



Benemérita Universidad Autónoma De Puebla

Facultad de Ciencias de la Computación

**“Sistema Administrador del Seguimiento de
Proyectos de Investigación”**

Tesina
Que para obtener el título de:
Licenciado en Ciencias de la Computación

Presenta:
Juan Zamora Cruz

Asesor: M. C. Eugenia Erica Vera Cervantes.
Puebla, Pue. Abril 2013

CONTENIDO

	Pág.
INTRODUCCION.	4
1. MARCO TEORICO.	
1.1. Sistemas de Información	6
1.2. Tecnologías de Información	6
1.3. Ingeniería de Software	8
1.3.1. Objetivos de la Ingeniería de Software	8
1.3.2. Aspectos del Proceso de Desarrollo	8
1.3.3. Ciclo de Vida	9
1.3.4. Tipos de modelos de Ciclos de Vida	10
1.3.4.1. Modelo de Cascada	10
1.3.4.2. Modelo Incremental	11
1.3.4.3. Modelo Espiral	12
1.3.4.4. Modelo de Construcción de Prototipos	14
1.3.5. Rol del Cliente	15
1.4. Metodologías de Desarrollo	16
1.5. Proceso Unificado de Desarrollo de Software (PUDS)	17
1.5.1. PUDS dirigido a Casos de Uso	18
1.5.2. PUDS centrado en la Arquitectura	19
1.5.3. PUDS Iterativo e Incremental	19
1.6. Elementos de Implementación	21
2. ANALISIS Y DISEÑO DEL SISTEMA.	
2.1. UML: Lenguaje Unificado de Modelado	23
2.1.1. Historia	23
2.1.2. PUDS emplea UML	24
2.2. Modelos UML	24
2.2.1. Diagramas	25
2.3. Análisis de Sistema de Tesina	25
2.4. Planeación del proyecto de software	26
2.4.1. Definición del problema	26
2.4.2. Desarrollo de estrategias de solución	26
2.4.3. Adecuación del Producto	26
2.5. Análisis de Requerimientos	27
2.5.1. Casos de Uso UML	27
2.5.2. Diseño arquitectónico o estructural	39

3. DISEÑO DE BASE DE DATOS.

3.1. Definición de Base de Datos	42
3.2. Sistema de Gestión de Base de Datos	42
3.3. Sistemas Manejadores de Base de Datos	43
3.4. Componentes de Base de Datos	44
3.5. Enfoque de Base de Datos	44
3.5.1. Enfoque Relacional	45
3.5.2. Enfoque Jerárquico	45
3.5.3. Enfoque de Red	45
3.6. Modelado de Base de Datos del Sistema	45
3.6.1. Modelado Entidad-Relación	45
3.6.2. Normalización del Modelado	48
3.6.2.1. Primera Forma Normal	48
3.6.2.2. Segunda Forma Normal	49
3.6.2.3. Tercera Forma Normal	50

4. DISEÑO DE ENTORNO WEB.

4.1. Antecedentes	54
4.1.1. Principios de Diseño Web	54
4.1.2. Funcionalidad o Usabilidad	54
4.1.3. Estética	56
4.2. Diseño Web del Sistema	58
4.2.1. Flujo de páginas WEB del Sistema	58

5. IMPLEMENTACION DEL SISTEMA.

5.1. Acceso al Sistema	64
5.2. Página de Inicio	64
5.3. Menú de acceso por Rol	65
5.4. Acceso de Alumnos	65
5.5. Menú de Alumnos	66
5.5.1. Actualización de Datos	66
5.5.2. Actualización de Password	67
5.5.3. Consulta de Proyectos	67
5.5.4. Visualización de Documentos	68
5.6. Acceso de Profesores	69
5.7. Menú de Profesores	70
5.7.1. Actualización de Datos	70
5.7.2. Actualización de Password	71
5.7.3. Consulta de Proyectos	71
5.7.4. Visualización de Documentos	72
5.8. Acceso de Administradores	73

5.9. Menú de Administrador	74
5.9.1. Gestor de Administradores	74
5.9.2. Gestor de Alumnos	75
5.9.3. Gestor de Profesores	77
5.9.4. Gestor de Proyectos	78
5.9.5. Gestor de Documentos	79
5.10. Menú de Catálogos	80
5.10.1. Lista y Mantenimiento de Catálogos	81
5.11. Menú de Reportes	82
5.11.1. Reporte de Alumnos, Profesores y Proyectos	83
5.11.2. Reportes Estadísticos	84
CONCLUSIONES.	85
TRABAJO FUTURO.	86
BIBLIOGRAFIA.	87

INTRODUCCION

Cualquier proyecto de investigación nos proporciona procedimientos científicos destinados a recabar información y formular hipótesis sobre un determinado fenómeno social o científico.

Por tanto; el presente trabajo es para que cualquier institución que trabaje con proyectos de investigación cuente con un sistema que le permita administrar los mismos, apoyándose en tener la información a la mano como son los objetivos del proyecto, antecedentes y justificación, teniendo al final la conclusión del mismo, recabando los informes que nos derive de su seguimiento realizado por el administrador del sistema , así como el guardado de la redacción final del mismo, para tener a la mano las conclusiones a las que se llegaron.

Esto nos conlleva a darle la importancia necesaria a un sistema que nos permita gestionar este tipo de documentos para que en el momento que sea requerido por cualquier situación como lo puede ser otro proyecto (de la misma área o similar) lo podamos vincular de forma más rápida y eficiente influyendo positivamente y positivamente en el tiempo de duración del mismo.

Los capítulos se enfocan en la implementación de todas las técnicas de TI para el desarrollo de un proyecto Web que gestione los trabajos de investigación de una institución educativa.

- En el primer capítulo se dan todos los antecedentes de la importancia de un sistema de TI y todo el Marco Teórico para el mejor entendimiento del mismo.
- Dentro del Segundo capítulo se describen todas las necesidades del cliente y a la vez son transformadas en diagramas UML.
- El tercer capítulo se observa las necesidades de almacenamiento mediante el diseño de la Base de Datos.
- El cuarto capítulo visualiza las técnicas para realizar páginas web mediante el diseño visual y navegación por las mismas para envolver al usuario de la manera correcta dentro de un sitio en internet.
- El ultimo capítulo es la implementación del sistema, en el cual se presentan los diferentes accesos a las paginas para gestionar los proyectos de investigación planteados.



Capítulo 1

MARCO TEORICO

1. MARCO TEORICO.

1.1. Sistemas de Información

Actualmente las empresas e instituciones necesitan manejar su información de forma precisa, eficiente y rápida, en la cual el siglo XXI es la época de la sociedad de la información. Todos los componentes de una empresa deben de funcionar de forma coordinada, mejorando su competitividad y disponiendo de la información adecuada para la toma de decisiones.

Un sistema según la Real Academia Española es “un conjunto de cosas que relacionadas entre sí ordenadamente contribuyen a un determinado objetivo”.

Mientras que el concepto de Información es una relación entre los datos recabados y el procesamiento de los mismos de forma que resulten útiles y significativos para el receptor de los mismos.

Los sistemas de Información existen desde que se creó la primera organización humana. Actualmente las Tecnologías de Información automatizan el trabajo de Sistemas de Información apoyando a ejércitos de Oficinistas, Científicos, Administradores, Profesores, Estudiantes, etc. que conllevan una manipulación de datos.

Esta automatización se proyecta mediante tecnologías de modelado de sistemas Web y Metodologías que se han ido mejorando cada vez con mayor fuerza, enfocándose a la globalización que hoy en día nos enfrentamos con el Internet y distribuyéndose mediante las nuevas formas de comunicación.

En la construcción y desarrollo de proyectos de software se aplican métodos y técnicas para resolver los problemas donde las nuevas Tecnologías de Información aportan herramientas y procedimientos sobre los que se apoya de la ingeniería de software en el desarrollo de las bases de datos, facilitando el control del proceso de desarrollo de software y suministrando a los desarrolladores las bases para construir software de alta calidad en una forma eficiente y rápida.

Definir una disciplina que garantice la producción y el mantenimiento de los productos desarrollados de software en el plazo fijado y dentro del costo estimado es una de las tareas a realizar por estas tecnologías.

1.2. Tecnologías de Información.

Es necesario establecer que la tecnología de la información (TI) se entiende como "aquellas herramientas y métodos empleados para recabar, retener, manipular o distribuir información. La tecnología de la información se encuentra generalmente asociada con las computadoras y las tecnologías afines aplicadas a la toma de decisiones.

Utilizando eficientemente la tecnología de la información se pueden obtener ventajas competitivas, pero es preciso encontrar procedimientos acertados para mantener tales ventajas como una constante, así como disponer de cursos y recursos alternativos de acción para adaptarlas a las necesidades del momento, pues las ventajas no siempre son permanentes. El sistema de información tiene que modificarse y actualizarse con regularidad si se desea percibir ventajas competitivas continuas [Jal1993].

En nuestros días, dada la importancia de la información como recurso estratégico que ayuda a generar ventajas competitivas para las instituciones, se tiende a construir sistemas de información cada vez más grandes y complejos donde intervienen equipos de trabajo formados por numerosas personas con distintas funciones y roles.

Sea cual fuese el tamaño del sistema es necesario que siempre nos apoyemos en un proceso de análisis y diseño como el que veremos a lo largo de esta tesina.

Esta unidad servirá de manera introductoria al proceso de desarrollo de software que explotamos en el resto de las demás unidades, vemos sus pilares y los principales conceptos en los que se apoya el desarrollo del sistema planteado.

El siguiente esquema conceptual nos presenta los ejes temáticos fundamentales y sus relaciones, y a modo anticipatorio lo orienta y ayuda a contextualizar cada uno de los conceptos desarrollados.

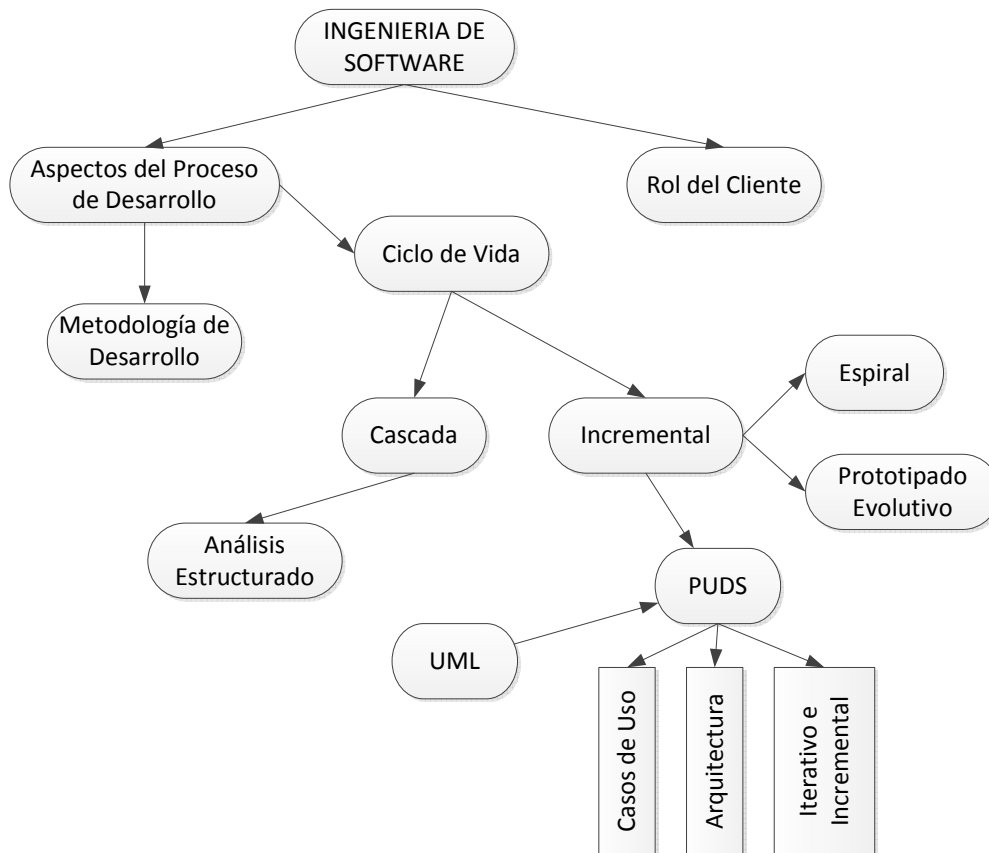


Fig. 1.1 Esquema Conceptual

1.3. Ingeniería de Software.

Ingeniería de software es el estudio de los principios y metodologías para el desarrollo y mantenimiento de sistemas de Información, siendo la aplicación práctica del conocimiento científico, diseño, desarrollo de software, además de la documentación asociada requerida para operarlo y mantenerlo. Conocido también como desarrollo de software o producción de software.

La Ingeniería de software trata el establecimiento de los principios y métodos de la ingeniería a fin de obtener software de modo rentable, que sea fiable y trabaje en máquinas reales.

1.3.1. Objetivos de la Ingeniería de Software.

El proceso del desarrollo de software ha sido el foco de atención en la última década y es el que define un enfoque que se toma cuando el software es tratado por la ingeniería.

En la construcción y desarrollo de proyectos de software se aplican métodos y técnicas para resolver los problemas, la informática aporta herramientas y procedimientos sobre los que se apoya la ingeniería de software para resolver problemas como:

- Mejorar la calidad de los productos de software.
- Aumentar la productividad y trabajo de los Ingenieros del software.
- Facilitar el control del proceso de desarrollo de software.
- Suministrar a los desarrolladores las bases para construir software de alta calidad en una forma eficiente.
- Definir una disciplina que garantice la producción y el mantenimiento de los productos software desarrollados en el plazo fijado y dentro del costo estimado.

1.3.2. Aspectos del Proceso de Desarrollo.

Para llevar a cabo un proceso de desarrollo necesitamos:

- Un Modelo de proceso (ciclo de vida) que nos dé orden de las fases en el desarrollo de software mediante criterios de transición entre las fases.
- Una Metodología de Software que realice la navegación entre cada fase.
- Un enfoque secuencial donde tengamos como premisa de nunca retornar a un paso completado (este es práctico para proyectos pequeños, no críticos) o Iterativo en el cual podamos retornar a cualquier paso ya completado para introducir un cambio, propagando su efecto hacia delante a lo largo del ciclo de vida (la mayoría de los ciclos son iterativos) o un Mecanismo Recursivo donde el cambio en el enfoque

puede ser replicado a los productos finales (todos los enfoques recursivos son iterativos, pero no todos los enfoques iterativos son recursivos).

- Una Metodología de Software
- Un *Análisis de Diseño estructurado* donde realicemos una descomposición funcional de los elementos orientada al flujo de datos.
- Un *Modelado Orientado* a la comunidad de Base de Datos donde concentremos la Información.
- Un *Diagrama de Flujo de Datos* donde mostremos las relaciones funcionales y gráficos cuyos nodos representan procesos y las líneas son el flujo de los datos
- Un *Diagrama de Estados* en el cual mostremos los aspectos dinámicos en el cual los nodos son los Estados y sus líneas son transiciones entre ellos causando eventos.
- Un *Diagrama de Objetos y Clases* donde los nodos de sus gráficos muestran las clases de objetos y sus arcos son relaciones entre las clases y que indica las características relevantes de una clase singular.
- Un *Diagrama Entidad-Relación* que muestre la estructura estática de los componentes del sistema y sus relaciones.

1.3.3. Ciclo de Vida.

El término de ciclo de vida se refiere a la idea de que un producto de software es el resultado de un proceso de desarrollo que se divide en fases.

Podemos definir el término ciclo de vida como la actividad de producir y mantener Sistemas de Información o Informáticos.

Sus funciones son el establecer un orden entre lo siguiente:

- Especificar
- Realizar Prototipo
- Diseñar
- Implementar
- Probar
- Mantener

Además de determinar los criterios para pasar de una fase a otra.

El ciclo de vida está de acuerdo con la metodología que se utilice. Hay metodologías en donde el ciclo de vida tiene un comienzo, se pasa por las distintas etapas antes mencionadas y llega a un fin.

En otras metodologías el ciclo de vida está formado por diversas iteraciones, cada una recorre todas las etapas y se van logrando versiones intermedias del producto. Luego se sigue con otra iteración que pasa nuevamente por todas las etapas y se sigue así

avanzando hasta que se logra el producto completo finalizado. En la definición del plan del proyecto el modelo del ciclo de vida seleccionado influye en el éxito del proyecto como cualquier otra decisión de planificación que se tome.

El modelo del ciclo de vida puede orientar su proyecto y ayudarlo a asegurar que cada paso se acerca más a la consecución del objetivo. Dependiendo del ciclo de vida que se seleccione se puede:

- Aumentar la velocidad del desarrollo
- Mejorar la calidad
- Mejorar el control
- Mejorar el seguimiento del proyecto
- Minimizar gastos
- Minimizar riesgos
- Mejorar las relaciones con el cliente

1.3.4. Tipos de Modelos de Ciclo de Vida

1.3.4.1. Modelo de Cascada

El ciclo de vida inicialmente propuesto por Royce en 1970, fue adaptado para el software a partir de ciclos de vida de otras ramas de la ingeniería. Es el primero de los propuestos y el más ampliamente seguido por las organizaciones (se estima que el 90% de los sistemas han sido desarrollados así).

Este modelo admite la posibilidad de hacer iteraciones, es decir, durante las modificaciones que se hacen en el mantenimiento se puede ver por ejemplo la necesidad de cambiar algo en el diseño, lo cual significa que se harán los cambios necesarios en la codificación y se tendrán que realizar de nuevo las pruebas, es decir, si se tiene que volver a una de las etapas anteriores al mantenimiento hay que recorrer de nuevo el resto de las etapas.

Después de cada etapa se realiza una revisión para comprobar si se puede pasar a la siguiente.

Trabaja en base a documentos, es decir, la entrada y la salida de cada fase es un tipo de documento específico. Idealmente, cada fase podría hacerla un equipo diferente gracias a la documentación generada entre las fases. Los documentos son:

- Análisis.
- Diseño.
- Codificación.
- Pruebas.

Sus *Ventajas* son:

- La planificación es sencilla.
- La calidad del producto resultante es alta.
- Permite trabajar con personal poco cualificado.

Pero sus *Inconvenientes* son:

- Lo peor es la necesidad de tener todos los requisitos al principio. Lo normal es que el cliente no tenga perfectamente definidas las especificaciones del sistema, o puede ser que surjan necesidades imprevistas.
- Si se han cometido errores en una fase es difícil volver atrás.
- No se tiene el producto hasta el final, esto quiere decir que:
 - Si se comete un error en la fase de análisis no lo descubrimos hasta la entrega, con el consiguiente gasto inútil de recursos.
 - El cliente no verá resultados hasta el final, con lo que puede impacientarse.
- No se tienen indicadores fiables del progreso del trabajo (síndrome del 90%).1.1
- Es comparativamente más lento que los demás y el costo es mayor también.

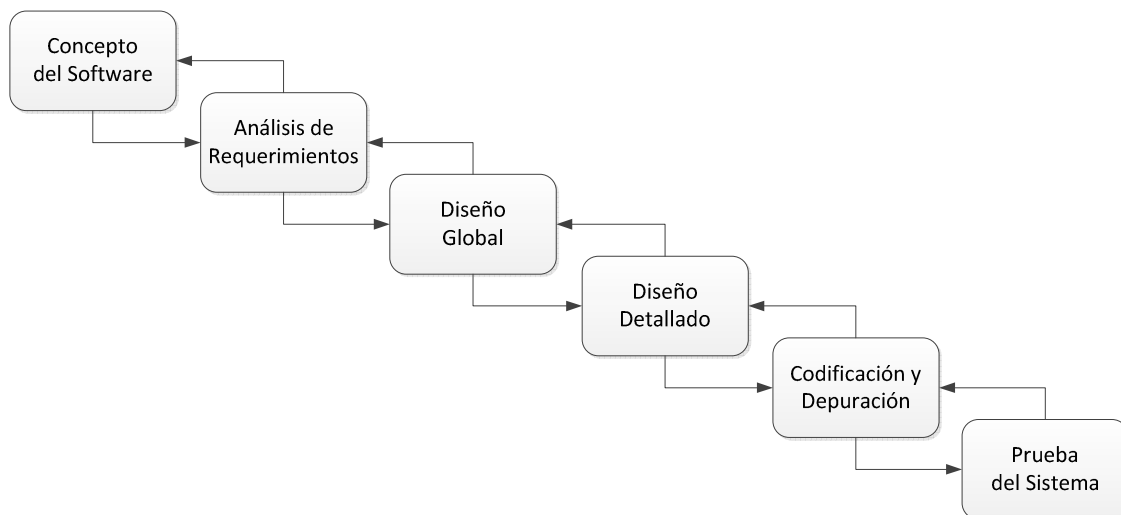


Fig. 2.3 Modelo de Ciclo de Vida "Cascada"

1.3.4.2. Modelo Incremental

Este modelo es basado en la entrega frecuente de software operativo a los clientes. Al realizar entregas semanales, quincenales o mensuales del producto, la comunicación es más efectiva, se reduce el riesgo dividiendo el proyecto en una serie de sub-proyectos más pequeños aumentando la visibilidad del progreso, proporcionando módulos terminados y operativos de un sistema grande antes de tenerlo operativo el sistema completo Tenemos una mayor capacidad para hacer modificaciones a mitad de camino por que el sistema está listo para ser entregado muchas veces durante el desarrollo .

Cada incremento paso por las distintas fases de desarrollo de software con lo cual en las primeras etapas podemos hacer un análisis y mitigación de riesgos mucho más efectiva que en un ciclo en cascada, ya que no tenemos que esperar hasta las fases finales para encontrar, por modificaciones sobre lo que ya hemos realizado. Al ir avanzando de esta forma vamos primero armando el esqueleto de la aplicación y poco a poco lo vamos rellenando hasta lograr una solución completa.

En este caso se va creando el sistema añadiendo pequeñas funcionalidades. Cada uno de los pequeños incrementos es parecido a lo que ocurre dentro de la fase de mantenimiento. La ventaja de este método es que no es necesario tener todos los requisitos en un principio. El inconveniente es que los errores en la detección de requisitos se encuentran tarde.

Hay dos partes en el ciclo de vida, similares al anterior. Por un lado está el análisis y el diseño global. Por otra parte están los pequeños incrementos, con las fases de diseño detallado, codificación y mantenimiento

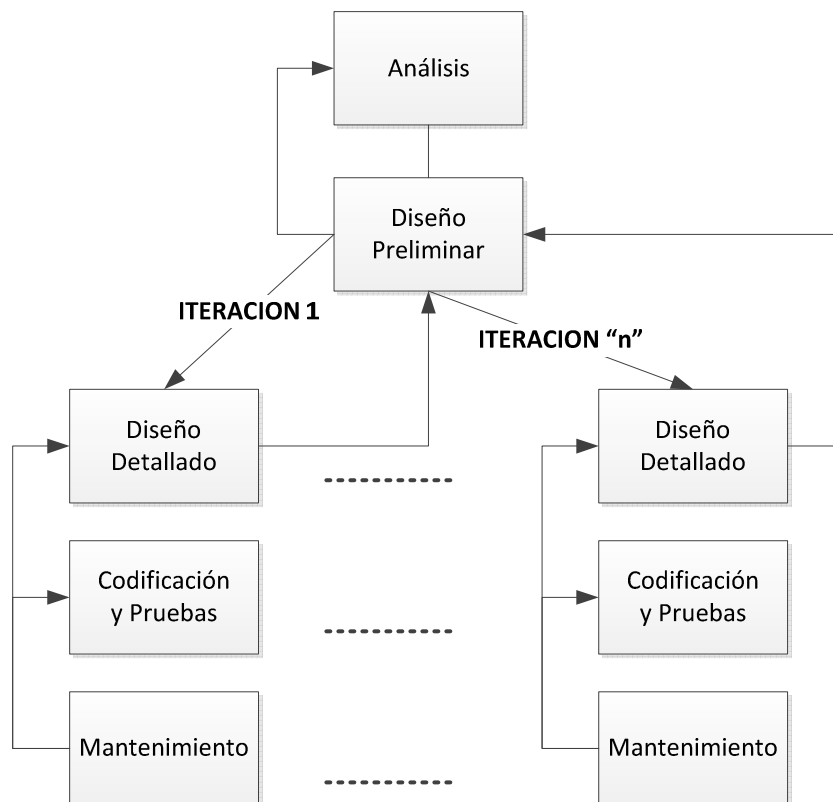


Fig. 4.5 Modelo de Ciclo de Vida "Incremental"

1.3.4.3. Modelo en Espiral

Propuesto inicialmente por Boehm en 1988. Consiste en una serie de ciclos que se repiten. Cada uno tiene las mismas fases y cuando termina da un producto ampliado con

respecto al ciclo anterior. En este sentido es parecido al modelo incremental, la diferencia importante es que tiene en cuenta el concepto de riesgo. Un riesgo puede ser muchas cosas: requisitos no comprendidos, mal diseño, errores en la implementación, etc.

El modelo en espiral se divide en un número de actividades estructurales también llamadas regiones de tareas. Generalmente existen entre tres y seis regiones de tareas.

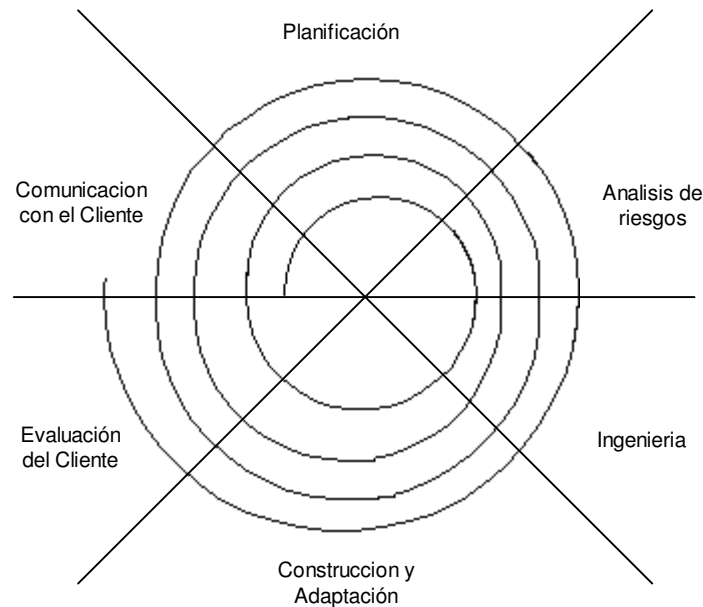


Fig. 6.7 Modelo de Ciclo de Vida "en Espiral"

Cada una de las regiones está poblada por una serie de tareas que se adaptan a las características del proyecto que va a emprender. Para proyectos pequeños, el número de tareas y su formalidad es bajo. Para proyectos mayores y más críticos, cada región contiene tareas que se definen para lograr un nivel más alto de formalidad.

