los procesos de eliminación y modificación de la información	72
4.4.6.	Diagrama de Colaboración del proceso de iniciar sesión .	73
5.	Diseño	74
5.1.	Diagrama de Actividades	74
5.1.1.	Diagrama de Actividades del proceso de pre-registro . . .	74
5.1.2.	Diagrama de Actividades del proceso de registro	75
5.1.3.	Diagrama de Actividades del proceso de creación de cuenta de usuario	76
5.1.4.	Diagrama de Actividades del proceso de modificación . .	77
5.1.5.	Diagrama de Actividades del proceso de eliminación . . .	78
5.1.6.	Diagrama de Actividades del proceso de consulta	79
5.2.	Diagrama de Estado	80
5.3.	Diagramas de Objetos	81
5.3.1.	Diagrama de Objetos de la aplicación	81
5.4.	Diagrama de Clases	82
6.	Desarrollo	84
6.1.	Diseño de la Interfaz de usuario	84
6.2.	Modelado de la base de datos	87
6.3.	Desarrollo de la aplicación Web	91

7. Pruebas	101
7.1. Menú principal	102
7.1.1. Consulta	102
7.1.2. Pre-Registro	105
7.1.3. Cuenta	107
7.1.4. Administrador	114
7.1.5. Contactos	117
Conclusiones	118
Glosario	120
Anexos	124
Bibliografía	159

Índice de figuras

1.1. Diagrama de la metodología web.	14
1.2. Diagrama de clases.	22
2.1. Portal especializado en Biblioteconomía, Documentación y disciplinas relacionadas.	44
2.2. Repositorio institucional de la Universidad Complutense de Madrid.	45
2.3. Repositorio de Tesis Doctorales en Red.	46
3.1. Gráfico del modelo iterativo incremental	48
4.1. Gráfica del tiempo en que tarda en validarse un tema de tesis	54
4.2. Divulgación de los temas de tesis propuestos por profesores	55
4.3. Porcentajes de alumnos titulados por tesis	55
4.4. Porcentaje de alumnos que registran su tesis	56
4.5. Caso de uso sobre las actividades del alumno.	59
4.6. Caso de uso sobre las actividades del profesor.	60
4.7. Caso de uso de las actividades generales del alumno.	60
4.8. Caso de uso sobre el proceso de consulta.	61
4.9. Caso de uso del proceso de eliminación de información.	62
4.10. Caso de uso sobre el proceso de modificación del registro.	62
4.11. Caso de uso sobre el proceso del pre-registro.	63
4.12. Caso de uso sobre proceso involucrado cuando en el pre-registro resulta no valido	64
4.13. Caso de uso sobre proceso involucrado cuando en el pre-registro resulta válido.	64
4.14. Diagrama de secuencia para el proceso de pre-registro.	65
4.15. Diagrama de secuencia para el proceso de iniciar sesión.	66
4.16. Diagrama de secuencia para el proceso de creación de nueva cuenta.	67
4.17. Diagrama de secuencia para el proceso de registro.	68
4.18. Diagrama de secuencia para el proceso de eliminación y modificación.	69
4.19. Diagrama de secuencia para el proceso de consulta.	70
4.20. Diagrama de colaboración referente al proceso de consulta.	70

4.21. Diagrama de colaboración referente al proceso de creación de nueva cuenta.	71
4.22. Diagrama de colaboración referente al proceso de pre-registro.	71
4.23. Diagrama de colaboración referente al proceso de registro.	72
4.24. Diagrama de colaboración referente al proceso de eliminación y modificación.	73
4.25. Diagrama de colaboración referente al proceso de iniciar sesión.	73
5.1. Diagrama de actividades del proceso de preregistro.	75
5.2. Diagrama de actividades del proceso de registro.	76
5.3. Diagrama de actividades del proceso de creación de cuenta de usuario	77
5.4. Diagrama de actividades del proceso de modificación.	78
5.5. Diagrama de actividades del proceso de eliminación.	79
5.6. Diagrama de actividades del proceso de consulta.	80
5.7. Diagrama de Estado.	81
5.8. Diagrama de Objetos.	82
5.9. Diagrama de Clases	83
6.1. Encabezado en Flash	85
6.2. Botones en Fiereworks	85
6.3. Fondo de la pantalla principal	86
6.4. Página principal en Dreamweaver	86
6.5. Formularios	87
6.6. Creación de tablas en DBDesigner	87
6.7. Modelo de la Base de Datos en DBDesigner	88
6.8. Crear Base de Datos en MySQL mediante importación de Scripts	89
6.9. Base de datos ya creada	89
6.10. Base de Datos Repositorio	90
6.11. Campos de la tabla de datos personales	90
6.12. Creación de una aplicación NetBeans	91
6.13. Creación de un nuevo proyecto en NetBeans	92
6.14. Creación de un JSP	92
6.15. Ventana que muestra un JSP	93
6.16. Ventana de búsqueda de servidores	94
6.17. Conexión entre aplicación y usuario	94
6.18. Conexión del servlet	100
7.1. Ventana principal	101
7.2. Menú	102
7.3. Ventana que muestra los dos tipos de consulta	103
7.4. Interfaz de los temas propuestos	104
7.5. Interfaz de los temas ya registrados para desarrollarse	104
7.6. Ventana principal de Pre-Registro	105
7.7. Ventana de la interfaz de Pre-Registro para un profesor	106
7.8. Ventana de la interfaz de Pre-Registro para un alumno	107

7.9. Cuenta - Ventana inicial	108
7.10. Inicio de sesión	108
7.11. Menú de las opciones con las que cuenta el profesor	109
7.12. Ventana del registro de tema del profesor	109
7.13. Modificación de datos del profesor	110
7.14. Mensajes dirigidos al profesor	111
7.15. Menú de opciones para el alumno	111
7.16. Ventana del registro de tema del alumno	112
7.17. Ventana del registro del proyecto a desarrollar	113
7.18. Ventana que muestra la opción de modificar datos del alumno	113
7.19. Ventana de mensajes dirigidos al alumno	114
7.20. Ventana principal del administrador. Inicio de sesión	114
7.21. Tres opciones a elegir (alumno/profesor/mensajes)	115
7.22. Consulta de datos sobre alumnos	115
7.23. Consulta de datos sobre profesores	116
7.24. Consulta de datos sobre profesores	116
7.25. Contactos	117

Agradecimientos

Agradecemos a Dios por darnos la fortaleza, paciencia y sabiduría para realizar este proyecto de tesis.

Agradecemos el gran apoyo de Zafra Rojas Ángel Rubén quien fue un guía y base importante en el desarrollo de este proyecto.

Agradezco a mis padres por darme la oportunidad de estudiar la licenciatura, por su apoyo incondicional que me han dado, por sus cuidados, preocupaciones y su amor. Nancy

Agradezco a mis padres por el apoyo incondicional y el entusiasmo con el que siempre me animan, ellos son parte de la razón de todo lo que hago. Diana

Agradezco el apoyo de familiares y amigos que han estado conmigo a lo largo de mi vida. Diana

Justificación

El motivo que impulsó el desarrollo de este sistema surgió a la hora de identificar la dificultad que tienen los alumnos universitarios de la carrera Licenciatura en Sistemas Computacionales al momento de buscar un tema de investigación de tesis, cuya elección puede llegar a ser sencilla o difícil, ya que cuando se cree que se ha encontrado el tema apropiado y de buen gusto, resulta ser que alguien ya lo había investigado anteriormente por lo tanto se debe de optar por otras alternativas, para esto sería muy apropiado contar con un sistema que brinde un conocimiento general de lo que trata cada proyecto de tesis que se encuentra registrado, así como tener acceso a los diferentes temas de investigación que proponen o están trabajando los profesores.

Por lo anteriormente dicho es importante contar con una herramienta que apoye a los pasantes de la carrera a culminar con su proyecto de investigación, brindándoles la posibilidad de consultar los proyectos desarrollados recientemente. Por otra parte se tendrá un mejor control de los proyectos vigentes y será mucho más sencillo su registro, ya que actualmente este proceso se efectúa de manera manual lo que conlleva a errores de ortografía, pérdida de la información y trabajo innecesario, ya que si éste ya se encontraba registrado, el trabajo realizado se pierde al tener que cambiar el tema, entre otros.

Planteamiento del Problema

La Universidad Autónoma del Estado de Hidalgo aprobó la carrera de Sistemas Computacionales en el año de 1997, desde entonces se ha tenido un gran número de estudiantes titulados por diversas modalidades y una de ellas es por la elaboración de tesis. Para esta modalidad en particular existe un problema, cuando los alumnos comienzan con el proceso de elaboración de tesis justamente al realizar la búsqueda de su tema, debido a que no encuentran una Problemática de su interés, o que al identificarla ésta ya fue investigada, o bien, no conocen los temas tentativos de investigación que proponen los profesores y que serían una opción para desarrollar.

Todo esto surge porque el control que actualmente utiliza la Universidad para el registro de temas de tesis se realiza de manera manual, esto provoca que no haya un buen control y los alumnos se vean afectados tener que invertir mas de su tiempo al esperar si su tema ya fue utilizado o sigue disponible. .

Objetivo general

Desarrollar un sitio web para controlar los proyectos elaborados por estudiantes de la Licenciatura en Sistemas Computacionales, que sirva como un apoyo que simplifique el trabajo desempeñado por las personas responsables del control de registros de tesis de la coordinación de esta carrera y brinde al estudiante o académicos que ingresen a él un panorama de los temas propuestos y la información de quien puede asesorarlos.

También se pretende que con la ayuda de este sitio, el alumno se vuelva autodidacta al momento en que alguno de los temas capte su interés y colabore en su desarrollo.

Objetivos Específicos

- Controlar las altas y bajas de temas propuestos para desarrollarlos.
- Tener un control de los registro de los tesis y sus proyectos de investigación.
- Crear una base de datos apoyándonos del programa XAMPP.
- Permita tener una visión mas amplia de los trabajos que están elaborando cada profesor.

Propuesta de Solución

La Problemática descrita anteriormente nos ha llevado a la propuesta de desarrollar un sitio web para controlar los proyectos de fin de carrera de los alumnos de la licenciatura en Sistemas Computacionales de la Universidad Autónoma del Estado de Hidalgo.

Esta propuesta está enfocada principalmente a mejorar:

- El proceso de registro de tesis.
- El control de tesis elaboradas.
- El servicio de información de temas propuestos por académicos en los cuales los alumnos pueden colaborar para su desarrollo.
- La calidad y tiempo de demora en dar respuesta a una petición de temas ya desarrollados, ya que aproximadamente se llega a tardar 15 días en verificarlo.

Con el implemento de este sistema se podrá brindar un apoyo fácil, sencillo y rápido a alumnos que están emprendiendo su búsqueda y realización de tesis. Fácil porque el sistema estará a su servicio en cualquier momento que deseen consultarlo, sencillo porque su utilización será apta para todo usuario, cumplirá con los 10 pasos que estandarizan un sitio web descritos por Shelden. Y rápido por que se reducirá el tiempo de consulta.

De esta manera un alumno o profesor que necesite de un tema de investigación podrá consultar los que se encuentran disponibles e interactuar con los que lo proponen por medio de este sitio web.

Capítulo 1

Marco Referencial

1.1. Metodologías

Como se sabe, hoy en día existen muchos tipos de metodologías que ayudan al análisis y desarrollo de sistemas y depende del programador elegir cual de ellas es más adecuada para utilizarla en la solución de su problema con el fin de obtener un resultado óptimo y facilitar el desarrollo. Aunque se debe de mencionar que la gran mayoría coinciden en ciertos pasos.

Ejemplos de algunas metodologías se presentan a continuación describiendo sus características, con el objetivo de ampliar un poco el panorama de cada una de ellas.

1.1.1. Metodología Web

Esta metodología es empleada para la construcción de sistemas Web [13]. A continuación se describe cada una de sus fases y en la figura. 1.1 se aprecia el su diagrama.



Figura 1.1: Diagrama de la metodología web.

1.1.1.1-Planificación del sistema o página Web

- Elección del tipo de web

Lo primero que se ha de decidir es el tipo de Web que queremos crear, es el punto de partida que afecta a todas las etapas posteriores de creación, realizaremos un diseño, aplicaciones, navegabilidad adecuadas al tipo de Web seleccionada.

Ejemplos:

1. sitio web comercial
2. sitio web profesional
3. sitio web de información
4. sitio web de ocio

- Definición de la Temática

Se ha de definir los temas que se van a exponer en el sitio Web, permitiendo definir términos claves de búsqueda para posteriormente realizar una metodología de posicionamiento. Ejemplos:

1. "Sitio Web comercial donde se vende todo tipo de objetos."
2. "Sitio Web comercial especializado en la compra y venta de pisos."
3. "Sitio Web profesional especializado en consultoría de nuevas tecnologías."
4. "Sitio Web de información acerca del museo del prado."

- Planteamiento de objetivos

Se han de plantear los objetivos que se quieren alcanzar con la creación del sitio Web, para posteriormente marcar estrategias funcional para la consecución de dichos objetivos. Ejemplos:

1. "Dar a conocer el negocio, captación de clientes y generación de nuevos pedidos."
2. "Vender publicidad contextual para obtener ingresos."
3. "Presentar mi Curriculum Vitae en la red."

- Escalabilidad

La escalabilidad se define como las visiones a corto y largo plazo acerca de nuestro sitio Web, si a lo largo del tiempo queremos ampliar nuestro sitio Web con nuevas aplicaciones, nuevas páginas, actualizaciones constantes.

- Definición del diseño

Dependiendo del tipo de Web, la temática seleccionada, los objetivos planteados y la escalabilidad definida, estamos preparados para definir sobre papel el diseño de la Web, incluyendo los fondos, tipos de letras, botones, formularios, links, plantillas, aplicaciones de tal forma que obtengamos "storyboard "de los elementos y diseño que queremos implementar en nuestro sitio Web.

1.1.1.2.-Producción o creación del sitio o página web

- Diseño visual y creación de la información a implementar

Creación del esqueleto de la Web, tablas, encabezados, espacio para imágenes, texto, botones.

Creación de las imágenes que acompañara a nuestro sitio Web, logos, cabeceras, fotografías, además del proceso concepción y materialización de la información que se va a ofrecer.

- Aplicaciones web

Creación de las aplicaciones que contendrán nuestro sitio Web, encuestas, foros, soporte al cliente, pedidos on-line, mediante programación específica y creación de bases de datos.

- Posicionamiento

Una vez que tenemos terminado nuestro sitio Web lo que queremos es que en cualquier buscador, especialmente google, aparezcamos en las primeras posiciones para recibir mas visitas a nuestra Web.

Es este punto donde se ha de implementar una metodología específica para alcanzar un posicionamiento óptimo, definiendo las palabras claves de búsqueda, creación de archivos robots.txt.

- Testeo

Realización de pruebas para comprobar la usabilidad y correcto funcionamiento de nuestro sitio, no queremos mostrar páginas con links rotos, páginas con un formato diferente a las anteriores, aplicaciones no funcionales ya que aportan una imagen negativa y harán que los visitantes no vuelvan a nuestra página.

1.1.1.3.-Mantenimiento y explotación del sitio o página web

- Ampliaciones y actualizaciones

Es muy importante ofrecer información novedosa que atraiga y fidelice a nuestros visitantes o clientes, es importante definir una página de novedades o noticias donde los visitantes obtengan rápidamente las informaciones novedosas y aprecien que el sitio Web se mantiene "vivo" puede ofrecerle nuevos servicios.

- Posicionamiento

El arte del posicionamiento es una metodología que requiere un seguimiento para poder posicionarnos y mantenernos en los primeros puestos de los buscadores, gran parte de nuestro éxito depende del número de visitas que reciba la Web.

- Marketing

Realizar una estrategia de marketing para dar a conocer nuestra Web ya puede ser mediante publicaciones en periódicos, panfletos, links de otras Web.

- Mantenimiento

Realización de programas de mantenimiento para que nuestra Web este en funcionamiento durante toda su vida.

1.1.2. Metodología de desarrollo de sistemas

Toda actividad debe estar basada en una metodología [5] y en principio, cualquier metodología es mejor que ninguna. Para desarrollar ésta se consideran las siguientes etapas:

1.1.2.1.-Análisis

1. Definición del problema
2. Estudio de la situación actual
3. Requisitos a considerar
4. Estudio de factibilidad

1.1.2.2.-Diseño lógico

1. Análisis funcional
2. Definición de datos y procesos
3. Modelización

1.1.2.3.-Diseño físico

1. Creación de ficheros y tablas
2. Elaboración de programas

1.1.2.4.-Implementación y control

1. Formación del usuario
2. Implantación del sistema
3. Explotación del sistema
4. Mantenimiento

1.1.3. Método de Cascada Pura

En un modelo en cascada, un proyecto progresa a través de una secuencia ordenada de pasos partiendo de la especificación de requerimientos hasta el mantenimiento del mismo.

El método realiza una revisión al final de cada etapa para determinar si está preparado para pasar a la siguiente etapa, por ejemplo, desde el análisis de requerimientos hasta el diseño. Cuando la revisión determina que el proyecto no está listo para pasar a la siguiente, permanece en la etapa actual hasta que esté preparado.

El modelo en cascada está dirigido por documentos. Ayuda a localizar errores en las primeras etapas del proyecto a un bajo costo. Ayuda a minimizar los gastos de la planificación porque permite realizarla sin planificación y sin problemas.

En resumen, los inconvenientes del modelo en cascada hacen que sea, a menudo, un modelo poco apropiado para un proyecto de desarrollo rápido. Incluso en los casos en los que las ventajas del modelo en cascada pura superan los inconvenientes, los modelos de cascada modificada (con retroceso) pueden funcionar mejor.

Las desventajas del modelo se centran en las dificultades para especificar claramente los requerimientos al comienzo del proyecto, antes de que se realice ningún

trabajo de diseño y antes de escribir ningún código.

No proporciona resultados tangibles en forma de software hasta el final del ciclo de vida de algunas herramientas, métodos y actividades que abarcan varias etapas de la cascada; estas actividades son difíciles de ajustar en las etapas discontinuas del modelo para un proyecto de desarrollo rápido, el modelo en cascada puede suponer una cantidad excesiva de documentación.

El modelo genera pocos signos visibles de progreso hasta el final. Esto puede dar la impresión de un desarrollo lento, existe la incertidumbre de los clientes si sus proyectos serán entregados a tiempo.

Sin embargo, puede servir como un modelo de proceso útil en situaciones donde los requerimientos están fijos [1].

1.1.4. Método Espiral

Es un modelo de ciclo de vida orientado a riesgos que divide un proyecto de software en mini-proyectos [1]. Cada mini proyecto se centra en uno o más riesgos importantes hasta que todos estén controlados.

Después de controlar todos los riesgos más importantes, el modelo en espiral finaliza del mismo modo que el ciclo de vida en cascada.

1.1.4.1.-Funcionamiento

1. Se parte de una escala pequeña en medio de la espiral, se localizan los riesgos, se genera un plan para manejar los riesgos, y a continuación se establece una aproximación a la siguiente interacción.
2. Cada iteración supone que el proyecto pasa a una escala superior. Se avanza un nivel en el Espiral, se comprueba que se tiene lo que se desea, y después se comienza a trabajar en el siguiente nivel
3. Con cada iteración a través del espiral se construye sucesivas versiones de software cada vez más completas. En cada bucle alrededor del espiral, la culminación del análisis de riesgo resulta una decisión de seguir o no seguir.
4. Cada interacción en el método espiral lleva consigo los seis pasos que a continuación se nombran: Determinar objetivos, alternativas y límites, Identificar y resolver riesgos, Evaluar alternativas, Generar las entregas de esa iteración, y comprobar que son correctas.
5. En el modelo en espiral, las primeras iteraciones son las menos costosas.
6. Supone menos gasto desarrollar el concepto de operación que realizar el desarrollo de los requerimientos, y también es menos costoso desarrollar los

requerimientos que llevar a cabo el desarrollo del diseño, la implementación del producto y la prueba del mismo.

En cada Cuadrante del Método espiral se realiza las siguientes actividades:

1. Planificación:
 - a) Determinación de objetivos, alternativas, restricciones, y elaboración del plan de desarrollo para el ciclo actual.
2. Análisis de Riesgos:
 - a) Evaluación de las alternativas, identificación y resolución de riesgos. Se decide si se sigue o no con el proyecto
3. Ingeniería:
 - a) Desarrollo del producto siguiendo un modelo: del ciclo de vida o cascada, prototipo, entre otros. Evaluación por el cliente.
 - b) Valoración de resultados.

1.1.5. Método de Codificar y Corregir (Code-and-fix)

Es un modelo poco útil [1], pero sin embargo bastante común. Se puede tener una especificación formal, o no tenerla. Si no se ha utilizado formalmente un método, probablemente ya se esté usando el método Codificar y Corregir en forma intuitiva. Cuando se utiliza éste método se empieza con una idea general de lo que se necesita construir, se utiliza cualquier combinación de diseño, código, depuración y métodos de prueba no formales que sirven hasta que se tiene el producto listo para entregarlo.

1.1.5.1.-Ventajas

1. No conlleva ninguna gestión; no se pierde tiempo en la planificación, en la documentación, en el control de calidad, en el cumplimiento de los estándares, o en cualquier otra actividad que no sea codificación pura.
2. Como se pasa directamente a codificar, se pueden mostrar inmediatamente indicios de progreso.
3. Requiere poca experiencia: cualquier persona que haya escrito alguna vez un programa está familiarizada con éste modelo.
4. Para proyectos pequeños que se intentan liquidar en un tiempo breve, o para modelos como programas de demostración o prototipos desechables, el modelo codificar y corregir puede ser útil.

1.1.5.2.-Desventajas

1. El modelo resulta peligroso para otro tipo de proyectos que no sean pequeños.
2. Puede que no suponga gestión alguna, pero tampoco ofrece medios de evaluación del progreso.
3. No proporciona medios de evaluación de la calidad o de identificación de riesgos.
4. Si al llevar tres cuartas partes de la codificación descubre que el diseño es incorrecto, no hay otra solución que desechar el trabajo y comenzar de nuevo.

1.1.6. Modelo Incremental

Entrega una serie de lanzamientos, llamados incrementos, que proporcionan en forma progresiva más funcionalidad para los clientes a medida que se entrega cada uno de los incrementos [1]:

1. El primer incremento es un producto esencial, sólo con los requisitos básicos
2. Los primeros incrementos son versiones "incompletas"del producto final, pero proporcionan al usuario la funcionalidad que necesita y una plataforma para evaluarlo.
3. El desarrollo incremental es útil sobre todo cuando el personal necesario para una implementación completa no está disponible.
4. Además, los incrementos se pueden planear para manejar los riesgos técnicos.

1.1.7. Modelo DRA (Desarrollo Rápido de Aplicaciones)

1. Modelo de proceso de software incremental que resalta un ciclo de desarrollo corto.[1]
2. Es una adaptación a "alta velocidad"del modelo en cascada en el que se logra el desarrollo rápido mediante un enfoque de construcción basado en componentes.
3. Permite que un equipo de desarrollo cree un "sistema completamente funcional"dentro de un periodo muy corto.

1.1.7.1.-Inconvenientes

1. Para proyectos grandes, pero escalables, el DRA necesita suficientes recursos humanos para crear el número correcto de equipos DRA
2. Si los desarrolladores y clientes no se comprometen con las actividades rápidas necesarias para completar un sistema en un marco de tiempo muy breve, los proyectos de DRA fallarán.
3. Si un sistema no se puede modular en forma apropiada, la construcción de los componentes necesarios para el DRA será Problemática.
4. El DRA sería inapropiado cuando los riesgos técnicos son altos, cuando se aplican muchas nuevas tecnologías.

1.1.8. Modelo de Proceso Incremental

Producen una versión completa en forma incremental con cada iteración [1].

Los caracteriza la forma en que permiten que los ingenieros de software desarrollen versiones cada vez más completas del software.

1.1.8.1.-Problemas

1. El cliente ve lo que parece una versión en funcionamiento del software, sin saber que el prototipo se va a desechar.
2. Además por la prisa de hacerlo funcionar no se ha considerado la calidad del software.
3. El cliente no entiende por qué el sistema se debe construir otra vez para incrementar la calidad.
4. Tal vez se utilice un sistema operativo o lenguaje de programación inadecuado sólo porque está disponible y es conocido.
5. Después de un tiempo, el desarrollador quizá se familiarice con estas selecciones y olvide las razones por las que son inapropiadas.

1.1.8.2.-Clave de éxito

1. Definir las reglas del juego desde el principio.
2. El cliente y el desarrollador se deben poner de acuerdo en que el prototipo se construya y sirva como un mecanismo para la definición de requisitos, en que se descarte, al menos en parte, y en que después se desarrolle el software real con un enfoque hacia la calidad.

1.1.9. UML

UML [3] que es un Lenguaje de Modelado Unificado, estándar para escribir planos de software. UML puede utilizarse para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra gran cantidad de software y se apoya de los siguientes diagramas:

1.1.9.1.-Diagrama de Clases

Un diagrama de clases sirve para visualizar las relaciones entre las clases que involucran el sistema, las cuales pueden ser asociativas, de herencia, de uso y de contenido.

Un diagrama de clases esta compuesto por los siguientes elementos:

1. Clase: atributos, métodos y visibilidad.
2. Relaciones: Herencia, Composición, Agregación, Asociación y Uso.

Elementos:

1. Clase.- Es la unidad básica que encapsula toda la información de un Objeto (un objeto es una instancia de una clase). A través de ella podemos modelar el entorno en estudio (una Casa, un Auto, una Cuenta Corriente, entre otros.).

En UML, una clase es representada por un rectángulo que posee tres divisiones véase figura. 1.2:

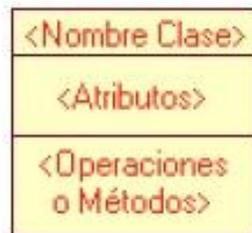


Figura 1.2: Diagrama de clases.

En donde:

1. Superior: Contiene el nombre de la Clase.
2. Intermedio: Contiene los atributos (o variables de instancia) que caracterizan a la Clase (pueden ser private, protected o public).

3. Inferior: Contiene los métodos u operaciones, los cuales son la forma como interactúa el objeto con su entorno (dependiendo de la visibilidad: private, protected o public).

1.1.9.2.-Diagrama de objetos

Un objeto es una instancia de clase (una entidad que tiene valores específicos de los atributos y acciones).

Un diagrama de Objeto se puede considerar un caso especial de un diagrama de clase. Los diagramas de objetos usan un sub conjunto de elementos de un diagrama de clase para enfatizar la relación entre las instancias de las clases en algún punto en el tiempo. Estos son útiles para entender los diagramas de clases. Estos no muestran nada diferente en su arquitectura a los diagramas de secuencia, pero reflejan multiplicidad y roles.

1.1.9.3.-Diagrama caso de uso

Un caso de uso es una descripción de las acciones de un sistema desde el punto de vista del usuario. Para los desarrolladores del sistema, ésta es una herramienta valiosa, ya que es una técnica de aciertos y errores para obtener los requerimientos del sistema desde el punto de vista del usuario. Esto es importante si la finalidad es crear un sistema que pueda ser utilizado por la gente en general.

