UNIVERSIDAD NACIONAL AUTONOMA DE NICARAGUA
UNAN-MANAGUA
RECINTO UNIVERSITARIO "RUBEN DARIO"
FACULTAD DE CIENCIAS E INGENIERIAS
DEPARTAMENTO DE COMPUTACION

TESIS MONOGRAFICA
PARA OPTAR AL TITULO DE: LICENCIATURA EN CIENCIAS DE LA COMPUTACION

TEMA DE INVESTIGACION

"Sistema de Información para el Registro Académico y Financiero del Programa de Maestría en Computación de la UNAN - Managua"

PRESENTADO POR:

- Br. Luis Miguel Martínez Olivera
- Br. Gerald Manuel Ortega Espinoza

TUTOR:
Msc. Reinerio Bermúdez Díaz

ASESOR METODOLOGICO:
Msc. Gerardo Mendoza Jiménez

MANAGUA, NICARAGUA 06 DE ABRIL DEL 2011

Contenido

Índice de figuras

Índice de tablas

Agradecimientos

Agradecemos infinitamente:

A Dios por permitirnos estar concluyendo una nueva etapa en nuestra formación profesional, sin su misericordia hubiese sido imposible haber cursado nuestros estudios universitarios ya que el nos provee la sabiduría necesaria para seguir adelante y vencer los obstáculos que se han presentado.

A todas las personas que de una u otra manera contribuyeron a la realización de esta tesis, a todos los profesores del Departamento de Computación, al Msc. Augusto Estrada Quintero por brindarnos la oportunidad de desarrollar la tesis, y estar anuente a ayudarnos en todo momento, al Msc. Reinerio Bermúdez Díaz quien fue nuestro tutor el cual dedico parte de su valioso tiempo asesorándonos y especialmente al Msc. Gerardo Mendoza Jiménez el cual siempre nos brindo los mejores consejos y su invaluable apoyo.

Los Autores

Dedicatoria

La educación es un factor indispensable para que la humildad pueda corregir los ideales de paz, libertad y justicia social.

Dedico este trabajo monográfico a mi madre Aura Elena Olivera Dávila; fuente de mi inspiración, quien me ha ayudado de forma inalcanzable para alcanzar mis metas, este trabajo es una forma de decirle que todo su sacrificio y esfuerzo no ha sido en vano.

Br. Luis Miguel Martínez Olivera.

Dedicatoria

Dedico este trabajo:

A Dios por darme la vida, por permitirme culminar esta etapa de mi vida y estar siempre presente en todo momento, gracias por tu inmensa bondad JEHOVA.

A mi madre María Elena Espinoza Baltodano a quien le debo todo lo que soy, gracias porque a pesar de las limitaciones, te has esforzado y me has brindado todo tu apoyo incondicionalmente, eres la persona a quien más admiro en la vida.

A mis Abuelos que tanto extraño Rodolfo Valentín Espinoza Quintanilla y Petronila Baltodano Espinoza (q.e.p.d), quienes me cuidaron, aconsejaron y ayudaron a complementar la educación que me dio mi madre.

A mis hermanas por ser un pilar fundamental en mi vida, y alentarme a seguir adelante para alcanzar mis metas.

Br. Gerald Manuel Ortega Espinoza

I. Resumen

Desde hace algunos años las organizaciones han reconocido que la información es el principal recurso que poseen, debido a que esta las alimenta, siendo uno de los factores más determinantes del fracaso o el éxito de las mismas.

En el presente trabajo monográfico se aborda el análisis y diseño de un sistema de información para el Registro Académico y Financiero del Programa de Maestría en Computación que ofrece el Departamento de Computación de la Facultad de Ciencias e Ingenierías de la UNAN – Managua.

En el inicio se presenta la situación actual de los mecanismos de control de pagos y expedientes de los estudiantes de la Maestría en Sistemas de Información los cuales sirvieron de base para tener una idea de la creación del sistema en mención.

A continuación se explica cada una de las metas u objetivos que se plantearon y posteriormente toda la información que hizo posible el análisis y diseño del sistema, así como los diferentes tipos de métricas (Producto, Proceso) que se pueden aplicar en el desarrollo de un sistema de información.

Así mismo, se muestran los procedimientos que se realizaron para el desarrollo del sistema de Registro Académico y Financiero del Programa de Maestría en Computación (PMCSIS), además de las herramientas que se utilizaron para la elaboración de dicho sistema (PMCSIS). Se hace una descripción del estudio de factibilidad tomando en cuenta la parte técnica, económica y operacional, así como los requerimientos necesarios para su debida implementación.

Luego se explica el funcionamiento de las diferentes opciones planteadas en el sistema para su debida aplicación cotidiana, y finalmente se presentan las herramientas (casos de usos, diagramas etc.) utilizadas para su debida implementación.

II. Introducción

Los sistemas de información y los sitios web son recursos muy valiosos para toda organización. La tendencia actual en lo que respecta al procesamiento de datos, es el desarrollo de sistemas de información debido a las ventajas que ofrece en lo que respecta a la organización y gestión de los datos.

Dentro de las necesidades de las organizaciones que manejan grandes volúmenes de información, se encuentra el mejorar la calidad y agilizar el trabajo, pero sobre todo tener una mejor seguridad, permitiendo así una administración más adecuada de los recursos.

En la presente investigación inicialmente se plantea la situación actual del manejo de los registros académicos y financieros de los estudiantes, así como el registro de docentes del Programa de Maestría en Computación, los cuales se llevan de una manera semi-automatizada.

Por lo tanto se observó la necesidad de diseñar e implementar un sistema de información que proporcione una mejor administración de los datos, así como una mejor atención a los estudiantes y docentes, agilizando los procesos de Registro Académico y Financiero de dicha maestría, de tal manera que los datos puedan estar a disposición en el momento preciso.

En la medida en que se profundice en el tema investigativo, se plantean las técnicas de análisis y diseño que permitieron la creación de este, destacando los requerimientos, herramientas de desarrollo, que fueron parte importante del proceso, las cuales contribuyeron con el desarrollo del presente tema de investigación.

III. Objetivos

Objetivo General

Implementar un sistema de información para el Registro Académico y Financiero del Programa de Maestría en Computación de la UNAN - Managua.

Objetivos Específicos

1. Desarrollar un sistema que facilite la administración del registro académico de los estudiantes de la maestría.
2. Crear un mecanismo que controle el pago de las mensualidades de los estudiantes de la maestría.
3. Diseñar una interfaz que genere los reportes académicos y financieros de los estudiantes para que sirva de apoyo a la toma de decisiones de las autoridades.
4. Diseñar un sitio Web que permita a los estudiantes consultar sus calificaciones y pagos.
5. Evaluar la calidad del sistema desarrollado aplicando métricas al producto software.

IV. Marco Teórico

En este apartado se muestra toda la información que se seleccionó de la literatura existente que sirvió de referencia para coordinar todos los conceptos y proposiciones que permitieron abordar el problema de la investigación.

4.1. Ingeniería de Requisitos (IR)

La ingeniería de requisitos ayuda a los ingenieros de software a entender mejor el problema en cuya solución trabajarán, incluye el conjunto de tareas que conducen a comprender cuál será el impacto del software sobre el negocio, qué es lo que el cliente quiere, cómo interactuarán los usuarios finales con el software.

Sommerville lo define como "la descripción de los servicios proporcionados por el sistema y sus restricciones operativas.", Pressman explica que este provee un mecanismo apropiado para entender lo que el cliente quiere, analizar las necesidades, evaluar la factibilidad, negociar una solución razonable y administrar los requisitos conforme éstos se transforman en un sistema operacional *[Sommerville, 2005] , [Pressman (b), 2006].*

Dentro de la ingeniería de requisitos se encuentra los requerimientos del usuario que no es más que declaraciones en lenguaje natural y en diagramas, de los servicios que se espera que el sistema proporcione y de las restricciones bajo las cuales debe funcionar y los requerimientos del sistema estos establecen con detalles las funciones, servicios y restricciones operativas del sistema *[Sommerville, 2005].*

4.1.1. Funciones de la Ingeniería de Requisitos

El proceso de ingeniería de requisitos se lleva a cabo a través de siete distintas funciones, algunas de estas funciones ocurren en paralelo.

- Inicio
- Obtención
- Elaboración
- Negociación
- Especificación

- Validación
- Gestión

A continuación se describen brevemente cada una de las funciones de la ingeniería de requisitos.

Inicio

La mayoría de los proyectos comienza cuando se identifica una necesidad de negocios o se descubre un nuevo mercado, los participantes de la comunidad de negocios definen un caso de negocios para la idea, tratan de identificar la amplitud y profundidad de mercado, hacen un análisis preliminar de factibilidad e identifican una descripción funcional del ámbito de proyecto *[Pressman (b), 2006].*

Al inicio los ingenieros hacen una serie de preguntas, con el objetivo de establecer una comprensión básica del problema, estas preguntas van dirigidas a el cliente.

Obtención

No es tan fácil preguntarle al cliente que es lo quiere, porque en muchos casos ni el mismo sabe que es lo que quiere, de ahí que surgen muchos problemas, tales como los que menciona Pressman *[Pressman (b), 2006]*:

- Problemas de ámbito: El límite del sistema está mal definido.
- Problemas de comprensión: El cliente no seguro por completo de que es lo necesita, no comprende del todo el dominio del problema.
- Problemas de volatilidad: Los problemas cambian conforme transcurre el tiempo.

Para poder resolver estos problemas los ingenieros deben de realizar de forma organizada la obtención de requisitos.

Elaboración

Es una acción del modelado de análisis y esta se conduce por medio de escenarios del usuario que describe la forma en que el usuario final (y otros actores) interactúan con el sistema, en esta etapa se producen una variedad de diagramas UML *[Pressman (b), 2006].*

El resultado final es un modelo de análisis que define el dominio de la información, las funciones y el comportamiento del problema.

Negociación

En muchas ocasiones hay ciertas diferencias entre lo que piden los clientes y los usuarios, entonces surgen algunos conflictos, por lo que el ingeniero debe de tratar de resolver esos conflictos por medio de la negociación.

Se pide a los clientes y a los usuarios que ordenen sus requisitos y después se discuten los problemas relacionados con la prioridad. Mediante un enfoque iterativo, los requisitos se eliminan, combinan o modifican de forma de que cada parte alcance cierto grado de satisfacción *[Pressman (b), 2006].*

Especificación

Para Pressman la especificación es el producto final que genera la ingeniería de requisitos. Sirve como base para las actividades de ingeniería de software subsecuentes. Describe la función y el desempeño de un sistema basado en computadoras y las restricciones que regirán su desarrollo *[Pressman (b), 2006].*

Validación

En esta se examina la especificación para asegurar que todos los requisitos se han especificado de manera precisa; que se han detectado las inconsistencias, omisiones, errores y que se han corregido, los productos de trabajo deben de cumplir con los estándares establecidos para el proceso, proyecto y producto *[Pressman (b), 2006].*

Gestión

La gestión de requisitos empieza por la identificación, donde cada requerimiento se asigna a un solo identificador, una vez identificados los requisitos se desarrollan las tablas de rastreabilidad. Entre las muchas tablas de rastreabilidad tenemos las siguientes *[Pressman (b), 2006]:*

- Tabla de rastreabilidad de las características
- Tabla de rastreabilidad de la fuente
- Tabla de rastreabilidad de dependencia
- Tabla de rastreabilidad del subsistema
- Tabla de rastreabilidad de la interfaz.

4.2. Sistemas

Sistema es un término que se usa comúnmente en todas partes del mundo, cuando hablamos por ejemplo de sistemas informáticos, sistemas organizacionales, sistemas gubernamentales, sistemas financieros, etc. Este término es bastante amplio y puede usarse de diferentes formas, aunque sin importar la forma en que se usen todos coinciden en que *sistema* es la unión de un conjunto de componentes, tal a como lo define Sommerville quien dice que "*Un sistema es una colección de componentes interrelacionados que trabajan conjuntamente para cumplir algún objetivo.*" *[Sommerville, 2005].*

A partir de un sistema se genera un ***subsistema*** este es la parte de otro sistema, ya que los sistemas son jerárquicos, entonces tendríamos un nodo raíz o nodo padre que sería el sistema principal, y los nodos hojas o nodos hijos serian los subsistemas, que estos últimos le brindarían información al nodo raíz.

Dentro de los sistemas se encuentran los ***Sistemas de Información*** que según Senn, depende en mayor o menor medida de un sistema organizacional. Por medio de este fluyen los datos de una persona o un departamento hacia otros y puede ser cualquier cosa, desde la comunicación interna entre los diferentes componentes de la

organización hasta sistemas de cómputos que generan reportes periódicos para varios usuarios *[Senn, 1992]*.

4.2.1. Características de los Sistemas de Información

Todo sistema de información es desarrollado para cumplir ciertos objetivos, por lo que Senn explica que para alcanzar sus objetivos este tiene que interactuar con su medio ambiente, que son todos los objetos que están fuera de la frontera de los sistemas, por lo que surgen dos tipos de sistemas: **Sistemas abiertos** estos sistemas son los que interactúan con su medio ambiente, es decir que producen entradas y salidas, todos los sistemas actuales son abiertos *[Senn, 1992]*.

En contraste, aquellos sistemas que no interactúan con su medio ambiente se conocen como **Sistemas Cerrados**, estos existen solo como un concepto, aunque muy importante.

4.2.2. Categorías de los sistemas de Información

Los sistemas de información se pueden clasificar según la función que desempeñan, dentro de los cuales están los siguientes:

- Sistemas de Procesamiento de Transacciones (TPS)
- Sistemas de Automatización de la Oficina (OAS)
- Sistemas de Trabajo del Conocimiento (KWS)
- Sistemas de Información Gerencial (MIS)
- Sistemas de Apoyo a la Toma de Decisiones (DSS)
- Sistemas Expertos e Inteligencia Artificial (SE)
- Sistemas de Apoyo a la Toma de Decisiones en Grupo (GDSS)
- Sistemas de Trabajo Colaborativo Apoyados por Computadora (CSCWS)
- Sistemas de Apoyo a Ejecutivos (ESS)
- Sistemas de información Geográfica (GIS)

A continuación se define el tipo de categoría relacionada con el sistema que se desarrolló.

4.2.3. Sistemas Información Gerencial (MIS)

Son sistemas de información computarizados cuyo propósito es contribuir a la correcta interacción entre los usuarios y las computadoras debido a que requieren que los usuarios, el software y el hardware funcionen de manera coordinada *[Kendall (b), 2005].*

Para acceder a la información los usuarios de un MIS comparten una base de datos en común. Ésta almacena datos y modelos que ayudan al usuario a interpretar y aplicar los datos, estos sistemas producen información que se emplea en la toma de decisiones, también pueden contribuir a unificar algunas de las funciones de información computarizadas de una empresa, a pesar de que no exista como una estructura individual en ninguna parte de esta *[Kendall (b), 2005].*

Los MIS incluyen los TPS, los cuales se encargan de procesar grandes cantidades de datos relacionadas con las operaciones comunes que tiene una empresa, Ahora bien, a diferencia de los TPS los MIS dan apoyo a un espectro más amplio de tareas organizacionales, incluyendo el análisis de decisiones y la toma de decisiones, el cual los TPS no tienen espectro tan amplio, porque estos no producen información que ayude a la toma de decisiones.

Laudon explica que los MIS resumen y preparan informes acerca de las operaciones básicas de la compañía, en contraste a los TPS que por lo regular presentan informes largos, en fechas preestablecidas *[Laudon, 2002].*

Por lo general los MIS sirven para los administradores interesados en informes o resultados no periódicos, Laudon los cataloga como “poco flexibles y con una capacidad analítica reducida”, ya que utilizan rutinas sencillas con resúmenes o comparaciones, sin tener que utilizar un modelo matemático avanzado.

Niveles de Toma de Decisiones

La toma de decisiones se puede clasificar según su nivel en la organización que corresponde a los niveles estratégicos, de administración, de conocimientos y operativo de la organización *[Laudon, 2002]:*

- La toma de decisiones estratégica determina los objetivos, los recursos y las políticas de la organización.
- La toma de decisiones para control gerencial se ocupa principalmente de la eficiencia y eficacia con que se utilizan los recursos y el desempeño de las utilidades operativas.
- La toma de decisiones en el nivel de conocimientos se ocupa de evaluar ideas nuevas de productos y servicios, formas de comunicar conocimientos nuevos y de distribuir información por toda la organización.
- La toma de decisiones para el control operativo determina la forma de llevar a cabo las tareas específicas propuestas por quienes toman decisiones estratégicas y por los administradores de nivel medio.

4.3. Análisis y Diseño de Sistemas

El desarrollo de sistemas puede considerarse, en general, formado por dos grandes componentes: el análisis y el diseño de sistemas.

El diseño de un sistema de información produce los detalles que establecen la forma en la que el sistema cumplirá con los requerimientos identificados durante la fase del análisis.

Los especialistas en sistemas se refieren, con frecuencia, a esta etapa como ***diseño lógico*** en contraste con el desarrollo del software, a la que se denominan ***diseño físico*** *[Senn, 1992].*

Dentro de las organizaciones el análisis y diseño de sistemas se refiere al proceso de examinar la situación de una empresa con el propósito de mejorarla con métodos y procedimientos adecuados.

En el Análisis y Diseño de sistemas se encuentran los ***orientados a objetos*** estos pueden ofrecer un enfoque que habilite los métodos lógicos, rápidos y minuciosos necesarios para crear nuevos sistemas en respuesta al cambiante entorno de un negocio, las técnicas orientadas a objetos son adecuadas en situaciones en las que los sistemas de información requieren de mantenimiento, adaptación y rediseños continuos *[Kendall (6), 2005]*.

Las tecnologías de objetos proporcionan grandes ventajas por medio de sus técnicas como por ejemplo la reutilización, la cual permite un desarrollo de software más rápido y de calidad, debido a que no es necesario describir un objeto desde el principio cada vez que se necesite. El software orientado a objetos es más fácil de mantener debido a su estructura inherentemente descompuesta.

Puede ser descrita como una disciplina de desarrollo y modelaje de software que hace fácil la construcción de los sistemas complejos. La orientación a objetos permite una representación directa de modelos del mundo real. El resultado es que, las transformaciones radicales de los requerimientos del sistema (definidos en términos de los usuarios) hacia especificaciones del sistema (definidas en términos computacionales) son enormemente reducidas.

Los tres aspectos fundamentales del paradigma de orientación a objetos son:

- Análisis Orientado a Objeto
- Diseño Orientado a Objeto
- Programación Orientado a Objeto

4.3.1. Análisis Orientado a Objetos

Los métodos de análisis orientado a objetos permiten al desarrollador modelar un problema a través de la representación de objetos, atributos y operaciones como las componentes primarias del modelado. Una amplia variedad de métodos de análisis a objetos han sido propuestos, pero todos poseen un conjunto de características comunes *[Pressman (a), 1998]*:

- Representación de clases o jerarquías de clases,
- Creación de modelos objeto-relación,
- Derivación de modelos objetos-comportamiento

La diferencia entre el Análisis Orientado a Objeto y el análisis función/datos es considerable. Mientras que métodos de análisis de función/datos comienzan por considerar el comportamiento del sistema y/o consideran los datos separadamente, el Análisis Orientado a Objeto puede ser caracterizado como una iteración entre analizar el comportamiento y la información del sistema.

El Análisis Orientado a Objeto involucra, en orden, las siguientes operaciones:

- Descubrir los objetos
- Organizar los objetos
- Identificar como los objetos interactúan
- Definir las operaciones de los objetos
- Definir los objetos internamente

4.3.2. Diseño Orientado a Objetos

"El diseño orientado a objetos transforma el modelo de análisis creado, usando el análisis orientado a objetos en un modelo de diseño que sirve para la construcción del software", En pocas palabras lo que Pressman quiere decir es que el diseño orientado a objetos lo que hace es traducir el análisis orientado a objetos del mundo real en un modelo especifico que pueda ser desarrollo en software *[Pressman (a), 1998].*

La diferencia del diseño orientado a objetos con los métodos convencionales radica en que este organiza en módulos los componentes principales de un sistema los cuales son denominados subsistemas.

Como en el análisis orientado a objetos existen varios métodos diferentes de diseño orientado a objeto. Aunque cada uno difiere del otro a partir de sus componentes, todos están de acuerdo con la pirámide de diseño y todos enfocan el proceso de diseño a través de dos niveles de abstracción: Diseño del sistema y los subsistemas, y diseños de objetos individuales *[Pressman (a), 1998].*

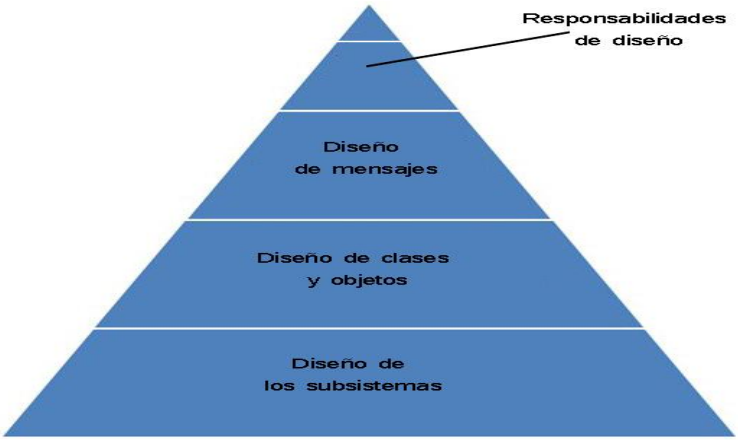


Figura 1- Diseño orientado a objeto en pirámide

4.3.3. Programación Orientada a Objetos (POO)

Consiste en programar usando objetos. El objetivo de la programación orientada a objetos es mejorar la productividad de los programadores facilitándoles reutilizar y extender el software ya existente, de modo que se reduzcan los costes de desarrollo y mantenimiento del software.

El mayor poder de la programación orientada a objetos motiva la reutilización y que usualmente es más fácil de comprender y mantener que en otros tipos de programación.

Básicamente el lenguaje Orientado a Objetos debe soportar lo siguiente:

- Objetos
- Clases
- Mensajes
- Encapsulamiento
- Herencia
- Polimorfismo

Objetos: Kendall lo define como "Una representación en computadora de alguna cosa o evento del mundo real". Ahora bien hay autores que definen un objeto como "una instancia de una clase" y no quiere decir que estén errados en la definición sino que el termino instancia y objeto son sinónimos, en la programación orientada a objetos estos términos son lo mismo *[Kendall (a), 1997].*

Clases: Una clase equivale a la generalización de un tipo específico de objetos, pero cada objeto que se construya de esa clase tendrá sus propios datos. Como ejemplo podemos pensar en un molde para hacer flanes; el molde es la clase y los flanes los objetos, si disponemos de un molde de un litro para hacer flanes de vainillas (Ingredientes: leche, vainilla, azúcar, etc.), el molde agrupa las propiedades comunes a todos los flanes de vainilla, pero no todos los flanes tienen por qué tener la misma cantidad de cada ingrediente *[Ceballos, 2006].*

Mensajes: Según Kendall, se puede definir mensajes como la información que se envía de un objeto a otro. Estos mensajes no son de forma libre en ningún sentido, sino que las clases son programadas cuidadosamente para recibir y enviar información *[Kendall (a), 1997].*

En la programación orientada a objetos un mensaje está asociado con un método, de tal forma que cuando un objeto recibe un mensaje la respuesta a ese mensaje es ejecutar el método asociado Ceballos *[Ceballos, 2006].*

Encapsulamiento: La información acerca de un objeto esta encapsulada por su comportamiento. Esto significa que hay datos que no pueden ser cambiados por procesos externos, sino que un objeto es el que puede pedir o decir que cambie sus propios datos con un mensaje *[Kendall (a), 1997].*

Herencia: La herencia es una de las cualidades más importantes de la programación orientada a objetos, porque permite que una clase herede los atributos y métodos de otra clase. Así se garantiza la reutilización de código *[Ceballos, 2006].*

Polimorfismo: La palabra “polimorfismo” significa “la facultad de asumir muchas formas”, refiriéndose a la facultad de llamar a las distintas versiones que adopta un método definido en una superclase y redefinido en sus clases derivadas, utilizando exactamente el mismo medio de acceso: una referencia a la superclase *[Ceballos, 2006].*

4.4. Modelos prescriptivos de proceso (Procesos de desarrollo de software, Paradigmas de la Ing. Del software).

Pressman dice que "Los modelos prescriptivos de procesos definen un conjunto distinto de actividades, acciones, tareas, fundamentos y productos de trabajo que se requieren para desarrollar software de alta calidad. Estos modelos de software no son perfectos, pero proporcionan una guía útil para el trabajo de la ingeniería del software" *[Pressman (6), 2006].*

A estos modelos se les llama "*prescriptivos"* por que prescriben un conjunto de elementos del proceso tales como: actividades del marco de trabajo, acciones de ingeniería del software, tareas, productos del software, aseguramiento de la calidad del software y mecanismos de control del cambio para cada proyecto.

Dentro de los modelos prescriptivos se encuentran los siguientes:

- Modelo en cascada (SDLC)
- Modelos de procesos incrementales
  - ✓ Modelo incremental
  - ✓ Modelo DRA
- Modelos de procesos evolutivos
  - ✓ Construcción de prototipos
  - ✓ Modelo en espiral
  - ✓ Modelo de desarrollo concurrente
- Modelos especializados de procesos
  - ✓ Desarrollo basado en componentes
  - ✓ Modelo de métodos formales
  - ✓ Desarrollo del software orientado a aspectos
- Proceso unificado
- Programación Extrema (XP)

A continuación se define el modelo de proceso con el que se desarrolló el sistema.

4.5. Modelo Incremental

El modelo incremental combina elementos del modelo en cascada aplicado en forma iterativa, va aplicando secuencias lineales de manera escalonada conforme avanza el tiempo en el calendario. Por cada secuencia lineal este produce incrementos en el software.

En este modelo los clientes identifican los servicios que brindará el sistema, identificando así cuales servicios son más importantes y cuales menos, entonces es donde el sistema se divide en varios incrementos en donde en cada incremento se debe proporcionar un producto funcional.

"Al utilizar el modelo incremental el primer incremento es un producto esencial, es decir que este incremento incorpora los requisitos básicos, pero varias características suplementarias (Algunas conocidas, otras no) no son incorporadas" *[Pressman (b), 2006].*

Desde el primer incremento es un producto operacional, este producto se le entrega al cliente para que lo someta a una evaluación del cual del resultado de esa evaluación saldrá el plan para desarrollar el siguiente incremento, este proceso se repite hasta que se haya entregado el producto completo.

El desarrollo incremental es útil sobre todo cuando el personal necesario para una implementación completa no está disponible, así los primeros incrementos se pueden implementar con menos personas, conforme el producto esencial es bien recibido se agrega más personal para implementar el siguiente incremento. Además los incrementos se pueden planear para manejar los riesgos técnicos, como por ejemplo que se necesite un equipo en especial cuya fecha de entrega es incierta, entonces se pueden planificar incrementos que no hagan uso de este tipo de equipos mientras no esté disponible y así se evita retrasos en la entrega del producto.

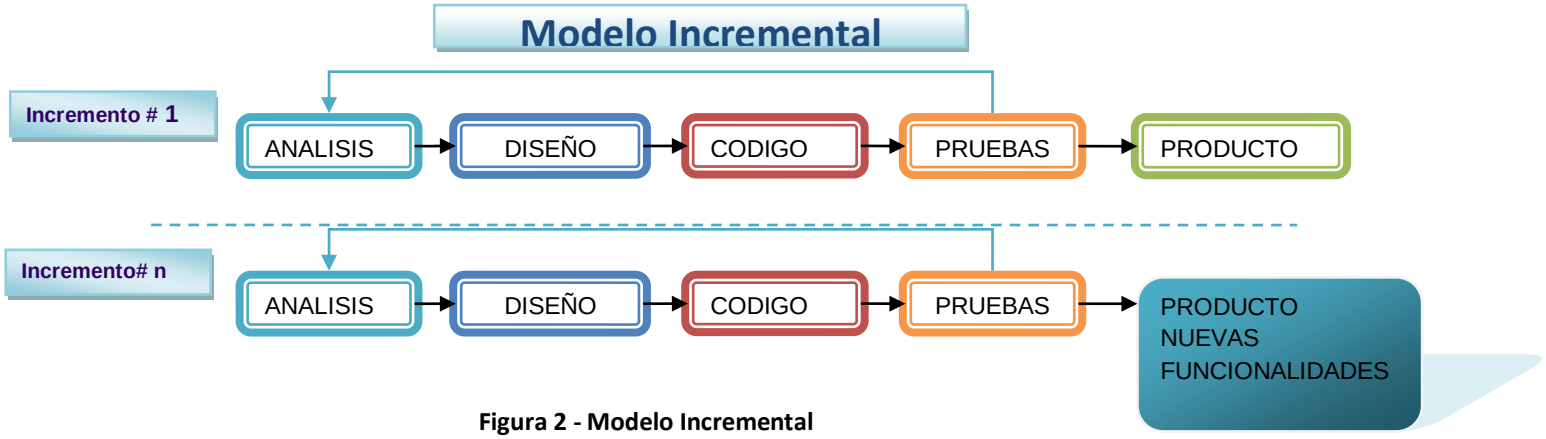


Figura 2 - Modelo Incremental

Sommerville menciona algunas de las ventajas que ofrece el modelo incremental, entre las cuales están las siguientes *[Sommerville, 2005]:*

- Los clientes no tienen que esperar hasta que el sistema completo se entregue para sacar provecho de él.
- Los clientes pueden utilizar los incrementos iníciales como prototipos y obtener experiencia sobre los requerimientos de los incrementos posteriores del sistema.
- Existe un bajo riesgo de fallo total del proyecto. Aunque se pueden encontrar problemas en algunos incrementos, lo normal es que el sistema se entregue de forma satisfactoria.
- Puesto a que los servicios de más alta calidad se entregan de primero y los incrementos posteriores se integran en ellos, es inevitable que los servicios más importantes del sistema sean a los que se le hagan más pruebas.

Sin embargo a pesar de ser un modelo de proceso de desarrollo excelente tiene sus desventajas, a como Sommerville explica los incrementos deben de ser relativamente pequeños (No más de 20,000 líneas) y cada uno debe de entregarse con alguna funcionalidad del sistema. Puede ser difícil adaptar los requerimientos del cliente a incrementos de tamaño apropiado, puede ser difícil identificar los recursos comunes que requieren todos los incrementos *[Sommerville, 2005].*

4.6. Prototipos

La elaboración de prototipos de un sistema de información es una técnica valiosa para la recopilación rápida de información específica a cerca de los requerimientos a cerca de los requerimientos de los usuarios.

Por medio del prototipo se desea saber cuál es la reacción de los usuarios, todas estas reacciones se recopilan, para verificar si el prototipo satisface sus necesidades o no, y así establecer cuáles serán las prioridades.

A continuación se define brevemente las clases de prototipos que existen, incluyendo el prototipo de características seleccionadas, que es con el que se trabajó en el sistema.

Clases de prototipos

Prototipo corregido (Parchado): Este tipo de prototipo tiene que ver con la construcción de un sistema que funciona pero se corrige simultáneamente. Este prototipo se desarrolla rápidamente con el objetivo de ser funcionales pero no eficaces *[Kendall (b), 2005].*

Prototipo no Funcional: Es un modelo no funcional a escala, configurado para probar ciertos aspectos del diseño, este podría producirse cuando la codificación requerida por las aplicaciones es demasiado extensa para incluirse en el prototipo pero cuando se puede conseguir una idea útil del sistema a través de la elaboración de un prototipo de la entrada y la salida. Sin embargo, aun se podrían tomar algunas decisiones sobre la utilidad del sistema como base en la entrada y la salida en el prototipo *[Kendall (b), 2005].*

Prototipo primero de una serie: involucra la creación de un primer modelo a escala completa de un sistema, con frecuencia llamado piloto. Este tipo de elaboración de prototipos es útil cuando se planean muchas instalaciones del mismo sistema de información, permitiendo a los usuarios experimentar la interacción real con el nuevo sistema, pero minimiza el costo de superar cualquier problema que se presente *[Kendall (b), 2005].*

Prototipo de características seleccionadas: involucra la creación de un modelo funcional que incluya alguna pero no todas, de las características que tendrá el sistema final *[Kendall (b), 2005].*

Cuando se recurre a este tipo de elaboración de prototipos, el sistema se contempla por módulos de forma que si las características que se incluyen en los prototipos se evalúan exitosamente, se pueden incorporar en el sistema final más grande sin necesidad de realizar demasiado esfuerzo en la interacción *[Kendall (b), 2005].*

4.7. Modelo Relacional

Las ideas fundamentales del modelo de datos relacional aparecieron en forma pública, en un artículo de Edward Codd de 1970, desde ese momento el modelo relacional se ha establecido como el principal modelo de datos para las aplicaciones de procesamiento de datos, debido a su simplicidad que facilita el trabajo del programador en comparación con otros modelos como el de red y jerárquico.

Una base de datos relacional consiste en un conjunto de tablas, a cada una de las cuales se les asigna un nombre exclusivo, donde cada fila de la tabla representa una relación entre un conjunto de valores, dado que cada tabla es un conjunto de dichas relaciones, hay una fuerte correspondencia entre el concepto de tabla y de relación, por tal razón es que el eje principal del modelo relacional reposa sobre el concepto de relación *[Silberschatz, 2002].*

4.7.1. Lenguaje de Consultas SQL (Base de Datos)

SQL fue desarrollado por IBM en su versión original en el Laboratorio de Investigación de San José, California, originalmente denominado Sequel, como parte del proyecto System R, a principios de 1970.

El lenguaje Sequel se basaba en el lenguaje SQUARE que utilizaba notaciones matemáticas y es el que introducía el concepto de bloque de *selección/proyección,* desde entonces ha evolucionado y su nombre ha pasado a ser SQL (Structured Query Language, Lenguaje estructurado de consultas).

Actualmente, numerosos productos son compatibles con el lenguaje SQL, este se ha establecido como el lenguaje estándar de bases de datos relacionales.

4.7.1.1. Configuración y Entorno de SQL

El SQL es un lenguaje relacional de consultas, constituido de tres componentes principales *[González, 1996]:*

- Un lenguaje de *control* de base de datos
- Un lenguaje de *descripción* de datos
- Un lenguaje de *manipulación* de datos

Catálogo de la base de datos

Es un conjunto de tablas es decir, es a su vez una base de datos, que contiene información de las tablas, vistas, índices, así como derechos de accesos a la base de datos.

Entre la información que contiene un catalogo se tiene el nombre, tamaño, tipo y valores validos en cada columna, además de la conformación de cómo se accedan a las tablas y las vistas *[González, 1996].*

Optimizador de consultas

Se encarga de escoger el mejor camino de acceso a los datos solicitados por una consulta del tipo SQL, con el optimizador lo que se persigue es minimizar el número de páginas que se deben intercambiar entre la memoria secundaria y la memoria principal *[González, 1996].*

Ejecución de una cláusula en SQL

En primer lugar, se tiene una fase de análisis y verificación, en donde se verifica si una cláusula se ha formulado de forma correcta. En segundo lugar se tiene la etapa de optimización, en donde se usan estadísticas de almacenamiento de los daos por medio del catalogo. Así se examinan varios posibles caminos de acceso a los datos, se calcula el costo de cada uno y se escoge el mejor. Finalmente se genera un plan para la ejecución de la cláusula.

Con el lenguaje SQL se pueden definir, suprimir y modificar bases de datos, tablas, índices y vistas.

Por ejemplo para definir una base de datos en SQL desde el QUERY ANALYZER (Analizador de consultas) solo escribimos el siguiente comando:

Create database nombre_base_de_datos;

4.7.1.2. Definición de la Base de Datos

Una vez que se ha definido una base de datos, el SABD (Sistema Administrador de Bases de Datos) asigna un espacio en disco y será en este espacio donde se podrán definir las tablas que conformaran la base de datos.

La definición de las tablas, y que serán llamadas tablas de base para diferenciarlas de las vistas o tablas derivadas, se hace por medio de la siguiente cláusula:

Create table nombre_tabla
(nombre-columna 1 tipo-dato1,
nombre-columna n tipo-dato n);

A modo de referencia, se usaran las características de algunos de los tipos datos definidos en SQL *[González, 1996]:*

- **char (n):** una palabra de n caracteres, con n ≤ 254
- **varchar:** para almacenar palabras de más de 254 caracteres
- **decimal(x, y)**: un numero decimal a lo sumo 15 dígitos, en donde y representa el numero de dígitos decimales y x el número total de dígitos.
- **integer:** un numero entero entre -2147483648 y +2147483647
- **smallint:** un numero entero entre -32768 y +32767
- **date:** una fecha *mm/dd/aa.*

Una vez que se han definido las tablas de la base de datos, es posible que se desee cambiar la estructura de alguna de ellas, es decir *[González, 1996]:*

- adicionar una nueva columna,
- suprimir una columna existente,
- modificar el tipo de datos de una columna existente.

Por ello se utilizan las siguientes tablas alter table, según se muestra a continuación *[González, 1996]:*

Alter table nombre_tabla **add**
(nombre-columna 1 tipo-dato 1,
nombre-columna n tipo-dato n);

Alter table nombre_tabla **drop**
(nombre-columna 1,
nombre-columna n);

Alter table nombre_tabla **modify**
(nombre-columna 1 nuevotipo1,
nombre-columna n nuevo tipo n);

Por otra parte si se desea suprimir una tabla de la base de datos, se utiliza la siguiente clausula *[González, 1996]:*

Drop table nombre_tabla;

Una vez que se ha definido la base de datos, con sus tablas y columnas, se procede a la carga de las tuplas usando la cláusula insert *[González, 1996]:*

Insert into nombre_tabla
[(Lista de columnas)]
Values (lista de valores);

Una vez que en la tabla de base se han almacenado los datos, se puede desear modificar su contenido. Para ello, se utiliza la siguiente cláusula: *[González, 1996]*

Update nombre-tabla **set**
Columna 1= expresión 1,
…
columna n = expresión n
[**Where** (condición de búsqueda)];

Por su parte si se desean suprimir algunas tuplas de una tabla se utiliza la clausula siguiente *[González, 1996]:*

Delete from nombre_tabla
[Where criterio de selección];

4.7.1.3. Manipulación de la Base de Datos

La forma general de la extracción de datos de una base de datos, vía el lenguaje relacional SQL, puede establecerse de la siguiente forma *[González, 1996]*:

- **select** lista de expresiones escalares del resultado - lista de atributos-
- **from** lista de los nombres de las relaciones involucradas
- **where** predicado de calificación sobre las tuplas-condiciones
- **group by** columna que sirve para particionar -lista de atributos-
- **having** predicado de calificación sobre las particiones -condición-
- **union** permite hacer la unión de dos bloques select
- **order by** columna retenida de la lista de select para un ordenamiento

Dentro de una expresión where se pueden utilizar los siguientes operadores *[González, 1996]:*

- de comparación: =, <, >, <=, >=, <>,
- de conjuntos: **union** –unión-, **intersect** -intersección- y **except** -diferencia-,
- de pertenencia de conjuntos: **in** y **not in,**
- de comparación de conjuntos: **contains** y **not contain** y
- las conectivas lógicas: **and, or y not.**

Operadores conjuntistas

- unión
- intersect
- except

Proyección

Select x1, x2,…, xn
From r;

En donde **r** representa una relación y x representa a los atributos.

Selección

Select *
From r
Where e;

En donde "*r*" representa una relación y "*e*" representa una condición. En vez de escribir como salida de consulta todos los atributos de una tabla, el SQL permite el uso de un asterisco.

Join

Select r *, s *
From r, s
Where $r.a_i = s.b_j$;

En donde "*r*" y "*s*" son relaciones, en este caso "r.a_i" significa que "a_i" es una variable atributo que recorre "*r*".

División

Con respecto al operador división y su definición, se debe recordar que lo que se busca es traducir el cuantificador universal de la lógica de predicados. La división se hace sobre dos relaciones R(X, Y) y S (Y); su resultado es el siguiente *[González, 1996]*:

$$R \div S = \{t \in R\ [X—Y] / \{t\} \times S \subseteq R\}$$

4.7.2. Normalización

El proceso de normalización es un método propio del modelo relacional y consiste en descomponer las relaciones originales en otras más pequeñas con el fin de eliminar una serie de anomalías de almacenamiento y manipulación que se pueden dar en las relaciones iníciales y que conformarían la futura base de datos relacional, así las relaciones además de ser más simples y estables, son más fácil de mantener que otras estructuras de datos.

Entre los beneficios que se pueden dar en una base de datos correctamente normalizada se encuentran los siguientes *[González, 1996]:*

- Reducir los problemas asociados con la supresión e inserción de tuplas.
- Reducir el tiempo asociado con modificaciones de las tuplas.
- Identificar problemas potenciales que pueden requerir un análisis adicional
- Mejorar la información para la toma de decisión referente a la organización física de los datos.

El proceso de normalización se compone de una serie de seis etapas llamadas formas normales, a continuación se definirán cada una de ellas:

4.7.2.1. Primera Forma Normal (1FN)

Se refiere a una representación de una relación en la cual los atributos son diferentes y los valores de cada uno de esos atributos son componentes atómicos, en esta 1FN lo que se busca es evitar los grupos repetitivos *[González, 1996].*

Dentro de esta se encuentran las dependencias multi-valuadas que pueden establecer asociaciones entre varios atributos de una misma relación. Así, los valores de algunos atributos en una relación pueden determinar en forma única, el conocimiento de los valores de otros atributos de la misma relación. Estas dependencias traducen reglas semánticas que, como se sabe, son muy comunes en el montaje del mundo real *[González, 1996].*

Axiomas de Inferencia de Armstrong

Sean X, Y, Z subconjuntos de atributos de una relación R en donde se verifican las dependencias funcionales X→Y y Y→Z. Entonces, las siguientes reglas se cumplen *[González, 1996]:*

Reflexividad	$X \rightarrow X$ se verifica siempre
Aumento	$X \rightarrow Y \Rightarrow X \cup Z \rightarrow Y$
Transitividad	$\{ X \rightarrow Y , Y \rightarrow Z \} \Rightarrow X \rightarrow Z$

A partir de este conjunto de reglas se pueden inferir otras como las siguientes. Sean X, Y, Z, y W subconjuntos de atributos de una relación R. Entonces se verifican las siguientes relaciones *[González, 1996]:*

Unión	$\{ X \rightarrow Y \text{ y } X \rightarrow Z \} \Rightarrow X \rightarrow Y \cup Z$
Descomposición	$X \rightarrow Y \Rightarrow X \rightarrow Z$ con $Z \subseteq Y$
Pseudo-transitividad	$\{ X \rightarrow Y \text{ y } Y \cup Z \rightarrow W\} \Rightarrow X \cup Z \rightarrow W$

Cerradura

Es el conjunto de dependencias funcionales que se pueden derivar de un conjunto de dependencias funcionales F. se denota por F^+, es decir *[González, 1996]*

$$F^+ = \{ f \ / F \vdash f\}$$

Saturación

Sea F un conjunto de dependencias funcionales. Se le llama *saturación* de un conjunto X con respecto a F, y se denota por X^+, al conjunto de atributos que son determinados por las dependencias funcionales que se derivan de F *[González, 1996].*

Teorema de descomposición

Sea un esquema de relación R (X, Y, Z), con X, Y y Z conjuntos de atributos de R, tal que la dependencia funcional X→Y se verifica en R. Entonces la relación R se descompone en las relaciones R1=R[X, Y] y R2=R[X, Z], R=R1 * R2 *[González, 1996].*

4.7.2.2. Segunda Forma Normal (2FN)

Para comprender lo que es la 2FN debemos conocer el concepto de dependencia parcial, González explica que sea R un esquema de relación, X una llave de R y A un atributo no llave. Se dice que el atributo A depende parcialmente de X si se verifica que Y→ A, en donde Y es un subconjunto propio de la llave X *[González, 1996].*

Una relación R se dice que se encuentra en segunda forma normal (2FN), si se encuentra en 1FN y si ningún atributo no llave depende parcialmente de la llave primaria.

4.7.2.3. Tercera Forma Normal (3FN)

González explica que sea R (X, Y, Z) un esquema de relación, en donde X, Y, Z son subconjuntos de atributos. Se dice que Z es transitivamente dependiente de X si existe Y tal que se dan las siguientes condiciones:

- Se verifica $X \rightarrow Y$
- No se verifica $Y \rightarrow X$
- Se verifica $Y \rightarrow Z$

Se dice que una relación R se encuentra en tercera forma normal (3FN) si se encuentra en 2FN y no existe una dependencia transitiva entre atributos no llave *[González, 1996].*

4.7.2.4. Tercera Forma Normal Boyce-Codd (3FNBC)

Una relación se encuentra en tercera forma normal Boyce-Codd (3FNBC) si todos los atributos son determinados solo por llaves, es decir, si cada vez que $X \rightarrow A$, $A \notin X$, se verifica en R, entonces X contiene una llave en R *[González, 1996].*

4.7.2.5. Dependencias Multivaluadas y la Cuarta Forma Normal (4FN)

Según González dice que una dependencia multivaluada existe entre dos conjuntos de atributos X y Y, y se denota por

$$X \rightarrow\rightarrow Y$$

Si solo el conocimiento de X, e independientemente de otros atributos, determina un conjunto de valores relativos a Y.

Entonces se dice que una relación <u>R</u> (X, Y, Z) se encuentra en 4FN si se encuentra en 3FNBC y si cada vez que una dependencia multivaluada $X \rightarrow\rightarrow Y$ se verifica, entonces X contiene a una llave de <u>R</u> *[González, 1996].*

4.7.2.6. Quinta Formal Normal (5FN)

Se refiere a las llamadas Dependencias Producto que garantizan descomposición de una relación en tres o más relaciones, manteniendo el contenido original y con menor redundancia *[González, 1996].*

Una relación R se dice que se encuentra en quinta forma normal (5FN) si cada dependencia producto * [X1] [X2]... [Xn] de R esta inducida por las llaves candidatas de R, es decir, cada Xi contiene una llave candidata de R *[González, 1996].*

4.8. UML (Lenguaje Modelado Unificado)

UML, por sus siglas en inglés, (Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Este permite modelar construir, y documentar los elementos que forman un sistema de software orientado a objetos.

UML es un lenguaje estándar que sirve para escribir los planos del software, puede utilizarse para visualizar, especificar, construir y documentar todos los artefactos que componen un sistema con gran cantidad de software. UML puede usarse para modelar desde sistemas de información hasta aplicaciones distribuidas basadas en Web, pasando por sistemas empotrados de tiempo real *[Alarcón, 2000].*

Es importante resaltar que UML es un "lenguaje" para especificar y no para describir métodos o procesos. Se utiliza para definir un sistema de software, 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. Se puede aplicar en una gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado Racional), pero no especifica en sí mismo qué metodología o proceso usar.

El lenguaje UML se compone de tres elementos básicos: Bloques de construcción, las reglas y algunos mecanismos comunes; los bloque de construcción a su vez se dividen en tres partes: Elementos, diagramas y relaciones.

Las relaciones, a su vez se dividen en: relaciones de dependencia, relaciones de asociación, relaciones de generalización y relaciones de realización.

Existen 4 tipos de elementos en UML, dependiendo del uso que se haga de ellos: Elementos estructurales, elementos de comportamiento, elementos de agrupación y elementos de anotación *[Alarcón, 2000]*.

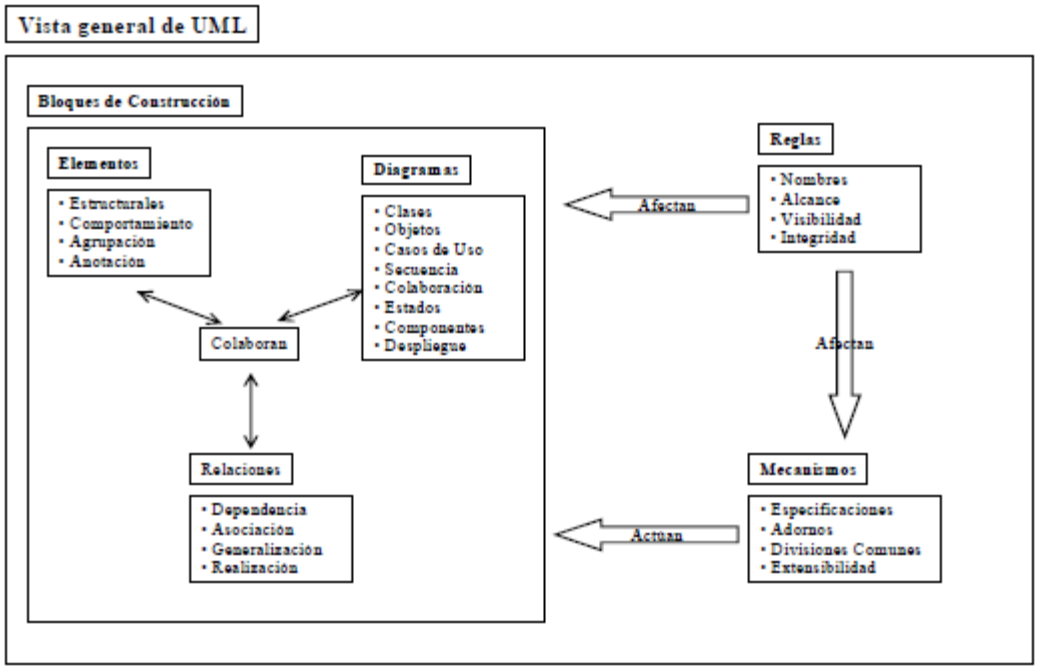


Figura 3- Vista general de los elementos de UML

A continuación se describirán los 4 tipos de elementos de UML que se mencionó en el párrafo anterior:

4.8.1. Elementos Estructurales

Los elementos estructurales en UML, en su mayoría, son las partes estáticas del modelo y representan cosas que no son conceptuales o materiales.