Cuando empieza este proceso, el equipo de ingeniería del software gira alrededor de la espiral en la dirección de las agujas del reloj, comenzando por el centro. El primer circuito de la espiral produce el desarrollo de una especificación de productos; los pasos siguientes en el espiral se podrían utilizar para desarrollar un prototipo y progresivamente versiones más sofisticadas del software. Cada paso de la región de planificación produce ajustes en el plan del proyecto. El coste y la planificación se ajustan según la reacción ante la evaluación del cliente. Además, el gestor de proyecto ajusta el número planificado de iteraciones requeridas para completar el software.

Al terminar una iteración se comprueba que lo que se ha hecho efectivamente cumple con los requisitos establecidos, también se verifica que funciona correctamente. El propio cliente evalúa el producto. No existe una diferencia muy clara entre cuando termina el proyecto y cuando empieza la fase de mantenimiento. Cuando hay que hacer un cambio, este puede consistir en un nuevo ciclo.

1.3.4.4. Modelo de Construcción de Prototipos

El paradigma de construcción de prototipos comienza con la recolección de requisitos. El desarrollador y el cliente encuentran y definen los objetivos globales para el software, identifican los requisitos conocidos, y las áreas del esquema en donde es obligatoria más definición. Entonces aparece un *diseño rápido*. El diseño rápido se centra en una representación de esos aspectos del software que serán visibles para el usuario/cliente (por ejemplo: enfoques de entrada y formatos de salida). El diseño rápido lleva a la construcción de un prototipo. El prototipo lo evalúa el cliente/usuario y lo utiliza para refinar los requisitos del software a desarrollar. La interacción ocurre cuando el prototipo satisface las necesidades del cliente, a la vez que permite que el desarrollador comprenda mejor lo que se necesita hacer.

Lo ideal sería que el prototipo sirviera como un mecanismo para identificar los requisitos del software. Si se construye un prototipo de trabajo, el desarrollador intenta hacer uso de los fragmentos del programa ya existentes o aplica herramientas (p. ej.: generadores de informes, gestores de ventanas, etc.) que permiten generar rápidamente programas de trabajo.

El prototipo puede servir como <<primer sistema>>. Aunque esta puede ser una visión idealizada, es verdad que a los clientes y a los que desarrollan les gusta el paradigma de construcción de prototipos. A los usuarios les gusta el sistema real y a los que desarrollan les gusta construir algo inmediatamente. Sin embargo, la construcción de prototipos también puede ser problemáticas en base a la calidad del software o la facilidad de mantenimiento a largo plazo. Además de esto es demasiado frecuente que la gestión de desarrollo del software sea muy lenta.

Para comenzar la primera iteración, usuarios y analistas identifican de manera conjunta los datos que son necesarios para el sistema y especifican la salida que debe producir la aplicación.

Las decisiones de diseño necesarias para desarrollar la salida del sistema cambian muy poco en relación con las tomadas en otros métodos de desarrollo. Sin embargo, con un prototipo, se espera que las especificaciones iniciales estén incompletas.

Al construir el prototipo se deben seguir los estándares para datos que emplea la organización. En esta etapa es más importante la rapidez con que se construye el prototipo que la eficiencia de operación. Es por esto que el analista no intenta optimizar la velocidad de operación del sistema.

Los cambios al prototipo son planificados con los usuarios antes de llevarlos a cabo. El analista es el responsable de realizar las modificaciones.

El proceso finaliza cuando los usuarios y analistas están de acuerdo en que el sistema ha evolucionado lo suficiente como para incluir todas las características necesarias o cuando ya es evidente que no se obtendrá mayor beneficio [PrR1993].

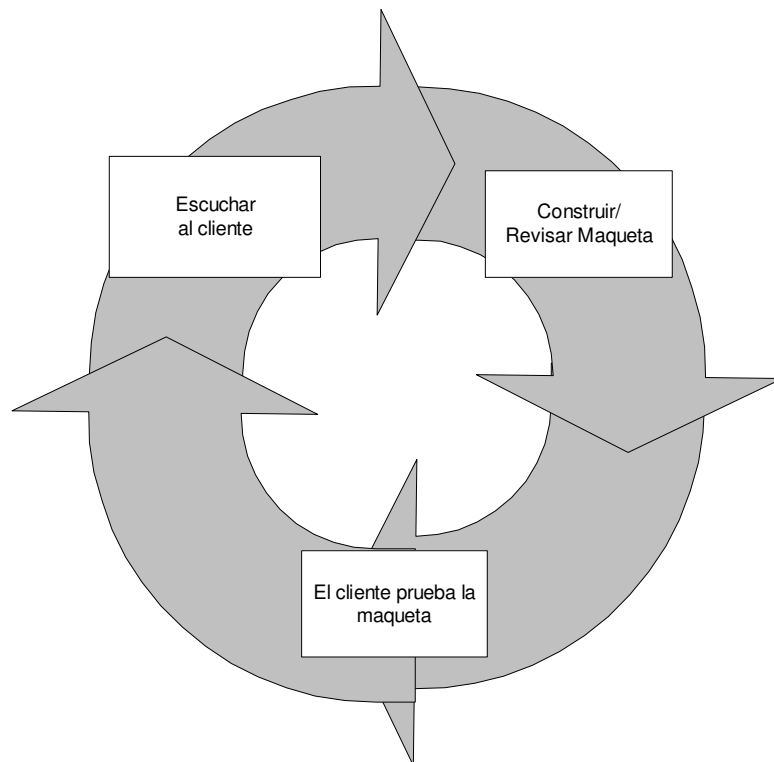


Fig. 8.9 Modelo de Ciclo de Vida "Construcción de Prototipos"

1.3.5. Rol del Cliente

Parte del trabajo de la ingeniera de software es educar a los clientes para que comprendan mejor el desarrollo de software, sin olvidarnos de que el proceso de desarrollo debe estar orientado a él.

En general, los clientes no entienden que el retraso de una semana para realizar la revisión o documentación importante se puede traducir en un retraso de una semana en la entrega del producto. En este sentido, uno de los errores más costosos es el desarrollar un software que sea rechazado por el usuario a último momento. En general, se suelen rechazar partes del software, esto significa que se tiene que volver a diseñar e implementar. Entonces, el efecto global es la entrega retrasada.

Hay 2 razones principales por las que se debe prestar atención a las relaciones con el cliente. Primero, porque aumenta la velocidad del desarrollo al eliminar una fuente de errores e ineficiencia en el desarrollo, Segundo, porque mejora la percepción en el cliente de la velocidad de desarrollo haciendo posible estructurar el proyecto de la manera que el usuario puede ver el progreso y aumentar la confianza.

Una vez que hemos hecho una introducción en la ingeniería de software comenzamos a analizar una metodología de desarrollo de software que debe seguir los lineamientos de los ciclos de vida incrementales.

1.4. Metodologías de Desarrollo.

Cabe Aclarar que cuando hablamos de dominio hacemos referencia a una parte de la realidad del cliente. El problema está en un dominio, mientras que el observador esta fuera del dicho dominio.

Se puede construir software para satisfacer un cliente o para investigar la realidad con un modelo computacional. Para representar un dominio mediante un modelo computacional disponemos de dos enfoques:

1. Representar de forma lo más completa posible un dominio y luego identificar aplicaciones (es un proceso lento , dos años aproximadamente)
2. Construir aplicaciones directamente a partir de los requerimientos (es más rápido y directo)

Deseamos modelar la realidad y para ello realizamos un modelo general y luego desarrollamos aplicaciones particulares de acuerdo a las necesidades del cliente (requerimientos).

Si se piensa que el software como un producto, seguramente ya habremos deducido que necesitamos de un proceso de fabricación. El proceso es la visión dinámica y la metodología es la visión estática.

Un método es una descripción formal de un procedimiento un conjunto de tareas realizadas en determinado orden para alcanzar un objetivo.

Una metodología es un conjunto de métodos a aplicar en un determinado procedimiento, es un conjunto de técnicas, métodos, herramientas, productos y roles para obtener software.

- Permite que se entienda el lenguaje usado
- Permite formar equipos con personas de diferentes niveles
- Está fuertemente asociada al desarrollo y al tipo de software

Así, el proceso es poner en funcionamiento una metodología [LoK2002].

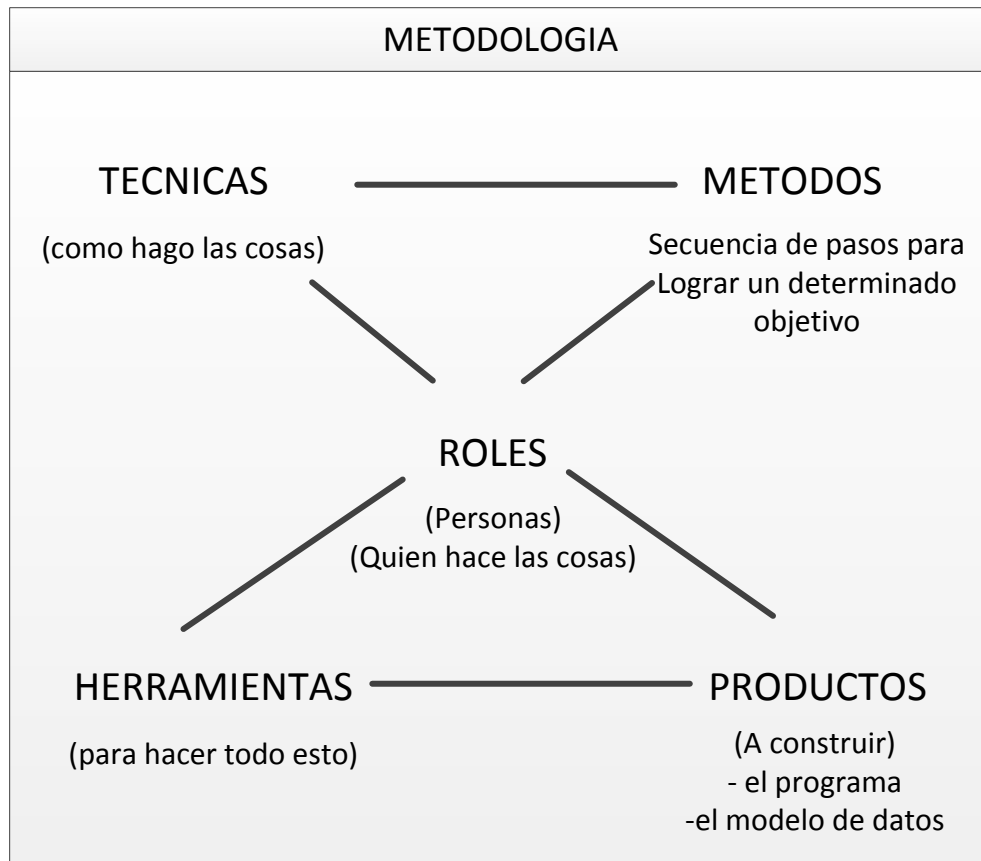


Fig. 10.11 Metodología de desarrollo

1.5. Proceso Unificado de Desarrollo de Software (PUDS).

Encarar cualquier tipo de proyecto sin contar con un proceso de desarrollo puede traer infinidad de problemas antes, durante o después del desarrollo del mismo. Contar con un proceso es contar con una herramienta que nos va ir guiando y sirviendo de bastón de apoyo durante el ciclo de vida de un Sistema, desde sus inicios hasta el mantenimiento del mismo.

Un alto porcentaje de los sistemas que fracasan es por no contar con un proceso de desarrollo, o a la inapropiada utilización del mismo. Muchos piensan que el desarrollo de software comienza con la programación del mismo en un lenguaje determinado, y ahí caen en una de las causantes más importantes de fracaso del sistema.

Sea cual sea el tamaño del sistema es crítico que nos apoyemos en un proceso. Este proceso nos va llevando paso a paso por cada etapa del desarrollo y va guiando y coordinando la intervención de los distintos roles involucrados en el desarrollo del mismo.

Ahora veremos cómo apoyarnos en un proceso para encarar proyectos de desarrollo de software. Este se denomina proceso unificado de desarrollo de software y abarca un

conjunto de actividades necesarias para transformar los requisitos de los distintos usuarios en el Sistema Informático

El proceso unificado utiliza un Lenguaje Unificado de Modelado (Unified Modeling Language, UML) para representar todos los esquemas necesarios en las distintas fases del desarrollo de software, los que veremos a medida que avancemos en el desarrollo de cada fase.

Este proceso se apoya en tres pilares alrededor de los cuales gira la totalidad de la metodología. Estos son los casos de uso, la arquitectura y el concepto de iterativo e incremental.

1.5.1. PUDS dirigido a Casos de Uso

Las necesidades del cliente no son fáciles de detectar. Esto nos lleva que utilicemos algún modo de capturar estas necesidades de manera que posteriormente puedan comunicarse al resto de los intervinientes en el proyecto. Existen entonces dos objetivos fundamentales: encontrar los verdaderos requisitos y representar los mismos de un modo adecuado para los usuarios, clientes y desarrolladores. Un sistema debe dar solución a los que el cliente espera, y los casos de uso nos van a permitir analizar minuciosamente esos requerimientos ellos han sido adoptados casi universalmente para la captura de requisitos de sistemas de información.

Debemos aclarar que dichos casos de uso son mucho más que una simple herramienta de captura de requisitos, ya que dirigen el proceso de desarrollo en su totalidad.

Los desarrolladores creen un modelo de análisis que utiliza el modelo de casos e uso como entrada y que es diferente del modelo de diseño porque es un modelo conceptual en lugar de ser un esquema de implementación. Cada caso de uso en el modelo de casos de uso se traduce en una realización de casos de uso en el modelo de análisis.

Tomando los casos de uso uno a uno, los desarrolladores pueden identificar las clases que participan en la realización de aquellos. Así obtenemos un conjunto de clases que juntos realizan los casos de uso.

Durante el análisis y el diseño transformamos el modelo de casos de uso desde un modelo de análisis en un modelo de diseño. El modelo de análisis va creciendo más y más en cada iteración. Por cada iteración se seleccionan determinados casos de uso y se los refleja en el modelo de análisis, se comienzan a construir clases de análisis y relaciones entre ellas

Luego es necesario complementar estos modelos de análisis con un modelo que permita plasmar la forma en que interactúan las distintas clases para llevar a cabo la realización de casos de uso. Aparecen de esta forma los diagramas de colaboración El modelo de diseño

se crea tomando como entrada principal el modelo de análisis y adaptándose al entorno de implementación elegido. Funciona como esquema para la implementación. Con esto queremos indicar que el modelo de diseño es más “Físico” por naturaleza, mientras que el modelo de análisis es más “Conceptual”.

El objetivo de este punto es, no solo probar el correcto funcionamiento del sistema, sino también controlar que el sistema desempeñe correctamente todas aquellas funcionalidades para las que fue creado, es decir que debemos chequear contra los casos de uso. Se utiliza un modelo compuesto por casos de prueba y procedimientos de prueba a partir del cual podemos ir chequeando la funcionalidad plasmada en los casos de uso contra los requerimientos esperados.

1.5.2. PUDS centrado en la Arquitectura

La arquitectura describe los cimientos del sistema que son necesarios como base para comprender, desarrollarlo y producirlo. El objetivo número uno de la fase de elaboración es construir una arquitectura sólida para que a partir de la misma podamos arrancar con la construcción de la totalidad del sistema sin tropiezos posteriores.

Al momento de ir pasando por las distintas fases del proyecto se necesita utilizar la arquitectura como guía general para situarse correctamente en el lugar en donde están parados y usaran los distintos diagramas más detallados para realizar su trabajo. Cuando hablamos de Arquitectura hablamos también del concepto de vista, ya que se generan vistas sobre los distintos modelos que hemos estado viendo y que abarcan solo aquellos casos de uso importantes en este contexto

La descripción de la arquitectura tiene cinco vistas, una para cada modelo:

1. Casos de uso
2. Análisis
3. Diseño
4. Despliegue
5. Implementación

1.5.3. PUDS Iterativo e Incremental

Ya vimos en los 2 puntos anteriores cuan critica es la relación entre casos de uso y arquitectura, unos condicionan lo otro y viceversa. Por eso es imprescindible utilizar un mecanismo que permita ir avanzando poco a poco en el refinamiento de unos como otros, es decir, vamos avanzando por etapas.

Esta es la clave del tercer pilar alrededor del cual gira el Proceso Unificado de desarrollo de software: proceso iterativo e incremental.

El decir, basarnos en un proceso iterativo e incremental nos diferencia mucho con metodologías de desarrollo tradicionales en las cuales el desarrollo también se realiza por etapas, pero las mismas se van realizando en serie, una detrás de la otra sin posibilidad por ejemplo de recibir retroalimentación en alguna y volver para atrás para modificar funcionalidad.

Se cumplen las etapas, se van cerrando y avanza a la siguiente. En nuestro caso la base es que el proceso sea realmente iterativo e incremental, es decir:

- Planificar un poco,
- Especificar, diseñar e implementar un poco
- Integrar, probar y ejecutar un poco cada iteración

Es decir, se divide el proyecto en un número de mini-proyectos, siendo cada uno de ellos una iteración. Cada iteración está compuesta por todas las fases de un proyecto de software, es decir que cada una de ellas sería como un mini-proyecto hecho con una metodología tradicional, como la de cascada.

Para Resumir, los principales objetivos de un desarrollo iterativo e incremental son:

- Atenuar los riesgos.
- Obtener una arquitectura robusta
- Gestionar requisitos cambiantes
- Permitir cambios tácticos.
- Conseguir una integración continua
- Conseguir un aprendizaje temprano

Para finalizar, el término de ciclo de vida intenta captar la idea de que un producto de software es el resultado de un proceso de desarrollo unificado que puede dividirse en fases. Podemos definir el ciclo de vida como la actividad de producir y mantener sistemas informáticos.

Su función es:

Establecer el orden en el que se

- Especifica
- Realiza un prototipo
- Diseña
- Implementa
- Prueba

Estableciendo criterios para pasar de una fase a otra.

1.6. Elemento de Implementación.

Para realizar cualquier tipo de desarrollo de software debemos tener en cuenta los puntos principales para el Análisis, la gestión, almacenamiento y explotación de toda esta información, tomando en cuenta las herramientas principales para su implementación:

- UML
- Base de Datos
- Sistemas de Gestión de Base de Datos
- Lenguajes de Manipulación y Definición de Datos
- Suite de Desarrollo Web

En resumen las Tecnologías de Información contribuyen a automatizar el Sistema de Información de la empresa mediante estrategias de Ingeniería de Software y mediante herramientas que nos permiten implementar de forma física lo que surge de cualquier idea, necesidad o requerimiento de un usuario de forma conceptual.



Capítulo 2

ANÁLISIS Y DISEÑO DEL SISTEMA

2. ANALISIS Y DISEÑO DEL SISTEMA.

Al conjunto o disposición de procedimientos o programas relacionados de manera que juntos forman una sola unidad es llamado "Análisis de Sistemas de Información", esto es; un conjunto de hechos, principios y reglas clasificadas y dispuestas de manera ordenada mostrando un plan lógico en la unión de las partes conceptuales y de diseño.

2.1. UML: Lenguaje Unificado de Modelado

El Lenguaje de Modelamiento Unificado (UML - Unified Modeling Language) es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software. UML entrega una forma de modelar cosas conceptuales como lo son procesos de negocio y funciones de sistema, además de cosas concretas como lo son escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de software reusables.

2.1.1. Historia

Durante los ochenta y principios de los noventa Grady Booch, James Rumbaugh, e Ivar Jacobson trabajaban por separado en desarrollo de notaciones para el análisis y diseño de sistemas orientados a objetos. Los tres llegaron por separado a obtener bastante reconocimiento.

Booch había escrito "Object-Oriented Analysis and Design with Applications" un libro de referencia en el análisis y diseño orientado a objetos desarrollando su propia notación.

Por su parte James Rumbaugh había desarrollado su propia notación de diseño orientado a objetos llamada OMT (Object Modeling Technique) en su libro "Object-Oriented Modeling and Design".

Por otro lado Jacobson se había revelado como un visionario del análisis (padre de los casos de uso) y sobre todo del diseño orientado a objetos, sorprendiendo a todo el mundo en "Object-Oriented Software Engineering: A Use Case Driven Approach".

A mediados de los noventa empezaron a intercambiar documentos y trabajar en conjunto produciendo grandes avances en el modelado de sistemas orientados a objetos.

En 1994 Rational contrató a Rumbaugh en donde ya trabajaba Booch, un año después Jacobson se unía a ellos en Rational.

En 1997 salió a la luz la versión 1.0 de UML.

Es importante remarcar que UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo.

2.1.2. PUDS emplea UML

UML (Unified Modeling Language) es un lenguaje estándar que se utiliza para realizar modelos de software. Se usa para visualizar, especifica, construir y documentar sistemas. UML es un lenguaje que brinda un vocabulario particular y reglas específicas que se focalizan en la representación conceptual y física el sistema. Este vocabulario y estas reglas dicen cómo crear y leer modelos, pero no dice que modelos se deben crear ni cuándo. Esta tarea le corresponde al Proceso Unificado de Desarrollo.

Un proceso bien definido nos dirá que artefactos producir, que actividades realizar y que personas (trabajadores) lo crean, usan y administran, y como emplear esto artefactos para medir y controlar el proyecto como un todo.

UML es:

- **Lenguaje de visualización:** Facilita la comunicación de los modelos conceptuales a otros y disminuye errores al permitir hablar el mismo lenguaje. Permite interpretar los modelos sin ambigüedades, ya que UML tiene una semántica bien definida.
- **Lenguaje de especificación:** Especificación significa construir modelos precisos, no ambiguos y completos.

UML especifica las decisiones del análisis y diseño que deben llevarse a cabo, como así también las de la implementación

- **Lenguajes de construcción:** No es un lenguaje de programación, pero sus modelos pueden ser conectados en una variedad de lenguajes de programación como Java, C++, Visual Basic. Se puede generar código a partir de un modelo UML y aun hacer ingeniería reversa (se puede construir un modelo a partir de una implementación). UML permite ejecución de modelos y simulación de sistemas.
- **Lenguaje de documentación:** Documenta requerimientos, arquitectura, diseño, codificación, plan de proyectos, pruebas, prototipo, versiones de software.

2.2. Modelos UML

Los modelos son simplificaciones de la realidad. Construimos modelos para comprender mejor los sistemas, porque debido a la complejidad de los mismos no es posible abordarlos en su totalidad tal como son en la realidad. A través del modelado se logran los siguientes objetivos:

- Ayudar a visualizar como es el sistema
- Permitir especificar la estructura del sistema
- Especificar el comportamiento del sistema
- Proporcionar plantillas que guían la construcción del sistema
- Documentar las decisiones adoptadas

Un modelo tiene dos aspectos esenciales:

- Semántica: Se trata de la información o significado del modelo
- Presentación Visual: es la notación que muestra el modelo de una forma comprensible. Organiza la presentación del modelo

2.2.1. Diagramas:

Volviendo a la definición de UML como "conjunto de herramientas", si nos imaginamos UML como una caja de herramientas con su martillo, destornillador, alicates, etc. Veamos qué contiene nuestra caja de herramientas:

- Diagrama de Casos de uso
- Diagrama de clases
- Diagrama de estados
- Diagrama de secuencias
- Diagrama de actividades
- Diagrama de colaboraciones

2.3. Análisis del Sistema de Tesina.

Para el análisis de nuestro sistema nos basaremos en el conjunto de herramientas de las metodologías UML. Siguiendo con la analogía, si vamos a colgar un cuadro no usaremos todas las herramientas de nuestra caja, posiblemente sólo usemos el martillo para clavar el clavo.

Lo mismo pasa con UML, una vez que conozcamos las herramientas usaremos en cada momento las más adecuadas a nuestras necesidades. No quiero decir que esto sea fácil, pues hay que saber para qué sirven y qué limitaciones tienen unas y otras para conocer su utilidad. Pero se puede alcanzar este conocimiento con un poco de práctica y sentido común.

Como lo hemos visto UML no es un método de desarrollo, o te va a decir cómo pasar del análisis al diseño y de este al código. No son una serie de pasos que te llevan a producir código a partir de unas especificaciones.

UML al no ser un método de desarrollo es independiente del ciclo de desarrollo que vayas a seguir, puede encajar en un tradicional ciclo en cascada, o en un evolutivo ciclo en espiral o incluso en los métodos ágiles de desarrollo.

2.4. Planeación del Proyecto de Software

2.4.1. Definición del Problema:

- a) Definición: Se requiere un sistema para gestionar cada uno de los proyectos que los investigadores mantienen para tener una administración de los mismos llevando a cabo el avance y documentación que cada uno de ellos puede desencadenar, teniendo un control de posibles documentos anexos al proyecto como pueden ser tesis, tesinas, etc. generando estadísticas que el administrador pueda recopilar para la toma de decisiones futuras que a los intereses de la institución educativa le competan.
- b) Especificaciones: Se requiere mínimamente una computadora la cual tenga conexión a internet y mediante un browser podamos acceder al sistema, además de una computadora central de donde tengamos nuestro proyecto lo cual mínimamente requeriremos no más de 2Gb de espacio en Disco duro, 1 MB de RAM e igualmente conexión a internet para que mediante solicitudes http o https podamos acceder a los servicios. Por ultimo requeriremos una computadora para soportar una BD en la cual mantendremos nuestra información de cada uno de los proyectos para ser accedidas en cualquier momento.

2.4.2. Desarrollo de estrategias de Solución:

- a) Estrategias: El sistema se podría realizarse mediante un sistema Web, con una BD, conllevando con ello mantener instalado el software en cada una de las maquinas. Otra estrategia es realizarse mediante una estrategia de 3 capas, en la cual podremos desarrollar el proyecto en la capa intermedia sin tener que reinstalar el producto en cada máquina si hubiese alguna modificación.
- b) Factibilidad: Esta factibilidad del sistema fue basada en que este sistema de proyectos es un subsistema complementario de otro de mayor envergadura, por lo cual se adaptó a las características del mismo ya desarrollado.