El valor verdadero de un caso de uso reposa en dos áreas:

1. La descripción escrita del comportamiento del sistema al afrontar una tarea de negocio o un requisito de negocio. Esta descripción se enfoca en el valor suministrado por el sistema a entidades externas tales como usuarios humanos u otros sistemas.
2. La posición o contexto del caso de uso entre otros casos de uso. Dado que es un mecanismo de organización, un conjunto de casos de uso coherentes, consistentes promueve una imagen fácil del comportamiento del sistema, un entendimiento común entre el cliente/propietario/usuario y el equipo de desarrollo.

Es práctica común crear especificaciones suplementarias para capturar detalles de requisitos que caen fuera del ámbito de las descripciones de los casos de uso. Ejemplos de esos temas incluyen rendimiento, temas de escalabilidad/gestión, o cumplimiento de estándares

1.1.9.4.-Diagrama de Estado

Es una manera de caracterizar un cambio en un sistema, es decir, que los objetos que lo componen modificaron su estado como respuesta a los sucesos y al tiempo. Ejemplos:

1. Cuando acciona el interruptor, la fuente de luz cambia su estado de apagada a encendida.
2. Cuando presiona un botón de un control remoto, una televisión cambia su estado para mostrarle un canal u otro.

El diagrama de estados UML captura este tipo de cambios. Presenta los estados en los que puede encontrarse un objeto junto con las transiciones entre los estados, y muestra los puntos inicial y final de una secuencia de cambios de estado.

Un diagrama de estados también se conoce como un motor de estados.

- Ventajas

1. Proporciona una gran variedad de símbolos y abarca varias ideas (todas para modelar los cambios por los que pasa un objeto).
2. Permite a los analistas, diseñadores y desarrolladores comprender el comportamiento de los objetos de un sistema.
3. Se aseguran de dar una clara representación del comportamiento del objeto para aumentar la probabilidad de que el equipo de desarrollo produzca un sistema que cumpla con los requerimientos.

1.1.9.5.-Diagrama de Secuencia

El diagrama de secuencia consta de objetos que se representan del modo usual: rectángulo con nombre (subrayado), mensajes representados por líneas continuas con una punta de flecha y el tiempo representado como una progresión vertical.

Este diagrama muestra la mecánica de la interacción con base en tiempos.

- Objetos

Los objetos se colocan cerca de la parte superior del diagrama de izquierda a derecha y se acomodan de manera que simplifiquen al diagrama. La extensión que está debajo (y en forma descendente) de cada objeto será una línea discontinua conocida como la línea de vida de un objeto. Junto con ésta se encuentra un pequeño rectángulo conocido como activación, el cual representa la ejecución de una operación que realiza el objeto. La longitud del rectángulo se interpreta como la duración de la activación.

- Mensaje

Un mensaje que va de un objeto a otro de la línea de vida de un objeto a la de otro. Un objeto puede enviarse un mensaje a sí mismo.

Un mensaje puede ser simple, sincrónico o asincrónico. Un mensaje simple es la transferencia del control de un objeto a otro. Si un objeto envía un mensaje

sincrónico, esperará la respuesta a tal mensaje antes de continuar con su trabajo. Si un objeto envía un mensaje sincrónico, no esperará una respuesta antes de continuar. En el diagrama de secuencia, los símbolos de mensaje varían, por ejemplo, la punta de la flecha de un mensaje simple está formada por líneas, la punta de la flecha de un mensaje sincrónico está rellena y la de un asincrónico tiene una sola línea.

- Tiempo

El diagrama representa al tiempo en dirección vertical. El tiempo se inicia en la parte superior y avanza hacia la parte inferior. Un mensaje que esté más cerca de la parte superior ocurrirá antes que uno que esté cerca de la parte inferior.

Con ello, el diagrama de secuencia tiene dos dimensiones. La dimensión horizontal es la disposición de los objetos y la dimensión vertical muestra el paso del tiempo.

1.1.9.6.-Diagrama de Actividades

Un diagrama de actividades ha sido diseñado para mostrar una visión simplificada de lo que ocurre durante una operación o proceso. Es una extensión de un diagrama de estados, mismo que ya conocí. El diagrama de estados muestra los estados de un objeto y representa las actividades con flechas conectan a los estados. El diagrama de actividades resalta, precisamente, a las actividades [2]

A cada actividad se le presenta por un rectángulo con las esquinas redondeadas. El procesamiento dentro de una actividad se lleva a cabo y al realizarse se continúa con la siguiente actividad. Una flecha representa la transición de una a otra actividad. Al igual que el diagrama de estados, el de actividades cuenta con un punto inicial (representado por un círculo) y uno final (representado por una flecha).

El diagrama de actividades del UML es muy parecido a un diagrama de flujo. Muestra los pasos, puntos de decisión y bifurcaciones. Este tipo de diagramas es útil para representar las operaciones de un objeto y los procesos de negocios.

1.1.9.7.-Diagrama de Colaboración

Un diagrama de objetos muestra a los objetos como tales y sus relaciones entre sí. Un diagrama de colaboraciones es una extensión de uno de objetos. Además de las relaciones entre objetos, el diagrama de colaboraciones muestra los mensajes que se enviarán los objetos entre sí. Por lo general, evitarán la multiplicidad dado que podría ser fuente de confusión.

Para representar un mensaje, dibujará una flecha cerca de la línea de asociación entre dos objetos, esta flecha apunta al objeto receptor. El tipo de mensaje se mostrará en una etiqueta cerca de la flecha; por lo general, el mensaje le indicará

al objeto receptor que ejecute una de sus operaciones. El mensaje finalizará con un par de parámetros con los que funcionará la operación.

1.1.9.8.-Diagrama de Escenarios

Es un diagrama que nos representa una secuencia de sucesos. Los sucesos se producen siempre que se intercambia información entre un objeto del sistema y un agente externo.

1.2. Herramientas de desarrollo de Sistemas para Internet

Con los cambios que va sufriendo la tecnología de información y comunicación en específico (creaciones e innovaciones), los métodos para el desarrollo de sistemas cada día se van modificando y causa de ello es que el programador requiere de herramientas que lo apoyen, facilitando su trabajo pero que le ayuden a realizarlo con calidad y eficiencia.

Las herramientas de programación, son aquellas que permiten realizar aplicativos, programas, rutinas, utilitarios y sistemas para que la parte física del computador u ordenador, funcione y pueda producir resultados.

Existen múltiples herramientas de programación en el mercado, tanto para analistas expertos como para analistas inexpertos. Las herramientas de programación más comunes, cuentan con programas de depuración o debugger, que son utilitarios que nos permiten detectar los posibles errores en tiempo de ejecución o corrida de rutinas y programas.

Entre otras herramientas de programación encontramos librerías y componentes, dados por algunos lenguajes de programación como son el C++ y delphi. Otras más son los lenguajes de programación, que nos permiten crear rutinas y programas.

Algunas de estas herramientas que se emplean en la construcción de sistemas son mencionadas en los próximos párrafos:

1.2.1. CSS

(Cascading Style Sheets, u Hojas de Estilo en Cascada)[2] es la tecnología desarrollada por el World Wide Web Consortium (W3C) con el fin de separar la estructura de la presentación. A pesar de que la recomendación oficial del grupo de trabajo de la W3C ya había alcanzado la estabilidad requerida para que fuera soportada por los principales navegadores comerciales, como Netscape e Internet Explorer, tan tempranamente como en el año 1998, la situación de

entonces, comúnmente conocida como la "guerra de los navegadores", hacía que los intereses comerciales de las dos compañías en lucha por el mercado de usuarios de Internet se interpusieran en el camino de las CSS.

1.2.2. ASP

El ASP [13] es una tecnología dinámica funcionando del lado del servidor, lo que significa que cuando el usuario solicita un documento ASP, las instrucciones de programación dentro del script son ejecutadas para enviar al navegador únicamente el código HTML resultante. La ventaja principal de las tecnologías dependientes del servidor radica en la seguridad que tiene el programador sobre su código, ya que éste se encuentra únicamente en los archivos del servidor que al ser solicitado a través del web, es ejecutado, por lo que los usuario no tienen acceso más que a la página resultante en su navegador.

El desarrollo que se ha venido dando a lo que es ASP ha sido bastante amplio. Entre sus funciones principales están el acceso a base de datos, envío de correo electrónico, creación dinámica de gráficos y otros. Esto debido a que el ASP es tan eficiente con escribir código directamente a la interfase de aplicación del servidor, con la ventaja de que es más eficiente que el CGI que depende de un compilador ya que el ASP corre como un servicio en el servidor, tomando ventaja de la arquitectura de multitareas.

1.2.3. XAJAX

Es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones [14].

Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se requieren al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página. JavaScript es el lenguaje interpretado (scripting language) en el que normalmente se efectúan las funciones de llamada de Ajax mientras que el acceso a los datos se realiza mediante XMLHttpRequest, objeto disponible en los navegadores actuales. En cualquier caso, no es necesario que el contenido asíncrono esté formateado en XML.

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

AJAX es una combinación de cuatro tecnologías ya existentes:

1. XHTML (o HTML) y hojas de estilos en cascada (CSS) para el diseño que acompaña a la información.
2. Document Object Model (DOM) accedido con un lenguaje de scripting por parte del usuario, especialmente implementaciones ECMAScript como JavaScript y JScript, para mostrar e interactuar dinámicamente con la información presentada.
3. XML es el formato usado generalmente para la transferencia de datos solicitados al servidor, aunque cualquier formato puede funcionar, incluyendo HTML preformateado, texto plano, JSON y hasta EBML.

Como el DHTML, LAMP o SPA, AJAX no constituye una tecnología en sí, sino que es un término que engloba a un grupo de éstas que trabajan conjuntamente.

1.2.4. XAMPP

Es un servidor independiente de plataforma, software libre, que consiste principalmente en la base de datos MySQL, el servidor Web Apache y los intérpretes para lenguajes de script: PHP y Perl. El programa está liberado bajo la licencia GNU y actúa como un servidor Web libre, fácil de usar y capaz de interpretar páginas dinámicas. Actualmente XAMPP esta disponible para Microsoft Windows, GNU/Linux, Solaris, y MacOS X [10].

El nombre proviene del acrónimo:

X (para cualquiera de los diferentes sistemas operativos)

A (Utiliza el servidor Apache)

M (Manejo de base de datos en MySQL)

P (PHP)

P (PERL)

1.2.4.1.-Características y Requisitos

XAMPP solamente requiere descargar y ejecutar un archivo zip, rar, o exe, con unas pequeñas configuraciones en alguno de sus componentes que el servidor Web necesitará. XAMPP se actualiza regularmente para incorporar las últimas versiones de Apache/MySQL/PHP y Perl. También incluye otros módulos como OpenSSL y phpMyAdmin. Para instalar XAMPP se requiere solamente una pequeña fracción del tiempo necesario para descargar y configurar los programas por separado.

1.2.4.2.-Aplicaciones

Oficialmente, los diseñadores de XAMPP solo pretendían su uso como una herramienta de desarrollo, para permitir a los diseñadores de sitios Webs y programadores testear su trabajo en sus propios ordenadores sin ningún acceso

a Internet. En la práctica sin embargo, XAMPP es utilizado actualmente para servidor de sitios Web, y con algunas modificaciones es generalmente lo suficientemente seguro para serlo. Con el paquete se incluye una herramienta especial para proteger fácilmente las partes más importantes.

1.2.4.3.- Apache

El servidor HTTP Apache [10] es un servidor web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, entre otros.), Windows, Macintosh y otras, que implementa el protocolo HTTP y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. Su nombre se debe a que Behelendorf eligió ese nombre porque quería que tuviese la connotación de algo que es firme y enérgico pero no agresivo, y la tribu Apache fue la última en rendirse al que pronto se convertiría en gobierno de EEUU, y en esos momentos la preocupación de su grupo era que llegasen las empresas y civilizaran el paisaje que habían creado los primeros ingenieros de internet.

Además Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. Era, en inglés, a patchy server (un servidor "parcheado").

El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation. Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

Apache tiene amplia aceptación en la red: desde 1996, Apache, es el servidor HTTP más usado. Alcanzó su máxima cuota de mercado en 2005 siendo el servidor empleado en el 70 por ciento de los sitios web en el mundo, sin embargo ha sufrido un descenso en su cuota de mercado en los últimos años. (Estadísticas históricas y de uso diario proporcionadas por Netcraft).

La mayoría de las vulnerabilidades de la seguridad descubiertas y resueltas tan sólo pueden ser aprovechadas por usuarios locales y no remotamente. Sin embargo, algunas se pueden accionar remotamente en ciertas situaciones, o explotar por los usuarios locales malévolos en las disposiciones de recibimiento compartidas que utilizan PHP como módulo de Apache.

1.2.5. MySql

MySQL [10] es un sistema de gestión de base de datos relacional, multiusuario y multiusuario con más de seis millones de instalaciones. MySQL AB desde enero de 2008 una subsidiaria de Sun Microsystems y ésta a su vez de Oracle Corporation desde abril de 2009- desarrolla MySQL como software libre en un esquema de licenciamiento dual. Por un lado se ofrece bajo la GNU GPL para

cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C.

Al contrario de proyectos como Apache, donde el software es desarrollado por una comunidad pública y el copyright del código está en poder del autor individual, MySQL es propietario y está patrocinado por una empresa privada, que posee el copyright de la mayor parte del código.

Esto es lo que posibilita el esquema de licenciamiento anteriormente mencionado. Además de la venta de licencias privativas, la compañía ofrece soporte y servicios. Para sus operaciones contratan trabajadores alrededor del mundo que colaboran vía Internet. MySQL AB fue fundado por David Axmark, Allan Larson y Michael Widenius.

1.2.6. JOOMLA

Joomla [8], es un sistema de administración de contenidos de código abierto construido con PHP bajo una licencia GPL. Este administrador de contenidos se usa para publicar en Internet e intranets utilizando una base de datos MySQL. En Joomla! se incluyen características como: hacer caché de páginas para mejorar el rendimiento, indexamiento web, feed RSS, versiones imprimibles de páginas, flash con noticias, blogs, foros, polls (encuestas), calendarios, búsqueda en el sitio web, e internacionalización del lenguaje. Su nombre es una pronunciación fonética para anglófonos de la palabra swahili jumla que significa todos juntos o como un todo. Se escogió como una reflexión del compromiso del grupo de desarrolladores y la comunidad del proyecto.

La primera versión de Joomla (Joomla 1.0.0) fue publicada el 16 de septiembre de 2005. Se trataba de una versión mejorada de Mambo 4.5.2.3 combinada con otras modificaciones de seguridad y anti-bugs. Actualmente los programadores han publicado Joomla 1.5 estable bajo un código completamente reescrito y construido bajo PHP 5.

1.2.7. Base de Datos

Una base de datos o banco de datos [12] es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta. En la actualidad, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital (electrónico), que ofrece un amplio rango de soluciones al problema de almacenar datos.

Existen unos programas denominados sistemas gestores de bases de datos, abreviado SGBD, que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Las propiedades de estos SGBD, así como su utilización y administración, se estudian dentro del ámbito de la informática.

Las aplicaciones más usuales son para la gestión de empresas e instituciones públicas. También son ampliamente utilizadas en entornos científicos con el objeto de almacenar la información experimental.

Aunque las bases de datos pueden contener muchos tipos de datos, algunos de ellos se encuentran protegidos por las leyes de varios países. Por ejemplo en España, los datos personales se encuentran protegidos por la Ley Orgánica de Protección de Datos de Carácter Personal (LOPD).

1.2.7.1.-Tipos de base de datos

Las bases de datos pueden clasificarse de varias maneras, de acuerdo al criterio elegido para su clasificación:

- Bases de datos estáticas

Éstas son bases de datos de sólo lectura, utilizadas primordialmente para almacenar datos históricos que posteriormente se pueden utilizar para estudiar el comportamiento de un conjunto de datos a través del tiempo, realizar proyecciones y tomar decisiones.

- Bases de datos dinámicas

Éstas son bases de datos donde la información almacenada se modifica con el tiempo, permitiendo operaciones como actualización y adición de datos, además de las operaciones fundamentales de consulta. Un ejemplo de esto puede ser la base de datos utilizada en un sistema de información de una tienda de abarrotes, una farmacia, un videoclub, entre otros.

1.2.7.2.-Según el contenido

- Bases de datos bibliográficas

Solo contienen un subrogante (representante) de la fuente primaria, que permite localizarla. Un registro típico de una base de datos bibliográfica contiene información sobre el autor, fecha de publicación, editorial, título, edición, de una determinada publicación, entre otros. Puede contener un resumen o extracto de la publicación original, pero nunca el texto completo, porque sino estaríamos en presencia de una base de datos a texto completo (o de fuentes primarias-ver más abajo). Como su nombre lo indica, el contenido son cifras o números. Por ejemplo, una colección de resultados de análisis de laboratorio, entre otras.

- Bases de datos de texto completo

Almacenan las fuentes primarias, como por ejemplo, todo el contenido de todas las ediciones de una colección de revistas científicas.

1.2.7.3.-Directorios

Un ejemplo son las guías telefónicas en formato electrónico.

1.2.8. Bases de datos o "bibliotecas"de información Biológica

Son bases de datos que almacenan diferentes tipos de información proveniente de las ciencias de la vida o médicas [12]. Se pueden considerar en varios subtipos:

1. Aquellas que almacenan secuencias de nucleótidos o proteínas.
2. Las bases de datos de rutas metabólicas
3. Bases de datos de estructura, comprende los registros de datos experimentales sobre estructuras 3D de biomoléculas
4. Bases de datos clínicas
5. Bases de datos bibliográficas (biológicas)

1.2.9. modelado de los datos

1. Las estructuras de datos de la base: El tipo de los datos que hay en la base y la forma en que se relacionan [6].
2. Las restricciones de integridad: Un conjunto de condiciones que deben cumplir los datos para reflejar correctamente la realidad deseada.
3. Operaciones de manipulación de los datos: típicamente, operaciones de agregado, borrado, modificación y recuperación de los datos de la base.

Otro enfoque es pensar que un modelo de datos permite describir los elementos de la realidad que intervienen en un problema dado y la forma en que se relacionan esos elementos entre sí. No hay que perder de vista que una Base de Datos siempre está orientada a resolver un problema determinado, por lo que los dos enfoques propuestos son necesarios en cualquier desarrollo de software.

1.3. Lenguajes de desarrollo de sistemas

Actualmente existen diferentes lenguajes de programación para desarrollar en la Web, estos han ido surgiendo debido a las tendencias y necesidades de las plataformas. En la presente tesis se pretende mostrar las ventajas y desventajas de los lenguajes más conocidos.

Desde los inicios de Internet, fueron surgiendo diferentes demandas por los usuarios y se dieron soluciones mediante lenguajes estáticos. A medida que paso el tiempo, las tecnologías fueron desarrollándose y surgieron nuevos problemas a dar solución. Esto dio lugar a desarrollar lenguajes de programación para la Web dinámicos, que permitieran interactuar con los usuarios y utilizaran sistemas de Bases de Datos. A continuación se da una introducción a los diferentes lenguajes de programación para la web.

1.3.1. PHP

PHP [11] es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK.

PHP es un lenguaje interpretado de propósito general ampliamente usado y que está diseñado especialmente para desarrollo web y puede ser incrustado dentro de código HTML. Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno.

1.3.1.1.-Ventajas

1. Es un lenguaje multiplataforma.
2. Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL.
3. Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
4. Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
5. Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
6. Permite las técnicas de Programación Orientada a Objetos.
7. Biblioteca nativa de funciones sumamente amplia e incluida.
8. No requiere definición de tipos de variables.
9. Tiene manejo de excepciones (desde PHP5).

1.3.1.2.-Desventajas

Si bien PHP no obliga a quien lo usa a seguir una determinada metodología a la hora de programar (muchos otros lenguajes tampoco lo hacen), aun estando dirigido a alguna en particular, el programador puede aplicar en su trabajo cualquier técnica de programación y/o desarrollo que le permita escribir código ordenado, estructurado y manejable. Un ejemplo de esto son los desarrollos que en PHP se han hecho del patrón de diseño Modelo Vista Controlador (o MVC), que permiten separar el tratamiento y acceso a los datos, la lógica de control y la interfaz de usuario en tres componentes independientes.

1.3.2. Programación Java

El lenguaje para la programación en Java [5], fue desarrollado por la compañía Sun Microsystems, con la idea original de usarlo para la creación de páginas WEB.

Esta programación Java tiene muchas similitudes con el lenguaje C y C++, así que si se tiene conocimiento de este lenguaje, el aprendizaje de la programación Java será de fácil comprensión por un programador que haya realizado programas en estos lenguajes.

Con la programación en Java, se pueden realizar distintos aplicativos, como son applets, que son aplicaciones especiales, que se ejecutan dentro de un navegador al ser cargada una página HTML en un servidor WEB, Por lo general los applets son programas pequeños y de propósitos específicos.

Java permite la modularidad por lo que se pueden hacer rutinas individuales que sean usadas por más de una aplicación, por ejemplo tenemos una rutina de impresión que puede servir para el procesador de palabras, como para la hoja de calculo.

1.3.3. Programación avanzada en Java

El lenguaje para la programación avanzada en Java, es un lenguaje orientado a objeto, de una plataforma independiente [7].

Otra de las utilidades de la programación avanzada en Java es el desarrollo de aplicaciones, que son programas que se ejecutan en forma independiente, es decir con la programación Java, se pueden realizar aplicaciones como un procesador de palabras, una hoja que sirva para cálculos, una aplicación gráfica, entre otros. en resumen cualquier tipo de aplicación se puede realizar con ella.

1.3.4. Programación Web

La programación Web, parte de las siglas WWW, que significan World Wide Web o telaraña mundial [2]. Para realizar una página con la programación

Web, se deben tener claros, tres conceptos fundamentales los cuales son, el URL(Uniform Resource Locators), es un sistema con el cual se localiza un recurso dentro de la red, este recurso puede ser una página web, un servicio o cualquier otra cosa. En resumen el URL no es más que un nombre, que identifica una computadora, dentro de esa computadora un archivo que indica el camino al recurso que se solicita.

El siguiente concepto dentro de la programación Web, es el protocolo encargado de llevar la información que contiene una página Web por toda la red de internet, como es el HTTP(Hypertext Transfer Protocol).

Y por último el lenguaje necesario cuya funcionalidad es la de representar cualquier clase de información que se encuentre almacenada en una página Web, este lenguaje es el HTML(Hypertext Markup Language).

Con el comienzo de Internet y la programación web , se desfasaron los diseños gráficos tradicionales, con lo que se empezaron a diseñar interfaces concretas para este medio, buscando ficheros pequeños para facilitar la carga de los mismos. La programación web se orientaba a un diseño muy cargado y interactuando con el usuario, mientras que al empezar a competir con millones de webs se ha optado más por el diseño sencillo y de fácil comprensión.

En programación web se creó la necesidad de conocer a fondo diferentes lenguajes de programación como HTML, JavaScript y DHTML.

Con esto se creó un nuevo profesional de la informática, el diseñador web, experto en estos menesteres, que viene siendo algo así como un experto en programación web, a caballo entre el diseñador gráfico tradicional y el programador de aplicaciones llevadas a Internet.

1.3.5. HTML

Desde el surgimiento de internet se han publicado sitios web gracias al lenguaje HTML. Es un lenguaje estático para el desarrollo de sitios web (acrónimo en inglés de HyperText Markup Language, en español Lenguaje de Marcas Hipertextuales). Desarrollado por el World Wide Web Consortium (W3C). Los archivos pueden tener las extensiones (htm, html)[11].

HTML es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML se escribe en forma de etiquetas, rodeadas por corchetes angulares. HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script (por ejemplo Javascript), el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML.

1.3.5.1.-Ventajas

1. Sencillo que permite describir hipertexto.
2. Texto presentado de forma estructurada y agradable.
3. No necesita de grandes conocimientos cuando se cuenta con un editor de páginas web o WYSIWYG.
4. Archivos pequeños.
5. Despliegue rápido.
6. Lenguaje de fácil aprendizaje.
7. Lo admiten todos los exploradores.

1.3.5.2.-Desventajas

1. Lenguaje estático.
2. La interpretación de cada navegador puede ser diferente.
3. Guarda muchas etiquetas que pueden convertirse en basura dificultando la corrección.
4. El diseño es más lento.
5. Las etiquetas son muy limitadas.

1.3.6. Java Script

Javascript [13] es un lenguaje de programación que permite a los desarrolladores crear acciones en sus páginas web. Pero ¿Qué es Javascript?, ¿Cómo nace Javascript? son algunas de las preguntas que a continuación se intenta resolver.

Javascript es un lenguaje que puede ser utilizado por profesionales y para quienes se inician en el desarrollo y diseño de sitios web. No requiere de compilación ya que el lenguaje funciona del lado del cliente, los navegadores son los encargados de interpretar estos códigos.

Javascript es un lenguaje con muchas posibilidades, utilizado para crear pequeños programas que luego son insertados en una página web y en programas más grandes, orientados a objetos mucho más complejos. Con Javascript podemos crear diferentes efectos e interactuar con nuestros usuarios.

Este lenguaje posee varias características, entre ellas podemos mencionar que es un lenguaje basado en acciones que posee menos restricciones. Además, es

un lenguaje que utiliza Windows y sistemas X-Windows, gran parte de la programación en este lenguaje está centrada en describir objetos, escribir funciones que respondan a movimientos del mouse, aperturas, utilización de teclas, cargas de páginas entre otros.

Es necesario resaltar que hay dos tipos de JavaScript: por un lado está el que se ejecuta en el cliente, este es el Javascript propiamente dicho, aunque técnicamente se denomina Navigator JavaScript. Pero también existe un Javascript que se ejecuta en el servidor, es más reciente y se denomina LiveWire Javascript.

1.3.6.1.-Ventajas

1. Lenguaje de scripting seguro y fiable.
2. Los script tienen capacidades limitadas, por razones de seguridad.
3. El código Javascript se ejecuta en el cliente.

1.3.6.2.-Desventajas

1. Código visible por cualquier usuario.
2. El código debe descargarse completamente.
3. Puede poner en riesgo la seguridad del sitio, con el actual problema llamado XSS (significa en inglés Cross Site Scripting renombrado a XSS por su similitud con las hojas de estilo CSS).

1.3.7. Lenguaje ASP

Es una tecnología del lado de servidor desarrollada por Microsoft para el desarrollo de sitio Web dinámicos. ASP significa en inglés (Active Server Pages), fue liberado por Microsoft en 1996. Las páginas Web desarrolladas bajo este lenguaje es necesario tener instalado Internet Information Server (IIS)[13].

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

1.3.7.1.-Ventajas

1. Usa Visual Basic Script, siendo fácil para los usuarios.
2. Comunicación óptima con SQL Server.
3. Soporta el lenguaje JScript (Javascript de Microsoft).

1.3.7.2.-Desventajas

1. Código desorganizado.
2. Se necesita escribir mucho código para realizar funciones sencillas.
3. Tecnología propietaria.
4. Hospedaje de sitios web costosos.