Clases: Una clase es una descripción de un conjunto de objetos que comparten los mismos atributos, operaciones, relaciones y semántica. Una clase implementa una o más interfaces *[Alarcón, 2000].*

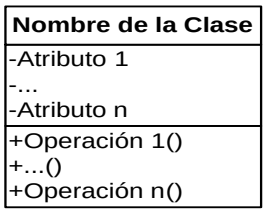


Figura 4- Representación gráfica de una clase

Interfaz: Es una colección de operaciones que especifican un servicio de una determinada clase o componente, la interfaz es la que describe el comportamiento visible externo de un elemento, además describe un conjunto de especificaciones de operaciones pero nunca su implementación *[Alarcón, 2000].*

Figura 5- Representación gráfica de una interfaz

Colaboración: Define una interacción y es una sociedad de roles y otros elementos que colaboran para proporcionar un comportamiento cooperativo mayor que la suma de los comportamientos de sus elementos. Una misma clase puede participar en diferentes colaboraciones *[Alarcón, 2000].*

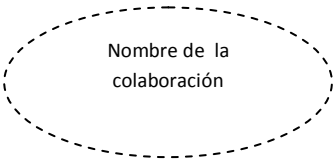


Figura 6- Representación gráfica de una colaboración

Casos de uso: Es la descripción de un conjunto de acciones que un sistema ejecuta y que produce un determinado resultado que es de interés para un actor particular. Un caso de uso se utiliza para organizar los aspectos del comportamiento en un modelo *[Alarcón, 2000].*

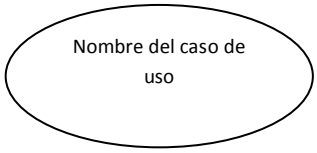


Figura 7- Representación gráfica de un caso de uso

Clase activa: Es una clase cuyos objetos tienen uno o más procesos o hilos de ejecución por lo tanto pueden dar lugar a actividades de control. Una clase activa se diferencia de una clase porque sus objetos representan elementos cuyo comportamiento es concurrente con otros elementos *[Alarcón, 2000].*

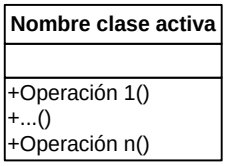


Figura 8- Representación gráfica de una clase activa

Componentes: Es una parte física y reemplazable de un sistema que conforma con un conjunto de interfaces y proporciona la implementación de dicho conjunto. Un componente representa típicamente el empaquetamiento físico de diferentes elementos lógicos, como clases, interfaces y colaboraciones *[Alarcón, 2000].*

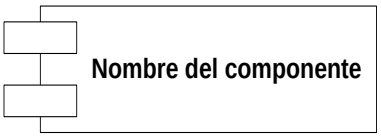


Figura 9- Representación gráfica de un componente

Nodos: Un nodo es un elemento físico que existe en tiempo de ejecución y representa un recurso computacional que, por lo general, dispone de algo de memoria y, con frecuencia, de capacidad de procesamiento. Un conjunto de componentes puede residir en un nodo *[Alarcón, 2000].*

Figura 10- Representación gráfica de un nodo

4.8.2. Elementos de Comportamiento

Son las partes dinámicas de modelo, se podría decir que son los verbos de un modelo y representan el comportamiento en el tiempo y en el espacio.

Interacción: Es un comportamiento que comprende un conjunto de mensajes intercambiados entre un conjunto de objetos, dentro de un contexto particular para conseguir un propósito específico. Una interacción involucra otros muchos elementos, incluyendo mensajes, secuencias de acción (comportamiento invocado por un objeto) y enlaces (conexiones entre objetos) *[Alarcón, 2000].*

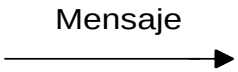


Figura 11- Representación gráfica de un mensaje

Maquinas de estados: Es un comportamiento que especifica las secuencias de estados por las que van pasando los objetos o las interacciones durante su vida en respuesta a eventos, junto con las respuestas a esos eventos *[Alarcón, 2000].*

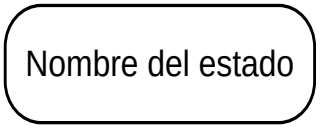


Figura 12- Representación gráfica de un estado

4.8.3. Elementos de Agrupación

Son los que forman la parte organizativa de UML, el principal elemento de agrupación es el **paquete**, el cual no es más que un mecanismo para organizar elementos en grupos.

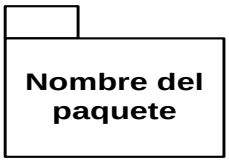


Figura 13- Representación gráfica de un paquete

4.8.4. Elementos de Anotación

Son la parte explicativa de los modelos de UML, estos nos permiten hacer comentarios sobre cualquier elemento de un modelo, el tipo principal de anotación es la nota.

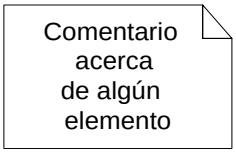


Figura 14- Representación gráfica de una nota

4.8.5. Diagramas

La finalidad de los diagramas es presentar diversas perspectivas de un sistema, a las cuales se les conoce como modelo.

UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas, a continuación se mencionaran los diagramas que son parte de UML y posteriormente se describirán los diagramas que se utilizaron en el desarrollo del sistema:

- Diagramas de clases
- Diagramas de objetos
- Diagramas de casos de usos
- Diagramas de estados
- Diagramas de secuencias
- Diagramas de actividades
- Diagramas de colaboración
- Diagramas de componentes
- Diagramas de distribución

4.8.5.1. Diagramas de Clases

Los diagramas de clase muestran las características estáticas del sistema y no representan ningún procesamiento en particular, también muestran la naturaleza de las relaciones entre las clases.

Un diagrama de clase podría mostrar simplemente el nombre de la clase; o el nombre de la clase y los atributos; o el nombre de la clase, los atributos y los métodos. Mostrar sólo el nombre de la clase es útil cuando el diagrama es muy complejo e incluye muchas clases *[Kendall (6), 2005].*

En el párrafo anterior se mencionaron dos conceptos bien importantes que contiene una clase que son los atributos y métodos a continuación se definirán estos conceptos:

- **Atributos:** Es una propiedad o característica de lo que la clase sabe acerca del objeto.
- **Métodos (Operaciones):** Es una operación que la clase puede realizar, en pocas palabras los métodos no son más que rutinas de código que trabajan con los atributos.

Relaciones en los Diagramas de Clases

Las relaciones las podríamos definir como conexiones en este caso entre clases, algo similar a lo que se realiza en el diagrama entidad-realización. A continuación se describirán las categorías de las relaciones en los diagramas de clases.

Asociaciones

Las asociaciones se muestran como una línea simple, al final de la línea se etiquetan con un símbolo que indica la multiplicidad (cardinalidad).

Tipos de asociaciones	Descripción
1 ——— 1	La relación 1 indica uno y solo uno, en este caso es una relación uno a uno.
1 ——— 1..*	La relación 1..* indica de 1 a muchos, en este caso es una relación uno a muchos (como mínimo 1).
1 ——— 0..*	La relación 0..* indica de 0 a muchos, en este caso es una relación de uno a muchos (como mínimo cero).
1 ——— *	La relación * indica de muchos, en este caso es una relación uno a muchos.

Tabla 1- Asociaciones de Clases

Relaciones todo/parte

Surgen cuando una clase representa al objeto total y otras clases representan parte del mismo, estas relaciones se muestran en los diagramas de clases mediante una línea con un diamante en un extremo. El diamante se conecta al objeto total *[Kendall (b), 2005].*

Las relaciones todo/parte tienen varias categorías las cuales se describirán a continuación:

Agregación: Se describe como una relación "tiene un", la agregación proporciona un medio para mostrar que el objeto total se compone de la suma de sus partes, estas relaciones se consideran relaciones débiles. El diamante al final de la línea no aparece solido *[Kendall (b), 2005].*

Colección: Una colección consta de un todo y sus miembros, esta se considera una asociación débil *[Kendall (b), 2005].*

Composición: Es una relación todo/parte en la que el todo tiene una responsabilidad por la parte, es una relación fuerte y el diamante al final de la línea se muestra sólido. En la composición una clase siempre contiene a otra clase, entonces si se elimina el todo se eliminan todas las partes *[Kendall (b), 2005].*

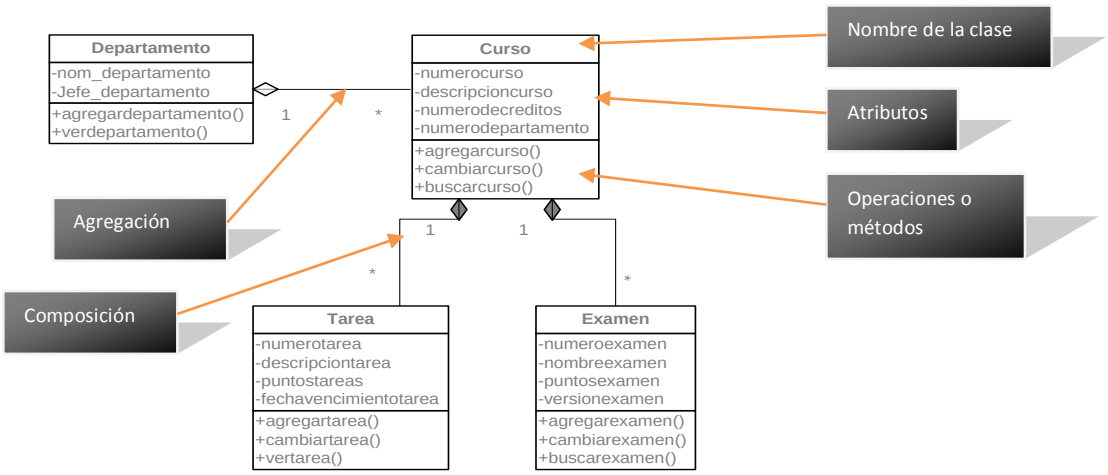


Figura 15- Diagrama de clases

4.8.5.2. Diagrama de Secuencias

Kendall describe los diagramas de secuencias como una sucesión de interacciones entre clases o instancias de objetos en un periodo determinado. Estos diagramas se derivan de los casos de usos y se utilizan para mostrar el patrón general de las actividades o interacciones en un caso de uso *[Kendall (b), 2005].*

En un diagrama de secuencia un *objeto* se representa como una caja en la parte superior de una línea vertical punteada (**Ver figura 19**), esta línea vertical se llama *línea de vida* del objeto. La línea de vida representa la vida del objeto durante la interacción.

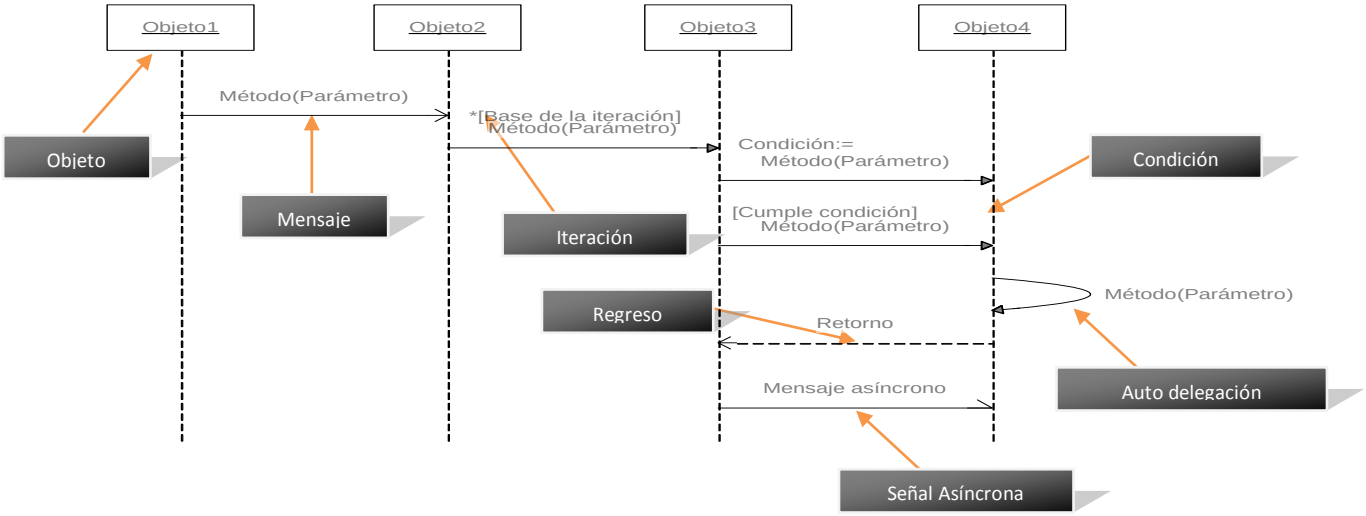


Figura 16- Estructura de un diagrama de secuencia

Cada *mensaje* se representa por medio de una flecha entre la línea de vida de dos objetos, el orden en el que transcurren los mensaje es de arriba hacia abajo, cada mensaje es etiquetado por lo menos con el nombre del mensaje, aunque también puede incluirse los argumentos o alguna información de control, también se puede mostrar la *auto delegación* que es un mensaje que un objeto se envía a sí mismo, regresando la flecha de mensaje de vuelta a la misma línea de vida *[Fowler, 1999].*

En la información de control existen dos partes valiosas que es la *condición* y el marcador de *iteración*, en el primero indica cuándo se envía un mensaje, este se envía solo si la condición es verdadera, mientras que en el segundo el mensaje es enviado muchas veces a varios objetos. La base de la iteración se puede mostrar entre corchetes como se observa en la **figura 16**.

En el diagrama de la **figura 16** también se observa un *regreso,* el cual indica el regreso de un mensaje, este se diferencia de los mensajes debido a que su línea es punteada y no continua.

Las flechas con media punta representan llamadas asíncronas, es decir llamadas que se emiten sin esperar a que sean devueltas al objeto que las emitió.

4.8.5.6. Diagramas de Colaboración

Los diagramas de colaboración describen la interacción entre dos o más cosas en el sistema, las cuales desempeñan en conjunto un comportamiento superior al que puede realizar cualquiera de las cosas por sí sola *[Kendall (6), 2005].*

Los diagramas de colaboración se diferencian de los de secuencias en la lectura ya que podría ser más difícil, además estos ponen énfasis en la organización de los objetos, mostrando la ruta que indica cómo se enlaza un objeto con otro.

En los diagramas de colaboración los objetos se muestran como una caja pero a diferencia de los diagramas de secuencias estas cajas no tienen la línea de vida (**Ver figura 17**). Las flechas al igual que en los diagramas de secuencias muestran los mensajes enviados, sin embargo en los diagramas de colaboración la secuencia de los mensajes se indica numerándolos.

Los diagramas de colaboración nombran los objetos de la forma *nombre_objeto:nombre_clase*, donde se puede omitir el nombre del objeto o el de la clase, pero si se omite el nombre del objeto hay que conservar los dos puntos para que quede claro que es la clase y no el objeto *[Fowler, 1999].*

Los diagramas de colaboración tiene dos formatos para enumerar los mensajes, esta la enumeración simple (Ejemplo: 1, 2, 3, 4) y la enumeración decimal (Ejemplo: 1, 1.1, 1.2, 1.2.1) que es la que utiliza UML, en este esquema es fácil apreciar la comunicación entre las operaciones como una llama a otra, aunque es un poco difícil apreciar la secuencia tal a como lo menciona Fowler *[Fowler, 1999].*

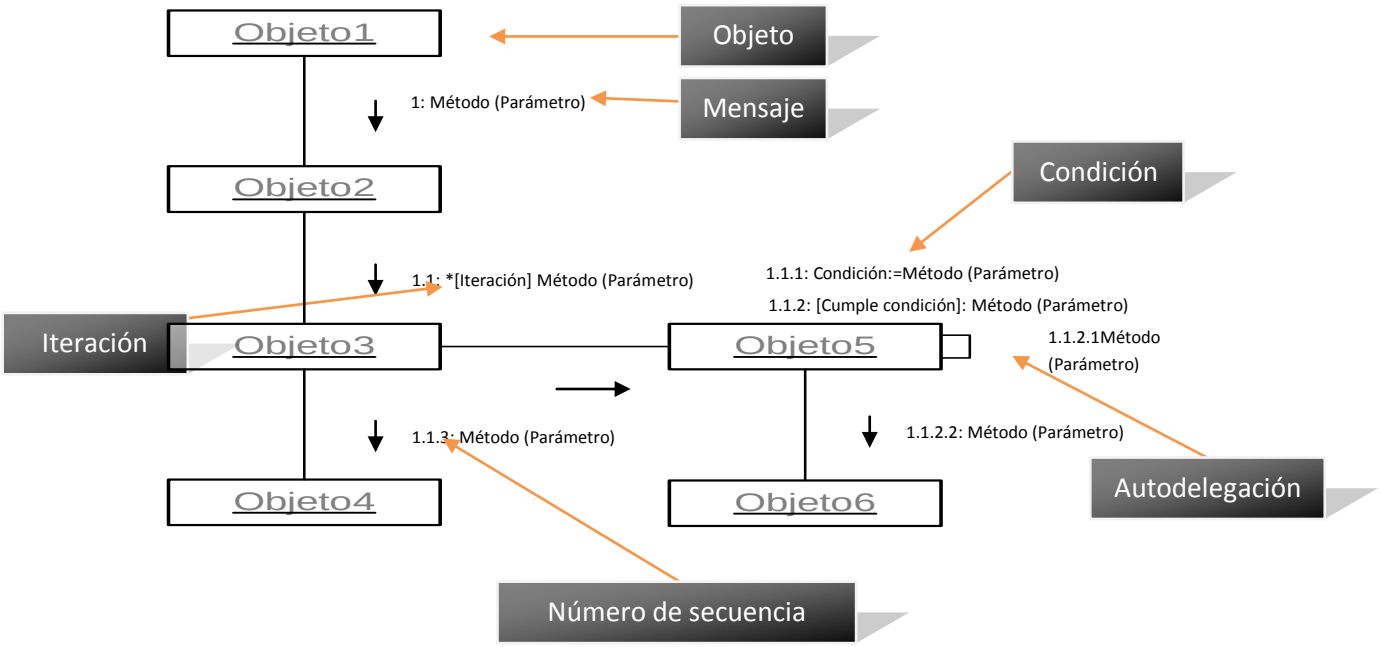


Figura 17- Estructura de un diagrama de colaboración con numeración decimal

4.8.5.3. Modelado de Casos de Uso

El UML está basado fundamentalmente en una técnica de análisis orientada a objetos conocida como modelado de casos de uso, el cual describe lo que hace un sistema sin describir como lo hace; es decir es modelo lógico del sistema. El UML se puede usar para analizar el modelo de caso de uso y para derivar objetos del sistema y sus interacciones entre sí y con los usuarios del sistema *[Kendall (6), 2005].*

El modelo de caso de uso proporciona medios eficaces de comunicación entre el equipo del negocio y el equipo de desarrollo. Un modelo de caso de uso divide la funcionalidad del sistema en comportamientos, servicios y respuestas que son significativos para los usuarios del sistema.

En los diagramas de casos de usos existen dos elementos importantes los casos de uso y los actores:

Casos de uso: Larman lo describe como un documento narrativo que describe la secuencia de eventos de un actor, son historias o casos de utilización del sistema, no son exactamente los requerimientos ni las especificaciones funcionales, sino que ejemplifican tácticamente los requerimientos en las historias que narran *[Larman, 1999].*

Actores: Es una entidad externa del sistema que de alguna forma participa en los casos de usos, por lo general estos estimulan al sistema con eventos de entrada de datos al sistema o algo de el actor *[Larman, 1999].*

Un método con que se identifican los casos de usos se basa en los actores:

1. Se identifican los actores relacionados con un sistema o empresa.
2. En cada actor, se identifican los procesos que inician o en que participan.

Un segundo método de identificación de los casos de usos se basa en los eventos:

1. Se identifican los eventos externos a los que un sistema ha de responder.
2. Se relacionan los eventos con los actores y con los casos de uso.

Tipos de casos de usos

- **Casos primarios de usos:** Representan los procesos comunes más importantes.
- **Casos secundarios de usos:** Representan procesos menores o raros.
- **Casos opcionales de usos:** Representan procesos que pueden no abordarse.
- **Casos esenciales de usos:** Son casos expandidos que se expresan en forma teórica que contiene poca tecnología y pocos detalles de implementación, las decisiones de diseño se posponen y abstraen de la realidad, especialmente los concernientes a la interfaz para el usuario *[Larman, 1999].*
- **Casos reales de usos:** Describe concretamente el proceso a partir de su diseño concreto actual, sujeto a las tecnologías específicas de entrada y salidas. Cuando se trata de la interfaz para el usuario a menudo presenta pantallas y explica la interacción con los artefactos *[Larman, 1999].*

Relaciones del caso de uso

Hay 4 tipos básicos de relaciones de comportamiento: Comunica, incluye, extiende y generaliza.

- **Comunica:** Se usa para conectar a un actor con un caso de uso
- **Incluye:** Describe la situación en que un caso de uso contiene un comportamiento que es común para más de un caso de uso
- **Extiende:** Describe la situación en la que un caso de uso posee el comportamiento que permite al nuevo caso de uso manejar una variación o excepción del caso de uso básico.
- **Generaliza:** Implica que una cosa es más típica que otra, esta relación podría existir entre dos actores o dos casos de usos.

Relación	Símbolo	Significado
Comunica		Un actor se conecta a un caso de uso usando una línea sin puntas de flechas.
Incluye	<<Incluye>>	Un caso de uso contiene un comportamiento que es más común que otro caso de uso. La flecha apunta al caso de uso común.
Extiende	<<Extiende>>	Un caso de uso diferente maneja las excepciones del caso de uso básico. La flecha apunta desde el caso de uso extendido hacia el básico.
Generaliza		Una "cosa" de UML es más general que otra "cosa". La flecha apunta a la "cosa" general

Tabla 2- Relaciones de Casos de Uso

Al diagramar un caso de uso empiece pidiendo a los usuarios que mencionen todo lo que el sistema debe hacer para ellos. Escriba quien está involucrado con cada caso de uso y los servicios que el caso de uso debe proporcionar a los actores. Use los siguientes lineamientos *[Kendall (b), 2005]:*

- Revise las especificaciones del negocio e identifique los actores en el dominio del problema.
- Identifique los eventos de alto nivel y desarrolle los casos de uso principales que describen dichos eventos y cómo los inician los actores.
- Revise cada caso de uso principal para determinar las posibles variaciones de flujo a través del caso de uso. Debido a que el flujo de eventos es normalmente diferente en cada caso busque actividades que podrían tener éxito o fallar.

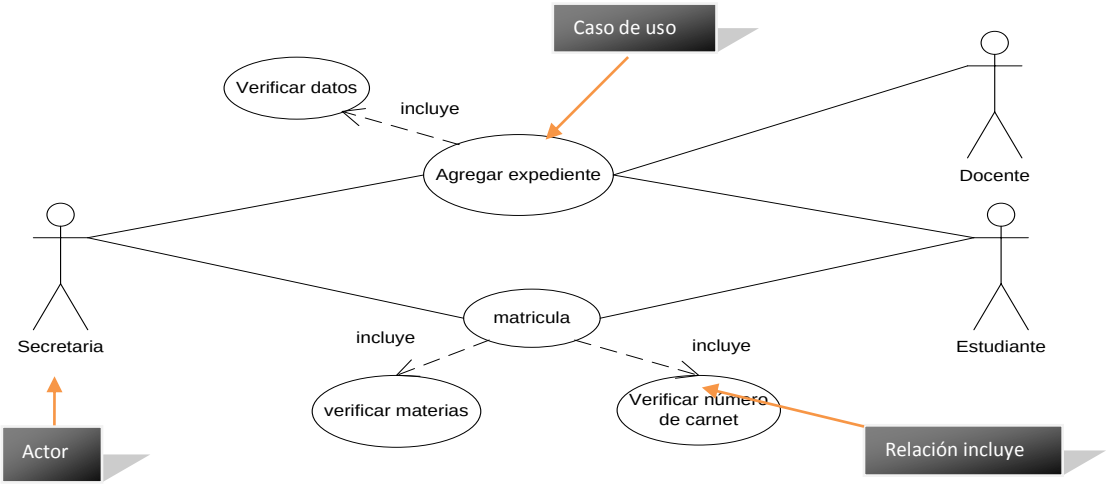


Figura 18- Ejemplo de un diagrama de casos de usos

4.9. Programación

4.9.1. Plataforma .NET

Microsoft .NET extiende las ideas de Internet y sistema operativo haciendo de la propia Internet la base de un sistema operativo.

En última instancia este permitirá a los desarrolladores crear programas que trasciendan los límites de los dispositivos y aprovechen por completo la conectividad de Internet y sus aplicaciones. Para ello proporciona una plataforma que incluye los siguientes componentes básicos *[Ceballos, 2006]:*

- Herramientas de programación para crear servicios Web con soporte multilingüe: Visual Studio .NET y .NET Framework.
- Infraestructura de servidores, incluyendo Windows y .NET Enterprise Server.
- Experiencias .NET utilizadas por los usuarios finales para acceder a los servicios Web.

4.9.2. NET Framework

Claramente se requiere de una infraestructura, no solo para facilitar el desarrollo de las aplicaciones, sino también para hacer el proceso de encontrar un servicio Web e integrarlo en una aplicación resulte transparente para usuarios y desarrolladores .NET Framework *[Ceballos, 2006].*

4.9.3. Visual Studio 2008

Es un conjunto completo de herramientas de desarrollo para construir aplicaciones Web, servicios Web, aplicaciones Windows, o de escritorio y aplicaciones para dispositivos móviles *[Ceballos, 2006].*

El entorno de desarrollo que ofrece esta plataforma con todas sus herramientas y con a la biblioteca de clases .NET Framework, es compartido en su totalidad por Visual Basic, Visual C++, y Visual C#, permitiendo así crear con facilidad soluciones en las que intervengan varios lenguajes y en las que el diseño se realiza separadamente respecto a la programación *[Ceballos, 2006].*

Dentro de la plataforma de visual Studio está incluido el lenguaje que se utilizo en el momento de creación del sistema PMCSIS el cual es:

4.9.4. Visual Basic .Net

Es un lenguaje orientado a objetos seguro y elegante que permite a los desarrolladores construir un amplio rango de aplicaciones seguras que se ejecutan sobre .NET Framework. Podemos utilizar Visual Basic para crear aplicaciones clientes Windows tradicionales, servicios Web XML, componentes distribuidos, aplicaciones cliente servidor, aplicaciones para acceso a base de datos, y muchas otras. Visual Basic 2008 proporciona un editor de código avanzado, diseñadores de interfaces de usuarios apropiados, depurador integrado, y muchas otras utilidades para facilitar el desarrollo rápido de aplicaciones *[Ceballos, 2006].*

4.9.5. ASP.NET

Es una plataforma que incorpora una serie de características y utilidades para diseñar aplicaciones Web: Formularios Web o Servicios Web.

ASP.NET se ejecuta en el servidor. En ASP.NET las aplicaciones web se suelen desarrollar utilizando formularios web, que están diseñado para hacer las aplicaciones web de forma sencilla, entre ellas se incluyen las aplicaciones que se pone a disposición del público a través del World Wide Web.

Permite aprovechar todas las características de la plataforma .NET para el diseño de aplicaciones modulares usando técnicas de orientación a objetos, en ASP.NET existen dos estilos:

- Incluir tanto los controles como el código en un único fichero .aspx.
- Permite mantener los controles de nuestra interfaz en un fichero .aspx y dejar todo el código en un lugar aparte (otro fichero).

4.10. Creación y Diseño de un Sitio Web

Una página Web no es más que un documento adaptado para que se pueda visualizar en la Web, la principal característica que tiene son los hiperenlaces que esta puede contener para poder mostrar otra página, tal a como lo define Berzal "Las páginas web son documentos de texto a los que se le añaden etiquetas que nos permiten visualizar el texto en distintas formas y establecer enlaces entre una página y otra" *[Berzal, 2005].*

4.10.1. Sitio Web

Es un conjunto de páginas web, localizadas en la World Wide Web (WWW), cada sitio web tiene una página de inicio llama por defecto índex, que es el primer documento que el usuario mira al ingresar a un sitio web.

El diseño web es una actividad multidisciplinar y reciente, debido a que esta surge con la aparición de internet en 1969 y luego en 1990 aparece lo que hoy conocemos como la World Wide Web (WWW) que es el medio por el cual se puede mirar las páginas web, estas se alimenta de fuentes como el diseño grafico y las artes visuales, la programación de aplicaciones informáticas, el diseño de interfaces, la redacción de contenidos, la animación tradicional, la publicidad, el marketing y otras muchas más.

Según Mariño "Es un proceso creativo que combina arte y tecnología para comunicar ideas" *[Mariño, 2005]*

El diseñador es quien trata de crear, elegir, organizar y disponer toda la información toda la información de la empresa.

El medio de comunicación de las páginas web es internet, y como todo acto de comunicación consta de los siguientes elementos: emisor, receptor, canal, código, mensaje.

- **Emisor:** Es el origen, de donde sale el mensaje.
- **Receptor:** Es el destino, a quien va dirigido el mensaje
- **Canal:** Es el medio físico por medio de donde se transmite el mensaje.
- **Código:** Es el conjunto de rasgos necesarios para que este pueda ser entendido por el receptor y emisor.
- **Mensaje:** es aquello que se quiere transmitir.

Las tres actividades fundamentales al visitar una página web son:

- Ver imágenes
- Leer textos
- Interaccionar con la interfaz

La informática tiene un peso muy importante en la creación y gestión de un sitio web, esta la trasmisión de información (Servidores, protocolos), gestión de datos (bases de datos, programación del lado del servidor), y la presentación en pantalla (Programación del lado del cliente).

4.10.2. Principales Facetas del Diseño Web

Dentro de las principales facetas tenemos las siguientes *[Mariño, 2005]:*

- **Contenido:** Son los datos, abarca aspectos como la organización, presentación, etc.
- **Aspecto visual:** Es el aspecto formal, su función es complementar y enriquecer los contenidos, lograr una sensación de equilibrio, orden, belleza y de proporcionar una navegación sencilla al usuario.
- **Programación:** Es el conjunto de instrucciones que permite la interactividad de un sitio web
- **Almacenamiento y distribución por la red:** Para que la página web sea accesible a los usuarios debe de ser alojada en un servidor (Equipo informático que se encarga de atender las solicitudes de los usuarios).
- **Finalidad de la web:** Es el propósito que se persigue, para que fue diseñada la página web.

El aspecto final de la web se verá influenciado por cada una de estas aéreas en menor o mayor medida, dependiendo del tipo de sitio web que se desarrolle.

4.10.3. Desarrollo de Aplicaciones Web con ASP.NET

Primero hay que definir que es una aplicación web según Berzal "Son aquellas cuya interfaz se construyen utilizando páginas web", las características común que tienen todas las aplicaciones web es el hecho de centralizar el software para facilitar las tareas de mantenimiento y actualización de grandes sistemas, en pocas palabras es evitar tener copias de las aplicaciones en todos los puestos de trabajos *[Berzal, 2005].*

Dentro de las aplicaciones web existen dos categorías:

- **Aplicaciones web estáticas:** En estas las páginas web se limitan a tener almacenado documentos en formato HTML (Por sus siglas en ingles de lenguaje de marcado de hipertexto), estos no son más que fichero de texto, con una serie de etiquetas.
- **Aplicaciones web dinámicas:** En estas el contenido de las páginas va cambiando dinámicamente en función del usuario que la visite.

Aunque las aplicaciones web se diseñen con la misma filosofía, se pueden diferenciar dos grupos de aplicaciones web en función de si la lógica de la aplicación se ejecuta en el cliente o en el servidor.

- **En el cliente:** El servidor y el navegador instalado en la maquina cliente se limita a mostrar paginas HTML.
- **En el servidor:** Las aplicaciones que se ejecutan el servidor pueden recibir información del cliente de distintas formas, luego como resultado de la ejecución se genera una respuesta que es enviada al cliente.

4.10.4. ASP: Active Server Page

ASP es la tecnología de la Microsoft que permite desarrollar aplicaciones web que se ejecutan en el servidor HTTP de Microsoft, el Internet Information Server (IIS).

Berzal explica que ASP consiste, básicamente en intercalar macros o fragmentos de código dentro de los documentos HTML que sirven para crear interfaces de usuarios de las aplicaciones web. Los fragmentos de HTML proporcionan la parte estática mientras que los fragmentos de código generan la parte dinámica *[Berzal, 2005]*.

Una página web no es más que un fichero HTML con extensión .asp (en nuestro caso con extensión .aspx por trabajar con ASP.NET) al que se le añade algo de código este código se puede implementar usando algunos lenguaje de programación tales como C# y Visual Basic, esto funciona de la siguiente manera: Cuando un usuario accede a una página por medio del navegador web el IIS interpreta que incluye la página y combina el resultado de su ejecución con la parte estática de ASP, una vez interpretada la página ASP, el resultado final se le envía al navegador web para que el usuario pueda accesar a la pagina.

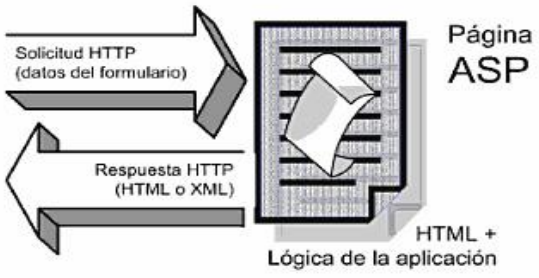


Figura 19– Funcionamiento de las paginas ASP.

En ASP el programador dispone de una serie de objetos predefinidos que permite simplificar el trabajo, estos objetos hacen uso de la encapsulación para omitir detalles de la comunicación del navegador web del cliente con el servidor HTTP. A continuación se mencionarán los objetos y la función que realiza:

- **Request:** La solicitud HTTP recibida
- **Response:** La respuesta HTTP generada
- **Server:** El estado del servidor
- **Application:** El estado de la aplicación web
- **Session:** La sesión de usuario

4.11. Factibilidad

El estudio de factibilidad no es un estudio completo de un sistema, sino más bien se usa el estudio para recopilar una serie de datos para la administración, para así poder permitir a la administración tomar una decisión acerca de continuar con el proyecto, para los proyectos de sistemas la factibilidad es valorada de tres formas: operacional, técnica y económicamente.

4.11.1. Factibilidad Técnica

Gran parte de la determinación de los recursos tiene que ve con la evaluación de la factibilidad técnica. El analista debe averiguar si es posible actualizar o incrementar los recursos técnicos actuales de tal manera que satisfagan los requerimientos bajo consideración. Sin embargo, los “agregados” a los sistemas existentes son costosos y no redituables, simplemente porque no cumplen las necesidades con eficiencia. Si no es posible actualizar los sistemas existentes, la siguiente pregunta es si hay tecnología disponible que cumpla las especificaciones.

Es común que la respuesta a la pregunta sobre si una tecnología especifica está disponible y puede satisfacer las necesidades de los usuarios, sea “Si”, y entonces la pregunta pasa ámbito económico *[Kendall (6), 2005].*

4.11.2. Factibilidad Económica

La factibilidad económica es la segunda parte de la determinación de recursos. Los recursos básicos que se deben considerar son el tiempo de usted y el del equipo de análisis de sistemas, el costo de realizar un estudio de sistemas completo (incluyendo el tiempo de los empleados con los que trabajara usted), el costo del tiempo de los empleados de la empresa, el costo estimado del hardware y el costo estimado del software comercial o del desarrollo del software *[Kendall (6), 2005].*

La empresa interesada debe tener la capacidad de calcular el valor de la inversión bajo evaluación antes de comprometerse a un estudio de sistemas completo. Si los costos a corto plazo no son opacados por las ganancias a largo plazo o no producen una reducción inmediata de los costos operativos, el sistema no es económicamente viable y el proyecto debe detenerse.

4.11.3. Factibilidad Operativa

Suponiendo por un momento que los recursos técnicos y económicos se evaluaron de manera adecuada. El analista de sistemas aun debe considerar la factibilidad operativa del proyecto solicitado. La factibilidad operativa depende de los recursos humanos disponibles para el proyecto e implica determinar si el sistema funcionará y será utilizado una vez que se instale *[Kendall (b), 2005].*

4.12. Métricas del Software

La medición es elemento clave en cualquier proceso de ingeniería, las medidas se utilizan para comprender mejor los atributos de los modelos que se crean y evaluar la calidad de los productos de ingeniería o de los sistemas que se construyen.

Mediante estas mediciones se asignan números o símbolos a los atributos para definirlas de acuerdo con reglas claramente establecidas.

4.12.1. Métricas de Productos

Aunque medida, medición y métrica son términos que suelen usarse de manera intercambiable, es importante observar las diferencias entre ellos: *[Pressman (b), 2006].*

- **Medida:** proporciona una indicación de cuantitativa de la extensión, la cantidad, la dimensión, la capacidad o el tamaño de algún atributo de un producto o proceso.
- **Medición:** Es el acto de determinar una medida.
- **Métrica:** Medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo determinado.

A continuación se mencionarán las áreas de mayor importancia en las métricas de producto:

- Métricas del modelo de análisis
- Métricas para el modelo de diseño
- Métricas para el código fuente
- Métricas para pruebas

En las métricas del modelo de análisis están las métricas basadas en la función la cual se describirán a continuación.

Métricas basadas en la función

La métrica de punto de función (PF) se usa de manera efectiva para la funcionalidad que entrega un sistema, el PF se usa para *[Pressman (b), 2006]:*

- Estimar el costo o esfuerzo requerido para diseñar, codificar y probar el software.
- Predecir el número de errores que se encontraran durante la prueba
- Pronosticar el número de componentes, de líneas de códigos proyectadas, o ambas en el sistema implementado.

Los valores del dominio de la función se definen de la siguiente manera:

- Número de entradas externas (EE)
- Número de salidas externas (SE)
- Número de consultas externas (CE)
- Número de archivos lógicos internos (ALI)
- Número de archivos de interfaz externos (AIE)

Las métricas de productos se dividen en dos clases según [Sommerville, 2005]:

- **Métricas dinámicas:** Son recogidas por las mediciones hechas en un programa en ejecución.
- **Métricas estáticas:** Son recogidas por las mediciones hechas en las representaciones del sistema como el diseño, el programa o la documentación.

Métricas estáticas de producto software	
Fan-in/Fan-out	Fan-in es una medida del número de funciones o métodos que llaman a otra función o método. Fan-out es el número de funciones que son llamadas por una función.
Longitud de código	Esta es una medida del tamaño del programa.
Complejidad ciclomática	Es una medida de complejidad del control del programa, está relacionada con la comprensión del programa.
Longitud de los identificadores	Es una medida de longitud promedio de los diferentes identificadores en un programa.
Profundidad de anidamiento de las condiciones	Es una medida de profundidad de anidamiento de las instrucciones condicionales "IF" en un programa.
Índice de Fog	Es una medida de longitud promedio de las palabras y las frases en los documentos.

Tabla 3- Métricas estáticas de producto software

Métricas orientadas a objetos	
Profundidad del árbol de herencia	**Representa el número de niveles discretos del árbol de herencia donde las subclases heredan atributos y operaciones de la superclases**
Método Fan-in/Fan-out	**Ver Tabla 3.**
Métodos pesados por clase	**Número de métodos incluidos en una clase con sus correspondientes pesos, que vendrá dado por la complejidad de cada método.**
Número de operaciones sobrescritas	**Número de operaciones en una subclase que se anulan en una superclase**

Tabla 4- Métricas orientadas a objetos

Métricas para productos web	
Cantidad de palabras	Total de palabras en una pagina
Porcentaje de texto en el cuerpo	Porcentaje de palabras que están en el cuerpo versus Texto visualizado (Es decir cabeceras).
Porcentaje de texto enfatizado en el cuerpo	Porción de texto enfatizado en el cuerpo (por ejemplo texto en negrita, letra capital)
Cantidad de texto posicionado	Cambios en el la posición del texto de alineación izquierda.
Cantidad de texto agrupado	Área de textos resaltados con color, regiones con bordes, reglas o listas.
Cantidad de enlaces	Total de enlaces en una pagina
Tamaño de la página	Número total de bytes de la página, así como elementos gráficos y hojas de estilo
Porcentajes de gráficos	Porcentaje de bytes en una página que son de los gráficos
Cantidad de gráficos	Total de gráficos en una página (no incluye gráficos especificados en un script, applet y objetos)
Cantidad de colores	Total de colores empleados
Cantidad de fuentes	Total de fuentes empleadas(es decir)

Tabla 5- Métricas para productos Web

4.12.1.1. Principios de Medición

Las métricas de productos ayudan a evaluar los modelos de análisis y diseño, ofrecen una indicación de la complejidad de los diseños procedimentales y el código fuente, facilita el diseño de pruebas más efectivas.

A continuación se describen las actividades que describen un proceso de medición *[Pressman (b), 2006]:*

- Formulación: La derivación de medidas y métricas apropiadas para la representación del software que se considera.
- Recolección: El mecanismo con que se acumulan los datos necesarios para derivar las métricas formuladas.
- Análisis: El cálculo de las métricas y la aplicación de las herramientas matemáticas.
- Interpretación: La evaluación de las métricas en un esfuerzo de conocer mejor la calidad de la representación.
- Retroalimentación: Recomendaciones derivadas de la interpretación de las métricas del producto transmitidas al equipo del software.

4.12.1.2. Atributos de las Métricas efectivas del Software

Pressman menciona un conjunto de atributos que toda métrica efectiva del software debe abarcar, la métrica derivada y las medidas que lleven a ella deben de ser: *[Pressman (b), 2006]*

- Simples y calculables: Debe de ser relativamente fácil aprender a derivar la métrica, y su cálculo no debe de exigir cantidades anormales de tiempo y esfuerzo.
- Empírica e intuitivamente persuasivas: La métrica debe de satisfacer las nociones intuitivas del ingeniero acerca del atributo del producto que se está construyendo.
- Consistentes y objetivas: La métrica siempre debe arrojar resultados que no permitan ambigüedad alguna.
- Consistentes en el uso de unidades y dimensiones: El cálculo matemático de la métrica debe de emplear medidas que no lleven a combinaciones extrañas de unidades.
- Independientes del lenguaje de programación: Las métricas deben de basarse en el modelo de análisis y diseño, o en la estructura del propio programa.
- Mecanismos efectivos para la retroalimentación de alta calidad: Es decir, la métrica debe de llevar a un producto final de la más alta calidad.

4.12.2. Métricas de Proceso

Las métricas del proceso se recopilan en el curso de todos los proyectos y durante largos periodos, su objetivo es proporcionar un conjunto de indicadores de proceso que conduzcan a la mejora de procesos a largo plazo *[Pressman (b), 2006].*

La única forma racional de mejorar cualquier proceso es medir sus atributos específicos, desarrollar un conjunto de métricas significativas con base en dichos atributos y luego emplearlas para ofrecer indicadores para conducir a una estrategia de mejora *[Pressman (b), 2006].*

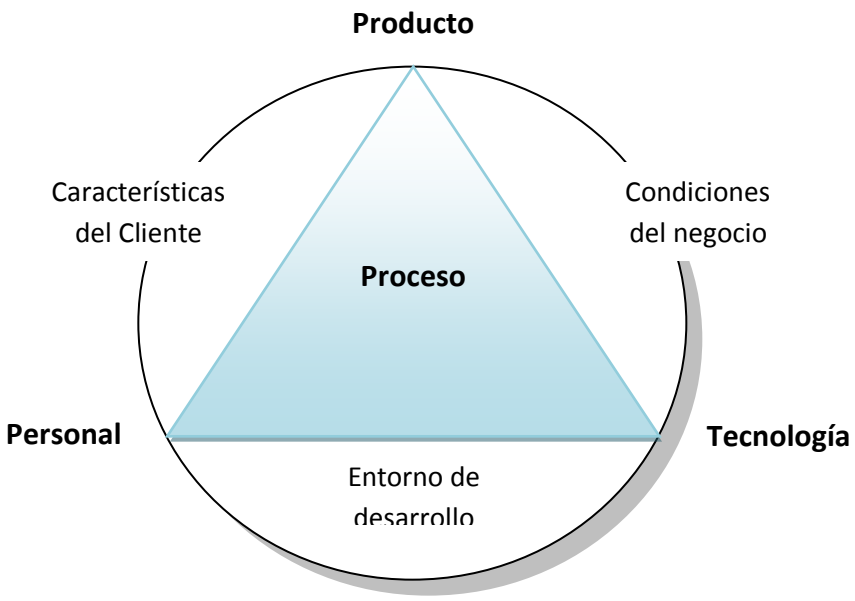


Figura 20- Determinantes para la calidad del software

En la **figura 20** el proceso se asienta en un triángulo que conecta tres factores con una profunda influencia en la calidad del software, la destreza y la motivación del personal se muestran como el factor individual más influyente en la calidad y el desempeño, la complejidad y la tecnología tienen un impacto sustancial sobre la calidad y el desempeño del proceso. Además, el triangulo del proceso existe dentro de un circulo de condiciones ambientales que incluye el entorno de desarrollo, condiciones del negocio y características del cliente.

4.12.3. Métricas de Proyecto

Las métricas de proyecto permiten que un gestor de proyecto de software 1) valore el estado de un proyecto en curso; 2) rastree los riesgos potenciales; 3) descubra las áreas con problemas antes que se conviertan en criticas; 4) ajuste el flujo de trabajo o las tareas; 5) evalúe la habilidad del equipo para controlar la calidad de los productos *[Pressman (b), 2006]*.

Las métricas de proyecto son tácticas, es decir que se emplean las métricas y sus indicadores para adaptar el flujo de trabajo del proyecto y las actividades técnicas.

La finalidad de las métricas de proyecto es doble: primero se emplea para minimizar el tiempo de desarrollo haciendo los ajustes necesarios para evitar demoras y reducir problemas y riesgos potenciales. Segundo, para valorar la calidad del producto sobre una base actual, y cuando es necesario, modificar el enfoque técnico para mejorar la calidad *[Pressman (b), 2006]*.

4.13. Diccionarios de Datos

El diccionario de datos es una aplicación especializada de los tipos de diccionarios usados como referencia en la vida cotidiana. El diccionario de datos es una obra de consulta con la información acerca de los datos (es decir *metadatos*), compilado por los analistas de sistemas para guiarse en el análisis y diseño *[Kendall (b), 2005]*.

Una razón importante para mantener los diccionarios de los datos es guardar los datos de manera ordenada, esto significa que los datos sean consistentes.

4.13.1. Elementos de Datos

Los siguientes son los elementos que comúnmente se incluyen en el formulario de descripción de elementos *[Kendall (b), 2005]*:

- ID del elemento
- Nombre del elemento
- Alias
- Descripción breve del elemento
- Si el elemento es base o derivado
- Longitud del elemento
- Tipo de datos
- Formato de entrada y salida
- Criterios de validación
- Cualquier valor predeterminado
- Un área adicional para observaciones o comentarios.

4.14. Pruebas del Software

La prueba es un conjunto de actividades que se planean con anticipación y se realizan de manera sistemática, por medio de esta se descubren errores cometidos sin darse cuenta al realizar su diseño y construcción.

La estrategia de prueba del software va depender del enfoque que este tenga, en este caso particular se trabajó con el enfoque incremental de pruebas debido a que el sistema se desarrolló con el proceso de desarrollo incremental **(Ver capitulo 4.5).** Este enfoque inicia con las pruebas individuales del programa, luego continua con las pruebas de integración de las unidades y culmina con las pruebas de entrega construido completamente.

4.14.1. Pruebas de Unidad

La prueba de unidad se concentra en la verificación de los componentes o módulos del sistema, estas se concentran en la lógica del procesamiento interno y en las estructuras de datos dentro de los límites de un módulo.

Pressman comenta algunas consideraciones acerca de las pruebas de unidad tales como *[Pressman (6), 2006]:*

- La interfaz del módulo se prueba para asegurar que la información fluye correctamente.
- Examinar estructuras de datos locales.
- Recorrer todos los caminos independientes (Caminos de base).
- Probar las condiciones limites.
- Probar todos los caminos de manejo de errores.

Procedimiento de las pruebas de unidad

El diseño de pruebas de unidad puede realizarse antes de que empiece la codificación o después de la codificación.

Según Pressman para cada prueba de unidad se debe de desarrollar un software controlar o de resguardo, en casi todas las aplicaciones un controlador, no es más que un programa principal que acepta los datos del caso de prueba, pasa estos datos al módulo e imprime los resultados, y los resguardos sirven para reemplazar módulos subordinados al módulo, un resguardo es "Un subprograma simulado" usa la interfaz del módulo subordinado, realiza una mínima manipulación de los datos proporciona verificación de entrada y devuelve el control al módulo de prueba *[Pressman (b), 2006].*

Los controladores y resguardos no forman parte del diseño, ni se entregan con el producto de software final, la prueba de unidad se simplifica cuando se diseña un componente con alta cohesión, en este caso las pruebas se reducen y es más fácil predecir y corregir errores.

4.14.2. Pruebas de Integración

La prueba de integración es una técnica sistemática para construir la arquitectura del software mientras, al mismo tiempo, se aplican las pruebas para descubrir errores asociados con la interfaz. El objetivo es tomar componentes a los que se aplicó una prueba de unidad y construir una estructura de programa que determine el diseño *[Pressman (b), 2006].*

En el caso de la integración incremental el programa se construye y prueba en pequeños incrementos, en lo que resulta más fácil corregir los errores.

Estrategias para la integración incremental *[Pressman (b), 2006]:*

- **Integración descendente:** Los módulos se integran al descender por medio de una jerarquía de control, empezando por el módulo de control principal (Programa principal).
- **Integración ascendente:** Los componentes se integran de abajo hacia arriba y se elimina la necesidad de resguardos.

- **Prueba de regresión:** Cada vez que se agregar un módulo el software cambia, es donde se aplica la prueba de regresión ejecutando un conjunto de pruebas para verificar que no haya efectos colaterales.
- **Prueba de humo:** Esta diseñado como mecanismo para marcar el ritmo en proyectos en los que el tiempo es crítico, lo que permite al equipo de desarrolladores evaluar el proyecto con frecuencia.

4.14.3. Prueba de entregas

El objetivo principal de estas pruebas es incrementar la confianza del desarrollador en que el sistema satisface los requerimientos, de esta manera puede entregarse al cliente. Para poder demostrar que satisface los requerimientos, tiene que demostrar que este entrega la funcionalidad especificada, rendimiento, confiabilidad y que no falle durante su uso normal.

Otro nombre para este es ***pruebas funcionales***, debido a que al probador solo le interesa la funcionalidad y no la implementación del software *[Sommerville, 2005]*.

Existen algunas guías para realizar las pruebas funcionales y que estas tengan éxito, por ejemplo:

1. Elegir entradas que fuerzan a que el sistema genere todos los mensajes de error.
2. Diseñar entradas que hacen que los búferes de entradas se desborden.
3. Repetir la misma entrada o series de entrada varias veces.
4. Forzar a que se generen las salidas inválidas.
5. Forzar los resultados de los cálculos para que sean demasiados grandes o demasiados pequeños.

Cuando los clientes se implican en este tipo de pruebas, estas a menudos se llaman ***pruebas de aceptación***, si la entrega es lo suficientemente buena, el cliente puede entonces aceptarla para su uso *[Sommerville, 2005]*.

4.15. Herramientas CASE con Rational Rose

CASE son las siglas en ingles “Computer Aided Software Engineering” que significa “ingeniería de software asistida por ordenador”, estas herramientas son diversas aplicaciones informáticas diseñadas para aumentar la productividad en el desarrollo de software. Estas herramientas se utilizan para ayudar a realizar la parte del diseño de un sistema, documentación, calcular costos, etc.

4.15.1. Rational Rose

Es una poderosa herramienta de modelado visual para asistir en el diseño de sistemas orientados a objetos, esta herramienta es utilizada para modelar el sistema que se va a desarrollar antes de escribir el código, por medio de los diferentes diagramas de UML ayuda a comprender mejor el funcionamiento del sistema que se está desarrollando.