2.4.3. Adecuación del Producto:

- a) Ciclo de Vida: Se define un modelo de Ciclo de vida de nuestro software es el modelo Espiral el cual vamos a ir desarrollando prototipo con aspectos controlados y sistemáticos del viejo modelo de cascada pero de una forma circular y cíclica.

b) Herramientas por Etapa: Cada fase tendremos los siguientes elementos:

- **Análisis de Requerimientos**: Editor de Textos Word Office.
- **Diseño Arquitectónico y Estructural**: Visual Paradigm con modelado de Diagramas UML mediante metodología PUDS como son:
 - Diagramas de Uso
 - Diagramas de Transición de Estados
 - Diagrama de Transición de Secuencias Normal y Alternativa.
- **Codificación, Desarrollo e Implementación**: Las herramientas planeadas en las capas son todas basadas en Tecnología Oracle como Base de Datos Oracle Estándar en la cual podremos tener nuestra información de forma Normalizada ya que su manejador es Relacional y además soporta terminología Orientada a Objetos, un Servidor tipo Internet Information Server donde Mantendremos nuestro proyecto junto con otro de mayor jerarquía que ya mantiene la seguridad y desarrollo en HTML.

2.5. Análisis de Requerimientos

2.5.1. Casos de uso UML:

- a. A continuación se listan lo Actores que se tendrán dentro de nuestro Sistema
 - i. **Alumno**: Usuario que mantiene sus datos personales actualizados de cada Alumno.
 - ii. **Profesor**: Usuario que mantiene sus datos personales y educativos como profesor.
 - iii. **Administrador**: Usuario que da seguimiento a los proyectos (asigna al investigador (es) los proyectos, etc.) proyecto, así como visualiza el cálculo de las Estadísticas.
- b. También tenemos los casos de uso generales.
 - i. **Mantenimiento del Datos de Alumnos**: Alta, baja, modificación de datos personales del Alumno ingresado, ingreso de Fotografía, actualización de Clave de Acceso al Sistema (password), Preinscripción de Materias, ingreso de kardex actual.
 - ii. **Mantenimiento de Datos del Profesor**: Alta, baja, modificación de datos personales del Profesor ingresado, ingreso de fotografía, actualización de claves de Acceso al sistema (password), revisión de kardex de cualquier Alumno.

- iii. **Mantenimiento de Datos del Proyecto**: Consulta de datos personales de alumnos, profesores, así como alta de profesores externos, alta de proyectos, guardado de Tesis finales, ingreso de alumnos titulados.

c. El Diagrama general de casos de uso sería de la siguiente manera:

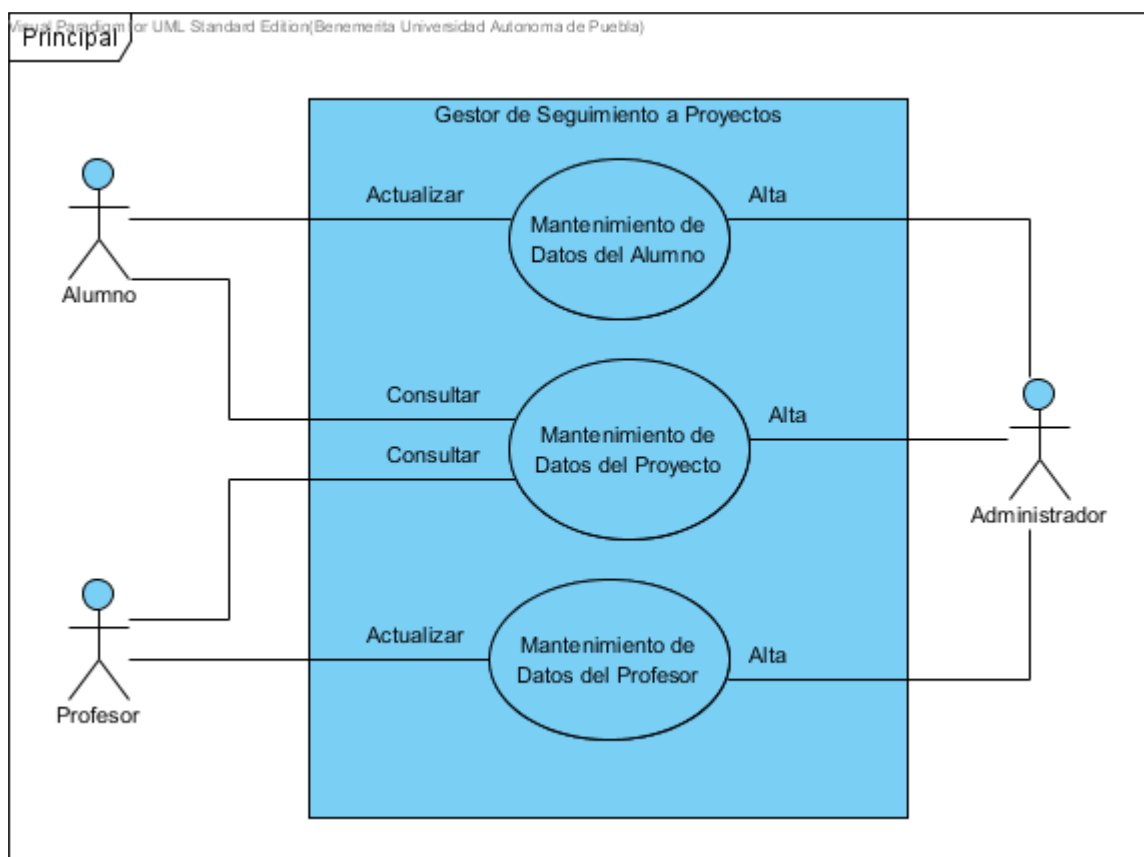


Fig. 2.1 Diagrama principal de "Casos de Uso" (Nivel de abstracción 0)

Desglosando el Mantenimiento de datos del Alumno a un nivel de abstracción mayor podríamos tener los siguientes casos de uso:

- i. **Alta de datos personales del Alumno**: El administrador ingresa los datos de los alumnos que tienen acceso al sistema (matricula, Apellidos, Nombres, teléfonos, domicilio, etc.) y el alumno puede actualizar sus datos una vez que ingreso al sistema.
- ii. **Ingreso de fotografía del Alumno**: Cada Alumno puede ingresar el Archivo perteneciente a la fotografía del Alumno, donde los administradores podrán visualizarlo junto con los datos personales del alumno.
- iii. **Actualización de Password**. Cada alumno puede cambiar su password de ingreso al sistema gestor de proyectos

- iv. **Pre-Inscripción de Materias**: El Alumno puede capturar las materias que ha sido pre-inscrito para ser tomadas en cuenta al momento de vincularle algún proyecto, ya que el proyecto puede tener ciertas áreas de interés.
- v. **Generación de Kardex**: Visualización de un reporte de kardex del alumno con el cual se muestra un desglose de las materias cursadas.

El diagrama de estos casos de uso seria el siguiente

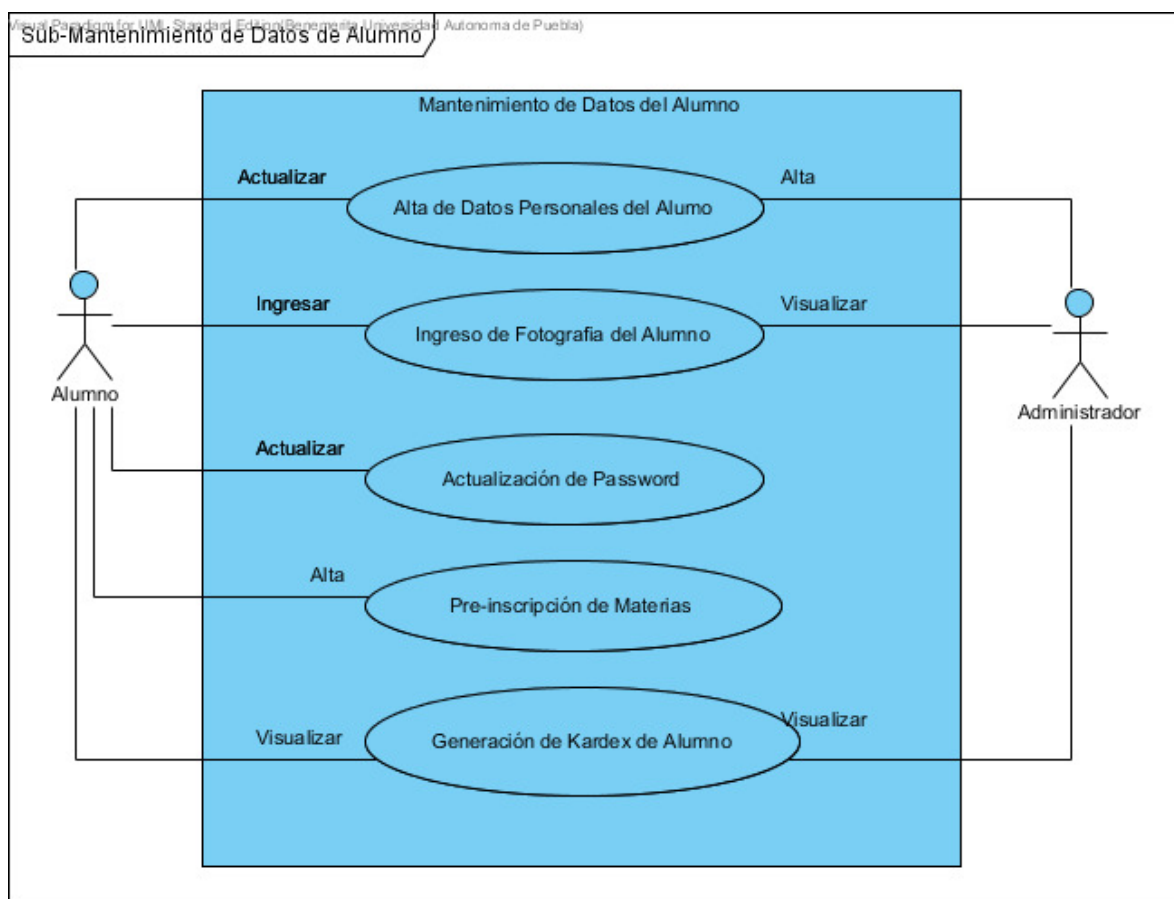


Fig. 2.2 Diagrama de Mantenimiento de datos del Alumno (Nivel de abstracción 1)

El caso de uso de Mantenimiento de datos del Profesor a un nivel de abstracción mayor podríamos tener los siguientes casos de uso:

- i. **Alta de datos personales de Profesores**: El administrador ingresa los datos del profesor que tienen acceso al sistema (Clave del Trabajador, Apellidos, Nombres, teléfonos, correo electrónico, domicilio, etc.) y el profesor puede actualizar sus datos una vez que ingreso al sistema.

- ii. **Ingreso de fotografía del Profesor**: Cada Profesor puede ingresar el Archivo perteneciente a la fotografía del mismo, donde los administradores podrán visualizarlo junto con los datos personales del profesor.
- iii. **Actualización de Password**: Cada profesor puede cambiar su password de ingreso al sistema gestor de proyectos

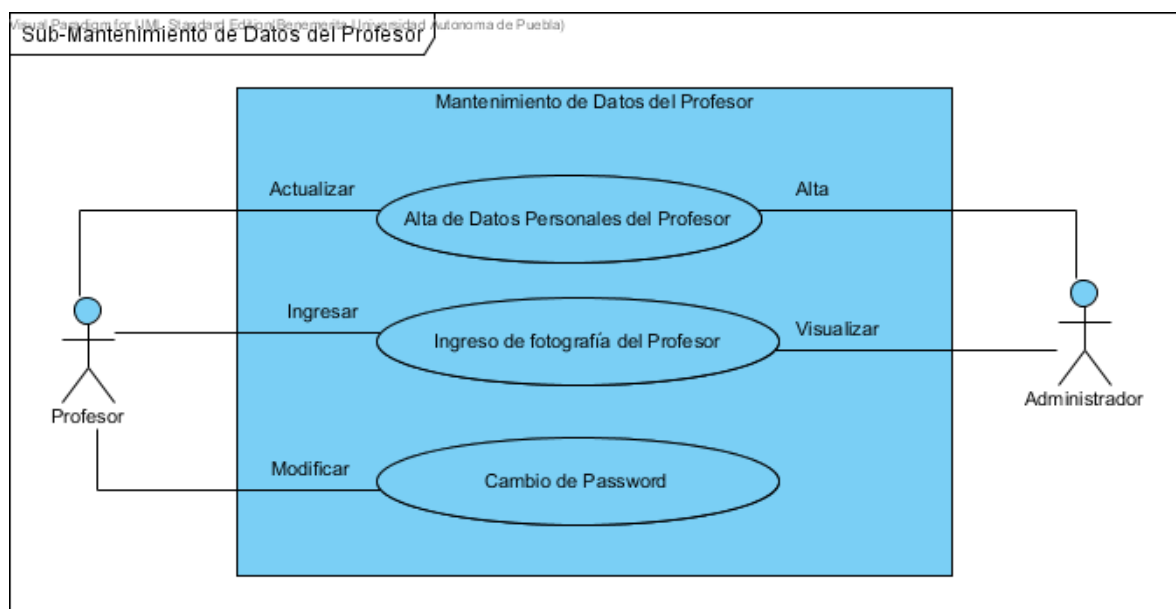


Fig. 2.3 Diagrama de Mantenimiento de datos del Profesores (Nivel de abstracción 1)

El caso de uso de Mantenimiento de datos del Proyecto a un nivel de abstracción mayor podríamos tener los siguientes casos de uso:

- i. **Alta de datos del Proyecto**: El administrador da de alta los datos del proyecto como pueden ser Título, patrocinadores, objetivos, Institución responsable, Actividades a realizar, tipo de proyecto, área académica, etc.
- ii. **Ingreso de Documentos**: El administrador ingresa los documentos que fuesen generados por el proyecto como es Tesis o Tesina.
- iii. **Captura de Profesores externos**: Cada profesor puede ser externo el cual puede haber dado de alta el proyecto por los intereses que a su institución le beneficiasen.
- iv. **Generación de Estadísticas**: Visualización del reporte de estadísticas generales de los proyectos.

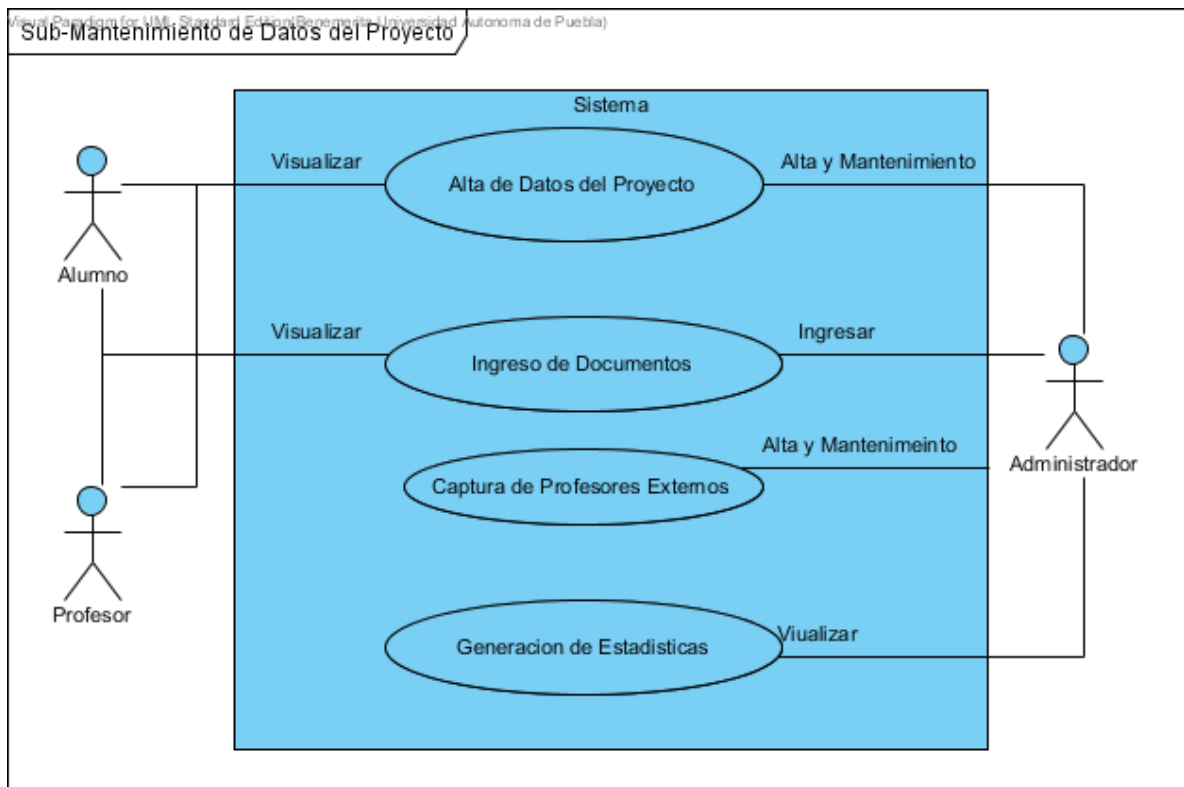


Fig. 2.4 Diagrama de Mantenimiento de datos del Proyectos (Nivel de abstracción 1)

La Secuencia de **Alta del Alumnos**:

- El administrador entra al menú general del sistema.
- Elige ingresar al sistema con su clave de administrador.
- Captura su clave y su password
- El sistema despliega el menú del Administrador.
- El administrador elige la opción de Alumnos
- El sistema despliega campo para capturar matricula de alumno ya existente o un botón de Nuevo Registro del alumno
- El administrador del sistema selecciona Nuevo Registro.
- El sistema captura los Datos personales del Alumno, así como un password inicial y elige la opción de enviar.
- El sistema ingresa los datos en el contenedor de datos.
- El catálogo regresa un mensaje de ingreso exitoso de los Datos del Alumno.
- El contenedor nos retorna hacia el menú principal del administrador.

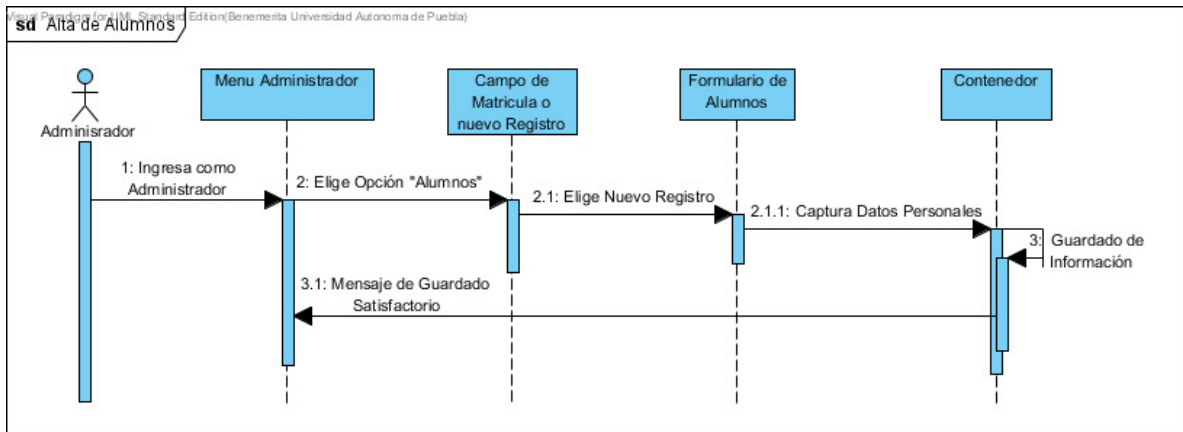


Fig. 2.5 Secuencia de Alta de Alumnos

Excepciones de Alta de Alumnos:

Mal ingreso por Usuario Administrador:

- El administrador entra al menú principal del sistema,
- Elige ingresar al sistema con su clave de administrador
- **El usuario y password es incorrecto.**
- El sistema retorna el mensaje de “Usuario o password Incorrecto, reintente

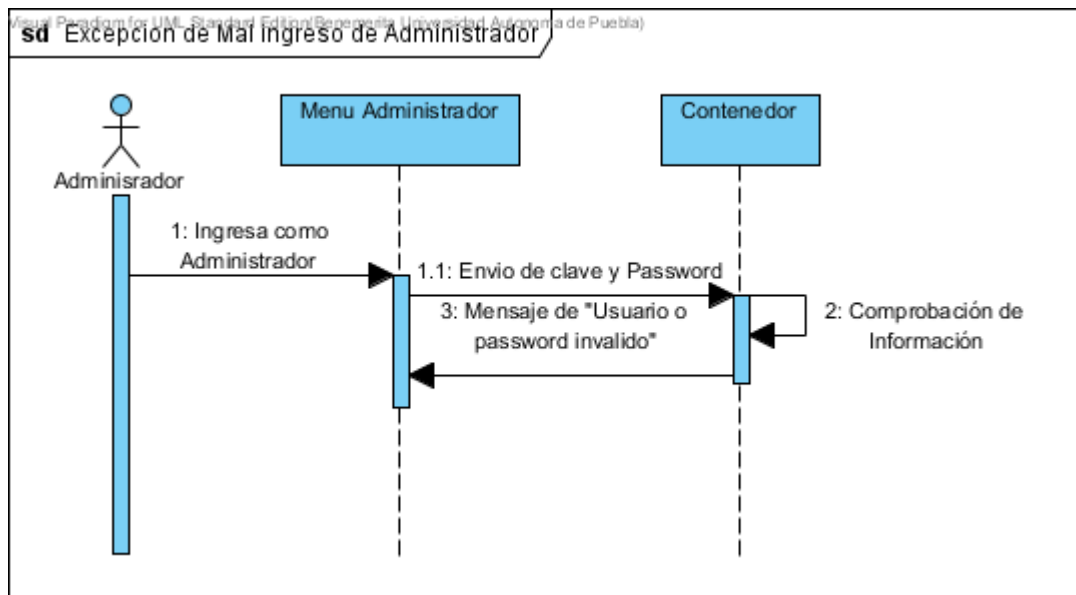


Fig. 2.6 Excepción de Mal ingreso del Administrador al Sistema

Falta de Información requerida del Alumno:

- El administrador entra al menú principal del sistema.
- Elige ingresar al sistema con su clave de administrador.
- Captura su clave y su password

- El sistema despliega el menú de Administrador.
- El administrador elige la opción de Alumnos
- El sistema despliega campo para capturar matricula de alumno ya existente o un botón de Nuevo Registro del alumno
- El administrador del sistema selecciona Nuevo Registro.
- El sistema captura los Datos personales del Alumno, así como un password inicial y elige la opción de enviar.
- El sistema ingresa los datos en el contenedor de datos.
- El sistema informa que faltan datos requeridos.
- El usuario ingresa los datos faltantes
- El catálogo regresa un mensaje de ingreso exitoso de los Datos del Alumno.
- El contenedor nos retorna hacia el menú principal del administrador.

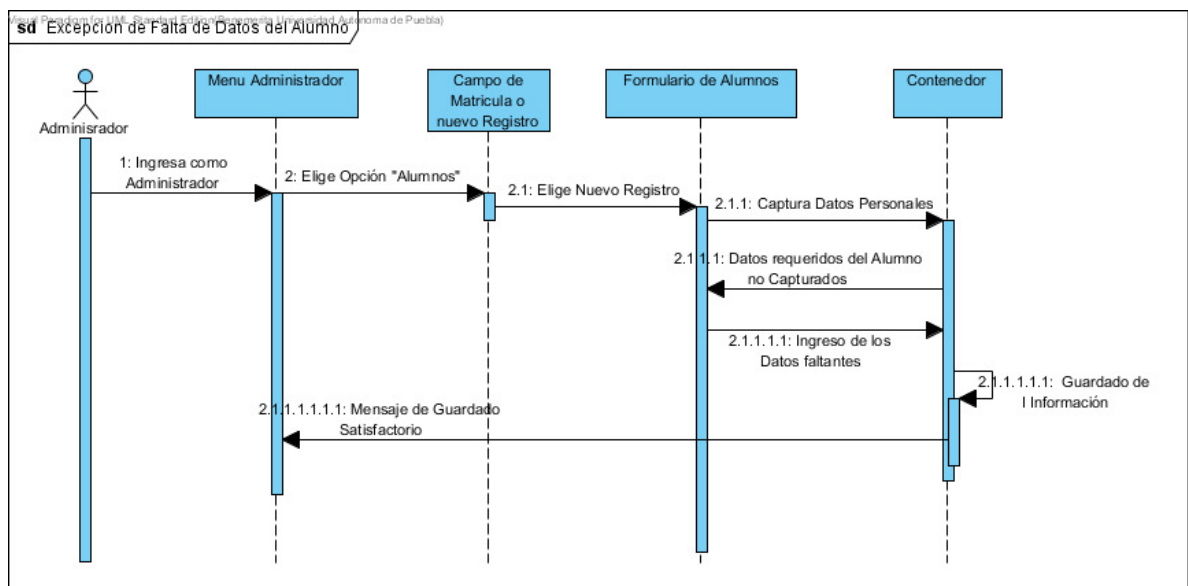


Fig. 2.7 Excepción de Falta de Datos requeridos del alumno

La Secuencia de **Alta del Profesores**:

- El administrador entra al menú general del sistema.
- Elige ingresar al sistema con su clave de administrador.
- Captura su clave y su password
- El sistema despliega el menú de Administrador.
- El administrador elige la opción de Profesores.
- El sistema despliega campo para capturar la clave del profesor ya existente o un botón de Nuevo Registro del profesor
- El administrador del sistema selecciona Nuevo Registro.

- El sistema captura los Datos personales del profesor, así como un password inicial y elige la opción de enviar.
- El sistema ingresa los datos en el contenedor de datos.
- El catálogo regresa un mensaje de ingreso exitoso de los Datos del Profesor.
- El contenedor nos retorna hacia el menú principal del administrador.