1.3.8. Lenguaje ASP.NET

Este es un lenguaje comercializado por Microsoft [5], y usado por programadores para desarrollar entre otras funciones, sitios web. ASP.NET es el sucesor de la tecnología ASP, fue lanzada al mercado mediante una estrategia de mercado denominada .NET.

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

1.3.8.1.-Ventajas

1. Completamente orientado a objetos.
2. Controles de usuario y personalizados.
3. División entre la capa de aplicación o diseño y el código.
4. Facilita el mantenimiento de grandes aplicaciones.
5. Incremento de velocidad de respuesta del servidor.
6. Mayor velocidad.
7. Mayor seguridad.

1.3.8.2.-Desventajas

1. Mayor consumo de recursos.

1.3.9. Lenguaje JSP

Es un lenguaje para la creación de sitios web dinámicos, acrónimo de Java Server Pages. Está orientado a desarrollar páginas web en Java. JSP es un lenguaje multiplataforma. Creado para ejecutarse del lado del servidor [13]. JSP fue desarrollado por Sun Microsystems. Comparte ventajas similares a las de ASP.NET, desarrollado para la creación de aplicaciones web potentes. Posee un motor de páginas basado en los servlets de Java. Para su funcionamiento se necesita tener instalado un servidor Tomcat.

1.3.9.1.-Características

1. Código separado de la lógica del programa.
2. Las páginas son compiladas en la primera petición.
3. Permite separar la parte dinámica de la estática en las páginas web.
4. Los archivos se encuentran con la extensión (jsp).
5. El código JSP puede ser incrustado en código HTML.

1.3.9.2.-Elementos de JSP

Los elementos que pueden ser insertados en las páginas JSP son los siguientes:

1. Código: se puede incrustar código "Java".
2. Directivas: permite controlar parámetros del servlet.
3. Acciones: permite alterar el flujo normal de ejecución de una página.

1.3.9.3.-Ventajas

1. Ejecución rápida del servlets.
2. Crear páginas del lado del servidor.
3. Multiplataforma.
4. Código bien estructurado.
5. Integridad con los módulos de Java.
6. La parte dinámica está escrita en Java.
7. Permite la utilización se servlets.

1.3.9.4.-Desventajas

1. Complejidad de aprendizaje.

1.3.10. Lenguaje PYTHON

Es un lenguaje de programación creado en el año 1990 por Guido Van Rossum [9], es el sucesor del lenguaje de programación ABC. Python es comparado habitualmente con Perl. Los usuarios lo consideran como un lenguaje más limpio para programar. Permite la creación de todo tipo de programas incluyendo los sitios web.

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

1. Programación orientada a objetos.
2. Programación estructurada.
3. Programación funcional.
4. Programación orientada a aspectos.

1.3.10.1.-Ventajas

1. Libre y fuente abierta.
2. Lenguaje de propósito general.
3. Gran cantidad de funciones y librerías.
4. Sencillo y rápido de programar.
5. Multiplataforma.
6. Licencia de código abierto (Opensource).
7. Orientado a Objetos.
8. Portable.

1.3.10.2.-Desventajas

1. Lentitud por ser un lenguaje interpretado.

1.3.11. Lenguaje Ruby

Es un lenguaje interpretado de muy alto nivel y orientado a objetos. Desarrollado en el 1993 por el programador japonés Yukihiro "Matz"Matsumoto. Su sintaxis está inspirada en Python, Perl. Es distribuido bajo licencia de software libre (Opensource) [9].

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

1.3.11.1.-Características

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

1.3.11.2.-Ventajas

1. Permite desarrollar soluciones a bajo Costo.
2. Software libre.
3. Multiplataforma.

Capítulo 2

Trabajos Relacionados

Hoy en día existe una variedad de universidades que ya cuentan con sitios web que funcionan como repositorios de tesis a nivel licenciatura, maestría o doctorado. Además hay organizaciones e instituciones que están implementando sistemas que cuentan con cierta similitud en funcionalidad al sitio que se pretende desarrollar.

2.1. ¿Qué es un repositorio de documentos?

Antes de continuar con las aplicaciones de algunos repositorio de tesis se debe de tener en claro el significado de ello.

Un repositorio de documentos de archivo significa una oficina de documento, una biblioteca, un museo o cualquier otra institución que custodia documentos que están abiertos para la inspección pública. (Debe observarse que en el Estándar Británico 5454 'repositorio' significa únicamente el lugar de almacenamiento).

Un repositorio es un sitio centralizado donde se almacena y mantiene información digital, habitualmente bases de datos o archivos informáticos.

El origen de la palabra española repositorio deriva del latín *repositorium*, que significaba armario, alacena. Este significado se generalizó (por metonimia) en español y es recogido en el Diccionario de la Real Academia (DRAE) como: "Lugar donde se guarda algo", y de ahí se aplicó al léxico específico de la informática para designar los repositorio de información digital.

Los repositorios están preparados para distribuirse habitualmente sirviéndose de una red informática como Internet o en un medio físico como un disco compacto. Y pueden ser de acceso público, o pueden estar protegidos y necesitar de una autenticación previa. Los repositorios más conocidos son los de carácter

académico y los institucionales. A diferencia de los ordenadores personales o de las PC de escritorio, los repositorios suelen contar con sistemas de Backup y mantenimiento preventivo y correctivo, lo que hace que nuestra información se pueda recuperar en el caso que nuestra máquina o PC quede inutilizable [7].

2.2. E-LIS Eprints in Library and Information Science (<http://eprints.rclis.org>)

Archivo abierto de carácter internacional especializado en Biblioteconomía, Documentación y disciplinas relacionadas. Su nacimiento se sitúa en 2003, como proyecto financiado por el Ministerio de Cultura, en la Universidad Politécnica de Valencia. Actualmente está alojado en máquinas del Consorcio Interuniversitario Lombardo por Elaborazione Automática (CILEA) en Italia. Ver figura.2.1

Su objetivo es mejorar la comunicación en la disciplina mediante la rápida distribución de documentos. Es el archivo abierto más grande de ésta disciplina con más de 6500 documentos almacenados. Es también el archivo más multilingüe, ya que recoge trabajos en turco, serbio, chino o ruso entre otras muchas lenguas.

E-LIS dispone de una política de remisión de trabajos donde se especifican aspectos como tipo de documentos que se aceptan, lenguas en que se admiten o copyright de los mismos. En un sentido amplio se acepta cualquier trabajo, publicado o no, relacionado con la disciplina, y en cualquier formato, aunque se prefiere el PDF. Igualmente se aceptan en cualquier lengua. Los autores son responsables de garantizar que los documentos depositados están libres de restricciones en cuanto a copyright. E-LIS no se hace responsable de los contenidos remitidos por los autores.

Una característica destacable de E-LIS es el equipo de más de 60 editores de diferentes países que son los encargados de garantizar que los trabajos remitidos se ajustan a la política del repositorio y que los metadatos son tan completos como sea posible, y que, independientemente del idioma empleado, siempre haya un resumen en inglés.



Figura 2.1: Portal especializado en Biblioteconomía, Documentación y disciplinas relacionadas.

2.3. E-PrintsUCM <http://www.ucm.es/eprints>

Se trata del repositorio institucional de la Universidad Complutense de Madrid, de uno de los archivos institucionales más antiguos ya que data de 2003. Entre los objetivos del archivo están el incrementar el acceso y la difusión de la investigación desarrollada en la universidad, así como permitir una mayor visibilidad e impacto de la investigación publicada.

Actualmente almacena más de 4100 documentos de los que la mayor parte son tesis depositadas por la Biblioteca. El software sobre el que está implementado es e-prints, igual que en el caso anterior. Ver figura. 2.2



Figura 2.2: Repositorio institucional de la Universidad Complutense de Madrid.

2.4. - TDX: Tesis Doctorals en Xarxa

TDR (Tesis Doctorales en Red) es un repositorio cooperativo que contiene, en formato digital, las tesis doctorales leídas en las Universidades de Cataluña y de otras comunidades autónomas. Es el repositorio con más tradición de ese país pues el proyecto nació a partir de un convenio firmado en 1999.

Permite la consulta remota del texto completo de las tesis, así como realizar búsquedas por autor, director, título, tema de la tesis, universidad y departamento donde se ha leído, año de defensa, entre otros, figura. 2.3. Los objetivos de este repositorio, gestionado por el Consorcio de Biblioteques Universitàries de Catalunya (CBUC) y el Centre de Supercomputació de Catalunya (CESCA), y patrocinado por la Generalitat de Catalunya, son:

1. Difundir, por todo el mundo y a través de Internet, los resultados de la investigación universitaria.
2. Ofrecer a los autores de las tesis una herramienta que incremente el acceso y la visibilidad de su trabajo.
3. Mejorar el control bibliográfico de las tesis.
4. Impulsar la edición electrónica y las bibliotecas digitales.
5. Incentivar la creación y el uso de la producción científica propia.

Son las propias universidades que participan en TDR las que difunden el repositorio entre sus estudiantes de doctorado, les dan las recomendaciones y la lista de formatos electrónicos aceptados para la redacción de las tesis y (una vez presentadas y aprobadas) las editan y cargan en TDR. Además, a medio plazo, está previsto reconvertir a formato digital algunas tesis ya presentadas durante los últimos años en otros formatos (microfichas o papel).

La consulta de las tesis es libre y no necesita ninguna clave de entrada al sistema. Los derechos del autor quedan protegidos mediante un contrato.



Figura 2.3: Repositorio de Tesis Doctorales en Red.

Capítulo 3

Método

En este capítulo se describirán las herramientas, la metodología y los lenguajes que se implementarán en todo el desarrollo de la aplicación.

3.1. Metodología empleada en el sistema

Para el desarrollo y secuencia de la aplicación se ha implementado la metodología incremental iterativo, que es aquel modelo incremental que combina elementos del modelo cascada (aplicado repetidamente) así como la filosofía iterativa del prototipado.

La parte inicial es el núcleo del producto (es la parte más importante).

Este modelo es aplicable cuando es difícil establecer los requisitos iniciales de un proyecto y es más apropiado para proyectos pequeños. Las nuevas versiones pueden ser planeadas de modo que los requisitos técnicos puedan ser administrados.

El objetivo es trabajar junto al usuario para descubrir sus requisitos de manera incremental antes de que el producto final sea obtenido.

El modelo iterativo de desarrollo de software se basa en que antes de entregar el sistema de una vez, tanto el desarrollo como las entregas se dividen en incrementos.

1. El primer incremento satisface los requisitos más críticos.
2. Los primeros incrementos sirven como prototipo y ayudan en la tarea de detectar los posteriores requisitos.

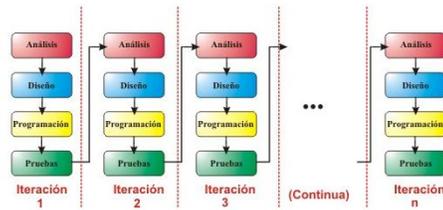


Figura 3.1: Gráfico del modelo iterativo incremental

3.2. Herramientas empleadas en el sistema

Las herramientas utilizadas fueron:

3.2.1. UML

Nos apoyamos de UML que como ya se había mencionado es un lenguaje de modelado que permite la representación conceptual y física de un sistema. Para ésto se utilizó las siguientes herramientas que éste proporciona para llevar a cabo el análisis y diseño del sistema.

Para el análisis nos apoyamos de:

1. Diagrama de casos de uso.
2. Diagrama secuencial.
3. Diagrama de colaboración.
4. Diagrama de escenarios.

Para el diseño se empleó:

1. Diagrama de Estados.
2. Diagrama de Actividades.
3. Diagrama de Objetos.
4. Diagrama de Clases.

3.2.2. DBDesigner

Es un proyecto de gran apoyo para el proceso de análisis y diseño de sistemas, permite el desarrollo de entidad relación y diccionario de datos, generar los script para bases de datos Oracle, SqlServer, MySQL, PostgreSQL y FireBird.

Fue empleado para la realización del diagrama de objetos.

3.2.3. StarUML

StarUML es un proyecto de código abierto para desarrollar rápido, flexible, ampliable, featureful y libremente disponible UML / MDA plataforma se ejecuta en plataforma Win32. El objetivo de la StarUML proyecto es construir una herramienta de modelado de software y también la plataforma que es una sustitución de las herramientas comerciales de UML como Rational Rose, juntos y así sucesivamente.

1. UML 2.0: UML está en continua expansión estándar administrado por OMG (Objeto Grupo de Gestión).
2. MDA (Arquitectura Dirigida por Modelos): MDA es una nueva tecnología introducida por OMG. Para obtener ventajas de la MDA, el software herramienta de modelado debe apoyar muchas variables de personalización. StarUML está diseñado para apoyar a la MDA y ofrece muchas variables como la personalización de perfil UML, Enfoque, Modelo Marco, NX (notación de extensión), la MDA y la plantilla de documento de código y así sucesivamente.
3. Arquitectura plug-in: Muchos usuarios exigen más y más funcionalidades a las herramientas de modelado de software. Para cumplir los requisitos, la herramienta debe tener bien definido el plug-in de plataforma. StarUML proporciona simple y potente plug-in de la arquitectura de modo que cualquier persona puede desarrollar módulos plug-in compatible con COM en lenguajes (C ++, Delphi, C, VB, entre otros)
4. Usabilidad: Es la cuestión más importante en el desarrollo de software. StarUML se implementa para proporcionar al usuario muchas funciones, como amigo rápido de diálogo, la manipulación del teclado, diagrama de visión general, entre otros.

StarUML está escrito principalmente en Delphi, pero cualquier lenguaje de programación pueden ser usado para desarrollar StarUML. (por ejemplo, C / C ++, Java, Visual Basic, Delphi, JScript, VBScript, C, VB.NET, entre otros)

Y en esta aplicación lo ocupamos para la realización de los siguientes diagramas:

1. Creación de casos de uso.
2. Diagrama de secuencia.
3. Diagrama de actividades.
4. Diagrama de colaboración.
5. Diagrama de objetos.

3.2.4. NetBeans

Es una herramienta que permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos.

◦ Características

1. Sistema de ventanas práctico para desarrollar las interfaces de usuario.
2. Sistema de ficheros virtual en el cual se montan los diferentes módulos con el cual se van adaptando automáticamente los menús, barra de herramientas, menús contextuales, entre otros, de la aplicación.
3. Su licencia permite construir tanto aplicaciones open source como comerciales.
4. Compatibilidad con Java Web Start.
5. No es obligatorio que una aplicación deba tener interfaz de usuario gráfica (GUI), ya que la plataforma permite dejar de lado la misma y seguir disfrutando del resto de los beneficios, por ejemplo la actualización de módulos desde un repositorio remoto.
6. Soporte completo para desarrollar desde NetBeans IDE, por lo que no se necesitará otra herramienta adicional para el desarrollo

3.2.5. Adobe Fireworks

Herramienta desarrollada para el manejo híbrido de gráficos vectoriales con Gráficos en mapa de bits y que ofrece un ambiente eficiente para la creación rápida de prototipos de sitios Web e interfaces de usuario como para la creación y Optimización de Imágenes para web. Originalmente fue desarrollado por Macromedia.

Permiten buscar el equilibrio entre la máxima calidad de imagen y el mínimo tamaño de compresión. Fireworks se integra con fluidez con Dreamweaver y Flash y proporciona las condiciones ideales para crear y optimizar imágenes sin perder información ni tiempo con la edición gracias a la compatibilidad de ida y vuelta.

3.2.6. Dreamweaver

Herramienta de diseño que se utilizó para el diseño y la programación de las páginas Web del repositorio de proyectos desarrollado, elegido por sus funcionalidades, su integración con otras herramientas como Adobe Flash y, recientemente, por su soporte de los estándares del World Wide Web Consortium.

La gran ventaja de este editor sobre otros es su gran poder de ampliación y

personalización del mismo, puesto que en este programa, sus rutinas (como la de insertar un hipervínculo, una imagen o añadir un comportamiento) están hechas en Javascript-C, lo que le ofrece una gran flexibilidad en estas materias. Esto hace que los archivos del programa no sean instrucciones de C++ sino, rutinas de Javascript que hace que sea un programa muy fluido, que todo ello hace, que programadores y editores web hagan extensiones para su programa y lo pongan a su gusto.

3.2.7. Flash

Flash es la tecnología más comúnmente utilizada en el Web que apoyo a este repositorio en la creación de animaciones, ya que permite realizarlas con poco peso, es decir, que tardan poco tiempo en ser cargadas por el navegador.

Se eligió por ser una aplicación en forma de estudio de animación que trabaja sobre "Fotogramas"destinado a la producción y entrega de contenido interactivo para diferentes audiencias alrededor del mundo sin importar la plataforma.

3.2.8. XAMPP

Se utilizó como servidor, en el se construyó la base de datos MySQL del repositorio de proyectos,ya que se empleó el servidor Web Apache y los intérpretes para lenguajes de script: PHP y Perl.

3.2.9. MySQL

◦ Las características principales de MySQL

1. Es un gestor de base de datos. Una base de datos es un conjunto de datos y un gestor de base de datos es una aplicación capaz de manejar este conjunto de datos de manera eficiente y cómoda.
2. Es una base de datos relacional. Una base de datos relacional es un conjunto de datos que están almacenados en tablas entre las cuales se establecen unas relaciones para manejar los datos de una forma eficiente y segura. Para usar y gestionar una base de datos relacional se usa el lenguaje estándar de programación SQL.
3. Es Open Source. El código fuente de MySQL se puede descargar y está accesible a cualquiera, por otra parte, usa la licencia GPL para aplicaciones no comerciales.
4. Es una base de datos muy rápida, segura y fácil de usar. Gracias a la colaboración de muchos usuarios, la base de datos se ha ido mejorando optimizándose en velocidad. Por eso es una de las bases de datos más usadas en Internet.

3.3. Lenguajes de programación usados en el sistema

Para la realización de esta aplicación fueron empleados los siguientes lenguajes de programación:

3.3.1. Html

HTML se utilizó para crear las páginas Web. Este lenguaje indica cómo debe mostrarse el contenido, facilitando el diseño al realizarlas y el control de los requisitos que debe de cumplir una página Web.

3.3.2. Java

Java es un lenguaje de programación orientado a objetos y al desarrollo de software con el cual se realizó este repositorio. Fue empleado para realizar las conexiones entre la interfaz gráfica y la base de datos.

Finalmente concluimos que Java nos brinda una solución para cada necesidad que pudiéramos tener.

3.3.3. JSP

Fue utilizada por el repositorio por ser una de las tecnologías de Java que permitió generar contenido dinámico para las páginas Web empleadas en el sistema. Facilitó la utilización de código Java mediante la construcción de scripts.

El motor de las páginas JSP está basado en los servlets de Java -programas en Java destinados a ejecutarse en el servidor-, aunque el número de desarrolladores que pueden afrontar la programación de JSP es mucho mayor, dado que resulta mucho más sencillo aprender que los servlets.

Capítulo 4

Análisis

Para conocer el proceso de registro de los temas de tesis que los estudiantes pretenden investigar y todos los elementos que se ven involucrados en esta acción, fue necesario realizar una serie de actividades para recopilar la mayor información, por ejemplo se realizó una entrevista a la persona encargada de dar de alta a los alumnos en el registro de su tema de tesis, para ello se realizó una serie de preguntas que arrojó información necesaria para estructurar un nuevo modelo a seguir.

Dicho cuestionario que se empleó es el siguiente:

CUESTIONARIO

- 1.-¿CUÁLES SON LOS REQUISITOS PARA EL REGISTRO DE UNA TESIS?
- 2.-¿DE CUÁNTAS ETAPAS ESTÁ COMPUESTO EL REGISTRO DE UNA TESIS?
- 3.-¿CÓMO SE CONTROLAN LOS REGISTROS?
- 4.-¿CUÁNTO TIEMPO TARDA EN VALIDARSE UN TEMA DE TESIS ELEGIDO POR UN ALUMNO?
- 5.-¿CÓMO SE DAN A CONOCER LOS TEMAS DE INVESTIGACIÓN PROPUESTOS POR LOS PROFESORES?
- 6.-¿SE LLEVA UN CONTROL DE ESOS TEMAS?
- 7.-¿CADA CUANTO SE ACTUALIZAN ESOS TEMAS?
- 8.-¿CUÁL ES EL PORCENTAJE ESTIMADO DE ALUMNOS QUE SE TITULAN POR TESIS?
- 9.-¿CUÁL ES EL PORCENTAJE ESTIMADO DE ALUMNOS QUE REGISTRAN SU TESIS Y LA ABANDONAN?
- 10.-¿QUÉ SUCEDE SI UN ALUMNO ABANDONA SU TESIS?
- 11.-¿CUÁL ES EL TIEMPO ESTIMADO SOBRE EL PROCESO DEL REGISTRO?
- 12.-¿CUÁL ES EL TIEMPO ESTIMADO PARA CONCLUIR LA TESIS DESPUÉS DE SER REGISTRADA?
- 13.-¿QUE FORMATOS SE UTILIZAN PARA EL REGISTRO?

Del cual se obtuvo lo siguiente:

- 1.- Solo es necesario llenar un formato de solicitud de registro de tesis.
- 2.-Este proceso solo consta de dos etapas, el registro del tema del alumno y la autorización o negación de éste.
- 3.- El control de dichos registros son actualmente manejados por medio de archivos en ficheros (papel y lápiz).
- 4.- El tiempo estimado es de 10 días pero puede variar, como se observa en la gráfica figura.4.1 es mas frecuente que tarde 10 días todo el proceso.



Figura 4.1: Gráfica del tiempo en que tarda en validarse un tema de tesis

- 5.- En la mayoría de los casos los estudiantes se han enterado de las propuestas de temas de investigación donde podrían realizar su tesis coordinadas por profesores que así lo han solicitado, por medio de carteles que la coordinación de la carrera pega en las vidrieras y/o pizarrones informativos, otros tantos lo han hecho porque ellos han ido a informarse personalmente a las oficinas encargadas de este trámite y una minoría por medio de folletos que encuentran en el instituto figura.4.2.



Figura 4.2: Divulgación de los temas de tesis propuestos por profesores

6.- La coordinación tiene la obligación y el deber de tener un control de los datos de los temas que se encuentran disponibles así como de los ya realizados para que el estudiante pueda realizar satisfactoriamente la elección de su tema de tesis con el cual procederá su titulación.

7.- El control que se tiene sobre esta información hace innecesario la actualización de los datos que se emplean, ya que si se ingresa, modifica o elimina algún tipo de dato, en ese mismo instante se libera el espacio y se corrigen los archivos.

8.-En los últimos años desde que se implemento la modalidad de titularse por el examen general de conocimiento mucho alumnos han optado por titularse por este medio, dejando muy por de bajo el porcentaje de los alumnos que eligen titularse por elaboración de tesis y en un nivel un poco mas arriba el de otras modalidades figura.4.3.



Figura 4.3: Porcentajes de alumnos titulados por tesis

9.-Lamentablemente este tipo de modalidad para titularse (por tesis) ha dejado de ser concurrida y de un aproximado de 10 alumnos que registran su tesis solo un cuarenta por ciento la llegan a terminar y mas de la mitad la abandonan por la posibilidad de titularse de otra forma figura.4.4.



Figura 4.4: Porcentaje de alumnos que registran su tesis

10.-Si un alumno llega a dejar su tesis no se le aplica ninguna amonestación solo lo borran del archivo y se sigue con otro proceso.

11.-El rango de tiempo para culminar con la tesis no debe exceder los años máximos que tiene el estudiante para titularse (2 años). De ahí el alumno puede terminar en el tiempo que el lo decida.

Para el modelaje y estructura de los datos utilizamos parte de la metodología UML, construyendo diagramas que nos permitan comprender de manera visual el flujo de cada proceso, en este caso implementamos el diagrama de escenarios, que describe de forma lógica cada paso que se lleva a cabo en el proceso. Es importante que antes de elaborar cualquier tipo de diagrama, primero hay que identificar los elementos que van a intervenir en el proceso, así como sus atributos correspondientes.

4.1. Diagramas de Escenarios

Este apartado nos permitirá observar cual es la secuencia de los sucesos que intervienen en todo el proceso de registro de una tesis.

Un escenario es una secuencia de sucesos. Los sucesos se producen siempre que se intercambia información entre un objeto del sistema y un agente externo.

Por lo tanto, este análisis nos arrojó los siguientes escenarios:

4.1.1. Diagrama de escenario de consulta de tesis en proceso

1. El sistema muestra la interfaz de consulta
2. El sistema muestra las categorías de tesis terminadas, en proceso y temas propuestos
3. El usuario elige la opción de tesis en proceso
4. El sistema muestra el tipo de búsqueda (materia, tema)
5. El cliente ingresa datos de la búsqueda
6. El sistema muestra la lista de la búsqueda
7. El usuario consulta.
8. El usuario termina la aplicación

4.1.2. Diagrama de escenario de consulta de tesis terminada

1. El sistema muestra la interfaz de consulta
2. El sistema muestra las categorías de tesis terminadas, en proceso y temas propuestos
3. El usuario elige la opción de temas propuestos
4. El sistema muestra el tipo de búsqueda (materia, tema)
5. El cliente ingresa datos de la búsqueda
6. El sistema muestra la lista de la búsqueda
7. El usuario consulta.
8. El usuario termina la aplicación

4.1.3. Diagrama de escenario pre-registro

1. El usuario ingresa a la a la interfaz gráfica de la aplicación
2. El usuario selecciona la opción pre- registro
3. El sistema muestra la interfaz de pre- registro
4. El sistema pide información del usuario
5. El usuario ingresa datos
6. El usuario envía o cancela el proceso
7. El usuario cierra el sistema

4.1.4. Diagrama de escenario de administrador modificar/eliminar

1. El usuario ingresa a la interfaz gráfica de la aplicación
2. El usuario selecciona la opción de administrador
3. El sistema solicita identificación de usuario
4. El usuario se identifica
5. El sistema valida y muestra la interfaz del administrador
6. El usuario elige opciones de revisar pre-registro, modificar, eliminar y dar de altas tesis terminadas
7. El sistema muestra la información
8. El usuario realiza las operaciones correspondientes
9. El usuario cierra el sistema

4.1.5. Diagrama de escenario de administrador

1. El usuario ingresa a la interfaz gráfica principal
2. El usuario elige crear nueva cuenta
3. El sistema solicita información al usuario
4. El usuario llena los campos vacíos
5. El sistema valida la información
6. El usuario envía o cancela la información en el proceso
7. El sistema la guarda en la base de datos
8. El usuario termina el proceso

4.1.6. Diagrama de escenario iniciar sesión

1. El usuario ingresa a la aplicación
2. El usuario ingresa a la opción de iniciar sesión
3. El sistema pide identificación de usuario
4. El usuario ingresa la información solicitada
5. El sistema valida la información
6. El usuario accede a su cuenta
7. El sistema muestra la información personal, de tesis y estado de proceso
8. El usuario cierra sesión

4.2. Diagrama de caso de uso

Con este tipo de diagramas hemos elaborado los siguientes modelos que ejemplifican las fases que sufre el sistema y en cada una de ellas se describen las acciones desde el punto de vista del usuario.