Según Boggs un modelo Rose es una imagen de un sistema desde varias perspectivas, incluidos todos los diagramas de UML tales como los casos de usos, Clases, Secuencias, Estado, etc. *[Boggs, 2002].*

Rational Rose permite generar código en diferentes lenguajes a partir de un diagrama UML, a como también permite la ingeniería inversa que significa que a partir del código genera los diagramas.

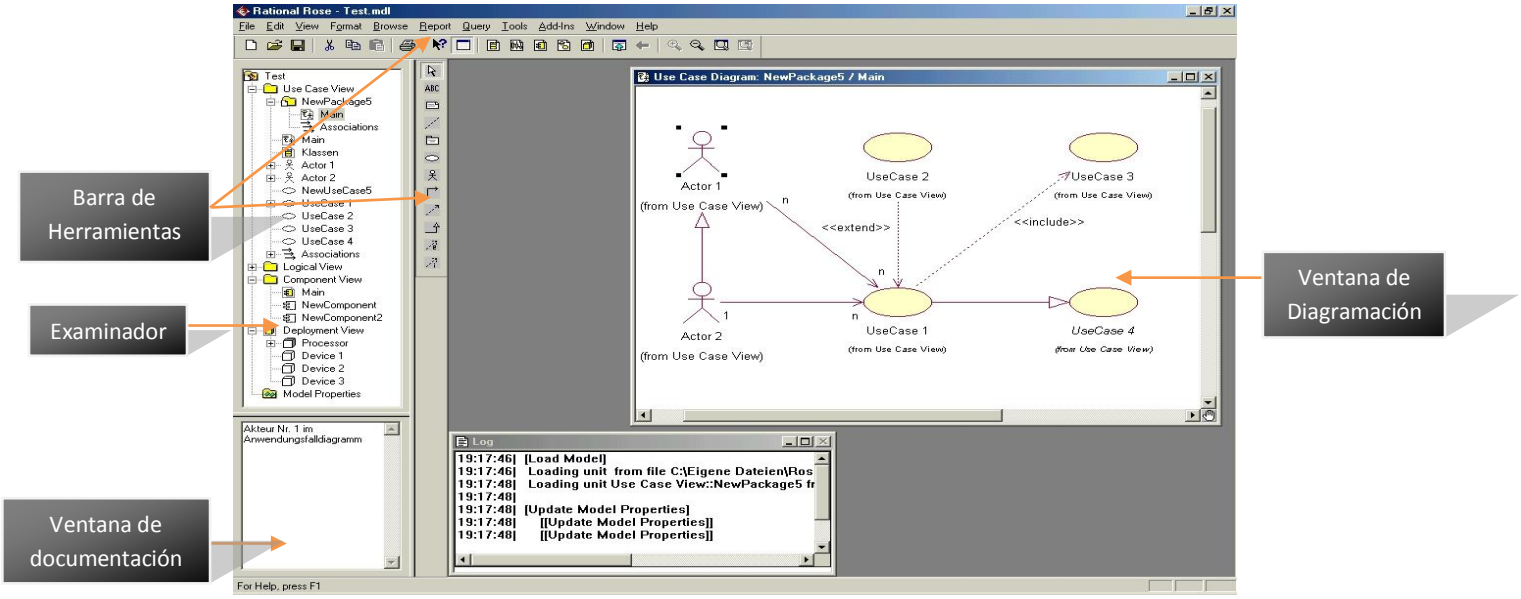


Figura 21– Interfaz de Rational Rose.

4.16. Conceptos Contables

4.16.1. Contabilidad

El propósito fundamental de la contabilidad es suministrar información valiosa para la toma de decisiones económicas a la persona encargada de tomarlas. Como se ilustra en la figura 22, la entrada al proceso contable es una actividad económica, la salida es información útil.

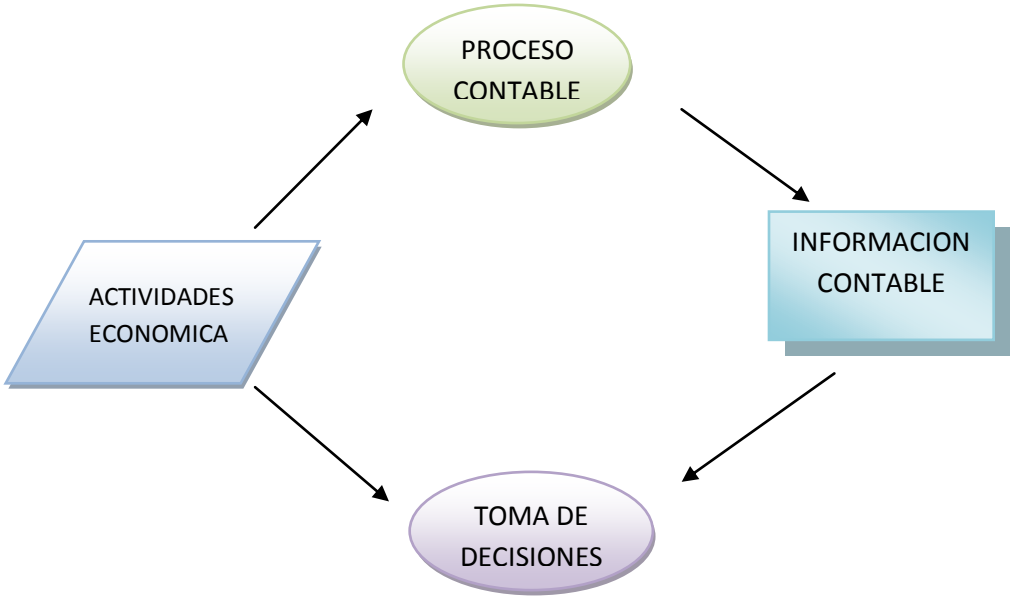


Figura 22 – Propósito de la contabilidad

La Contabilidad vincula a quien toma las decisiones con las actividades económicas y con los resultados de sus decisiones

4.16.2. Informe Financiero

Toda la información contable dentro de una organización es valiosa para la administración. El proceso de suministrar información financiera de propósito general para personas fuera de la organización es llamado "Informe Financiero" *[Meigs, 2009].*

Estados Financieros

Es el principal medio para reportar información financiera de propósito general. Las personas que reciben estos informes son llamados usuarios de Estados Financieros.

Un paquete de Estado Financiero incluye:

Balance General: Muestra los datos específicos de la empresa para indicar los recursos que poseen, las obligaciones que debe y el monto del capital propio en el negocio *[Meigs, 2009].*

Un Estado de Resultado: Indica la rentabilidad del negocio con relación al año anterior (u otro periodo) *[Meigs, 2009].*

Un flujo de efectivos: Resume el efectivo recibido y los pagos del negocio respecto del mismo periodo cubierto por el estado de resultados *[Meigs, 2009].*

El propósito fundamental de los Estados Financieros es ayudar a los usuarios en la evaluación de la posición financiera, la rentabilidad y los proyectos futuros de una empresa.

Informe del Patrimonio del Balance General.

- ✓ Negocios de un solo propietario: Incluye un solo rubro, el del dueño (propietario). Ejemplo:

Patrimonio	
Capital de PMCSIS ……	$ 800.00

- ✓ Sociedades de Personas: Esta tiene dos o más propietarios y se relacionan separadamente por cada socio, la cantidad del patrimonio social en la empresa. Ejemplo:

Patrimonio	
Capital de PMCSIS ……	$ 800.00
Capital de Otra maestría……	$700.00
Total de Patrimonio	$15,000 .00

CICLO CONTABLE

Sistema Contable Computarizado – Personal

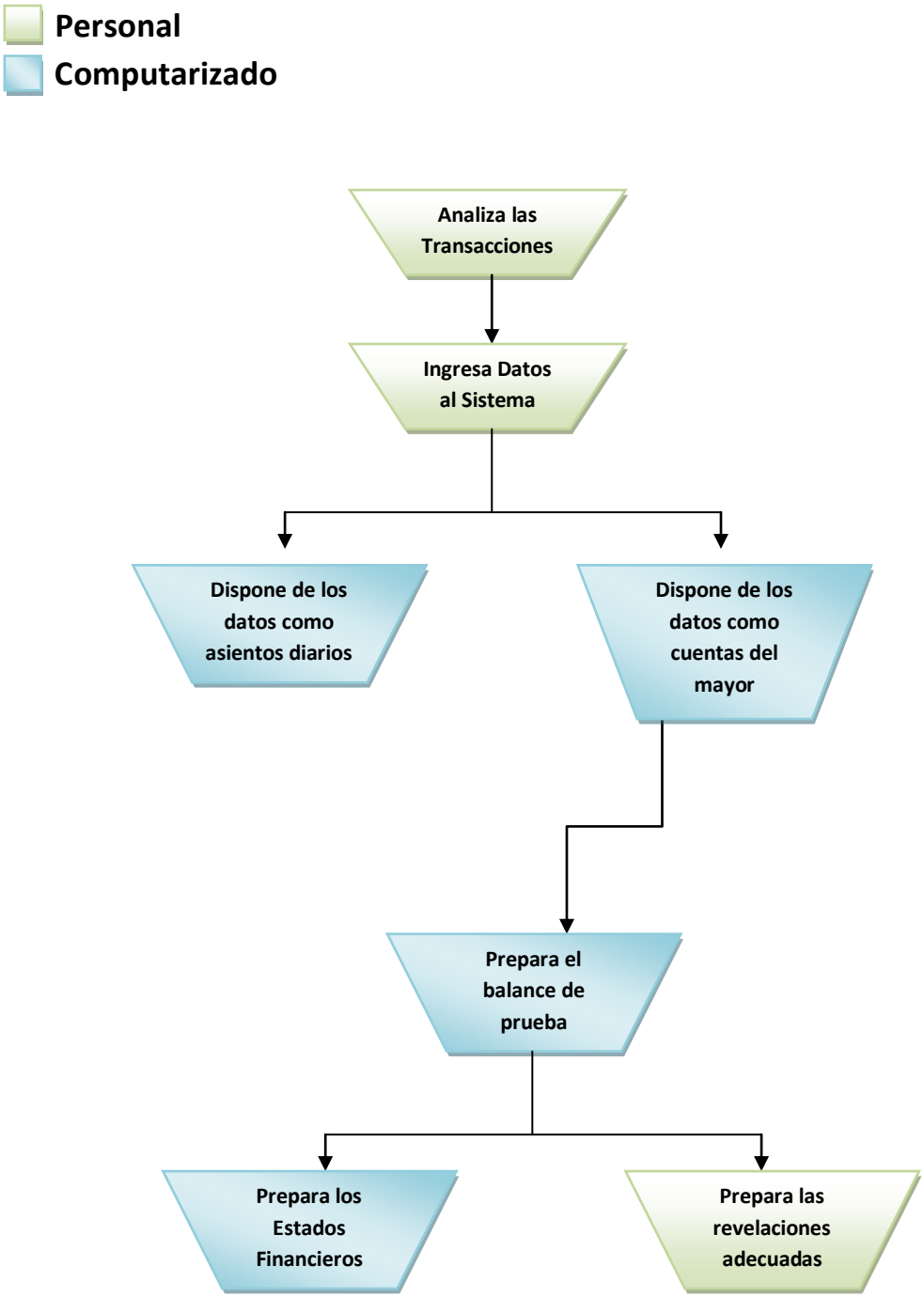


Figura 23 – Ciclo contable

4.17. Conceptos Estadísticos

4.17.1. Estadística

La estadística es comúnmente considerada como una colección de hechos numéricos expresados en términos de una relación sumisa, y que han sido recopilados a partir de otros datos numéricos *[Meigs, 2009]*..

Datos Estadísticos

Los datos estadísticos no son otra cosa que el producto de las observaciones efectuadas en las personas y objetos en los cuales se produce el fenómeno que queremos estudiar. Dicho en otras palabras, son los antecedentes (en cifras) necesarios para llegar al conocimiento de un hecho o para reducir las consecuencias de este.

Clasificación de los datos

Los datos estadísticos pueden ser clasificados en cualitativos, cuantitativos, cronológicos y geográficos *[Meigs, 2009]*.

- ✓ **Datos Cualitativos:** cuando los datos son cuantitativos, la diferencia entre ellos es de clase y no de cantidad.
- ✓ **Datos cuantitativos:** cuando los valores de los datos representan diferentes magnitudes, decimos que son datos cuantitativos.
- ✓ **Datos cronológicos:** cuando los valores de los datos varían en diferentes instantes o períodos de tiempo, los datos son reconocidos como cronológicos.
- ✓ **Datos geográficos:** cuando los datos están referidos a una localidad geográfica se dicen que son datos geográficos.

Términos Estadísticos

Población: Es el conjunto de elementos, individuos o entes sujetos a estudio y de los cuales queremos obtener un resultado.

Variable: Es la característica que estamos midiendo.

Muestra: Conjunto de elementos que forman parte de población. La muestra representa a esta población.

Dato: Cada uno de los individuos, cosas, entes abstractos que integran una población o universo determinado. Dicho de otra forma, cada valor observado de la variable.

Porcentaje

La palabra porcentaje, como indica su nombre, se refiere al número de partes que nos interesan de un total de 100.

4.17.2. Cuadro Estadístico

Llamado también "tabla estadística", es un instrumento que sirve para presentar los resultados de la conceptualización y cuantificación de ciertos aspectos particulares de la realidad. Como tal, es el ámbito descriptivo que establece las relaciones e interconexiones posibles que existen entre los conceptos que se cuantifican y las magnitudes que adquieren unos con respecto a otros.

Se define también como el conjunto de datos estadísticos ordenados en columnas y filas, que permite leer, comparar e interpretar las características de una o más variables. Los datos son el resultado de la ejecución de una investigación estadística o el aprovechamiento de un registro administrativo con fines estadísticos *[Meigs, 2009]*.

Elementos de un cuadro estadístico

Todo cuadro estadístico debe tener un título, encabezamiento, columna matriz, cuerpo y pie. Así, los elementos de un cuadro estadístico son en detalle:

- ✓ Código o número de cuadro
- ✓ Título
- ✓ Encabezamiento
- ✓ Listado de clasificaciones
- ✓ Cuerpo
- ✓ Pie

La información que se muestra a continuación fue tomada del sitio web del programa de Maestría.

4.18. Descripción del Programa de Maestría en Computación

La UNAN-Managua en cooperación con el Instituto Tecnológico de Costa Rica (ITCR), ofrece la Maestría en Computación con énfasis en Sistemas de Información.

El programa tiene una cobertura nacional con sede central ubicada en la Facultad de Ciencias de la UNAN-MANAGUA, Recinto Universitario Rubén Darío.

Para el desarrollo del programa se cuenta con personal docente altamente capacitado y experimentado, integrado por prestigiosos académicos nacionales y extranjeros con vasta experiencia profesional en el área de los sistemas de información.

Objetivo

Para los próximos años, el PMC se ve como un programa sólido y atractivo, comprometido con el desarrollo y el progreso tecnológico del país, a través de la formación de profesionales altamente calificados en el área de la computación y con resultados visibles en el incremento de la competitividad y productividad de las diferentes Empresas, Instituciones, Ministerios y Entes Autónomos, entre otros.

Misión

Desarrollar en los participantes del programa, la capacidad para dirigir y gestionar proyectos de interés nacional en el área de la computación. Así como, realizar investigaciones para la construcción, implementación y ejecución de proyectos informáticos, con un alto grado de confiabilidad y efectividad.

Proceso de Admisión para aplicar a Maestrías

Los aspirantes a Maestría deben realizar el siguiente proceso para ser admitidos:

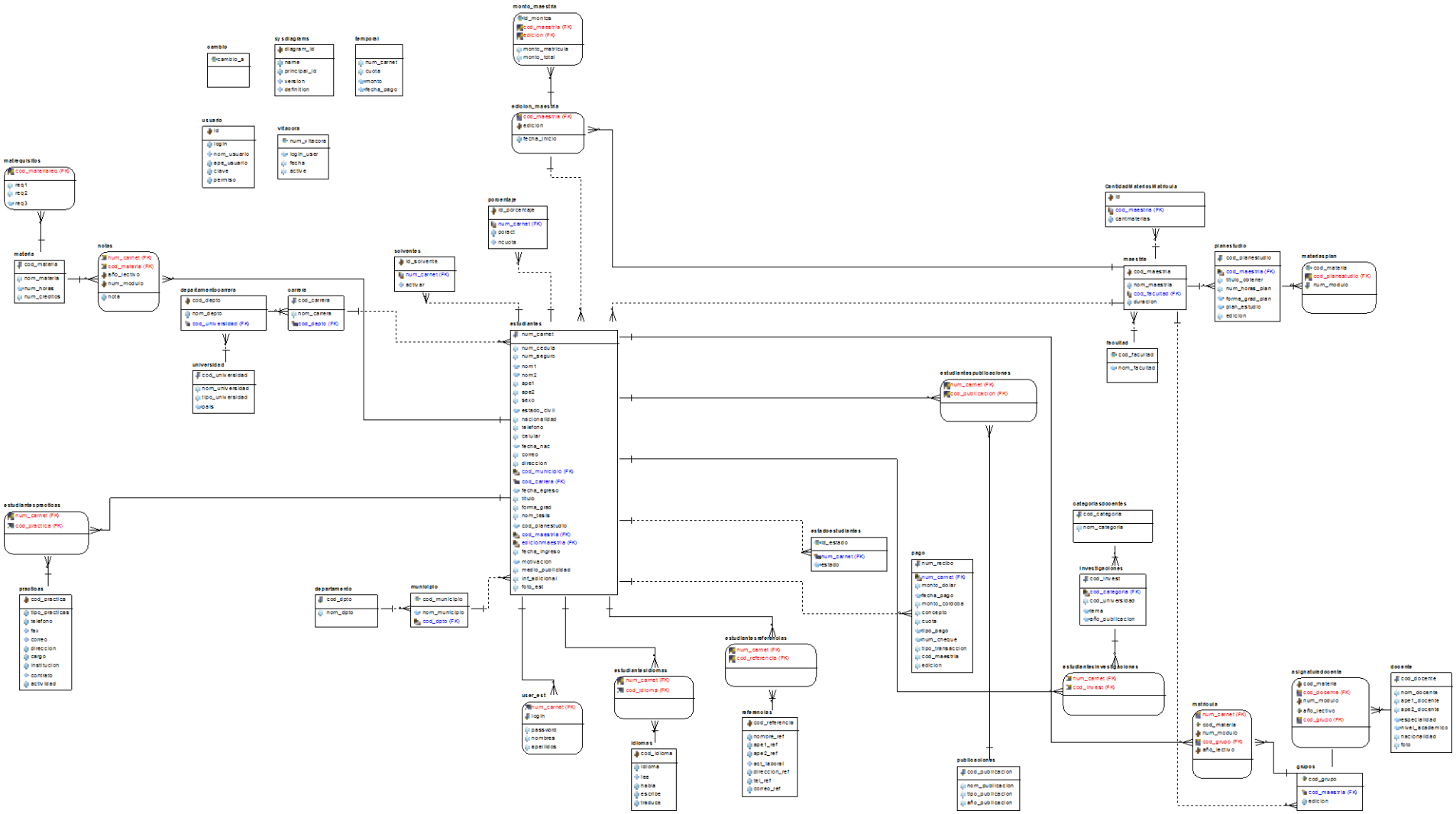
- Revisar requisitos de admisión y/o postulación
- Crear una cuenta de usuario (si hace la postulación en línea)
- Llenar formulario de solicitud
- Revisar resultados de la comisión.

V. Resultados

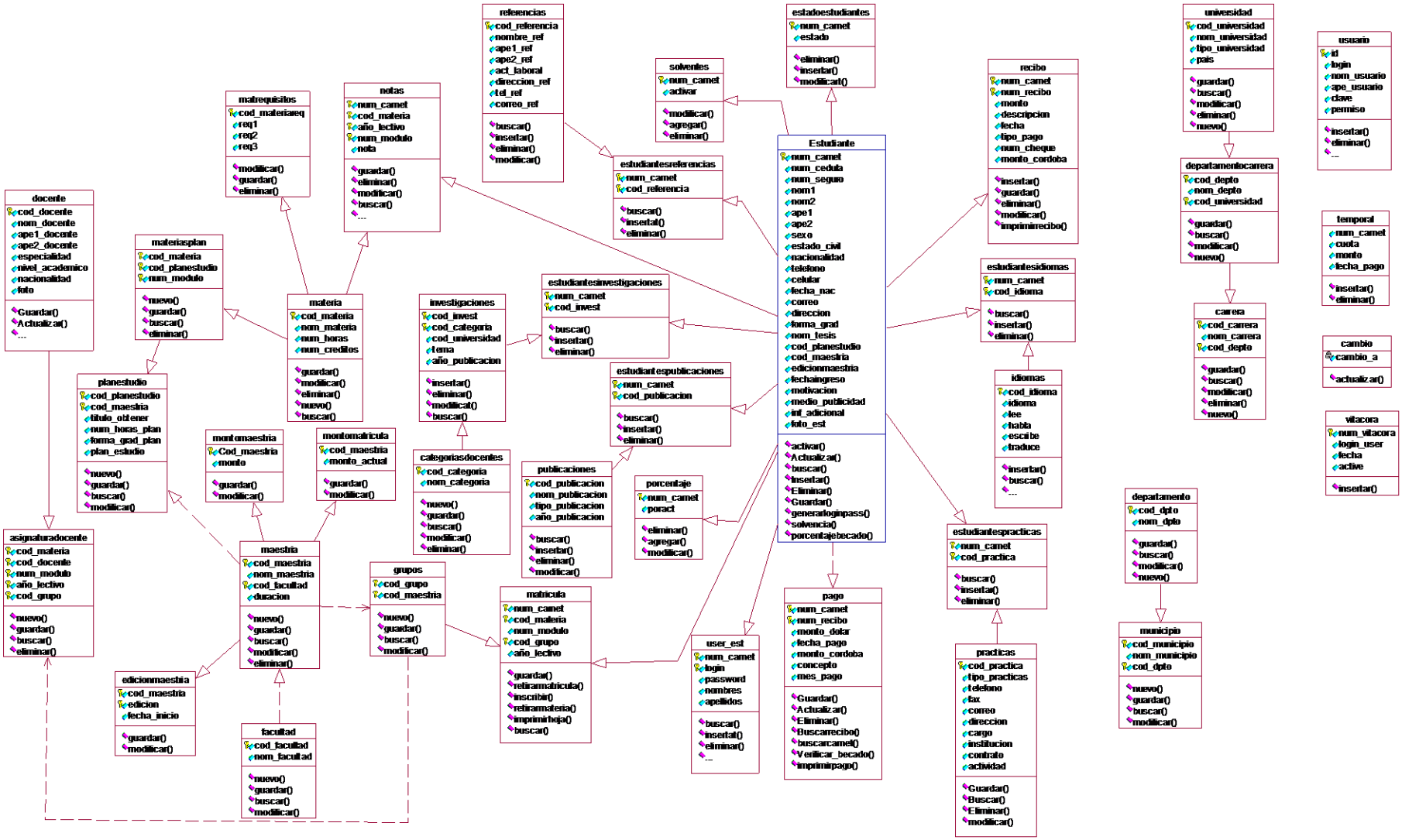
En este apartado se muestra los resultados obtenidos durante el desarrollo del sistema, a continuación se enumeran cada uno de los resultados:

- Base de datos normalizada
- Diagrama de clase
- Diagrama de casos de uso
- Diagrama de secuencia
- Diagrama de colaboración
- Diseño de pantallas del sistema PMCSIS
- Reportes
- Diseño de pantallas del Sitio web de Consultas Calificaciones y pagos
- Prueba d unidad
- Prueba de integración
- Prueba de entrega
- Métricas

Resultado 1 - Base de datos normalizada

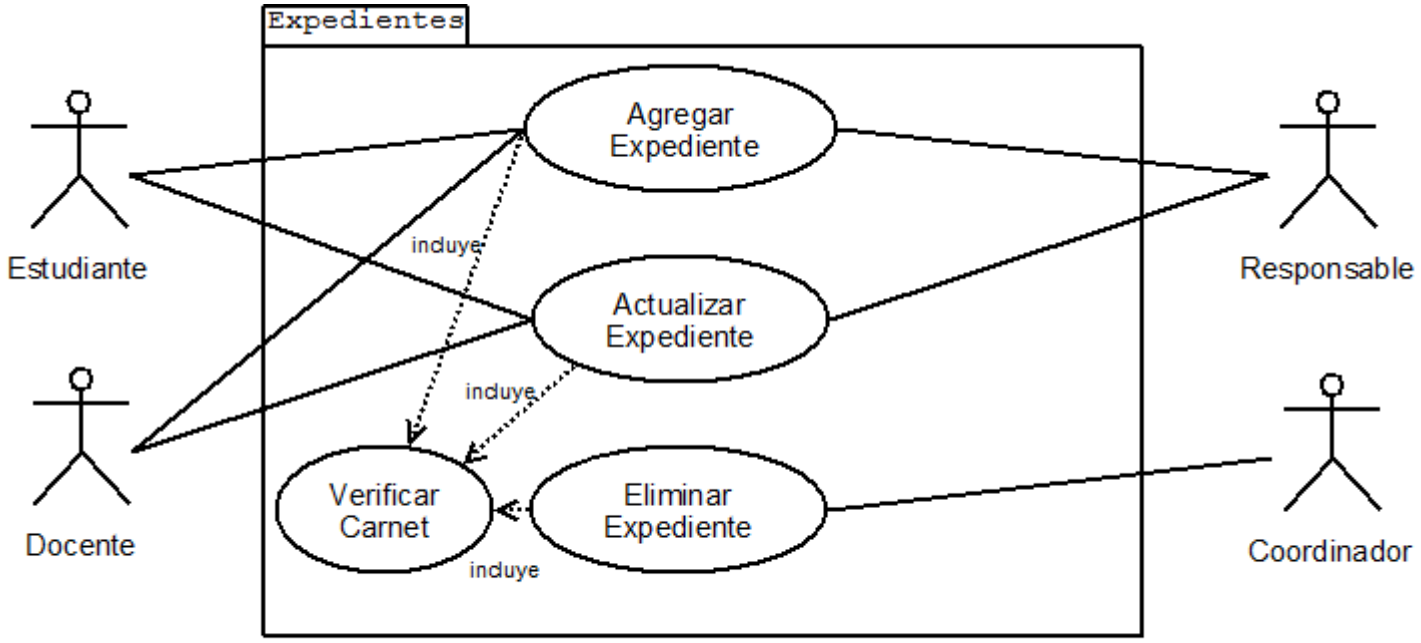


Resultado 2 – Diagrama de clases

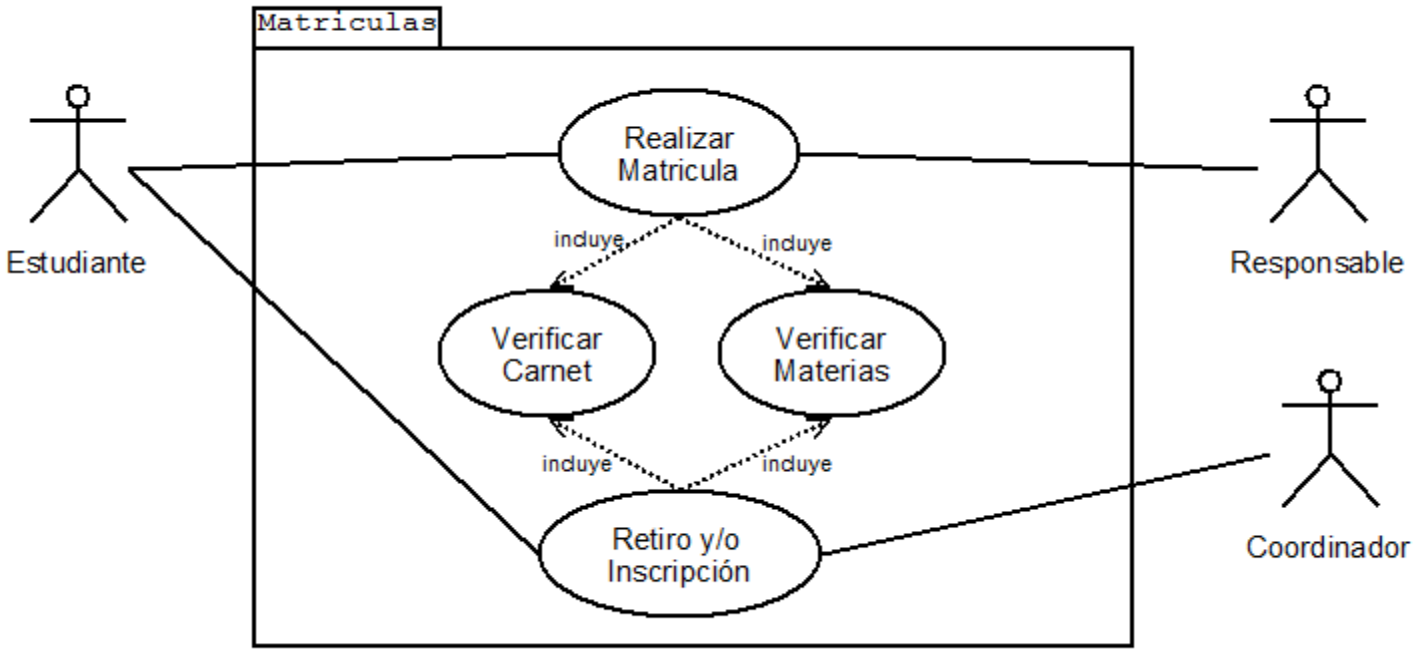


Resultado 3 - Diagrama de casos de uso

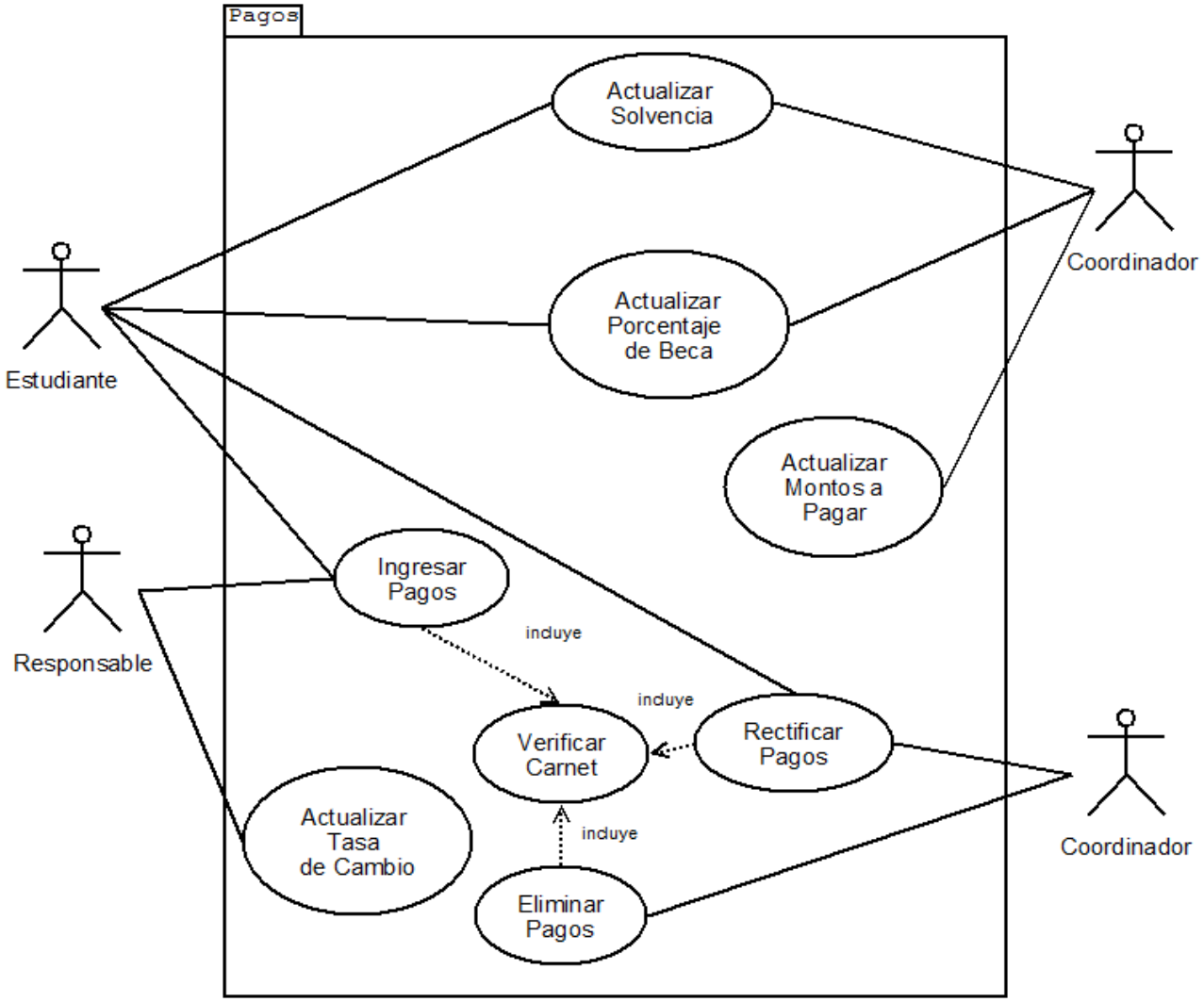
1 – DIAGRAMA DE EXPEDIENTES



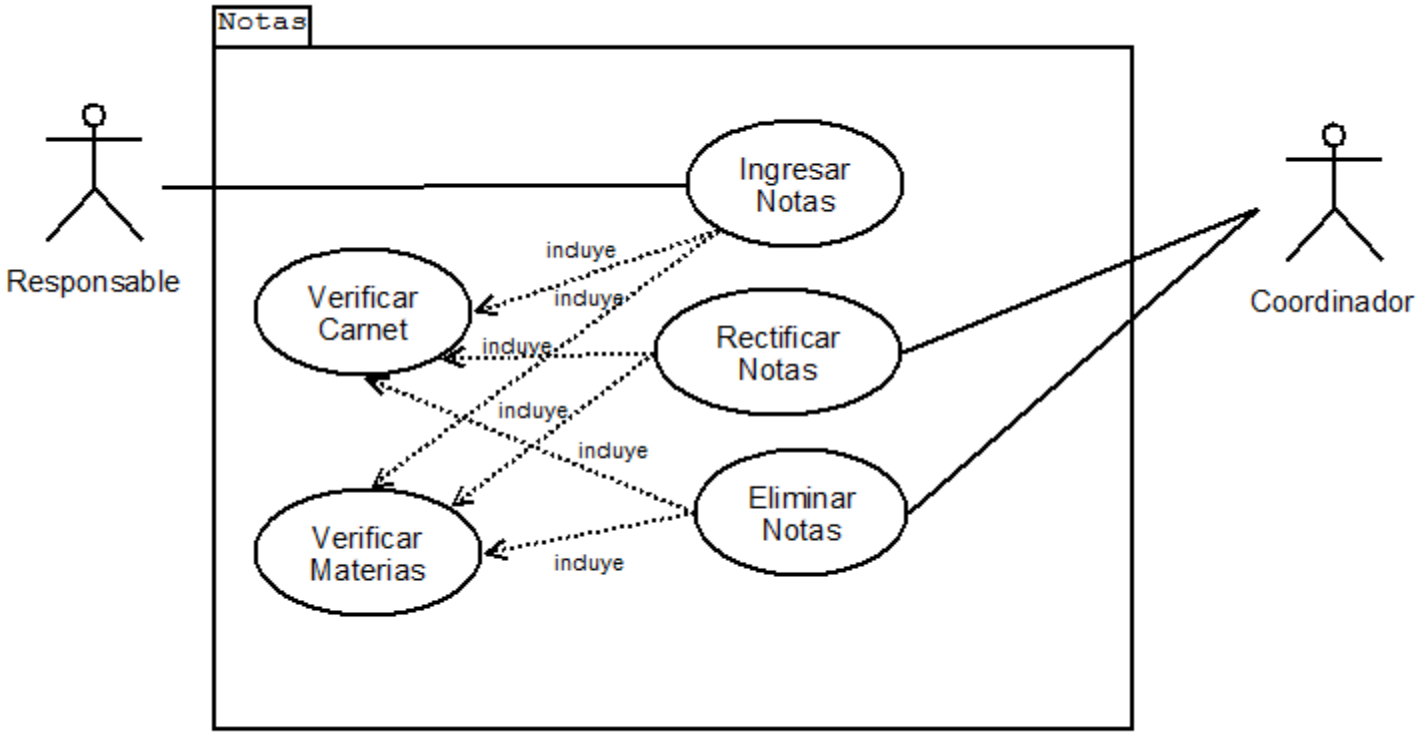
2 – DIAGRAMA DE MATRICULAS



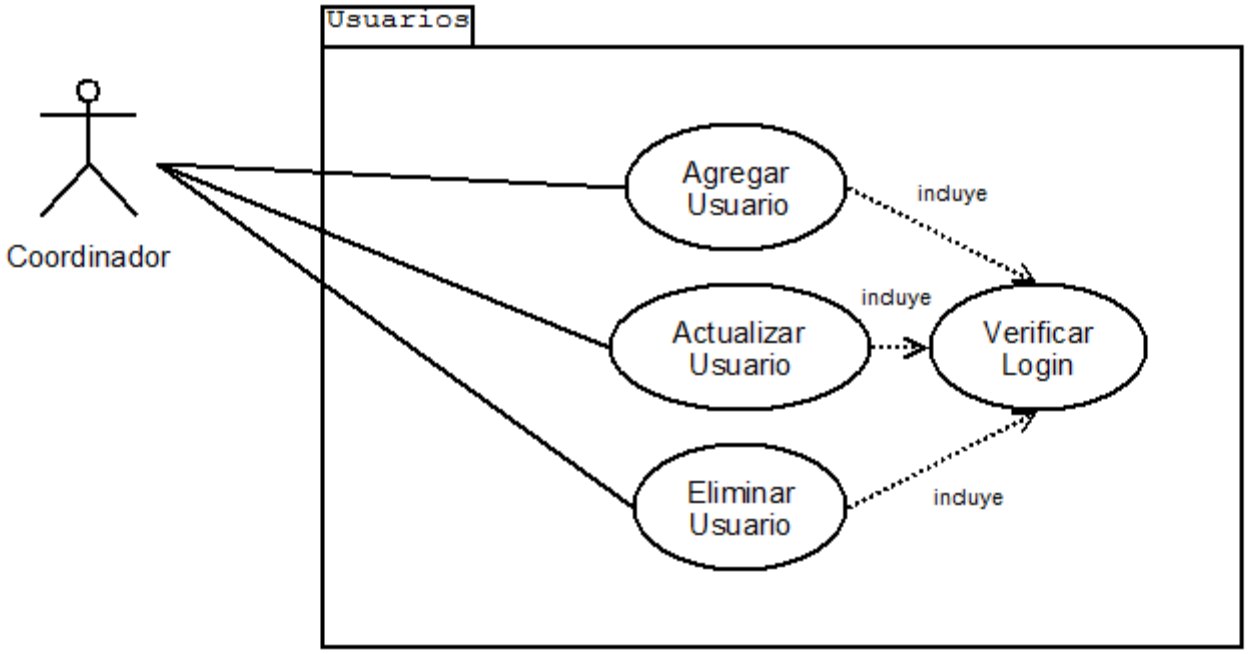
3 – DIAGRAMA DE PAGOS



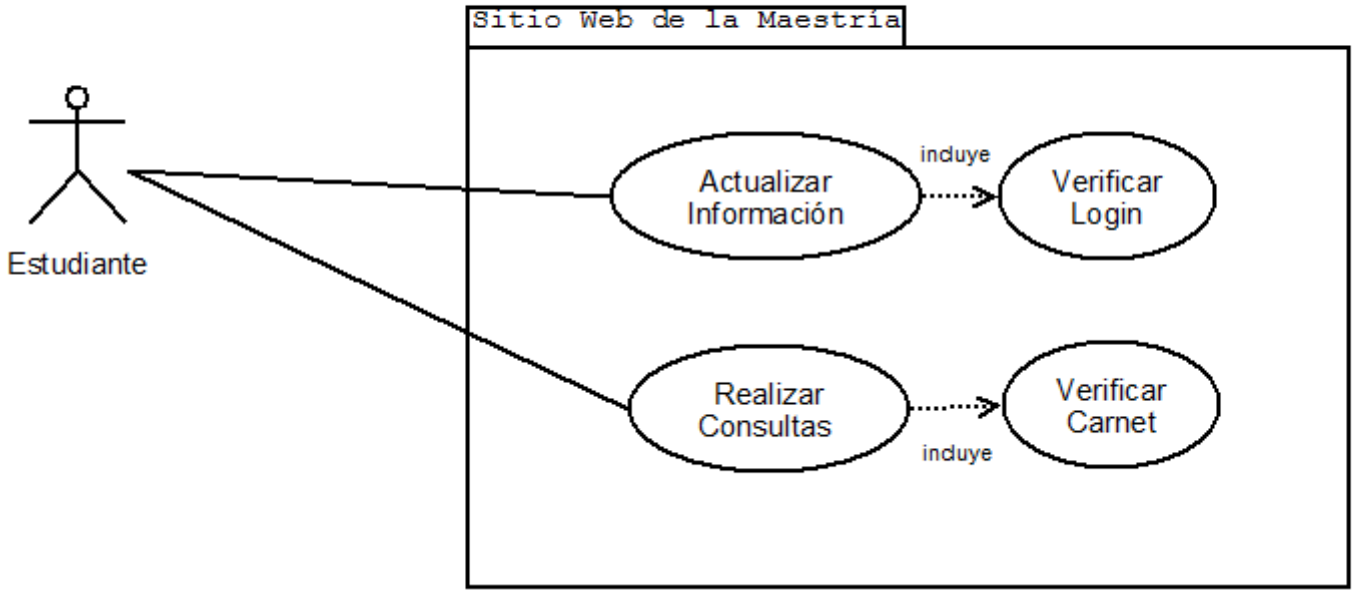
4– DIAGRAMA DE NOTAS



5 – DIAGRAMA DE USUARIOS



6 – DIAGRAMA DE SITIO WEB



7- ESCENARIOS DE CASOS DE USOS

UC-0001		**Nombre del caso de uso: *Agregar expediente***
Actor(es): Estudiante, Docente, Responsable		
Descripción: Cuando el Estudiante y el Docente ingresan por primera vez a la maestría se le toman ciertos datos básicos, para crearle un expediente. Como su cédula, nombre, dirección, teléfono.		
Activar el evento: El *actor Responsable* ingresa al sistema, hace clic en la pestaña registro, luego selecciona el expediente que desea crear, ya sea de Estudiante o docente.		
Tipo: Primario y esencial		
Paso	**Acción**	
1	El sistema inicia este caso de uso cuando el ***actor Responsable*** ingresa en la pestaña registro y selecciona Estudiante o Docente.	
2	El ***actor Responsable*** ingresa el número de identificación del Estudiante o Docente	
3	El sistema valida que el número de identificación sea nuevo, que no esté registrado en el sistema.	
4	El ***actor Responsable*** ingresa los datos básicos como: nombre, nacionalidad, cédula, dirección, teléfono.	
5	El sistema valida que los datos ingresados estén correctos y/o completos.	
6	El sistema registra al Estudiante o Docente.	
Precondición: El *actor Responsable* debe estar conectado al sistema.		
Poscondición: Se ha creado un nuevo registro de expediente.		
Excepciones	**Paso**	**Acción**
	3	Si el número de identificación ya está registrado, el sistema envía un mensaje, a continuación el caso de uso queda sin efecto.
	5	Si los datos ingresados no están correctos y/o completos el sistema verificara dato por dato y lo devolverá al paso 4, el caso de uso continúa.
	6	En caso de ser un Estudiante el sistema le creara una cuenta para consultar sus notas y pagos al sitio web de la maestría.
Suposiciones: El *actor Responsable* tiene un usuario y una contraseña valida.		

UC-0002	Nombre del caso de uso: *Actualizar expediente*	
Actor(es): Estudiante, Docente, Responsable		
Descripción: Cuando se desea modificar datos del expediente de un Estudiante o Docente, el sistema se comportará de la siguiente manera.		
Activar el evento: El *actor Responsable* ingresa al sistema, hace clic en la pestaña registro, luego selecciona el tipo de expediente que desea actualizar, ya sea de Estudiante o docente, da clic en el botón buscar, luego ingresa el número de identificación y presiona Enter, por ultimo hace clic en el botón modificar.		
Tipo: Primario y esencial		
Paso	**Acción**	
1	El sistema inicia este caso de uso cuando el ***actor Responsable*** ingresa en la pestaña registro y selecciona Estudiante o Docente.	
2	El ***actor Responsable*** ingresa el número de identificación del Estudiante o Docente.	
3	El sistema valida que el número de identificación exista.	
4	El sistema muestra los datos del expediente.	
5	El sistema activa los datos para que puedan ser modificados.	
6	El ***actor Responsable*** modifica los datos.	
7	El sistema valida que los datos sean correctos y/o completos.	
8	El sistema actualiza el registro del Estudiante o Docente.	
Precondición: El *actor Responsable* debe estar conectado al sistema.		
Poscondición: Expediente actualizado en el sistema.		
Excepciones	**Paso**	**Acción**
	3	Si el número de identificación no está registrado, el sistema envía un mensaje y lo devolverá al paso 2, el caso de uso continúa.
	7	Si los datos ingresados no están correctos y/o completos el sistema verificara dato por dato y lo devolverá al paso 6, el caso de uso continúa.
Suposiciones: El *actor Responsable* tiene un usuario y una contraseña valida.		

UC-0003	Nombre del caso de uso: *Eliminar expediente*	
Actor(es): Coordinador		
Descripción: Cuando se desea eliminar un expediente, el sistema se comportará de la siguiente manera.		
Activar el evento: El *actor Coordinador* ingresa al sistema, hace clic en la pestaña registro, luego selecciona el tipo de expediente que desea actualizar, ya sea de Estudiante o docente, da clic en el botón buscar, luego ingresa el número de identificación y presiona Enter, por ultimo hace clic en el botón eliminar.		
Tipo: Primario y esencial		
Paso	**Acción**	
1	El sistema inicia este caso de uso cuando el ***actor Coordinador*** ingresa en la pestaña registro y selecciona Estudiante o Docente.	
2	El ***actor Coordinador*** ingresa el número de identificación del Estudiante o Docente.	
3	El sistema valida que el número de identificación exista.	
4	El sistema muestra los datos del expediente.	
5	El ***actor Coordinador*** da clic en eliminar.	
6	El sistema envía un mensaje para verificar la eliminación.	
7	El ***actor Coordinador*** acepta.	
8	El sistema elimina el expediente.	
Precondición: El *actor Coordinador* debe estar conectado al sistema y cuenta con los permisos necesarios para la eliminación.		
Poscondición: Expediente eliminado en el sistema.		
Excepciones	**Paso**	**Acción**
	3	Si el número de identificación no está registrado, el sistema envía un mensaje y lo devolverá al paso 2, el caso de uso continúa.
	7	Si el ***actor Coordinador*** cancela, a continuación el caso de uso queda sin efecto.
Suposiciones: El *actor Coordinador* tiene un usuario y una contraseña valida.		

UC-0004		Nombre del caso de uso: *Realizar matricula*
Actor(es): Estudiante, Responsable		
Descripción: El sistema deberá comportarse tal como se describe en siguiente caso de uso cuando el *actor Responsable* desea realizar una matrícula.		
Activar el evento: El *actor Responsable* ingresa al sistema, selecciona la pestaña registro, da clic en matricula, luego da clic en nuevo.		
Tipo: Primario y esencial		
Paso	**Acción**	
1	El sistema inicia este caso de uso cuando el ***actor Responsable*** selecciona la pestaña registro, da clic en matricula, luego en da clic en nuevo.	
2	El ***actor Responsable*** ingresa el número de carnet del Estudiante.	
3	El sistema valida que el número de carnet este registrado.	
4	El sistema verifica que el Estudiante de nuevo ingreso se haya efectuado un pago para la matricula.	
5	El sistema muestra la lista de materias que el Estudiante puede matricular de acuerdo al plan de estudio entre los datos que muestra están: código, nombre, horas, creditos.	
6	El sistema le asigna el grupo de estudio.	
7	El sistema registra la matricula	
Precondición: El *actor Responsable* debe estar conectado al sistema.		
Poscondición: Una nueva matricula registrada en el sistema.		
Excepciones	**Paso**	**Acción**
	3	Si el número de identificación ya está registrado, el sistema envía un mensaje y lo devolverá al paso 2, el caso de uso continúa.
	5	El sistema verifica si el Estudiante ya ha matriculado todos los módulos disponibles, le envía un mensaje, en este caso si el Estudiante debe materias, entonces se trasladaría al caso de uso ***UC-0005 Retiro y/o Inscripción.***
	6	El sistema verifica que un grupo esté relacionado a un modulo, si no existe un grupo relacionado a un modulo, le envía un mensaje, a continuación el caso de uso queda sin efecto.
Suposiciones: El *actor Responsable* tiene un usuario y una contraseña valida.		

UC-0005		Nombre del caso de uso: *Retiro y/o Inscripción*
Actor(es): Estudiante, Coordinador		
Descripción: El sistema deberá comportarse tal como se describe en siguiente caso de uso cuando el *actor Coordinador* desea realizar un retiro y/o inscripción.		
Activar el evento: El *actor Coordinador* ingresa al sistema, selecciona la pestaña registro, da clic en matricula, luego da clic en retirar o inscribir.		
Tipo: Primario y esencial		
Paso	**Acción**	
1	El sistema inicia este caso de uso cuando el ***actor Coordinador*** selecciona la pestaña registro, da clic en matricula, luego en da clic en : a) Inscribir ***véase la Sección Realizar Inscripción*** b) Retirar ***véase la Sección Realizar Retiro.***	
Sección: Realizar Inscripción		
Paso	**Acción**	
1	El ***actor Coordinador*** ingresa el número de carnet del Estudiante.	
2	El sistema valida que el número de carnet este registrado.	
3	El sistema muestra la lista de materias que puede inscribir de acuerdo al plan de estudio.	
4	El ***actor Coordinador*** selecciona las materias	
5	El sistema registra la inscripción.	
Excepciones	**Paso**	**Acción**
	2	Si el número de identificación no está registrado, el sistema envía un mensaje y lo devolverá al paso 1, el caso de uso continúa.
	3	Si el Estudiante no tiene materias que inscribir, le envía un mensaje, a continuación el caso de uso queda sin efecto.
Sección: Realizar Retiro		
Paso	**Acción**	
1	El ***actor Coordinador*** ingresa el número de carnet del Estudiante.	
2	El sistema valida que el número de carnet este registrado.	
3	El sistema muestra la lista de materias que puede retirar de acuerdo al plan de estudio y que no tengan notas registradas.	
4	El ***actor Coordinador*** selecciona las materias	
5	El sistema registra el retiro de las materias.	
Excepciones	**Paso**	**Acción**
	2	Si el número de identificación no está registrado, el sistema envía un mensaje y lo devolverá al paso 1, el caso de uso continúa.
	3	El sistema verificara si el Estudiante puede retirar materias, sino puede le enviará un mensaje, a continuación el caso de uso queda sin efecto.
	4	En caso de que el Estudiante desee ***Retirar Matricula*** el ***actor Coordinador*** debe de dar clic en el botón retirar matricula, siempre y cuando el sistema lo permita.
Precondición: El *actor Coordinador* debe estar conectado al sistema y cuenta con los permisos necesarios para la inscripción y/o retiro.		
Poscondición: Una Inscripción y/o Retiro registrado en el sistema.		
Suposiciones: El *actor Coordinador* tiene un usuario y una contraseña valida.		