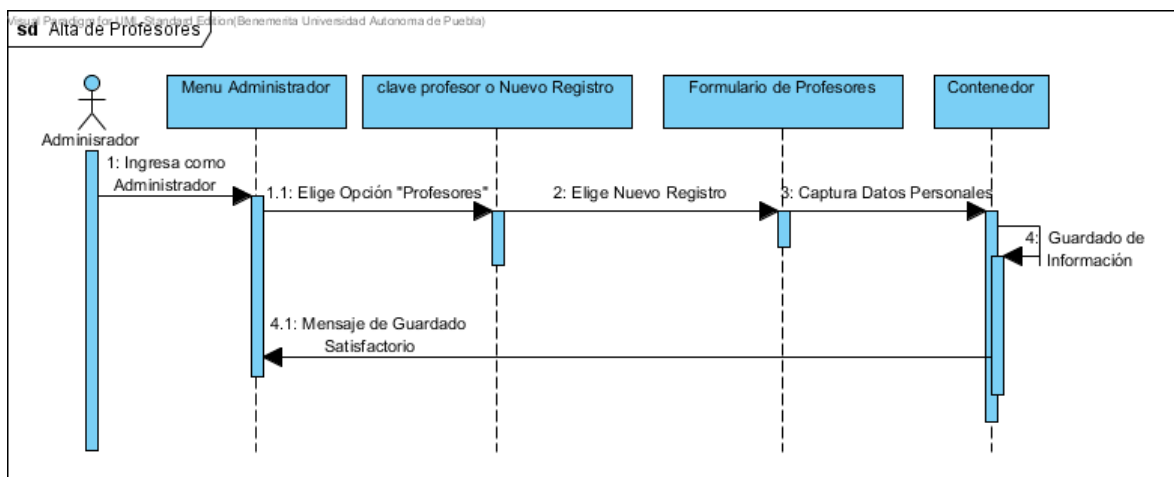


Fig. 2.8 Secuencia de Alta de Profesores

Excepciones de Alta de Profesores:

Falta de Información requerida del Profesor:

- El administrador entra al menú general del sistema.
- Elige ingresar al sistema con su clave de administrador.
- Captura su clave y su password
- El sistema despliega el menú de Administrador.
- El administrador elige la opción de Profesores.
- El sistema despliega campo para capturar la clave del profesor ya existente o un botón de Nuevo Registro del profesor
- El administrador del sistema selecciona Nuevo Registro.
- El sistema captura los Datos personales del profesor, así como un password inicial y elige la opción de enviar.
- El sistema ingresa los datos en el contenedor de datos.
- El sistema informa que faltan datos requeridos.
- El usuario ingresa los datos faltantes
- El catálogo regresa un mensaje de ingreso exitoso de los Datos del Profesor.
- El contenedor nos retorna hacia el menú principal del administrador.

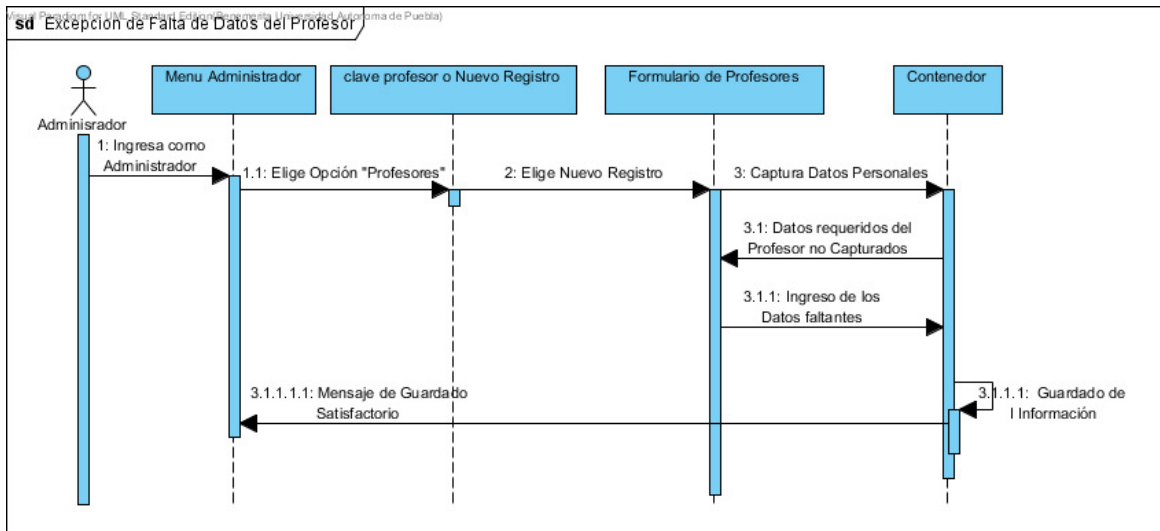


Fig. 2.9 Excepción de Falta de Datos requeridos del Profesor

La Secuencia de **Alta del Proyectos**:

- El administrador entra al menú general del sistema.
- Elige ingresar al sistema con su clave de administrador.
- Captura su clave y su password
- El sistema despliega el menú de Administrador.
- El administrador elige la opción de Proyectos.
- El sistema despliega listado de Proyectos y un botón de Registro de Proyecto
- El administrador del sistema selecciona Registro del proyecto.
- El sistema captura los Datos del proyecto y elige la opción de enviar.
- El sistema ingresa los datos en el contenedor de datos.
- El catálogo regresa un mensaje de ingreso exitoso de los Datos del Proyecto.
- El contenedor nos retorna hacia el menú principal del administrador.

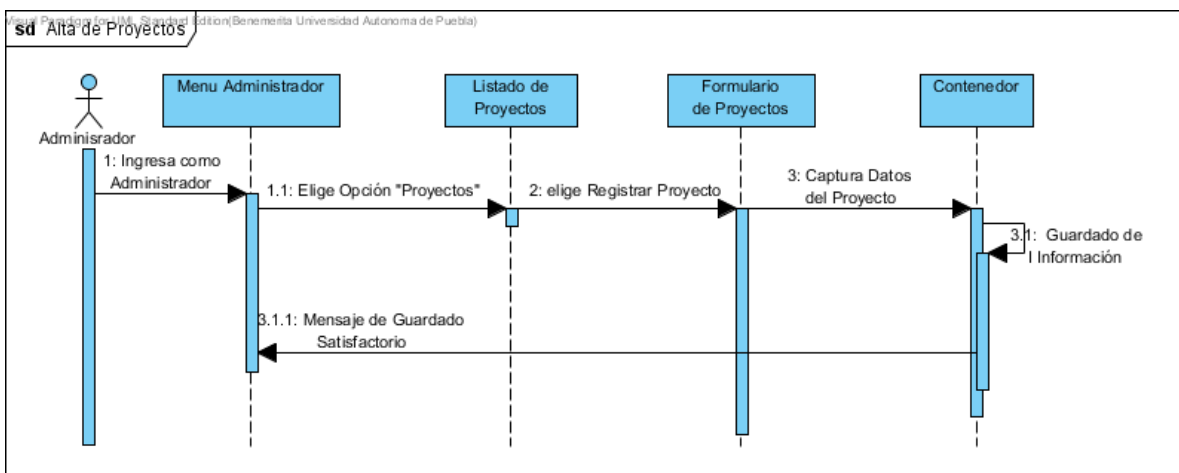


Fig. 2.10 Secuencia de Alta de Proyectos

Excepciones de **Alta de Proyectos**:

Falta de Información requerida del Proyecto:

- El administrador entra al menú general del sistema.
- Elige ingresar al sistema con su clave de administrador.
- Captura su clave y su password
- El sistema despliega el menú de Administrador.
- El administrador elige la opción de Proyectos.
- El sistema despliega listado de Proyectos y un botón de Registro de Proyecto
- El administrador del sistema selecciona Registro del proyecto.
- El sistema captura los Datos del proyecto y elige la opción de enviar.
- El sistema ingresa los datos en el contenedor de datos.
- El sistema informa que faltan datos requeridos.
- El usuario ingresa los datos faltantes
- El catálogo regresa un mensaje de ingreso exitoso de los Datos del Proyecto.
- El contenedor nos retorna hacia el menú principal del administrador.

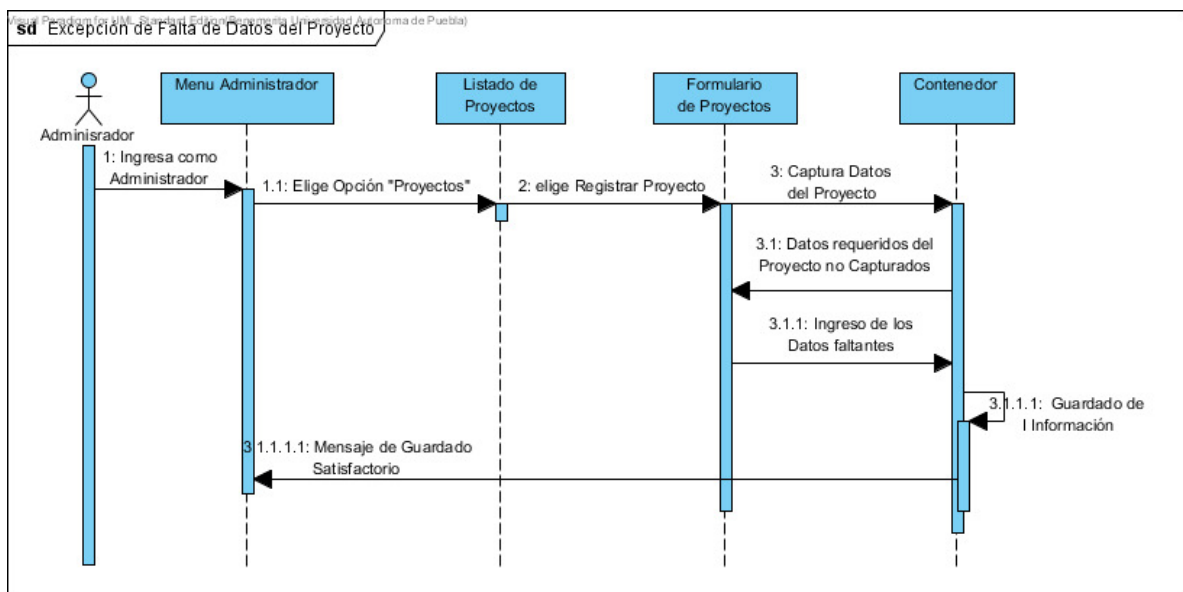
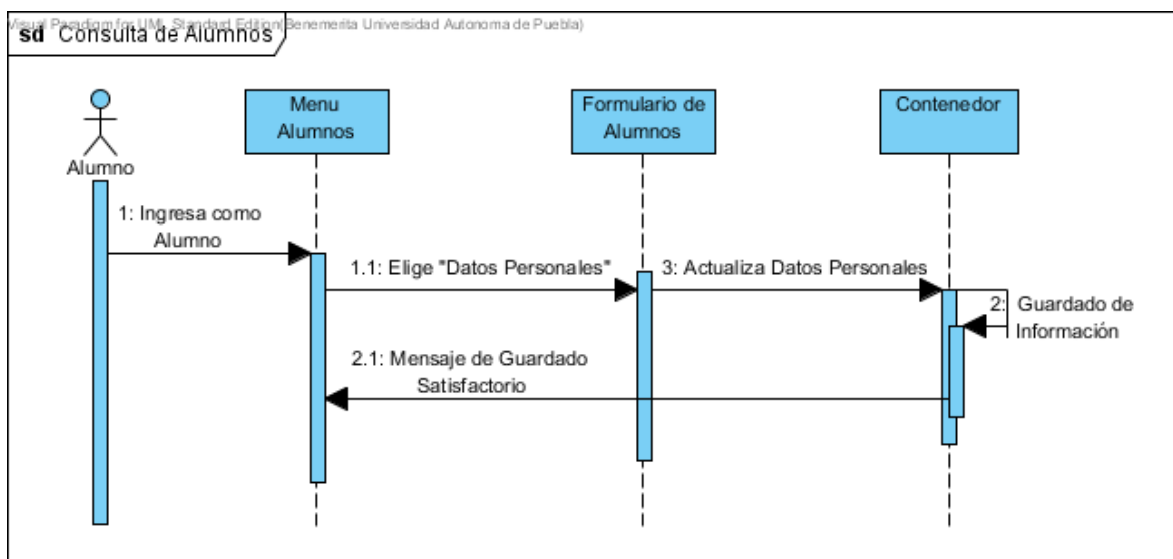


Fig. 2.11 Excepción de Falta de Datos requeridos del Proyecto

La Secuencia de **Consulta de datos del Alumnos**:

- El alumno entra al menú general del sistema.
- Elige ingresar al sistema con su matrícula.
- Captura su matrícula y su password
- El sistema despliega el menú del Alumnos.

- El alumno elige la opción de Datos personales
- El sistema despliega el formulario con los datos personales del alumno
- El alumno actualiza su información.
- El Alumno elige el botón de Enviar para guardar la información
- El sistema ingresa los datos en el contenedor de datos.
- El catálogo regresa un mensaje de actualización exitosa de los Datos del Alumno.
- El contenedor nos retorna hacia el menú principal del Alumno.



Nota: Las excepciones son las mismas que la Alta de datos del Alumno

La Secuencia de **Consulta de datos del Profesor:**

- El profesor entra al menú general del sistema.
- Elige ingresar al sistema con su clave de profesor.
- El sistema ingresa el pasaporte para verificar su clave
- El profesor captura su clave y su password
- El sistema despliega el menú del Profesores.
- El Profesor elige la opción de Datos personales
- El sistema despliega el formulario con los datos personales del Profesor
- El Profesor actualiza su información.
- El Profesor elige el botón de Enviar para guardar la información
- El sistema ingresa los datos en el contenedor de datos.
- El catálogo regresa un mensaje de actualización exitosa de los Datos del Profesor.
- El contenedor nos retorna hacia el menú principal del Profesor.

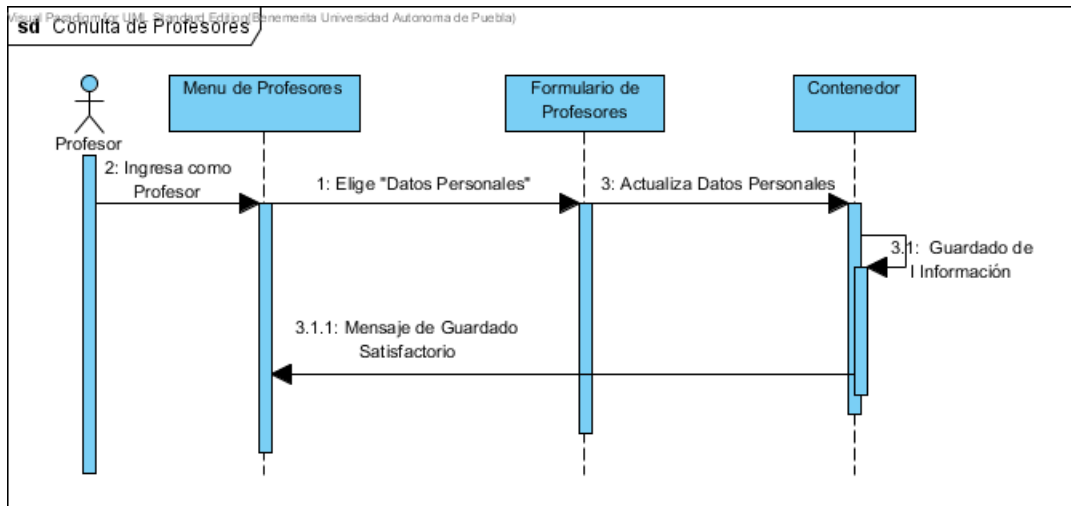


Fig. 2.13 Secuencia de Consulta de Profesores

Nota: Las excepciones son las mismas que la Alta de datos del Profesor

La Secuencia de **Cambio de Password**:

- El alumno entra al menú general del sistema.
- Elige ingresar al sistema con su clave de Alumno.
- Captura su clave y su password actual
- El sistema despliega el menú del Alumnos.
- El administrador elige la opción “Cambio de Clave”
- El sistema despliega la página de cambio de password
- El alumno captura el nuevo password y su confirmación y elige la opción de enviar.
- El sistema ingresa los datos en el contenedor de datos.
- El catálogo regresa un mensaje de cambio exitoso del Alumno.
- El contenedor nos retorna hacia el menú principal del Alumno.

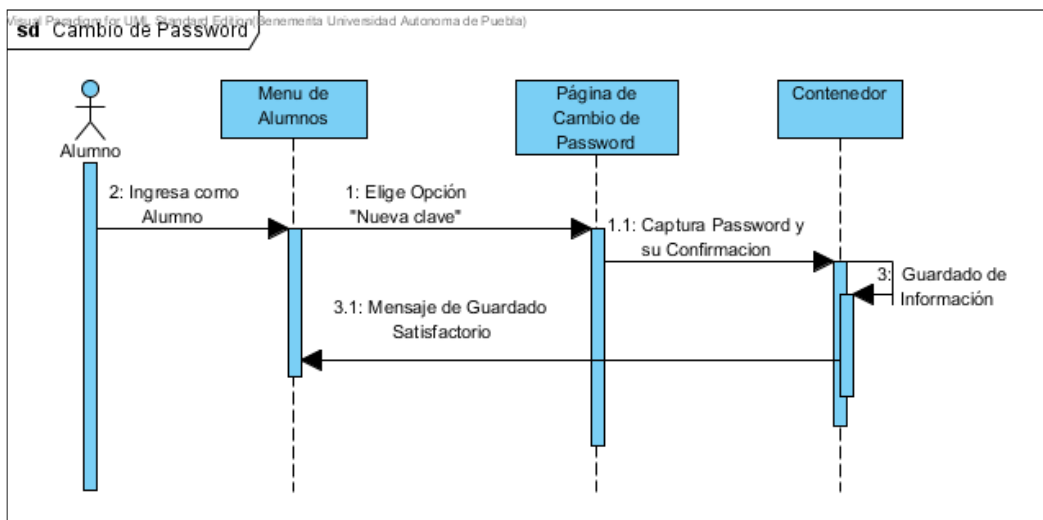


Fig. 2.14 Secuencia de Cambio de Password

La Secuencia de **Ingreso de documentos (tesis)**:

- El administrador entra al menú general del sistema.
- Elige ingresar al sistema con su clave de Administrador.
- Captura su clave y su password actual
- El sistema despliega el menú del Administrador.
- El administrador elige la opción “tesis”
- El sistema despliega la página con un listado de proyectos
- El administrador elige el campo “sube archivo” del proyecto seleccionado.
- El sistema despliega la página para dar la información del nombre y ruta del archivo
- El usuario captura los datos del archivo y elige el botón Upload.
- El sistema ingresa los archivos en el contenedor de datos.
- El catálogo regresa un mensaje de archivo subido exitosamente.
- El contenedor nos retorna hacia el menú principal del Administrador.

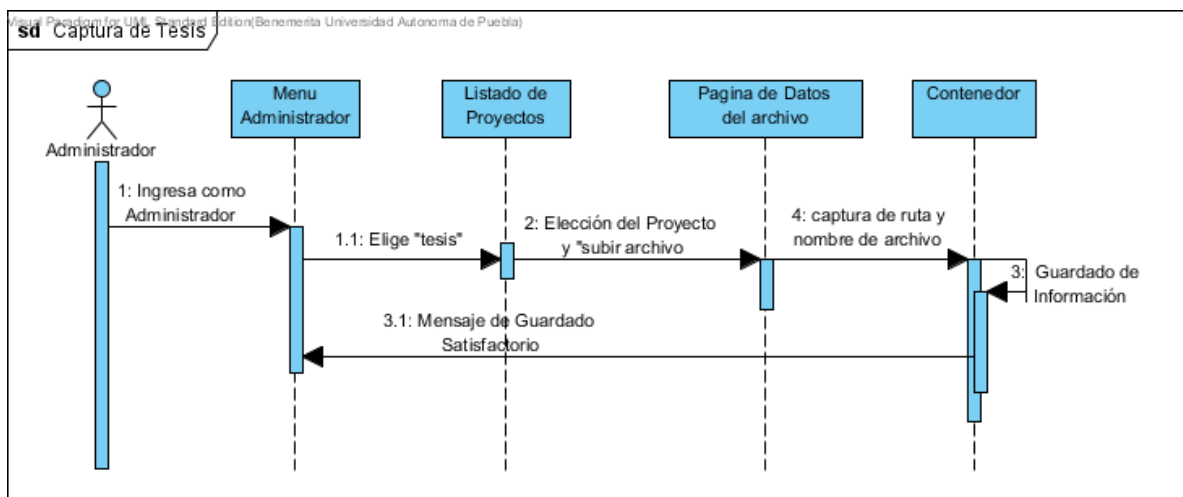


Fig. 2.15 Secuencia de Captura de Tesis

2.5.2. Diseño Arquitectónico o Estructural:

Diagrama de Clases UML

a) Identificando las clases u objetos tendremos:

- Administrador: Esta clase es la que inicia todos los demás sucesos, ya que sus objetos registrarán uno o más proyectos, da de alta profesores y Alumnos para poderles vincular un proyecto de investigación.

- Proyecto: Esta clase solo puede tener asociado un profesor como asesor y un alumno como investigador.
- Alumno: Esta clase es a la cual se le asignara un proyecto de investigación.
- Profesor: Esta clase es por lo cual se le asigna para ser responsable de asesorar al alumno con el proyecto de investigación.
- Tesis: Es parte del material de documentación que se puede anexar a un Proyecto

b) El diagrama de clases quedaría de la siguiente manera:

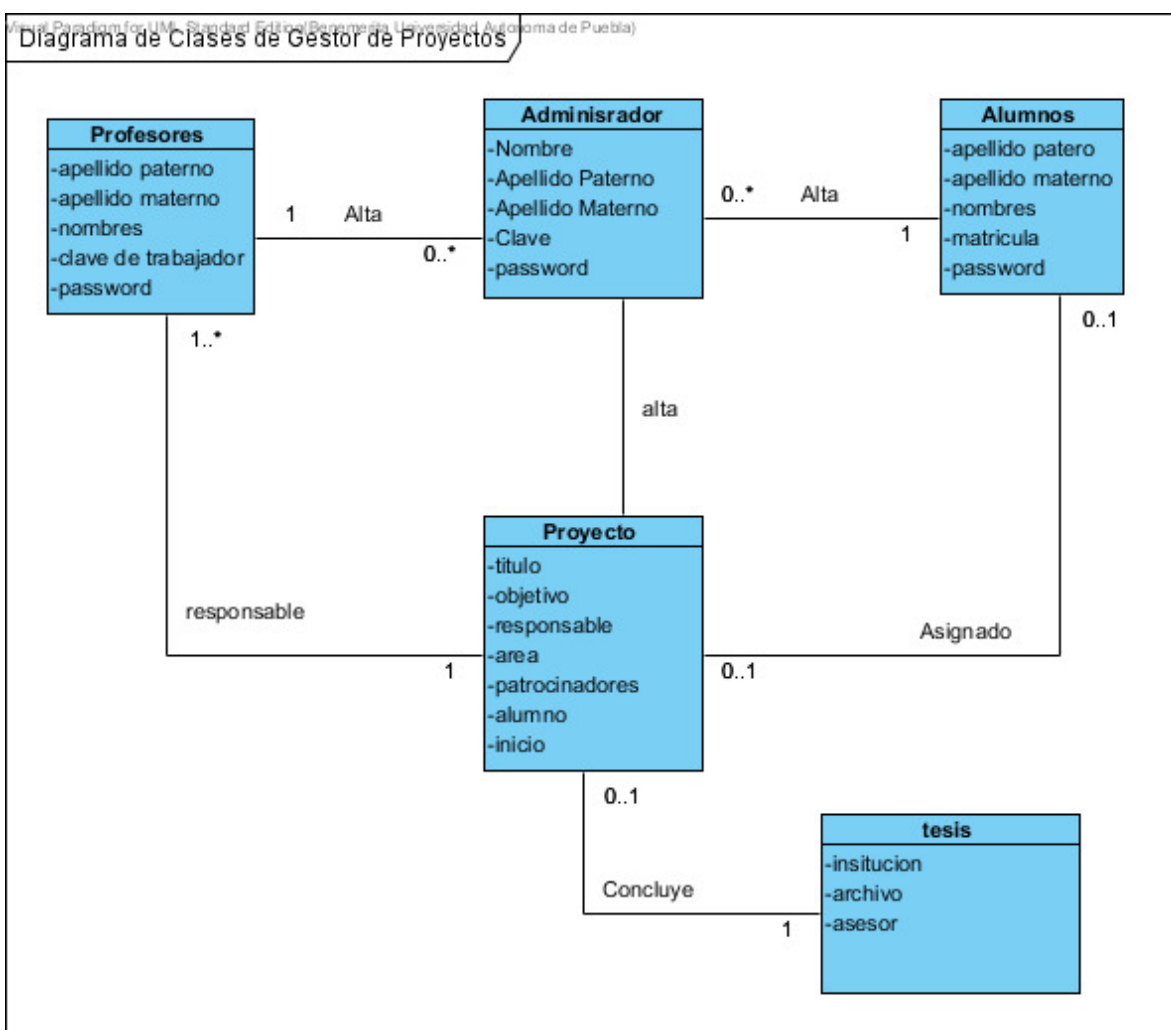


Fig. 2.16 Diagrama de Clases



Capítulo 3

DISEÑO DE BASE DE DATOS



3. DISEÑO DE BASE DE DATOS.

3.1. Definición de Base de Datos

El término “Base de Datos” aparece por primera vez al comienzo de los años setenta, para expresar un conjunto de datos relacionados entre sí, que están estructurados de forma que puede accederse a ellos automáticamente e independientemente de los programas que gestionan esos datos. Esta independencia se refiere a la posibilidad de modificar la estructura de los datos sin necesidad de modificar los programas que los manipulan, evitando con ello los problemas de actualización de datos previamente existentes.

Una base o banco de datos es un conjunto de datos que pertenecen al mismo contexto 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 gracias al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos tienen formato electrónico, que ofrece un amplio rango de soluciones al problema de almacenar datos.

En informática existen los sistemas gestores de bases de datos (SGBD), que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Las propiedades de los sistemas gestores de bases de datos se estudian en informática.

La base de datos más popular en desarrollo web es MySQL, seguida por Oracle, SQL Server y PostgreSQL, también puede usarse perfectamente Firebird o HSQL.

3.2. Sistemas de Gestión de Base de Datos

Los Sistemas de gestión de base de datos son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. En los textos que tratan este tema, o temas relacionados, se mencionan los términos SGBD y DBMS, siendo ambos equivalentes, y acrónimos, respectivamente, de Sistema Gestor de Bases de Datos y Database Management System, su expresión inglesa.

El propósito general de los sistemas de gestión de base de datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos.