4.2.1. Diagrama de caso de uso sobre las actividades del alumno

El siguiente diagrama (figura 4.5) describe el modo de acceso para realizar el pre registro de tema de tesis y para ello es necesario consultar los temas que ya han sido dados de alta para verificar que su tema no se encuentre registrado.

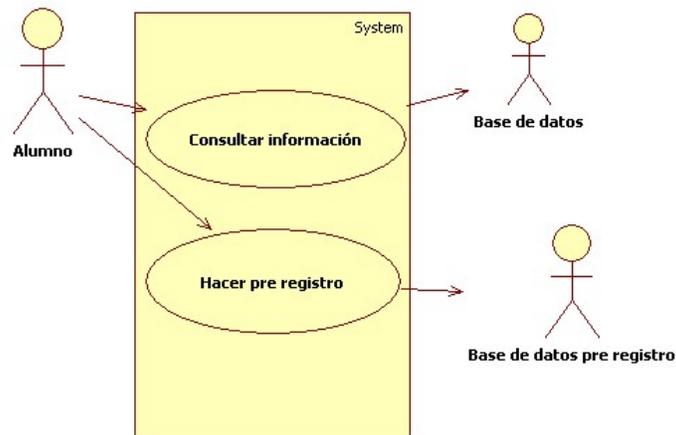


Figura 4.5: Caso de uso sobre las actividades del alumno.

4.2.2. Diagrama de caso de uso sobre las actividades del profesor

El siguiente diagrama (figura 4.6) muestra el caso de uso donde el profesor podrá realizar un pre registro para dar de alta algún tema o temas de investigación que propone como alternativa a los alumnos para que puedan desarrollar un trabajo de tesis.

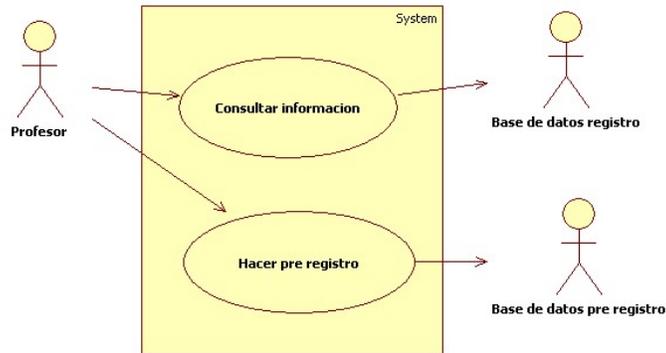


Figura 4.6: Caso de uso sobre las actividades del profesor.

4.2.3. Diagrama de caso de uso sobre las actividades generales del alumno

A continuación se presenta el diagrama (figura 4.7) que muestra el caso de uso donde el administrador realice las operaciones de eliminación, modificación, consulta y altas.

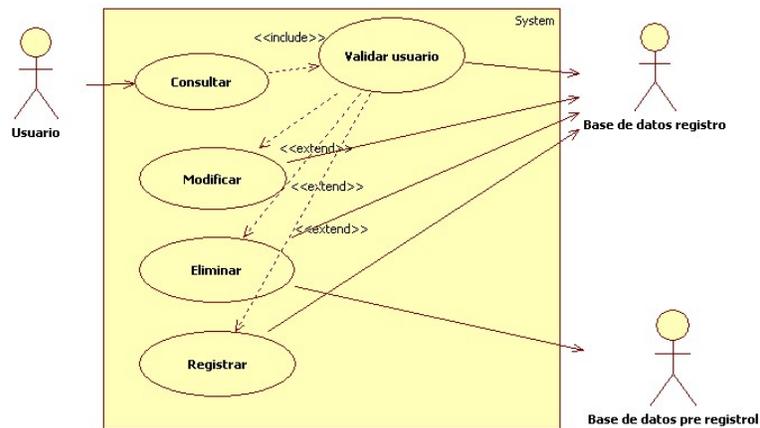


Figura 4.7: Caso de uso de las actividades generales del alumno.

4.2.4. Diagrama de caso de uso sobre el proceso de consulta

En este caso de uso el actor denominado alumno solicita consultar el registro en la base de datos, lo hace mediante una búsqueda que él eligirá, para finalmente obtener la información que requiere, como se muestra en la figura 4.8:

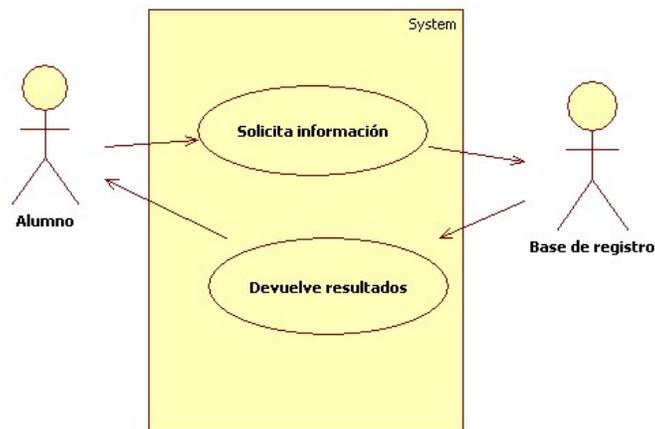


Figura 4.8: Caso de uso sobre el proceso de consulta.

4.2.5. Diagrama de caso de uso sobre el proceso de eliminación de información

En el siguiente diagrama (figura 4.9) se ejemplifica el proceso de eliminación de información por parte del administrador del sistema (actor), desde que consulta la base de datos y elimina la información que sea requerida, para que finalmente la envíe y la guarde nuevamente en la base de datos.

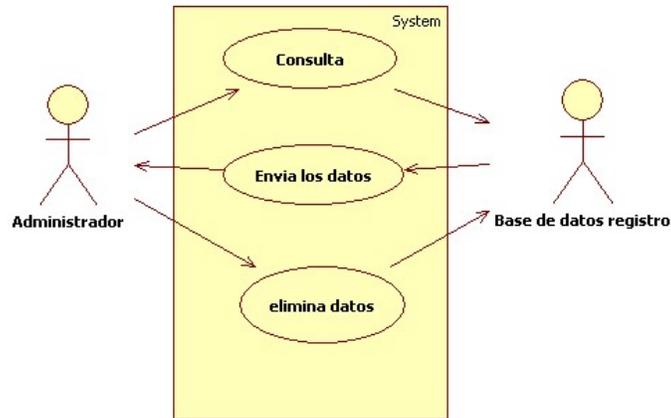


Figura 4.9: Caso de uso del proceso de eliminación de información.

4.2.6. Diagrama de caso de uso sobre el proceso de modificación del registro

La figura 4.10 muestra el caso de uso donde es necesario que se modifiquen algunos datos y para ello se necesita que el administrador lleve a cabo la consulta en el registro los datos del alumno a modificar, realice cambios y guarde.

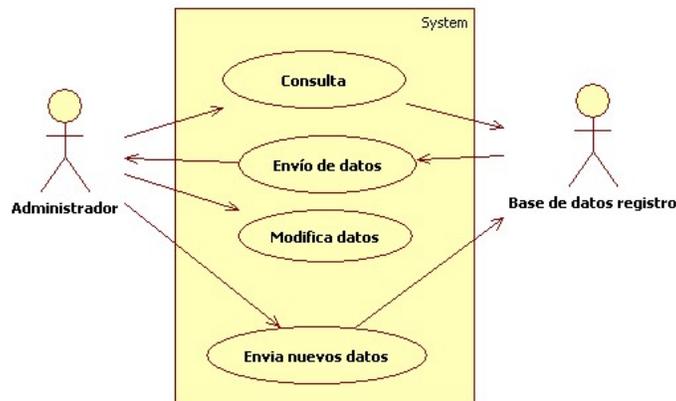


Figura 4.10: Caso de uso sobre el proceso de modificación del registro.

4.2.7. Diagrama de caso de uso sobre el proceso de pre-registro

La figura 4.11 muestra el caso de uso para llevar a cabo el pre-registro del tema de tesis, donde intervienen tres actores principales, el alumno, la base de datos registro y base de datos pre registro en el cual es necesario que el alumno consulte el registro para verificar que su tema propuesto no se encuentre dado de alta en la base de datos por otra persona y de esta manera pueda iniciar el proceso de pre registro.

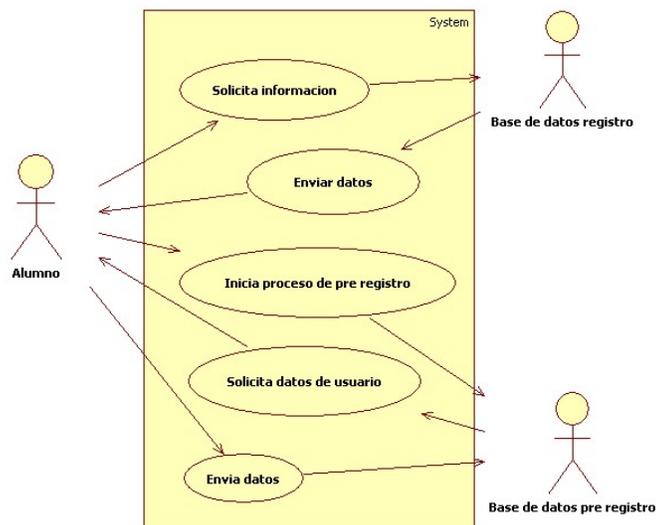


Figura 4.11: Caso de uso sobre el proceso del pre-registro.

4.2.8. Diagrama de caso de uso de un pre-registro inválido

En este caso de uso intervienen tres actores siendo el administrador, el alumno y la base de datos pre registro en donde el administrador consulta el pre registro y verifica la información de los alumnos existentes y si la información es incorrecta enviará un mensaje al alumno donde sugiere corregir errores (figura 4.12).

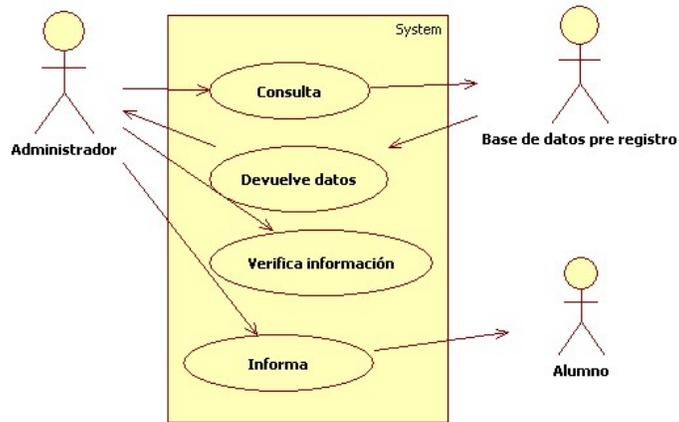


Figura 4.12: Caso de uso sobre proceso involucrado cuando en el pre-registro resulta no valido

4.2.9. Diagrama de caso de uso de un pre-registro valido

En este caso de uso como en el anterior se ven involucrados tres actores: el administrador, alumno y la base de datos pre registro en donde el administrador consulta el pre registro y verifica la información de los alumnos existentes y si la información es correcta enviará a la base de datos de registros y se notificará al usuario que fue aceptado el tema (figura 4.13).

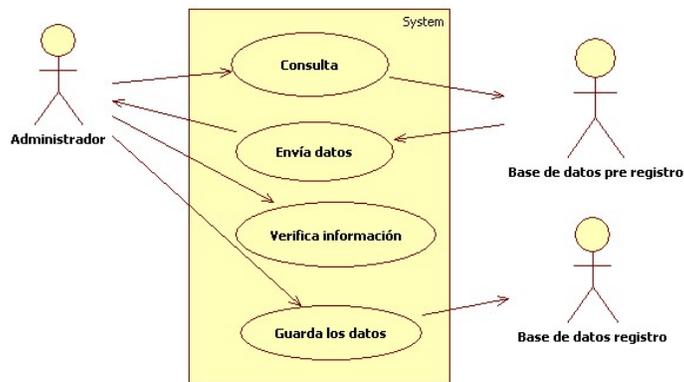


Figura 4.13: Caso de uso sobre proceso involucrado cuando en el pre-registro resulta válido.

4.3. Diagrama de secuencia

En este apartado observaremos una serie de diagramas de secuencia que muestra la mecánica de interacción con base en los tiempos de ejecución de cada acción en cada uno de los proceso que se desarrollaran en la aplicación.

4.3.1. Diagrama de secuencia de pre-registro

La figura 4.14 muestra el diagrama para realizar el proceso de pre registro donde la secuencia inicia con el acceso a la aplicación para que el sistema proporcione el formato para llenar campos con información requerida y una vez ingresado todos los datos el sistema envía y guarda la información .

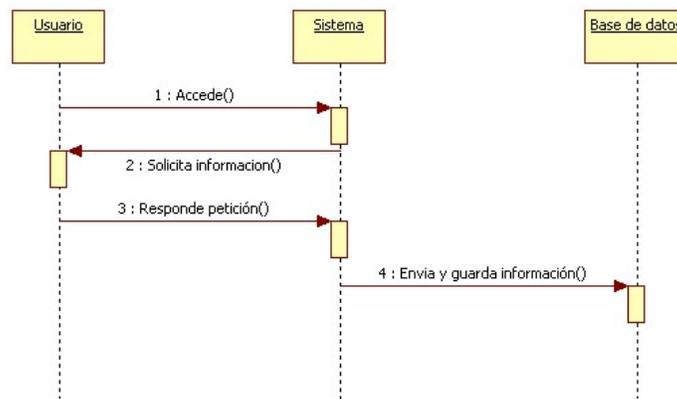


Figura 4.14: Diagrama de secuencia para el proceso de pre-registro.

4.3.2. Diagrama de secuencia del proceso de iniciar sesión

El siguiente diagrama (figura 4.15) muestra la secuencia para realizar el proceso de iniciar sesión de usuario en la aplicación, donde el alumno podrá revisar el proceso de su trámite así como información personal.

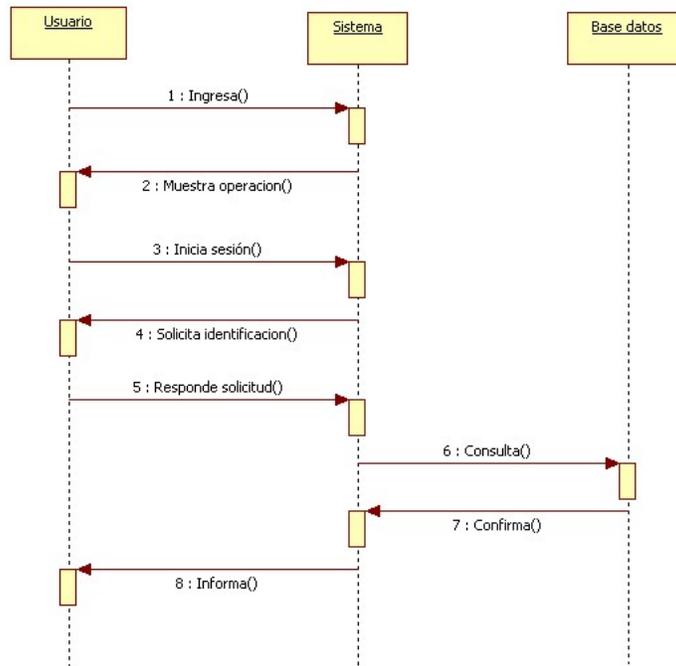


Figura 4.15: Diagrama de secuencia para el proceso de iniciar sesión.

4.3.3. Diagrama de secuencia del proceso de creación nueva cuenta

La figura 4.16 describe cada paso que se emplea al momento de generar una nueva cuenta de usuario, los pasos son los mostrados a continuación:

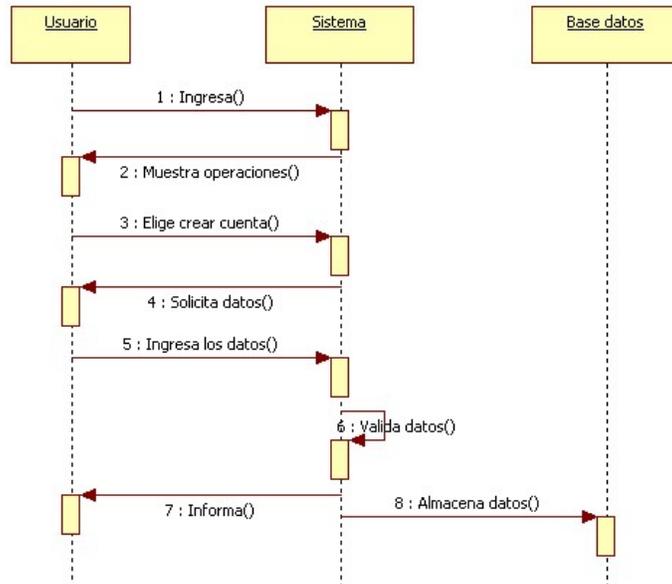


Figura 4.16: Diagrama de secuencia para el proceso de creación de nueva cuenta.

4.3.4. Diagrama de secuencia del registro

La figura 4.17 muestra el diagrama de secuencia que describe el flujo principal de datos y actividades para llevar a cabo el registro de una tesis.

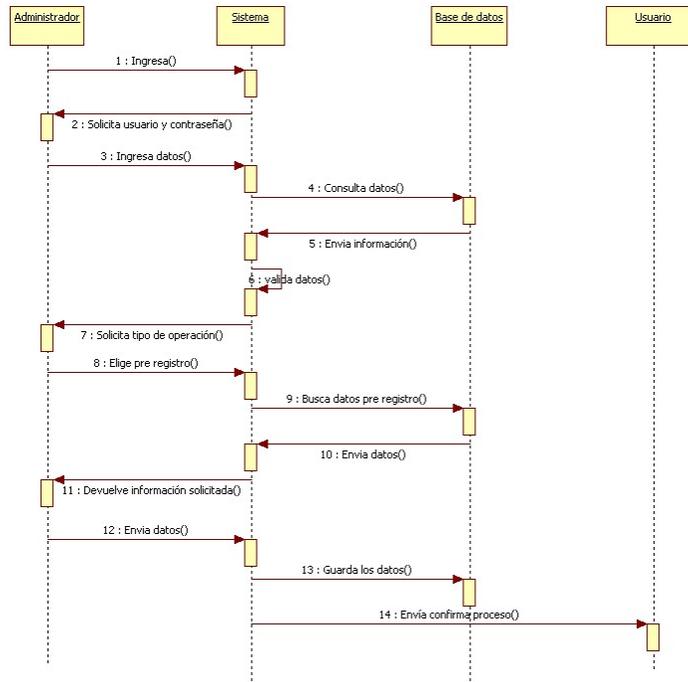


Figura 4.17: Diagrama de secuencia para el proceso de registro.

4.3.5. Diagrama de secuencia de eliminación y modificación

Con la figura 4.18 se muestra el diagrama de secuencia para realizar operaciones de consulta, eliminación o modificación de información por parte de un administrador.

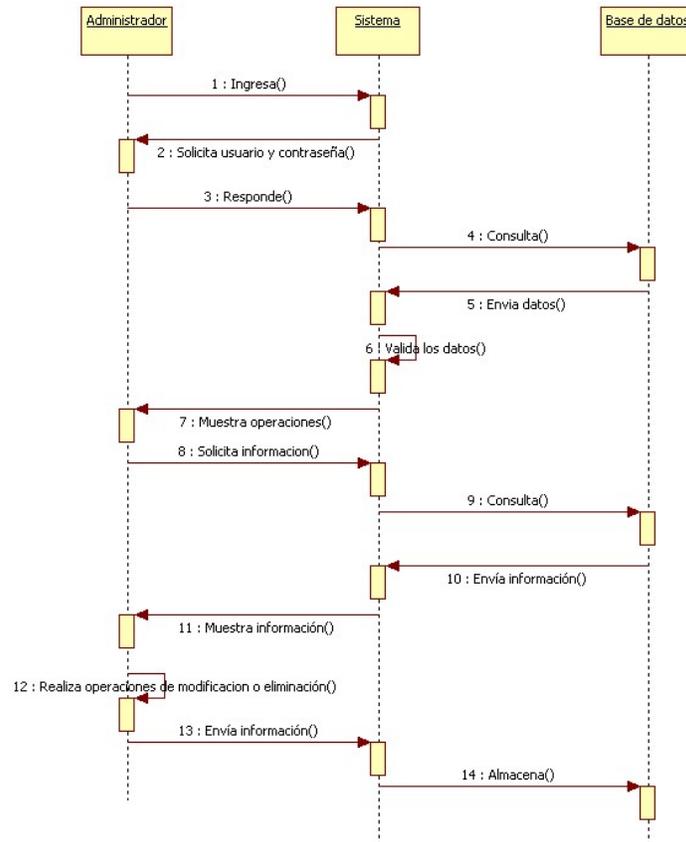


Figura 4.18: Diagrama de secuencia para el proceso de eliminación y modificación.

4.3.6. Diagrama de secuencia de consulta

El siguiente diagrama (figura 4.19) describe el flujo principal para realizar consultas en la aplicación.

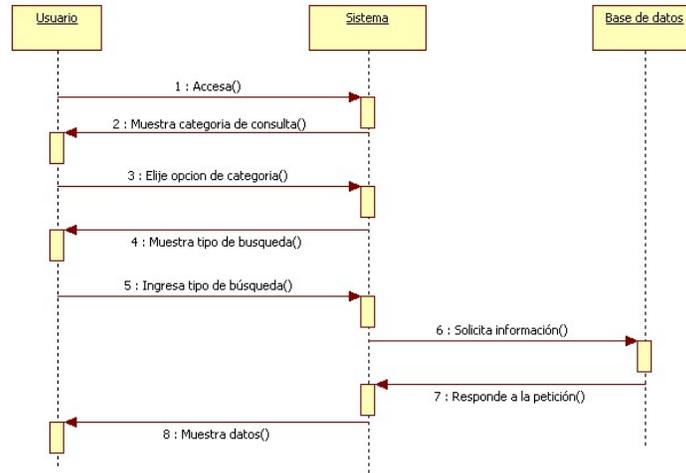


Figura 4.19: Diagrama de secuencia para el proceso de consulta.

4.4. Diagrama de Colaboración

Los siguientes diagramas muestran las relaciones entre los diferentes objetos que se emplean en esta aplicación, así como los mensajes que circulan entre ellos. Todo esto con el fin de brindar un panorama desde otra perspectiva de los procesos que se van a manejar en este sistema.

4.4.1. Diagrama de Colaboración de la consulta

Con el siguiente diagrama (figura 4.20) mostramos el escenario donde actúan los objetos: usuario, sistema y base de datos y sus interacciones entre si, ejemplificando el proceso que se realiza cuando se solicita y se ejecuta una consulta.

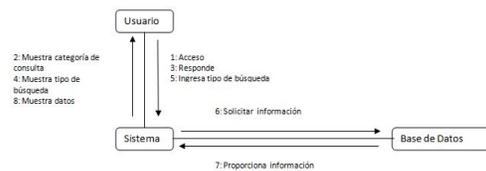


Figura 4.20: Diagrama de colaboración referente al proceso de consulta.

4.4.2. Diagrama de Colaboración del proceso de creación de nueva cuenta

Este diagrama (figura 4.21) muestra el escenario donde los objetos usuario y sistema se comunican entre sí para llevar a cabo el proceso de la creación de una nueva cuenta de usuario, cuando así se requiera.

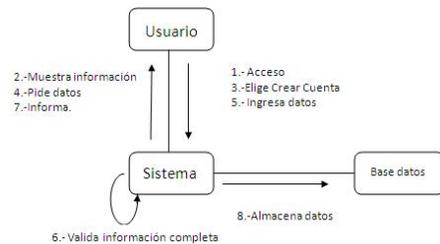


Figura 4.21: Diagrama de colaboración referente al proceso de creación de nueva cuenta.

4.4.3. Diagrama de Colaboración del proceso de pre-registro

La siguiente imagen (figura 4.22) ejemplifica todo el proceso que esta aplicación realizará cuando el usuario entre en el apartado de realizar un pre-registro. En ésta actúan los objetos: usuario, sistema y la base de datos finalmente guardará la información que se obtenga de la interacción de dichos objetos.

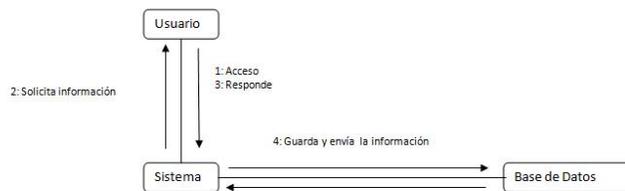


Figura 4.22: Diagrama de colaboración referente al proceso de pre-registro.

4.4.4. Diagrama de Colaboración del proceso de registro

A continuación mostramos el diagrama de colaboración (figura 4.23) que muestra el proceso de registro. En él intervienen el administrador, el sistema, la base de datos y el usuario que desea realizar el proceso y se observan las interacciones entre ellos para la realización satisfactoria de estas actividades.

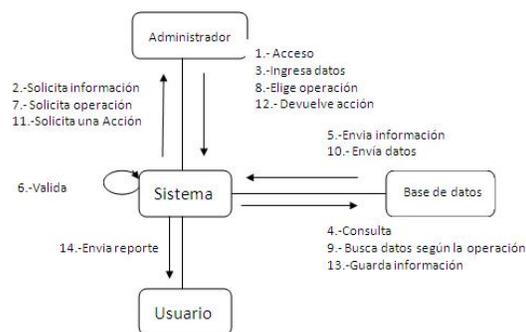


Figura 4.23: Diagrama de colaboración referente al proceso de registro.

4.4.5. Diagrama de Colaboración de los procesos de eliminación y modificación de la información

En el diagrama que se observa en la parte inferior de este texto (figura 4.24), mostramos el escenario que ejemplifica al proceso de eliminación y modificación de la información por parte del administrador, con las respectivas relaciones y dependencias entre los actores u objetos.

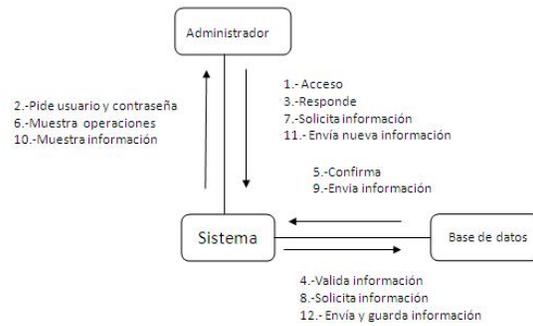


Figura 4.24: Diagrama de colaboración referente al proceso de eliminación y modificación.

4.4.6. Diagrama de Colaboración del proceso de iniciar sesión

A continuación se muestra el diagrama de colaboración que ejemplifica el escenario que emplea el proceso de iniciar sesión por parte del usuario, donde también se observan las relaciones entre el usuario, sistema y base de datos respectivamente (figura 4.25).

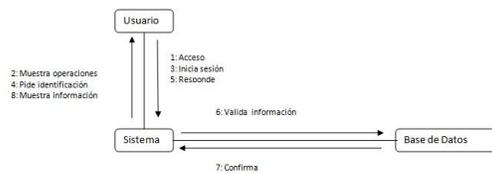


Figura 4.25: Diagrama de colaboración referente al proceso de iniciar sesión.