UC-0006	Nombre del caso de uso: *Ingresar notas*	
Actor(es): Responsable		
Descripción: Cuando se desea ingresar las notas de los Estudiantes el sistema se comportará de la siguiente manera.		
Activar el evento: El *actor Responsable* primero ingresa al sistema en donde selecciona la pestaña registro, da clic en el botón calificaciones, luego da clic en nuevo, selecciona como va ingresar las notas.		
Tipo: Primario y esencial		
Paso	**Acción**	
1	El sistema inicia este caso de uso cuando el ***actor Responsable*** ingresa en la pestaña registro y da clic en el botón notas.	
2	El ***actor Responsable*** da clic en botón nuevo.	
3	El ***actor Responsable*** selecciona como ingresara las notas: *a)* Si desea ingresar las notas de forma individual, *véase la sección ingresar notas de forma individual.* b) Si desea ingresar notas por grupo, *véase la sección ingresar notas por grupo*	
4	El ***actor Responsable*** ingresa la nota.	
5	El sistema verifica las notas.	
6	El sistema registra la nota.	
Sección: *Ingresar notas de forma individual*		
Paso	**Acción**	
1	El ***actor Responsable*** selecciona ingresar notas de forma individual.	
2	El ***actor Responsable*** ingresa el número de carnet del Estudiante.	
3	El sistema verifica si el número de carnet existe.	
4	El ***actor Responsable*** selecciona el código de la materia e ingresa el año lectivo.	
5	El sistema verifica que el Estudiante tenga matriculada esa materia.	
6	El sistema muestra el número de carnet, nombre y apellidos del Estudiante.	
Excepciones	**Paso**	**Acción**
	3	Si el número de carnet no está registrado, el sistema envía un mensaje y lo devolverá al paso 2, el caso de uso continúa.
	5	Si el Estudiante no tiene matriculada esa materia, el sistema envía un mensaje, a continuación el caso de uso queda sin efecto.
Sección: *Ingresar notas por grupo*		
Paso	**Acción**	
1	El ***actor Responsable*** selecciona ingresar notas por grupo.	
2	El ***actor Responsable*** selecciona el código y la edición de la maestría.	
3	El ***actor Responsable*** selecciona el código de la materia e ingresa el año lectivo.	
4	El sistema verifica que el Estudiante tenga matriculada esa materia.	
5	El sistema selecciona los Estudiantes que tienen matricula la materia y muestra el número de carnet, nombre y apellidos de los Estudiantes.	
Excepciones	**Acción**	
	3	Si no existen Estudiantes con la materia matriculada en ese año lectivo, el sistema le envía un mensaje, a continuación el caso de uso queda sin efecto.
Precondición: El *actor Responsable* debe estar conectado al sistema.		
Poscondición: Se ha creado un nuevo registro de notas.		
Excepciones	**Paso**	**Acción**
	5	Si los datos ingresados no están correctos y/o completos el sistema verificara dato por dato y lo devolverá al paso 4, el caso de uso continúa.
Suposiciones: El *actor Responsable* tiene un usuario y una contraseña valida.		

UC-0007		Nombre del caso de uso: *Rectificar notas*
Actor(es): Coordinador		
Descripción: Cuando se desea rectificar la nota de un Estudiante el sistema se comportara de la siguiente manera.		
Activar el evento: El *actor Coordinador* selecciona la pestaña registro, da clic en calificaciones, luego da clic en buscar, ingresa el número de carnet, código de materia, año lectivo y por ultimo da clic en modificar.		
Tipo: Primario y esencial		
Paso	**Acción**	
1	El sistema inicia este caso de uso cuando el ***actor Coordinador*** selecciona la pestaña registro, luego da clic en calificaciones y por ultimo da clic en buscar.	
2	El ***actor Coordinador*** ingresa el número de carnet del Estudiante.	
3	El sistema valida que exista el número de carnet.	
4	El ***actor Coordinador*** selecciona el código de materia y año lectivo.	
5	El sistema valida que el Estudiante tenga esa nota registrada.	
6	El sistema muestra los datos.	
7	El ***actor Coordinador*** modifica la nota.	
8	El sistema verifica que los datos estén correctos y/o completos.	
	El sistema actualiza el registro de nota.	
Precondición: El *actor Coordinador* debe estar conectado al sistema y cuenta con los permisos necesarios para rectificar las notas.		
Poscondición: Se ha actualizado el registro de notas.		
Excepciones	**Paso**	**Acción**
	3	Si el número de identificación ya está registrado, el sistema envía un mensaje y lo devolverá al paso 2, el caso de uso continúa.
	5	Si el Estudiante no tiene esa nota registrada el sistema le envía un mensaje, a continuación el caso de uso queda sin efecto.
	8	Si los datos ingresados no están correctos y/o completos el sistema verificara dato por dato, le enviará un mensaje y lo devolverá al paso 7, el caso de uso continúa.
Suposiciones: El *actor Coordinador* tiene un usuario y una contraseña valida.		

UC-0008	Nombre del caso de uso: *Eliminar notas*	
Actor(es): Coordinador		
Descripción: Cuando se desea eliminar una nota, el sistema se comportará de la siguiente manera.		
Activar el evento: El *actor Coordinador* selecciona la pestaña registro, da clic en calificaciones, luego da clic en buscar, ingresa el número de carnet, código de materia, año lectivo y por ultimo da clic en eliminar.		
Tipo: Primario y esencial		
Paso	**Acción**	
1	El sistema inicia este caso de uso cuando el ***actor Coordinador*** selecciona la pestaña registro, luego da clic en calificaciones y por ultimo da clic en buscar.	
2	El ***actor Coordinador*** ingresa el número de carnet del Estudiante.	
3	El sistema valida que exista el número de carnet.	
4	El ***actor Coordinador*** selecciona el código de materia y año lectivo.	
5	El sistema valida que el Estudiante tenga esa nota registrada.	
6	El sistema muestra los datos.	
7	El ***actor Coordinador*** da clic en eliminar.	
8	El sistema envía un mensaje para verificar la eliminación	
9	El ***actor Coordinador*** acepta.	
10	El sistema elimina el registro de nota.	
Precondición: El *actor Coordinador* debe estar conectado al sistema y cuenta con los permisos necesarios para la eliminación.		
Poscondición: Registro de nota eliminado.		
Excepciones	**Paso**	**Acción**
	3	Si el número de identificación no está registrado, el sistema envía un mensaje y lo devolverá al paso 2, el caso de uso continúa.
	5	Si el Estudiante no tiene esa nota registrada, el sistema le envía un mensaje, a continuación el caso de uso queda sin efecto.
	9	Si el ***actor Coordinador*** cancela, a continuación el caso de uso queda sin efecto.
Suposiciones: El *actor Coordinador* tiene un usuario y una contraseña valida.		

UC-0009		Nombre del caso de uso: *Actualizar tasa de cambio*
Actor(es): Responsable		
Descripción: Cuando se desee actualizar la tasa de cambio C$ Córdoba - $ Dólar, el caso de uso se comportará de la siguiente manera.		
Activar el evento: El *actor Responsable* ingresa al sistema, hace clic en la pestaña utilidades, luego da clic en *Cambiar valor del Dólar.*		
Tipo: Primario y esencial		
Paso	**Acción**	
1	El sistema inicia este caso de uso cuando el ***actor Responsable*** selecciona en la pestaña utilidades *Cambiar valor del Dólar*.	
2	El sistema le muestra el valor actual.	
3	El ***actor Responsable*** ingresa el nuevo monto.	
4	El sistema registra el nuevo valor del Dólar.	
Precondición: El *actor Responsable* debe estar conectado al sistema.		
Poscondición: Se ha actualizado el valor del Dólar.		
Excepciones	**Paso**	**Acción**
	2	El sistema por defecto le asigna cero a la tasa de cambio, cuando crea el registro de la maestría.
Suposiciones: El *actor Responsable* tiene un usuario y una contraseña valida.		

UC-0010		Nombre del caso de uso: *Actualizar montos a pagar*
Actor(es): Coordinador		
Descripción: Cuando se desee actualizar montos a pagar de la maestría y la matricula, el caso de uso se comportará de la siguiente manera.		
Activar el evento: El *actor Coordinador* ingresa al sistema, hace clic en la pestaña utilidades, luego da clic en *Cambiar montos de maestría.*		
Tipo: Primario y esencial		
Paso	**Acción**	
1	El sistema inicia este caso de uso cuando el ***actor Coordinador*** selecciona en la pestaña utilidades *Cambiar montos de maestría.*	
2	El sistema le muestra el valor actual.	
3	El ***actor Coordinador*** ingresa el nuevo monto.	
4	El sistema registra el nuevo valor.	
Precondición: El *actor Coordinador* debe estar conectado al sistema y cuenta con los permisos necesarios para actualizar estos datos.		
Poscondición: Se ha actualizado el valor.		
Excepciones	**Paso**	**Acción**
	2	El sistema por defecto le asigna cero al monto de la matricula y de la maestría, cuando crea el registro de la maestría.
Suposiciones: El *actor Coordinador* tiene un usuario y una contraseña valida.		

UC-0011	Nombre del caso de uso: *Actualizar Solvencia*	
Actor(es): Coordinador, Estudiante		
Descripción: Cuando se desee actualizar el estado de un Estudiante si esta solvente o no, el caso de uso se comportará de la siguiente manera.		
Activar el evento: El *actor Coordinador* ingresa al sistema, hace clic en la pestaña utilidades, luego da clic en *Actualizar solvencia de pago*.		
Tipo: Primario y esencial		
Paso	**Acción**	
1	El sistema inicia este caso de uso cuando el ***actor Coordinador*** selecciona en la pestaña utilidades *Actualizar solvencia de pago.*	
2	El ***actor Coordinador*** ingresa el número de carnet del Estudiante.	
3	El sistema verifica si existe ese registro.	
4	El ***actor Coordinador*** selecciona el estado.	
5	El sistema registra el nuevo estado.	
Precondición: El *actor Coordinador* debe estar conectado al sistema y cuenta con los permisos necesarios para actualizar estos datos.		
Poscondición: Se ha actualizado el estado.		
Excepciones	**Paso**	**Acción**
	3	Si el número de identificación no está registrado, el sistema envía un mensaje y lo devolverá al paso 2, el caso de uso continúa.
	5	El sistema por defecto le asigna al estado del Estudiante *Solvente* crea el registro de estudiante.
Suposiciones: El *actor Coordinador* tiene un usuario y una contraseña valida.		

UC-0012		Nombre del caso de uso: *Actualizar Porcentaje de Beca*
Actor(es): Coordinador, Estudiante		
Descripción: Cuando se desee actualizar el porcentaje de beca de un Estudiante, el caso de uso se comportará de la siguiente manera.		
Activar el evento: El *actor Coordinador* ingresa al sistema, hace clic en la pestaña utilidades, luego da clic en *Actualizar porcentaje de becado.*		
Tipo: Primario y esencial		
Paso	**Acción**	
1	El sistema inicia este caso de uso cuando el ***actor Coordinador*** selecciona en la pestaña utilidades *Actualizar porcentaje de becado.*	
2	El ***actor Coordinador*** ingresa el número de carnet del Estudiante.	
3	El sistema verifica si existe ese registro.	
4	El sistema mostrará el porcentaje actual de beca.	
5	El ***actor Coordinador*** ingresa el nuevo porcentaje.	
6	El sistema registra el nuevo estado.	
Precondición: El *actor Coordinador* debe estar conectado al sistema y cuenta con los permisos necesarios para actualizar estos datos.		
Poscondición: Se ha actualizado el estado.		
Excepciones	**Paso**	**Acción**
	3	Si el número de identificación no está registrado, el sistema envía un mensaje y lo devolverá al paso 2, el caso de uso continúa.
	6	El sistema por defecto le asigna cero al porcentaje de beca cuando crea el registro de estudiante.
Suposiciones: El *actor Coordinador* tiene un usuario y una contraseña valida.		

UC-0013	Nombre del caso de uso: *Ingresar Pagos*	
Actor(es): Responsable, Estudiante		
Descripción: Cuando se desee ingresar el pago de un Estudiante, el caso de uso se comportará de la siguiente manera.		
Activar el evento: El *actor Responsable* ingresa al sistema, selecciona la pestaña registro, luego va a la sección de pagos y da clic en el que desea realizar (Mensualidad, Matricula).		
Tipo: Primario y esencial		
Paso	**Acción**	
1	El sistema inicia este caso de uso cuando el ***actor Responsable*** selecciona en la pestaña registro *Pagos* y luego da clic en: *a)* Matricula *véase la sección ingresar pago de matrícula.* b) Mensualidad *véase la sección ingresar pago de mensualidad.*	
2	El sistema registra el nuevo pago.	
	Sección: Ingresar pago de matricula	
Paso	**Acción**	
1	El ***actor Responsable*** ingresa la fecha de pago.	
2	El sistema verifica que la fecha esta correcta.	
3	El ***actor Responsable*** ingresa el número de carnet.	
4	El sistema verifica que el exista el registro del Estudiante.	
5	El ***actor Responsable*** ingresa los datos del pago como: número de recibo, monto, tipo de pago, concepto.	
6	El sistema valida que los datos estén correctos y/completos.	
Excepciones	**Paso**	**Acción**
	2	Si la fecha es incorrecta el sistema le envía un mensaje y lo devolverá al paso 1, el caso de uso continúa.
	4	Si el número de carnet no existe el sistema le envía un mensaje y lo devolverá al paso 3, el caso de uso continúa.
	6	El sistema verificará dato por dato, si un dato esta incorrecto le envía un mensaje y lo devolverá al paso 5, el caso de uso continúa.
	Sección: Ingresar pago de mensualidad	
1	El ***actor Responsable*** ingresa el número de carnet.	
2	El sistema valida que el número de carnet exista.	
3	El sistema muestra los detalles de pago tales como: monto a pagar, porcentaje de beca, deducción porcentaje de beca, monto pagado, monto restante. Así como de los detalles de los meses pagados.	
4	El ***actor Responsable*** ingresa los datos del pago como: monto, fecha, concepto, mes a pagar.	
5	El sistema valida que los datos estén correctos y/o completos.	
Excepciones	**Paso**	**Acción**
	2	Si el número de carnet no existe el sistema le envía un mensaje y lo devolverá al paso 1, el caso de uso continúa.
	5	El sistema verificará dato por dato, si un dato esta incorrecto le envía un mensaje y lo devolverá al paso 4, el caso de uso continúa.
Precondición: El *actor Responsable* debe estar conectado al sistema.		
Poscondición: Se ha registrado un nuevo pago.		
Suposiciones: El *actor Responsable* tiene un usuario y una contraseña valida.		

UC-0014	Nombre del caso de uso: *Rectificar Pagos*	
Actor(es): Coordinador, Estudiante		
Descripción: Cuando se desee modificar el pago de un Estudiante, el caso de uso se comportará de la siguiente manera.		
Activar el evento: El *actor Coordinador* ingresa al sistema, selecciona la pestaña registro, luego va a la sección de pagos y da clic en el que desea modificar (Mensualidad, Matricula).		
Tipo: Primario y esencial		
Paso	**Acción**	
1	El sistema inicia este caso de uso cuando el ***actor Coordinador*** selecciona en la pestaña registro *Pagos* y luego da clic en: *a)* Matricula *véase la sección Actualizar pago de matrícula.* b) Mensualidad *véase la sección Actualizar pago de mensualidad.*	
2	El sistema actualiza el registro del pago.	
Sección: Actualizar pago de matricula		
Paso	**Acción**	
1	El ***actor Coordinador*** da clic en Buscar.	
2	El ***actor Coordinador*** ingresa el número de carnet del Estudiante.	
3	El sistema valida que el número de carnet exista.	
4	El sistema muestra los datos del pago.	
5	El ***actor Coordinador*** modifica los datos del pago como: número de recibo, monto, tipo de pago, concepto.	
6	El sistema valida que los datos estén correctos y/completos.	
Excepciones	**Paso**	**Acción**
	3	Si el número de carnet no existe el sistema le envía un mensaje y lo devolverá al paso 2, el caso de uso continúa.
	6	El sistema verificará dato por dato, si un dato esta incorrecto le envía un mensaje y lo devolverá al paso 5, el caso de uso continúa.
Sección: Actualizar pago de mensualidad		
1	El ***actor Coordinador*** da clic en Buscar.	
2	El ***actor Coordinador*** ingresa el número de carnet y/o número de recibo.	
3	El sistema valida que el número de carnet y/o número de recibo exista.	
4	El sistema muestra los datos del pago.	
5	El ***actor Coordinador*** modifica los datos del pago como: monto, fecha, concepto, mes a pagar.	
6	El sistema valida que los datos estén correctos y/o completos.	
Excepciones	**Paso**	**Acción**
	3	Si el número de carnet y/o número de recibo no existe el sistema le envía un mensaje y lo devolverá al paso 2, el caso de uso continúa.
	6	El sistema verificará dato por dato, si un dato esta incorrecto le envía un mensaje y lo devolverá al paso 5, el caso de uso continúa.
Precondición: El *actor Coordinador* debe estar conectado al sistema y debe de tener los permisos necesarios para modificar datos.		
Poscondición: Se ha actualizado un registro de pago.		
Suposiciones: El *actor Coordinador* tiene un usuario y una contraseña valida.		

UC-0015	Nombre del caso de uso: *Eliminar Pagos*	
Actor(es): Coordinador		
Descripción: Cuando se desee eliminar el pago de un Estudiante, el caso de uso se comportará de la siguiente manera.		
Activar el evento: El *actor Coordinador* ingresa al sistema, selecciona la pestaña registro, luego va a la sección de pagos y da clic en el que desea eliminar (Mensualidad, Matricula).		
Tipo: Primario y esencial		
Paso	**Acción**	
1	El sistema inicia este caso de uso cuando el ***actor Coordinador*** selecciona en la pestaña registro *Pagos* y luego da clic en: *a)* Matricula *véase la sección Eliminar pago de matrícula.* b) Mensualidad *véase la sección Eliminar pago de mensualidad.*	
2	El sistema elimina el registro del pago.	
	Sección: Elimina pago de matricula	
Paso	**Acción**	
1	El ***actor Coordinador*** da clic en Buscar.	
2	El ***actor Coordinador*** ingresa el número de carnet del Estudiante.	
3	El sistema valida que el número de carnet exista.	
4	El sistema muestra los datos del pago.	
5	El ***actor Coordinador*** da clic en eliminar	
6	El sistema envía un mensaje para verificar la eliminación.	
7	El ***actor Coordinador*** acepta	
Excepciones	**Paso**	**Acción**
	3	Si el número de carnet no existe el sistema le envía un mensaje y lo devolverá al paso 2, el caso de uso continúa.
	7	Si el ***actor Coordinador*** cancela, a continuación el caso de uso queda sin efecto.
	Sección: Actualizar pago de mensualidad	
1	El ***actor Coordinador*** da clic en Buscar.	
2	El ***actor Coordinador*** ingresa el número de carnet y/o número de recibo.	
3	El sistema valida que el número de carnet y/o número de recibo exista.	
4	El sistema muestra los datos del pago.	
5	El ***actor Coordinador*** clic en eliminar.	
6	El sistema envía un mensaje para la verificación de la eliminación.	
7	El ***actor Coordinador*** acepta.	
Excepciones	**Paso**	**Acción**
	3	Si el número de carnet y/o número de recibo no existe el sistema le envía un mensaje y lo devolverá al paso 2, el caso de uso continúa.
	6	Si el ***actor Coordinador*** cancela, a continuación el caso de uso queda sin efecto.
Precondición: El *actor Coordinador* debe estar conectado al sistema y debe de tener los permisos necesarios para eliminar datos.		
Poscondición: Se ha eliminado un registro de pago.		
Suposiciones: El *actor Coordinador* tiene un usuario y una contraseña valida.		

UC-0016	Nombre del caso de uso: *Agregar usuarios*	
Actor(es): Coordinador		
Descripción: Cuando se desee agregar un nuevo usuario al sistema, el caso de uso se comportará de la siguiente manera.		
Activar el evento: El *actor Coordinador* ingresa al sistema, hace clic en la pestaña Usuarios, luego da clic en *Administrar cuentas.*		
Tipo: Primario y esencial		
Paso	**Acción**	
1	El sistema inicia este caso de uso cuando el ***actor Coordinador*** selecciona en la pestaña Usuarios *Administrar cuentas*, Luego da clic en nuevo.	
2	El ***actor Coordinador*** ingresa login del nuevo usuario.	
3	El sistema verifica si existe otro usuario con ese login.	
4	El ***actor Coordinador*** ingresa los datos del usuario como: nombre, apellidos, contraseña, permisos.	
5	El sistema verifica que los datos estén correctos y/o completos.	
6	El sistema registra el usuario.	
Precondición: El *actor Coordinador* debe estar conectado al sistema y cuenta con los permisos necesarios para crear usuarios.		
Poscondición: Se ha creado un nuevo usuario.		
Excepciones	**Paso**	**Acción**
	3	Si el login ya está registrado el sistema envía un mensaje y lo devolverá al paso 2, el caso de uso continúa.
	5	Si un dato esta incorrecto y/o incompleto el sistema le envía un mensaje y lo devolverá al paso 4, el caso de uso continúa.
Suposiciones: El *actor Coordinador* tiene un usuario y una contraseña valida.		

UC-0017		Nombre del caso de uso: *Actualizar usuarios*
Actor(es): Coordinador		
Descripción: Cuando se desee modificar un usuario del sistema, el caso de uso se comportará de la siguiente manera.		
Activar el evento: El *actor Coordinador* ingresa al sistema, hace clic en la pestaña Usuarios, luego da clic en *Administrar cuentas.*		
Tipo: Primario y esencial		
Paso	**Acción**	
1	El sistema inicia este caso de uso cuando el ***actor Coordinador*** selecciona en la pestaña Usuarios *Administrar cuentas*, Luego da clic en buscar.	
2	El ***actor Coordinador*** ingresa login del usuario.	
3	El sistema verifica si existe login.	
4	El sistema muestra los datos.	
5	El ***actor Coordinador*** modifica los datos.	
6	El sistema valida que los datos estén correctos y/o completos.	
7	El sistema actualiza el registro del usuario.	
Precondición: El *actor Coordinador* debe estar conectado al sistema y cuenta con los permisos necesarios para modificar usuarios.		
Poscondición: Se ha actualizado un registro de un usuario.		
Excepciones	**Paso**	**Acción**
	3	Si el login no está registrado el sistema envía un mensaje y lo devolverá al paso 2, el caso de uso continúa.
	6	Si un dato esta incorrecto y/o incompleto el sistema le envía un mensaje y lo devolverá al paso 5, el caso de uso continúa.
Suposiciones: El *actor Coordinador* tiene un usuario y una contraseña valida.		

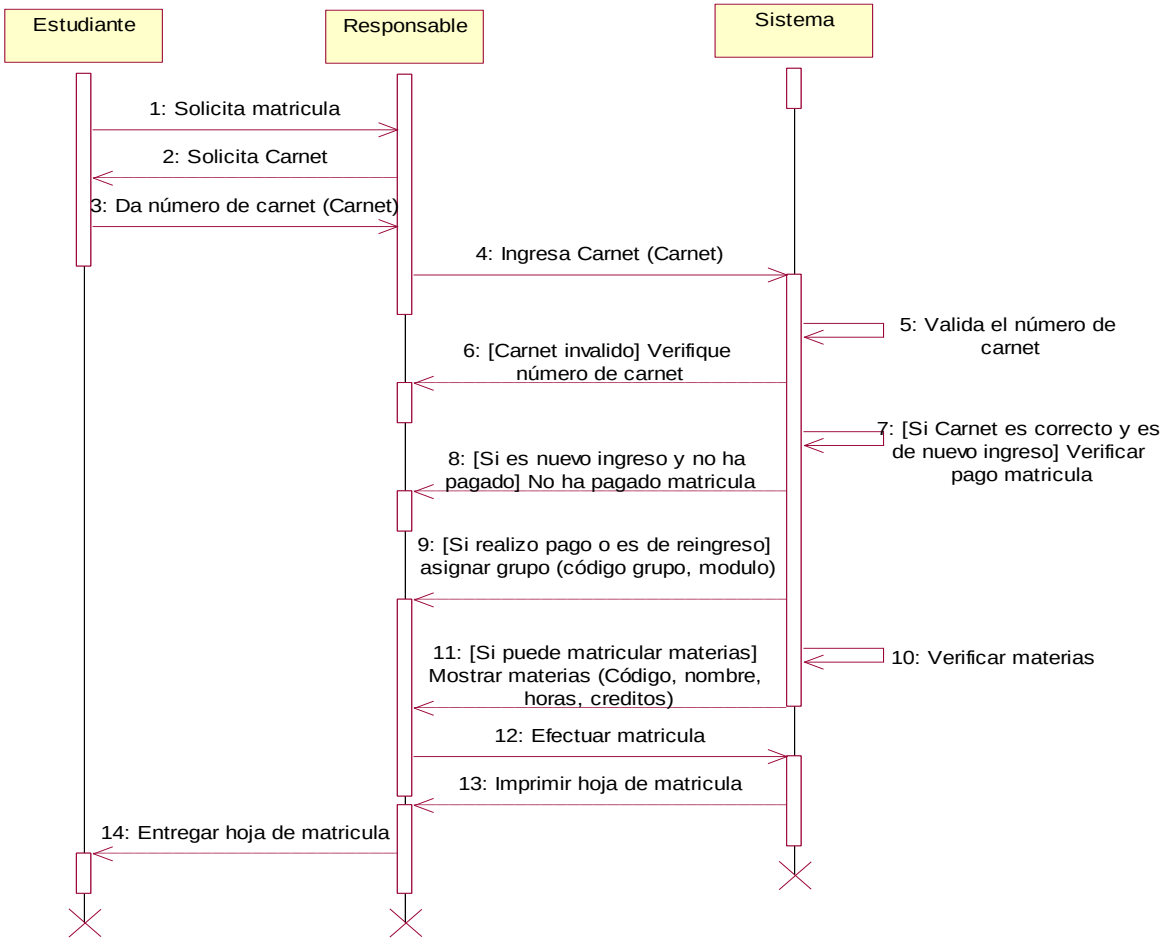
UC-0018	Nombre del caso de uso: *Eliminar usuarios*	
Actor(es): Coordinador		
Descripción: Cuando se desee eliminar un usuario del sistema, el caso de uso se comportará de la siguiente manera.		
Activar el evento: El *actor Coordinador* ingresa al sistema, hace clic en la pestaña Usuarios, luego da clic en *Administrar cuentas.*		
Tipo: Primario y esencial		
Paso	**Acción**	
1	El sistema inicia este caso de uso cuando el ***actor Coordinador*** selecciona en la pestaña Usuarios *Administrar cuentas*, Luego da clic en buscar.	
2	El ***actor Coordinador*** ingresa login del usuario.	
3	El sistema verifica si existe login.	
4	El sistema muestra los datos.	
5	El ***actor Coordinador*** da clic en eliminar.	
6	El sistema envía un mensaje de verificación de eliminación.	
7	El ***actor Coordinador*** acepta.	
8	El sistema elimina el registro del usuario.	
Precondición: El *actor Coordinador* debe estar conectado al sistema y cuenta con los permisos necesarios para eliminar usuarios.		
Poscondición: Se ha actualizado un registro de un usuario.		
Excepciones	**Paso**	**Acción**
	3	Si el login no está registrado el sistema envía un mensaje y lo devolverá al paso 2, el caso de uso continúa.
	5	El sistema verifica que el usuario a eliminar no sea el que inicio sesión, a continuación el caso de uso queda sin efecto.
	6	Si el ***actor Coordinador*** cancela, a continuación el caso de uso queda sin efecto.
Suposiciones: El *actor Coordinador* tiene un usuario y una contraseña valida.		

UC-0019	Nombre del caso de uso: *Actualizar información*	
Actor(es): Estudiante		
Descripción: Cuando un Estudiante desea modificar información de su cuenta en el sitio web de la maestría, el caso de uso se comportará de la siguiente manera.		
Activar el evento: El *actor Estudiante* usa el Sitio Web para cambiar la información, dando clic en cambiar contraseña.		
Tipo: Primario y esencial		
Paso	**Acción**	
1	El ***actor Estudiante*** se conecta al Sitio Web y da clic en *Cambiar contraseña.*	
2	El ***actor Estudiante*** ingresa los datos tales como: número de carnet, login, antigua contraseña, nueva contraseña y da clic en *cambiar contraseña*.	
3	El servidor valida que los datos estén correctos y/o completos.	
4	El servidor registra los cambios.	
5	El servidor envía mensaje de confirmación.	
Precondición: El *actor Estudiante* está en la página web *Cambiar contraseña*.		
Poscondición: El Estudiante ha cambiado exitosamente la contraseña.		
Excepciones	**Paso**	**Acción**
	3	Si un dato esta incorrecto y/o incompleto el servidor le envía un mensaje y lo devolverá al paso 2, el caso de uso continúa.
Suposiciones: El *actor Estudiante* tiene un navegador, un número de carnet, login y clave válidas.		

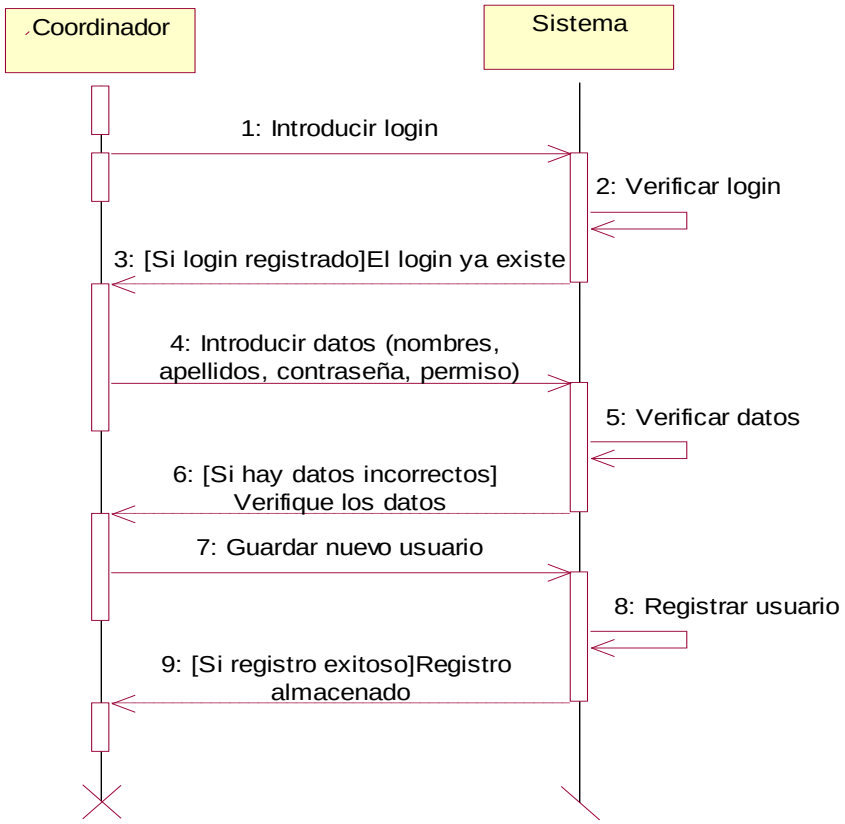
UC-0020	Nombre del caso de uso: *Realizar Consultas*	
Actor(es): Estudiante		
Descripción: Cuando un Estudiante desee realizar consultas acerca de sus notas y pagos en el sitio web de la maestría, el caso de uso se comportará de la siguiente manera.		
Activar el evento: El *actor Estudiante* usa el Sitio Web para consultar sus notas y pagos, introduciendo su login y clave de usuario, luego elige el reporte que desea e introduce su número de carnet.		
Tipo: Primario y esencial		
Paso	**Acción**	
1	El ***actor Estudiante*** se conecta al Sitio Web y da clic en *Iniciar Sesión.*	
2	El ***actor Estudiante*** ingresa su login y contraseña y da clic en entrar.	
3	El servidor valida que los datos estén correctos y/o completos.	
4	El servidor lo redirecciona a la página de consultas.	
5	El ***actor Estudiante*** da clic en el reporte ya sea pago o notas e ingresa su número de carnet.	
6	El servidor web valida que sea su número de carnet.	
7	El servidor web lo redirecciona al reporte.	
Precondición: El *actor Estudiante* está en la página web iniciar sesión.		
Poscondición: El Estudiante ha realizado exitosamente las consultas.		
Excepciones	**Paso**	**Acción**
	3	Si el login o la contraseña son incorrectas, el servidor le envía un mensaje y lo devolverá al paso 2, el caso de uso continúa.
	6	Si el número de carnet no le pertenece el servidor le envía un mensaje, y lo devolverá al paso 5, el caso de uso continúa.
	6	Si el usuario logra entrar si n iniciar sesión, el servidor valida que el usuario haya iniciado sesión en caso contrario al tratar de consultar un reporte, este lo redirecciona a la página de inicio.
Suposiciones: El *actor Estudiante* tiene un navegador, un número de carnet, login y clave válidas.		

Resultado 4 - Diagramas de secuencia

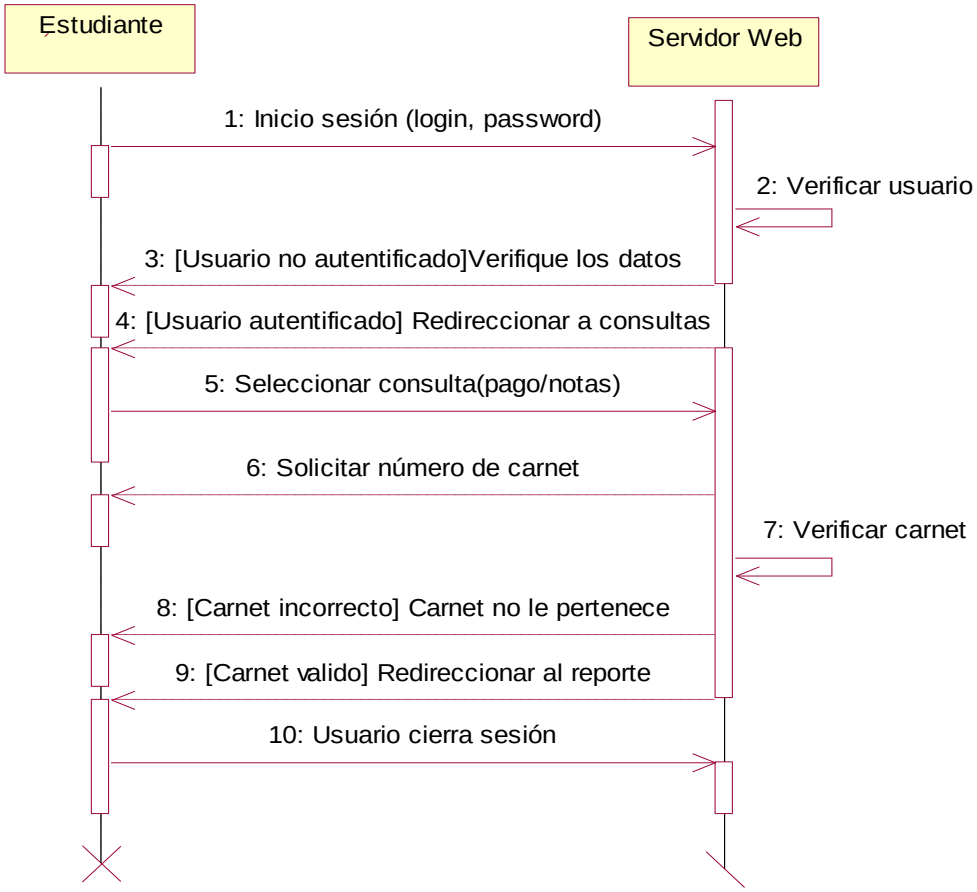
1 – REALIZAR MATRICULA



2 – INGRESAR USUARIO

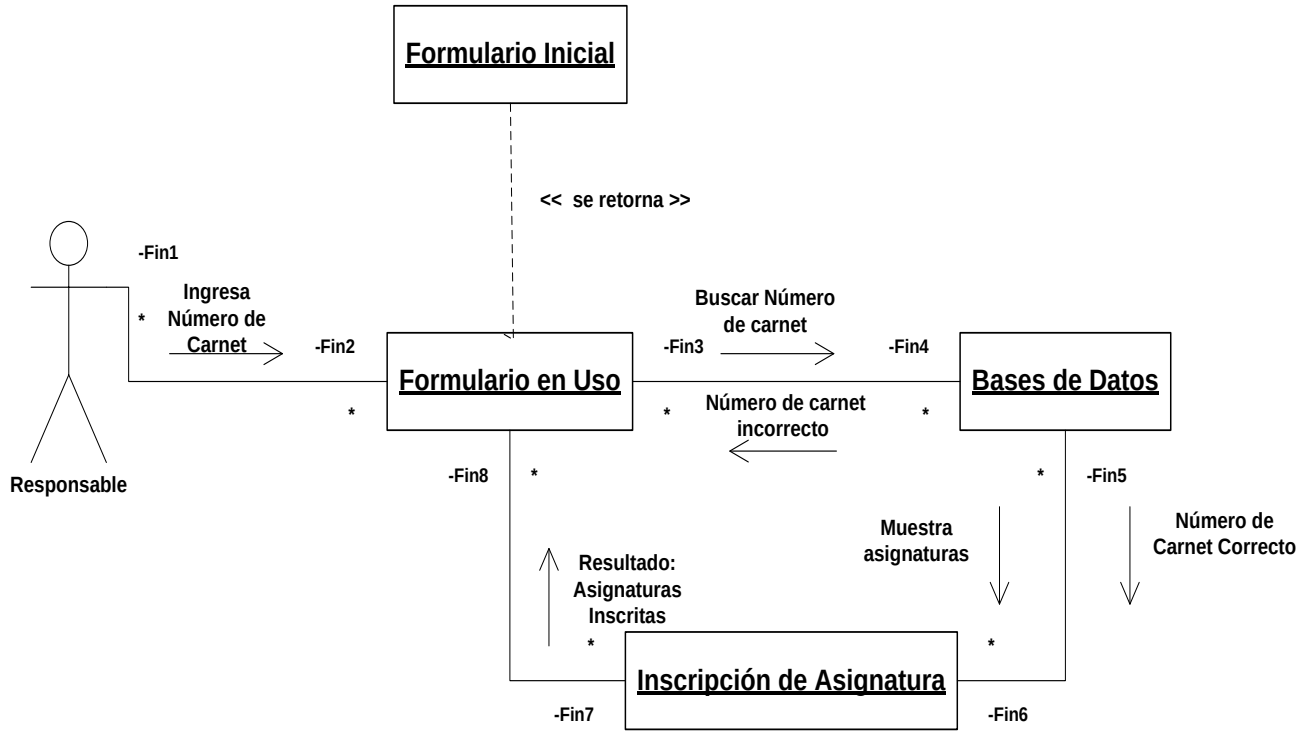


3 – REZLIZAR COSULTAS

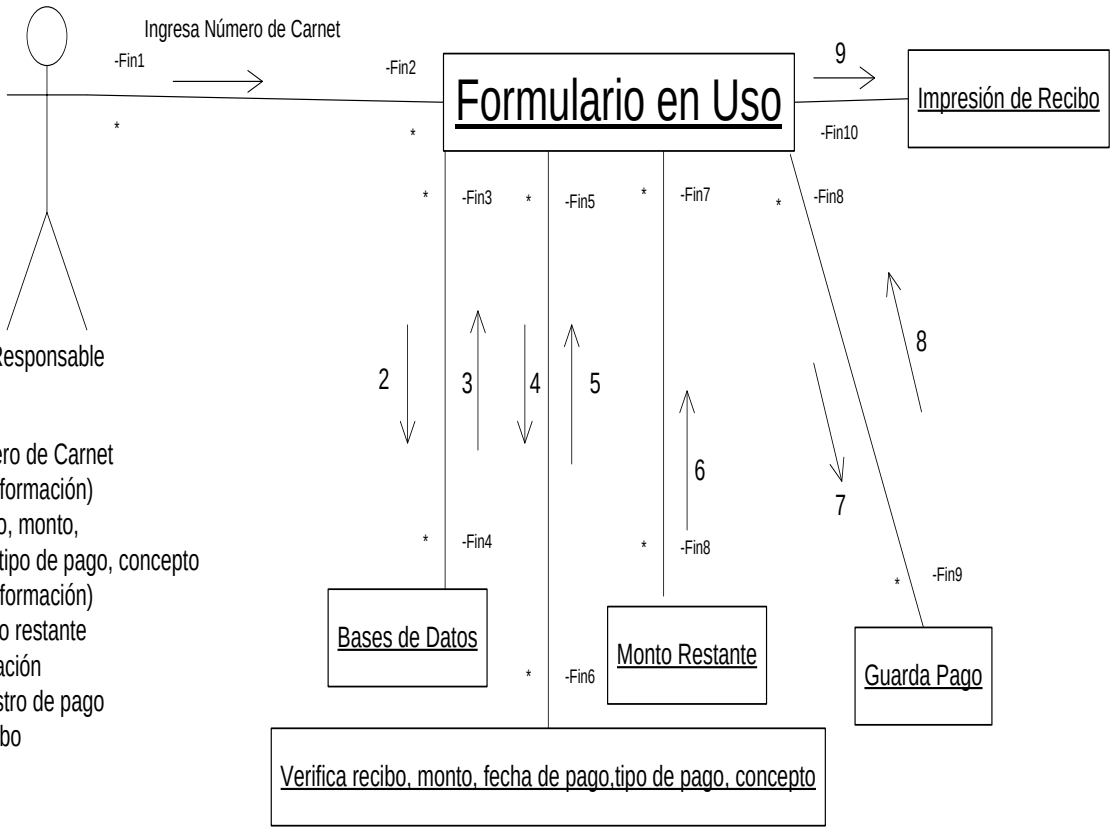


Resultado 5 - Diagrama de colaboración

1 – INSCRIBIR ASIGNATURAS

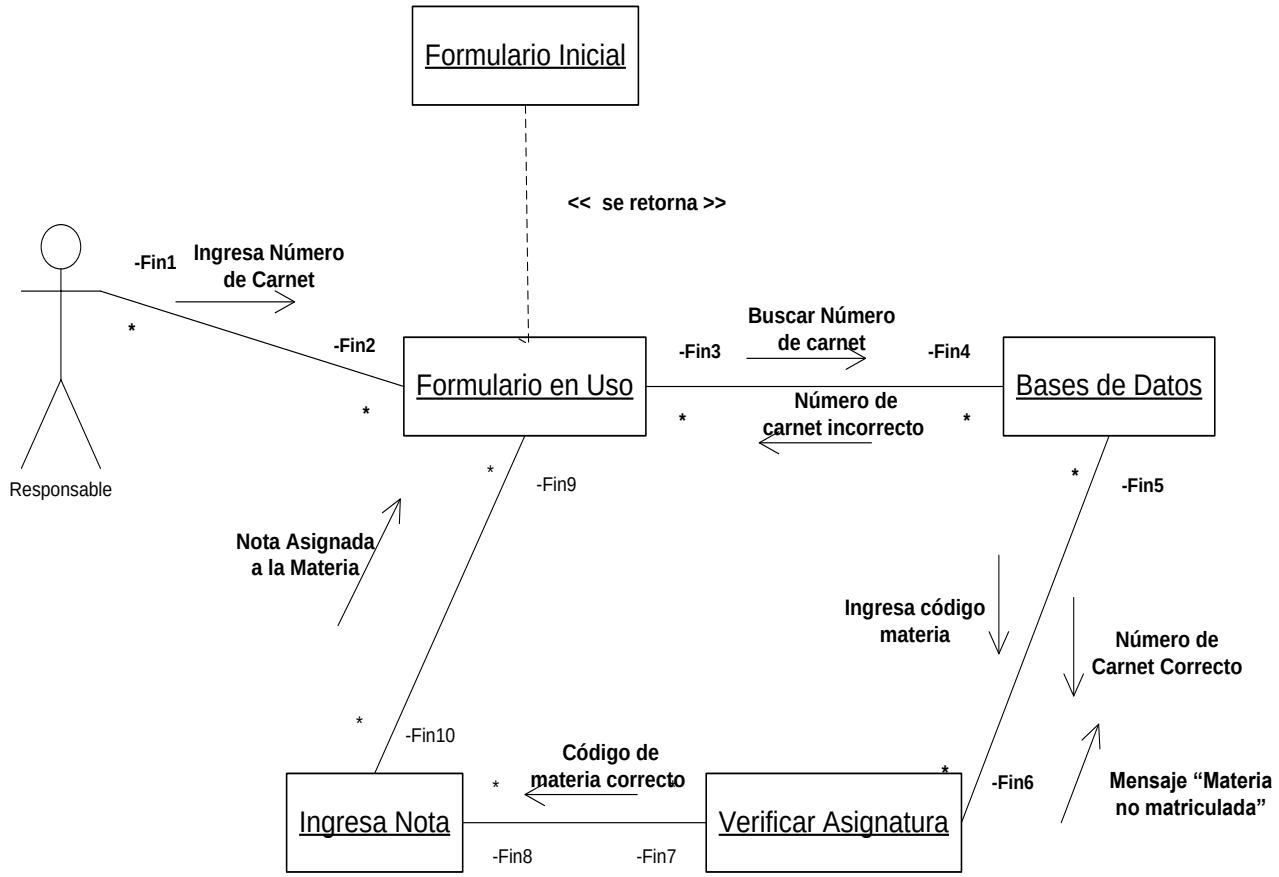


2 – REALIZAR PAGO DE MENSUALIDAD



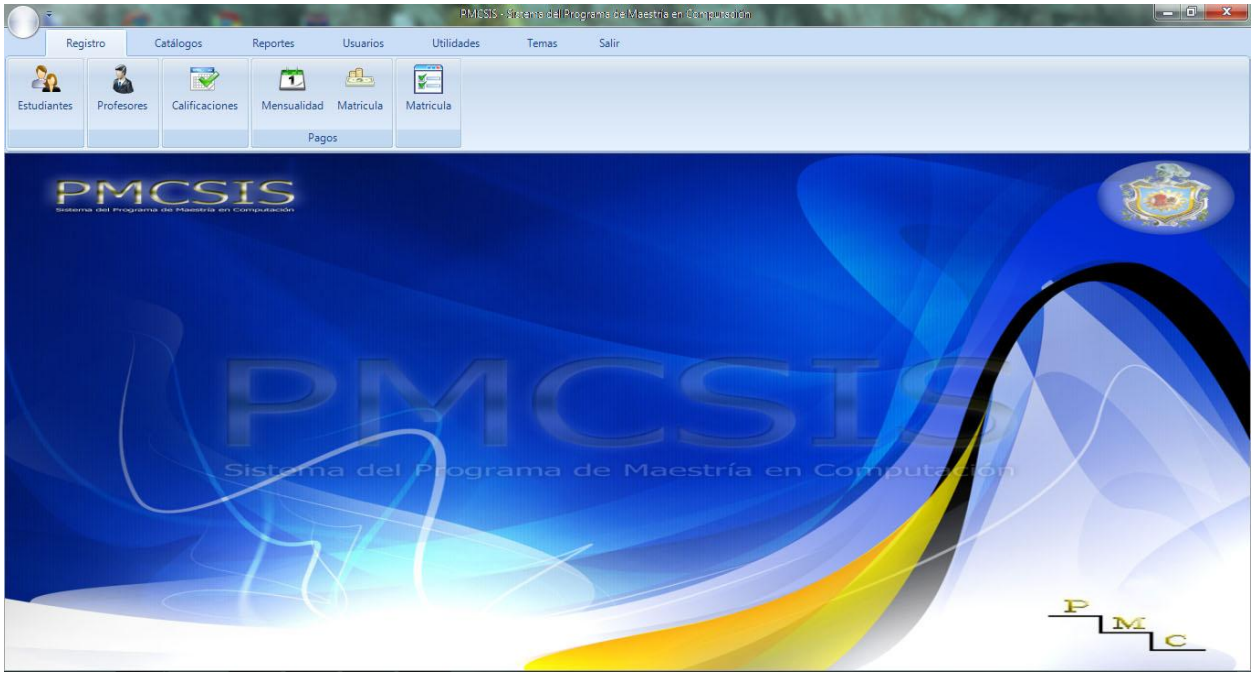
2 – Buscar Número de Carnet
3 – Resultado (información)
4 – ingresa recibo, monto,
Fecha de pago, tipo de pago, concepto
5 – Resultado (información)
6 – Calcula monto restante
7 – Envía información
8 – Guardar registro de pago
9 – Imprime Recibo