SGBD libres

- PostgreSQL (<http://www.postgresql.org> Postgresql) Licencia BSD
- MySQL Licencia Dual, depende el uso.
- Firebird basada en la versión 6 de Interbase, Initial Developer's PUBLIC LICENSE Versión 1.0.
- SQLite (<http://www.sqlite.org> SQLite) Licencia Dominio Público
- Sybase ASE Express Edition para Linux (Edición gratuita para Linux)
- Microsoft SQL Server Compact Edition

SGBD comerciales

- dBase
- FileMaker
- Fox Pro
- IBM DB2 Universal Database (DB2 UDB)
- IBM Informix
- MAGIC
- Microsoft SQL Server
- Open Access
- Oracle
- Paradox
- PervasiveSQL
- Progress (DBMS)
- Sybase ASE
- Sybase ASA
- Sybase IQ

3.3. Sistemas Manejadores de Base de Datos

Los sistemas manejadores de bases de datos han sido usados ampliamente para Organizar y Manipular grandes Volúmenes de Datos de las Empresas.

Estos Sistemas Manejadores de Bases de Datos solo podían ser corridos en instalaciones computacionales grandes (Mainframes). Afortunadamente esta situación empezó a cambiar debido al bajo costo de las mini/microcomputadoras y a la aparición de Sistemas Manejadores de Bases de Datos para las microcomputadoras.

Un sistema manejador de base de datos no es más que “Un sistema computarizado para llevar registros”.

Algunas de las facilidades que proporciona el sistema a los usuarios son:

- Agregar Nuevos Archivos a la Base de Datos.
- Agregar Nuevos Registros a los Archivos existentes.

- Recuperación de Datos.
- Actualización de Datos.
- Borrar registros.
- Borrar Archivos.
- Proporcionar los mecanismos para el control del acceso concurrente a los datos.

Una base de datos es una colección de información útil organizada en una manera específica. Por ejemplo para llevar el control de los teléfonos de personas podría ser útil una base de datos que constará de los datos de la siguiente figura[Hal1994]:

Nombre	Teléfono
Vargas Rubén	18-20-67
Díaz Raúl	13-67-90
Ruiz Pedro	12-89-90
Martínez Raúl	12-54-46

Tabla para llevar una agenda telefónica.

3.4. Componentes de Base de Datos

Toda base de datos está formada por uno o varios bloques de información llamados "Tablas" que normalmente tendrán alguna característica en común. Una Tabla o archivo de datos es un conjunto conexo de información del mismo tipo.

Cada tabla está formada por "registros". Un "registro" es la unidad elemental de información de la tabla o fichero. Cada registro está formado por uno o más elementos llamados "campos". Un "campo" es cada una de las informaciones que se interesa almacenar en cada registro y es por tanto la unidad elemental de información del registro. Si hablamos de una Universidad como un ejemplo, las entidades de interés podrían ser: Alumnos, Profesores, Salones, etc. Respecto a la entidad Alumno, los atributos o características importantes podrían ser: Nombre, Dirección, Teléfono, Carrera, Matricula, edad, etc.

Un archivo de una base de datos también puede ser pensado como una tabla en la que tenemos renglones y columnas, cada renglón correspondiendo a un registro del archivo y cada columna correspondiendo a un campo.

3.5. Enfoque de Base de Datos

Existen 3 enfoques principales para el manejo de Bases de Datos:

- Enfoque Relacional.
- Enfoque Jerárquico.
- Enfoque de Redes.

3.5.1. Enfoque Relacional

El Enfoque Relacional proyecta a los datos como un conjunto de Tablas (Archivos) constando de un conjunto de Renglones (Registros) y cada renglón constando de un conjunto de Columnas (Campos). La relación entre las Tablas se establece solo por nombres de atributos comunes.

3.5.2. Enfoque Jerárquico

La base de datos Jerárquica consta de muchos registros. Cada nodo representa un Tipo de registros conceptual a un segmento. Cada registro o segmento está constituido por un cierto número de campos que lo describen.

3.5.3. Enfoque de Red

Para representar este tipo de relación, es necesario que los dos tipos de registro estén interconectados por medio de un registro conector llamado conjunto conector [Hal1994].

3.6. Modelado de Base de Datos del Sistema.

Basados en el enfoque Relacional nos centramos en realizar un modelado de Base de Datos que concuerde con las necesidades especificadas, para esto los pasos que tenemos que seguir son:

- Realizar el Modelado Lógico de Entidad-Relación
- Realizar el Modelado Entidad-Relación
- Normalización de Atributos y Entidades
- Diagrama Entidad Relación

3.6.1. Modelado Entidad-Relación

Se pretende desarrollar el modelado del proyecto de Base de Datos “Sistema Administrador del Seguimiento de Proyectos de Investigación” que comprende los Proyectos asignados a un (unos) Investigador(es) para su desarrollo.

- a) El administrador dará de alta los Profesores que pueden acceder al sistema.
- b) El administrador dará de alta los Alumnos involucrados a los proyectos.
- c) El alumno y profesor podrá actualizar sus Datos personales, para tenerlos al día.
- d) Cualquier usuario del sistema podrá cambiar su password de ingreso al sistema.

- e) Los profesores y Alumnos podrán ingresar una imagen fotográfica dentro de sus Datos personales.
- f) El administrador podrá dar de alta los proyectos, así como vincular el asesor (Profesor) y el investigador (Alumno) al mismo.
- g) Se mantendrá la información para los proyectos identificando cada uno e informando su título, fecha de Alta, así como la fecha de inicio y Fecha Final, identificando si el proyecto es Nacional o Extranjero.
- h) Cada proyecto tendrá un Status el cual nos dará una idea del avance del mismo.
- i) En el proyecto participaran Alumnos dependiendo del tipo de proyecto (Individual o Grupal)
- j) Los Alumnos realizan Documentos como parte de la investigación de los proyectos.

De acuerdo a estas premisas se realiza el siguiente Modelado Lógico:

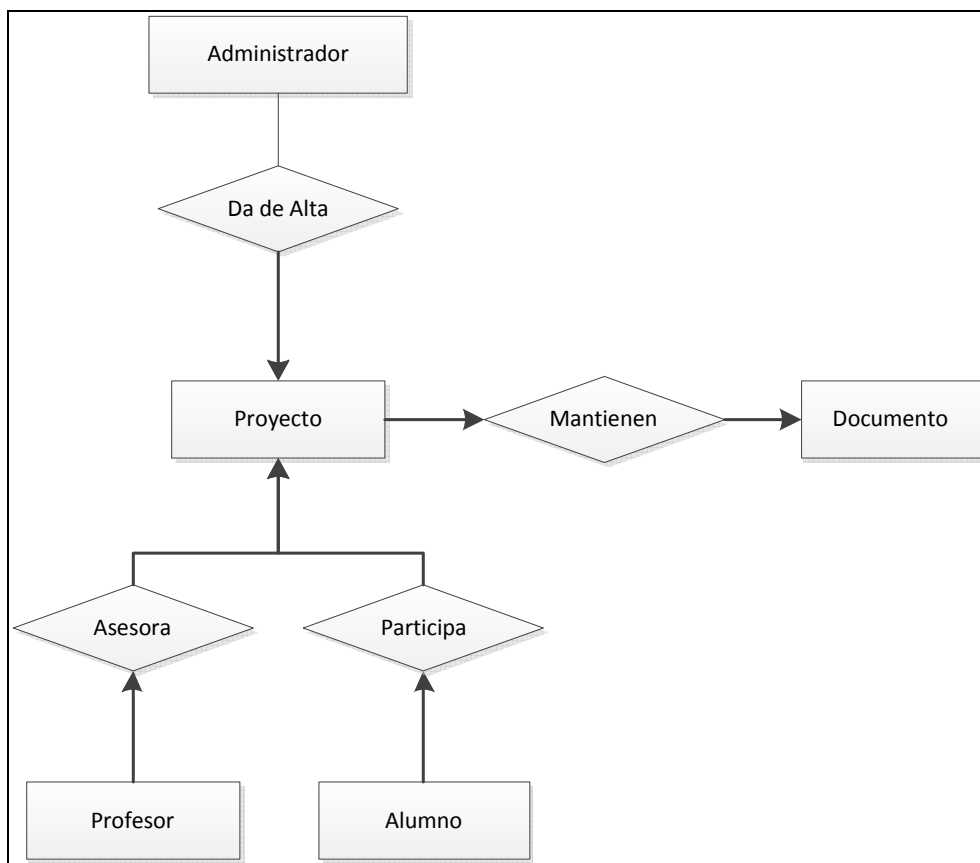


Fig. 3.1 Modelado Lógico General del Sistema

Del cual podemos detallar cada parte y generar los detalles y atributos del diagrama Entidad-Relación, para así tener el diagrama del Administrador como sigue:

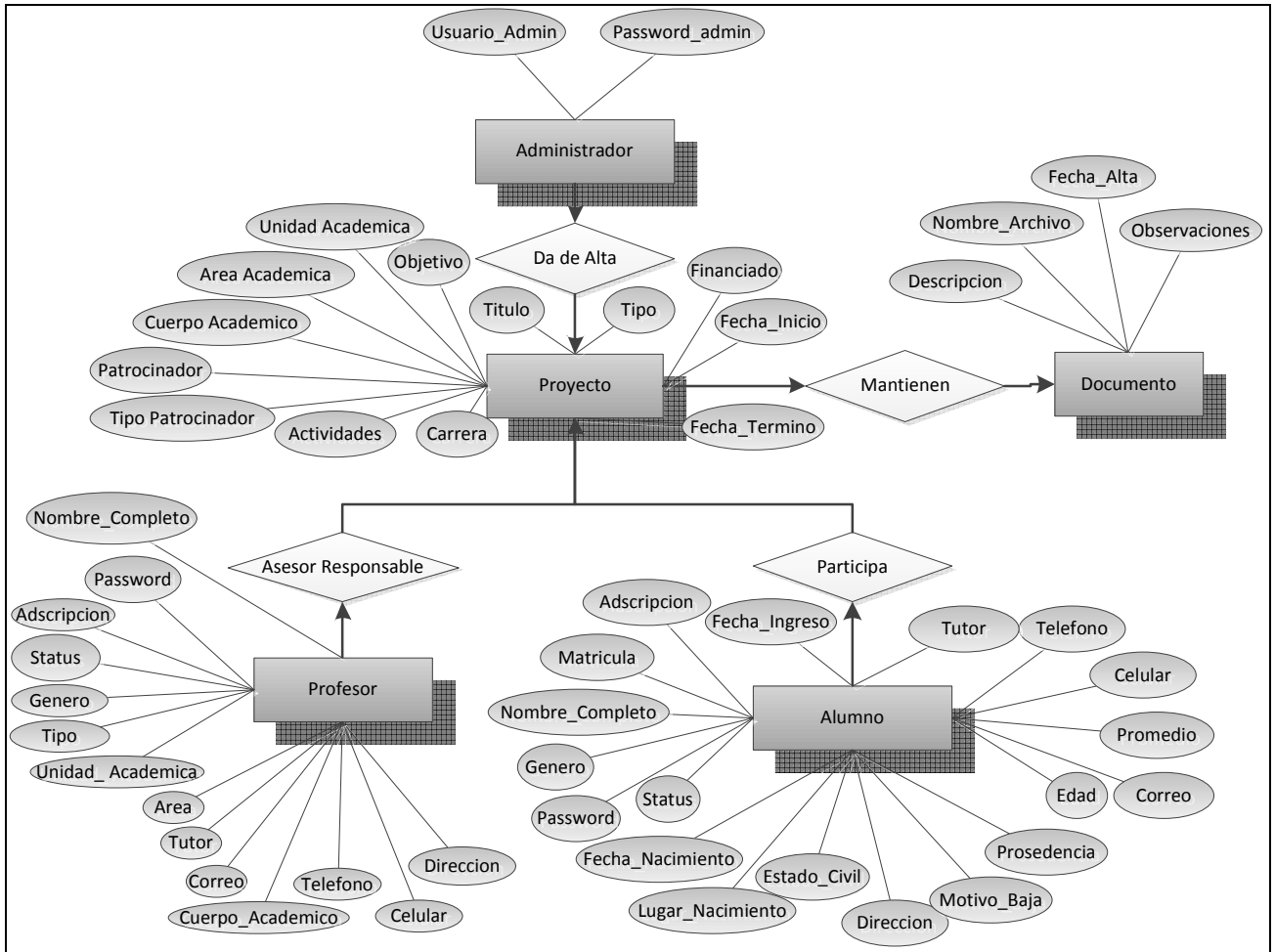


Fig. 3.2 Atributos E-R de Entidad Administrador

Tomando las entidades de nuestro 1er diagrama E-R tenemos linealmente las siguientes entidades y sus atributos:

Proyectos						
Titulo	Objetivo	Unidad_Academica	Area_Academica	Cuerpo_Academico	Patrocinador	Responsable
Tipo_Patrocinador	Actividades	Carrera	Fecha_Termino	Fecha_Inicio	Financiado	Tipo
Participante						

Profesor						
Nombre_Completo	Password	Adscripcion	Status	Genero	Tipo	Celular
Unidad_academica	Area	Tutor	Correo	Cuerpo_Academico	Telefono	Direccion

Alumno						
Nombre_Completo	Matricula	Fecha_Ingreso	Adscripcion	Genero	Password	Status
Fecha_Nacimiento	Estado_Civil	Lugar_Nacimiento	Direccion	Motivo de Baja	Procedencia	Edad
Correo	Promedio	Celular	Telefono	Tutor		
Administrador						
Usuario_Admin	Password_admin					
Documentos						
Descripcion	Nombre_Archivo	Fecha_Alta	Observaciones			

Fig. 3.3 Atributos de Entidades del Sistema

3.6.2. Normalización de Modelado

Las formas normales, vistas como las diferentes propiedades que deben cumplir los esquemas de relación con dependencias, tienen una importancia significativa en el diseño de bases de datos. Particularmente importantes son las denominadas “Tercera Forma Normal” y “Forma Normal de Boyce-Codd”. Las formas normales garantizan que muchos de los problemas de redundancia y de anomalías en una base de datos, no ocurran.

Tomando las entidades de nuestro 1er diagrama E-R tenemos linealmente las siguientes entidades y sus atributos:

3.6.2.1. Primera Forma Normal

Codd definió: “Una relación se encuentre en Primera Forma Normal (1FN), si tiene la propiedad de que ninguno de sus dominios tienen elementos que sean en sí mismos conjuntos”, lo que significa que los registros de una relación necesitan tener el mismo número de atributos.

Un esquema de relación **R** esta en primera forma normal, si el dominio de cada atributo consiste de valores individuales.

Con esto las entidades no cumplen el requisito de la Primera Forma Normal (1NF) de sólo tener campos atómicos, pues el nombre de Profesor y Alumno es un campo que puede (y conviene) descomponerse en apellido paterno, apellido materno y nombres. Entonces para llevarlas a la 1ra forma normal tenemos lo siguiente:

Proyectos						
Titulo	Objetivo	Unidad_Academica	Area_Academica	Cuerpo_Academico	Patrocinador	Apellido_Pat_Responsable
Apellido_Mat_Responsable	Nombre_Responsable	Tipo_Patrocinador	Actividades	Carrera	Fecha_Termino	Fecha_Inicio
Financiado	Tipo	Apellido_Pat_Participante	Apellido_Mat_Participante	Nombre_Participante		
Profesor						
Apellido_Pat	Apellido_Mat	Nombre	Password	Adscripcion	Status	Genero
Tipo	Celular	Unidad_academica	Area	Tutor	Correo	Cuerpo_Academico
Telefono	Pais	Direccion				
Alumno						
Apellido_Pat	Apellido_Mat	Nombre	Matricula	Fecha_Ingreso	Adscripcion	Genero
Password	Status	Fecha_Nacimiento	Estado_Civil	Lugar_Nacimiento	Pais	Direccion
Motivo de Baja	Procedencia	Edad	Correo	Promedio	Celular	Telefono
Apellido_Pat_Tutor	Apellido_Mat_Tutor	Nombre_Tutor				
Administrador						
Usuario_Admin	Password_admin					
Documentos						
Descripcion	Nombre_Archivo	Fecha_Alta	Observaciones			

Fig. 3.4 Atributos en 1ra Forma Normal (Atomicidad)

3.6.2.2. Segunda Forma Normal

Un esquema de Relación **R** esta en Segunda Forma Normal, si se encuentra en Primera forma Normal y todos los atributos no primos de **R**, están en dependencia funcional completa con cada candidato a llave de **R**.

En otras palabras, una relación **R** viola la Segunda Forma Normal, si algún atributo no primo, es funcionalmente dependiente de un subconjunto de una llave, es decir, si existen dependencias parciales. Este hecho es relevante, en aquellos casos en que las llaves sean compuestas.

Así podemos llevar nuestro diseño a la 2da forma Normal:

Proyectos							
Clave_Proyecto	Titulo	Objetivo	Unidad_Academica	Area_Academica	Cuerpo_Academico	Patrocinador	Apellido_Pat_Responsable
	Apellido_Mat_Responsable	Nombre_Responsable	Tipo_Patrocinador	Actividades	Carrera	Fecha_Termino	Fecha_Inicio
	Financiado	Tipo	Apellido_Pat_Participante	Apellido_Mat_Participante	Nombre_Participante		
Profesor							
No_Profesor	Apellido_Pat	Apellido_Mat	Nombre	Password	Adscripcion	Status	Genero
	Tipo	Celular	Unidad_academica	Area	Tutor	Correo	Cuerpo_Academico
	Telefono	Pais	Direccion				
Alumno							
No_Alumno	Apellido_Pat	Apellido_Mat	Nombre	Matricula	Fecha_Ingreso	Adscripcion	Genero
	Password	Status	Fecha_Nacimiento	Estado_Civil	Lugar_Nacimiento	Pais	Direccion
	Motivo de Baja	Procedencia	Edad	Correo	Promedio	Celular	Telefono
	Apellido_Pat_Tutor	Apellido_Mat_Tutor	Nombre_Tutor				
Administrador							
Clave_Admin	Usuario_Admin	Password_admin					
Documentos							
Clave_Proyecto	Descripcion	Nombre_Archivo	Fecha_Alta	Observaciones			

Fig. 125 Atributos en 2da. Forma Normal

Los atributos no primos dependen funcionalmente de la llave primaria, y como la llave primaria es también llave candidata, se cumple con la Segunda forma Normal.

3.6.2.3. Tercera Forma Normal

Un esquema de Relación **R** está en tercera Forma Normal, si cuando **X** es una llave de **R**, **Y** ⊂ **R** es un conjunto de atributos y **A** es un atributo no-primario de **R** que no está ni en **X** ni en **Y**, no ocurre que:

1. $X \Rightarrow Y$ se cumple en **R**
2. $Y \Rightarrow A$ se cumple en **R**, pero
3. $Y \Rightarrow X$ no se cumple en **R**

Como se ve en el sistema tienen existencias transitivas que no dependen de la llave primaria, por lo cual formamos unas nuevas entidades para llevarla a una 3ra. Forma Normal:

Proyectos							
Clave_Proyecto	Titulo	Objetivo	Clave_Unidad	Area_Academica	No_Cuerpo	Patrocinador	No_Profesor_Responsable
Tipo_Patrocinador	Actividades	Clave_Carrera	Fecha_Termino	Fecha_Inicio	Financiado	Tipo	No_Alumno_Participante
Unidad_Academica							
Clave_Unidad	Nombre	Pagina_Web	Area_Buap	Ubicacion			
Area							
Clave_Area	Nombre	Descripcion	Adscripcion	Clave_Unidad			
Area_Buap				Pais			
Clave_Area_Buap	Nombre	Descripcion			Clave_Pais	Nombre	
Cuerpo_Academico							
No_Cuerpo	Nombre	Descripcion	Siglas	estado	Clave_Unidad		
Carrera							
Clave_Carrera	Nombre	Clave_Unidad	Institucion	Ubicacion	telefono	fax	
Profesor							
No_Profesor	Apellido_Pat	Apellido_Mat	Nombre	Password	Clave_Carrera (Adscripcion)	Status	Genero
Tipo	Celular	Clave_Unidad	Clave_Area	Tutor	Correo	No_Cuerpo	Telefono
Clave_Pais	Direccion						
Alumno							
No_Alumno	Apellido_Pat	Apellido_Mat	Nombre	Matricula	Fecha_Ingreso	Clave_Carrera (Adscripcion)	Genero
Password	Status	Fecha_Nacimiento	Estado_Civil	Lugar_Nacimiento	Clave_Pais	Direccion	Motivo de Baja
Procedencia	Edad	Correo	Promedio	Celular	Telefono	No_Profesor_Tutor	

Fig. 3.6 Atributos en 3da. Forma Normal

De lo cual podemos realizar un diagrama Entidad-Relación de la forma más apegado a una Base de Datos de la forma que las relaciones sean representadas por líneas continuas o discontinuas y al finalizar las dichas "patas de gallo" para visualizar su grado como el siguiente:

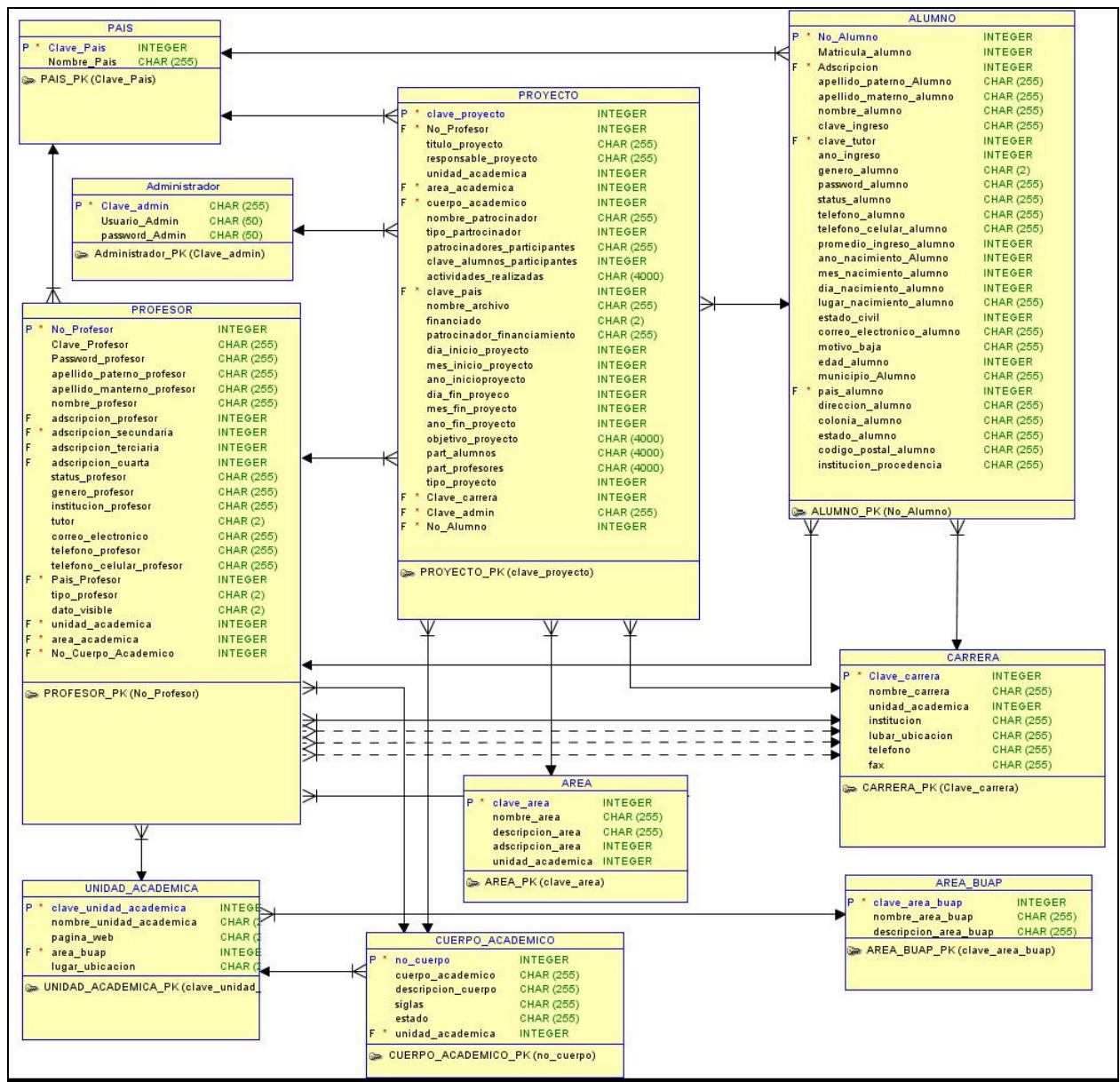
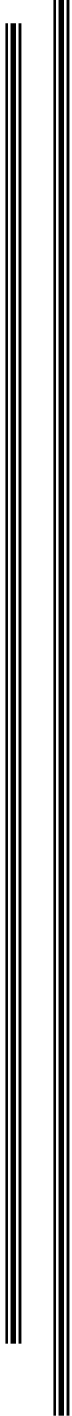


Fig. 3.7 Diagrama Entidad-Relación



Capítulo 4

DISEÑO DE ENTORNO WEB

4. DISEÑO DE ENTORNO WEB.

4.1. Antecedentes

El diseño web es mucho más que una aplicación de buen gusto. Se trata de una serie de técnicas destinadas a hacer al sitio útil y atractivo a la vez

La cantidad de páginas web existentes se pueden contar en miles de millones. Sin embargo, un bajo porcentaje de ellas tienen un buen diseño, debido fundamentalmente a que muchos diseñadores web no siguen los principios básicos del diseño web, y la gran mayoría no son profesionales ni saben bien lo que hacen.

Podría pensarse que es nada más que la aplicación del buen gusto. Sin embargo, el diseño web es una técnica que se basa en una serie de reglas definidas que sirven para crear algo que resulte útil y atractivo a la vez.

Y de la mano del diseño puramente gráfico del sitio, va toda una planificación de los contenidos del sitio, la estructura de los menús, cómo será la interacción con el usuario, rutas de navegación, arquitectura de información y en definitiva, todo un diálogo que ha de establecerse entre el sitio web y el visitante, que debe pensarse y planificarse para que éste último entienda de qué va el sitio y sepa usarlo y no saber hacer. El diseño web es mucho más que darle color y estilo a una página web.

4.1.1. Principios de Diseño WEB

Durante la planificación del proyecto, debe tenerse en cuenta que es muy probable que exista algo similar en la web, lo que hace que el diseñador se vea obligado a hacer un buen diseño para poder competir. El diseño web tiene dos partes bien diferenciadas:

- La Funcionalidad o Usabilidad.
- La Estética

Ambas son imprescindibles, ya que una página funcional pero con mala estética no atraerá la atención de posibles usuarios o visitantes, mientras que una página muy bonita pero que funciona mal hará que los visitantes que lleguen atraídos por la buena estética del sitio no regresen a él por no encontrar lo que necesitan o les resulte difícil.