Capítulo 5

Diseño

En este capítulo se realiza el prototipo de diseño del sistema, apoyándose de UML . Por ello se describe a cada proceso que se realizará en la aplicación utilizando el diagrama de estado, los diagramas de actividades y el de clases, siendo observados y analizados desde todos los posibles puntos de vista.

5.1. Diagrama de Actividades

A continuación se describen los diagramas de actividades que han sido diseñados para mostrar una visión simplificada de lo que ocurre durante una operación o proceso que realizará la aplicación de acuerdo a los requerimientos del usuario.

5.1.1. Diagrama de Actividades del proceso de pre-registro

Este diagrama (figura 5.1) describe las actividades a realizar para llevar a cabo el pre-registro del tema de tesis, donde inicialmente seleccionamos la operación hacer pre-registro, posteriormente es necesario ingresar los datos que el sistema solicita al usuario quien podrá completar la información, enviarla y guardarla o bien tendrá la facultad de cancelar esta operación.

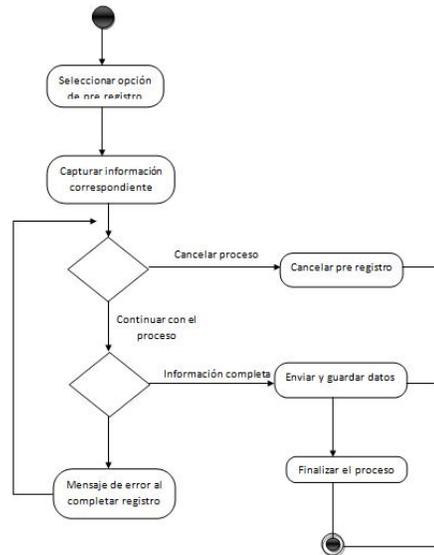


Figura 5.1: Diagrama de actividades del proceso de preregistro.

5.1.2. Diagrama de Actividades del proceso de registro

El diagrama de actividades para dar de alta a un tesista consiste en que el usuario denominado administrador ingresará su nombre de identificación y contraseña, si los datos son correctos se tendrá acceso al sistema, como primer paso se consulta el pre-registro, se verifica la información existente de los alumnos que han realizado la solicitud del registro y de acuerdo a la veracidad de los datos el administrador podrá enviar y guardar los datos en el registro e informa al usuario que su tema ya ha sido dado de alta. En caso de que la información del pre-registro sea faltante o no sea correcta el administrador envía un mensaje al usuario sugiriendo modificar sus datos. Una vez terminadas estas actividades el administrador podrá hacer un nuevo registro o cerrar la aplicación (figura 5.2).

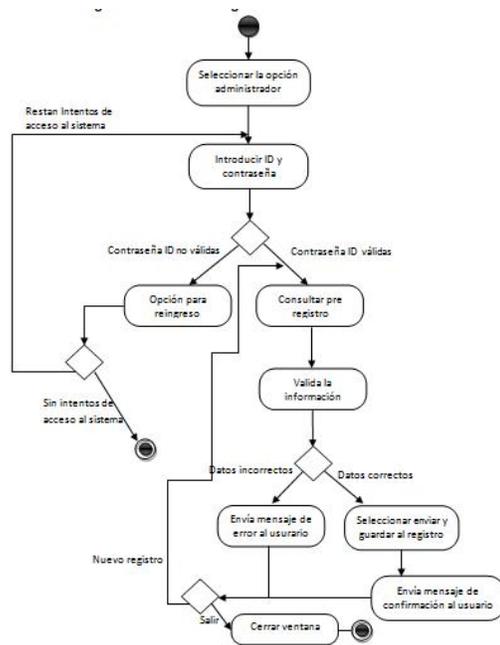


Figura 5.2: Diagrama de actividades del proceso de registro.

5.1.3. Diagrama de Actividades del proceso de creación de cuenta de usuario

Para crear una cuenta de usuario elegimos la operación nueva cuenta donde el usuario ingresará sus datos correspondientes y tendrá como opción cancelar esta función y salir o bien enviar la información completada, guardarla y finalizar el proceso, en caso de que los datos estén incompletos el sistema enviará un mensaje de error donde informa al usuario que debe completar la información para poder continuar con el proceso, observe la figura 5.3.

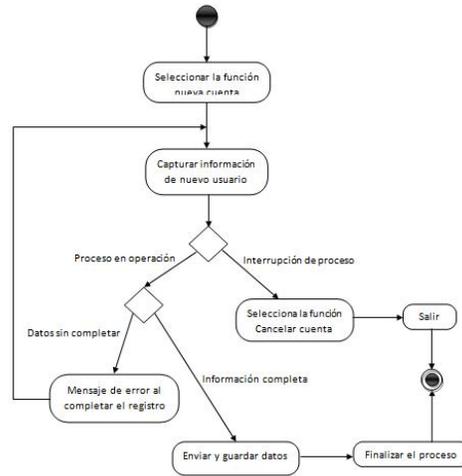


Figura 5.3: Diagrama de actividades del proceso de creación de cuenta de usuario

5.1.4. Diagrama de Actividades del proceso de modificación

Este diagrama (figura 5.4) despliega las actividades para modificar datos del registro, inicialmente se abre la aplicación y se accesa al sistema como administrador quien puede elegir la opción de modificar los datos del registro de profesor o alumno según sea el caso, el sistema solicita el numero de cuenta de alumno a modificar o bien el numero de matricula del profesor, se realizan las respectivas modificaciones y se guardan los cambios, si el administrador desea cancelar el proceso solo elige la opción de cancelar y cerrar la aplicación.

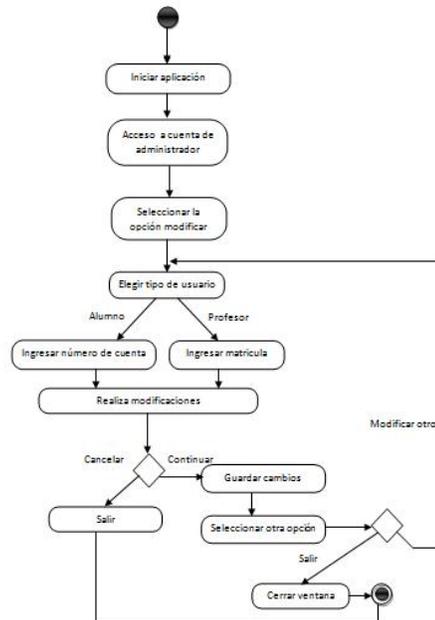


Figura 5.4: Diagrama de actividades del proceso de modificación.

5.1.5. Diagrama de Actividades del proceso de eliminación

Este diagrama (figura 5.5) despliega las actividades para eliminar datos del registro, inicialmente se abre la aplicación y se accesa al sistema como administrador quien puede elegir la opción de eliminar los datos del registro de profesor o alumno según sea el caso, el sistema solicita el numero de cuenta del alumno o matrícula del profesor a eliminar y el administrador elige la opción de continuar si quiere eliminar definitivamente el registro o bien cancelar esta operación y salir de la aplicación.

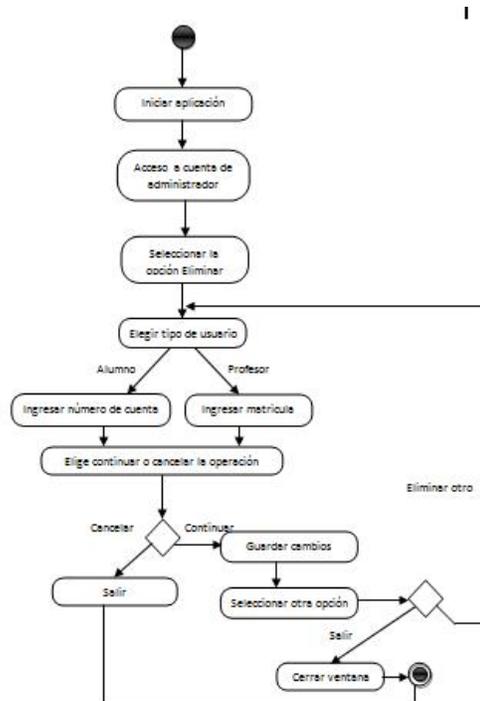


Figura 5.5: Diagrama de actividades del proceso de eliminación.

5.1.6. Diagrama de Actividades del proceso de consulta

Las actividades de consulta inician cuando el usuario elige esta operación en la cual es necesario indicar el estado de los temas a consultar, en este caso puede ser tesis en proceso o bien tesis concluidas, una vez indicada esta actividad se elige que tipo de consulta, puede ser por tema o por proyectos, posteriormente se ingresan los datos con los que se desea obtener resultados de búsqueda. Dados los resultados el usuario podrá realizar una nueva búsqueda o bien salir de la función, figura 5.6.

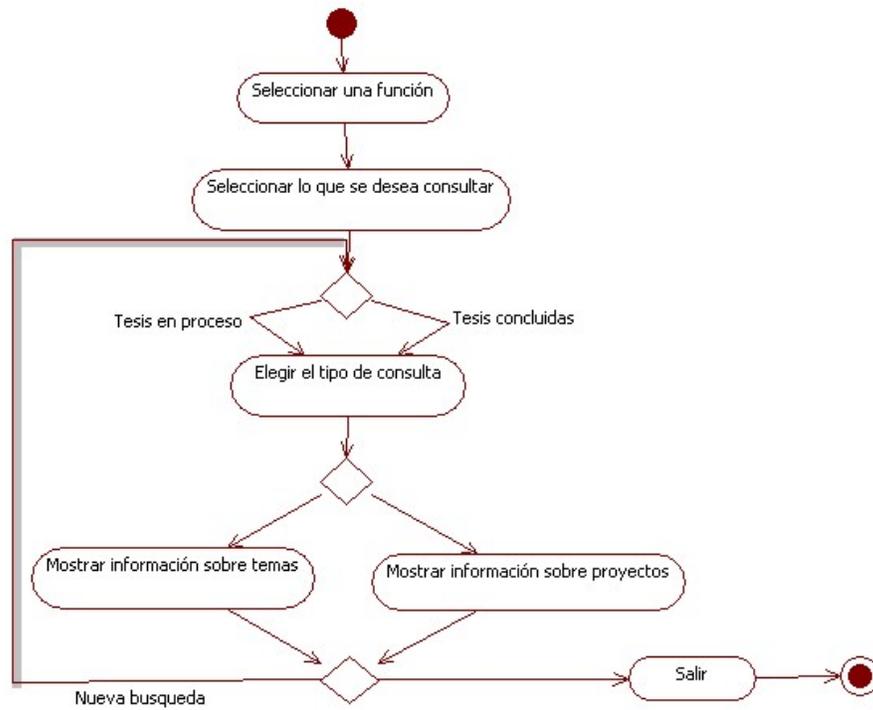


Figura 5.6: Diagrama de actividades del proceso de consulta.

5.2. Diagrama de Estado

En este apartado se muestra el diagrama que representa los diferentes estados de la aplicación en desarrollo en relación a todos los procesos que contendrá, y los estímulos que provocan los cambios de estado en cada objeto, todo considerando los diversos procesos que contiene el sistema, observe la figura 5.7.

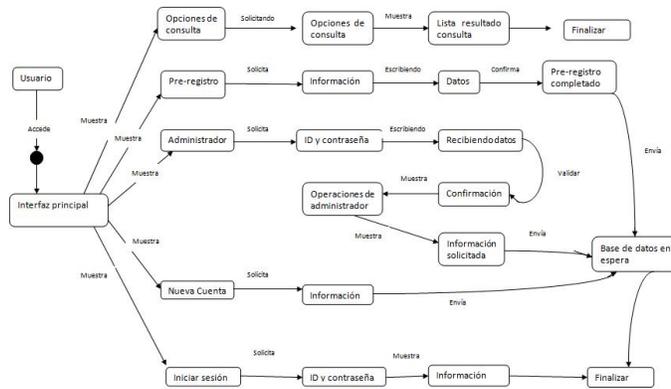


Figura 5.7: Diagrama de Estado.

5.3. Diagramas de Objetos

Basándonos en las normas que establece el modelo de objetos, se diseñó el siguiente diagrama que muestra las diferentes instancias de las clases que se utilizarán en el desarrollo de la aplicación.

5.3.1. Diagrama de Objetos de la aplicación

La figura 5.8 le muestra la forma en que UML representa a los objetos, en este caso estamos trabajando con nueve y como se observa todos se encuentran relacionados con el objetivo de interactuar y funcionar en conjunto para que el sistema realice con eficiencia y eficacia su función.

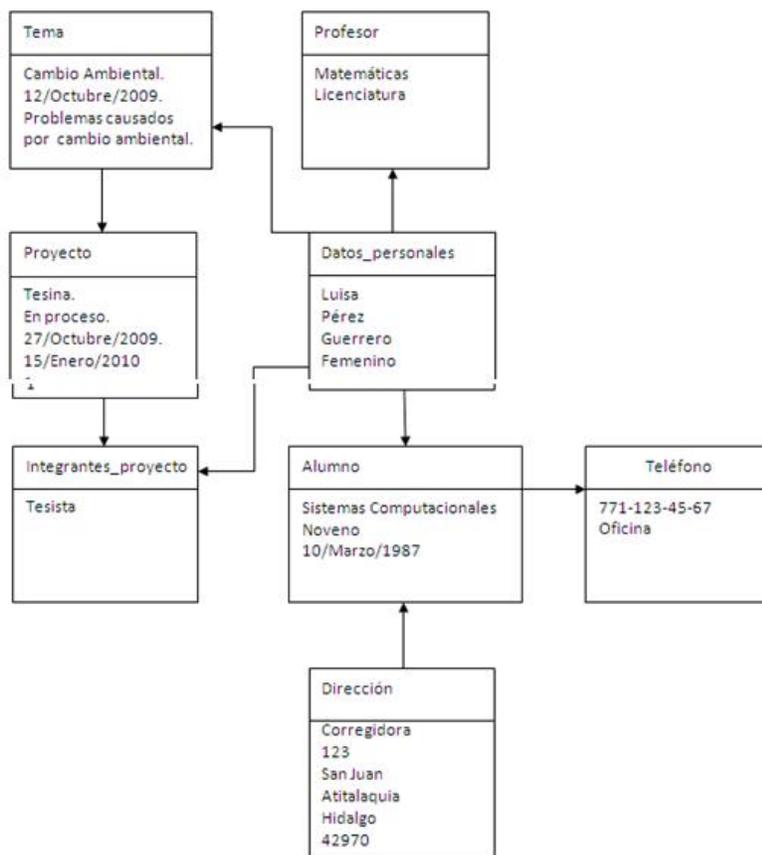


Figura 5.8: Diagrama de Objetos.

5.4. Diagrama de Clases

La figura 5.9 muestra el diagrama de clases que visualiza las relaciones entre las clases que involucran el sistema, las cuales pueden ser asociativas, de

herencia, de uso y de contenido.

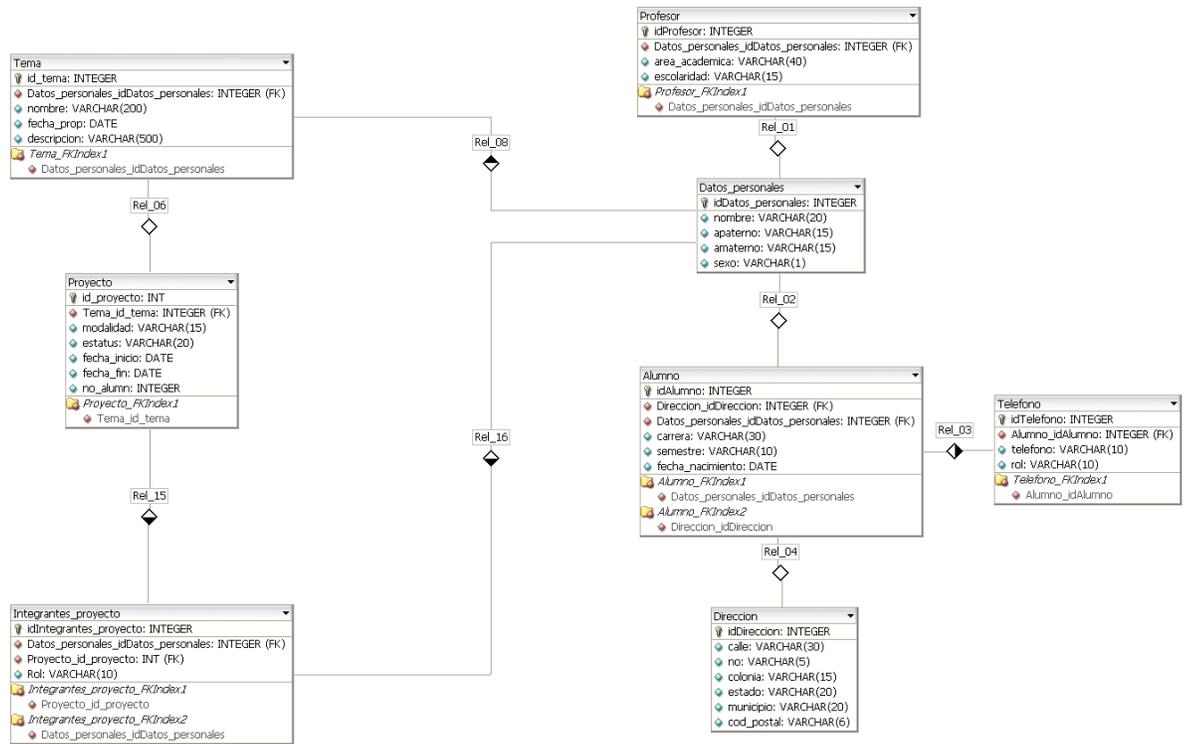


Figura 5.9: Diagrama de Clases

Capítulo 6

Desarrollo

En este capítulo se abordará el proceso de como se desarrolló el repositorio de proyectos, mencionando las herramientas y lenguajes que fueron necesarios utilizar para su construcción.

6.1. Diseño de la Interfaz de usuario

El diseño de la interfaz de usuario debe centrarse en forma directa a la comunicación efectiva entre el usuario y el sistema. Es necesario que se elabore un buen diseño en base al análisis de la información donde se definieron los objetos y acciones de la interfaz.

La interfaz del sistema Repositorio de Proyectos de Investigación ha sido diseñada por el conjunto de elementos construidos con distintas herramientas de diseño Web.

1.-Para elaborar el encabezado del sistema, se utilizó Flash 8, herramienta que efectuó animación a elementos de texto e imágenes. Figura 6.1

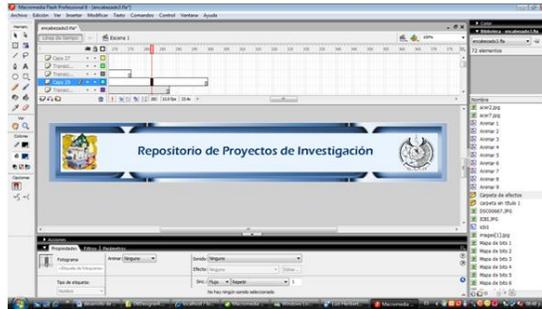


Figura 6.1: Encabezado en Flash

Para la elaboración del diseño gráfico, se utilizó Fireworks 8, herramienta que permite el retoque fotográfico ó la creación de nuevos elementos. Figura 6.2



Figura 6.2: Botones en Fiereworks

Cada elemento gráfico del sistema fue elaborado en fireworks con la finalidad de que las imágenes sean de mayor calidad.

La figura 6.3 muestra la creación del fondo de la pantalla principal del sistema repositorio.

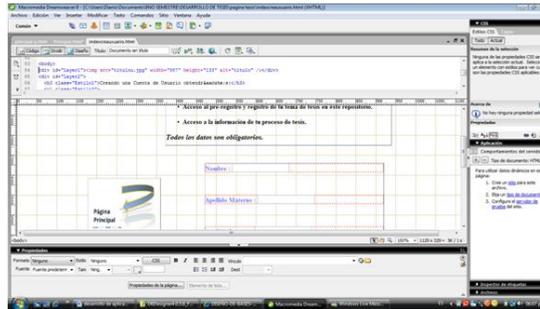


Figura 6.5: Formularios

6.2. Modelado de la base de datos

Es muy importante un buen diseño de base de datos ya que es la que establecerá comunicación directa con el sistema. Para la construcción y modelado implementamos DB Designer versión 4.0.5.6 con el cual se crearon las tablas que representan datos y relaciones. Figura 6.6

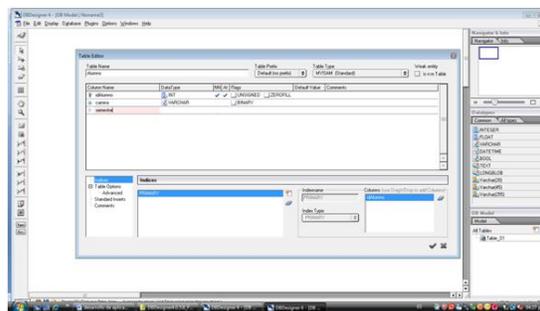


Figura 6.6: Creación de tablas en DBDesigner

Cada tabla tiene una clave primaria, un identificador único. Para establecer una relación entre dos tablas fue necesario incluir en una de ellas la clave primaria de la otra es decir una clave secundaria. Figura 6.7

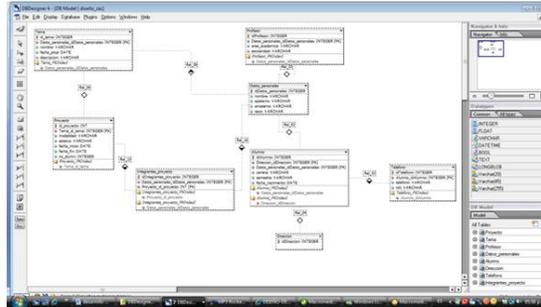


Figura 6.7: Modelo de la Base de Datos en DBDesigner

Una vez que se concluyó con el modelado de la base de datos se prosiguió a exportar el diseño de la base a sentencias SQL. DB Designer genera de manera automática los scripts.

Para el uso y organización de los datos se ha utilizado la herramienta XAMPP que es un paquete formado por un servidor web Apache, una base de datos MySQL y los intérpretes para los lenguajes PHP y Per. La instalación de este paquete es sencilla y en este caso se ha utilizado la versión 2.5.

Una de las ventajas XAMPP es que podemos hacer uso de las herramientas fácilmente, en este caso haremos usos de MySQL para crear la base de datos física la cual una vez construida se conectará a la aplicación Web.

Como primer paso es necesario acceder a la administración de MySQL, crear una nueva base de datos e importar los scripts generados por DBDesigner como se muestra en la figura 6.8.

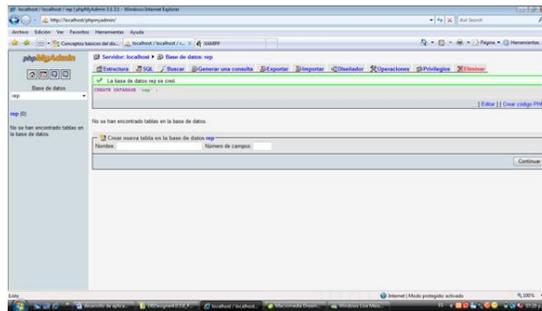


Figura 6.8: Crear Base de Datos en MySQL mediante importación de Scripts

1.- Creación de la base de datos con el nombre Repositorio. Figura 6.9.

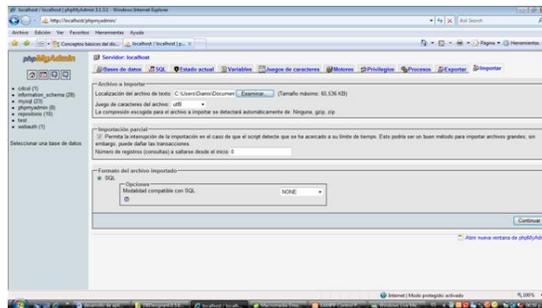


Figura 6.9: Base de datos ya creada

2.- Importación de los scripts generados por DB Designer. Para importar el archivo es necesario elegir la opción importar que se encuentra en la parte posterior derecha y de ahí proseguimos a localizar el archivo en examinar y una vez encontrado damos click en continuar.

La figura 6.10 muestra un mensaje que confirma que la importación de la base de datos se ha ejecutado correctamente. Ahora ya tenemos la base de datos del Repositorio integrada por diez tablas: tema, proyecto, integrantes del proyecto, cuenta, mensajes profesor, datos personales, dirección, alumno y teléfono.

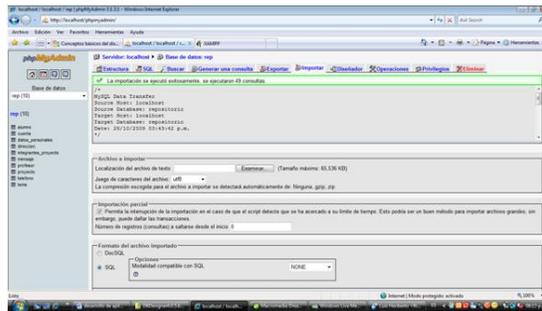


Figura 6.10: Base de Datos Repositorio

La figura 6.11 muestra la tabla datos personales con sus respectivos campos.

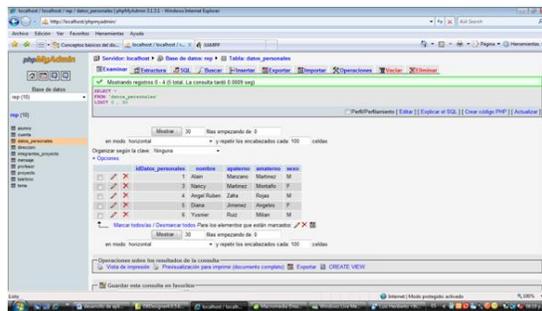


Figura 6.11: Campos de la tabla de datos personales

La programación del sistema se desarrolló sobre la tecnología de Java que implementa aplicaciones web.

Para continuar con el desarrollo de la aplicación es necesario tener instalado correctamente NetBeans 6.1.7

6.3. Desarrollo de la aplicación Web

Un JSP compila a un programa en Java la primera vez que se invoca y del programa en Java se crea una clase que se empieza a ejecutar en el servidor como un servlet. La principal diferencia entre los servlets y los JSPs es el enfoque de la programación: un JSP es una página Web con etiquetas especiales y código Java incrustado, mientras que un servlet es un programa que recibe peticiones y genera a partir de ella una página web.

Para la creación de aplicaciones se utilizó NetBeans versión 6.7.1 empleando la opción de proyecto de aplicaciones Web.