3 – INGRESAR UNA CALIFICACION DE UNA ASIGNATURA DE UN ESTUDIANTE

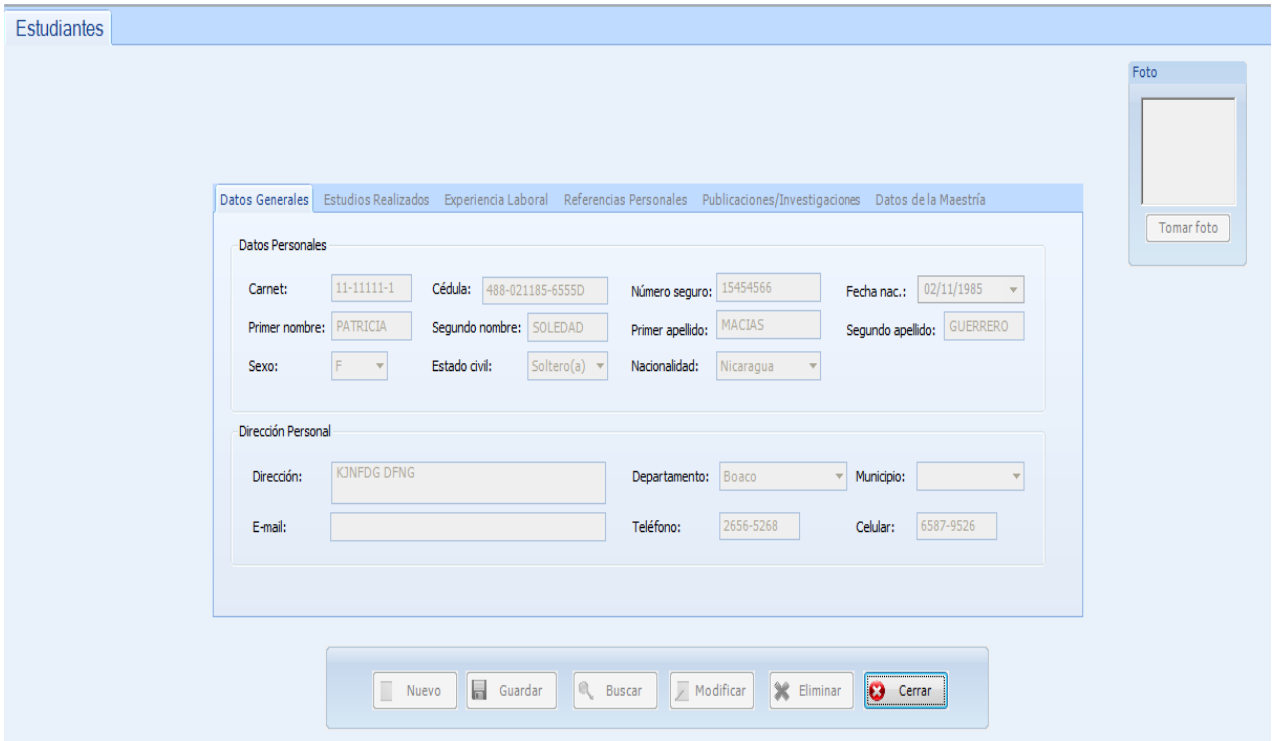


Resultado 6 – Pantallas

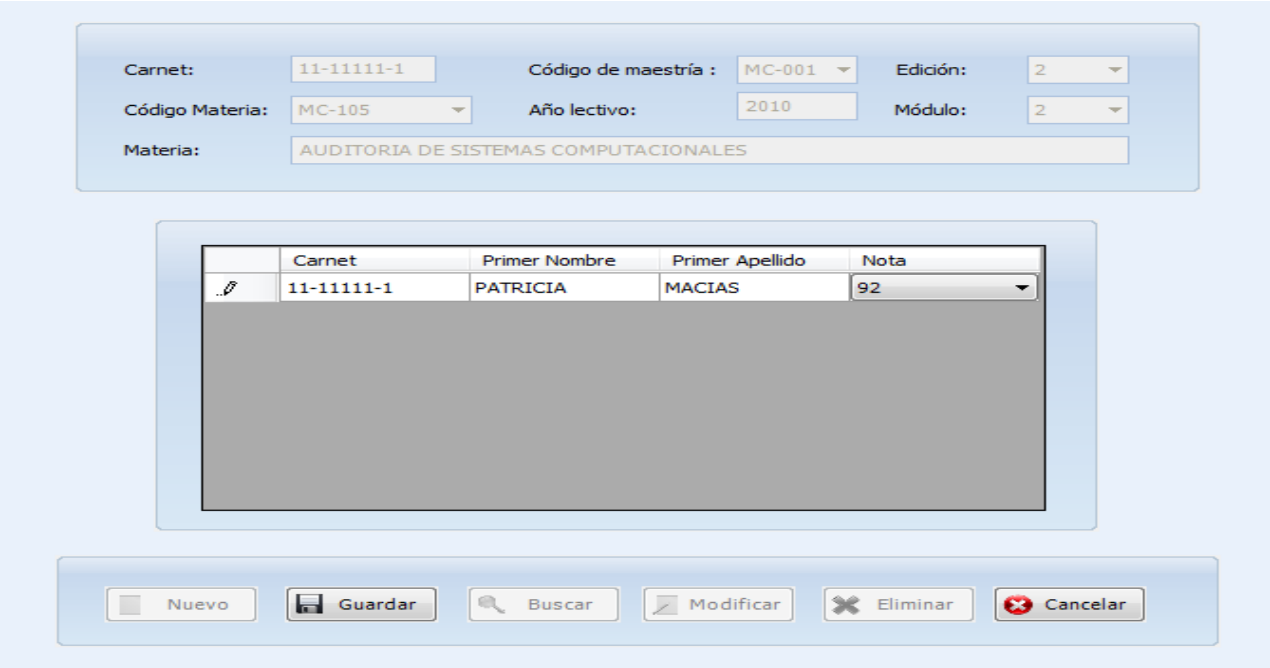
1 - PANTALLA PRINCIPAL



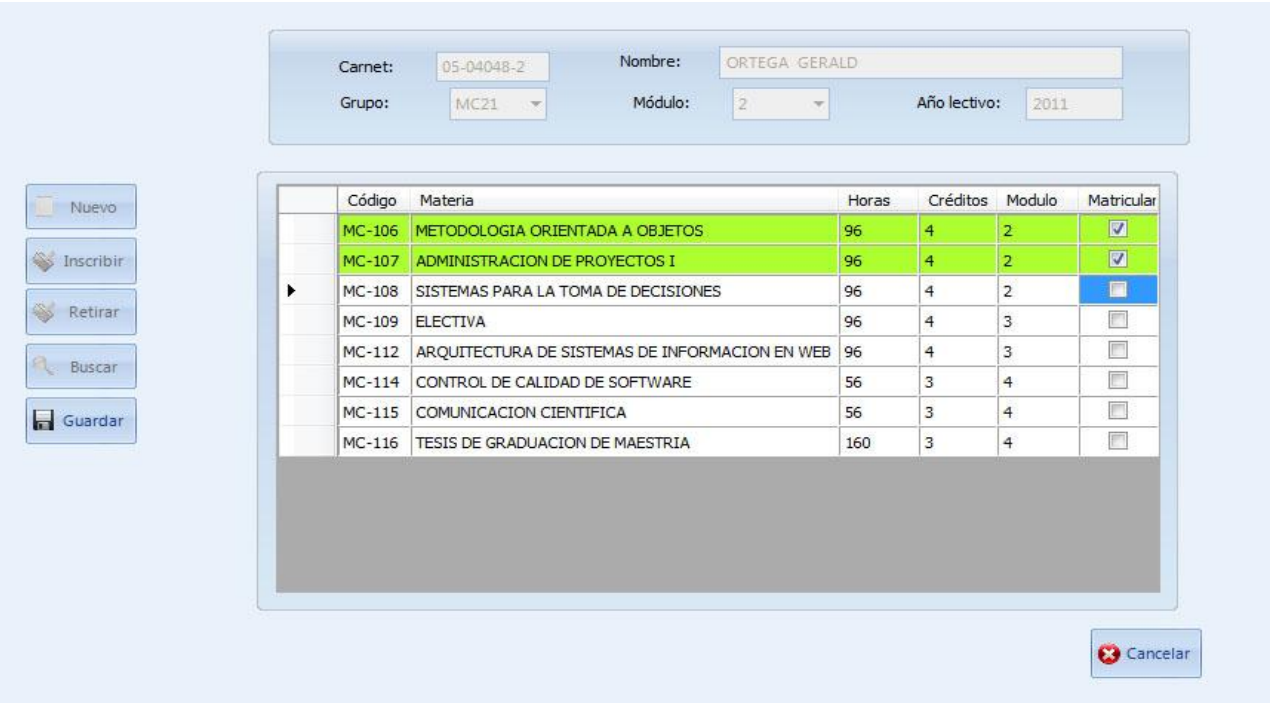
2 - REGISTRO DE EXPEDIENTES DE ESTUDIANTES



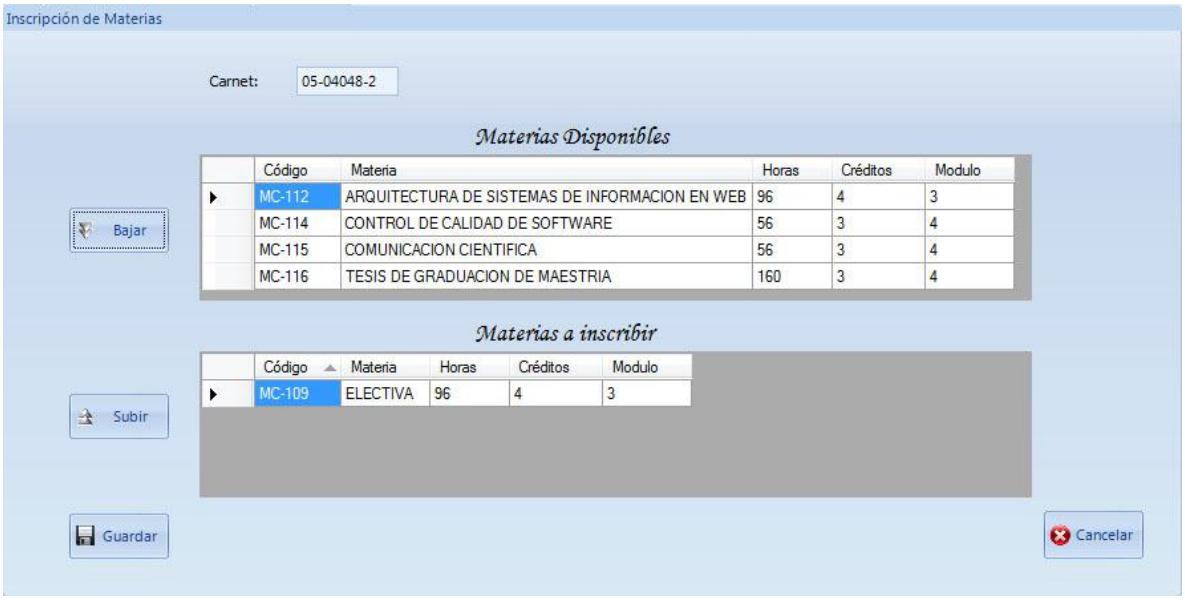
3 - REGISTRO DE CALIFICACIONES DEL ESTUDIANTE



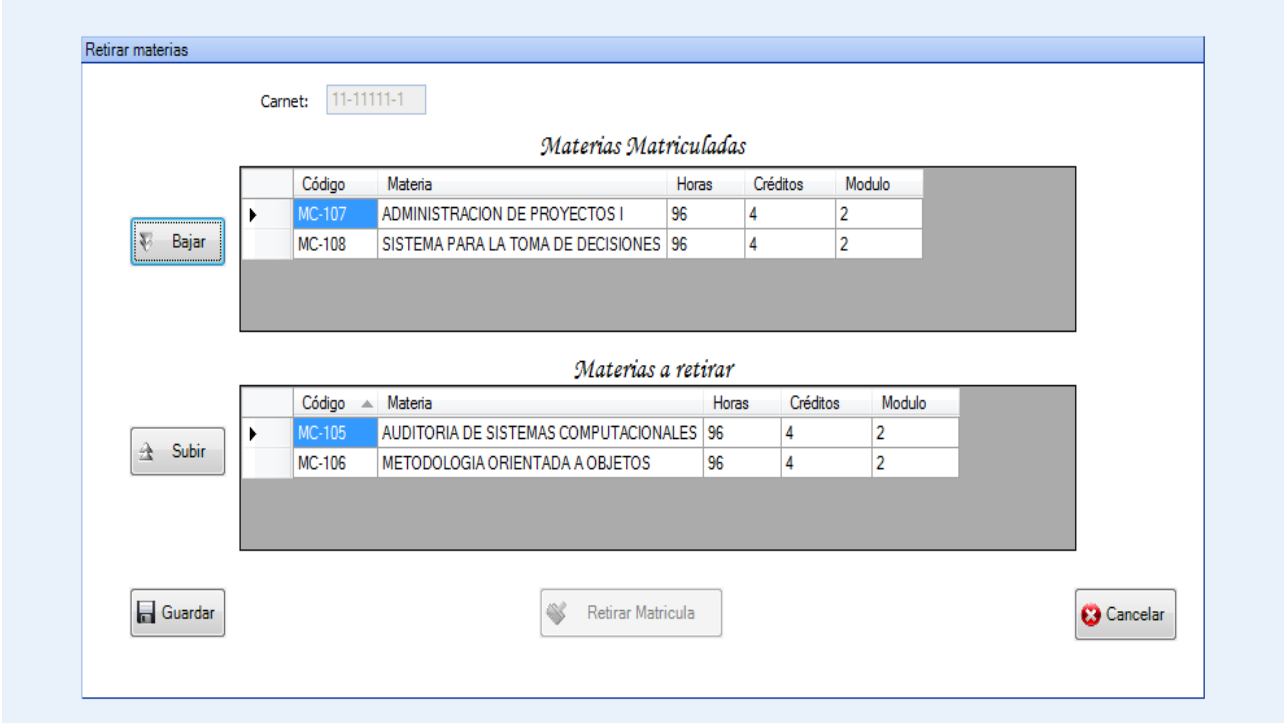
4 - MATRICULA DE ASIGNATURAS



5 - INSCRIPCIÓN DE ASIGNATURAS



6 - RETIRO DE ASIGNATURAS



7 - REGISTRO DE DOCENTES

8 - CATALOGOS

REGISTRO DE FACULTAD

Código de facultad: FC-001

Nombre de facultad: CIENCIAS E INGENIERIAS

Nuevo | Guardar | Buscar | Eliminar

REGISTRO DE GRUPOS

Código de Grupo:

Código de Maestria:

Nuevo | Guardar | Buscar | Modificar

REGISTRO DE DEPARTAMENTOS

Código Departamento: 08

Nombre Departamento: Boaco

Nuevo | Guardar | Buscar | Modificar

REGISTRO DE GRUPOS

Código de Grupo: MC11

Código de Maestria: MC-001

Nuevo | Guardar | Buscar | Modificar

REGISTRO DE MUNICIPIOS

Código Departamento: 07

Código Municipio: 07-03

Nombre Municipio: belen

Nuevo | Guardar | Buscar | Modificar

REGISTRO DE CATEGORIA - DOCENTE

Código Categoria: 1

Nombre Categoria: TITULAR

Nuevo | Guardar | Buscar | Modificar | Eliminar

REGISTRO DE DOCENTE - MATERIA

Nombres y Apellidos: LUIS JBFDS DFJBSD

Código de docente: 11-111 Código de materia: MC-101

Módulo: 1 Año lectivo: 2010

Código de Grupo: MC-11

Nuevo Guardar Buscar Eliminar

REGISTRO DE MATERIAS PLAN DE ESTUDIO

Código plan de estudio: PL-009

Código de materia: MC-101

Modulo: 1

Nuevo Guardar Buscar Eliminar

REGISTRO DE MATERIAS

Código de materia: MC-101 Nombre materia: NEGOCIOS ELECTRONICOS

Número de horas: 96

Requisitos

Si No

Número de creditos: 4

Nuevo Guardar Buscar Modificar Eliminar

REGISTRO DE PLAN DE ESTUDIO

Código plan de estudio: PL-099 Código Maestria: MC-001

Plan de estudio: 2009 Forma de graduacion del plan: Tesis

Número de horas del plan: 325 Titulo a Obtener Master en Sistemas de Información

Nuevo Guardar Buscar Modificar

REGISTRO DE CARRERAS

Código de carrera: CA-007

Código de Departamento: DC-005

Nombre carrera: CC COMPUTACION

Nuevo Guardar Buscar Modificar Eliminar

REGISTRO DE MAESTRIA

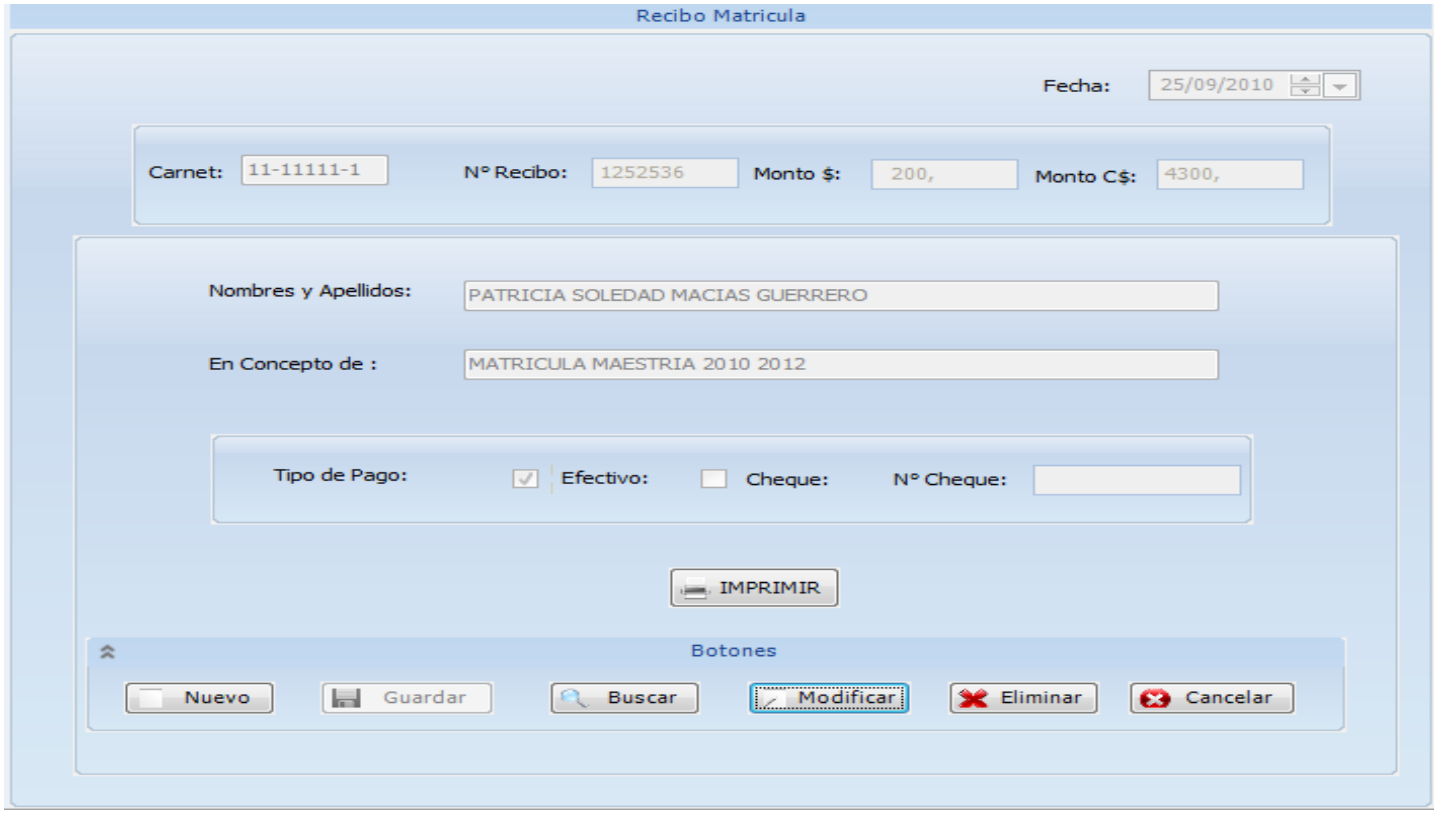
Código de maestría: MC-001 Fecha Inicio: 24/09/2010

Código Facultad: fc-001 Duración maestría: 2

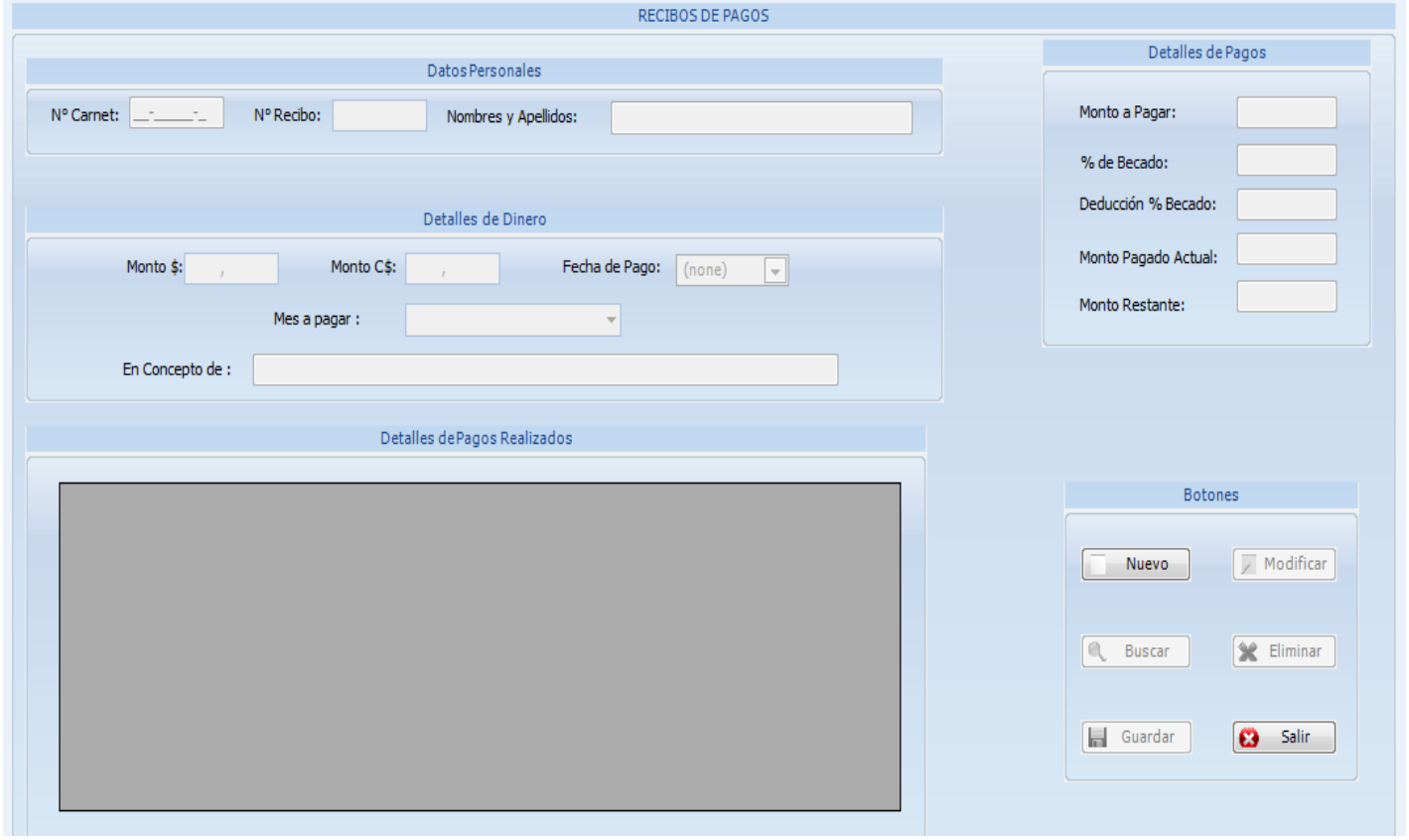
Nombre maestría: Sistema de Información Edición: 2

Nuevo Guardar Buscar Modificar Eliminar

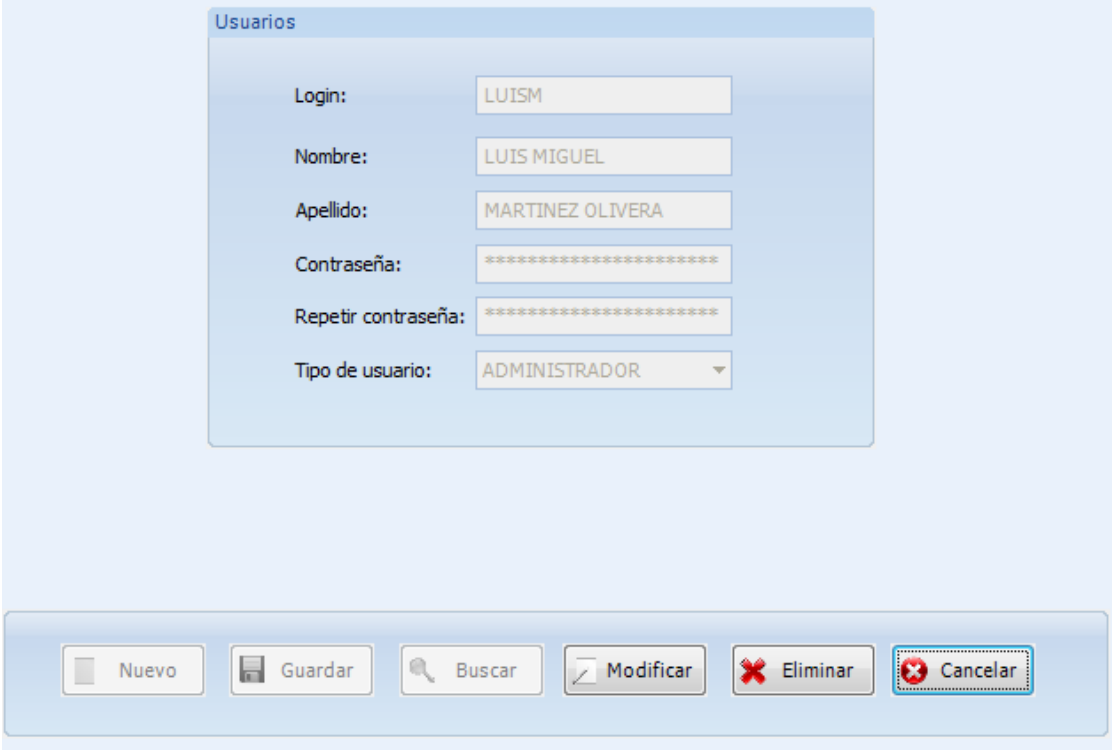
9 - REGISTRO DE PAGO DE MATRICULA



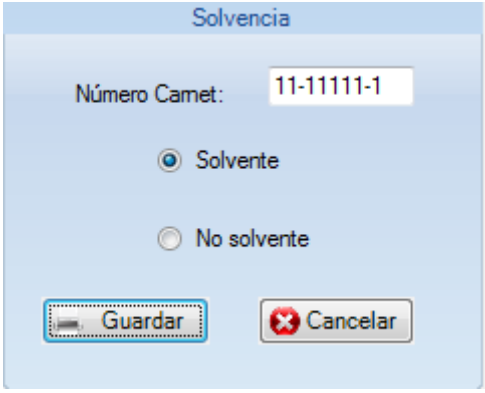
10 - REGISTRO DE PAGO DE MENSUALIDAD



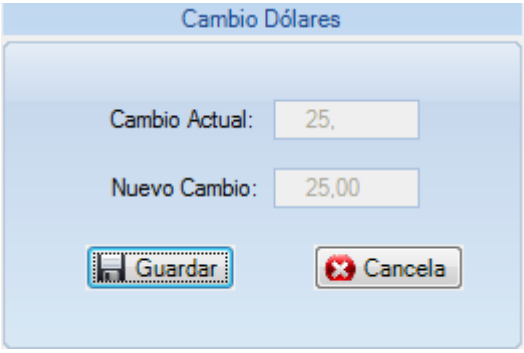
11 - REGISTRO DE USUARIO



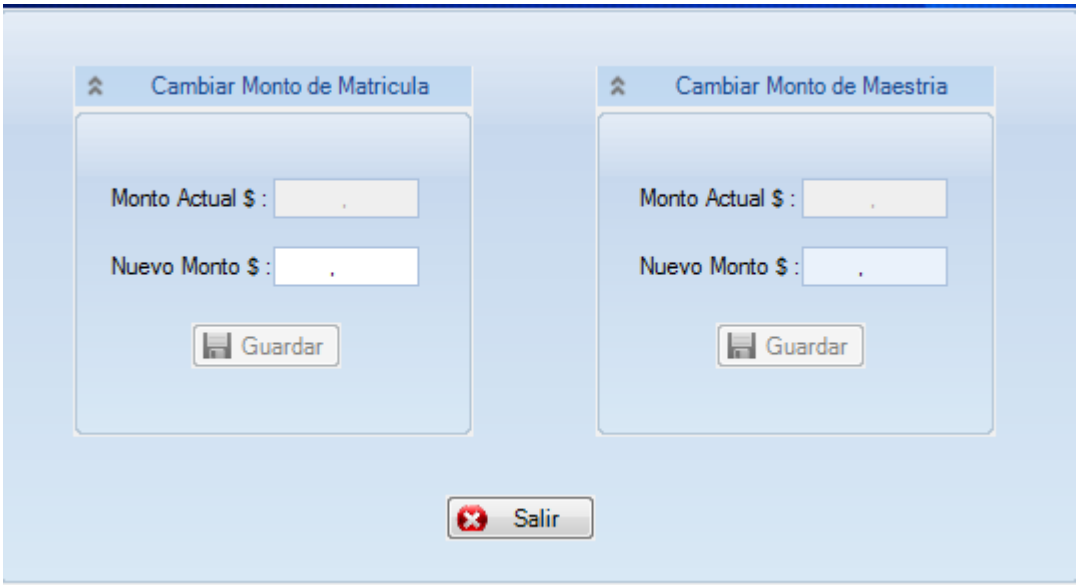
12 - SOLVENCIA DE ESTUDIANTE



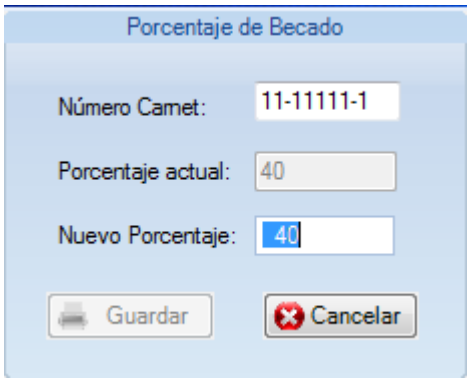
13 - CAMBIO DE DÓLAR



14 - ACTUALIZAR MONTOS DE MAESTRÍA

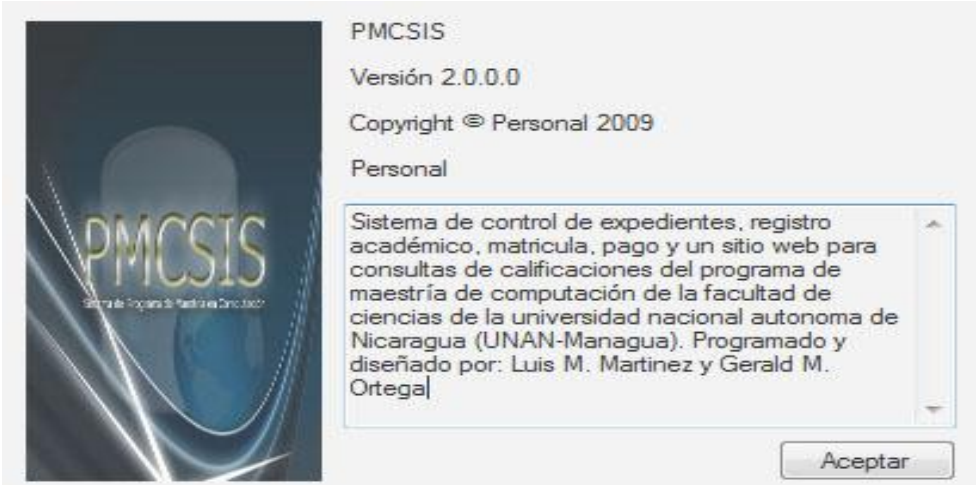


15 - PORCENTAJE DE BECADOS

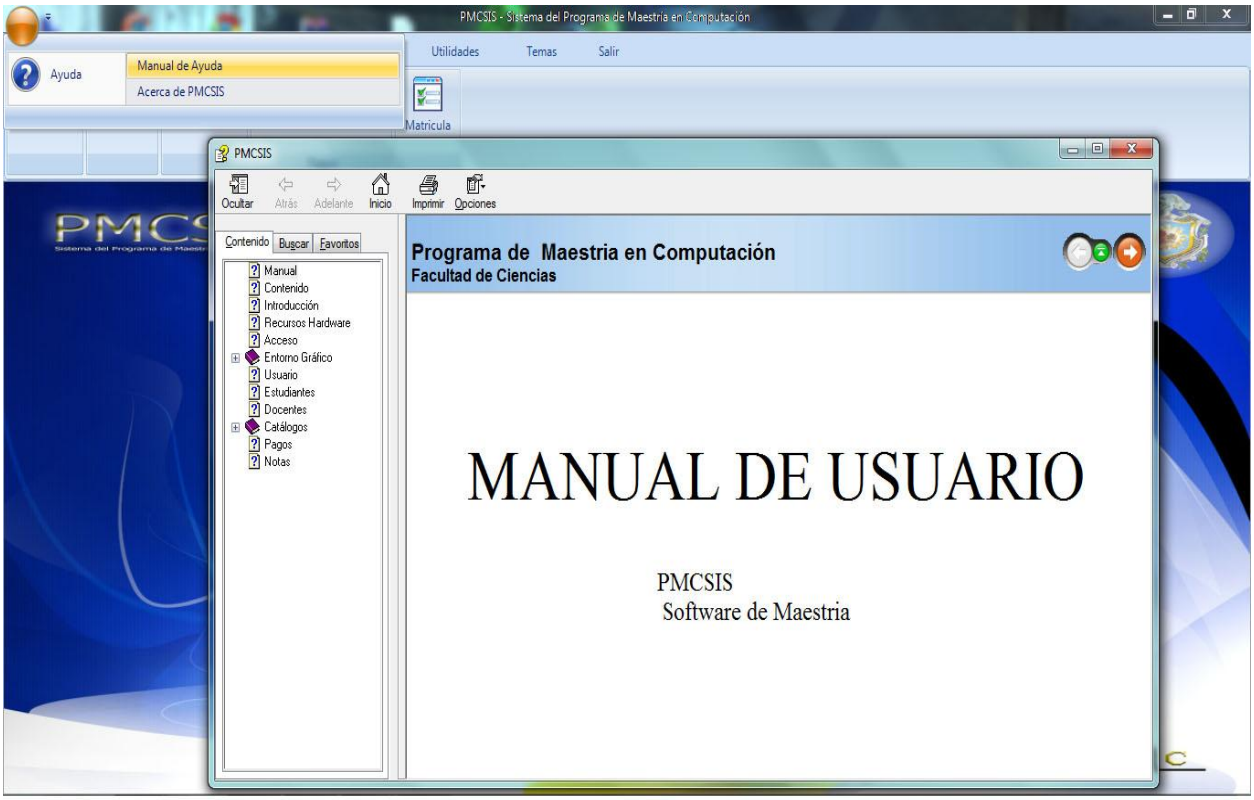


16 – CANTIDAD DE MATERIAS MATRICULA

17 - ACERCA DE



18 – MANUAL DE AYUDA



Resultado 7- Reportes

REPORTES NO ESTADISTICOS

1- MATRICULA DE ASIGNATURA

UNIVERSIDAD NACIONAL AUTONOMA DE NICARAGUA
RECINTO UNIVERSITARIO RUBEN DARIO
DEPARTAMENTO DE COMPUTACION
MAESTRIA EN COMPUTACION CON ENFASIS EN SISTEMAS DE INFORMACION

Carnet: 11-11111-1 Nombre: MACIAS GUERRERO PATRICIA SOLEDAD

Grupo: MC11 Modulo: 1 Año lectivo: 2010

Hoja de Matricula

Código	Materia	Creditos
MC-101	NEGOCIOS ELECTRONICOS	4
MC-102	INGENIERIA DEL SOFTWARE	4
MC-103	INTELIGENCIA DE NEGOCIOS	4
MC-104	BASE DE DATOS AVANZADAS	4

Firma Responsable Firma Estudiante

PMCSIS Usuario: luism

26/09/2010 18:24:58

2 - RETIRO DE ASIGNATURA

UNIVERSIDAD NACIONAL AUTONOMA DE NICARAGUA
RECINTO UNIVERSITARIO RUBEN DARIO
DEPARTAMENTO DE COMPUTACION
MAESTRIA EN COMPUTACION CON ENFASIS EN SISTEMAS DE INFORMACION

Carnet: 11-11111-1 Nombre: MACIAS GUERRERO PATRICIA SOLEDAD

Grupo: MC21 Modulo: 2 Año lectivo: 2010

Hoja de retiro de materias

	Código	Materia	Creditos
M	MC-106	METODOLOGIA ORIENTADA A OBJETOS	4
M	MC-107	ADMINISTRACION DE PROYECTOS I	4
M	MC-108	SISTEMA PARA LA TOMA DE DECISIONES	4
R	MC-105	AUDITORIA DE SISTEMAS COMPUTACIONALES	4

Firma Estudiante Firma Responsable

PMCSIS Usuario:

3 - RECIBO DE PAGO DE MATRICULA

UNIVERSIDAD NACIONAL AUTONOMA DE NICARAGUA
RECINTO UNIVERSITARIO RUBEN DARIO
DEPARTAMENTO DE COMPUTACION
MAESTRIA EN COMPUTACION CON ENFASIS EN SISTEMAS DE INFORMACION

RECIBO DE PAGO DE MATRICULA

Carnet : 11-11111-1 **Nº Recibo :** 1252536

Nombres y Apellidos : PATRICIA SOLEDAD MACIAS GUERRERO

DETALLES DEL PAGO

Cantidad en Dolares : $ 200 **Fecha de Pago :** 25/09/2010

Cantidad en Cordoba : C$ 4300

En Concepto de : MATRICULA MAESTRIA 2010 2012

Efectivo: X

Firma Responsable **Firma Estudiante**

PMCSIS **Usuario :** luism

4 - RECIBO DE PAGO DE MENSUALIDAD

UNIVERSIDAD NACIONAL AUTONOMA DE NICARAGUA
RECINTO UNIVERSITARIO RUBEN DARIO
DEPARTAMENTO DE COMPUTACION
MAESTRIA EN COMPUTACION CON ENFASIS EN SISTEMAS DE INFORMACION

RECIBO DE PAGO DE MENSUALIDAD

21/11/2010

Número de Carnet: 11-11111-1 **Número de Recibo:** 4654643

Nombres y Apellidos : LUIS MIGUEL MARTINEZ OLIVERA

DETALLES DEL PAGO

Cantidad en Dolares : $ 200.00 **Fecha de Pago :** 21/11/2010

Cantidad en Cordoba : C$ 4300 **Nº Cuota :** 1

En Concepto de : DFGFDG JKF G

Tipo Pago: EFECTIVO **Número Cheque:**

Firma Responsable **Firma Estudiante**

PMCSIS **Usuario:** luism

5 - LISTADO DE ESTUDIANTES POR MAESTRÍA

UNIVERSIDAD NACIONAL AUTONOMA DE NICARAGUA
RECINTO UNIVERSITARIO RUBEN DARIO
DEPARTAMENTO DE COMPUTACION
MAESTRIA EN COMPUTACION CON ENFASIS EN SISTEMAS DE INFORMACION

Lista de Estudiantes

Maestría: Sistema de Información **Edición:** 2

Carnet	Primer apellido	Segundo apellido	Primer nombre	Segundo nombre	Sexo	Nacionalidad
10-00000-0	PALMA	VIVAS	HENRY	FERNANDO	M	
10-11111-1	JUAREZ	VEGA	CARLOS	ENRIQUE	F	
11-11111-1	MACIAS	GUERRERO	PATRICIA	SOLEDAD	F	
12-22222-2	HERRERA	SUAREZ	ALMA	IRIS	M	
13-33333-3	CASTILLO	URBINA	HUMBERTO	NOEL	M	
14-44444-4	ANTON	FLORES	YADER	JOSE	M	
14-72583-6	HERNANDEZ	UGARTE	EVING	JOSE	M	
15-55555-5	SILVA	OPORTA	TAMARA	DEL SOCORRO	F	
16-66666-6	ALTAMIRANO	LOPEZ	AUGUSTO	CESAR	M	
20-00000-0	MARTINEZ	OLIVERA	LUIS	MIGUEL	M	

6 - LISTADO DE DOCENTES

7 –

UNIVERSIDAD NACIONAL AUTONOMA DE NICARAGUA
RECINTO UNIVERSITARIO RUBEN DARIO
DEPARTAMENTO DE COMPUTACION
MAESTRIA EN COMPUTACION CON ENFASIS EN SISTEMAS DE INFORMACION

INFORME DE DOCENTES DE LA MAESTRIA

27/10/2010

CODIGO	NOMBRES Y APELLIDOS	ESPECIALIDAD	NIVEL ACADEMICO	NACIONALIDAD
11-111	LUIS JBFDS DFJBSD	JSDBFHDS	Doctor	Afganistán
55-555	LUIS SJDHFJUDSF KFSHDFGS	DEWPOFSDK	Licenciado	Nicaragua

7- PAGOS POR ESTUDIANTES

UNIVERSIDAD NACIONAL AUTONOMA DE NICARAGUA
RECINTO UNIVERSITARIO RUBEN DARIO
DEPARTAMENTO DE COMPUTACION
MAESTRIA EN COMPUTACION CON ENFASIS EN SISTEMAS DE INFORMACION

INFORME DE MESES CANCELADOS


Fotografia

Carnet: 11-11111-1

Nombres y Apellidos: PATRICIA SOLEDAD MACIAS GUERRERO

N° Recibo	Fecha Pago	Monto Dólar	Monto Córdoba	Mes Cancelado
1234567	26/10/2010	167.00	3590	SEPTIEMBRE
1234568	26/11/2010	167.00	3590	NOVIEMBRE
2566969	26/09/2010	167.00	3590,5	OCTUBRE

1

8 - HISTORIAL ACADÉMICO

UNIVERSIDAD NACIONAL AUTONOMA DE NICARAGUA
RECINTO UNIVERSITARIO RUBEN DARIO
DEPARTAMENTO DE COMPUTACION
MAESTRIA EN COMPUTACION CON ENFASIS EN SISTEMAS DE INFORMACION

Historial Academico

Carnet; 05-04048-2 Maestría: MC-001 Edición: 1

Nombre: GERALD Apellido: ORTEGA

Modulo	Código	Asignatura	Año	Nota
1	MC-101	NEGOCIOS ELECTRONICOS	2011	85
1	MC-102	INGENIERIA DEL SOFTWARE	2011	89
			Promedio:	87.00
			Promedio General:	87.00

NSP: No se presento
Advertencia: Este historial no tiene válidez sino lleva firma ni sello del coordinador de la maestria

9 - PLAN DE ESTUDIO

UNIVERSIDAD NACIONAL AUTONOMA DE NICARAGUA
RECINTO UNIVERSITARIO RUBEN DARIO
DEPARTAMENTO DE COMPUTACION
MAESTRIA EN COMPUTACION CON ENFASIS EN SISTEMAS DE INFORMACION

PLAN ESTUDIO

FACULTAD : fc-001 CIENCIAS E INGENIERIAS **DURACION :** 2 años

MAESTRIA : MC-001 Sistema de Información **HORAS DEL PLAN** 325

PLAN ESTUDIO : 2009 **TITULO A OBTENER :** Master en Sistemas de Información

FORMA DE GRADUACION : Tesis

MODULO 1	Código	Asignatura	Horas	Créditos
	MC-102	INGENIERIA DEL SOFTWARE	96	4
	MC-103	INTELIGENCIA DE NEGOCIOS	96	4
	MC-104	BASE DE DATOS AVANZADAS	96	4

MODULO 2	Código	Asignatura	Horas	Créditos
	MC-105	AUDITORIA DE SISTEMAS COMPUTACIONALES	96	4
	MC-106	METODOLOGIA ORIENTADA A OBJETOS	96	4
	MC-107	ADMINISTRACION DE PROYECTOS I	96	4
	MC-108	SISTEMA PARA LA TOMA DE DECISIONES	96	4

MODULO 3	Código	Asignatura	Horas	Créditos
	MC-109	ELECTIVA	96	4
	MC-110	PROGRAMACION ORIENTADA A OBJETOS	96	4
	MC-111	ADMINISTRACION DE PROYECTOS II	96	4
	MC-112	ARQUITECTURA DE SISTEMAS DE INFORMACION	96	4

MODULO 4	Código	Asignatura	Horas	Créditos
	MC-113	PLANIFICACION ESTRATEGICA DE TI	56	3
	MC-114	CONTROL DE CALIDAD DEL SOFTWARE	56	3
	MC-115	COMUNICACION CIENTIFICA	56	3
	MC-116	TESIS DE GRADUACION PARA MAESTRIA	160	4

REPORTES ESTADISTICOS

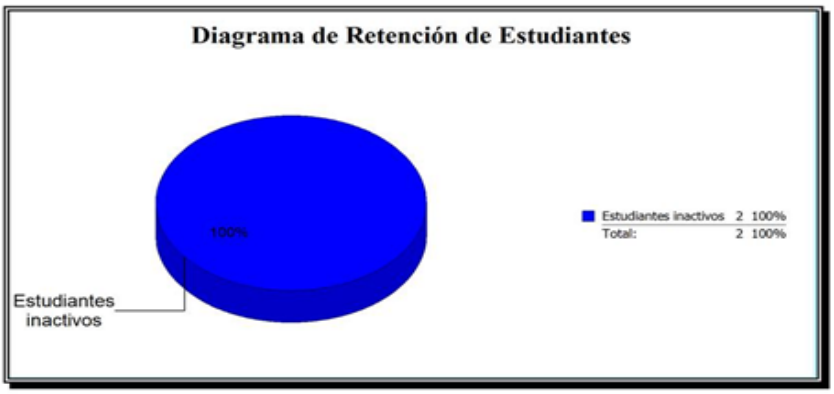
1 - PORCENTAJE DE BECADOS

UNIVERSIDAD NACIONAL AUTONOMA DE NICARAGUA
RECINTO UNIVERSITARIO RUBEN DARIO
DEPARTAMENTO DE COMPUTACION
MAESTRIA EN COMPUTACION CON ENFASIS EN SISTEMAS DE INFORMACION

2 - PORCENTAJE DE RETENCIÓN

UNIVERSIDAD NACIONAL AUTONOMA DE NICARAGUA
RECINTO UNIVERSITARIO RUBEN DARIO
DEPARTAMENTO DE COMPUTACION
MAESTRIA EN COMPUTACION CON ENFASIS EN SISTEMAS DE INFORMACION

DIAGRAMA



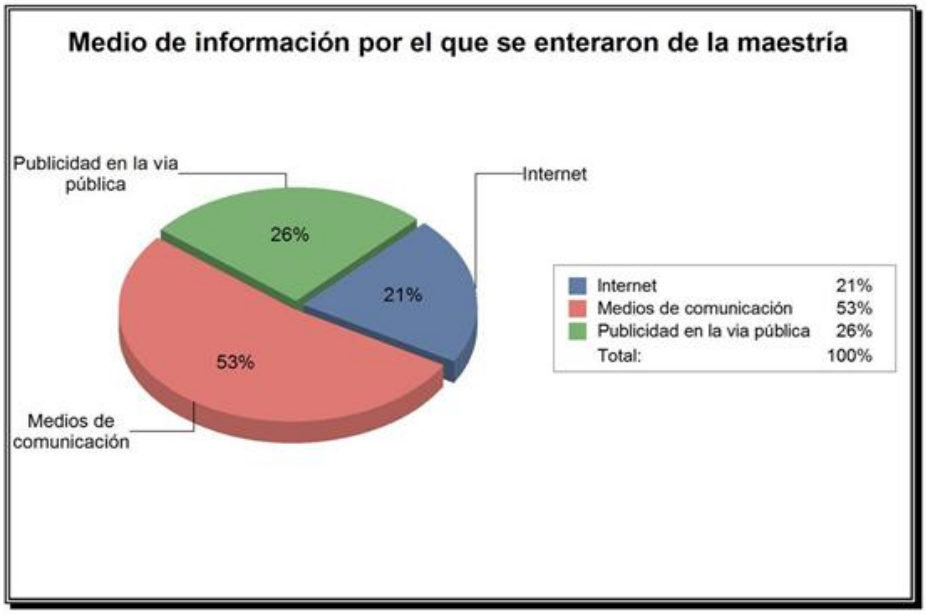
Lista de Estudiantes

	Carnet	Primer nombre	Primer apellido
Estudiantes inactivos			
	14-72583-6	EVING	HERNANDEZ
	48-85858-5	FGG	DFFGDF

3 - PORCENTAJE DEL MEDIO DE INFORMACIÓN

UNIVERSIDAD NACIONAL AUTONOMA DE NICARAGUA
RECINTO UNIVERSITARIO RUBEN DARIO
DEPARTAMENTO DE COMPUTACION
MAESTRIA EN COMPUTACION CON ENFASIS EN SISTEMAS DE INFORMACION

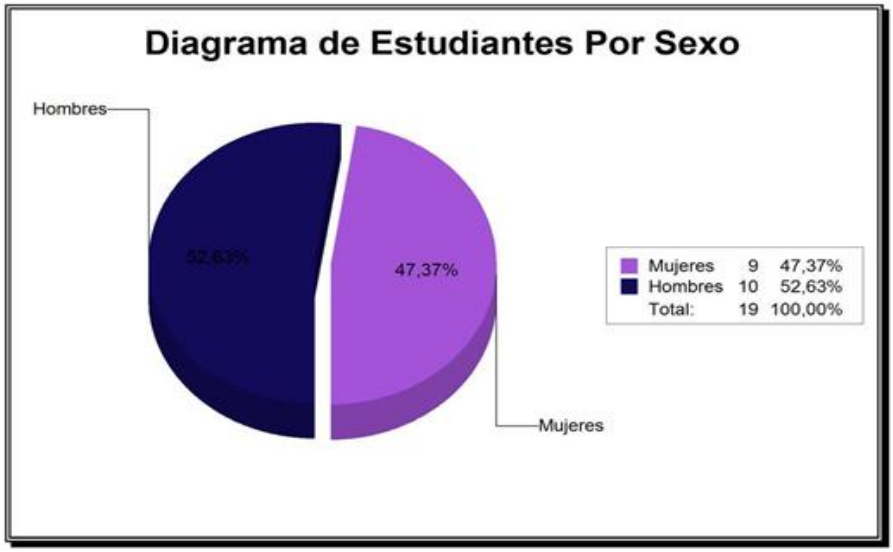
Diagrama



4 - PORCENTAJE DE ESTUDIANTES POR SEXO

UNIVERSIDAD NACIONAL AUTONOMA DE NICARAGUA
RECINTO UNIVERSITARIO RUBEN DARIO
DEPARTAMENTO DE COMPUTACION
MAESTRIA EN COMPUTACION CON ENFASIS EN SISTEMAS DE INFORMACION

Diagrama



Resultado 8 - Pantallas del sitio web de consultas de calificaciones y pagos

1 - BIENVENIDA

2 - INICIO DE SESIÓN

3 – CAMBIAR CONTRASEÑA

4 – SELECCIÓN DE REPORTES

5 - REPORTE DE PAGO

6 -

UNIVERSIDAD NACIONAL AUTONOMA DE NICARAGUA
RECINTO UNIVERSITARIO RUBEN DARIO
DEPARTAMENTO DE COMPUTACION
MAESTRIA EN COMPUTACION CON ENFASIS EN SISTEMAS DE INFORMACION

INFORME DE MESES CANCELADOS

Fotografia

Carnet: 11-11111-1

Nombres y Apellidos: PATRICIA SOLEDAD MACIAS GUERRERO

Nº Recibo	Fecha Pago	Monto Dólar	Monto Córdoba	Mes Cancelado
1234567	26/10/2010	167.00	3590	SEPTIEMBRE
1234568	26/11/2010	167.00	3590	NOVIEMBRE
2566969	26/09/2010	167.00	3590,5	OCTUBRE

REPORTE DE CALIFICACIONES

UNIVERSIDAD NACIONAL AUTONOMA DE NICARAGUA
RECINTO UNIVERSITARIO RUBEN DARIO
DEPARTAMENTO DE COMPUTACION
MAESTRIA EN COMPUTACION CON ENFASIS EN SISTEMAS DE INFORMACION

Historial Academico

Carnet; 05-04048-2 **Maestría:** MC-001 **Edición:** 1

Nombre: GERALD **Apellido:** ORTEGA

Modulo	Código	Asignatura	Año	Nota
1	MC-101	NEGOCIOS ELECTRONICOS	2011	85
1	MC-102	INGENIERIA DEL SOFTWARE	2011	89

Promedio: 87.00

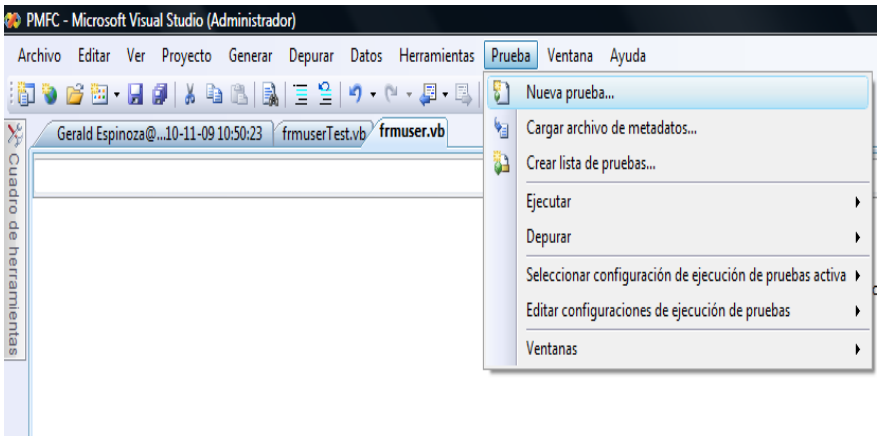
Promedio General: 87.00

NSP: No se presento
Advertencia: Este historial no tiene válidez sino lleva firma ni sello del coordinador de la maestria

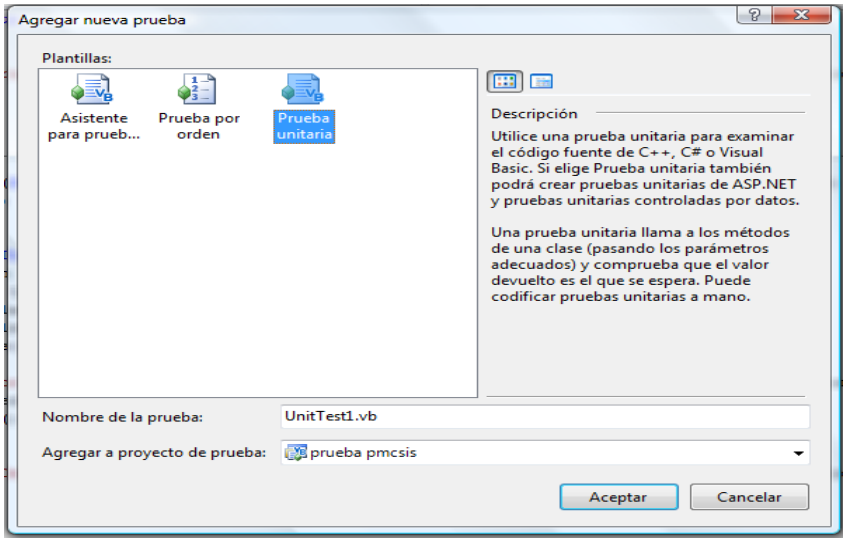
Resultado 9 – Prueba de unidad

Esta prueba tiene como objetivo encontrar defectos en los componentes o módulos del sistema, aquí se analizan los métodos y/o funciones para ver si los datos fluyen correctamente.