4.1.2. Funcionalidad o Usabilidad

Cuando accedemos a alguna de las muchas fuentes de información, desde los simples libros hasta los ordenadores, encontramos que la información contenida está organizada

de forma tal que puede accederse a los contenidos con facilidad. En algunos casos, como en los libros, por ejemplo, el uso continuo por generaciones ha hecho que dicha organización se encuentre estandarizada y sea muy sencillo para los lectores encontrar cada uno de los elementos que lo constituyen.

Siguiendo el ejemplo de los libros, el orden en que aparecen los contenidos está descrito en el índice, que generalmente se encuentra al principio o al final del mismo. Además, la propia forma física del libro, la forma en que se encuentran ordenadas las páginas, hacen que la interfaz sea muy sencilla e intuitiva.

En las páginas web, esta forma de ordenar las páginas no es posible, por lo que el ordenamiento y la clasificación de las secciones, páginas y contenidos deben realizarse empleando otros métodos, pero que permitan acceder a cada uno de ellos con la misma facilidad.

La interfaz

El concepto de interfaz es muy amplio y se refiere a todo sistema que permite el contacto y la funcionalidad entre dos sistemas diferentes. Por ejemplo, los botones y la pantalla del teléfono celular conforman la interfaz, ya que permite que el usuario pueda emplear las funciones que este aparato ofrece.

Una página web puede contar con los mejores contenidos en el género que se desarrolla, pero indefectiblemente fracasará si su interfaz no permite un rápido y cómodo acceso a los mismos por parte de los usuarios. Por el contrario, una página web cuyos contenidos sean de menor calidad (sin que éstos sean malos, por supuesto) pero cuya interfaz permite que sus usuarios naveguen en forma sencilla, tengan acceso en forma inmediata al contenido que desean e interactúen en forma sencilla, tendrá un mayor éxito.

Simplicidad

La facilidad y comodidad con que los usuarios acceden a los servicios que brinda una web está fundado en dos principios fundamentales para el desarrollo de una interfaz efectiva:

- **Simplicidad** La simplicidad con que se desarrolle esta interfaz es crucial para determinar que un usuario se sienta satisfecho y desee regresar a un sitio. El hecho de que una persona deba realizar una extensa navegación por el sitio para hallar lo que busca en él es totalmente contraproducente. Por el contrario, si alguien que visita un sitio cuenta con varias herramientas que le permiten acceder rápidamente a aquello que le interesa, seguramente volverá.

Un ejemplo de simplicidad lo encontramos en la página principal del buscador “Google”.



Fig. 4.1. Simplicidad en una Página

Si un usuario llega a un sitio cuya navegación se le hace compleja, terminará por volver al buscador para seleccionar otro sitio que trate de temas similares.

La Navegabilidad

Una de las razones por las que los usuarios de Internet seleccionan los sitios de su preferencia es la facilidad con la que pueden acceder a los diferentes contenidos y poder navegar en forma sencilla entre las diferentes páginas, secciones, etc.

Todo diseñador debe tener en cuenta que el usuario debe sentirse cómodo mientras permanece en el sitio. Esta facilidad de manejo del sitio es una de las razones por las cuales las personas deciden volver o no a un sitio determinado.

4.1.3. Estética

El diseño web desde el punto de vista estético, equivale a explicar cuáles son los criterios que se deben seguir para realizar una página atractiva y efectiva.

Selección del Tipo de Diseño:

El diseño de una página web debe estar acorde con el contenido de la misma. Esto quiere decir que la selección del tipo de diseño que va a llevar nuestra web debe estar acorde con lo que el usuario busca en ella y con el producto o servicio ofrecido por la misma.

El diseñador deberá analizar estos factores y planificar su trabajo, haciendo una maqueta. Esta maqueta no es más que un dibujo de lo que programará posteriormente.

Para hacer una maqueta primero se piensan y deciden los contenidos que va a tener el sitio, con papel y lápiz se estructuran y bocetan las secciones principales, posicionando cada elemento de la página en el sitio que se considere apropiado, y una vez se tiene en papel un boceto de cómo será la página web, se pasa a diseñar la parte gráfica

empleando programas de dibujo. El Photoshop es muy usado para tal efecto, pero Gimp cumple perfectamente su función y es gratis.

Esta maqueta hecha en Photoshop tendrá además otra funcionalidad, ya que podrá extraerse de ella las imágenes para crear fondos, menús, botones, etc.

Debemos comentar que el diseño web debe seguir los mismos principios que rigen cualquier clase de diseño. Una página web, la portada de un libro, una revista, un periódico o cualquier otra publicación deben estar diseñados bajo los mismos principios, entendiendo las diferencias que corresponden a cada medio.

Los principios que rigen el diseño web son:

- **Balance**– Es el equilibrio que debe existir entre los diferentes elementos que componen la página. Imágenes y texto, elementos grandes y pequeños, zonas oscuras y claras, etc., deben estar balanceadas de forma tal que la página no solo resulte atractiva, sino que además logre enfocar la atención del usuario sobre aquellos elementos que nos interesa que lo haga.
- **Contraste**– El contraste es una forma de distinguir los elementos que deseamos resaltar. Diferencias en la tipografía, diferencias en las formas (círculos, cuadrados, rectángulos, etc.), diferencias de tamaño (una imagen más grande con otras más pequeñas, por ejemplo), texturas y fondos diferentes, etc., son algunos de los contrastes que pueden emplearse para enfocar la atención del usuario sobre los elementos más importantes de la página
- **Énfasis** – Es una forma de distinguir algo dentro de un elemento. Un ejemplo de esto es lo que sucede con este texto, donde los títulos se encuentran resaltados así como algunas palabras o frases, de forma que facilita la comprensión de lo que se quiere transmitir.
- **Repetición o ritmo**– El ritmo o repetición es una forma de establecer patrones con los que se ordenan los elementos de la página. Si los elementos de una página web siguen un patrón determinado, es mucho más sencillo para las personas comprender el contenido y acceder a la mayor parte del mismo en poco tiempo. Este punto es clave para mantener la usabilidad del sitio.
- **Proximidad o unidad**– Este concepto se refiere a colocar juntos o próximos los elementos que estén relacionados, formando unidades visuales que brindan coherencia al contenido de la página.

4.2. Diseño Web del Sistema

Debido a que es un sistema relacionado con Base de datos y accedido por muchos usuarios para el ingreso y manipulación de información se torna el siguiente diseño de pantallas para nuestro sistema gestor de Proyectos de Investigación

4.2.1. Flujo de Páginas WEB del Sistema

Detallando la página principal y teniendo en cuenta los errores de logueo de usuario tenemos la siguiente secuencia de navegación de pantallas WEB del Sistema gestor de Proyectos:

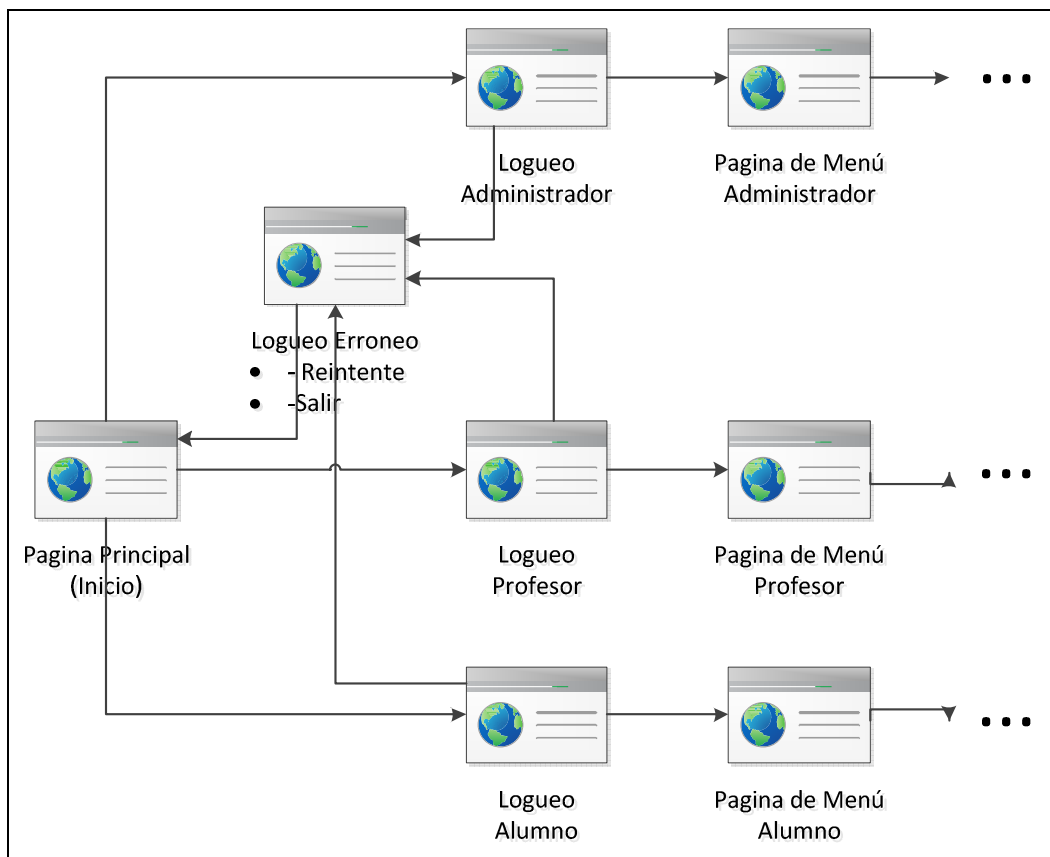


Fig. 4.2 Diseño de Navegación de Página Principal

Una vez que hemos entrado a la página del Menú de Alumnos nos encontraremos la navegación hacia las páginas de visualización y actualización de cierta información concerniente a los datos de cada Alumno, así como el cambio de password para la entrada al sistema, y también podremos navegar por un listado de proyectos ya agregados

por el administrador del sistema, así como podremos navegar a la página de descargar de los documentos asociados al proyecto seleccionado en la página de listado.

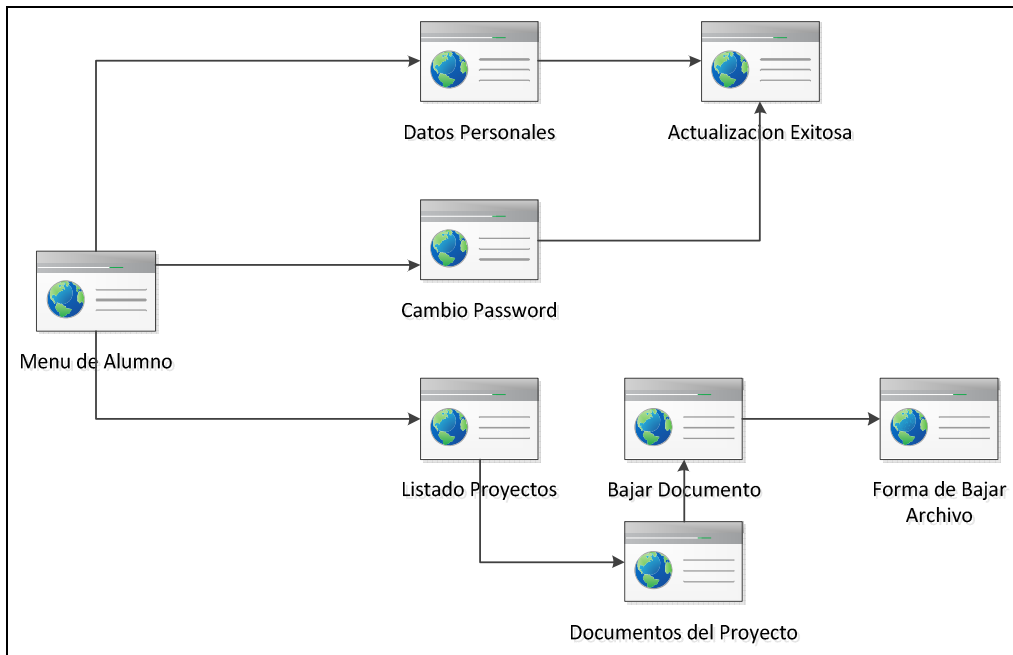


Fig. 4.3 Navegación por Menú con Rol de Alumno

De una forma similar es la navegación por las páginas del Menú de Profesor ya que conllevan los mismos privilegios, solo que los datos registrados en cada página son con referencias diferentes, ya que los profesores tienen características específicas a la institución:

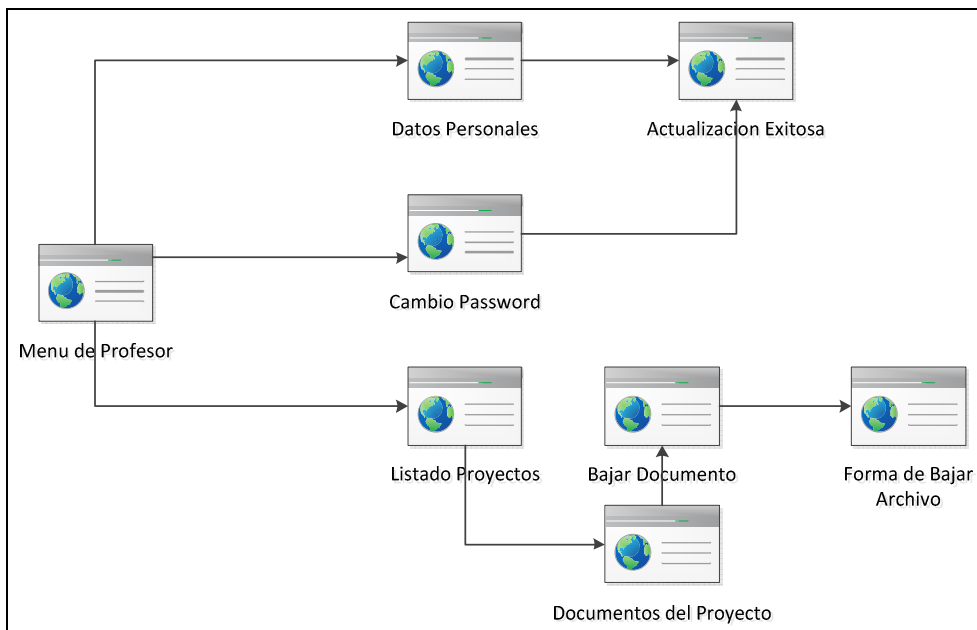


Fig. 4.4 Navegación por Menú con Rol de Profesor

La navegación por parte del Rol Administrador está un poco más compleja ya que conlleva las páginas que pueden acceder a toda la información del sistema, como son catálogos y realizar reportes estadísticos los cuales visualizaremos dentro de otros Submenús, pero que los privilegios para este rol son específicamente para darle soporte y mantenimiento a toda la información del Sistema Gestor de Proyectos de Investigación, desde Registrar Alumnos, Profesores que puedan ingresar al sistema, hasta la manera de imprimir reportes los cuales nos den un panorama general de cómo se están comportando nuestros proyectos a nivel estadístico:

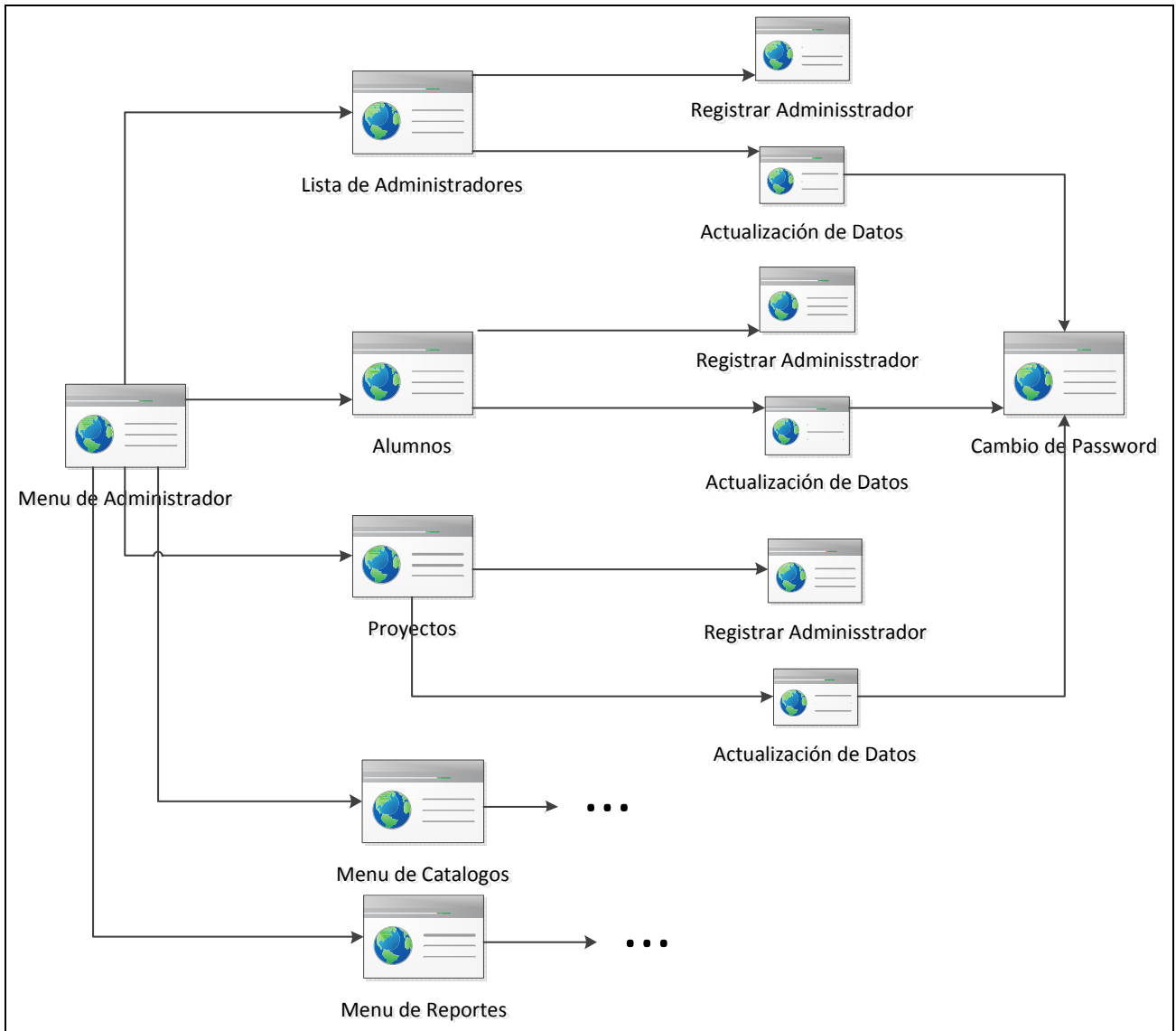


Fig. 4.5 Páginas de navegación del rol Administrador

Detallando la navegación de los catálogos es simple, ya que es la información de nuestro sistema que el 90% de ella no se mueve (solo en la carga inicial) pero que tenemos que tener contemplada para darle dinamismo a nuestro sistema y que podamos efectuar un

cambio sin que esto conlleve programación al momento de actualizar la información, por lo cual la secuencia de navegación se muestra en las siguientes pantallas:

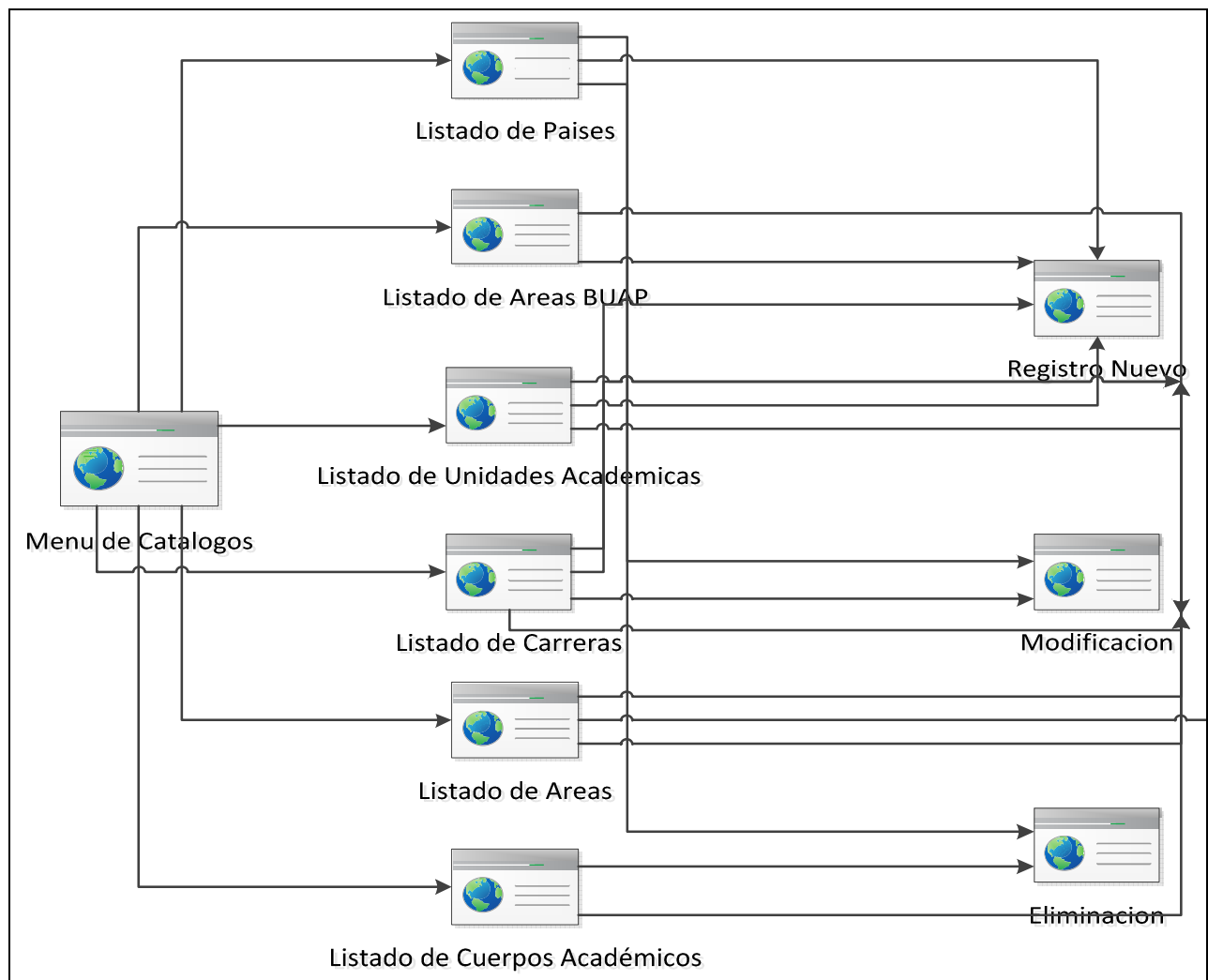


Fig. 4.6 Navegación de páginas del Submenú de Catálogos

Por ultimo pero no menos importantes tenemos la navegación de nuestras páginas de reportes la cuales están divididas en 2 partes, la parte de reportes generales de Alumnos, profesores y Administradores, la cual podremos filtrar por nombre o apellido para poder analizar el reporte de una manera eficaz; y la navegación de los informes estadísticos, que serán basados en proyectos, pero que podremos filtrarlos por las siguientes 4 características

- Área Académica
- Profesor
- Carrera
- Status

Y que cada uno contendrá un diferente tipo de Grafica para poder visualizar de forma rápida el comportamiento de cada proyecto.