Para comenzar con el proyecto Repositorio se inicia NetBeans y una vez cargado el programa seleccionamos en la barra de menú Archivo - Nuevo proyecto, se despliega una nueva ventana y en la parte de categorías se selecciona Java y en la columna de projects seleccionamos Aplicaciones Java y por último damos clic en siguiente cuando aparezcan las ventanas posteriores y finalizamos el proyecto. Véase la figura. 6.12

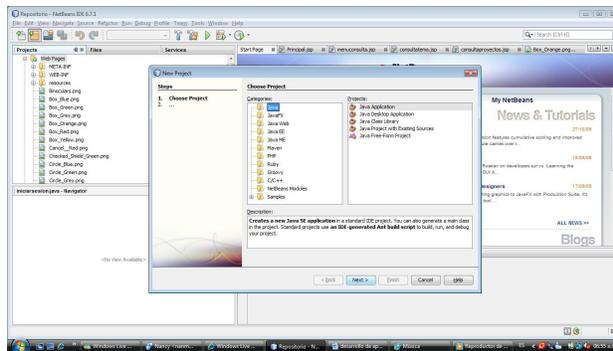


Figura 6.12: Creación de una aplicación NetBeans

La figura. 6.13 muestra la creación de un nuevo proyecto en NetBeans, en este caso nuestro proyecto es llamado Repositorio.

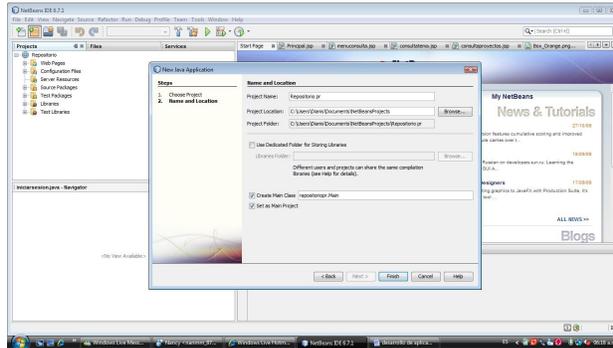


Figura 6.13: Creación de un nuevo proyecto en NetBeans

Una vez que se crea el proyecto se procede a la elaboración de las pantallas comenzando con lo siguiente. Como primer paso creamos un nuevo JSP, en la parte izquierda de la ventana seleccionamos con clic derecho sobre Web Pages - nuevo - JSP - asignamos un nombre de la página, en este caso se crea con el nombre Principal.jsp. Figura.6.14.

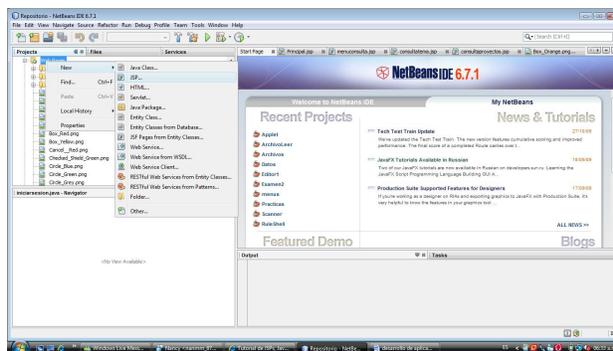


Figura 6.14: Creación de un JSP

Seleccionamos el contenido de la ventana principal y eliminamos el código que se encuentra predeterminado posteriormente se procede a agregar el código del diseño que se creó en DreamWeaver de la misma. Figura.6.15.

Para el correcto funcionamiento de la página tenemos que migrar todas las imágenes, animaciones, entre otros que se hayan utilizado al momento de crear el diseño en DreamWeaver, para esto tenemos que seleccionarlas de la carpeta origen, seleccionar, copiar y pegar en Web Pages

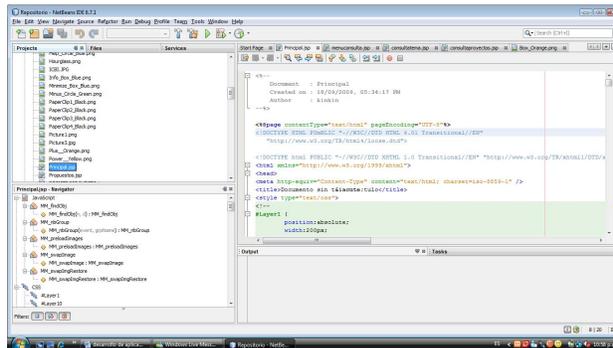


Figura 6.15: Ventana que muestra un JSP

De ésta forma se crean todas las páginas que va a contener el sistema y una vez finalizado se prosigue a la conexión entre la aplicación y la base de datos.

Para este procedimiento se necesita un servidor que va a permitir que se cargue la aplicación. Para ello agregamos el servidor Tomcat 6.0.

1.- En la parte izquierda de la ventana de NetBeans se selecciona la pestaña de servicios, buscamos la opción de servidores - clic derecho - agregar un nuevo servidor - Tomcat 5.0 - finalizar. Figura.6.16


```

package conexiones;

import entities.Alumno;
import entities.Cuenta;
import entities.DatosPersonales;
import entities.Direccion;
import entities.Telefono;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.Date;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.swing.JOptionPane;
import jpa.AlumnoJpaController;
import jpa.CuentaJpaController;
import jpa.DatosPersonalesJpaController;
import jpa.DireccionJpaController;
import jpa.TelefonoJpaController;

public class conexionAlumno extends HttpServlet {

    protected void processRequest
    (HttpServletRequest request, HttpServletResponse response)

        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();

        try {
            DatosPersonales datPer = new DatosPersonales();
            Alumno alumno = new Alumno();
            Cuenta cuenta = new Cuenta();
            Direccion direccion = new Direccion();
            Telefono tel = new Telefono();
            boolean b = true;
            String msj = "";

            DatosPersonalesJpaController jpaDatosPersonales =
                new DatosPersonalesJpaController();

            CuentaJpaController jpaCuenta = new CuentaJpaController();
            AlumnoJpaController jpaAlumno = new AlumnoJpaController();
            DireccionJpaController jpaDireccion = new DireccionJpaController();
            TelefonoJpaController jpaTelefono = new TelefonoJpaController();

```

```

datPer.setNombre(request.getParameter("nombre"));
datPer.setApaterno(request.getParameter("paterno"));
datPer.setAmaterno(request.getParameter("materno"));
datPer.setSexo(request.getParameter("sexo"));

if (datPer.getNombre().isEmpty()) {
    msj = "Nombre\n";
    b = false;
}

if (datPer.getApaterno().isEmpty()) {
    msj += "Apellido paterno\n";
    b = false;
}

if (datPer.getAmaterno().isEmpty()) {
    msj += "Apellido materno\n";
    b = false;
}

Date d = !request.getParameter("fnac").isEmpty()?
new Date(request.getParameter("fnac")) : new Date();

alumno.setDatospersonalesidDatospersonales(datPer);
alumno.setCarrera(request.getParameter("carrera"));
alumno.setFechaNacimiento(d);
alumno.setIdNoCuenta(request.getParameter("cuenta"));
alumno.setSemestre(request.getParameter("semestre"));

if (alumno.getCarrera().isEmpty()) {
    msj += "Carrera\n";
    b = false;
}

tel.setDatospersonalesidDatospersonales(datPer);
tel.setTelefono(request.getParameter("telcel"));
tel.setRol("Celular");

if (tel.getTelefono().isEmpty()) {
    msj += "Telefono\n";
    b = false;
}

if (!tel.getTelefono().matches("[0-9]+") || tel.getTelefono().length()
> 12 || tel.getTelefono().length() < 7) {

```

```

        JOptionPane.showMessageDialog(null, "Telefono invalido");
        b = false;
    }

    direccion.setAlumnoidnocuenta(alumno);
    direccion.setCalle(request.getParameter("calle"));
    direccion.setColonia(request.getParameter("colonia"));
    direccion.setNo(request.getParameter("numero"));
    direccion.setEstado(request.getParameter("estado"));
    direccion.setCodPostal(request.getParameter("cp"));
    direccion.setMunicipio(request.getParameter("municipio"));

    cuenta.setDatospersonalesidDatospersonales(datPer);
    cuenta.setUsuario(request.getParameter("usuario"));
    cuenta.setCorreoAlternativo(request.getParameter("correo"));
    cuenta.setPreguntaSecreta(request.getParameter("pregunta"));
    cuenta.setRespuestaSecreta(request.getParameter("respuesta"));
    cuenta.setContraseña(util.MD5.getMd5(request.getParameter
("pass"))));

    if (cuenta.getUsuario().isEmpty()) {
        msj += "Usuario\n";
        b = false;
    }

    if (cuenta.getCorreoAlternativo().isEmpty()) {
        msj += "Correo\n";
        b = false;
    }

    if (request.getParameter("pass").isEmpty() ||
        request.getParameter("pass2").isEmpty()) {

        msj += "Contraseña\n";
        b = false;
    }

    if (!request.getParameter("pass").equals(request.getParameter
("pass2"))) {
        JOptionPane.showMessageDialog(null, "La contraseña no coincide");
        b = false;
    }

    if (!b) {
        if (!msj.isEmpty()) {

```

```

        JOptionPane.showMessageDialog(null,
"Los siguientes datos son requeridos: \n" + msj);
    }
    response.sendRedirect("indexprealum.jsp?nombre=" +
datPer.getNombre() + "&paterno=" + datPer.getApaterno()
+ "&materno=" + datPer.getAmaterno() + "&fnac=" +
request.getParameter("fnac") + "&carrera="
+ alumno.getCarrera() + "&cuenta=" + alumno.getIdNoCuenta() +
"&semestre=" + alumno.getSemestre()
+ "&telcel=" + tel.getTelefono() + "&calle=" + direccion.getCalle()
"&numero=" + direccion.getNo()
+ "&colonia=" + direccion.getColonia() + "&municipio=" +
direccion.getMunicipio() + "&estado=" + direccion.getEstado()
+ "&usuario=" + cuenta.getUsuario() + "&correo=" +
cuenta.getCorreoAlternativo());
    return;
}

for(Cuenta c: jpaCuenta.findCuentaEntities())
    if(c.getUsuario().equals(cuenta.getUsuario()))
    {
        JOptionPane.showMessageDialog(null, "Nombre de usuario existente");
        response.sendRedirect("indexprealum.jsp?nombre=" +
datPer.getNombre() + "&paterno=" + datPer.getApaterno()
+ "&materno=" + datPer.getAmaterno() + "&fnac=" +
request.getParameter("fnac") + "&carrera="
+ alumno.getCarrera() + "&cuenta=" +
alumno.getIdNoCuenta() + "&semestre=" + alumno.getSemestre()
+ "&telcel=" + tel.getTelefono() + "&calle=" +
direccion.getCalle() + "&numero=" + direccion.getNo()
+ "&colonia=" + direccion.getColonia() + "&municipio=" +
direccion.getMunicipio() + "&estado=" + direccion.getEstado()
+ "&usuario=" + cuenta.getUsuario() + "&correo=" +
cuenta.getCorreoAlternativo());
        b = false;
        break;
    }

/*

VALIDAR NÚMERO DE CUENTA

*/

```

```

        if(b)
        {

                jpaDatosPersonales.create(datPer);
                jpaAlumno.create(alumno);
                jpaDireccion.create(direccion);
                jpaCuenta.create(cuenta);
                jpaTelefono.create(tel);

                response.sendRedirect("datosBien.jsp");
        }
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        out.close();
    }
}

// <editor-fold defaultstate="collapsed" desc="HttpServlet methods.
Click on the + sign on the left to edit the code.">
/**
 * Handles the HTTP <code>GET</code> method.
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doGet(HttpServletRequest request,
HttpServletResponse response)
        throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Handles the HTTP <code>POST</code> method.
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doPost(HttpServletRequest request,
HttpServletResponse response)
        throws ServletException, IOException {
    processRequest(request, response);
}

```

```

}

/**
 * Returns a short description of the servlet.
 * @return a String containing servlet description
 */
@Override
public String getServletInfo() {
    return "Short description";
} // </editor-fold>
}

```

La figura.6.18 muestra la sección del servlet que es el que se cargará directamente en el servidor.

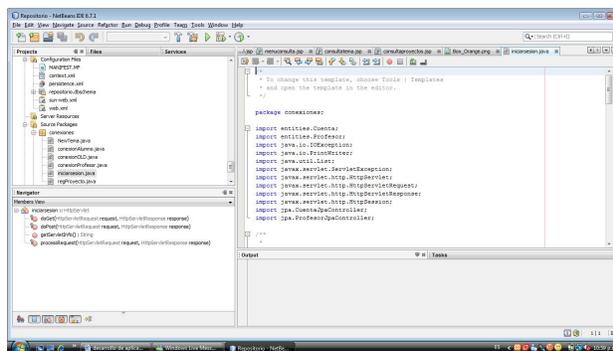


Figura 6.18: Conexión del servlet

Capítulo 7

Pruebas

A continuación se presentan algunas pruebas de la utilización del sistema Repositorio de proyectos de investigación.

Primero se comenzará explicando la interfaz gráfica de la página principal, que es la que se muestra en la figura.7.1.

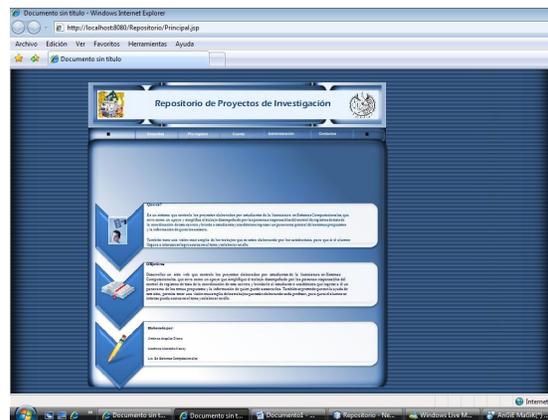


Figura 7.1: Ventana principal

La figura 7.1 contiene un encabezado que muestra el logo institucional así como el nombre del sistema, posteriormente cuenta con un menú para que el usuario pueda elegir una de las cinco opciones y como parte final se expone el tipo de sistema, su utilidad, objetivos y por quién fue elaborado.

7.1. Menú principal

Los elementos que conforman el menú principal son los que se muestran en la figura 7.2.

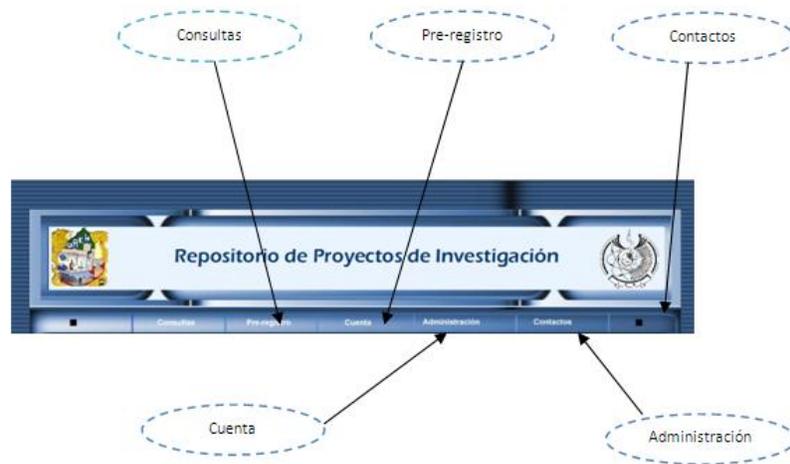


Figura 7.2: Menú

7.1.1. Consulta

Esta opción nos permite buscar información acerca de temas propuestos o temas ya registrados como proyecto para titulación. El usuario podrá elegir cualquiera de las dos opciones. Figura. 7.3

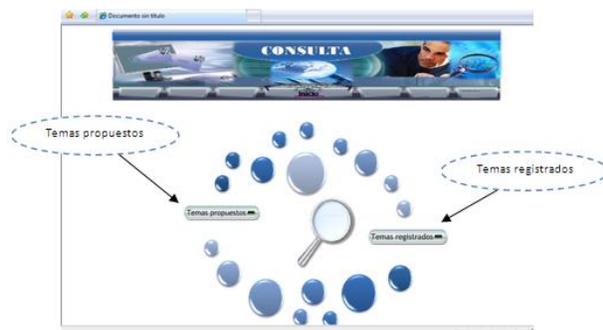


Figura 7.3: Ventana que muestra los dos tipos de consulta

Consulta de temas registrados: Muestra información de los temas que ya han sido dados de alta.

Consulta de temas propuestos: Muestra la información de temas propuestos por algunos profesores.

Permite al usuario realizar consultas de temas de proyectos de investigación que estén ya registrados o bien los temas que proponen los catedráticos o alumnos de la institución.

1.-Temas Propuestos

Este apartado nos permite consultar los temas, autores, descripción y la fecha de publicación de cada uno, donde los autores pueden ser alumnos o profesores.

Figura 7.4



TEMA	AUTOR	DESCRIPCION	INICIO
Seguridad en redes	Alain Manzano Martinez	Lo mismo	Tue Oct 17 00:00:00 CST 2009
Implementación de una central PBX con VPNs	Angel Ruben Zafra Rojas	Implementación de una central telefonica PBX (VoIP) bajo linux con el servidor Asterisk y redes privadas virtuales. Sistema para dar de alta y administrar proyectos como tesis, seminarios, monografias o articulos.	Tue Oct 28 00:00:00 CST 2009
Repositorio de tesis	Diana Jimenez Angeles	Sistema para dar de alta y administrar proyectos como tesis, seminarios, monografias o articulos.	Tue Oct 28 00:00:00 CST 2009
Visita guiada por el museo del papagallo virtualmente	Nancy Martinez Montaña	Museo virtualmente todas las instalaciones y los departamentos con los que cuenta el museo para que los usuarios puedan realizar su visita virtualmente.	Tue Oct 28 00:00:00 CST 2009

Figura 7.4: Interfaz de los temas propuestos

2.-Temas Registrados

Muestra información de los temas que ya han sido registrados y que por ende no podrán ser desarrollados por otros alumnos, además de mostrar información sobre quién lo registro, el asesor, la modalidad y el estado en que se encuentra el proceso. Figura 7.5



TEMA	ASESOR	INTEGRANTE(S)	ESTADO	MODALIDAD
Seguridad en redes	Nancy Martinez Montaña	nula martinez montaña	pre registrado	tesis

Figura 7.5: Interfaz de los temas ya registrados para desarrollarse

7.1.2. Pre-Registro

Eligiendo esta opción el usuario debe dar clic en el botón según su rol (alumno, profesor) para registro de información. Figura. 7.6



Figura 7.6: Ventana principal de Pre-Registro

1.-Pre-Registro-Profesor

El profesor ingresará sus datos personales, nombre de usuario y contraseña con lo cual podrá ingresar posteriormente al sistema en la opción de cuenta. Figura. 7.7.



Formato de solicitud de pre-registro Profesor

El primer paso es llenar el formulario de pre-registro, con el cual se podrá obtener un panorama global de ustedes usuarios.

Crear tu cuenta te permitirá:

- Acceso al pre-registro y registro de tu tema de tesis en este repositorio.
- Acceso a la información de tu proceso de tesis.

Es muy importante revisar todas sus respuestas antes de oprimir el botón de Enviar.

Nombre:

Apellido Paterno:

Apellido Materno:

Sexo:

Area académica:

Número de Matrícula:

Escolaridad:

Número telefónico:

Número telefónico (oficina):



Profesor



Página Principal



Alumno

Figura 7.7: Ventana de la interfaz de Pre-Registro para un profesor

2.-Pre-Registro-Alumno

El alumno ingresará sus datos personales y dará de alta una cuenta con la cual podrá ingresar posteriormente al sistema. Figura.7.8



Figura 7.9: Cuenta - Ventana inicial

El usuario podrá ingresar al sistema siempre y cuando se haya registrado anteriormente tecleando su nombre de usuario y contraseña. Figura. 7.10



Figura 7.10: Inicio de sesión

1.- Usuario-Profesor

Una vez que el profesor haya ingresado al sistema tendrá las opciones de registrar nuevos temas de proyectos de investigación, modificar sus datos y leer los mensajes enviados por administrador del sitio. Figura. 7.11



Figura 7.11: Menú de las opciones con las que cuenta el profesor

1.1.- Usuario-Profesor-Registro

Si un profesor desea registrar un tema para proyecto de investigación, deberá ingresar los datos correspondientes como son el nombre y una breve descripción de lo que trata su tema, esta información será guardada en la base de datos que inmediatamente la publicará en el apartado de consultas en la opción de temas publicados para que quien este interesado pueda desarrollarlo. Figura. 7.12

Figura 7.12: Ventana del registro de tema del profesor

1.2.-Usuario-Profesor-Modificar datos

En caso de que el profesor haya ingresado algún dato erróneo tiene la posibilidad de realizar las respectivas modificaciones en sus datos pero solo con respecto a su área académica, escolaridad y teléfonos, por eso es importante que cuando un usuario llene su pre-registro lo haga con los datos correctos. Figura.7.13



The screenshot shows a web interface for modifying a professor's data. At the top is a blue banner with the text 'MODIFICAR DATOS' in a stylized font. Below the banner are three navigation icons: 'MENSAJES' (Messages), 'REGISTRO' (Registration), and a profile icon. The main content area is titled 'Modifica tus datos que hayan cambiado:' and includes a 'Confirmar información' link. The form contains four input fields: 'Escolaridad:' with the value 'Licenciatura', 'Área Académica:' with the value 'LIC', 'Teléfono:' with the value '7711212464', and 'Teléfono de Oficina:'. At the bottom, there is a warning message: 'Al guardar, modificar o eliminar datos debe de ser informado el usuario que se ve afectado directamente.' and two buttons: 'GUARDAR' and 'CANCELAR'.

Figura 7.13: Modificación de datos del profesor

1.3-Usuario-Profesor-Mensajes

El administrador del sitio web estará enviando mensajes en este apartado a la cuenta del profesor acerca del estado de sus temas propuestos o de cualquier otro tipo de información. Figura.7.14



Figura 7.14: Mensajes dirigidos al profesor

2.- Usuario- Alumno

En la cuenta de alumno están disponibles cuatro de las opciones posibles que se brindan al usuario, registro de un tema, registro del proyecto, modificar su información y mensajes. Figura. 7.15.



Figura 7.15: Menú de opciones para el alumno

2.1-Usuario-Alumno-Registro

El registro es una de las opciones que permite al alumno publicar un tema de proyecto de investigación que desee compartir con alguien para su desarrollo o simplemente proporcionar una opción a otros compañeros para que ellos desarrollen. El alumno solo debe de escribir el tema y realizar una pequeña introducción de éste, al momento de enviar, será inmediatamente publicado en la opción de consultas en temas registrados. Figura. 7.16.

The screenshot shows a web interface for registering a research topic. At the top, there is a banner with the word 'REGISTRO' in large, stylized letters. Below the banner, there is a section titled 'Registro del tema para el desarrollo del proyecto de investigación'. This section contains several lines of text: 'Los siguientes datos se consideran altamente importantes, se pide que las respuestas escritas sean verdicas y correctas.', 'Una vez enviada la informacion sera verificada.', 'Revise todas sus respuestas antes de oprimir el boton de Enviar.', and 'Todos los datos son obligatorios.'. There are three icons on the left side: 'MONITOREAR DATOS', 'MENSAJES', and 'MENSAJES'. The form itself has a 'Tema:' label followed by a text input field containing 'Administración de las TICs'. Below that is a 'Describe tu tema:' label followed by a larger text area containing 'REGISTRAR EN COMO ADMINISTRACION DE LAS NUEVAS TECNOLOGIAS'. At the bottom of the form, there is a small disclaimer: 'Al hacer clic en Enviar manifiesta su conformidad con el servicio que ofrece el repositorio de proyectos de investigacion y confirma la veracidad de su informacion capturada.' and two buttons: 'Enviar' and 'Cancelar'.

Figura 7.16: Ventana del registro de tema del alumno

2.2-Usuario-Alumno-Registro de Proyecto

Una vez que el alumno haya elegido su tema a desarrollar, podrá darlo de alta como proyecto de investigación en el sistema. Para ello deberá ingresar los datos de los integrantes y del asesor, además de la modalidad en la cuál trabajará. Figura. 7.17.



Figura 7.17: Ventana del registro del proyecto a desarrollar

2.3- Usuario- Alumno- Modificar datos

En caso de que en los datos del alumno existan errores o sea necesario actualizar la información, se podrán realizar los cambios necesarios en esta opción, pero solo en datos como su dirección, teléfono y semestre. Por eso es indispensable que toda la información que ingrese sea la correcta. Figura. 7.18.



Figura 7.18: Ventana que muestra la opción de modificar datos del alumno

2.4-Usuario-Alumno-Mensajes

Para mantener informado al alumno del estado de su proyecto el sistema enviará los mensajes correspondientes a esta opción. Figura. 7.19.



Figura 7.19: Ventana de mensajes dirigidos al alumno

7.1.4. Administrador

Esta opción es únicamente para el administrador del sistema el cual cuenta con la opción de ver toda la información de los usuarios (profesores, alumnos) y de enviar mensajes a ellos. Debe de ingresar con su usuario y contraseña que el sistema le proporciona. Figura 7.20.



Figura 7.20: Ventana principal del administrador. Inicio de sesión

1.- Administrador-Opción

Una vez que el administrador haya ingresado al sistema podrá consultar la información de los usuarios que estén registrados en el sitio. Figura. 7.21.



Figura 7.21: Tres opciones a elegir (alumno/profesor/mensajes)

1.2.- Administrador-Alumno

En esta ventana se muestra la información del registro del alumno, sus datos personales e información del proyecto de investigación. Figura. 7.22.

The image shows a form titled 'Formato de solicitud de registro Alumno.' It has a search bar for 'Número de Cuenta:'. Below the search bar are two tabs: 'Profesor' and 'Alumno'. The 'Alumno' tab is selected. The form contains two tables. The first table is for 'DATOS PERSONALES' and the second is for 'DATOS DE TEMA'.

DATOS PERSONALES		INFORMACIÓN
Nombre		
Correo alternativo		

DATOS DE TEMA		INFORMACIÓN
Tema		
Asesor		
Fecha de inicio		
Fecha fin		

Figura 7.22: Consulta de datos sobre alumnos

1.3.- Administrador-Profesor

En este apartado muestra la información correspondiente a los datos del profesor y del tema o temas que han sido registrados como propuesta para su desarrollo. Figura 7.23.

Formato de solicitud de registro profesor

Matricula

DATOS PERSONALES	INFORMACION
Nombre	Apellido
Apellido Paterno	Apellido Materno
Sexo	Telefono
Telefono	Telefono oficina
Cuenta	Correo Alternativo

DATOS DE TEMA	INFORMACION
Tema	*Nota guárdela por el caso de paginas respaldadas

Figura 7.23: Consulta de datos sobre profesores

1.4.- Administrador-Mensaje

En esta sección el administrador puede enviar mensajes a los alumnos y/o profesores registrados en este repositorio por medio de su número de cuenta o matrícula. Figura. 7.24.

Destinatario (Cuenta/Matricula):

Escribir el mensaje:

Siempre lea a esta repositorio.

El tema fue aceptado.

Figura 7.24: Consulta de datos sobre profesores

7.1.5. Contactos

Opción que muestra información de las personas encargadas del funcionamiento del sistema, de su operación y elaboración. Figura. 7.25.



Figura 7.25: Contactos

Conclusiones

La tecnología está cambiando constantemente y nos proporciona muchas facilidades para simplificar el trabajo y mejorar los resultados. Los Repositorios Institucionales son una nueva e incipiente clase de servicios y software.