Para crear la prueba nos dirigimos a la barra de menú, elegimos pruebas y luego nueva prueba.



Después seleccionamos el tipo de prueba a realizar.



Y por último se codifica las pruebas necesarias para los métodos que se desee.

A continuación se muestra el código de la prueba de unidad para el módulo usuario.

Código controlador para la prueba de unidad

```
Imports Infragistics.Win.UltraWinEditors
Imports Infragistics.Win.Misc
Imports System.Windows.Forms
Imports System
Imports Microsoft.VisualStudio.TestTools.UnitTestin
Imports PMFC

'''Se trata de una clase de prueba para frmuserTest y se pretende que
'''contenga todas las pruebas unitarias frmuserTest.
<TestClass()> _
Public Class frmuserTest

    Private testContextInstance As TestContext

    '''Obtiene o establece el contexto de la prueba que proporciona
    '''la información y funcionalidad para la ejecución de pruebas
actual.
    Public Property TestContext() As TestContext
        Get
            Return testContextInstance
        End Get
        Set(ByVal value As TestContext)
            testContextInstance = Value
        End Set
    End Property

    '''Una prueba de UltraGroupBox1
    <TestMethod()> _
    Public Sub UltraGroupBox1Test()
        Dim target As frmuser = New frmuser
        Dim expected As UltraGroupBox = Nothing
        Dim actual As UltraGroupBox
        target.UltraGroupBox1 = expected
        actual = target.UltraGroupBox1
        Assert.AreEqual(expected, actual)
    End Sub

    '''Una prueba de txtrcontraseñaU
    <TestMethod()> _
    Public Sub txtrcontraseñaUTest()
        Dim target As frmuser = New frmuser
        Dim expected As UltraTextEditor = Nothing
        Dim actual As UltraTextEditor
        target.txtrcontraseñaU = expected
        actual = target.txtrcontraseñaU
        Assert.AreEqual(expected, actual)
    End Sub

    '''Una prueba de txtnomU
```

```
<TestMethod()> _
Public Sub txtnomUTest()
    Dim target As frmuser = New frmuser
    Dim expected As UltraTextEditor = Nothing
    Dim actual As UltraTextEditor
    target.txtnomU = expected
    actual = target.txtnomU
    Assert.AreEqual(expected, actual)
End Sub

  '''Una prueba de txtloginU
  <TestMethod()> _
Public Sub txtloginUTest()
    Dim target As frmuser = New frmuser
    Dim expected As UltraTextEditor = Nothing
    Dim actual As UltraTextEditor
    target.txtloginU = expected
    actual = target.txtloginU
    Assert.AreEqual(expected, actual)
End Sub

  '''Una prueba de txtcontraseñaU
  <TestMethod()> _
Public Sub txtcontraseñaUTest()
    Dim target As frmuser = New frmuser
    Dim expected As UltraTextEditor = Nothing
    Dim actual As UltraTextEditor
    target.txtcontraseñaU = expected
    actual = target.txtcontraseñaU
    Assert.AreEqual(expected, actual)
End Sub

  '''Una prueba de txtapeU
  <TestMethod()> _
Public Sub txtapeUTest()
    Dim target As frmuser = New frmuser
    Dim expected As UltraTextEditor = Nothing
    Dim actual As UltraTextEditor
    target.txtapeU = expected
    actual = target.txtapeU
    Assert.AreEqual(expected, actual)
 End Sub

'''Una prueba de nuevo
<TestMethod()> _
Public Sub nuevoTest()
    Dim target As frmuser = New frmuser
    Dim expected As UltraButton = Nothing
    Dim actual As UltraButton
    target.nuevo = expected
    actual = target.nuevo
    Assert.AreEqual(expected, actual)
End Sub
```

```
'''Una prueba de modificar
<TestMethod()> _
Public Sub modificarTest()
    Dim target As frmuser = New frmuser
    Dim expected As UltraButton = Nothing
    Dim actual As UltraButton
    target.modificar = expected
    actual = target.modificar
    Assert.AreEqual(expected, actual)
End Sub

'''Una prueba de Label6
<TestMethod()> _
Public Sub Label6Test()
    Dim target As frmuser = New frmuser
    Dim expected As Label = Nothing
    Dim actual As Label
    target.Label6 = expected
    actual = target.Label6
    Assert.AreEqual(expected, actual)
 End Sub

'''Una prueba de Label5
<TestMethod()> _
Public Sub Label5Test()
    Dim target As frmuser = New frmuser
    Dim expected As Label = Nothing
    Dim actual As Label
    target.Label5 = expected
    actual = target.Label5
    Assert.AreEqual(expected, actual)
End Sub

'''Una prueba de Label4
<TestMethod()> _
Public Sub Label4Test()
    Dim target As frmuser = New frmuser
    Dim expected As Label = Nothing
    Dim actual As Label
    target.Label4 = expected
    actual = target.Label4
    Assert.AreEqual(expected, actual)
 End Sub

'''Una prueba de Label3
 <TestMethod()> _
Public Sub Label3Test()
    Dim target As frmuser = New frmuser
    Dim expected As Label = Nothing
    Dim actual As Label
    target.Label3 = expected
    actual = target.Label3
    Assert.AreEqual(expected, actual)
End Sub
```

```
'''Una prueba de Label2
<TestMethod()> _
Public Sub Label2Test()
    Dim target As frmuser = New frmuser
    Dim expected As Label = Nothing
    Dim actual As Label
    target.Label2 = expected
    actual = target.Label2
    Assert.AreEqual(expected, actual)
End Sub

'''Una prueba de Label1
<TestMethod()> _
Public Sub Label1Test()
    Dim target As frmuser = New frmuser
    Dim expected As Label = Nothing
    Dim actual As Label
    target.Label1 = expected
    actual = target.Label1
    Assert.AreEqual(expected, actual)
 End Sub

'''Una prueba de guardar
<TestMethod()> _
Public Sub guardarTest()
    Dim target As frmuser = New frmuser
    Dim expected As UltraButton = Nothing
    Dim actual As UltraButton
    target.guardar = expected
    actual = target.guardar
    Assert.AreEqual(expected, actual)
End Sub

'''Una prueba de Gbusuarios
<TestMethod()> _
Public Sub GbusuariosTest()
    Dim target As frmuser = New frmuser
    Dim expected As UltraGroupBox = Nothing
    Dim actual As UltraGroupBox
    target.Gbusuarios = expected
    actual = target.Gbusuarios
    Assert.AreEqual(expected, actual)
 End Sub

'''Una prueba de eliminar
<TestMethod()> _
Public Sub eliminarTest()
    Dim target As frmuser = New frmuser
    Dim expected As UltraButton = Nothing
    Dim actual As UltraButton
    target.eliminar = expected
    actual = target.eliminar
    Assert.AreEqual(expected, actual)
End Sub
```

```
'''Una prueba de cerrar
    <TestMethod()> _
    Public Sub cerrarTest()
        Dim target As frmuser = New frmuser
        Dim expected As UltraButton = Nothing
        Dim actual As UltraButton
        target.cerrar = expected
        actual = target.cerrar
        Assert.AreEqual(expected, actual)
     End Sub

    '''Una prueba de CbxtipousuarioU
    <TestMethod()> _
    Public Sub CbxtipousuarioUTest()
        Dim target As frmuser = New frmuser
        Dim expected As UltraComboEditor = Nothing
        Dim actual As UltraComboEditor
        target.CbxtipousuarioU = expected
        actual = target.CbxtipousuarioU
        Assert.AreEqual(expected, actual)
    End Sub

    '''Una prueba de buscar
    <TestMethod()> _
    Public Sub buscarTest()
        Dim target As frmuser = New frmuser
        Dim expected As UltraButton = Nothing
        Dim actual As UltraButton
        target.buscar = expected
        actual = target.buscar
        Assert.AreEqual(expected, actual)
    End Sub

    '''Una prueba de txtrcontraseñaU_KeyPress
    <TestMethod(), _
     DeploymentItem("PMFC.exe")> _
    Public Sub txtrcontraseñaU_KeyPressTest()
        Dim target As frmuser_Accessor = New frmuser_Accessor
        Dim sender As Object = Nothing
        Dim e As KeyPressEventArgs = Nothing

        sender = target.txtrcontraseñaU
        target.txtrcontraseñaU.Text = "123"
        target.txtrcontraseñaU_KeyPress(sender, e)
    End Sub
   '''Una prueba de txtnomU_KeyPress
     <TestMethod(), _
     DeploymentItem("PMFC.exe")> _
    Public Sub txtnomU_KeyPressTest()
        Dim target As frmuser_Accessor = New frmuser_Accessor
        Dim sender As Object = Nothing
        Dim e As KeyPressEventArgs = Nothing

        sender = target.txtnomU
        target.txtnomU.Text = "gerald"
        target.txtnomU_KeyPress(sender, e)
    End Sub
```

```
'''Una prueba de txtloginU_KeyPress
<TestMethod(), _
 DeploymentItem("PMFC.exe")> _
Public Sub txtloginU_KeyPressTest()
    Dim target As frmuser_Accessor = New frmuser_Accessor
    Dim sender As Object = Nothing
    Dim e As KeyPressEventArgs = Nothing

    sender = target.txtloginU
    target.txtloginU.Text = "gerald87"
    target.txtloginU_KeyPress(sender, e)
End Sub

'''Una prueba de txtcontraseñaU_KeyPress
<TestMethod(), _
 DeploymentItem("PMFC.exe")> _
Public Sub txtcontraseñaU_KeyPressTest()
    Dim target As frmuser_Accessor = New frmuser_Accessor
    Dim sender As Object = Nothing
    Dim e As KeyPressEventArgs = Nothing
    sender = target.txtcontraseñaU
    target.txtcontraseñaU.Text = "123"
    target.txtcontraseñaU_KeyPress(sender, e)
End Sub

'''Una prueba de txtapeU_KeyPress
 <TestMethod(), _
 DeploymentItem("PMFC.exe")> _
Public Sub txtapeU_KeyPressTest()
    Dim target As frmuser_Accessor = New frmuser_Accessor
    Dim sender As Object = Nothing
    Dim e As KeyPressEventArgs = Nothing
    sender = target.txtapeU
    target.txtapeU.Text = "ortega"
    target.txtapeU_KeyPress(sender, e)
End Sub

'''Una prueba de nuevo_Click
<TestMethod(), _
 DeploymentItem("PMFC.exe")> _
Public Sub nuevo_ClickTest()
    Dim target As frmuser_Accessor = New frmuser_Accessor
    Dim sender As Object = Nothing
    Dim e As EventArgs = Nothing
    target.nuevo_Click(sender, e)
End Sub

'''Una prueba de modificar_Click
<TestMethod(), _
 DeploymentItem("PMFC.exe")> _
Public Sub modificar_ClickTest()
    Dim target As frmuser_Accessor = New frmuser_Accessor
    Dim sender As Object = Nothing
    Dim e As EventArgs = Nothing
    target.modificar_Click(sender, e)
End Sub
```

```
'''Una prueba de limpiar
    <TestMethod(), _
     DeploymentItem("PMFC.exe")> _
    Public Sub limpiarTest()
        Dim target As frmuser_Accessor = New frmuser_Accessor
        target.limpiar()
    End Sub

    '''Una prueba de InitializeComponent
    <TestMethod(), _
     DeploymentItem("PMFC.exe")> _
    Public Sub InitializeComponentTest()
        Dim target As frmuser_Accessor = New frmuser_Accessor
        target.InitializeComponent()
    End Sub

    '''Una prueba de inicio
    <TestMethod(), _
     DeploymentItem("PMFC.exe")> _
    Public Sub inicioTest()
        Dim target As frmuser_Accessor = New frmuser_Accessor
        target.inicio()
    End Sub

    '''Una prueba de guardar_reg
    <TestMethod(), _
     DeploymentItem("PMFC.exe")> _
    Public Sub guardar_regTest()
        Dim target As frmuser_Accessor = New frmuser_Accessor
        target.guardar_reg()
    End Sub

    '''Una prueba de guardar_Click
     <TestMethod(), _
     DeploymentItem("PMFC.exe")> _
    Public Sub guardar_ClickTest()
        Dim target As frmuser_Accessor = New frmuser_Accessor
        Dim sender As Object = Nothing
        Dim e As EventArgs = Nothing
        target.guardar_Click(sender, e)
    End Sub

   '''Una prueba de frmuser_Load
    <TestMethod(), _
     DeploymentItem("PMFC.exe")> _
    Public Sub frmuser_LoadTest()
        Dim target As frmuser_Accessor = New frmuser_Accessor
        Dim sender As Object = Nothing
        Dim e As EventArgs = Nothing
        target.frmuser_Load(sender, e)
    End Sub
```

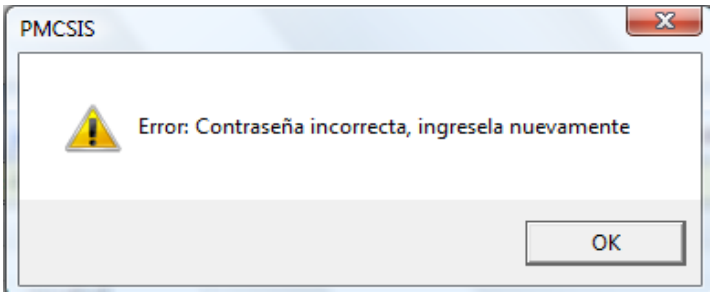
```
'''Una prueba de eliminar_reg
<TestMethod(), _
 DeploymentItem("PMFC.exe")> _
Public Sub eliminar_regTest()
    Dim target As frmuser_Accessor = New frmuser_Accessor
    target.eliminar_reg()
End Sub

'''Una prueba de eliminar_Click
<TestMethod(), _
 DeploymentItem("PMFC.exe")> _
Public Sub eliminar_ClickTest()
    Dim target As frmuser_Accessor = New frmuser_Accessor
    Dim sender As Object = Nothing
    Dim e As EventArgs = Nothing
    target.eliminar_Click(sender, e)
End Sub

'''Una prueba de Dispose
 <TestMethod(), _
 DeploymentItem("PMFC.exe")> _
Public Sub DisposeTest()
    Dim target As frmuser_Accessor = New frmuser_Accessor
    Dim disposing As Boolean = False
    target.Dispose(disposing)
End Sub

'''Una prueba de contador_datos
<TestMethod(), _
 DeploymentItem("PMFC.exe")> _
Public Sub contador_datosTest()
    Dim target As frmuser_Accessor = New frmuser_Accessor
    target.contador_datos()
End Sub

'''Una prueba de cerrar_Click
<TestMethod(), _
 DeploymentItem("PMFC.exe")> _
Public Sub cerrar_ClickTest()
    Dim target As frmuser_Accessor = New frmuser_Accessor
    Dim sender As Object = Nothing
    Dim e As EventArgs = Nothing
    target.cerrar_Click(sender, e)
End Sub

'''Una prueba de CbxtipousuarioU_KeyPress
<TestMethod(), _
 DeploymentItem("PMFC.exe")> _
Public Sub CbxtipousuarioU_KeyPressTest()
    Dim target As frmuser_Accessor = New frmuser_Accessor
    Dim sender As Object = Nothing
    Dim e As KeyPressEventArgs = Nothing
    target.CbxtipousuarioU_KeyPress(sender, e)
End Sub
```

```
    '''Una prueba de buscar_reg
    <TestMethod(), _
     DeploymentItem("PMFC.exe")> _
    Public Sub buscar_regTest()
        Dim target As frmuser_Accessor = New frmuser_Accessor
        target.buscar_reg()
    End Sub

    '''Una prueba de buscar_Click
    <TestMethod(), _
     DeploymentItem("PMFC.exe")> _
    Public Sub buscar_ClickTest()
        Dim target As frmuser_Accessor = New frmuser_Accessor
        Dim sender As Object = Nothing
        Dim e As EventArgs = Nothing
        target.buscar_Click(sender, e)
    End Sub

    '''Una prueba de Constructor frmuser
    <TestMethod()> _
    Public Sub frmuserConstructorTest()
        Dim target As frmuser = New frmuser
    End Sub
End Class
```

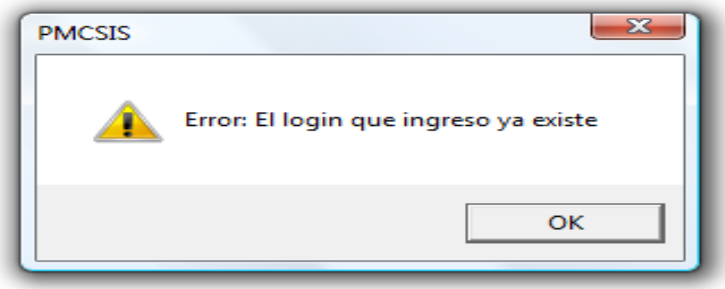
Ejecución de la pruebas de unidad

Al ejecutar la prueba de unidad, muestra los siguientes mensajes, que corresponden al modulo de usuarios.

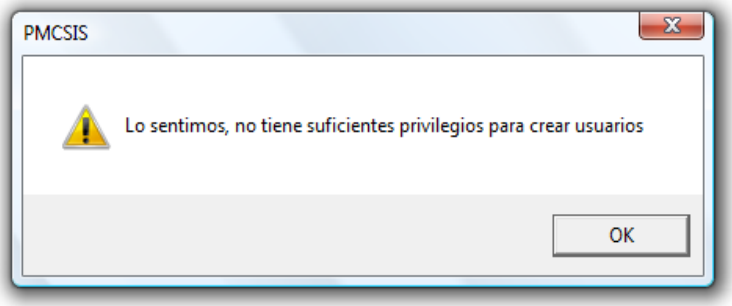
Al ingresar una contraseña invalida



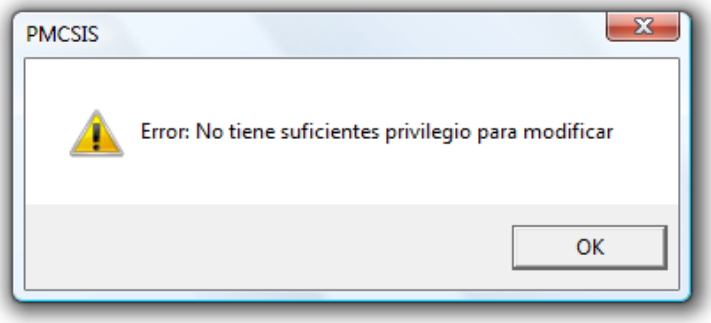
Al crear un login que ya existe



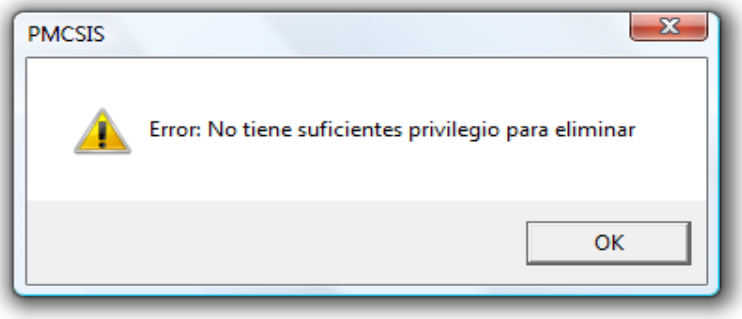
Cuando un usuario con cuenta de invitado desea crear un usuario



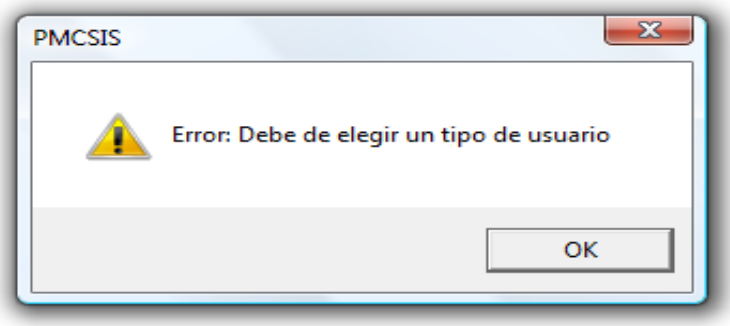
Cuando un usuario con cuenta de invitado desea modificar un usuario



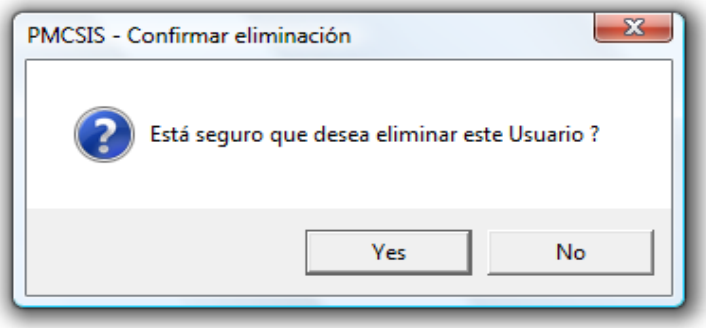
Cuando un usuario con cuenta de invitado desea eliminar un usuario



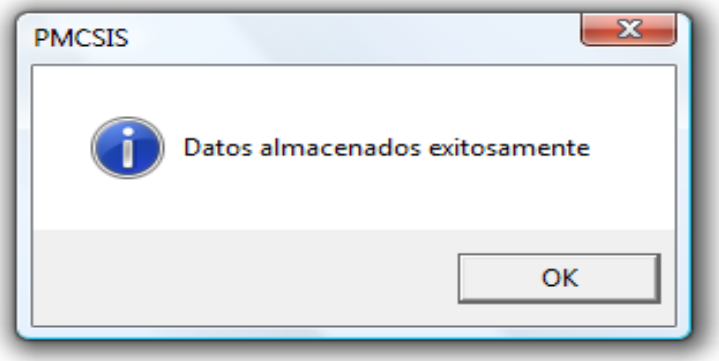
Cuando se crea un usuario y no se selecciona el tipo de usuario



Cuando un usuario con cuenta de administrados desea eliminar un usuario



Cuando se crea exitosamente un usuario.



Al finalizar la prueba se obtienen los resultados, a como se puede observar en la imagen siguiente el módulo completo todas las pruebas de manera satisfactoria.

Resultados de pruebas

Gerald Espinoza@GERALD-PC 201 | Ejecutar | Depurar | Agrupar por: [Ninguno] | [Todas las c | <Escribir palabra clave

Ejecución de pruebas completado Resultados: 94/94 correctas; Elementos comprobados: 0

Resultado	Nombre de la prueba	Proyecto	Mensaje de error
Pasada	UltraGroupBox1Test	prueba2	
Pasada	txtrcontraseñaUTest	prueba2	
Pasada	txtnomUTest	prueba2	
Pasada	txtloginUTest	prueba2	
Pasada	txtcontraseñaUTest	prueba2	
Pasada	txtapeUTest	prueba2	
Pasada	nuevoTest	prueba2	
Pasada	modificarTest	prueba2	
Pasada	Label6Test	prueba2	
Pasada	Label5Test	prueba2	
Pasada	Label4Test	prueba2	

Este es un resumen de los resultados de la prueba de unidad.

Resumen de resultados

Nombre de la ejecución de pruebas:	Gerald Espinoza@GERALD-PC 2010-11-09 15:45:12
Resultado de la ejecución:	94/94 pruebas correctas, 0 incorrectas, 0 omitidas
Configuración de ejecución:	Ejecución de pruebas local
Enviada por:	Gerald-PC\Gerald Espinoza
Iniciado:	09/11/2010 03:45:17 p.m.
Completada en:	09/11/2010 04:01:17 p.m.

En el caso de que se hubiera encontrado algún error en las pruebas los errores se reflejarían en los resultados de las pruebas como lo muestra la siguiente imagen.

Resultados de pruebas

Gerald Espinoza@GERALD-PC 201 | Ejecutar | Depurar | Agrupar por: [Ninguno] | [Todas las c | <Escribir palabra clave

Ejecución de pruebas error Resultados: 21/94 correctas; Elementos comprobados: 1

Resultado	Nombre de la prueba	Proyecto	Mensaje de error
Pasada	CbxtipousuarioUTest	prueba2	
Pasada	buscarTest	prueba2	
Error	txtrcontraseñaU_KeyPressTest	prueba2	Test method prueba2.frmuserTest.txtrcontraseñaU_KeyPressT
Pasada	txtnomU_KeyPressTest	prueba2	
En curso	txtloginU_KeyPressTest	prueba2	

A continuación se muestra el código de la prueba de unidad para el módulo inicio de sesión.

Código controlador para la prueba de unidad

```
Imports Infragistics.Win.UltraWinToolbars
Imports System.Windows.Forms
Imports System
Imports Microsoft.VisualStudio.TestTools.UnitTesting
Imports PMFC

<TestClass()> _
Public Class LoginForm1Test

    Private testContextInstance As TestContext

    Public Property TestContext() As TestContext
        Get
            Return testContextInstance
        End Get
        Set(ByVal value As TestContext)
            testContextInstance = Value
        End Set
    End Property

    '''Una prueba de UltraToolbarsManager1
    <TestMethod(), _
     DeploymentItem("PMFC.exe")> _
    Public Sub UltraToolbarsManager1Test()
        Dim target As LoginForm1_Accessor = New LoginForm1_Accessor
        Dim expected As UltraToolbarsManager = Nothing
        Dim actual As UltraToolbarsManager
        target.UltraToolbarsManager1 = expected
        actual = target.UltraToolbarsManager1
        Assert.AreEqual(expected, actual)
    End Sub

    '''Una prueba de txtusername
    <TestMethod(), _
     DeploymentItem("PMFC.exe")> _
    Public Sub txtusernameTest()
        Dim target As LoginForm1_Accessor = New LoginForm1_Accessor
        Dim expected As TextBox = Nothing
        Dim actual As TextBox
        target.txtusername = expected
        actual = target.txtusername
        Assert.AreEqual(expected, actual)
    End Sub
```

```
    '''Una prueba de txtpassword
    <TestMethod(), _
     DeploymentItem("PMFC.exe")> _
    Public Sub txtpasswordTest()
        Dim target As LoginForm1_Accessor = New LoginForm1_Accessor
        Dim expected As TextBox = Nothing
        Dim actual As TextBox
        target.txtpassword = expected
        actual = target.txtpassword
        Assert.AreEqual(expected, actual)
    End Sub

    '''Una prueba de PictureBox1
    <TestMethod(), _
     DeploymentItem("PMFC.exe")> _
    Public Sub PictureBox1Test()
        Dim target As LoginForm1_Accessor = New LoginForm1_Accessor
        Dim expected As PictureBox = Nothing
        Dim actual As PictureBox
        target.PictureBox1 = expected
        actual = target.PictureBox1
        Assert.AreEqual(expected, actual)
    End Sub
      '''Una prueba de Panel1
    <TestMethod(), _
     DeploymentItem("PMFC.exe")> _
    Public Sub Panel1Test()
        Dim target As LoginForm1_Accessor = New LoginForm1_Accessor
        Dim expected As Panel = Nothing
        Dim actual As Panel
        target.Panel1 = expected
        actual = target.Panel1
        Assert.AreEqual(expected, actual)
    End Sub

    '''Una prueba de LoginForm1_Fill_Panel
    <TestMethod(), _
     DeploymentItem("PMFC.exe")> _
    Public Sub LoginForm1_Fill_PanelTest()
        Dim target As LoginForm1_Accessor = New LoginForm1_Accessor
        Dim expected As Panel = Nothing
        Dim actual As Panel
        target.LoginForm1_Fill_Panel = expected
        actual = target.LoginForm1_Fill_Panel
        Assert.AreEqual(expected, actual)
End Sub
    '''Una prueba de Label2
    <TestMethod(), _
     DeploymentItem("PMFC.exe")> _
    Public Sub Label2Test()
        Dim target As LoginForm1_Accessor = New LoginForm1_Accessor
        Dim expected As Label = Nothing
        Dim actual As Label
        target.Label2 = expected
        actual = target.Label2
        Assert.AreEqual(expected, actual)
    End Sub
```

```
'''Una prueba de Label1
<TestMethod(), _
 DeploymentItem("PMFC.exe")> _
Public Sub Label1Test()
    Dim target As LoginForm1_Accessor = New LoginForm1_Accessor
    Dim expected As Label = Nothing
    Dim actual As Label
    target.Label1 = expected
    actual = target.Label1
    Assert.AreEqual(expected, actual)
End Sub
'''Una prueba de BtnOK
<TestMethod(), _
 DeploymentItem("PMFC.exe")> _
Public Sub BtnOKTest()
    Dim target As LoginForm1_Accessor = New LoginForm1_Accessor
    Dim expected As Button = Nothing
    Dim actual As Button
    target.BtnOK = expected
    actual = target.BtnOK
    Assert.AreEqual(expected, actual)
End Sub
 '''Una prueba de Btncancel
<TestMethod(), _
 DeploymentItem("PMFC.exe")> _
Public Sub BtncancelTest()
    Dim target As LoginForm1_Accessor = New LoginForm1_Accessor
    Dim expected As Button = Nothing
    Dim actual As Button
    target.Btncancel = expected
    actual = target.Btncancel
    Assert.AreEqual(expected, actual)
End Sub
'''Una prueba de _LoginForm1_Toolbars_Dock_Area_Top
<TestMethod(), _
 DeploymentItem("PMFC.exe")> _
Public Sub LoginForm1_Toolbars_Dock_Area_TopTest()
    Dim target As LoginForm1_Accessor = New LoginForm1_Accessor
    Dim expected As UltraToolbarsDockArea = Nothing
    Dim actual As UltraToolbarsDockArea
    target._LoginForm1_Toolbars_Dock_Area_Top = expected
    actual = target._LoginForm1_Toolbars_Dock_Area_Top
    Assert.AreEqual(expected, actual)
End Sub

'''Una prueba de _LoginForm1_Toolbars_Dock_Area_Right
<TestMethod(), _
 DeploymentItem("PMFC.exe")> _
Public Sub LoginForm1_Toolbars_Dock_Area_RightTest()
    Dim target As LoginForm1_Accessor = New LoginForm1_Accessor
    Dim expected As UltraToolbarsDockArea = Nothing
    Dim actual As UltraToolbarsDockArea
    target._LoginForm1_Toolbars_Dock_Area_Right = expected
    actual = target._LoginForm1_Toolbars_Dock_Area_Right
    Assert.AreEqual(expected, actual)
End Sub
```

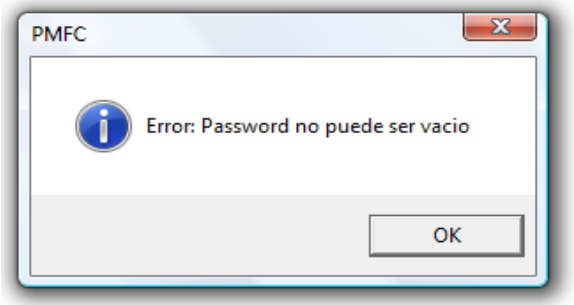
```
'''Una prueba de _LoginForm1_Toolbars_Dock_Area_Left
<TestMethod(), _
 DeploymentItem("PMFC.exe")> _
Public Sub LoginForm1_Toolbars_Dock_Area_LeftTest()
    Dim target As LoginForm1_Accessor = New LoginForm1_Accessor
    Dim expected As UltraToolbarsDockArea = Nothing
    Dim actual As UltraToolbarsDockArea
    target._LoginForm1_Toolbars_Dock_Area_Left = expected
    actual = target._LoginForm1_Toolbars_Dock_Area_Left
    Assert.AreEqual(expected, actual)
End Sub
'''Una prueba de _LoginForm1_Toolbars_Dock_Area_Bottom
<TestMethod(), _
 DeploymentItem("PMFC.exe")> _
Public Sub LoginForm1_Toolbars_Dock_Area_BottomTest()
    Dim target As LoginForm1_Accessor = New LoginForm1_Accessor
    Dim expected As UltraToolbarsDockArea = Nothing
    Dim actual As UltraToolbarsDockArea
    target._LoginForm1_Toolbars_Dock_Area_Bottom = expected
    actual = target._LoginForm1_Toolbars_Dock_Area_Bottom
    Assert.AreEqual(expected, actual)
End Sub

'''Una prueba de txtusername_KeyPress
<TestMethod(), _
 DeploymentItem("PMFC.exe")> _
Public Sub txtusername_KeyPressTest()
    Dim target As LoginForm1_Accessor = New LoginForm1_Accessor
    Dim sender As Object = Nothing
    Dim e As KeyPressEventArgs = Nothing
    sender = target.txtusername
    target.txtusername.Text = "gerald87"
    target.txtusername_KeyPress(sender, e)
End Sub

'''Una prueba de Txtpassword_KeyPress
<TestMethod(), _
 DeploymentItem("PMFC.exe")> _
Public Sub Txtpassword_KeyPressTest()
    Dim target As LoginForm1_Accessor = New LoginForm1_Accessor
    Dim sender As Object = Nothing
    Dim e As KeyPressEventArgs = Nothing
    sender = target.txtpassword
    target.txtpassword.Text = "123"
    target.Txtpassword_KeyPress(sender, e)
End Sub

'''Una prueba de OK_Click
<TestMethod(), _
 DeploymentItem("PMFC.exe")> _
Public Sub OK_ClickTest()
    Dim target As LoginForm1_Accessor = New LoginForm1_Accessor
    Dim sender As Object = Nothing
    Dim e As EventArgs = Nothing
    target.OK_Click(sender, e)
End Sub
```

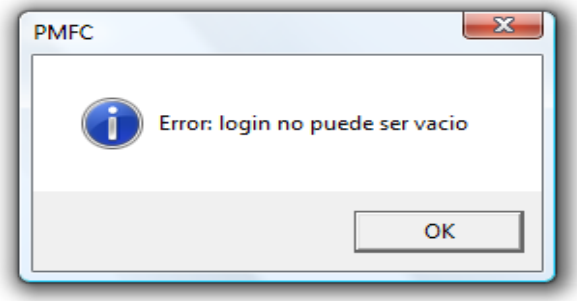
```
    '''Una prueba de LoginForm1_Load
    <TestMethod(), _
     DeploymentItem("PMFC.exe")> _
    Public Sub LoginForm1_LoadTest()
        Dim target As LoginForm1_Accessor = New LoginForm1_Accessor
        Dim sender As Object = Nothing
        Dim e As EventArgs = Nothing
        target.LoginForm1_Load(sender, e)
    End Sub
'''Una prueba de InitializeComponent
    <TestMethod(), _
     DeploymentItem("PMFC.exe")> _
    Public Sub InitializeComponentTest()
        Dim target As LoginForm1_Accessor = New LoginForm1_Accessor '
        target.InitializeComponent()
    End Sub

    '''Una prueba de Dispose
    <TestMethod(), _
     DeploymentItem("PMFC.exe")> _
    Public Sub DisposeTest()
        Dim target As LoginForm1_Accessor = New LoginForm1_Accessor
        Dim disposing As Boolean = False
        target.Dispose(disposing)
    End Sub

    '''Una prueba de contador
    <TestMethod(), _
     DeploymentItem("PMFC.exe")> _
    Public Sub contadorTest()
        Dim target As LoginForm1_Accessor = New LoginForm1_Accessor
        target.contador()
    End Sub

    '''Una prueba de conectar
    <TestMethod(), _
     DeploymentItem("PMFC.exe")> _
    Public Sub conectarTest()
        Dim target As LoginForm1_Accessor = New LoginForm1_Accessor
        target.conectar()
    End Sub

    '''Una prueba de Cancel_Click
    <TestMethod(), _
     DeploymentItem("PMFC.exe")> _
    Public Sub Cancel_ClickTest()
        Dim target As LoginForm1_Accessor = New LoginForm1_Accessor
        Dim sender As Object = Nothing
        Dim e As EventArgs = Nothing
        target.Cancel_Click(sender, e)
    End Sub
    '''Una prueba de Constructor LoginForm1
    <TestMethod()> _
    Public Sub LoginForm1ConstructorTest()
        Dim target As LoginForm1 = New LoginForm1
    End Sub
End Class
```

Ejecución de la pruebas de unidad

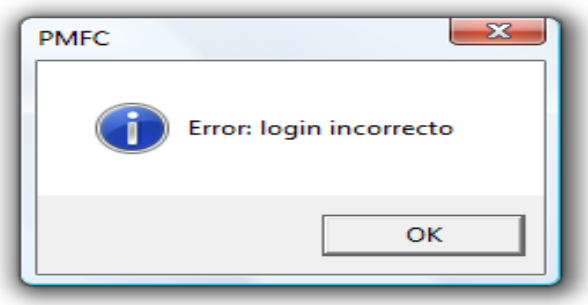
Al ejecutar la prueba de unidad, muestra los siguientes mensajes, que corresponden al módulo de usuarios

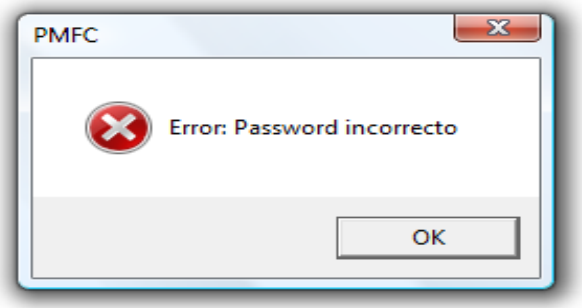
En caso de dejar vacio los campos login y/o password





En caso de un usuario invalido o contraseña incorrecta





Al finalizar la prueba se obtienen los resultados, a como se puede observar en la imagen siguiente el módulo completo todas las pruebas de manera satisfactoria.

Resultados de pruebas

Gerald Espinoza@GERALD-PC 201 | Ejecutar | Depurar

Ejecución de pruebas completado Resultados: 24/24 correctas; Elementos comprobados: 0

Resultado	Nombre de la prueba	Proyecto	Mensaje de error
Pasada	UltraToolbarsManager1Test	prueba login	
Pasada	txtusernameTest	prueba login	
Pasada	txtpasswordTest	prueba login	
Pasada	PictureBox1Test	prueba login	
Pasada	Panel1Test	prueba login	
Pasada	LoginForm1_Fill_PanelTest	prueba login	
Pasada	Label2Test	prueba login	
Pasada	Label1Test	prueba login	
Pasada	BtnOKTest	prueba login	
Pasada	BtncancelTest	prueba login	
Pasada	txtusername_KeyPressTest	prueba login	
Pasada	Txtpassword_KeyPressTest	prueba login	
Pasada	OK_ClickTest	prueba login	
Pasada	LoginForm1_LoadTest	prueba login	
Pasada	InitializeComponentTest	prueba login	

Este es un resumen de los resultados de la prueba de unidad.

Resumen de resultados

Nombre de la ejecución de pruebas:	Gerald Espinoza@GERALD-PC 2010-11-10 15:10:38
Resultado de la ejecución:	24/24 pruebas correctas, 0 incorrectas, 0 omitidas
Configuración de ejecución:	Ejecución de pruebas local
Enviada por:	Gerald-PC\Gerald Espinoza
Iniciado:	10/11/2010 03:10:43 p.m.
Completada en:	10/11/2010 03:18:14 p.m.

En el caso de que se hubiera encontrado algún error en las pruebas los errores se reflejarían en los resultados de las pruebas como lo muestra la siguiente imagen.

Resultados de pruebas

Gerald Espinoza@GERALD-PC 201 | Ejecutar | Depurar

Ejecución de pruebas error Resultados: 23/24 correctas; Elementos comprobados: 1

Resultado	Nombre de la prueba	Proyecto	Mensaje de error
Pasada	UltraToolbarsManager1Test	prueba login	
Pasada	txtusernameTest	prueba login	
Pasada	txtpasswordTest	prueba login	
Pasada	PictureBox1Test	prueba login	
Pasada	Panel1Test	prueba login	
Pasada	LoginForm1_Fill_PanelTest	prueba login	
Pasada	Label2Test	prueba login	
Pasada	Label1Test	prueba login	
Pasada	BtnOKTest	prueba login	
Pasada	BtncancelTest	prueba login	
Pasada	txtusername_KeyPressTest	prueba login	
Error	Txtpassword_KeyPressTest	prueba login	Test method prueba_
Pasada	OK_ClickTest	prueba login	
Pasada	LoginForm1_LoadTest	prueba login	
Pasada	InitializeComponentTest	prueba login	

Resultado 10 – Prueba de integración

Pantallas a utilizar: LoginForm, Ingresar Usuario las cuales se utilizaron para las pruebas de unidad.