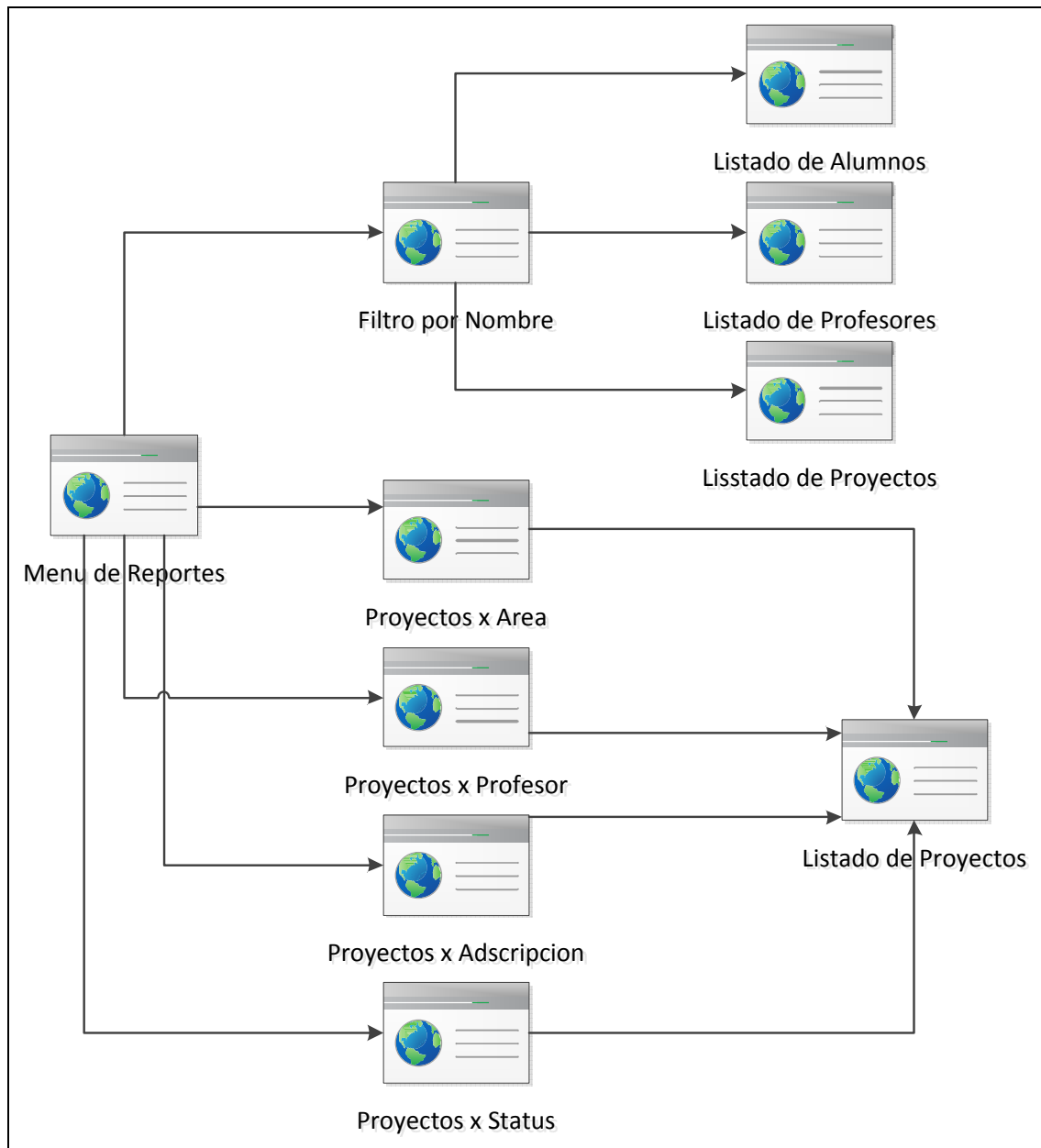


Fig. 4.7 Navegación de reportes generales y Estadísticos



Capítulo 5

IMPLEMENTACION DEL SISTEMA

5. IMPLEMENTACION DEL SISTEMA.

5.1. Acceso al Sistema.

El acceso a nuestro sistema es mediante una página web, para efectos de nuestra tesis se realizara de acuerdo a una computadora la cual tomaremos como servidor denominando “**localhost**” y será mediante la dirección <http://localhost/ffyl/index.html> como se ve en la siguiente figura:

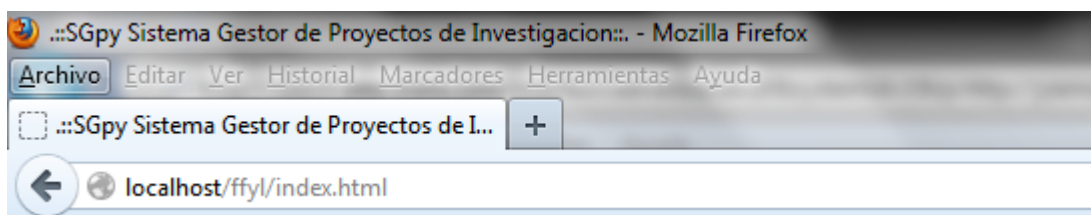


Fig. 5.1 Página de Acceso al sistema

5.2. Página de Inicio.

Una vez que entramos a la página antes mencionada, se muestra una página de bienvenida al sistema en la cual podremos ver el nombre de nuestro sistema, además de visualizar ciertas imágenes de un entorno académico, lo cual va muy relacionado con nuestro proyecto, y lo más importante que es el acceso a nuestros Formularios, mediante el rol que tengamos en nuestro sistema:

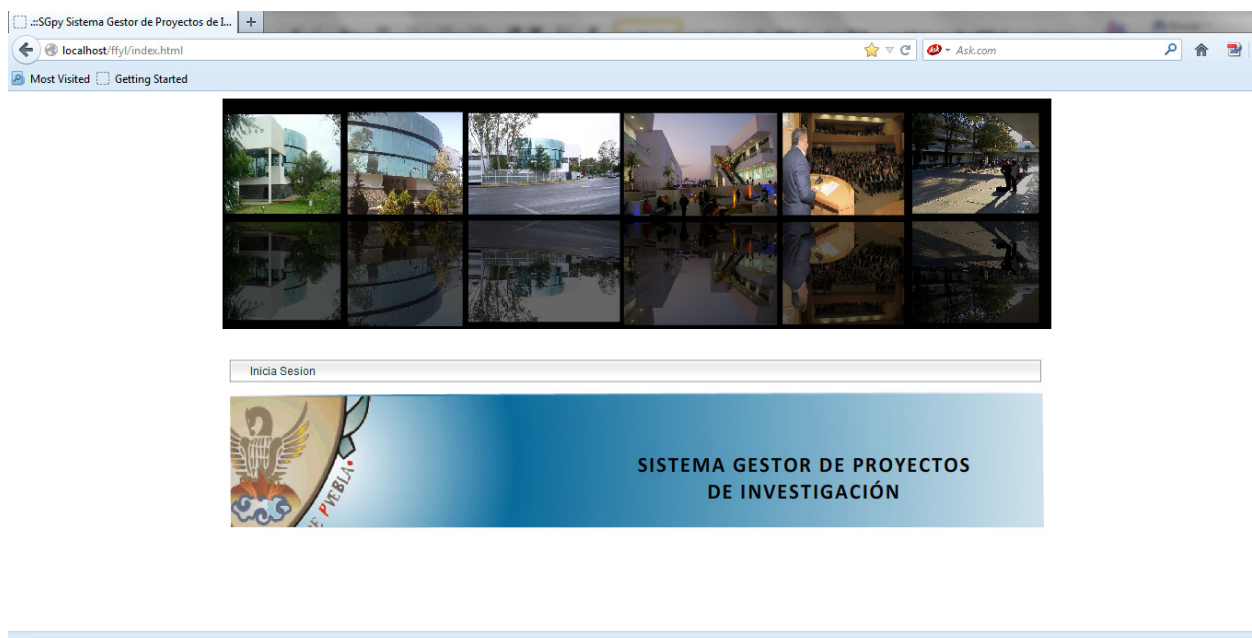


Fig. 5.2 Página de Inicio y bienvenida del Sistema

5.3. Menú de Acceso por Rol

Como ya mencionamos, podremos ingresar mediante el Rol que tengamos lo cual distingue a cada usuario por los privilegios que tenga para poder gestionar los proyectos, como lo son:

- **Alumno**
- **Profesor**
- **Administrador**

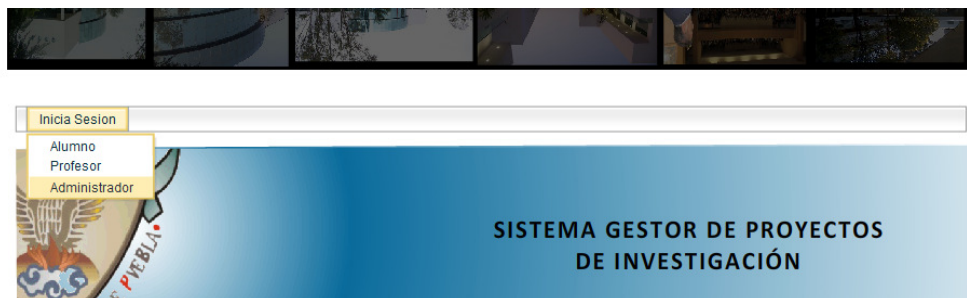


Fig. 5.3 Roles de Usuario para su Acceso al Sistema

5.4. Acceso de Alumnos

Una vez que ingresamos como alumno, nos muestra una pantalla para capturar nuestra Matricula y nuestro password que es la seguridad de nuestro sistema:



Matrícula del alumno:

Password del alumno:

- [Regresar](#)

Fig. 5.4 Página de entrada al sistema para Rol de alumnos

5.5. Menú de Alumnos

Una vez que nos firmamos como alumno valido, nos muestra la página que contiene el menú sobre los privilegios que tenemos dentro del sistema:

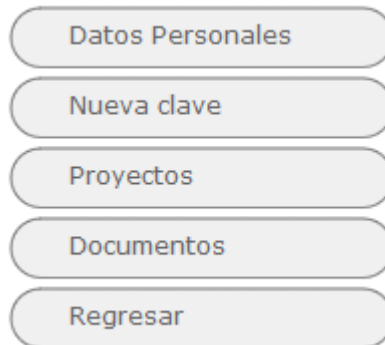


Fig. 5.5 Menú del Rol de Alumno

5.5.1. Actualización de Datos

La primera opción de nuestro menú es la de Actualización de datos, la cual al seleccionarla nos muestra la página con nuestro nombre y solo los datos que podemos actualizar debido a que hayamos cambiado los datos más volátiles:

 A screenshot of a web browser showing a student data update page. The browser address bar shows "localhost/ffyl/principal/alumnos/index.html". The page has a red header with "ZAMORA CRUZ JUAN". On the left is a sidebar menu with buttons for "Datos Personales", "Nueva clave", "Proyectos", "Documentos", and "Regresar". The main content area has a "Enviar datos" button and is divided into sections: "Identificación (no modificables)" with fields for Matricula (950012566), Apellido Paterno (ZAMORA), Apellido Materno (CRUZ), Nombre (JUAN), and Adscripción (LIC EN CIENCIAS DE LA COMPUTACIÓN); "Contacto" with fields for Telefono (2222207143), Telefono Celular (5529631164), and Correo electrónico (jzamorac007@hotmail.com); and "Domicilio" with fields for Calle (17 NORTE), Colonia (LA LOMA), Estado (PUEBLA), Codigo postal (71230), and Pais (Mexico).

Fig. 5.6 Página de Actualización de Datos del Alumno

5.5.2. Actualización de Password

La segunda opción es la de Actualizar nuestra clave de entrada, esto es en beneficio de nuestra seguridad, ya que es recomendable cambiarla con regularidad:

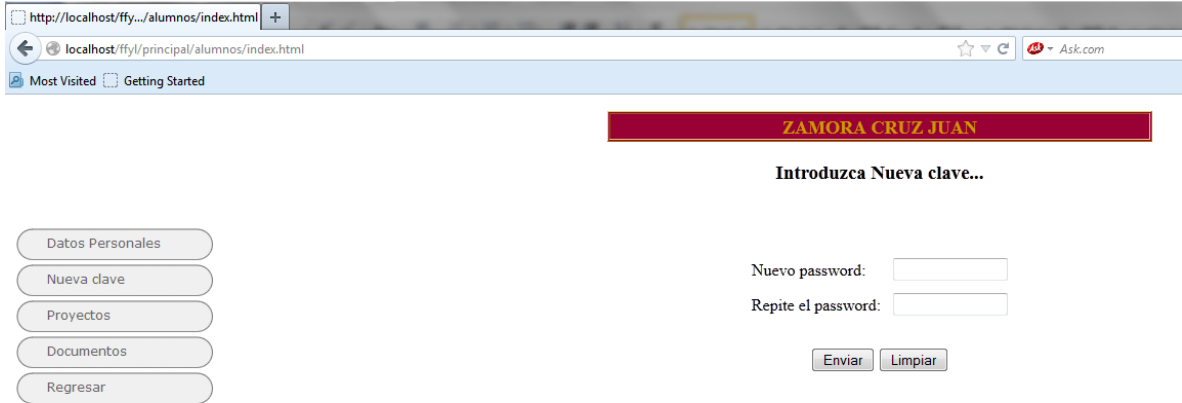


Fig. 5.7 Página de actualización de Contraseña de seguridad

5.5.3. Consulta de Proyectos

Como tercera opción tendremos la lista de nuestros proyectos, la cual nos da la posibilidad de seleccionar cualquier proyecto para visualizar su información, más detallada, además de si querer imprimir, tenemos la opción en la esquina inferior derecha con el grafico de una impresora



Fig. 5.8 Página de Lista de Proyectos

Si oprimimos el botón de Información del proyecto, podremos visualizar la página que contiene los detalles del mismo como en la siguiente figura:

PROYECTO: "Biblioteca Personal"

Datos del Proyecto

Status:

Patrocinador:

Patrocinadores Participantes:

Institución responsable del proyecto:

Objetivo:

Actividades a realizar:

Participantes profesores:

Participantes alumnos:

Datos de Origen

Responsable del Proyecto:

Tipo de Proyecto:

Colegiado o posgrado:

Area Académica:

Cuerpo Académico:

Escala de Inicio del Proyecto:

Fig. 5.9 Página que detalle los pormenores del Proyecto

5.5.4. Visualización de Documentos

Como cuarta opción tenemos la lista de proyectos, la cual visualizamos el nombre del proyecto, y el archivo ya ingresado, para poder obtener la tesis mediante un archivo con extensión "**PDF**"

Documentos ingresados

Titulo	Tipo	Reg.	Estado	Archivo	Director de tesis
FUNDAMENTACIÓN DE UN FEMINISMO FILOSÓFICO EN LA OBRA DE CELIA AMORÓS	Tesis	2008	rechazada	bruebel.pdf	GARCIA AGUILAR MARIA DEL CARMEN
REFLEXIONES DESDE PAULO FREIRE	Tesis	2008	rechazada		FERNANDEZ Y DIAZ RODOLFO MAXIMINO
LA FILOSOFÍA DE LA VIDA ARTÍSTICA EN SAMUEL RAMOS	Tesis	2008	registrada		CABEZAS HIGAREDA ANTONIO
TOMÁS DE MERCADO Y LA HERMENÉUTICA ANALÓGICA	Tesis	2008	registrada		CAMPOS BENTEZ JUAN MANUEL
LA FILOSOFÍA POLÍTICA DE LA DEMOCRACIA DELIBERATIVA DE JÜRGEN HABERMAS	Tesis	2007	registrada		CARRERA ALVAREZ VICENTE
LNCHAMIENTO: MORALIDAD SUI GÉNERIS Y PARADOJA MORAL	Tesis	2007	registrada		ROMANO RODRIGUEZ MARIA DEL CARMEN
EL CONCEPTO DE ACCIÓN EN VITA ACTIVA DE HANNAH ARENDT	Tesis	2008	registrada		CARRERA ALVAREZ VICENTE
ILUSTRACIÓN Y MITO EN DIALÉCTICA DE THEODOR ADORNO Y MAX HORKHEIMER	Tesis	2008	registrada		CARRERA ALVAREZ VICENTE
LAS MULTIPLES DETERMINACIONES DEL YO EN DESCARTES	Tesis	2008	registrada		GIBU SHIMABUKURORICARDO ANTONIO
EL CONCEPTO DE AMOR EN SAN AGUSTÍN Y JOSÉ ORTEGA Y GASSET	Tesis	2008	registrada		GIBU SHIMABUKURORICARDO ANTONIO
EL PAPEL DE LA INDICACIÓN FORMAL EN LAS PRIMERAS LECCIONES DE MARTÍN H	Tesis	2009	registrada		XOLOCOTZI YAÑEZ ANGEL
LA PERCEPCIÓN EN JEAN PAUL SARTRE	Tesis	2007	registrada		GJHGJ GJHGJGJHGJH
LA NEUROSIS NEOGÉNA COMO UNA CONSECUENCIA DE LA ENAJENACIÓN	Tesis	2008	registrada		PETER SILVARICARDO AGUSTÍN
LA TEORÍA CRÍTICA DE MAX HORKHEIMER	Tesis	2008	registrada		CARRERA ALVAREZ VICENTE
EL CONCEPTO DE LO ROMÁNTICO EN HEGEL	Tesis	2008	registrada		RIVAS LOPEZ VICTOR GERARDO
VALORACIONES Y PROBLEMAS DE LA FILOSOFIA MEXICANA CONTEMPORÁNEA	Tesis	2008	registrada		HERNANDEZ Y ALVAREZ JOSÉ DAVID

Fig. 5.10 Lista de Documentos Ingresados

Si seleccionamos el archivo el navegador nos mostrara una página para poder tener opciones de visualización, o para bajarlo a nuestra máquina del Cliente.

Documentos ingresados

Titulo	Tipo	Reg.	Estado	Archivo	Director de
FUNDAMENTACIÓN DE UN FEMINISMO FILOSÓFICO EN LA OBRA DE CELIA AMORÓS	Tesis	2008	rechazada	prueba1.pdf	GARCIA AGUILARMARIA D
REFLEXIONES DESDE PAULO FREIRE	Tesis	2008	rechazada		FERNANDEZ Y DIAZRODOI
LA FILOSOFÍA DE LA VIDA ARTÍSTICA EN SAMUEL RAMOS	Tesis	2008	registrada		CABEZAS HIGAREDAANTO
TOMÁS DE MERCADO Y LA HERMENÉUTICA ANALÓGICA	Tesis	2008	registrada		CAMPOS BENITEZJUANMA
LA FILOSOFÍA POLÍTICA DE LA DEMOCRACÍA DELIBERATIVA I					
LENCHAMIENTO: MORALIDAD SUI GÉNERIS Y PARADOJA MOR					
EL CONCEPTO DE ACCIÓN EN VITA ACTIVA DE HANNAH AREN					
ILUSRACIÓN Y MITO EN DIALÉCTICA DE THEODOR. ADORNO Y					
LAS MULTIPLES DETERMINACIONES DEL YO EN DESCARTES					
EL CONCEPTO DE AMOR EN SAN AGUSTÍN Y JOSÉ ORTEGA Y G					
EL PAPEL DE LA INDICACIÓN FORMAL EN LAS PRIMERAS LECCIONES DE MARTÍN H	Tesis	2009	registrada		XOLOCOITZI YANEANGEL
LA REPRESENTACIÓN EN EL ARTE. GOTTFR					



Fig. 5.11 Opciones de Visualización para el Documentos asociado al Proyecto

5.6. Acceso de Profesores

El segundo rol con el que podemos entrar a nuestro sistema es el de Profesores, el cual nos proporciona una página en la cual podremos identificarnos para que nos pase a una página con los privilegios correspondientes:



Clave del profesor:

Password del profesor:

Entrar

Limpiar

- [Regresar](#)

Fig. 5.12 Acceso a Rol de Profesores

5.7. Menú de Profesores

De la misma forma que el rol anterior, una vez que nos firmamos como profesor valido, pasamos a la página que contiene el menú sobre los privilegios que tienen los profesores dentro del sistema:

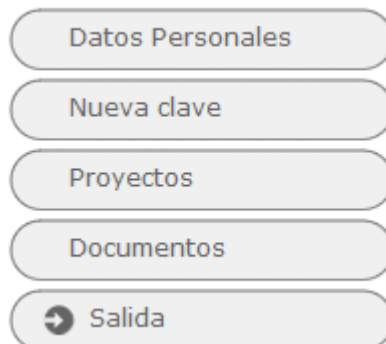


Fig. 5.13 Menú del Rol de Profesores

5.7.1. Actualización de Datos

La primera opción es la Actualización de nuestros datos como Profesor, ya que tenemos si de alguna manera están desactualizados, podremos darle mantenimiento mediante la siguiente página y enviarlos a la Base de Datos

RIVEROLL TORICHE PANTALEON

Identificación

Clave:

Apellido Paterno:

Apellido Materno:

Nombre:

Genero :

Tutor :

Adscripción 1a.:

Adscripción 2a.:

Adscripción 3a.:

Adscripción 4a.:

Contacto

Status :

Telefono:

Telefono Celular:

Correo Electrónico:

Complementos

Fig. 5.14 Actualización de Datos del Rol de Profesores

5.7.2. Actualización de Password

En este rol también encontramos la actualización de nuestro password como segunda opción ya que para darle mayor seguridad a nuestra información debemos actualizarlo periódicamente.

RIVEROLL TORICHE PANTALEON

Introduzca Nueva clave...

Nuevo password:

Repita el password:

Fig. 5.15 Página de Actualización de Password del Rol Profesores

5.7.3. Consulta de Proyectos

En nuestra tercera opción del menú de profesores nos muestra la página de la lista sobre los proyectos que tenemos en la institución, la cual nos da la posibilidad de seleccionar cualquier proyecto para visualizar su información más detalladamente, además imprimirlo.

Lista de Proyectos				
Titulo	Tipo	Fecha de Inicio	Fecha de Fin	Selección
Sistema Administrador del seguimiento de Proyectos de Investigación	Interno	2013	2013	<input type="radio"/>
Sistema simulador en líneas de Exámenes Departamentales del área de Programación de la FCC	Interno	2013	2013	<input type="radio"/>
Servicio Telemático de Foro de discusión no supervisado a partir de una aplicación Web	Interno	2013	2013	<input type="radio"/>
Servicio de una Agenda Electrónica en la Web	Interno	2013	2013	<input type="radio"/>
Control e Inventario en línea de la Infraestructura de la FCC	Interno	2013	2013	<input type="radio"/>
Sistema de asesorías en línea para profesores y alumnos de la FCC	Interno	2013	2013	<input type="radio"/>
Reservación de espacio para un evento	Interno	2013	2013	<input type="radio"/>
Tabla Interactiva para la Venta de Mascotas	Interno	2013	2013	<input type="radio"/>
Formulario dinámico para reservar un coche de renta.	Interno	2013	2013	<input type="radio"/>
Gestor de Gastos Personales	Interno	2013	2013	<input type="radio"/>
Biblioteca Personal	Interno	2013	2013	<input checked="" type="radio"/>

Fig. 5.16 Página que muestra la Lista de Proyectos

Podemos seleccionar algún proyecto del cual estemos interesados y oprimir el botón de información del proyecto para visualizar sus detalles como nos muestra la siguiente figura:

PROYECTO: "Biblioteca Personal"

Datos del Proyecto

Status:

Patrocinador:

Patrocinadores Participantes:

Institución responsable del proyecto:

Objetivo:

Actividades a realizar:

Participantes profesores:

Participantes alumnos:

Datos de Origen

Responsable del Proyecto:

Tipo de Proyecto:

Colegiado o posgrado:

Area Académica:

Cuerpo Académico:

Fecha de inicio del Proyecto:

Fig. 5.17 Página que detalla la información del Proyecto

5.7.4. Visualización de Documentos

En este rol también solo podremos visualizar los documentos referentes al proyecto desde la cuarta opción de nuestro menú, para que observemos aquellos archivos como pueden ser tesis, investigación, o cualquier otro documento que avale el proyecto:

Documentos ingresados

Titulo	Tipo	Reg.	Estado	Archivo	Director de tesis
FUNDAMENTACIÓN DE UN FEMINISMO FILOSÓFICO EN LA OBRA DE CELIA AMORÓS	Tesis	2008	rechazada	prueba1.pdf	GARCIA AGUILARMARIA DEL CARMEN
REFLEXIONES DESDE PAULO FREIRE	Tesis	2008	rechazada		FERNANDEZ Y DIAZ RODOLFO MAXIMINO
LA FILOSOFÍA DE LA VIDA ARTÍSTICA EN SAMUEL RAMOS	Tesis	2008	registrada		CABEZAS HIGAREDA ANTONIO
TOMÁS DE MERCADO Y LA HERMENÉUTICA ANALÓGICA	Tesis	2008	registrada		CAMPOS BENITEZ JUAN MANUEL
LA FILOSOFÍA POLÍTICA DE LA DEMOCRACÍA DELIBERATIVA DE JÜRGEN HABERMAS	Tesis	2007	registrada		CARRERA ALVAREZ VICENTE
LINCHAMIENTO: MORALIDAD SUI GÉNERIS Y PARADOJA MORAL	Tesis	2007	registrada		ROMANO RODRIGUEZ MARIA DEL CARMEN
EL CONCEPTO DE ACCIÓN EN VITA ACTIVA DE HANNAH ARENDT	Tesis	2008	registrada		CARRERA ALVAREZ VICENTE
ILUSTRACIÓN Y MITO EN DIALÉCTICA DE THEODOR ADORNO Y MAX HORKHEIMER	Tesis	2008	registrada		CARRERA ALVAREZ VICENTE
LAS MÚLTIPLES DETERMINACIONES DEL YO EN DESCARTES	Tesis	2008	registrada		GIBU SHIMABUKURORICARDO ANTONIO
EL CONCEPTO DE AMOR EN SAN AGUSTÍN Y JOSÉ ORTEGA Y GASSET	Tesis	2008	registrada		GIBU SHIMABUKURORICARDO ANTONIO
EL PAPEL DE LA INDICACIÓN FORMAL EN LAS PRIMERAS LECCIONES DE MARTÍN H	Tesis	2009	registrada		XOLOCOTZI YAÑEZ ANGEL
LA PERCEPCIÓN EN JEAN PAUL SARTRE	Tesis	2007	registrada		GJHG JHGHJGJHGJH
LA NEUROSIS NEOGÉNA COMO UNA CONSECUENCIA DE LA ENAJENACIÓN	Tesis	2008	registrada		PETER SILVARICARDO AGUSTÍN
LA TEORÍA CRÍTICA DE MAX HORKHEIMER	Tesis	2008	registrada		CARRERA ALVAREZ VICENTE
EL CONCEPTO DE LO ROMÁNTICO EN HEGEL	Tesis	2008	registrada		RIVAS LOPEZ VICTOR GERARDO
16 VALORACIONES Y PROBLEMAS DE LA FILOSOFÍA MEXICANA CONTEMPORÁNEA	Tesis	2008	registrada		HERNANDEZ Y ALVAREZ JOSÉ DAVID

Fig. 5.18 Lista de Documentos anexos al Proyecto

Seleccionando el nombre del archivo que esta en color azul se nos mostrara una pantalla para que indiquemos la manera que queremos descargarlo.

Documentos ingresados

Titulo	Tipo	Reg.	Estado	Archivo	Director de
FUNDAMENTACIÓN DE UN FEMINISMO FILOSÓFICO EN LA OBRA DE CELIA AMORÓS	Tesis	2008	rechazada	prneba1.pdf	GARCIA AGUILARMARIA D
REFLEXIONES DESDE PAULO FREIRE	Tesis	2008	rechazada		FERNANDEZ Y DIAZRODOI
LA FILOSOFÍA DE LA VIDA ARTÍSTICA EN SAMUEL RAMOS	Tesis	2008	registrada		CABEZAS HIGAREDAANTO
TOMÁS DE MERCADO Y LA HERMENÉUTICA ANALÓGICA	Tesis	2008	registrada		CAMPOS BENITEZJUAN MA
LA FILOSOFÍA POLÍTICA DE LA DEMOCRACÍA DELIBERATIVA I					ENT.
LENCHAMIENTO: MORALIDAD SUI GÉNERIS Y PARADOJA MOR					LAR
EL CONCEPTO DE ACCIÓN EN VITA ACTIVA DE HANNAH AREN					ENT.
ILUSURACIÓN Y MITO EN DIALÉCTICA DE THEODOR. ADORNO Y					ENT.
LAS MULTIPLES DETERMINACIONES DEL YO EN DESCARTES					CAI
EL CONCEPTO DE AMOR EN SAN AGUSTÍN Y JOSÉ ORTEGA Y G					CAI
EL PAPEL DE LA INDICACIÓN FORMAL EN LAS PRIMERAS LECCIONES DE MARTÍN H	Tesis	2009	registrada		XOLOCOITZI YANEZANGEL
LA PERCEPCIÓN EN TRANSDUC. CARTER	Tesis	2007	registrada		PERCEPCION

Fig. 5.19 Descarga del Archivo relacionado al Proyecto

5.8. Acceso de Administrador

Como tercer y último rol aunque no menos importante sino todo lo contrario tenemos el Rol de Administrador que por ser más delicados por tener acceso y poder manipular el total de la información del sistema lo estamos filtrando y validando mediante una captcha como se muestra en la figura:

Clave del administrador:

Password del admin:

JINGLES

Escribe los caracteres

- [Regresar](#)

Fig. 5.20 Acceso al sistema por Rol de Administrador

5.9. Menú de Administrador

Como vemos en la siguiente figura, tenemos más opciones que los roles anteriores, y esto es por lo que es delicado entrar con este Rol, ya que damos mantenimiento a la información total de nuestro sistema.