Por lo que al elaborar este sistema se buscó facilitar el proceso de registro para proyectos de investigación que se desarrollan en la Universidad Autónoma del Estado de Hidalgo para el Instituto de Ciencias básicas e Ingeniería

Al concluirlo pudimos apreciar que el objetivo se logró en su totalidad. Debido a que el sitio web fue elaborado pensando en automatizar el proceso de registro de tesis y se cumplió con la meta abarcando un poco más de ello, claro ésta que este sistema puede ser mejorado en un futuro.

El servicio que ofrece el Repositorio Institucional abarca toda la información que se maneja en el proceso de registro de temas y/o proyectos de investigación que los alumnos registran como tesis.

Durante el proceso para la realización de este proyecto se emplearon diversas herramientas que aprendimos en la carrera tales como programación, diseño web, análisis, metodología de investigación, metodología de desarrollo de software, entre otros, lo cual nos sirvió para practicar y reforzar los conocimientos y lograr que el producto resultante cumpla con las expectativas deseadas.

Se cuenta también con una visión para trabajo futuro que va a consistir en implementar el sistema en la coordinación de la Licenciatura de Sistemas Computacionales, seguido de realizar pruebas de su desempeño y si lo requiere, hacer modificaciones en su diseño lógico y físico del repositorio, así como las actualizaciones que sean necesarias.

En conclusión este proyecto es de gran utilidad tanto para nosotros como para la institución, ya que se logró un mutuo beneficio. Y nos produjo satisfacción emplear el sistema que facilita el proceso de selección y registro de temas de tesis mejorando la eficiencia y eficacia del servicio. Además de ayudar a mejorar la comunicación académica, conservar materiales de aprendizaje, aumentar del prestigio de la Universidad exponiendo sus investigaciones académicas.

Glosario

- **Ajax.** Acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML). Técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications).
- **Atributos.** Valores que corresponden a un objeto, como color, material, cantidad, ubicación. Generalmente se conoce como la información detallada del objeto. Suponiendo que el objeto es una puerta, sus propiedades serían: la marca, tamaño, color y peso.
- **Base de datos.** Conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.
- **Base.** Es en donde se encuentra toda la información de todos los registros sin que se haga ninguna validación adicional.
- **Campo.** Corresponde al nombre de la columna. Debe ser único y además de tener un tipo de dato asociado.
- **Caso de uso.** Técnica para la captura de requisitos potenciales de un nuevo sistema o una actualización de software.
- **Ciclo de vida clásico del desarrollo de sistemas.** Es el conjunto de actividades que los analistas, diseñadores y usuarios realizan para desarrollar e implantar un sistema de información.
- **Ciclo de vida de un sistema.** Es un enfoque por fases del análisis y diseño que sostiene que los sistemas son desarrollados de mejor manera mediante el uso de un ciclo específico de actividades del analista y del usuario.
- **Diagrama de actividades.** Es un tipo especial de diagrama de estados que muestra el flujo secuencial o ramificado de actividades en un sistema. Conjunto de actividades, Son especialmente importantes para modelar la función del sistema, así como para resaltar el flujo de control entre objetos.
- **Diagrama de clases.** Presenta un conjunto de clases, interfaces y colaboraciones, y las relaciones entre ellas.

- **Diagrama de colaboración.** Es un diagrama de interacción que resalta la organización estructural de los objetos que envían y reciben mensajes. Un diagrama de colaboración muestra un conjunto de objetos, enlaces entre estos objetos y mensajes enviados y recibidos por estos objetos. Los objetos normalmente son instancias con nombre o anónimas de clases, pero también pueden representar instancias de otros elementos, como colaboraciones, componentes y nodos.
- **Diagrama de componentes.** Muestra un conjunto de componentes y sus relaciones, se utilizan para describir la vista de implementación estática de un sistema.
- **Diagrama de despliegue.** Muestra un conjunto de nodos y sus relaciones, se utilizan para describir la vista de despliegue estática de un sistema.
- **Diagrama de estado.** Representa una máquina de estados, constituida por estados, transiciones, eventos y actividades. Son especialmente importantes para modelar el comportamiento de una interfaz, una clase o una colaboración. Los diagramas de estados resaltan el comportamiento dirigido por eventos de un objeto, lo que es especialmente útil al modelar sistemas reactivos.
- **Diagrama de objetos.** Representa un conjunto de objetos y sus relaciones. Se utilizan para describir estructuras de datos, instantáneas de los elementos encontrados en los diagramas de clases. Cubre los mismos aspectos que los diagramas de clases pero desde una perspectiva de casos reales o prototípicos.
- **Diagrama de secuencia.** Es un diagrama de interacción que resalta la ordenación temporal de los mensajes.
- **Diagrama de secuencia.** Presenta un conjunto de objetos y los mensajes enviados y recibidos por ellos., Los objetos suelen ser instancias con nombre o anónimas de clases, pero también pueden representarse instancias de otros elementos, tales como colaboraciones, componentes y nodos.
- **Diagramas de casos de uso.** Organizan los comportamientos del sistema. Un diagrama de caso de uso representa un conjunto de casos de uso y actores (un tipo especial de clases) y sus relaciones.
- **Diagramas de interacción.** Se da este nombre colectivo a los diagramas de secuencia y los diagramas de colaboración. Ambos diagramas son isomorfos, es decir, se puede convertir de uno a otro sin pérdida de información.
- **Herencia.** Reutilización de un objeto padre ya definido para poder extender la funcionalidad en un objeto hijo. Los objetos hijos heredan todas las operaciones y/o propiedades de un objeto padre.

- **HTML.** Siglas de HyperText Markup Language (Lenguaje de Marcas de Hipertexto). Lenguaje de marcado predominante para la construcción de páginas web.
- **Iniciar sesión.** Acción de comenzar una sesión de un usuario específico, permitiendo así identificarse frente a un sistema o servicio.
- **Interfaz.** Conjunto de operaciones que permiten a un objeto comportarse de cierta manera, por lo que define los requerimientos mínimos del objeto.
- **Investigación Preliminar.** La solicitud para recibir ayuda de un sistema de información puede originarse por varias razones: sin importar cuales sean estas, el proceso se inicia siempre con la petición de una persona.
- **Java.** Es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.
- **JSP.** Es un método de creación de páginas web dinámicas en servidor usando el lenguaje Java.
- **Lenguaje de programación.** Es un idioma artificial diseñado para expresar computaciones que pueden ser llevadas a cabo por máquinas como las computadoras.
- **Metodología.** Hace referencia al conjunto de métodos de investigación utilizados para alcanzar una gama de objetivos en una ciencia.
- **MySQL.** Es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones.
- **Operaciones.** Llamados métodos, son aquellas actividades o verbos que se pueden realizar con/para este objeto, como por ejemplo abrir, cerrar, buscar, cancelar, acreditar, cargar.
- **Orientado a objetos.** Tipo de lenguaje de programación basado en la idea de encapsular estado y operaciones en objetos.
- **Página de inicio.** Es el primer documento que ve el usuario cuando entra en el sitio web poniendo el nombre del dominio de ese sitio web en un navegador.
- **Registro.** Corresponde a cada fila que compone la tabla. Allí se componen los datos y los registros. Eventualmente pueden ser nulos en su almacenamiento.
- **Repositorio.** Es un sitio centralizado donde se almacena y mantiene información digital, habitualmente bases de datos o archivos informáticos.

- **Servidor de Base de Datos.** (database server) provee servicios de base de datos a otros programas u otras computadoras, como es definido por el modelo cliente-servidor.
- **Servidor web.** 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.
- **Servidor.** Es una computadora que, formando parte de una red, provee servicios a otras computadoras denominadas clientes.
- **Servlet.** Es un programa que se ejecuta en un servidor. Los servlets son objetos que corren dentro del contexto de un contenedor de servlets y extienden su funcionalidad. También podrían correr dentro de un servidor de aplicaciones.
- **Sesión.** Es el marco espacio-temporal en el que se desarrolla una actividad concreta, en la cual se participa como actor o bien como espectador.
- **Sitio web.** Es un conjunto de páginas web, típicamente comunes a un dominio de Internet o subdominio en la World Wide Web en Internet.
- **SQL.** (Lenguaje de consulta estructurado), es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en éstas.
- **Tabla.** En las bases de datos, se refiere al tipo de modelado de datos, donde se guardan los datos recogidos por un programa. Su estructura general se asemeja a la vista general de un programa de Hoja de cálculo.
- **UML. Lenguaje Unificado de Modelado,** por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group).
- **Usuario.** Individuo que utiliza una computadora, sistema operativo, servicio o cualquier sistema informático. Por lo general es una única persona.
- **Vistas.** Es una vista o relación que se hace en referencia a una fila o columna específica.
- **XAMPP.** Es un servidor independiente de plataforma, software libre, que consiste principalmente en la base de datos MySQL, el servidor Web Apache y los intérpretes para lenguajes de script: PHP y Perl. El nombre proviene del acrónimo de X (para cualquiera de los diferentes sistemas operativos).

Anexos

Para tener un panorama amplio de la estructura interna del presente proyecto se anexa el código generado de la construcción del sistema, específicamente de páginas Web y servlets que constituyen la conexión del repositorio de proyectos con la base de datos.

Anexo 1.-Código de la pagina Principal.jsp, elaborada con la herramienta Dreamweaver 8 y con el lenguaje de programación Java, la cual, se muestra al usuario al inicio del sistema.

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"

    "http://www.w3.org/TR/html4/loose.dtd">

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859
-1" />
<title>UAEH - Repositorio de Proyectos de Investigaci&oacuten - ICBI
</title>
<style type="text/css">
<!--
#Layer1 {
position:absolute;
width:200px;
height:56px;
z-index:1;
left: 261px;
top: 198px;
}
body,td,th {
```

```
color: #F0F0F0;
}
body {
background-color: #F0F0F0;
background-image: url(resources/images/menu_r1_c1.jpg);
}
#Layer2 {
position:absolute;
width:200px;
height:115px;
z-index:2;
left: 560px;
top: 277px;
}
#Layer3 {
position:absolute;
width:620px;
height:822px;
z-index:3;
left: 259px;
top: 233px;
}
#Layer4 {
position:absolute;
width:200px;
height:115px;
z-index:4;
left: 261px;
top: 22px;
}
#Layer5 {
position:absolute;
width:200px;
height:175px;
z-index:4;
top: 32px;
left: 256px;
}
#Layer6 {
position:absolute;
width:200px;
height:54px;
z-index:5;
left: 211px;
top: 191px;
}
```

```
#Layer7 {
position:absolute;
width:200px;
height:115px;
z-index:6;
left: 261px;
top: 44px;
}
#Layer8 {
position:absolute;
width:59px;
height:29px;
z-index:7;
left: 443px;
top: 220px;
}
#Layer9 {
position:absolute;
width:84px;
height:28px;
z-index:8;
left: 585px;
top: 223px;
}
#Layer10 {
position:absolute;
width:121px;
height:38px;
z-index:9;
left: 732px;
top: 221px;
}
#Layer11 {
position:absolute;
width:708px;
height:296px;
z-index:10;
left: 453px;
top: 309px;
}
#Layer12 {
position:absolute;
width:104px;
height:31px;
z-index:11;
left: 877px;
```

```
top: 221px;
}
#Layer13 {
position:absolute;
width:194px;
height:31px;
z-index:12;
left: 1033px;
top: 219px;
}
#Layer14 {
position:absolute;
width:200px;
height:115px;
z-index:13;
left: 853px;
top: 325px;
}
#Layer15 {
position:absolute;
width:474px;
height:416px;
z-index:14;
left: 451px;
top: 641px;
}
#Layer16 {
position:absolute;
width:770px;
height:495px;
z-index:13;
left: 1316px;
top: 382px;
}
#Layer17 {
position:absolute;
width:439px;
height:400px;
z-index:1;
left: 571px;
top: -61px;
}
#Layer18 {
position:absolute;
width:720px;
height:119px;
```

```
z-index:14;
left: 967px;
top: 393px;
}
#Layer19 {
position:absolute;
width:200px;
height:115px;
z-index:15;
left: 79px;
top: 1155px;
}
#Layer20 {
position:absolute;
width:1205px;
height:1037px;
z-index:15;
left: 47px;
top: 231px;
}
#Layer21 {
position:absolute;
width:760px;
height:115px;
z-index:16;
left: 457px;
top: 447px;
}
#Layer22 {
position:absolute;
width:269px;
height:241px;
z-index:1;
left: 197px;
top: 174px;
}
#Layer23 {
position:absolute;
width:763px;
height:115px;
z-index:17;
left: 455px;
top: 646px;
}
.Estilo9 {
color: #000033;
```

```
font-size: 13px;
}
#Layer24 {
position:absolute;
width:200px;
height:115px;
z-index:18;
left: 297px;
top: 653px;
}
#Layer25 {
position:absolute;
width:413px;
height:115px;
z-index:2;
left: 415px;
top: 630px;
}
.Estilo11 {
font-size: 12px;
font-family: Georgia, "Times New Roman", Times, serif;
color: #000033;
}
.Estilo12 {font-size: 12px; font-family: Georgia, "Times New Roman",
Times, serif; color: #000033; font-weight: bold; }
.Estilo14 {font-size: 10px}
#Layer26 {
position:absolute;
width:200px;
height:115px;
z-index:19;
left: 314px;
top: 887px;
}
.Estilo17 {font-family: Georgia, "Times New Roman", Times, serif}
.Estilo18 {
font-weight: bold;
color: #003366;
font-family: Georgia, "Times New Roman", Times, serif;
font-size: 13px;
}
.Estilo19 {
color: #003366;
font-family: Georgia, "Times New Roman", Times, serif;
font-size: 13px;
}
```

```

#Layer27 {
position:absolute;
left:353px;
top:273px;
width:77px;
height:38px;
z-index:20;
}
-->
</style>
<script type="text/JavaScript">
<!--
function MM_preloadImages() { //v3.0
    var d=document; if(d.images){ if(!d.MM_p) d.MM_p=new Array();
        var i,j=d.MM_p.length,a=MM_preloadImages.arguments; for(i=0;
            i<a.length; i++)
            if (a[i].indexOf("#")!=0){ d.MM_p[j]=new Image; d.MM_p[j++].
                src=a[i];}}
}

function MM_findObj(n, d) { //v4.01
    var p,i,x;  if(!d) d=document; if((p=n.indexOf("?"))>0&&parent
        .frames.length) {
        d=parent.frames[n.substring(p+1)].document; n=n.substring(0,p);}
    if(!(x=d[n])&&d.all) x=d.all[n]; for (i=0;!x&&i<d.forms.length
        ;i++) x=d.forms[i][n];
    for(i=0;!x&&d.layers&&i<d.layers.length;i++) x=MM_findObj(n,d.
        layers[i].document);
    if(!x && d.getElementById) x=d.getElementById(n); return x;
}

function MM_nbGroup(event, grpName) { //v6.0
    var i,img,nbArr,args=MM_nbGroup.arguments;
    if (event == "init" && args.length > 2) {
        if ((img = MM_findObj(args[2])) != null && !img.MM_init) {
            img.MM_init = true; img.MM_up = args[3]; img.MM_dn = img.
                src;
            if ((nbArr = document[grpName]) == null) nbArr = document
                [grpName] = new Array();
            nbArr[nbArr.length] = img;
            for (i=4; i < args.length-1; i+=2) if ((img = MM_findObj(
                args[i])) != null) {
                if (!img.MM_up) img.MM_up = img.src;
                img.src = img.MM_dn = args[i+1];
                nbArr[nbArr.length] = img;
            } }
    } }
}

```

```

} else if (event == "over") {    document.MM_nbOver = nbArr
= new Array();
  for (i=1; i < args.length-1; i+=3) if ((img = MM_findObj(
args[i])) != null) {          if (!img.MM_up) img.MM_up =
img.src;
  img.src = (img.MM_dn && args[i+2]) ? args[i+2] :
((args[i+1])? args[i+1] : img.MM_up);
  nbArr[nbArr.length] = img;
}
} else if (event == "out" ) {
for (i=0; i < document.MM_nbOver.length; i++) {
  img = document.MM_nbOver[i]; img.src = (img.MM_dn) ?
img.MM_dn : img.MM_up; }
} else if (event == "down") {
nbArr = document[grpName];
if (nbArr)
  for (i=0; i < nbArr.length; i++) { img=nbArr[i]; img.
src = img.MM_up; img.MM_dn = 0; }
document[grpName] = nbArr = new Array();
for (i=2; i < args.length-1; i+=2) if ((img = MM_findObj
args[i])) != null) {
  if (!img.MM_up) img.MM_up = img.src;
  img.src = img.MM_dn = (args[i+1])? args[i+1] : img.MM_up;
  nbArr[nbArr.length] = img;
} }
}
}

function MM_swapImgRestore() { //v3.0
var i,x,a=document.MM_sr; for(i=0;a&&i<a.length&&(x=a[i])&&
x.oSrc;i++) x.src=x.oSrc;
}

function MM_swapImage() { //v3.0
var i,j=0,x,a=MM_swapImage.arguments; document.MM_sr=new
Array; for(i=0;i<(a.length-2);i+=3)
  if ((x=MM_findObj(a[i]))!=null){document.MM_sr[j++]=x;
  if(!x.oSrc) x.oSrc=x.src; x.src=a[i+2];}
}
//-->
</script>
</head>

<body onload="MM_preloadImages('resources/images/menu_r2_c2_
f3.jpg','resources/images/menu_r2_c2_f2.jpg','resources/
images/menu_r2_c2_f4.jpg','resources/images/menu_r2_c4_f2.jpg'
,'resources/images/menu_r2_c6_f2.jpg','resources/images/menu_

```

```

r2_c8_f2.jpg',',',resources/images/menu_r2_c10_f2.jpg')">
<div id="Layer3"></div>
<div id="Layer7">
  <object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
  codebase="http://download.macromedia.com/pub/shockwave/cabs
  /flash/swflash.cab#version=7,0,19,0" width="1000" height="
  150">
    <param name="movie" value="resources/flash/oopc2.swf" />
    <param name="quality" value="high" />
    <embed src="resources/flash/oopc2.swf" quality="high" pluginspage=
    "http://www.macromedia.com/go/getflashplayer" type="application/
    x-shockwave-flash" width="1000" height="150"></embed>
  </object>
</div>
<div id="Layer24"></div>
<div class="Estilo14" id="Layer21">
  <p align="justify" class="Estilo18">Que es? </p>
  <p align="justify" class="Estilo19">Es un sistema que
  controla los proyectos elaborados por estudiantes de la
  licenciatura en Sistemas Computacionales, que sirve como
  un apoyo y simplifica el trabajo desempe&ntilde;ado por
  las personas responsables del control de registros de
  tesis de la coordinaci&oacute;n de esta carrera y brinda
  a estudiantes y acad&eacute;micos ingresar un panorama
  general de los temas propuestos y la informaci&oacute;n
  de quien los asesora.</p>
  <p align="justify" class="Estilo19"> Tambi&eacute;n
  tiene una visi&oacute;n mas amplia de los trabajos que
  se est&aacute;n elaborando por los catedraticos, para
  que si el alumno llegara a interesarse logre acercarse
  al tema y colaborar en ello.</p>
</div>
<div id="Layer1">
  <table width="1000" border="0" cellpadding="0"
  cellspacing="0">
    <!-- fwtable fwsrc="menu.png" fwbase="menu.jpg"
    fwstyle="Dreamweaver" fwdocid = "974161121" fwnested="0" -->
    <tr>
      <td colspan="11"></td>
    </tr>
    <tr>
      <td rowspan="2"></td>
<td><a href="menuconsulta.jsp" target="_self"
onmouseout="MM_nbGroup('out');" onmouseover="MM_nb
Group('over','menu_r2_c2','resources/images/menu_
r2_c2_f2.jpg','resources/images/menu_r2_c2_f4.jpg',1);"
onclick="MM_nbGroup('down','navbar1','menu_r2_c2','
resources/images/menu_r2_c2_f3.jpg',1);"></a></td><td rowspan="2">
</td><td><a href="indexprepre.jsp" onmouseout="MM_swap
ImgRestore()" onmouseover="MM_swapImage('menu_r2_c41','',
resources/images/menu_r2_c4_f2.jpg',1)"></a></td>
<td rowspan="2"></td>
<td><a href="indexusuarioalpro.jsp" onmouseout="MM_
swapImgRestore()" onmouseover="MM_swapImage('menu_r2_c61',
'',resources/images/menu_r2_c6_f2.jpg',1)"></a></td>
<td rowspan="2"></td>
<td><a href="indexconta.jsp" onmouseout="MM_swapImgRestore()"
onmouseover="MM_swapImage('menu_r2_c81','',resources/images/
menu_r2_c8_f2.jpg',1)"></a></td>
<td rowspan="2"></td>
<td><a href="indexconta.jsp" onmouseout="MM_swapImgRestore()"
onmouseover="MM_swapImage('menu_r2_c101','',resources/images/
menu_r2_c10_f2.jpg',1)"></a></td>
<td rowspan="2"></td>
</tr>

```

```

<tr>
  <td></td>
  <td></td>
  <td></td>
  <td></td>
  <td></td>
</tr>
</table>
</div>
<div id="Layer20">
  
  <div id="Layer25">
    <p class="Estilo12">Elaborado por:</p>
    <p class="Estilo11">Jim&eacute;nez Angeles Diana</p>
    <p class="Estilo11">Mart&iacute;nez Monta&ntilde;o Nancy </p>
    <p class="Estilo11"> Lic. En Sistemas Computacionales </p>
  </div>
  <div id="Layer22"></div>
</div>
<div id="Layer23">
  <div align="justify" class="Estilo9">
    <p class="Estilo17"><strong>Objetivos</strong></p>
    <p class="Estilo17">Desarrollar un sitio web que controle
los proyectos elaborados por estudiantes de la licenciatura
en Sistemas Computacionales, que sirva como un apoyo que
simplifique el trabajo desempe&ntilde;ado por las personas
responsables del control de registros de tesis de la
coordinaci&oacute;n de esta carrera y brindarle al estudiante
o acad&eacute;micos que ingrese a &eacute;l un panorama
de los temas propuestos y la informaci&oacute;n de quien
puede asesorarlos. Tambi&eacute;n se pretende que con
la ayuda de este sitio, permita tener una visi&oacute;n
mas amplia de los trabajos que est&aaacute;n elaborando
cada profesor, para que si el alumno se interesa pueda
acercarse al tema y colaborar en ello. </p>
  </div>
</div>
<div id="Layer26"></div>

```

```

<div id="Layer27">
  <object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
    codebase="http://download.macromedia.com/pub/shockwave/cabs
    /flash/swflash.cab#version=7,0,19,0" width="40" height="40">
    <param name="movie" value="resources/flash/sound.swf" />
    <param name="quality" value="high" />
    <embed src="resources/flash/sound.swf" quality="high"
    pluginspage="http://www.macromedia.com/go/getflashplayer" type="
    application/x-shockwave-flash" width="40" height="
    40"></embed>
  </object>
</div>
</body>
</html>

```

Anexo 2.-La clase se llama *admin.java* quien tiene la instrucción de realizar la conexión con la base de datos repositorio y accesa a la tabla administrador donde se encuentran los datos correspondientes a los que se utilizan en la página.

```

package conexiones;

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.swing.JOptionPane;

public class admin extends HttpServlet {

    protected void processRequest(HttpServletRequest request,
    HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();

        try {

            if(!request.getParameter("user").equals("admin") ||
            !request.getParameter("pass").equals("4dm!N"))
            {
                JOptionPane.showMessageDialog(null, "Nombre de
                usuario o contraseña incorrectos");
                response.sendRedirect("indeex.jsp");
                return;
            }
        }
    }
}

```

```

        }

        response.sendRedirect("administrador1.jsp");

    } finally {
        out.close();
    }
}

protected void doGet(HttpServletRequest request,
    HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

protected void doPost(HttpServletRequest request,
    HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

public String getServletInfo() {
    return "Short description";
}
}

```

Anexo 3.-En la clase llamada *NewTema.java* tiene la instrucción de establecer la conexión del sistema con la base de datos, la cual hace referencia a la tabla tema , que contiene los campos nombre, descripción, entre otros.

```

package conexiones;

import entities.Alumno;
import entities.Cuenta;
import entities.Tema;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.Date;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import javax.swing.JOptionPane;
import jpa.TemaJpaController;

protected void processRequest(HttpServletRequest request,

```

```

HttpServletResponse response)
throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out = response.getWriter();
    try {
        out.print("<%@page session='true'%>");

        HttpSession sesion = request.getSession();
        Tema tema = new Tema();
        TemaJpaController jpaTema = new TemaJpaController();
        boolean b = true;

        tema.setNombre(request.getParameter("tema"));
        tema.setDescripcion(request.getParameter("descripcion"));

        if(tema.getNombre().isEmpty())
        {
            JOptionPane.showMessageDialog(null, "Nombre de tema
requerido");
            response.sendRedirect("indexusuarioregistro.jsp?
descripcion=" +tema.getDescripcion());
            b = false;
        }

        else if(tema.getDescripcion().isEmpty())
        {
            JOptionPane.showMessageDialog(null, "Descripción de
tema requerido");
            response.sendRedirect("indexusuarioregistro.jsp?tema="
+ tema.getNombre());
            b = false;
        }

        tema.setDatospersonalesidDatospersonales(((Cuenta)sesion.getAt
tribute("cuenta")).getDatospersonalesidDatospersonales());
        tema.setFechaProp(new Date());
        if(b)
        {
            jpaTema.create(tema);
            response.sendRedirect("Principal.jsp");
        }

    } finally {
        out.close();
    }
}

```

```

methods. Click on the + sign on the left to edit the code.">
protected void doGet(HttpServletRequest request,
    HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
    }

protected void doPost(HttpServletRequest request,
    HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
    }

    public String getServletInfo() {
        return "Short description";
    }
}
}

```

Anexo 4.-La clase mensajes se utilizar para enviar e informar a los alumnos el estado en el que se encuentra el su proyecto, es decir, si el registro se ha validado correctamente. Para ello es necesario realizar la conexión entre el sistema y la base de datos, en este caso se llama *NuevoMensaje.java* que contiene la instrucción de acceso a la tabla Mensaje de la base de datos Repositorio indicando los campos que el sistema requiere.