- Registrar Usuario
- LoginForm

Estados del Ejemplo Planteado

Elemento	Estado
Login	Presente, Ausente
Contraseña	Presente, Ausente
Diálogos	Presente, Ausente
Base de Datos	Activa, Con Problemas

Valores de Entrada a Considerar

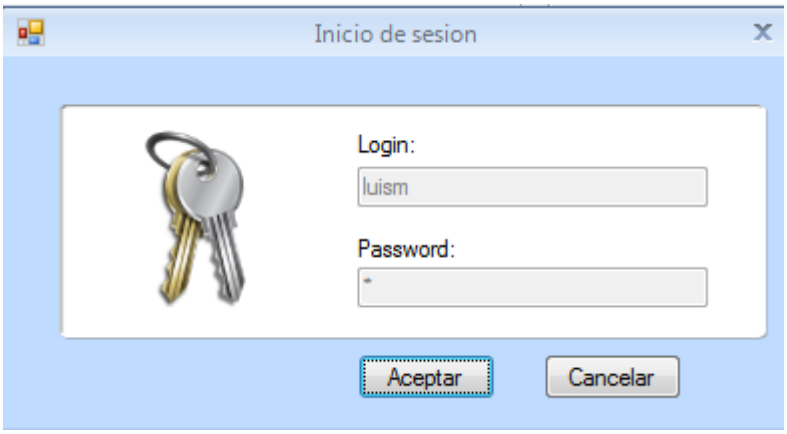
Elemento	Estado
Gerald87, admin1234	(Nombre y Contraseña Registrada)
Gerald87, admin	(Nombre , Contraseña no Registrada)
Gerald87,	(Falta Contraseña)
,admin1234	(Falta Nombre)
Luis, 546hhd	(Nombre no registrado)

Casos de Pruebas

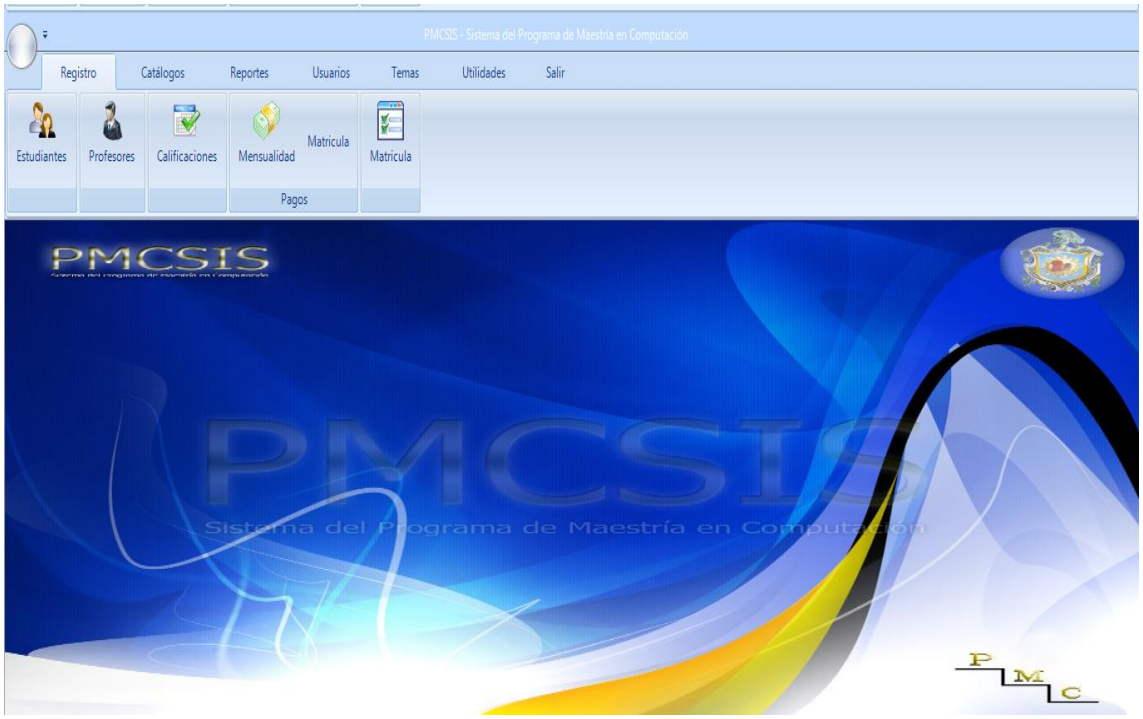
Entrada	Condiciones de Entrada	Salida Esperada
1. Gerald87, admin1234	• Login presente • Contraseña presente • Bases de Datos Activa	Entrar a la Pantalla principal de la Aplicación (Administración del Sistema)
2. Gerald87, admin1234	• Base de Datos con problemas	Falla de Ejecución
3. Gerald87,	• Login Presente • Contraseña Ausente • Base de Datos Activa	Mensaje de Error de Contraseña
4. ,admin1234	• Login Ausente • Base de Datos Activa	Mensaje de Error de Login
5. Luis, 546hhd	• Login no existe • Base de Datos Activa	Mensaje de Error de Login
6. Gerald87, admin	• Login Presente • Contraseña Incorrecta • Base de Datos Activa	Mensaje de Error de Contraseña Incorrecta

Pasos a seguir

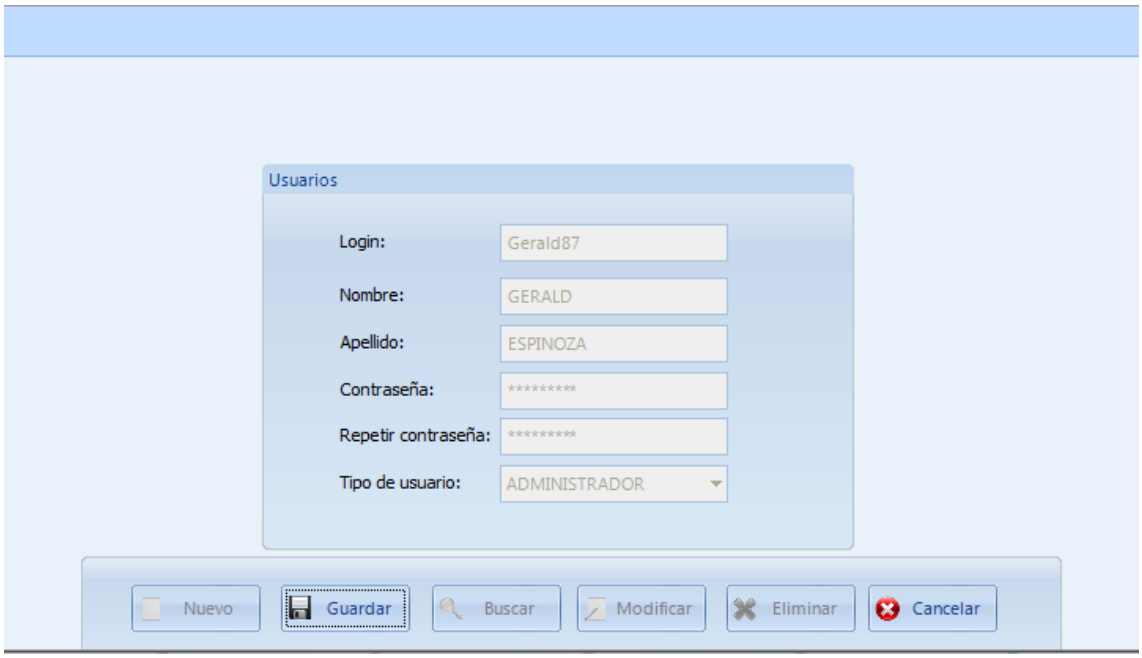
1. Ingresar al sistema con un usuario ya creado



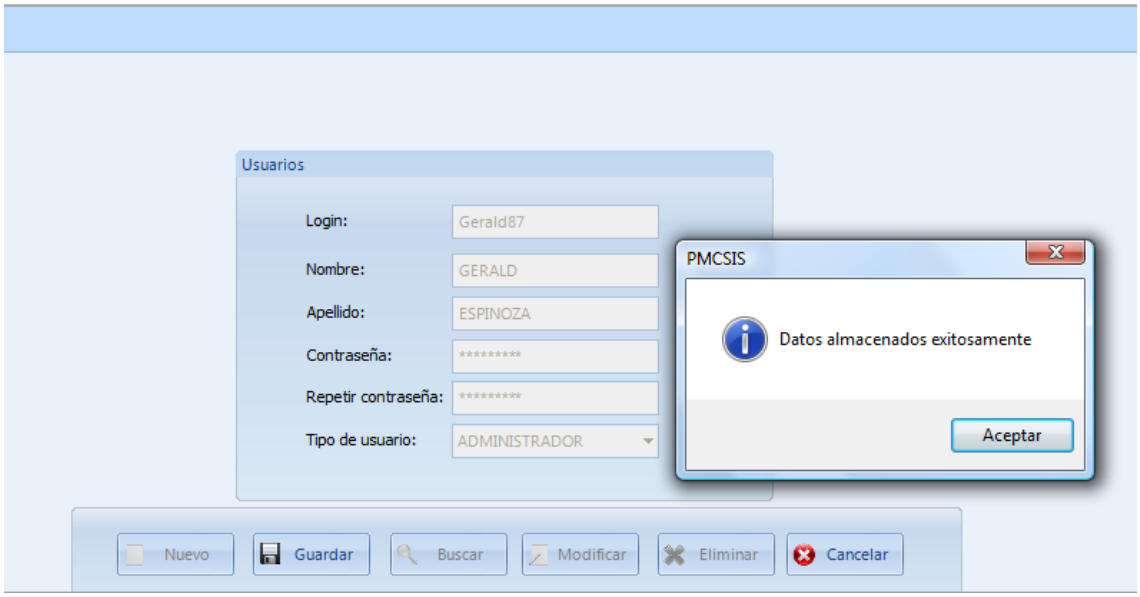
Pantalla Principal



Agregar Usuario



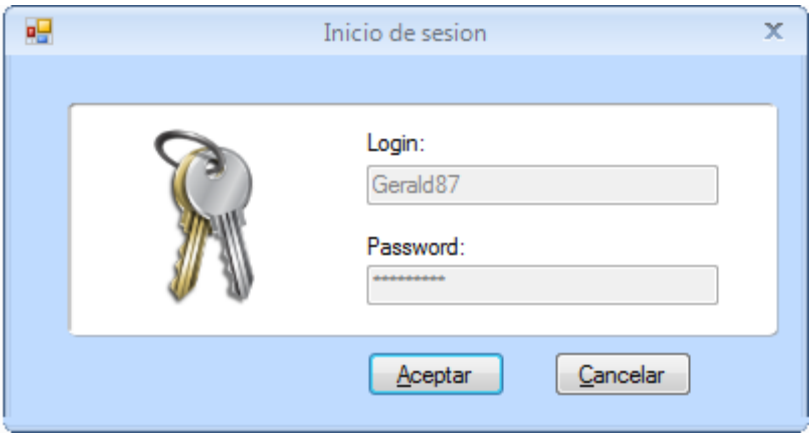
Damos Clic en Aceptar y cerramos la Aplicación



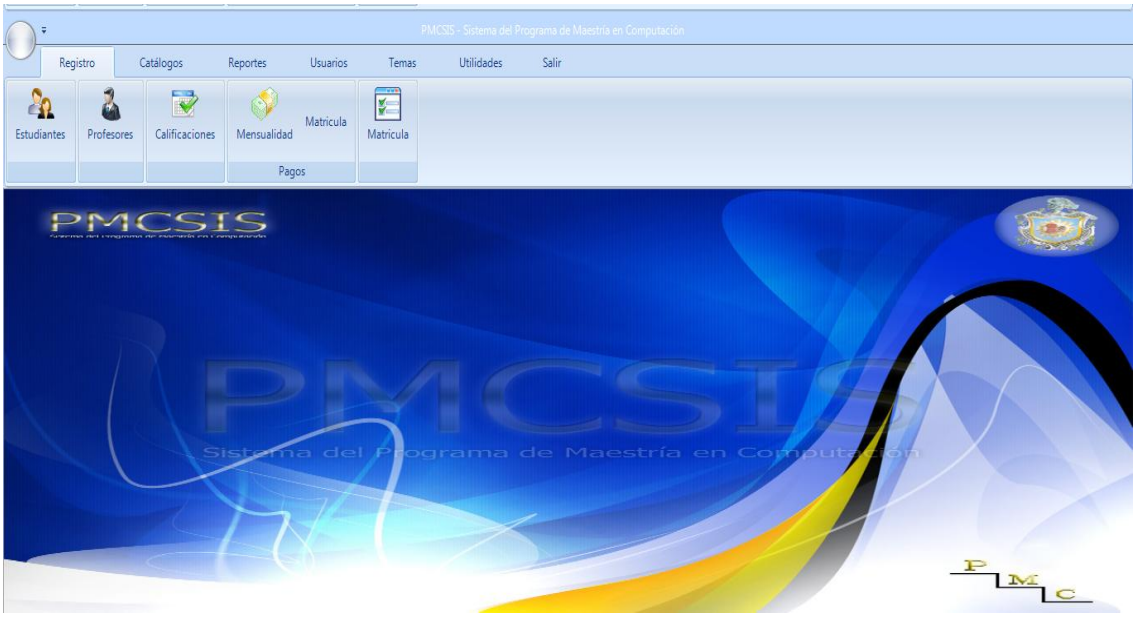
Probar el usuario Agregado

Entradas

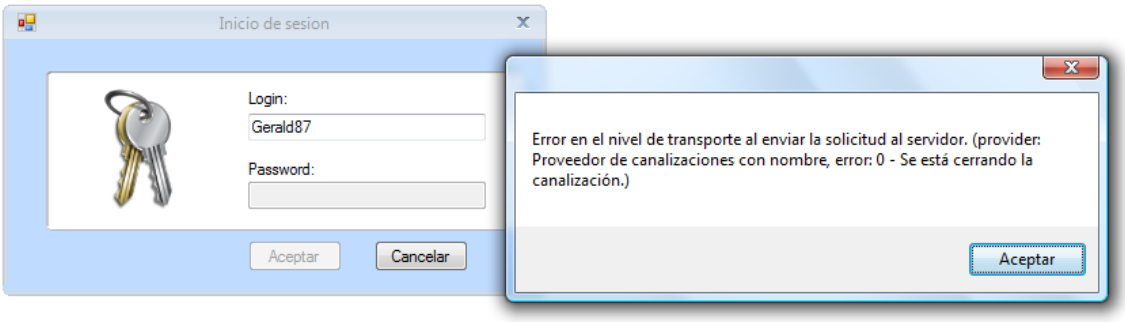
1. Gerald87, admin1234



Pantalla Principal

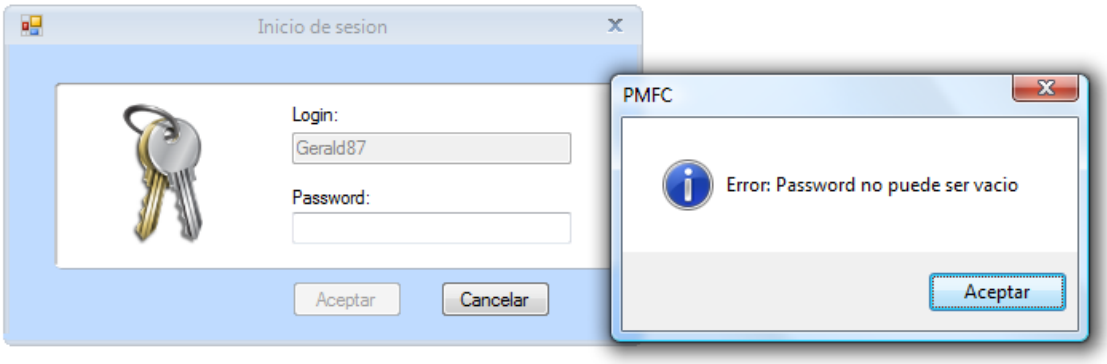


2. Gerald87, admin1234

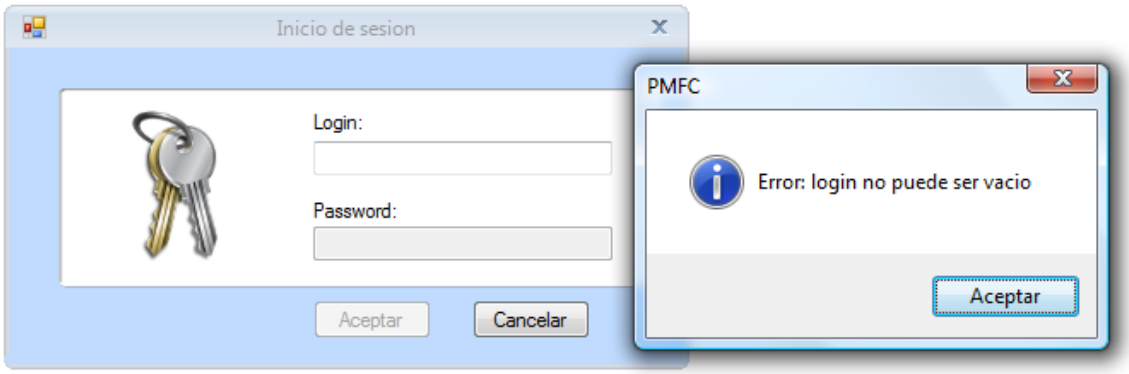


No encuentra Conexión con la Base de datos, por tanto nos manda una falla en la Ejecución.

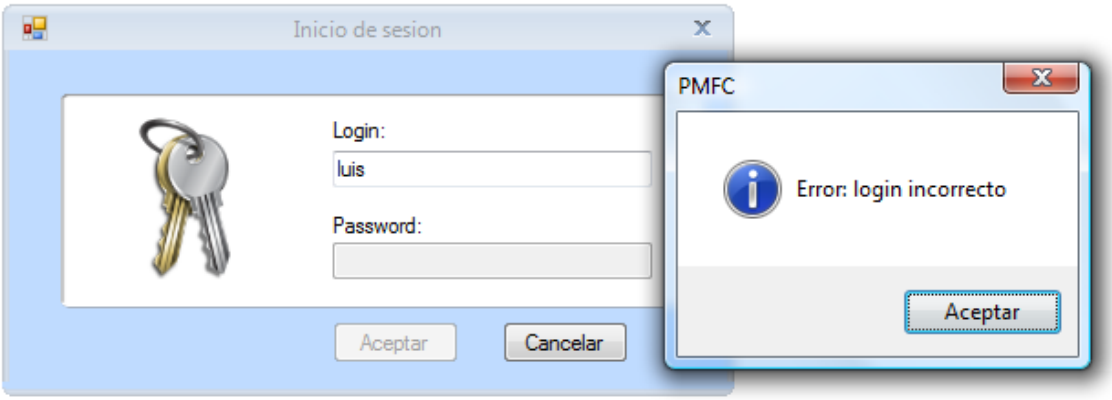
3. Gerald87,



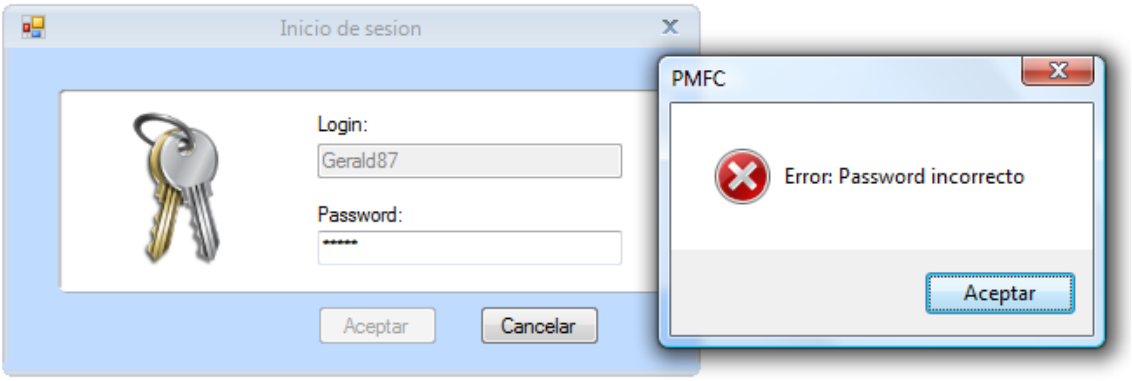
4. ,admin1234



5. Luis, 546hhd



6. Gerald87, admin



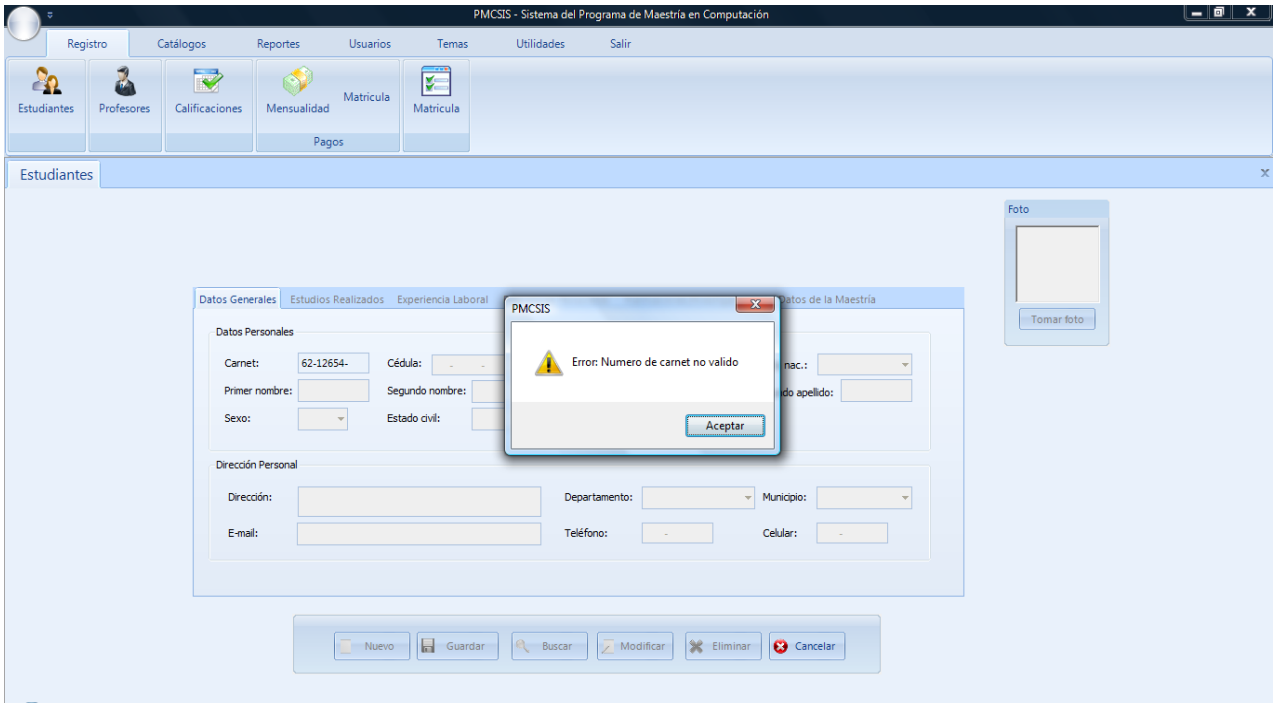
Resultado 11 – Prueba de entrega

La prueba funcional sirve para verificar que cada procedimiento de cada módulo se está realizando correctamente.

Caso de Uso: Procedimientos para ingresar a un estudiante correctamente.

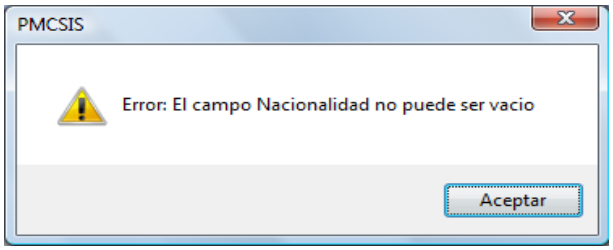
Procedimiento

- Ingreso número de carnet del estudiante que es un campo de 8 caracteres.
- Si el número de carnet es correcto me permitirá ingresar el resto de la información, en caso contrario no me permitirá continuar y enviara un mensaje de Número de carnet no valido.



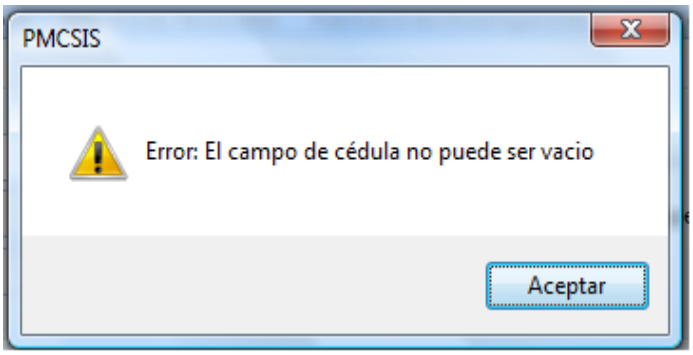
- En el caso de que el número de carnet sea correcto se podrá ingresar los datos del estudiante.
- En el campo de Nacionalidad se escoge la nacionalidad del estudiante (no puede quedar vacio).

Mensaje de Error

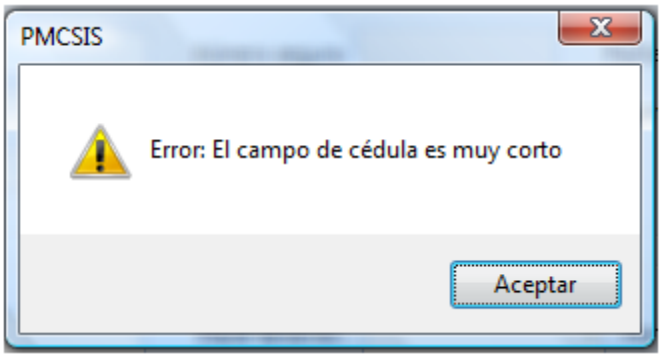


- En el campo cedula se digita el numero de cedula del estudiante (este campo no puede ser nulo, ni tampoco puede menos de 10 caracteres para los que no son de Nicaragua, para Nicaragua el campo es de 14 caracteres).

Mensaje de error campo nulo.

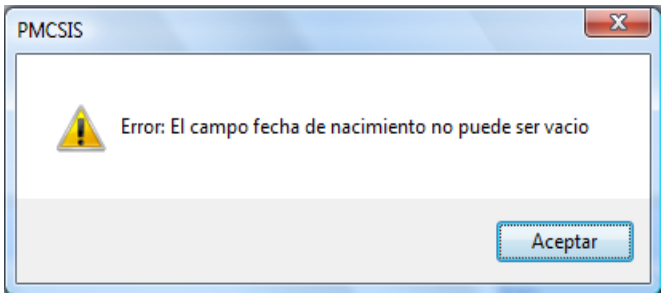


Mensaje de aviso



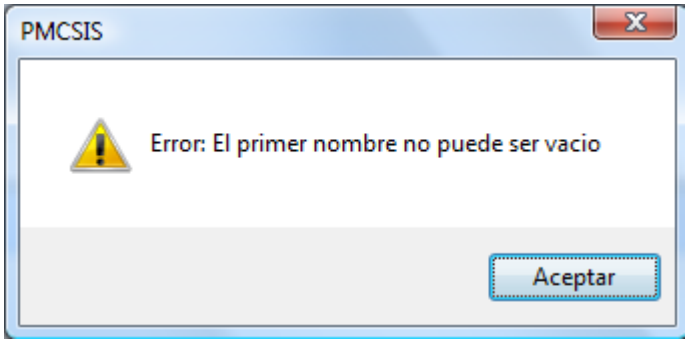
- Luego en el campo número de seguro se digita el seguro del estudiante (Este campo puede ser vacio).
- El campo fecha_nac para los que no son de Nicaragua es requerido que el digitador ingrese su fecha de nacimiento, en cambio si es de Nicaragua la fecha de nacimiento la toma de el numero de cedula por tanto no la digitara).

Mensaje de Error si el estudiante es de otra nacionalidad en el campo fecha

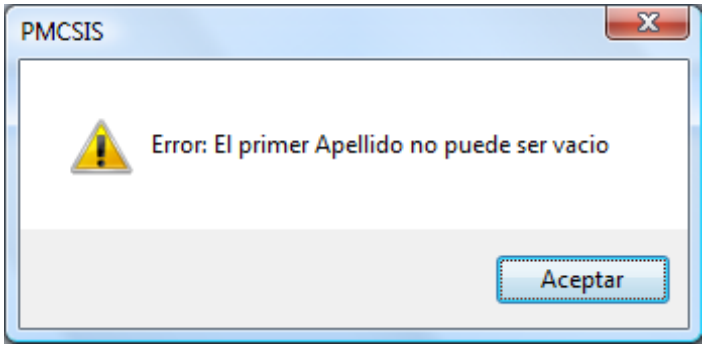


- Luego los campos de Primer nombre, Segundo nombre, Primer Apellido, Segundo apellido (Los campos requeridos son Primer nombre y Primer Apellido, los otros dos pueden ser nulo).

Mensaje de Error del Primer Nombre

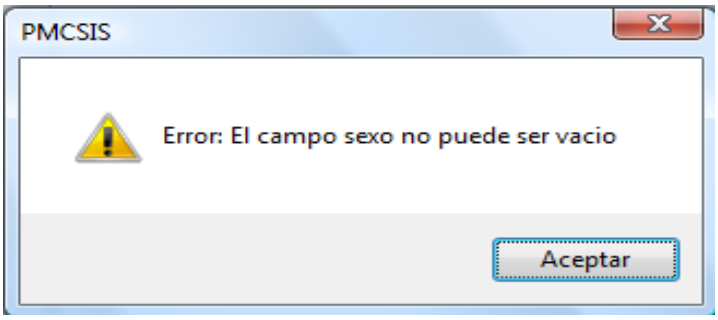


Mensaje de Error del Primer Apellido



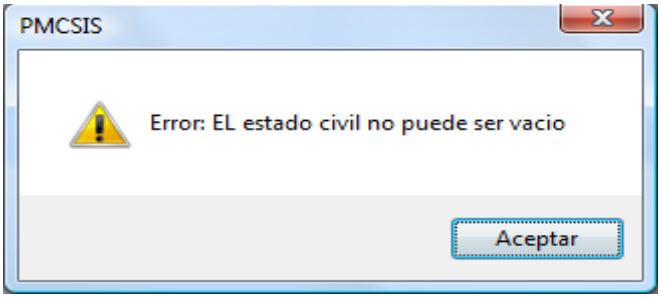
- El campo sexo es requerido , de él depende que podamos seguir introduciendo la información del estudiante, de lo contrario nos manda el siguiente mensaje:

Mensaje de Error



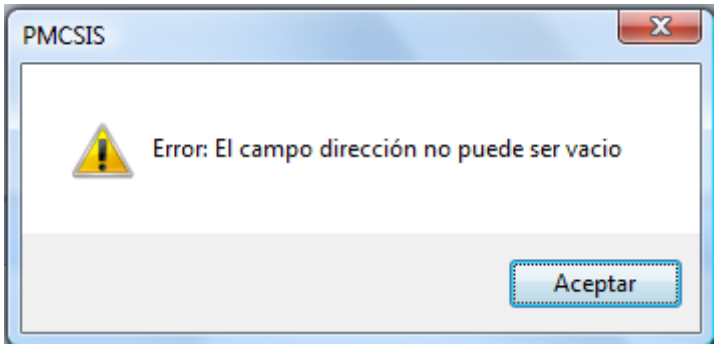
- El campo estado civil es requerido de él depende que podamos seguir introduciendo la información del estudiante , de lo contrario nos manda el siguiente mensaje:

Mensaje de Error

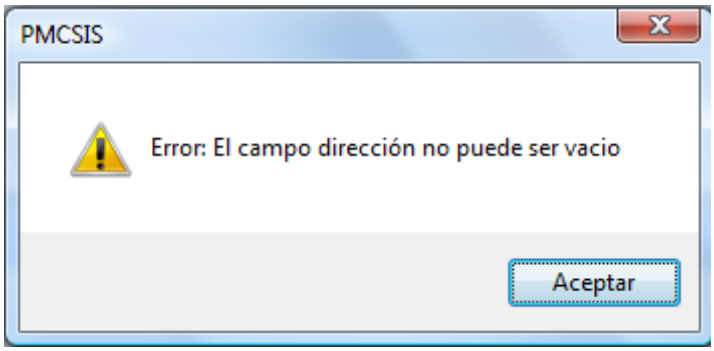


- El campo de dirección es requerido, al igual que Departamento y Municipio para poder seguir con la parte de estudios realizados. Mensajes de Errores.

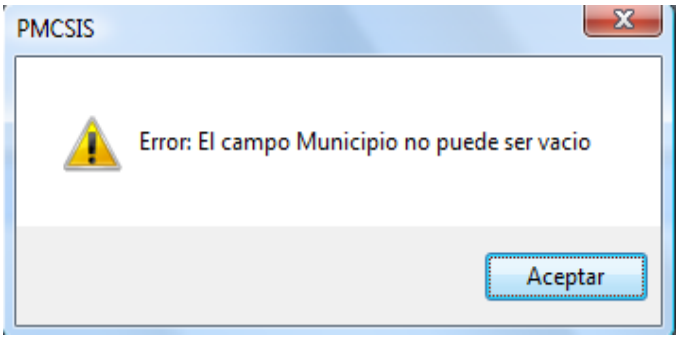
Mensaje de Error de Dirección



Mensaje de Error de Departamento

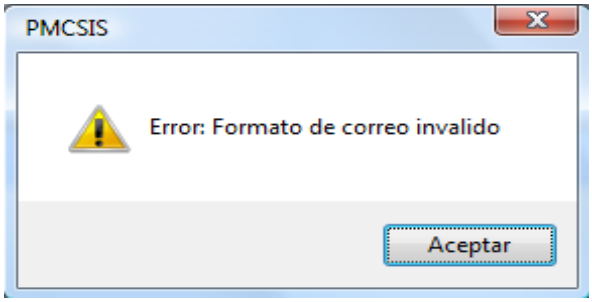


Mensaje de Error de Municipio



- El campo correo no es requerido ,pero tiene que tener la siguiente estructura por ejemplo : maestria@yahoo.es de lo contrario nos manda un error:

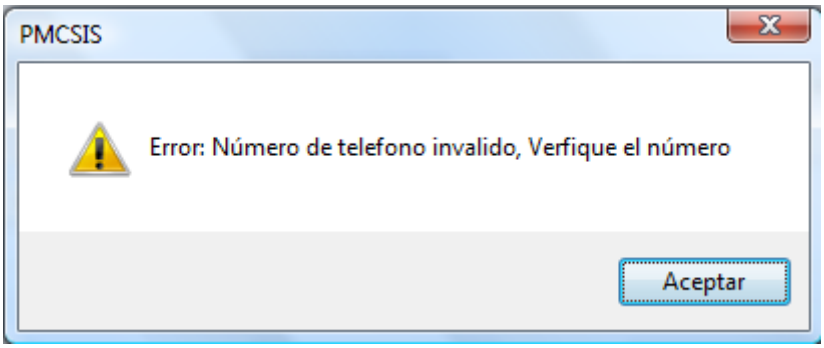
Mensaje de Error de Correo



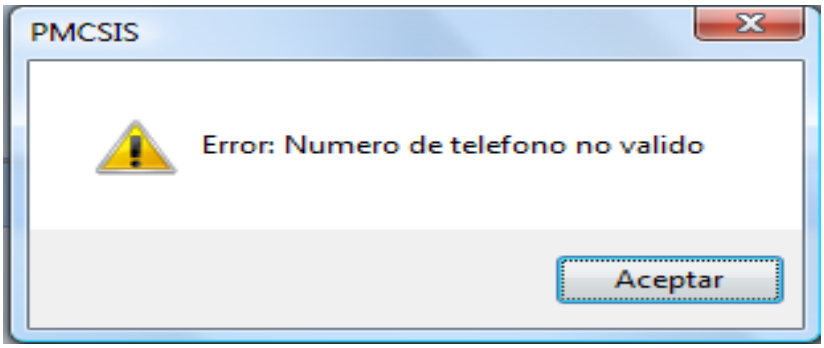
- El campo de número de teléfono al igual que el de celular no es requerido, pero si tiene un formato que es de 8 caracteres y no debe empezar con un número menor que 3 si no se escribe así nos mandara el siguiente mensaje:

Mensaje de Error de número de Teléfono

Ejemplo: 1111-2561

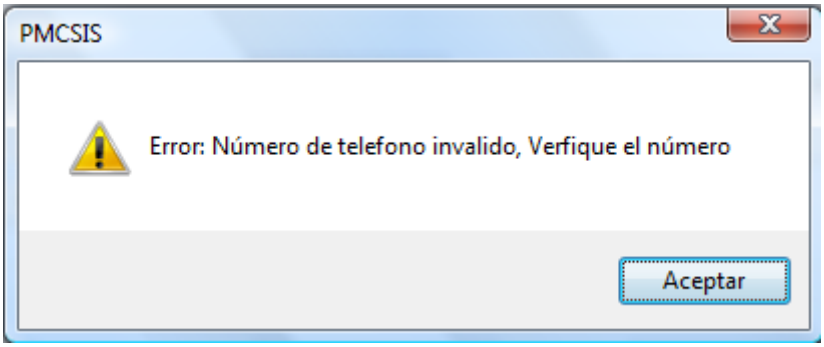


Ejemplo: 6258-4

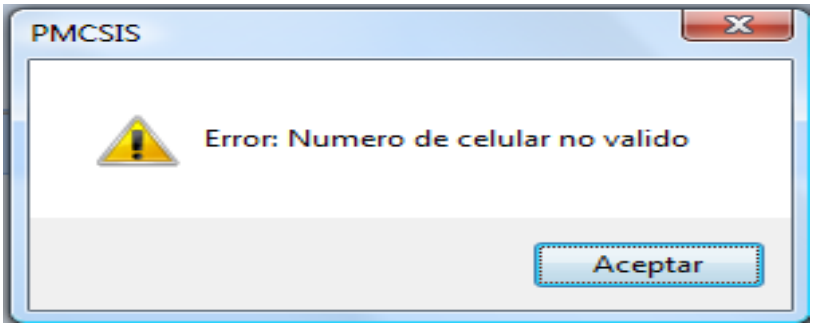


Mensaje de Error de número de Celular

Ejemplo: 1111-2561



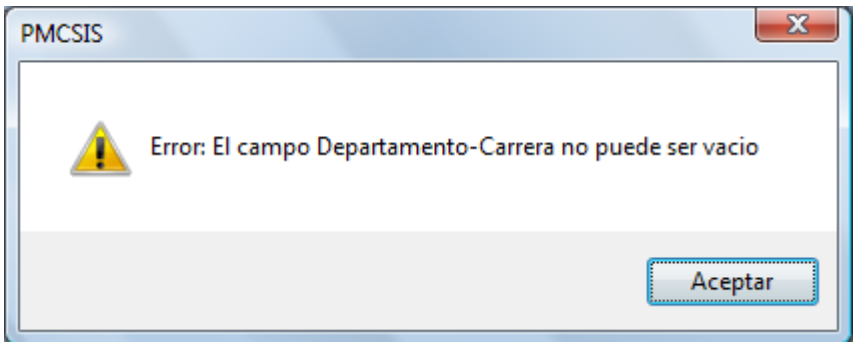
Ejemplo: 6258-4



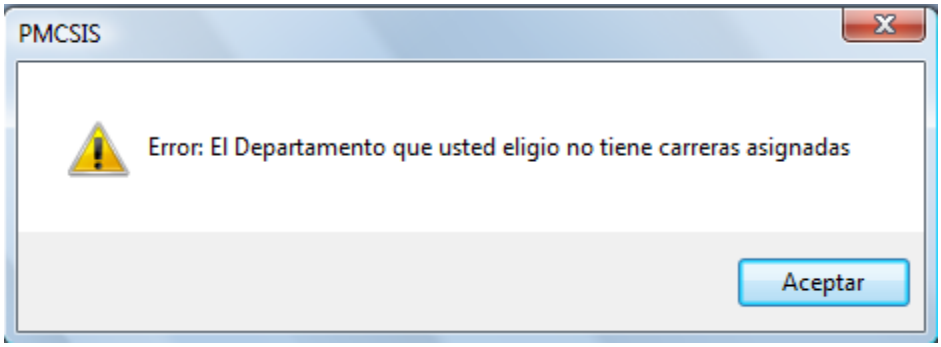
Ahora viene la parte de Estudios Realizados que activa una vez llenado los datos generales.

- El campo Departamento no puede ser nulo y debe de tener carreras incluidas ya que si no tiene manda mensaje de error.

Mensaje de Error (Cuando esta nulo)

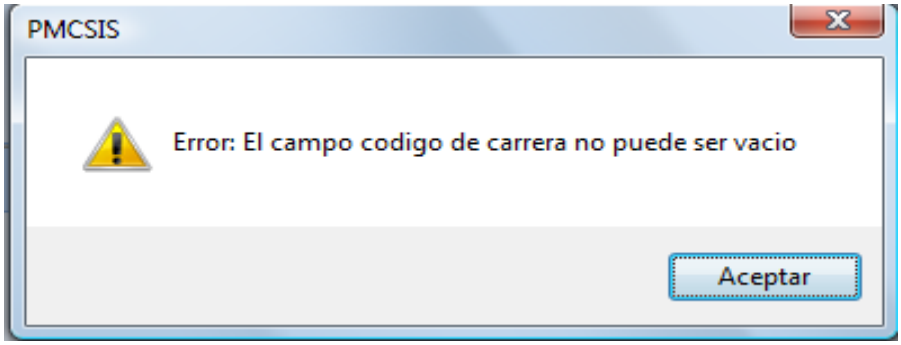


Mensaje de Error (Cuando no tiene carreras incluidas)



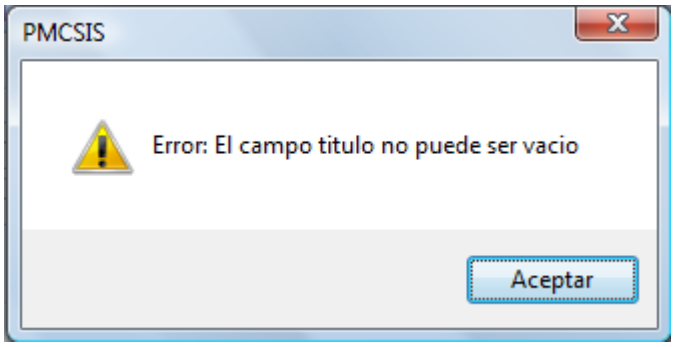
- Estando en el campo carrera este no puede ser nulo , al no ingresar información no manda el siguiente mensaje:

Mensaje de Error



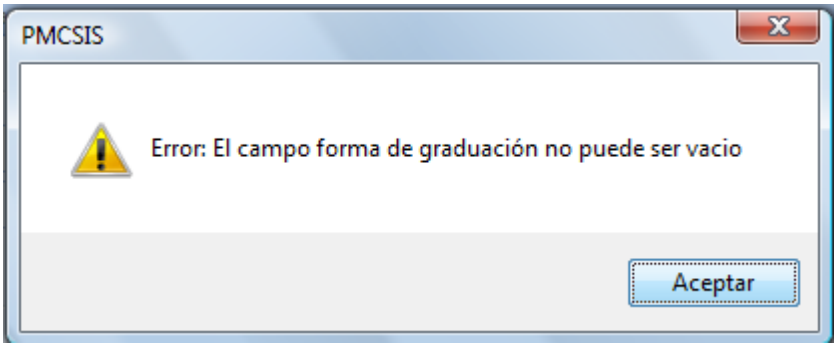
- El campo titulo no puede ser nulo se selecciona (licenciado, Ingeniero), siendo nulo o vacio nos mandaría el siguiente mensaje:

Mensaje de Error



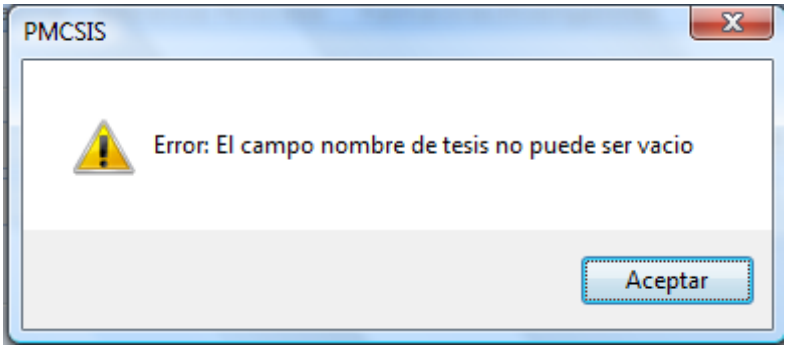
- El campo forma de graduación se escoge la manera en que se graduó (Monografía, tesis, seminario, examen de grado) no puede ser nulo.

Mensaje de Error



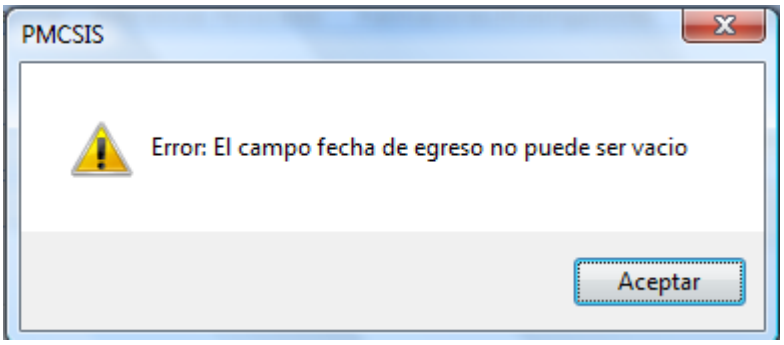
- El campo nombre de tesis es requerido por tanto no puede ser nulo en este caso no refleja el siguiente mensaje:

Mensaje de Error

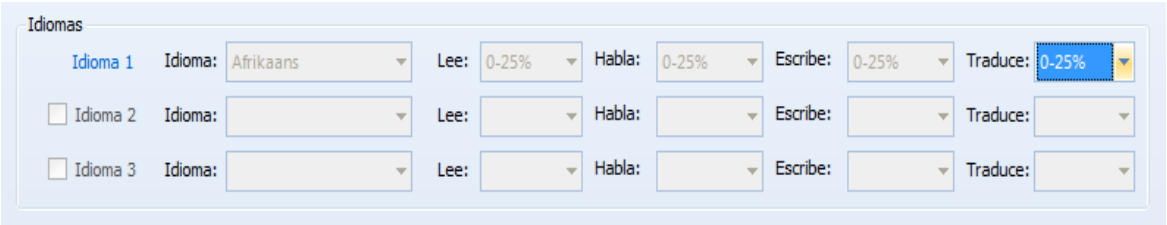


- El campo fecha de egresado no puede ser nulo debe de seleccionar del calendario.

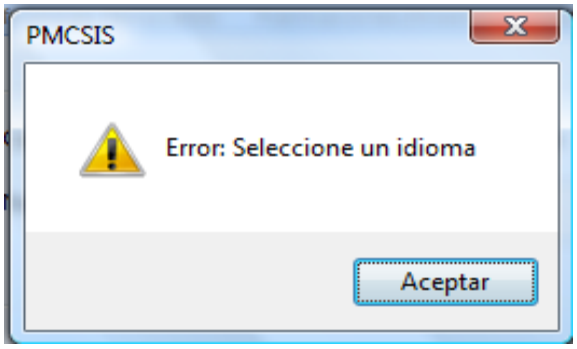
Mensaje de Error

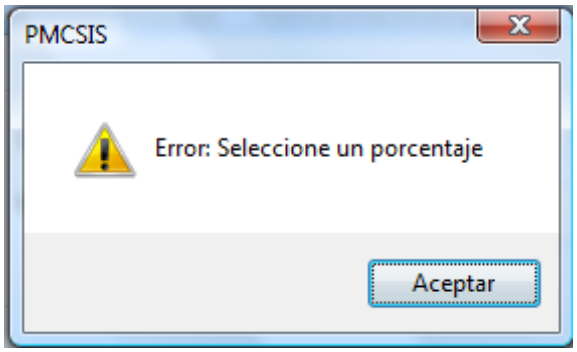


- En la sección de idiomas aquí se escoge como máximo tres idiomas y el porcentaje de habla, lee y escribe, el primer idioma es requerido, los otros dos idiomas si no tiene, entonces los omite dándole check al segundo idioma.

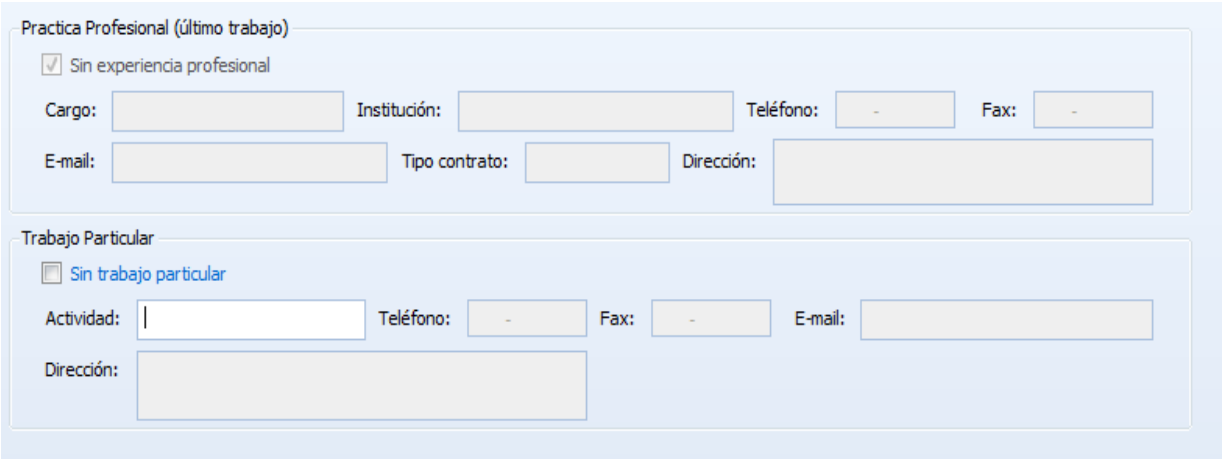


Mensaje de Error



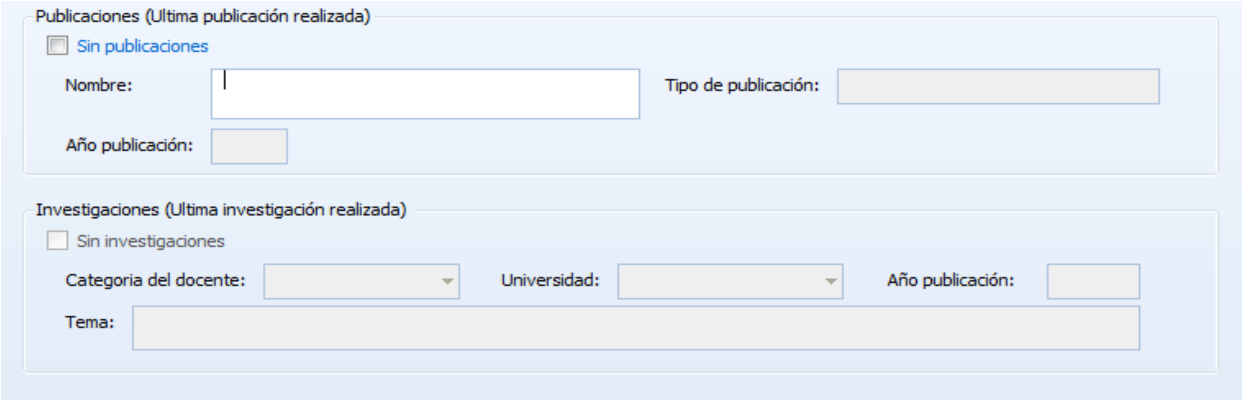


- En la parte de experiencia laboral los campos que se encuentran pueden quedar vacios dándole check no son requeridos.



- En lo que respecta a las referencias personales aquí como en los casos anteriores Nombre y primer apellido es requerido al igual que profesión, dirección, teléfono y el único no requerido es el email que tiene la misma estructura explicada anteriormente.

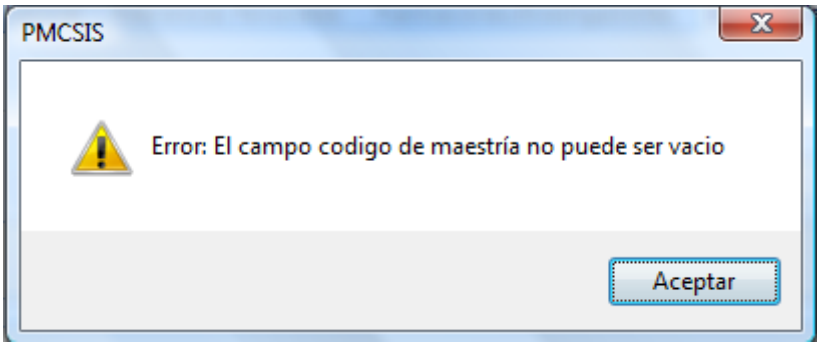
- En las publicaciones e investigaciones estas no son requeridas, por tanto pueden quedar vacías dándole check en los checkbox , en caso contrario que tenga todos los campos son requeridos.



En los datos de la maestría:

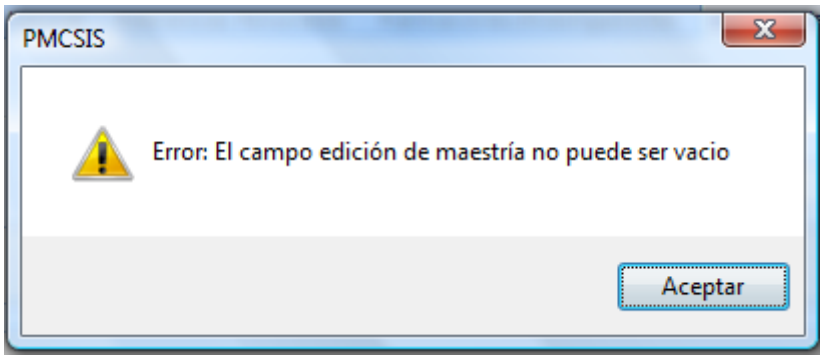
- En el combo maestría se selecciona el código de maestría al que va pertenecer el estudiante de lo contrario si no se selecciona no manda el siguiente mensaje:

Mensaje de Error



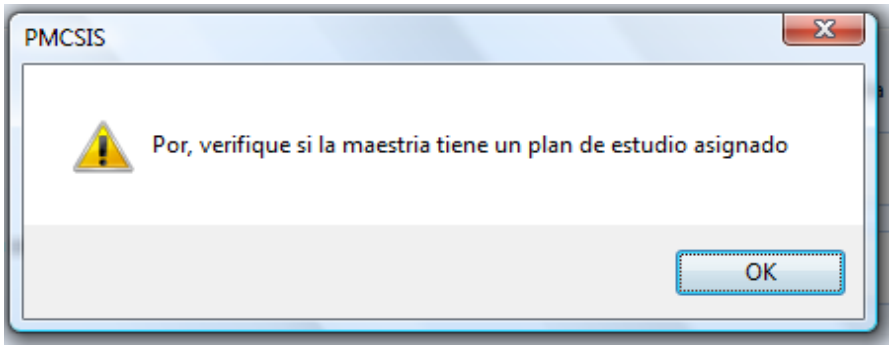
- El combo edición seleccione de el número de edición de la maestría de lo contrario nos manda el siguiente mensaje :

  Mensaje de Error



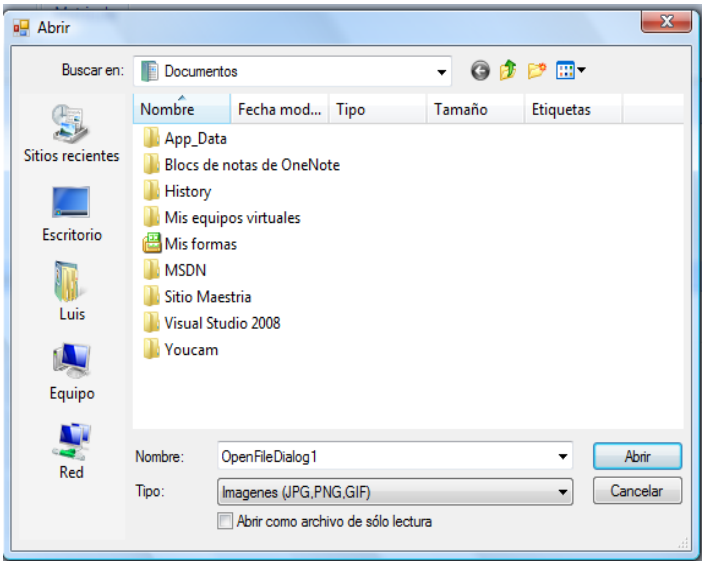
- El combo plan de estudio se selecciona el plan correspondiente a la maestría a la que va pertenecer el estudiante, de lo contrario manda un mensaje:

  Mensaje de Error

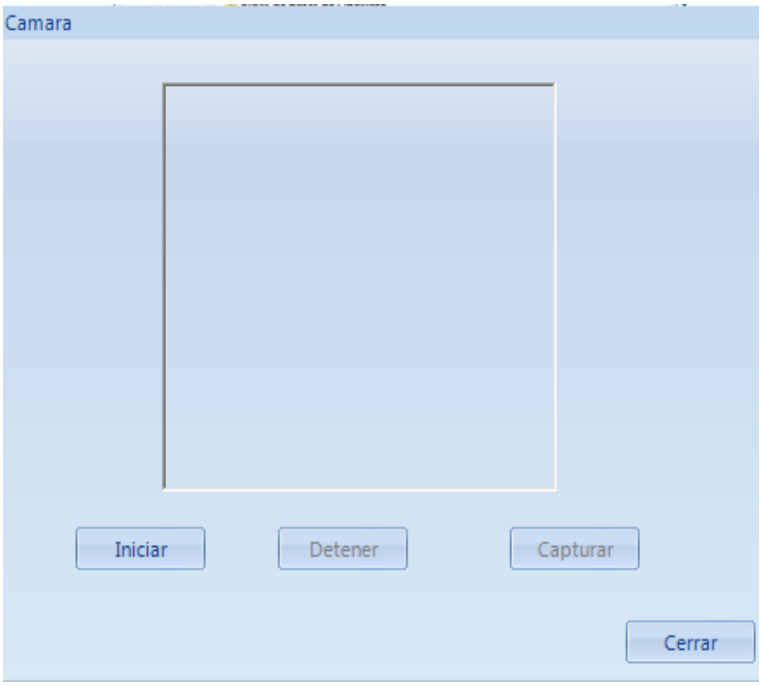


- El campo fecha de ingreso la da de acuerdo con la el código de maestría y la edición, este campo no lo digita el digitador.
- El campo de motivación e información adicional no son requeridos
- Con respecto al medio de comunicación por el cual se entero debe de seleccionar uno
- Con la fotografía del estudiante hay dos formas de ingresarla 1- Vía archivo y 2- que la maquina tenga cámara y de ahí se la tomen.