Fig. 5.21 Menú de Administrador

5.9.1. Gestor de Administradores

La primera opción de nuestro sistema es para dar mantenimiento a los administradores, esto es para configurarlos y modificarlos y hasta poder Eliminarlos de nuestra Base de Datos

ADMINISTRADORES

Clave	Usuario	Selección
100203199	evera	<input type="radio"/>
100101511	quiroz	<input type="radio"/>
jzamora	jzamora	<input checked="" type="radio"/>

Fig. 5.22 Página que lista los Administradores del Sistema

Si elegimos la Opción de Nuevo registro, nos pasara a la página para poder dar de alta un nuevo administrador:

Fig. 5.23 Página para dar de Alta un nuevo Administrador

Pero si elegimos el botón de “Modificar” nos llevara a la página que contiene la clave y el usuario con el que ingresamos al sistema con el Rol de Administrador:

Fig. 5.24 Página para dar mantenimiento a nuestro Administrador Actual

5.9.2. Gestor de Alumnos

La segunda opción es gestionar los Alumnos que pueden ingresar al sistema, ya que no todos los alumnos de la institución podrán hacerlos y pasado algún tiempo también darles mantenimiento como es ponerlos inactivos en el sistema:

Fig. 5.25 Página de Mantenimiento de Alumnos que ingresan al Sistema

Si elegimos la opción de enviar, nos pasara a la página para que visualicemos los datos actuales del Alumno, así como poder darle mantenimiento a sus datos, como vemos en esta pantalla podemos actualizar todos los datos ya que con el Rol de Alumno teníamos datos que no pueden ser actualizables.

ZAMORA CRUZ JUAN

Identificación	
Matricula:	<input type="text" value="950012508"/>
Apellido Paterno:	<input type="text" value="ZAMORA"/>
Apellido Materno:	<input type="text" value="CRUZ"/>
Nombre:	<input type="text" value="JUAN"/>
Adscripción:	<input type="text" value="LIC EN CIENCIAS DE LA COMPUTACIÓN"/>
Estado del alumno:	<input type="text" value="ACTIVO"/>
Tutor Asignado:	<input type="text" value="EUGENIA VERA CERVANTES"/>
Contacto	
Telefono:	<input type="text" value="2222207143"/>
Telefono Celular:	<input type="text" value="5529831184"/>
Correo electrónico:	<input type="text" value="jzamora007@hotmail.com"/>
Domicilio	
Calle:	<input type="text" value="17 NORTE"/>
Colonia:	<input type="text" value="LA LOMA"/>
Estado:	<input type="text" value="PUEBLA"/>
Codigo postal:	<input type="text" value="71230"/>
Pais:	<input type="text" value="Mexico"/>

Fig. 5.26 Página de Mantenimiento de datos del Alumno

Si oprimimos el botón de Nuevo registro nos mostrara una página similar a la anterior, pero sin datos, en la cual podremos dar de alta un nuevo alumno:

Alta de alumno

Datos personales del alumno	
Matricula :	<input type="text"/>
Password del alumno:	<input type="text"/>
Repetir password del alumno:	<input type="text"/>
Apellido Paterno:	<input type="text"/>
Apellido Materno:	<input type="text"/>
Nombre:	<input type="text"/>
Adscripción:	<input type="text" value="--Adscripción--"/>
Tutor:	<input type="text" value="--Tutor--"/>
Sección:	<input type="text"/>
Año de Ingreso:	<input type="text" value="--año--"/>
Genero:	<input type="text" value="Femenino"/>
Contacto	
Telefono:	<input type="text"/>
Telefono Celular:	<input type="text"/>
Correo electrónico:	<input type="text"/>
Domicilio	
Calle:	<input type="text"/>
Colonia:	<input type="text"/>

Fig. 5.27 Página de Alta de un nuevo Alumno

Con el Rol de administrador también tenemos la opción de cambiar el password, por si algún alumno no recuerda su clave de acceso, podremos cambiarla, aunque no tenemos acceso a los password actuales.

Fig. 5.28 Página de Actualización de Password del Alumno

5.9.3. Gestor de Profesores

De la misma forma que realizamos el mantenimiento del rol anterior (Alumnos) también tenemos en el menú del Administrador el mantenimiento a Datos de los profesores, ya que con nuestro rol gestionamos los demás roles.

Fig. 5.29 Mantenimiento y Registro de Profesores

5.9.4. Gestor de Proyectos

La siguiente opción es la de Mantenimiento de Proyectos, la que a diferencia de los demás mantenimientos es que entramos con una lista, dentro de la cual nos muestra los títulos y la información más relevante del proyecto, así como el documento ya ingresado y la opción para poder subir otro archivo relacionado con el proyecto.



Fig. 5.30 Página que muestra las opciones para dar mantenimiento al proyecto

Eligiendo la opción de subir archivo nos muestra una página donde podremos dar directamente la ruta y nombre donde se encuentra el archivo que deseamos relacionar al proyecto y/o un botón con el cual podemos tener acceso a las rutas



Fig. 5.31 Página que permite cargar el archivo relacionado al Proyecto

Los botones de alta y Modificación son similares a los mantenimientos anteriores donde podemos visualizar los datos del proyecto seleccionado desde la lista, actualizar los datos del Proyecto o para dar de alta un nuevo proyecto.

Mantenimiento de Proyectos

Enviar

Datos del Proyecto

Título del Proyecto:

Patrocinador:

Patrocinadores Participantes:

Institución responsable del proyecto:

Objetivo:

Actividades a realizar:

Participantes profesores:

Participantes alumnos:

Datos de Origen

Responsable del Proyecto:

Tipo de Proyecto:

Colegiado o posgrado:

Area Académica:

Cuerpo Académico:

Fecha de inicio del Proyecto: **Día:** **Mes:** **Año:**

Fig. 5.32 Página de Visualización y Actualización de Datos del Proyecto.

5.9.5. Gestor de Documentos

Otra de las opciones que tenemos dentro de nuestro rol de Administrador es el Mantenimiento a Documentos, en el cual le podemos dar un nombre al documento, o subir documentos relacionados al proyecto:

DOCUMENTOS

Titulo	Tipo	Fecha de registro	Estado	Archivo	Insertar Archivo	Selección
FUNDAMENTACIÓN DE UN FEMINISMO FILOSÓFICO EN LA OBRA DE CELIA AMORÓS	Tesis	2008	Terminada	arrebai.pdf	• Sube archivo	<input type="radio"/>
REFLEXIONES DESDE PAULO FREIRE	Tesis	2008	Terminada		• Sube archivo	<input type="radio"/>
LA FILOSOFÍA DE LA VIDA ARTÍSTICA EN SAMUEL RAMOS	Tesis	2008	Registrada		• Sube archivo	<input type="radio"/>
TOMÁS DE MERCADO Y LA HERMENÉUTICA ANALÓGICA	Tesis	2008	Registrada		• Sube archivo	<input type="radio"/>
LA FILOSOFÍA POLÍTICA DE LA DEMOCRACÍA DELIBERATIVA DE JÜRGEN HABERMAS	Tesis	2007	Registrada		• Sube archivo	<input type="radio"/>
LENCHAMIENTO: MORALIDAD SUI GÉNERIS Y PARADOJA MORAL	Tesis	2007	Registrada		• Sube archivo	<input type="radio"/>
EL CONCEPTO DE ACCIÓN EN VITA ACTIVA DE HANNAH ARENDT	Tesis	2008	Registrada		• Sube archivo	<input type="radio"/>
ILUSTRACIÓN Y MITO EN DIALÉCTICA DE THEODOR ADORNO Y MAX HORKHEIMER	Tesis	2008	Registrada		• Sube archivo	<input type="radio"/>
LAS MÚLTIPLES DETERMINACIONES DEL YO EN DESCARTES	Tesis	2008	Registrada		• Sube archivo	<input type="radio"/>
EL CONCEPTO DE AMOR EN SAN AGUSTÍN Y JOSÉ ORTEGA Y GASSET	Tesis	2008	Registrada		• Sube archivo	<input type="radio"/>
EL PAPEL DE LA INDICACIÓN FORMAL EN LAS PRIMERAS LECCIONES DE MARTÍN H	Tesis	2009	Registrada		• Sube archivo	<input type="radio"/>
LA PERCEPCIÓN EN JEAN PAUL SARTRE	Tesis	2007	Registrada		• Sube archivo	<input type="radio"/>
LA NEUROSIS NEOGÉNA COMO UNA CONSECUENCIA DE LA ENAJENACIÓN	Tesis	2008	Registrada		• Sube archivo	<input type="radio"/>
LA TEORÍA CRÍTICA DE MAX HORKHEIMER	Tesis	2008	Registrada		• Sube archivo	<input type="radio"/>
EL CONCEPTO DE LO ROMÁNTICO EN HEGEL	Tesis	2008	Registrada		• Sube archivo	<input type="radio"/>

Fig. 5.33 Lista de Documentos Asociados al Proyecto

Que de la misma forma al seleccionar subir archivo podremos cargar un documento para asociarlo al Proyecto.



Fig. 5.34 Página para cargar el documento asociado al Proyecto

5.10. Menú de Catálogos

La primer opcion diferente de este rol está relacionada a los Catálogos del sistema, la cual es un submenu completo sobre nuestras entidades de registros que son pocas veces modificadas, pero que aun así debemos tener su mantenimiento.

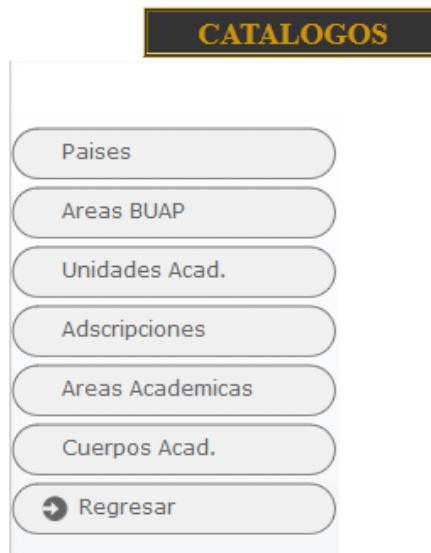


Fig. 5.35 Submenú de Catálogos del Sistema

5.10.1. Lista y Mantenimiento de Catálogos

Este catálogo es para aparecer en cada uno de nuestros mantenimientos a proyectos, alumnos profesores los diferentes países donde se ubican.

Lista de Países

Clave	nombre	Selección
1	Afganistan	<input type="radio"/>
2	Africa del Sur	<input type="radio"/>
3	Albania	<input type="radio"/>
4	Alemania	<input type="radio"/>
5	Andorra	<input type="radio"/>
6	Angola	<input type="radio"/>
7	Antigua y Barbuda	<input type="radio"/>
8	Antillas Holandesas	<input type="radio"/>
9	Arabia Saudita	<input type="radio"/>
10	Argelia	<input type="radio"/>
11	Argentina	<input type="radio"/>
12	Armenia	<input type="radio"/>

Fig. 5.36 Página de Mantenimiento a Países

Otra opcion son las Listas de Areas B.U.A.P. para asociarlas al proyecto

Lista de Areas B.U.A.P.

Clave	nombre	descripcion	Selección
1	Área de Educación y Humanidades	AREA DE EDUCACION Y HUMANIDADES DE LA COMPUTACION	<input type="radio"/>
2	Área de Ciencias Naturales y Exactas		<input type="radio"/>
3	Área de Ciencias de la Salud		<input type="radio"/>
4	Área de Ciencias Sociales y Administrativas		<input type="radio"/>
5	Área de ingeniería y Tecnología		<input type="radio"/>
6	Área de Ciencias Agropecuarias	123	<input checked="" type="radio"/>

Fig. 5.37 Página de Mantenimiento a Áreas BUAP

Tenemos la opcion del Mantenimiento de Unidades Academicas las cuales ademas de asociarlas al prooyecto podremos realizar Informes que nos detallen las estadisticas de proyectos por Unidad.

Lista de Unidades Academicas

Clave	nombre	pagina web	Selección
1	Escuela de Artes	www.artes.buap.mx	<input type="radio"/>
2	Escuela de Biología	www.biologia.buap.mx	<input type="radio"/>
3	Escuela de Cultura Física	www.licufi.buap.mx	<input type="radio"/>
4	Escuela de Ingeniería Agrohidráulica	www.buap.mx/aspirantes/licenciaturas/agro	<input type="radio"/>
5	Escuela de Ingeniería Agrohidráulica	www.buap.mx/aspirantes/licenciaturas/agroforestal	<input type="radio"/>
6	Facultad de Administración	www.administracion.buap.mx	<input type="radio"/>
7	Facultad de Arquitectura	www.arquitectura.buap.mx	<input type="radio"/>
8	Facultad de Ciencias de la Computación	www.cs.buap.mx	<input type="radio"/>

Fig. 5.38 Página de Mantenimiento de Unidades Académicas

Los demas mantenimientos se comportan de la misma forma, mostrando la lista y dandonos opciones para poder actualizar la informacion, asi como para borrar cualquier registro de nuestra Base de Datos.

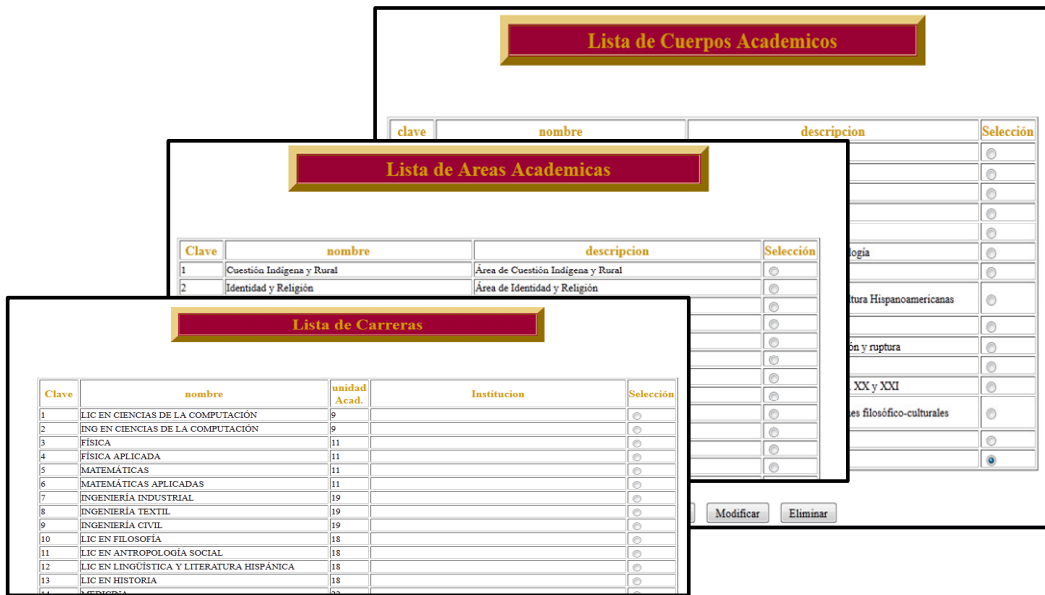


Fig. 1339 Páginas de Mantenimiento de Catálogos de Información

5.11. Menú de Reportes (Estadísticas)

Por ultima opcion tenemos el submenu de Reportes, del cual podremos ver las estadisticas de los proyectos, asi como el listado por cualquier informacion relacionada.

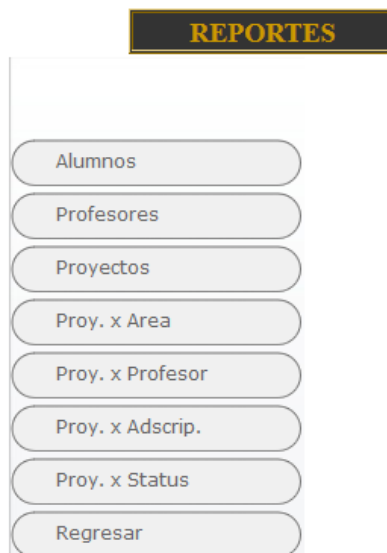


Fig. 5.40 Submenú de Reportes

5.11.1. Reporte de Alumnos, Profesores y Proyectos

En estas primeras 3 opciones tendremos que dar una condicion de filtrado, para que podamos desplegar la informacion de una forma que el usuario pueda analizarla.



Fig. 5.41 Página de Filtrado para la Búsqueda de información

Una vez que filtramos la informacion, podremos visualizar una lista con lo mas relevante informacion del reporte que hayamos solicitado (estos son a consideracion del Cliente), una vez listado, podremos imprimirlo mediante la imagen de la impresora para que nos lo muestre en un archivo con format "PDF":

Reporte de Alumnos

Matricula	Nombre	Ingreso	Status
200433560	ZAMORA ARROYO, MARCOS	2004	BAJA
200816545	ZAMORA BELLO, FELIPE	2002	ACTIVO
950012566	ZAMORA CRUZ, JUAN	1995	ACTIVO
201011189	ZAMORA FLORES, DENISSE	2010	ACTIVO
200726113	ZAMORA GONZALEZ, OBED DE JESUS	2007	ACTIVO
200907234	ZAMORA HERNANDEZ, TONALXOCHIT	2009	ACTIVO
200722179	ZAMORA LARA, CARLOS	2007	ACTIVO
200700751	ZAMORA MENDEZ, YOLOTZIN	2007	ACTIVO
200711354	ZAMORA RODRIGUEZ, ALEJANDRA	2007	ACTIVO
200923138	ZAMORA ROJAS, NICANDRO DANIEL	2009	ACTIVO
200829395	ZAMORA SANCHEZ, JOSE RODOLFO	2008	ACTIVO
200803765	ZAMORA SANTIAGO, ROSITA	2008	ACTIVO
200814917	ZAMORANO BARRANCO, GEMMA	2008	ACTIVO
200801374	ZAMORANO MORALES, SERGIO OMAR	2008	ACTIVO

Regresar

Fig. 5.42 Lista de Alumnos filtradas por Nombre

Si elegimos el boton de regresar nos llevara a la página anterior de Filtrado, para poder filtrar la informacion para obtener una menor cantidad de registros para poderlo analizar de la mejor manera.

5.11.2. Reportes Estadísticos.

Pasando las 3 opciones finales, tendremos nuestros reportes estadísticos, los cuales nos muestran cada reporte una gráfica del comportamiento de nuestro proyecto en base a sus diversos datos.

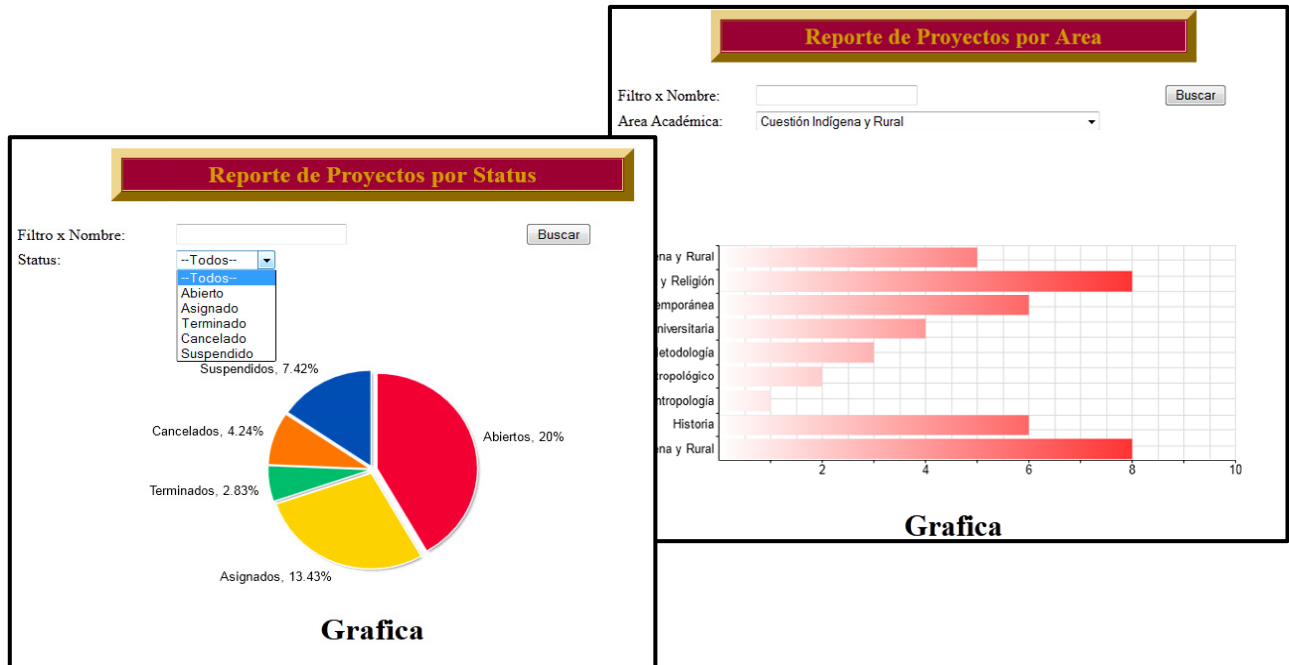


Fig. 5.43 Páginas de Reportes Estadísticos

Así una vez filtrados, podremos revisar los reportes referentes a las características del proyecto y podremos imprimirlos, de la misma forma en que imprimimos los demás listas en nuestro sistema (mediante el icono de la impresora).

Reporte de Proyectos por Status		
Status: ABIERTO		
Titulo	Responsable	Tipo
Formulario dinámico para reservar un coche de renta.	BUAP	INDIVIDUAL
Registros 1		
Status: ASIGNADO		
Titulo	Responsable	Tipo
Sistema simulador en línea de Exámenes Departamentales del área de Programación de la FCC	UTP	INDIVIDUAL
Servicio de una Agenda Electrónica en la Web	BUAP	INDIVIDUAL
Control e Inventario en línea de la Infraestructura de la FCC	BUAP	INDIVIDUAL
Reservación de espacio para un evento	UNAM	INDIVIDUAL
Registros 4		
Status: CANCELADO		
Titulo	Responsable	Tipo
Servicio Telemático de Foro de discusión no supervisado a partir de una aplicación Web	BUAP	INDIVIDUAL
Distintos Document	BUAP	INDIVIDUAL

Fig. 5.44 Página que lista los filtros registrados.

CONCLUSIONES

Basado en el proyecto actual nos damos cuenta que la base primordial de cualquier sistema son las etapas de su desarrollo, por que como se plantea; es más costoso un cambio a nivel análisis que a nivel programación de sistemas.

Como se observó en un principio, se lograron los objetivos principales basados en los requerimientos del usuario final del sistema, como es el almacenamiento de su información llevando a cabo las metodologías planteadas en el transcurso del Diplomado.

Como cualquier desarrollo de sistemas, se puede hacer mejora a cada una de la opciones planteadas, pero esto es mediante el equilibrio de Tiempo, herramientas y costos necesarios para esto, con lo que hay que plantear al usuario final, para tener en cuenta cuáles son sus siguientes necesidades y en cuales nos podemos esforzar para una de las posibilidades de robustecer el sistema, ya que cualquier característica nueva puede no ser eficaz para ciertos desarrollos aunque para otros suele ser básica.

TRABAJO FUTURO

Como observamos en las conclusiones anteriores se logró el objetivo requerido, con esto no queremos decir que el sistema no pueda mejorar, y como sabemos este punto es primordial tomarlo en cuenta para la duración de cada sistema, ya que la tendencia de cada sistema es mejorar al mismo ritmo que la tecnología o estaremos observando como en un corto tiempo parece el mismo.

Por lo anterior y como consecuencia de los avances que día con día estamos teniendo para cada uno de los desarrollos web, viéndolo desde el punto de vista de mejora continua tocamos los siguientes puntos de mejora para nuestro sistema en un tiempo futuro:

- Desarrollo para dispositivos móviles: llevándolo a plataformas como son celulares, para que cualquier alumno, profesor, o administrador pueda conectarse desde cualquier punto del planeta como lo vemos con otras páginas.
- Énfasis en el aspecto Multimedia: diseñando una interfaz más amigable donde podremos dar al usuario una mejor opción en la cual pueda visualizar ciertas ligas con las páginas o avisos de la Universidad en cuestión.
- Mejorar la Robustez: Tener una mayor despliegue de mensajes apropiados para cualquier situación error y que la aplicación pueda terminar de forma más limpia y no con mensajes del gestor de Base de Datos o del navegador en uso.
- Mejoramiento de Seguridad: Mayor seguridad en cuanto a cifrado de datos, sesiones, etc.
- Migración de Bases de datos: Migrar a una plataforma de Base de datos que podamos tener una mayor amplitud de herramientas con las cuales podamos relacionar con otros sistemas teniendo en cuenta la disponibilidad, tolerancia al fallo, Granularidad de información, Triggers de Auditoria, etc.

BIBLIOGRAFIA

- [Jal1993] Jacobson, Ivar ***El Proceso Unificado de Desarrollo***, Addison Wesley, España, 2000
- [PrR1993] Roger S. Pressman, ***Ingeniería del Software***, Mc. Graw Hill, Tercera Edición, 1993
- [Hal1994] I. T. Hawryskiewicz, ***Análisis y diseño de bases de datos***, Megabyte Noriega editores, Primera edición, 1994
- [LoK2002] Kevin Loney, Marlene Theriault, **Oracle 9i Manual del administrador, técnicas de gestión de bases de datos Oracle robustas y de alto rendimiento**, Oracle Press - Mc Graw Hill – Osborne, 2002
- [WaJ2008] John Watson, **OCA Oracle 11g, Administration**, Oracle Press Mc Graw Hill, 2008

LINK EN INTERNET

<http://www.slideshare.net/AmbarGamboa/la-importancia-de-los-sistemas-de-informacin>

<http://es.scribd.com/doc/69908114/Practico-4-PUDS>

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

<http://es.kioskea.net/contents/secu/secuintro.php3>

<http://www.slideshare.net/Financieros2008/seguridad-y-control-de-los-sistemas-de-informacin>

<http://forums.aspfree.com/code-bank-54/asp-classic-md5-hash-82164.html>

<http://snipt.org/uhjhe3>

<http://www.rgraph.net/examples/pie.html>