```

package conexiones;
import entities.Alumno;
import entities.Mensaje;
import entities.Profesor;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.swing.JOptionPane;
import jpa.AlumnoJpaController;
import jpa.MensajeJpaController;
import jpa.ProfesorJpaController;

public class NuevoMensaje extends HttpServlet {
protected void processRequest(HttpServletRequest request,

```

```

HttpServletResponse response)
throws ServletException, IOException {
response.setContentType("text/html;charset=UTF-8");
PrintWriter out = response.getWriter();
    try {

        String cuenta = request.getParameter("cuenta");
        String msj = request.getParameter("mensaje");

        Mensaje mensaje;
        MensajeJpaController mensajeJpaController = new
        MensajeJpaController();
        boolean b = true;
        for(Alumno a : new
        AlumnoJpaController().findAlumnoEntities())
            if(a.getIdNoCuenta().equals(cuenta))
                {
                    mensaje = new Mensaje();
                    mensaje.setMensaje(msj);
                    mensaje.setDatospersonalesidDatospersonales(a.
                    getDatospersonalesidDatospersonales());
                    mensajeJpaController.create(mensaje);
                    b = false;
                    response.sendRedirect("administrador1.jsp");
                    break;
                }
        if(b)
        {
            for(Profesor p : new
            ProfesorJpaController().findProfesorEntities())
                if(p.getIdMatricula().equals(cuenta))
                    {
                        mensaje = new Mensaje();
                        mensaje.setMensaje(msj);
                        mensaje.setDatospersonalesidDatospersonales
                        (p.getDatospersonalesidDatospersonales());
                        mensajeJpaController.create(mensaje);
                        b = false;
                        response.sendRedirect("administrador1.jsp");
                        break;
                    }
        }
        if(b)
        {
            JOptionPane.showMessageDialog(null, "No existe el
            usuario: " + cuenta);
        }
    }
}

```

```

        response.sendRedirect("administrador1.jsp");
    }

    } finally {
        out.close();
    }
}

methods. Click on the + sign on the left to edit the code.">
protected void doGet(HttpServletRequest request,
    HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

protected void doPost(HttpServletRequest request,
    HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

public String getServletInfo() {
    return "Short description";
}
}

```

Anexo 5.-La clase alumno tiene como instrucción realizar la conexión entre los campos que solicita el sistema y la base de datos que es la que permite el correcto almacenamiento así como el acceso y búsqueda de la información ya almacenada o bien los datos que están por ser guardados. En este caso se ha creado un servlet llamado *conexionAlumno.java*.

```

package conexiones;
import entities.Alumno;
import entities.Cuenta;
import entities.DatosPersonales;
import entities.Direccion;
import entities.Telefono;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.Date;
import javax.servlet.ServletException;

```

```

import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.swing.JOptionPane;
import jpa.AlumnoJpaController;
import jpa.CuentaJpaController;
import jpa.DatosPersonalesJpaController;
import jpa.DireccionJpaController;
import jpa.TelefonoJpaController;

public class conexionAlumno extends HttpServlet {
protected void processRequest(HttpServletRequest
request, HttpServletResponse response)
throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out = response.getWriter();

    try {
        DatosPersonales datPer = new DatosPersonales();
        Alumno alumno = new Alumno();
        Cuenta cuenta = new Cuenta();
        Direccion direccion = new Direccion();
        Telefono tel = new Telefono();
        boolean b = true;
        String msj = "";

        DatosPersonalesJpaController jpaDatosPersonales =
        new DatosPersonalesJpaController();
        CuentaJpaController jpaCuenta = new
        CuentaJpaController();
        AlumnoJpaController jpaAlumno = new
        AlumnoJpaController();
        DireccionJpaController jpaDireccion = new
        DireccionJpaController();
        TelefonoJpaController jpaTelefono = new
        TelefonoJpaController();

        datPer.setNombre(request.getParameter("nombre"));
        datPer.setApaterno(request.getParameter("paterno"));
        datPer.setAmaterno(request.getParameter("materno"));
        datPer.setSexo(request.getParameter("sexo"));

        if (datPer.getNombre().isEmpty()) {
            msj = "Nombre\n";
            b = false;
        }
    }
}

```

```

        if (datPer.getApaterno().isEmpty()) {
            msj += "Apellido paterno\n";
            b = false;
        }

        if (datPer.getAmaterno().isEmpty()) {
            msj += "Apellido materno\n";
            b = false;
        }

        Date d = !request.getParameter("fnac").isEmpty()?
new Date(request.getParameter("fnac")) : new Date();

        alumno.setDatospersonalesidDatospersonales(datPer);
        alumno.setCarrera(request.getParameter("carrera"));
        alumno.setFechaNacimiento(d);
        alumno.setIdNoCuenta(request.getParameter("cuenta"));
        alumno.setSemestre(request.getParameter("semestre"));

        if (alumno.getCarrera().isEmpty()) {
            msj += "Carrera\n";
            b = false;
        }

        tel.setDatospersonalesidDatospersonales(datPer);
        tel.setTelefono(request.getParameter("telcel"));
        tel.setRol("Celular");

        if (tel.getTelefono().isEmpty()) {
            msj += "Telefono\n";
            b = false;
        }

        if (!tel.getTelefono().matches("[0-9]+") ||
tel.getTelefono().length() > 12 ||
tel.getTelefono().length() < 7) {
            JOptionPane.showMessageDialog(null, "Telefono
invalido");
            b = false;
        }

        direccion.setAlumnoidnocuenta(alumno);
        direccion.setCalle(request.getParameter("calle"));
        direccion.setColonia(request.getParameter("colonia"));
        direccion.setNo(request.getParameter("numero"));

```

```

direccion.setEstado(request.getParameter("estado"));
direccion.setCodPostal(request.getParameter("cp"));
direccion.setMunicipio(request.getParameter("municipio"));
cuenta.setDatospersonalesidDatospersonales(datPer);
cuenta.setUsuario(request.getParameter("usuario"));
cuenta.setCorreoAlternativo(request.getParameter("correo"
));
        cuenta.setPreguntaSecreta(request.getParameter("
pregunta"));
        cuenta.setRespuestaSecreta(request.getParameter("
respuesta"));
        cuenta.setContraseña(util.MD5.getMd5(request.get
Parameter("pass")));

        if (cuenta.getUsuario().isEmpty()) {
            msj += "Usuario\n";
            b = false;
        }

        if (cuenta.getCorreoAlternativo().isEmpty()) {
            msj += "Correo\n";
            b = false;
        }

        if (request.getParameter("pass").isEmpty() ||
request.getParameter("pass2").isEmpty()) {
            msj += "Contraseña\n";
            b = false;
        }

        if (!request.getParameter("pass").equals(request.
getParameter("pass2"))) {
            JOptionPane.showMessageDialog(null, "La contraseña
no coincide");
            b = false;
        }

        if (!b) {
            if (!msj.isEmpty()) {
                JOptionPane.showMessageDialog(null, "Los
siguientes datos son requeridos: \n" + msj);
            }
            response.sendRedirect("indexprealum.jsp?nombre=" +
datPer.getNombre() + "&paterno=" + datPer.
getApaterno()+ "&materno=" + datPer.getAmaterno()+
"&fnac=" + request.getParameter("fnac") + "&carrera="

```

```

+ alumno.getCarrera() + "&cuenta=" + alumno.getIdNo
Cuenta() + "&semestre=" + alumno.getSemestre()
+ "&telcel=" + tel.getTelefono() + "&calle=" +
direccion.getCalle() + "&numero=" + direccion.getNo()
+ "&colonia=" + direccion.getColonia() + "&municipio="
+ direccion.getMunicipio() + "&estado=" +
direccion.getEstado() + "&usuario=" + cuenta.getUsuario()
+ "&correo=" + cuenta.getCorreoAlternativo());
return;
}

for(Cuenta c: jpaCuenta.findCuentaEntities())
if(c.getUsuario().equals(cuenta.getUsuario()))
{
    JOptionPane.showMessageDialog(null, "Nombre
de usuario existente");
    response.sendRedirect("indexprealum.jsp?nombre=" +
datPer.getNombre() + "&paterno=" + datPer.
getApaterno() + "&materno=" + datPer.
getAmaterno() + "&fnac=" + request.getParameter
("fnac") + "&carrera=" + alumno.getCarrera()
+ "&cuenta=" + alumno.getIdNoCuenta() +
"&semestre=" + alumno.getSemestre()
+ "&telcel=" + tel.getTelefono() + "&calle=" +
direccion.getCalle() + "&numero=" +
direccion.getNo() + "&colonia=" + direccion
.getColonia() + "&municipio=" + direccion.
getMunicipio() + "&estado=" + direccion.getEstado()
+ "&usuario=" + cuenta.getUsuario() +
"&correo=" + cuenta.getCorreoAlternativo());
    b = false;
    break;
}
if(b)
{

    jpaDatosPersonales.create(datPer);
    jpaAlumno.create(alumno);
    jpaDireccion.create(direccion);
    jpaCuenta.create(cuenta);
    jpaTelefono.create(tel);

    response.sendRedirect("datosBien.jsp");
}
} catch (Exception e) {
    e.printStackTrace();
}

```

```

        } finally {
            out.close();
        }
    }
}
methods. Click on the + sign on the left to edit the code.">
protected void doGet(HttpServletRequest request,
    HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

protected void doPost(HttpServletRequest request,
    HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

public String getServletInfo() {
    return "Short description";
}
}

```

Anexo 6.-En la clase *conexionProfesor.java* se creó un servlet que contiene las instrucciones para realizar la conexión entre el sistema Web y la tabla profesor. Los campos a los que el sistema puede ingresar, en esta clase, son nombre, apaterno, amaterno, matrícula, entre otros.

```

package conexiones;
import entities.Cuenta;
import entities.DatosPersonales;
import entities.Profesor;
import entities.Telefono;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.swing.JOptionPane;
import jpa.CuentaJpaController;
import jpa.DatosPersonalesJpaController;
import jpa.ProfesorJpaController;
import jpa.TelefonoJpaController;
import jpa.exceptions.PreexistingEntityException;

```

```

public class conexionProfesor extends HttpServlet {
protected void processRequest(HttpServletRequest request,
HttpServletResponse response)
throws ServletException, IOException {
response.setContentType("text/html;charset=UTF-8");
PrintWriter out = response.getWriter();
    try {
        DatosPersonales datPer = new DatosPersonales();
        Profesor prof = new Profesor();
        Cuenta cuenta = new Cuenta();
        Telefono tel = new Telefono();
        boolean b = true;
        String msj = "";

        DatosPersonalesJpaController jpaDatosPersonales =
        new DatosPersonalesJpaController();
        ProfesorJpaController jpaProfesor = new
        ProfesorJpaController();
        CuentaJpaController jpaCuenta = new
        CuentaJpaController();
        TelefonoJpaController jpaTelefono = new
        TelefonoJpaController();

        datPer.setNombre(request.getParameter("nombre"));
        datPer.setApaterno(request.getParameter("paterno"));
        datPer.setAmaterno(request.getParameter("materno"));
        datPer.setSexo(request.getParameter("sexo"));

        if (datPer.getNombre().isEmpty()) {
            msj = "Nombre\n";
            b = false;
        }

        if (datPer.getApaterno().isEmpty()) {
            msj += "Apellido paterno\n";
            b = false;
        }

        if (datPer.getAmaterno().isEmpty()) {
            msj += "Apellido materno\n";
            b = false;
        }

        prof.setDatospersonalesidDatospersonales(datPer);
        prof.setIdMatricula(request.getParameter("matricula"));
    }
}

```

```

prof.setAreaAcademica(request.getParameter("areaacademica"));
prof.setEscolaridad(request.getParameter("escolaridad"));

    if (prof.getIdMatricula().isEmpty()) {
        msj += "Matricula\n";
        b = false;
    } else if (prof.getIdMatricula().length() < 4 ||
prof.getIdMatricula().length() > 15) {
        JOptionPane.showMessageDialog(null, "Matricula
invalida");
        b = false;
    }

    if (prof.getAreaAcademica().isEmpty()) {
        msj += "Area academica\n";
        b = false;
    }
}

cuenta.setDatosPersonalesidDatosPersonales(datPer);
cuenta.setUsuario(request.getParameter("usuario"));
cuenta.setCorreoAlternativo(request.getParameter("correo"));
cuenta.setPreguntaSecreta(request.getParameter("pregunta"));
cuenta.setRespuestaSecreta(request.getParameter("respuesta"));
cuenta.setContraseña(util.MD5.getMD5(request.getParameter("pass")));

    if (cuenta.getUsuario().isEmpty()) {
        msj += "Usuario\n";
        b = false;
    }

    if (cuenta.getCorreoAlternativo().isEmpty()) {
        msj += "Correo\n";
        b = false;
    }

if(request.getParameter("pass").isEmpty() ||
request.getParameter("pass2").isEmpty())
{
    msj += "Contraseña\n";
    b = false;
}
}

```

```

    if (!request.getParameter("pass").equals(request.getParameter("pass2"))) {
        JOptionPane.showMessageDialog(null, "La contraseña no coincide");
        b = false;
    }

    tel.setDatospersonalesidDatospersonales(datPer);
    tel.setTelefono(request.getParameter("telcel"));
    tel.setRol("Celular");

    if (tel.getTelefono().isEmpty()) {
        msj += "Telefono\n";
        b = false;
    }

    if (!tel.getTelefono().matches("[0-9]+") ||
tel.getTelefono().length() > 12 ||
tel.getTelefono().length() < 7) {
        JOptionPane.showMessageDialog(null,
"Telefono invalido");
        b = false;
    }

    if(!cuenta.getCorreoAlternativo().matches("^[0-9a-
zA-Z]([_\\.w]*[0-9a-zA-Z])*@[0-9a-zA-Z]([-w]*[0-9a-zA-
Z].)+([a-zA-Z]{2,9}.)+[a-zA-Z]{2,3}$"))
    {
        JOptionPane.showMessageDialog(null, "Correo
invalido");
        b = false;
    }

    if (!b) {
        if(!msj.isEmpty())
            JOptionPane.showMessageDialog(null, "Los
siguientes datos son requeridos: \n" + msj);

        response.sendRedirect("indexpreprof.jsp?nombre=" +
datPer.getNombre() + "&paterno=" + datPer.getApaterno() +
"&materno=" + datPer.getAmaterno() + "&matricula=" +
prof.getIdMatricula() + "&areaacademica=" +
prof.getAreaAcademica() + "&usuario=" +
cuenta.getUsuario() + "&correo=" +
cuenta.getCorreoAlternativo());
        return;
    }

```

```

    }

    for(Cuenta c: jpaCuenta.findCuentaEntities())
        if(c.getUsuario().equals(cuenta.getUsuario()))
        {
            JOptionPane.showMessageDialog(null, "Nombre
            de usuario existente");
            response.sendRedirect("indexpreprof.jsp?nombre=" +
            datPer.getNombre() + "&paterno=" + datPer.getApaterno() +
            "&materno=" + datPer.getAmaterno() + "&matricula=" +
            prof.getIdMatricula() + "&areaacademica=" +
            prof.getAreaAcademica() + "&usuario=" +
            cuenta.getUsuario() + "&correo=" +
            cuenta.getCorreoAlternativo());
            b = false;
            break;
        }

    for(Profesor p: jpaProfesor.findProfesorEntities())
        if(p.getIdMatricula().equals(prof.getIdMatricula()))
        {
            JOptionPane.showMessageDialog(null, "Matricula
            existente");
            response.sendRedirect("indexpreprof.jsp?nombre=" +
            datPer.getNombre() + "&paterno=" + datPer.getApaterno() +
            "&materno=" + datPer.getAmaterno() + "&matricula=" +
            prof.getIdMatricula() + "&areaacademica=" +
            prof.getAreaAcademica() + "&usuario=" +
            cuenta.getUsuario() + "&correo=" +
            cuenta.getCorreoAlternativo());
            b = false;
            break;
        }

    if(b)
    {
        jpaDatosPersonales.create(datPer);
        jpaProfesor.create(prof);
        jpaCuenta.create(cuenta);
        jpaTelefono.create(tel);

        tel = new Telefono();

        tel.setDatospersonalesidDatospersonales(datPer);
        tel.setTelefono(request.getParameter("teloficina"));
        tel.setRol("Oficina");
    }
}

```

```

        if(!tel.getTelefono().isEmpty())
            jpaTelefono.create(tel);

        response.sendRedirect("datosBien.jsp");
    }

    } catch (PreexistingEntityException peee) {
        peee.printStackTrace();
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        out.close();
    }
}

}

methods. Click on the + sign on the left to edit the code.">
protected void doGet(HttpServletRequest request,
HttpServletResponse response)
throws ServletException, IOException {
processRequest(request, response);
}

protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
processRequest(request, response);
}

public String getServletInfo() {
return "Short description";
}
}
}

```

Anexo 7.-La clase **regProyecto.java** va a controlar todos los datos que se utilizan al momento de solicitar el registro del tema de un proyecto y para ello se tiene como instrucción establecer la conexión entre el sistema y la base de datos la cual contiene información requerida al momento de realizar dicha solicitud, en este caso se necesitan los datos personales del alumno y datos del proyecto a registrar.

En este apartado se realiza cualquier tipo de operación que se deseen hacer a los datos, es decir, si necesitamos realizar una consulta o bien modificarlos.

```

package conexiones;
import entities.Alumno;
import entities.IntegrantesProyecto;
import entities.Profesor;
import entities.Proyecto;

```

```

import entities.Tema;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.Calendar;
import java.util.Date;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.swing.JOptionPane;
import jpa.AlumnoJpaController;
import jpa.IntegrantesProyectoJpaController;
import jpa.ProfesorJpaController;
import jpa.ProyectoJpaController;

public class regProyecto extends HttpServlet {

    protected void processRequest(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        try {

            String strA1 = request.getParameter("alumno1");
            String strA2 = request.getParameter("alumno2");
            String strProfe = request.getParameter("profe");

            if(strProfe.isEmpty())
            {

                JOptionPane.showMessageDialog(null, "Necesitas un
                asesor para registrar el tema");
                response.sendRedirect("REGISTROPROYECTO.jsp?alumno1=" +
                strA1 + "&alumno2=" + strA2);
                return;
            }
            AlumnoJpaController alumnoJpa = new
            AlumnoJpaController();
            ProfesorJpaController profesorJpa = new
            ProfesorJpaController();
            IntegrantesProyectoJpaController integrantesJpa
            Controller = new IntegrantesProyectoJpaController();

            Alumno alumno1 = null;

```

```

        Alumno alumno2 = null;
        Profesor profesor = null;

        boolean b = true;

        //Buscando alumnos

        for(Alumno a: alumnoJpa.findAlumnoEntities())
        {
            if(a.getIdNoCuenta().equals(strA1))
                alumno1 = a;
            if(a.getIdNoCuenta().equals(strA2))
                alumno2 = a;
        }

        for(IntegrantesProyecto ip : integrantesJpaController.
        findIntegrantesProyectoEntities())
        {
            if(alumno1 != null && ip.get
            DatospersonalesidDatospersonales().
            equals(alumno1.getDatospersonales
            idDatospersonales()))
            {
                JOptionPane.showMessageDialog(null, "El alumno " +
                ip.getDatospersonalesidDatospersonales().getNombre()+
                " " + ip.getDatospersonalesidDatospersonales().getApaterno()
                + " " + ip.getDatospersonalesidDatospersonales().getAmaterno()
                + " ya se encuentra registrado en un proyecto");
                b = false;
                response.sendRedirect("indexusuario.jsp");
            }

            if(alumno2 != null && ip.getDatospersonalesidDatospersonales
            ().equals(alumno1.getDatospersonalesidDatospersonales()))
            {
                JOptionPane.showMessageDialog(null, "El alumno " + ip
                .getDatospersonalesidDatospersonales().getNombre()
                + " " + ip.getDatospersonalesidDatospersonales().getApaterno()
                + " " + ip.getDatospersonalesidDatospersonales().getAmaterno()
                + " ya se encuentra registrado en un proyecto");
                b = false;
                response.sendRedirect("indexusuario.jsp");
            }
        }
    }
}

```

```

//Buscando Profesor
for(Profesor p : profesorJpa.findProfesorEntities())
if(p.getIdMatricula().equals(strProfe))
{
    profesor = p;
    break;
}

if(!strA1.isEmpty() && alumno1 == null)
{
    JOptionPane.showMessageDialog(null, "No existe un
alumno con el numero de cuenta " + strA1
+ ". Necesitas darlo de alta en la seccion de
pre- registro");
    b = false;
}

if(!strA2.isEmpty() && alumno2 == null)
{
    JOptionPane.showMessageDialog(null, "No existe un
alumno con el numero de cuenta " + strA2
+ ". Necesitas darlo de alta en la seccion de
pre- registro");
    b = false;
}

if(profesor == null)
{
    JOptionPane.showMessageDialog(null, "No existe
un profesor con la matricula " + strProfe + "
. Necesitas darlo de alta en la seccion de
pre-registro");
    b = false;
}

if(b)
{
    ProyectoJpaController proyectoJpaController = new
ProyectoJpaController();
    //IntegrantesProyectoJpaController
integrantesJpaController = new
IntegrantesProyectoJpaController();

    Proyecto proyecto = new Proyecto();
    IntegrantesProyecto integrantes;
}

```

```

Calendar calendario = Calendar.getInstance();

Calendario.add(Calendar.YEAR, 2);
Calendario.add(Calendar.MONTH, 6);

    proyecto.setTemaidtema(new Tema(Integer.parseInt(
    request.getParameter("temas"))));
    proyecto.setModalidad(request.getParameter
("modalidad"));
    proyecto.setEstatus("pre registrado");
    proyecto.setFechaInicio(new Date());
    proyecto.setFechaFin(calendario.getTime());

    proyectoJpaController.create(proyecto);

    if(alumno1 != null)
    {
        integrantes = new IntegrantesProyecto();
        integrantes.setDatospersonalesidDatospersonales(alumno1.get
DatospersonalesidDatospersonales());
        integrantes.setProyectooidproyecto(proyecto);
        integrantes.setRol("Tesista");
        integrantesJpaController.create(integrantes);
    }

    if(alumno2 != null)
    {
        integrantes = new IntegrantesProyecto();
        integrantes.setDatospersonalesidDatospersonales
(alumno2.getDatospersonalesidDatospersonales());
        integrantes.setProyectooidproyecto(proyecto);
        integrantes.setRol("Tesista");
        integrantesJpaController.create(integrantes);
    }

    integrantes = new IntegrantesProyecto();

    integrantes.setDatospersonalesidDatospersonales
(profesor.getDatospersonalesidDatospersonales());
    integrantes.setProyectooidproyecto(proyecto);
    integrantes.setRol("Asesor");
    integrantesJpaController.create(integrantes);

    response.sendRedirect("proyectobien.jsp");
}
} finally {

```

```

        out.close();
    }
}

methods. Click on the + sign on the left to edit the code.">
protected void doGet(HttpServletRequest request,
    HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

protected void doPost(HttpServletRequest request,
    HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

public String getServletInfo() {
    return "Short description";
}
}

```

Anexo 8.-En la clase llamada *Iniciarsesion.java* es un servlet que se conectará con el servidor y mandará las instrucciones para que se lleve a cabo la conexión entre el sistema y la base de datos, enlazándose específicamente con la tabla llamada cuenta que contiene los campos como usuario y contraseña.

```

package conexiones;
import entities.Alumno;
import entities.Cuenta;
import entities.Profesor;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.List;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import jpa.AlumnoJpaController;
import jpa.CuentaJpaController;
import jpa.ProfesorJpaController;

public class iniciarsesion extends HttpServlet {

```

```

protected void processRequest(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {
response.setContentType("text/html;charset=UTF-8");
PrintWriter out = response.getWriter();

out.println("<%@page session='true'%>");
HttpSession sesion = request.getSession();
List<Cuenta> listaCuenta;
List<Profesor> listaProfe;
Cuenta cuenta = null;
Profesor prof = null;
CuentaJpaController jpaCuenta = new
CuentaJpaController();
ProfesorJpaController jpaProfesor = new ProfesorJpaController();
AlumnoJpaController jpaAlumno = new AlumnoJpaController();

String usuario = request.getParameter("usuario");
String pass = request.getParameter("contraseña");

try {
out.println("<html>");
out.println("<head>");
out.println("<title>Servlet iniciarsesion</title>");
out.println("</head>");
out.println("<body>");

listaCuenta = jpaCuenta.findCuentaEntities();

for(int i = 0; i < listaCuenta.size(); i++)
{
if(listaCuenta.get(i).getUsuario().equals(usuario)
&&listaCuenta.get(i).getContraseña().equals(util
.MD5.getMD5(pass)))
{
cuenta = listaCuenta.get(i);
break;
}
}

if(cuenta != null)
{
sesion.setAttribute("cuenta", cuenta);
listaProfe = jpaProfesor.findProfesorEntities();

```

```

for(int i = 0; i < listaProfe.size(); i++)
{
    if(listaProfe.get(i).getDatospersonalesid
Datospersonales().equals(cuenta.getDatospersonales
idDatospersonales()))
    {
        prof = listaProfe.get(i);
        break;
    }
}

if(prof != null)
{
    sesion.setAttribute("profesor", prof);
    sesion.setMaxInactiveInterval(300);
    sesion.setAttribute("alumno", null);
    response.sendRedirect("indexusuarioprofe.jsp");
}
else
{
    for(Alumno al : jpaAlumno.findAlumnoEntities())
        if(al.getDatospersonalesidDatospersonales().equals
(cuenta.getDatospersonalesidDatospersonales()))
        {
            sesion.setAttribute("alumno", al);
            sesion.setMaxInactiveInterval(300);
            sesion.setAttribute("profesor", null);
            break;
        }
    response.sendRedirect("indexusuario.jsp");
}

else
    response.sendRedirect("datosMal.jsp");

out.println("</body>");
out.println("</html>");

} finally {
    out.close();
}
}

```

methods. Click on the + sign on the left to edit the code.">
protected void doGet(HttpServletRequest request,

```
    HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
    }

    protected void doPost(HttpServletRequest request,
    HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
    }

    public String getServletInfo() {
    return "Short description";
    }
    }
```

Bibliografía

- [1] ROGER S. PRESSMAN, *Ingeniería del Software, Un enfoque práctico*, Mc Graw Hill, 6a. Edición.
- [2] PAUL S. WANG, *Java Con programación orientada a objetos y aplicaciones en la www*, Ed. Thomson, 1994.
- [3] JACOPSON RUMBAUGH BOOCH, *Lenguaje unificado del modelado*, 1995.
- [4] H.M DEITEL Y P.J. DEITEL, *Como programar en C/C++*, Segunda Edición.
- [5] SUN MICROSYSTEMS, *Object-Oriented Application analysis and Design for Java Technology (UML)*, Ed. Sun Microsystems, OO-226.
- [6] SUN MICROSYSTEMS, *Programación Java para no Programadores*, Ed. Sun Microsystems.
- [7] <http://www.orientedigital.com.mx/secc/faq.php?idfaq=4>.
- [8] <http://www.desarrolloweb.com/manuales/>.
- [9] <http://www.maestrosdelweb.com/editorial/aspintro/>.
- [10] ULLMAN. LARRY, *MySQL*, Ed. Pearson Educación S.A, 2004.
- [11] PIERRE. ARMAND, *HTML - Creacion y difusión de páginas web*, Ed. Trillas, 1999.
- [12] ELMASRI. RAMIREZ A. / NAVATHE, SHAMKANT B., *Fundamentos de Sistemas de Bases de Datos*, España: Pearson Prentice Hall, 3a. Edición.
- [13] ORÓS, JUAN CARLOS, *Diseño de páginas web interactivas con Javascript y CSS*, Mexico: Alfaomega Ra.Ma., 3a. Edición.
- [14] VILLEGAS LARA, MAGIN, *Desarrollo de un sistema de información orientado a la web. Caso de Estudio: Área Académica de Ciencias Computacionales*, Tesis de licenciatura, Universidad Autonoma del Estado de Hidalgo.