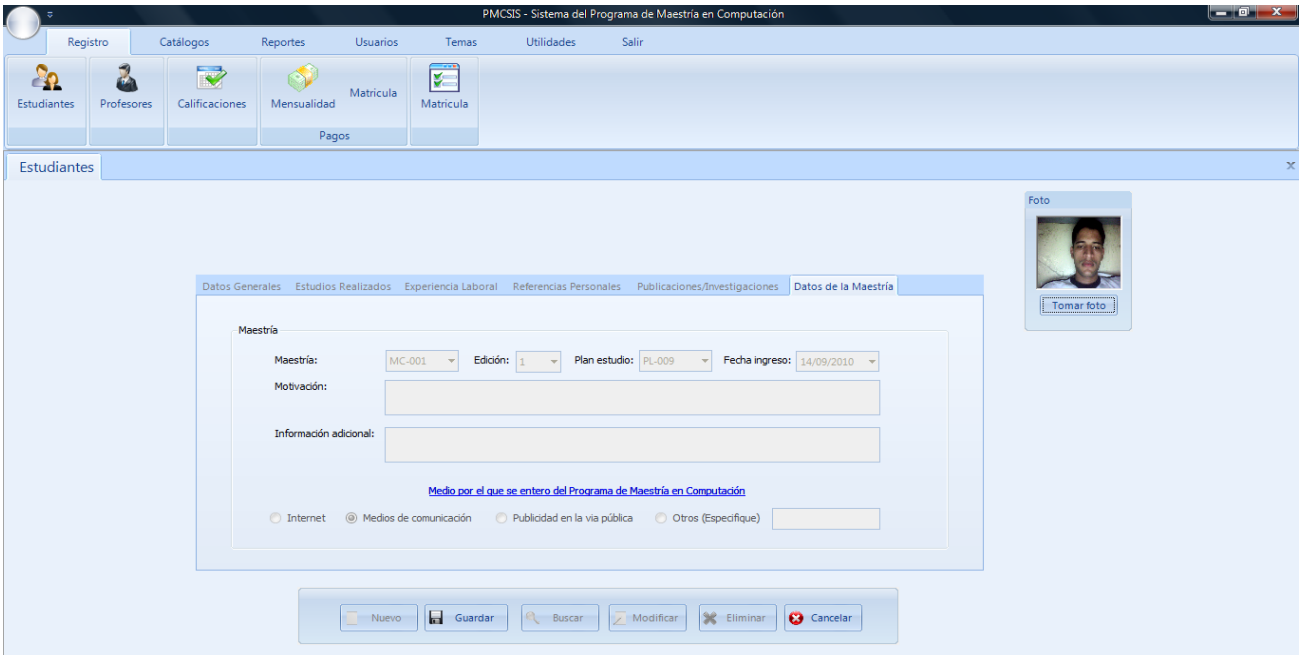
  1- **Vía archivo:** Doble clic en la caja nos aparece la caja de dialogo donde escogemos la imagen ser guardada (tipo de archivo JPG, PNG, GIF).



2- **Vía Cámara** : dándole clic en el botón Tomar Foto



Clic en Iniciar se activa la cámara y luego en capturar imagen



Luego se activa el botón 

Guardar

Resultado 12 – Tabla de métricas evaluadas en el sistema

La siguiente tabla resume las métricas aplicadas al sistema PMCSIS para la medición del producto software, y sus respectivos valores. Cabe recalcar que estos valores son aproximados.

Métrica	Valor	Observación
Fan-in	0.8	La cantidad de métodos o funciones que llaman a otros métodos o funciones es aproximadamente del 80%.
Fan-out	0.3	La cantidad de métodos subordinados o métodos que son llamados por otros métodos o funciones es aproximadamente 25%
Longitud de los identificadores	0.5	La longitud de los identificadores es moderada para la comprensión de estos.
Métodos pesados por clase	0.3	El peso de los métodos viene dado por la complejidad de los métodos, en particular la complejidad de los métodos pesados por clases es moderada.
Complejidad ciclomática	0.5	Esta medida está relacionada con la comprensión del programa, el grado de comprensión del programa es intermedio.

Tabla 6– Métricas aplicadas al sistema PMCSIS

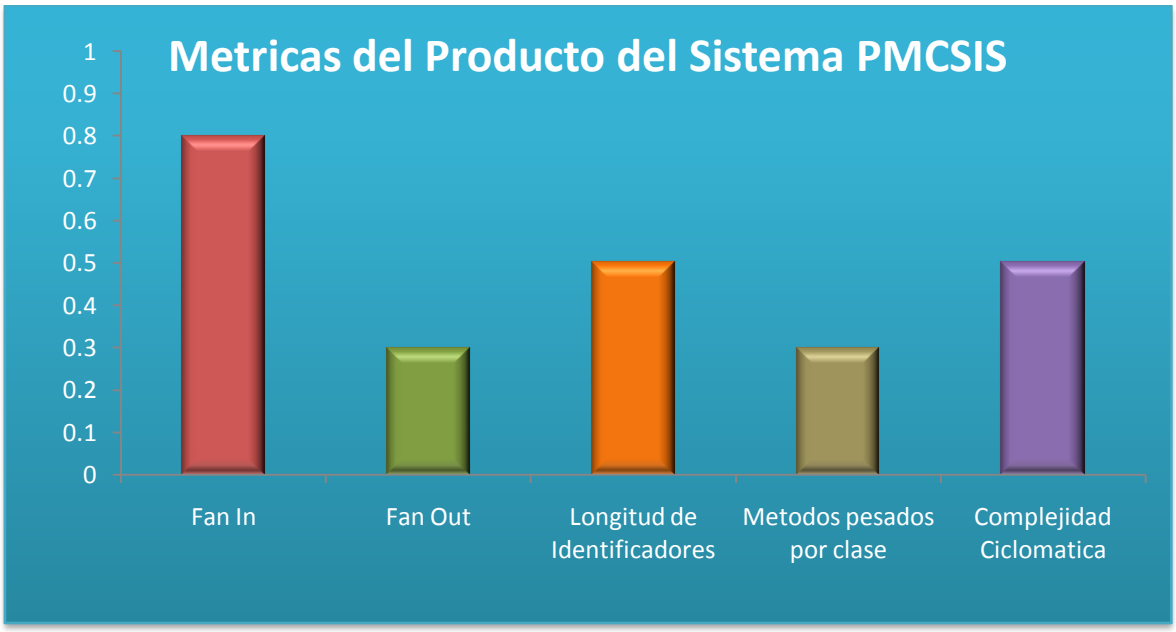


Figura 24– Diagrama de métricas aplicadas a PMCSIS

Métrica	Valor	Observación
Cantidad de gráficos	0.4	El Sitio Web no cuenta con muchas imágenes para que este pueda cargarse en el navegador de forma más rápida.
Porcentaje de texto en el cuerpo	0.1	El sitio Web cuenta con pocas líneas de texto en el cuerpo de las páginas.
Tamaño de la pagina	0.3	El peso de las páginas Web es moderado.
Cantidad de colores	0.2	La cantidad de colores utilizados es mínima.

Tabla 7– Métricas aplicadas al Sitio Web

Figura 25– Diagrama de métricas aplicadas al Sitio Web

VI. Diseño Metodológico

La finalidad de este apartado es establecer el nivel de profundidad que se busca mediante el conocimiento propuesto, así como la forma de acceder a la información referente al estudio.

Para cumplir con tal fin, el primer aspecto a definir fue el tipo y diseño de la investigación, seguidamente describir el procedimiento a seguir en la realización del trabajo.

6.1. Tipo de estudio

Este trabajo está orientado hacia una investigación Aplicada, cuya realización resolvió problemas concretos que demandaban solución.

Para su realización se tomaron como bases un conjunto de conocimientos teóricos y prácticos para la debida construcción del sistema por medio del modelo incremental.

A partir de esta investigación Aplicada se originó el proyecto factible, ya que está dirigido a la automatización del proceso de Registro Académico y financiero del programa de maestría del Departamento de Computación de la UNAN – Managua a objeto de mejorar la eficiencia de los procesos realizados en el departamento, y optimizar la calidad de servicio al estudiante.

6.2. Procedimientos

Para la elaboración del presente trabajo se utilizó uno de los modelos de desarrollo de software el cual es el **Modelo Incremental** que consta de 5 etapas que comprenden:

- ✓ Análisis
- ✓ Diseño
- ✓ Código
- ✓ Pruebas
- ✓ Producto (Implantación)

Así como también el proyecto se clasifica dentro de los tipos de sistemas de información en **MIS** (Sistemas de información Gerencial).

Incrementos

Como no es posible obtener desde un inicio todos los requerimientos, se desarrollaron varias versiones del software (prototipos), por tal motivo se realizaron incrementos en los que se tuvo que regresar de la etapa 4 a la etapa 1 iterativamente hasta obtener el producto completo. (**Ver fig. 26**)

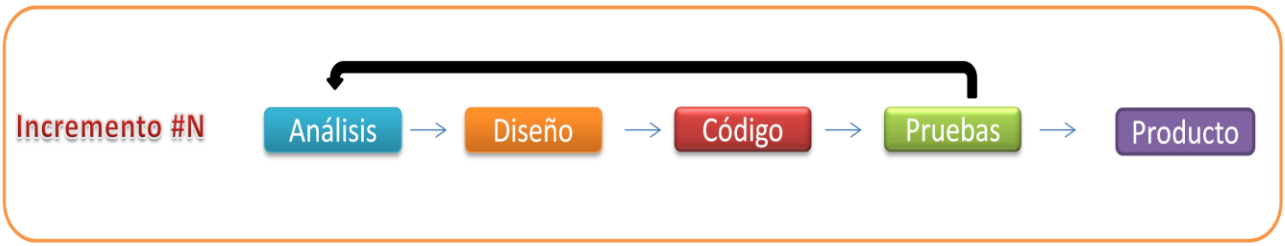


Figura 26– Etapas del Modelo Incremental.

En este gráfico de muestran los incrementos del Sistema PMCSIS (Sistema para el Programa de Maestría en Computación) que se realizaron **Ver fig. 27**

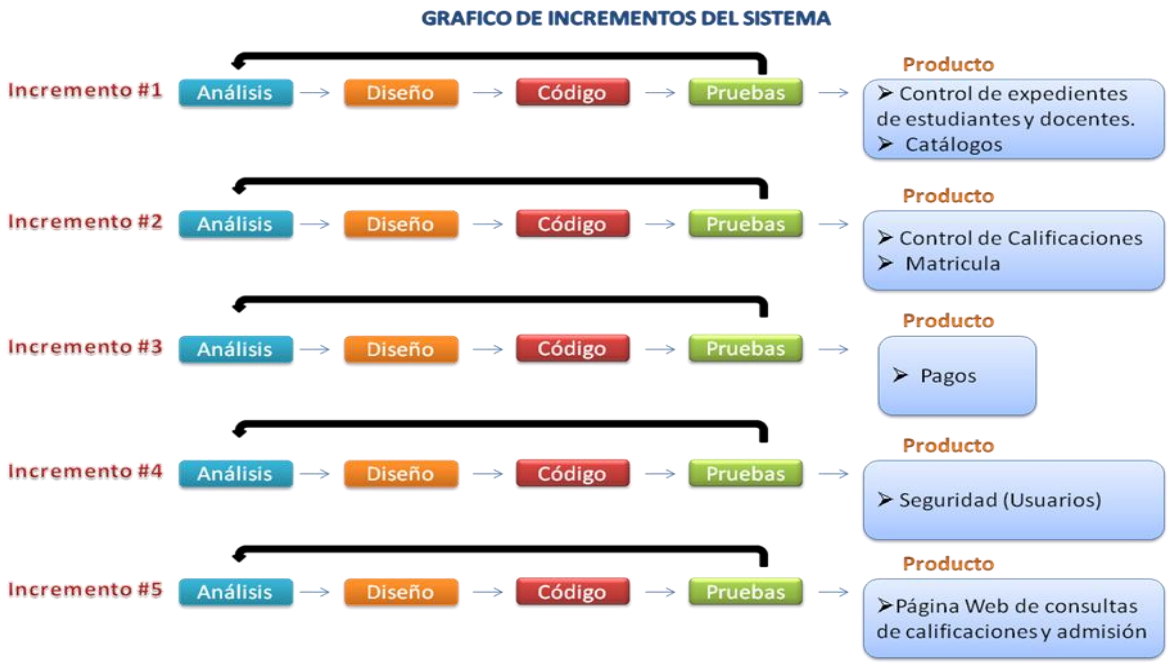


Figura 27– Incrementos del sistema PMCSIS

6.2.1. Etapa 1. Análisis

En esta etapa se definieron las funcionalidades que tienen los incrementos del sistema desarrollado. Como primer paso se realizaron entrevistas al Director del Departamento de Computación responsable del Programa de Maestría para obtener información y/o observar que se requería.

Para los elementos teóricos se consultó bibliografía de análisis y diseño de sistemas, ingeniería del Software en:

- La biblioteca central de la UNAN-Managua y en buscadores de Internet
- Documentos existentes en la biblioteca del departamento de computación (libros, monografías, etc.) y la observación.

Se realizó un estudio de factibilidad para determinar si el sistema era viable en la funcionalidad, en este estudio se tomaron en cuenta los requerimientos que tiene el sistema por incremento. Así también se procedió a realizar casos de usos por cada incremento, para tener una mejor visión de la forma en que se deben de manejarse los datos y los cambios que sufrirán para la generación de informes requeridos.

6.2.2. Etapa 2. Diseño

Ya definidos los requisitos (requerimientos) en la fase anterior por cada incremento mediante el estudio de factibilidad**,** estos se tomaron en cuenta para abordar al desarrollo del sistema en sí.

El desarrollo se inicio con la realización de un diseño lógico por cada incremento, el cual no incluía todas las funcionalidades, pero sí, las más importantes, como diagramas (casos de uso, actividades, entidad – relación, secuencias etc.), el diseño de los procedimientos de captura de datos, la base de datos y las salidas en pantalla o impresas hacia el usuario de acuerdo con las necesidades de información.

- ✓ Para la creación de diagramas se utilizó Microsoft Visio 2007, Rational Rose (UML. Normalización, entidad-relación, diseño de la bases de datos.)
- ✓ Para el diseño de las pantallas de entrada y salida se utilizó Visual Studio 2008 (Visual Basic)
- ✓ Para diseño de imágenes para la página y el sistema se utilizó Adobe Photoshop CS3 y Fireworks 8
- ✓ Para los reportes se utilizó Crystal Report v10
- ✓ Asp.net para la elaboración de la página web de consultas (calificaciones, pagos).

6.2.3. Etapa 3. Codificación

En esta penúltima etapa, se redujo toda la parte de diseño a código, con el objetivo de asegurarse de que funcione de acuerdo a las especificaciones y necesidades requeridas.

Para iniciar con la codificación se partió del conocimiento práctico referente a la programación orientada a objetos y en bases de datos para su debida implementación, el cual se utilizó el lenguaje Visual Basic de la plataforma de Visual Studio.NET 2008 y como gestor y administrador de bases de datos SQL Server 2008.

6.2.4. Etapa 4. Prueba

Durante esta etapa se realizaron implementaciones de cada incremento, aquí el usuario final por primera vez pudo interactuar con la aplicación (incremento) y probar así la ejecución de los procesos automatizados.

Esta etapa de pruebas se determinó si se logró cumplir con los objetivos definidos en la primera etapa. Se verificaron los problemas que puedan ocurrir en cada módulo (pruebas de unidad), continuando con las pruebas de integración de todos los incrementos, y para finalizar las pruebas de entrega así como las pruebas de aceptación.

Cabe recalcar que el desarrollo del modelo incremental implicó que se pudieran crear varias versiones preliminares de lo que es ahora el sistema en su totalidad.

6.2.5. Etapa 5. Producto

Una vez ya realizados todos los diferentes incrementos se hizo entrega del software completo al coordinador del programa de Maestría para su debida implementación; esto consistió en la instalación del sistema (software) en las máquinas donde se va a trabajar.

Una vez terminada la instalación del software, el mismo paso a la fase de producción (operatividad), durante la cual cumplió las funciones para las que fue desarrollado, es decir, es finalmente fue utilizado por el (o los) usuario final, produciendo los resultados esperados. Entre las pruebas que se aplicaron están:

- Pruebas de unidad: Se verificó que el flujo de los datos sea el correcto**.(ver Resultado 9)**
- Prueba de integración: Aquí se verificó si se encontraron defectos al momento de integrar cada incremento. **(ver Resultado 10)**
- Prueba entrega: Aquí se demostró la funcionalidad especificada, rendimiento y confiabilidad. **(ver Resultado 11)**

Aplicación de Métricas

Posteriormente de haber culminado con las etapas anteriores y haber realizados todos los incrementos teniendo el producto final, se procedió a aplicar métricas de software que llevaron a medir la calidad del producto.

Para ello se aplicó métricas de software basados en una escala que proyectó la idea del grado de medición ya sea menor o mayor que puede tomar una medida particular (**Ver fig. 28**).

Figura 28– Escala de Medición de Métricas.

Las métricas aplicadas al sistema PMCSIS y al Sitio Web de Consultas de Notas y Pagos, así como sus respectivos valores están resumidas en el **Resultado 12.**

Finalmente se puede observar en el siguiente gráfico un resume todas las fases anteriormente explicadas, las distintas fases que se llevaron a cabo en el desarrollo del proyecto (**Ver fig. 29**).

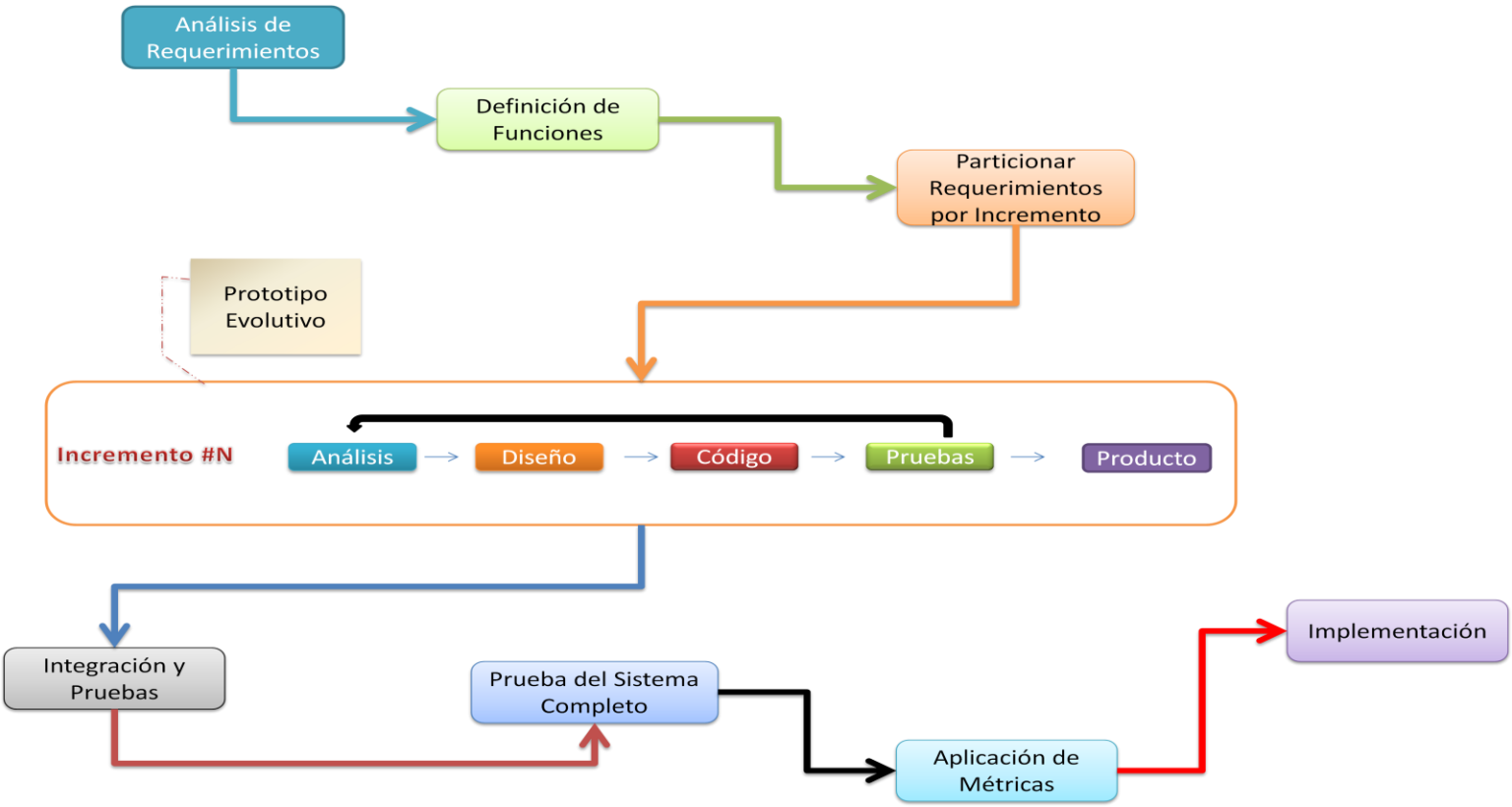


Figura 29– Fases que se desarrollarán en el proyecto.

6.3. Estudio de Factibilidad

Al iniciar con el análisis y diseño del sistema en cuestión se enfoco en el aspecto técnico (equipo a utilizar), económico (referente a los gastos de equipo, licencias de software, pago de realizadores del sistema entre otros) y operacional.

6.3.1. Factibilidad Técnica

CANTIDAD	RECURSOS	CARACTERISTICAS

Actualmente el programa de Maestría de la Facultad de Ciencias e Ingenierías no cuenta con ningún sistema de información.

El Departamento no cuenta con ninguna computadora destinada para la implantación del sistema, pero ha aceptado adquirir la tecnología necesaria para la implantación del sistema. En la (**Tabla 9**) se presentan los requerimientos del sistema.

6.3.2. Factibilidad Económica

Se ha estimado el costo mínimo necesario para el desarrollo del sistema:

DESCRIPCIÓN	CANTIDAD	COSTO UNITARIO $	SUBTOTAL $
Servidores	1	2,493.20	2,493.20
Equipo para programadores	2	580.75	1,161.50
Equipo para usuarios del sistema	1	580.75	580.75
Licencia de Visual Studio 2008	2	1,068.88	2,137.76
Licencia de SQL Server 2008	1	8,592.00	8,592.00
Papelería y otros	-	-	500
Muebles de Oficina (Sillas, Mesas)	4 (2 sillas, 2mesas)	87.46	174.92
Impresora	1	50.6	50.6
Pago de los desarrolladores	2	1500	3,000
MONTOS TOTALES			**$** !Especificación de carácter no válida

Tabla 8 – Recursos de la Factibilidad Técnica

1	Servidor de Bases de Datos	Capacidad de almacenamiento mínimo 500GB, preferiblemente en dos discos.
		Capacidad de memoria 6GB (Expandible)
		Procesadores Intel Xeon velocidad mínima de 2.0 GHz
		Otras características especificadas en cotizaciones del producto
2	Equipos de cómputo para desarrollo del sistema.	Capacidad de almacenamiento de 320GB
		Capacidad de Memoria 2GB (Expandible)
		Procesador doble núcleo, Preferiblemente marca Intel
		Monitores LCD de 17´´
		Teclado USB
		Mouse Óptico USB
1	Equipos de cómputo para usuario con acceso al sistema.	Capacidad de almacenamiento de 320GB
		Capacidad de Memoria 2GB (Expandible)
		Procesador doble núcleo, Preferiblemente marca Intel
		Monitores LCD de 17´´
		Teclado USB
		Mouse Óptico USB
1	Sistema Operativo	Microsoft Windows Vista
2	Plataforma .NET (Lenguaje de Programación)	Microsoft Visual Studio 2008 (Visual Basic)
1	Gestor Manejador de Bases de Datos	Microsoft SQL Server 2008
2	Mueble para computadora	
1	Impresora Tipo DeskJet	
2	Silla sencilla giratoria	

Tabla 9– Resumen de los Costos del Sistema

6.3.3. Factibilidad Operacional

Cuando el sistema se implemente, se espera a que simplifique:

- Los tiempos de búsqueda y respuesta.
- Elaboración de reportes entre otros

Claro que para que esto se cumpla será necesario tener los recursos que se plantean en el estudio de factibilidad para que pueda dar repuestas positivas, además de la disponibilidad de los usuarios finales a los cuales se les dará una previa capacitación del uso del sistema PMCSIS.

6.4. Descripción del sistema

El sistema de información implementado es el “PMCSIS” para el departamento de computación de la Facultad de Ciencias de la UNAN – Managua, que permite realizar los procesos de matricula (inscripción y/o retiro de materias), manejo de expedientes académicos, control de pagos, grabación de notas y un modulo de consulta de calificaciones y pagos en línea; este una vez instalado servirá de apoyo importante a la administración que actualmente está a cargo de la maestría.

El sistema genera información actualizada, inmediata a manera de reportes impresos y presentaciones en pantallas de: estudiantes matriculados, pagos realizados, estadísticas académicas y de pagos, docentes, certificado de calificaciones (tanto en el sistema como en línea a través del sitio web de consultas) y actualización de datos (estudiantes, docentes); además lleva un control de usuario que lo manejaran. El sistema tiene como opciones:

1. **Registro**
   1.1. Estudiantes
   1.2. Profesores
   1.3. Calificaciones
   1.4. Pago Matricula
   1.5. Pago Mensualidad
   1.6. Matricula Materias
2. **Catálogos**
   2.1. Universidad
   2.2. Facultad
   2.3. Carrera
   2.4. Maestría
   2.5. Materias
   2.6. Plan de Estudio
   2.7. Materias del Plan de Estudio
   2.8. Docentes materias
   2.9. Categoría Docente
   2.10. Departamento Carrera
   2.11. Grupo
   2.12. Departamento
   2.13. Municipio

3. **Reportes**
   3.1. Docentes
   3.2. Pagos
   3.3. Maestría
   3.4. Calificaciones
   3.5. Estudiantes
4. **Usuarios**
   4.1. Administrar Usuario
5. **Temas (Apariencia)**
6. **Utilidades**
   6.1. Actualizar Porcentaje de Becado
   6.2. Cambiar Montos de Maestría
   6.3. Cambiar Valor del Dólar
   6.4. Actualizar Solvencia de Pagos
7. **Salir**
   7.1. Cerrar Sesión
   7.2. Salir
8. **Ayuda**
   8.1. Manual de Usuario
   8.2. Acerca de PMCSIS

Descripción de cada opción

1. Registro

Permite ingresar la información del estudiante como: expediente, matricula de asignaturas (retiro y/o inscripción) calificaciones obtenidas por modulo, pagos de matrícula y mensualidades: así como la información de cada Docente de la maestría.

2. Catálogos

Es importante porque permite realizar ingresos, actualizaciones, eliminaciones de todas las tablas de la base de datos, este se encuentra restringido para los usuarios normales, solo el administrador puede realizar cambios.

3. Reportes

En estos se pueden encontrar reportes estadísticos y no estadísticos

.Estadísticos: se refiere a reportes que realicen cálculos como: promedios, porcentaje de becados entre otros.

No estadísticos: son informes comunes que no necesitan realizar cálculos como: listado de estudiantes, docentes etc.

4. Usuarios

En esta sección se puede agregar, modificar la información de los usuarios que usaran el sistema.

5. Temas (Apariencia)

En esta parte el usuario podrá cambiar la apariencia de la interfaz del sistema según sea su gusto (Azul, Negro, Plata).

6. Utilidades

Permite modificar los datos de algunas tablas como son cambio de valor del dólar, porcentaje de becados entre otros.

7. Salir

Permite dos opciones que son cerrar sesión que este nos sacara del sistema dándonos la opción de ingresar con otro usuario y la otra es salir que nos saca completamente del sistema.

8. Ayuda

Presenta el manual de usuario que sirve de guía para utilizar correctamente el sistema; así como una pantalla de *Acerca de* que muestra la información de la versión y programadores del sistema.

6.5. Análisis del sistema (PMCSIS)

Diccionario de elementos de Datos

Es un manual que contiene las definiciones y organización de los objetos utilizados y producidos por el sistema PMCSIS (**Ver Inciso 4.13, Anexo 1**).

Base de Datos Normalizada

Aquí se muestran los elementos del sistema analizados; esta se encuentra normalizada hasta la Tercera forma Normal (3FN), junto con sus respectivas relaciones entre las diferentes entidades que forma la BD (**Ver inciso 4.7.2, Resultado 1**).

Diagrama de clases

Herramienta grafica del Lenguaje Unificado Modelado (UML) que muestra los atributos y los métodos de cada tabla del Sistema (**Ver Inciso 4.8.5, Resultado 2**).

Diagrama de casos de usos

Muestran una secuencia de interacciones que se desarrollan entre el sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el sistema; también especifican la comunicación y el comportamiento de el sistema mediante su interacción con los usuarios y/u otros sistemas (**Ver Inciso 4.8.5, Resultado 3**).

Diagrama de secuencia

Muestran la interacción del conjunto de objetos del sistema y contienen detalles de implementación del escenario, incluyendo los objetos, clases y mensajes intercambiados entre los objetos. (**Ver Inciso 4.8.5, Resultado 3**).

Diagrama de colaboración

En estos diagramas se destaca la organización estructural de objetos que envían y reciben mensajes. (**Ver Inciso 4.8.5, Resultado 4)**

Diseño de pantallas del sistema PMCSIS (Ver Resultado 6)

Se presenta la forma en que el usuario final va a interactuar con el contenido del sistema dentro de ellas están:

1. **Pantalla principal**
2. **Académicos**

- Registro de Expedientes del Estudiante
- Registro de Calificaciones del Estudiante
- Matricula de Asignaturas
- Inscripción de Asignaturas
- Retiro de Asignaturas
- Registro de Docentes

3. **Catálogos**
4. **Pagos**

- Registro de Pago de Matricula
- Registro de Pago de Mensualidad

5. **Usuarios**

- Registro de usuario

6. **Utilidades**

- Solvencia de Estudiantes
- Cambio de Dólar
- Actualizar Montos de Maestría
- Porcentaje de Becados
- Actualizar cantidad de materias de matricula

7. **Manual de Ayuda**
8. **Acerca de**

Reportes (Ver Resultado 7)

Se presenta de forma impresa o visual dentro de ellos están los estadísticos y no estadísticos estos a su vez ayudan a la toma de decisiones.

No estadísticos

- Matricula de asignatura
- Retiro de Asignatura
- Recibo de Pago de Matricula
- Recibo de Pago de Mensualidad
- Listado de Estudiantes por Maestría
- Listado de Docentes
- Pagos por Estudiantes
- Historial Académico
- Plan de Estudio

Estadísticos

- Porcentaje de becados
- Porcentaje de Retención
- Porcentaje del Medio de Información
- Porcentaje de estudiantes por Sexo

Aplicación de Métricas

Medio que se utilizo para valorar, entender y probar el desarrollo del sistema y su mantenimiento **(Ver inciso 4.12)**

En la tabla de métricas vemos reflejado los valores que tiene nuestro sistema basándose una escala **(ver figura 28)**

Si desea ver los resultados al aplicar las métricas al sistema puede consultar el **Resultado 12.**

Diseño de pantallas del Sitio web de Consultas Calificaciones y pagos (Ver Resultado 8)

Se presenta las páginas que el estudiante va a interactuar en el sitio web de consultas de calificaciones y pagos dentro de ellas están:

- Bienvenida
- Inicio de Sesión
- Cambiar Contraseña
- Selección de Reportes
- Reporte de pago
- Reporte de Calificaciones

Pruebas del sistema

Consistió en comprobar que el software realizar correctamente las tareas indicadas en la especificación del caso, dentro de las pruebas se realizaron:

- Prueba unidad (**Ver Resultado 9**)
- Prueba de integración (**Ver Resultado 10**)
- Prueba de entrega (**Ver Resultado 11**)

VII. Conclusiones

Una vez desarrollado el sistema se puede concluir que este brinda una mayor seguridad y agilidad en el registro académico, lo cual conlleva a mejorar la atención de cada uno de los estudiantes de la maestría en sistemas de información. Este permite almacenar grandes cantidades de información para que pueda ser accesada en el momento que se necesite.

.

El sistema permitirá llevar un seguimiento constante de la evolución de cada estudiante en lo que respecta a las calificaciones.

Por otro lado permite emplear un mecanismo de pagos eficiente llevando el registro de estos, el monto cancelado en un determinado mes y a su vez verifica si el estudiante es becado, por tanto le refleja el monto total a pagar e imprime su respectivo recibo, toda esta información se va almacenando para su respectivo uso cuando fuese necesario (modificar, eliminar y/o buscar un pago correspondiente); además lleva el control del pago de matrícula de cada estudiante.

En lo académico a partir de la generación de reportes; la administración de la maestría puede conocer la información de cada uno de los estudiantes como: rendimiento académico y en lo referido a la publicidad definir el medio de comunicación por el cual los interesados se enteraron de la oferta de la maestría en sistemas de información del departamento de computación de la UNAN - Managua; estos reportes reducen los posibles atrasos en que puedan incurrir para así tomar las decisiones más adecuadas**.**

En lo administrativo; el sistema resultó factible ya que permite tener más posibilidades de análisis de las situaciones que presentan los estudiantes como: becas, pagos atrasados, monto de la maestría (matricula y mensualidades).

El sitio web proporciona una mayor comunicación maestría – estudiante, ya que este ultimo puede accesar desde cualquier lugar donde se encuentre por medio de un login y contraseña que se será entregado por la administración, a su vez podrá cambiar su contraseña, consultar sus pagos y calificaciones; cabe recalcar que estas solo pueden ser accesadas si el estudiante se encuentra solvente con sus pagos.

Con la aplicación de métricas al sistema se evaluó la calidad logrando un producto estable en gran medida con los requerimientos planteados.

Para facilitar el acceso al sistema se creó un manual de usuario en donde se explican las funciones o procedimientos que se deben de realizar al momento de utilizarlo, este servirá de apoyo a los usuarios encargados del sistema.

VIII. Recomendaciones

- Para manejar el sistema se requiere de un profesional que cuente con conocimientos de computación, además este debe capacitado antes usarlo.

- Proveer de mecanismos avanzados de tolerancia a fallos al servidor que atienden el sistema (fuentes de poder redundantes y almacenamiento en disco), para asegurar el funcionamiento permanente de los mismos.

- Se sugiere que los servidores permanezcan físicamente en un área (Nodo de Computacion) que posea las características mínimas adecuadas para tal efecto, dentro de las características se recomienda los siguientes puntos:

    - El área física debe ser exclusivamente para contener el servidor y equipos de infraestructura tecnológica asociada al sistema.
    - El acceso físico al área (servidor) debe ser restringido.
    - El acceso a la base de datos debe de ser restringida
    - Se debe contar con uno o más circuitos eléctricos regulados, de la misma forma se sugieren la utilización de equipos de respaldo y regulación eléctrica en caso de fallas.

- Es importante que el manejo de usuarios y roles del sistema se realice de forma ordenada, y asegurar que los usuarios y contraseñas sean de uso estrictamente individual, esto para garantizar el correcto acceso a la información y evitar se vulneren los mecanismos de seguridad.

- Por ser un sistema desarrollado por el modelo incremental se pueden incluir nuevas opciones o actualizaciones en el caso de que este se emplee en nuevas maestría en el departamento o en otras áreas de la Facultad de Ciencias.

IX. Bibliografía

Consultada

1. [Alarcón, 2000]
   Alarcón, Raúl;
   "Diseño orientado a objetos con UML"
   Grupo Eidos, España (Madrid), S.L., 2000.

2. [Berzal, 2007]
   Berzal, Fernando; Cortijo, Francisco José; Cubero, Juan Carlos;
   "Desarrollo profesional de aplicaciones web con ASP.NET"
   Berzal Fernando, España, 2007.

3. [Boggs, 2002]
   Boggs, Wendy; Boggs, Michael;
   "Mastering UML with Rational Rose 2002"
   SIBEX Inc., México, 2002

4. [Ceballos, 2006]
   Ceballos, Francisco Javier;
   "Enciclopedia de Microsoft Visual Basic"
   Primera edición, Alfa omega, México, 2006.

5. [Fowler, 1999]
   Fowler, Martin; Scott, Kendall;
   "UML gota a gota"
   Primera edición, Addison Wesley Iberoamericana, México, 1999.

6. [González,1996]
   González, Carlos;
   "Sistemas de Bases de Datos"
   Editorial Tecnológica de Costa Rica, Costa Rica, 1996.

7. [Hernández, 2006]
   Hernández Sampieri, Roberto; Fernández Collado, Carlos;
   Baptista Lucio, Pilar;
   "Metodología de la investigación"
   Cuarta edición, McGraw Hill, México, 2006.

8. [Kendall (a), 1993]
E. Kendall, Kenneth; E. Kendall, Julie;
“Análisis y Diseño de Sistemas”
Tercera edición, McGraw Hill, México, 1993.

9. [Kendall (b), 2005]
E. Kendall, Kenneth; E. Kendall, Julie;
“Análisis y Diseño de Sistemas”
Sexta edición, McGraw Hill, México, 2005.

10. [Larman, 1999]
Larman, Craig;
“UML y PATRONES introducción al análisis y diseño orientado a objetos”
Primera edición, Prentice Hall, México, 1999

11. [Laudon, 2002]
C. Laudon, Kenneth; P. Laudon, Jane;
“Sistemas de información gerencial, organización y tecnología de la empresa conectada en red”
Sexta edición, Pearson educación, México, 2002.

12. [Mariño, 2005]
Mariño, Ramón;
“Diseño de páginas web y diseño grafico. Metodología y técnicas para la implementación de sitios web y para el diseño grafico”;
Primera edición, ideas propias editorial, España, 2005.

13. [Meigs, 2009]
Meigs Robert, Meigs Mary, Bettner Mark, Whittington Ray
“Contabilidad, La base para decisiones gerenciales”
Decima Edición, McGraw Hill, Colombia 2009

14. [Pressman (a), 1998]
Pressman, Roger;
“Ingeniería del Software: Un enfoque practico”
Cuarta edición, McGraw Hill, España, 1998.

15. [Pressman (b), 2006]
Pressman, Roger;
“Ingeniería del Software: Un enfoque practico”
Sexta edición, McGraw Hill, España, 2006.

16. [Schmuller, 1998]
Schmuller, Joseph;
“Aprendiendo UML en 24 horas”
Pearson educación Latinoamérica, México, 1998.

17. [Senn, 1992]
Senn, James A.;
“Análisis y Diseño de Sistemas de Información”
Segunda edición, McGraw Hill, México, 1992.

18. [Silberschatz, 2002]
Silberschatz, Abraham; F. Korth, Henry; S. Sudarshan;
“Fundamentos de bases de datos”
Cuarta edición, McGraw-Hill, España, 2002.

19. [Sommerville, 2005]
Sommerville, Ian;
“Ingeniería del software”
Sexta edición, Editorial Pearson Educación S.A., España (Madrid), 2005.

X. Web grafía

1. [Ivory, 2001] - Ivory, M., R. Sinha y M. Hearst, "Empirically Validated Web Page Design Metrics", ACM, SIGHI '01, Seattle, WA, abril de 2001, disponible en http://webtango.berkeley.edu/papers/chi2001/

XI. ANEXOS

ANEXOS

Anexo 1 – Diccionario de elementos de datos

Nº	NOMBRE	TIPO	LONGITUD	DESCRIPCION
1	act_laboral	varchar	40	Actividad laboral de una persona de referencia
2	Activar	int	4	Activa al estudiante para poder ver certificado de notas en el sitio web , si no está al día con los pagos
3	Active	bit	1	Activa al usuario cuando ingresa al sistema (1 - Activado , 2 - Desactivado)
4	Actividad	varchar	50	Tipo de actividad que realiza en el trabajo
5	año_lectivo	smallint	2	Año lectivo en que la materia se va impartir por un Docente
6	año_publicacion	smallint	2	Año de publicación de un tema
7	ape_usuario	varchar	50	Apellidos del usuario que maneja el sistema
8	ape1	varchar	12	Primer apellido de un docente
9	ape1_docente	varchar	12	Primer apellido del docente
10	ape1_ref	varchar	12	Primer apellido de una persona de referencia
11	ape2	varchar	12	Segundo apellido de un docente
12	ape2_docente	varchar	12	Segundo apellido del docente
13	ape2_ref	varchar	12	Segundo apellido de una persona de referencia
14	Apellidos	varchar	50	Apellidos del estudiante

Nº	NOMBRE	TIPO	LONGITUD	DESCRIPCION
15	cambio_a	nchar	8	Cambio de Dólar según tasa
16	Cargo	varchar	40	Cargo que tenia o tiene en el lugar de trabajo
17	Celular	varchar	9	Número de celular del estudiante
18	Clave	varchar	200	Contraseña del estudiante que maneja el sistema
19	cod_carrera	varchar	6	Código de identificación de una Carrera
20	cod_categoria	int	4	Código de identificación de una categoría de un estudiante
21	cod_depto	varchar	6	Código de identificación de un departamento académico
22	cod_docente	varchar	6	Código de identificación de un Docente
23	cod_dpto	varchar	2	Código de identificación de un departamento geográfico de Nicaragua
24	cod_facultad	varchar	6	Código que identifica a la facultad
25	cod_grupo	varchar	4	Código de identificación de un grupo
26	cod_grupo	varchar	4	Código que identifica a un grupo de estudiantes
27	cod_idioma	int	4	Código que identifica un determinado idioma
28	cod_invest	int	4	Código que identifica a una investigación
29	cod_maestria	varchar	6	Código de identificación de una maestría

Nº	NOMBRE	TIPO	LONGITUD	DESCRIPCION
30	cod_materia	varchar	6	Código de identificación de una Materia
31	cod_materiareq	varchar	6	Código que identifica a una materia con requisito
32	cod_municipio	varchar	5	Código de identificación de un municipio de un departamento geográfico
33	cod_planestudio	varchar	6	Código que identifica al plan de estudio de la maestría
34	cod_practica	int	4	Código que identifica a una práctica
35	cod_publicacion	int	4	Código que identifica a una publicidad
36	cod_publicidad	int	4	Código que identifica a una publicidad
37	cod_referencia	int	4	Código que identifica a una referencia personal de un estudiante
38	cod_universidad	varchar	6	Código de identificación de una universidad
39	Concepto	varchar	50	Descripción por la cual se emite un recibo
40	Contrato	varchar	40	Tipo de contrato que tiene en el trabajo
41	Correo	varchar	40	Correo electrónico del estudiante o lugar de practicas
42	correo_ref	varchar	40	Correo electrónico de la persona de referencia
43	Cuota	int	4	Numero de cuota de pago mensual de un estudiante
44	Descripción	varchar	50	Descripción por la cual se emite un recibo

Nº	NOMBRE	TIPO	LONGITUD	DESCRIPCION
45	direccion	text	16	Lugar de residencia actual del estudiante o lugar de prácticas
46	direccion_ref	text	16	Lugar de residencia actual de la persona de referencia
47	Duración	int	4	Duración de la maestría
48	Edición	smallint	2	Número de edición de una maestría
49	Edicionmaestria	smallint	2	Número de edición de la maestría en que entra a cursar un estudiante
50	Escribe	varchar	10	Porcentaje que escribe un determinado idioma
51	Especialidad	varchar	20	Nombre de especialidad del docente
52	Estado	bit	1	Estado del estudiante respecto a los pagos 1- Activo , 0-Desactivado
53	estado_civil	varchar	20	Estado civil de un estudiante (Soltero, Casado, Divorciado, Viudo)
54	Fax	varchar	9	Fax del lugar del lugar donde realizo las prácticas
55	Fecha	varchar	10	Fecha de pago de recibo de matricula
56	Fecha	varchar	10	fecha en que el usuario ingreso al sistema
57	fecha_egreso	varchar	10	Fecha en que egreso el estudiante de una determinada carrera
58	fecha_ingreso	varchar	10	Fecha de ingreso a la maestría
59	fecha_inicio	varchar	10	Fecha de inicio de una maestría

Nº	NOMBRE	TIPO	LONGITUD	DESCRIPCION
60	fecha_pago	varchar	10	Fecha de pago de recibo de pago mensual o matricula de la maestría
61	forma_grad	varchar	20	Forma de Graduación del estudiante de su carrera
62	forma_grad_plan	varchar	40	Forma de Graduación del plan de estudio de la maestría
63	Foto	image	16	Foto que identifica a un docente
64	foto_est	image	16	Foto que identifica a un estudiante
65	Habla	varchar	10	Porcentaje que habla un determinado idioma
66	Idioma	varchar	80	Nombre del idioma que habla el estudiante
67	inf_adicional	text	16	Información adicional que el estudiante quiere ingresar
68	institucion	varchar	50	Nombre de institución donde trabaja
69	Lee	varchar	10	Porcentaje que lee un determinado idioma
70	Login	varchar	50	login del estudiante en el sitio web generado por el sistema
71	Login	varchar	50	login del usuario en el sistema
72	login_user	varchar	40	Login del usuario en uso del sistema
73	medio_publicidad	varchar	40	Medio de Comunicación por el cual se entero de la maestría
74	mes_pago	varchar	20	Mes cancelado del pago mensual

Nº	NOMBRE	TIPO	LONGITUD	DESCRIPCION
75	Monto	nchar	8	Monto equivalente de lo que vale la maestría ó matricula ó pago mensual
76	monto_actual	nchar	8	Monto equivalente de lo que vale la matricula de la maestría
77	monto_cordoba	nchar	8	Total que percibe por el pago mensual ó matricula en córdoba
78	monto_dolar	nchar	19	Total que percibe por el pago mensual de la matricula o pago mensual en dólar
79	Motivación	text	16	Párrafo que explica el porqué va a estudiar la maestría
80	Nacionalidad	varchar	50	Nacionalidad del estudiante y/o Docente
81	nivel_academico	varchar	20	Grado ó nivel académico del docente
82	nom_carrera	varchar	50	Nombre que se le asigna a una carrera determinada
83	nom_categoria	varchar	20	Nombre que se le asigna a una categoría determinada
84	nom_depto	varchar	6	Nombre que se le asigna a un departamento académico
85	nom_docente	varchar	40	Especifica los nombres del docente
86	nom_dpto	varchar	50	Nombre que se le asigna a un departamento geográfico de Nicaragua del estudiante
87	nom_facultad	varchar	50	Nombre que identifica a la facultad
88	nom_maestria	text	16	Nombre que se le asigna a la maestría
89	nom_materia	varchar	50	Nombre que se le asigna a una determinada carrera

Nº	NOMBRE	TIPO	LONGITUD	DESCRIPCION
90	nom_municipio	varchar	50	Nombre de un municipio que pertenece a un departamento geográfico
91	nom_publicacion	text	16	Nombre que se le asigna a una publicación realizada por un estudiante
92	nom_tesis	text	16	Nombre de la tesis de graduación
93	nom_universidad	varchar	50	Nombre que se le asigna a una universidad donde estudio el estudiante
94	nom_usuario	varchar	40	Nombres del usuario que maneja el sistema
95	nom1	varchar	12	Primer nombre de un docente
96	nom2	varchar	20	Segundo nombre de un docente
97	nombre_ref	varchar	40	Nombres de las personas de referencia de un estudiante
98	Nombres	varchar	50	Nombres del estudiante
99	Nota	varchar	3	Nota final de una asignatura
100	num_carnet	varchar	10	Número de Carnet que identifica un estudiante (único)
101	num_cedula	varchar	10	Número de cedula de un estudiante (único)
102	num_cheque	char	20	Número de cheque que paga la matricula
103	num_creditos	smallint	2	Número de créditos de una materia en el semestre
104	num_horas	smallint	2	Número de horas de una materia en el semestre

Nº	NOMBRE	TIPO	LONGITUD	DESCRIPCION
105	num_horas_plan	smallint	2	Número de horas del plan de estudio de la maestría
106	num_modulo	smallint	2	Número de módulo de una Materia
107	num_recibo	varchar	7	Número de identificación única de un recibo de pago en matricula ó pago
108	num_seguro	varchar	15	Número de Seguro de un estudiante (único)
109	num_vitacora	int	4	Número de veces ingresado al sistema un determinado usuario
110	País	varchar	40	Nombre del país de la universidad
111	Password	varchar	200	Contraseña del estudiante para el sitio web generado por el sistema
112	Permiso	varchar	20	Tipo de permiso del usuario al ingresar al sistema (Administrador, Invitado)
113	plan_estudio	int	4	año del plan de estudio de la maestría
114	Poract	int	4	Porcentaje de Becado de un estudiante
115	Porcentaje	int	4	Porcentaje de Becado de un estudiante
116	req1	varchar	6	Código de materia prerrequisito necesario para cursar una asignatura
117	req2	varchar	6	Código de materia prerrequisito necesario para cursar una asignatura
118	req3	varchar	6	Código de materia prerrequisito necesario para cursar una asignatura
119	Sexo	char	1	Género del Estudiante

Nº	NOMBRE	TIPO	LONGITUD	DESCRIPCION
120	tel_ref	varchar	9	Número teléfono de la persona de referencia
121	Teléfono	varchar	9	Número teléfono del estudiante o lugar de prácticas
122	Tema	text	16	Tema de publicación de un estudiante
123	tipo_pago	char	10	Tipo de pago de recibo de matricula
124	tipo_practicas	varchar	15	Tipo de prácticas realizadas por un estudiante
125	tipo_publicacion	varchar	40	tipo de publicaciones de un estudiante
126	tipo_universidad	varchar	20	Tipo de universidad donde estudio el estudiante (Pública ó Privada)
127	Titulo	varchar	20	Nombre del título que obtuvo el estudiante al egresar de la carrera que estudio
128	titulo_obtener	varchar	50	Título profesional a obtener en la maestría
129	Traduce	varchar	10	Porcentaje que traduce un determinado idioma

Anexo 2 – Codificación de catalogo de departamentos y municipios

Departamentos Geográficos

Código	Nombre del Departamento
01	Chinandega
02	León
03	Managua
04	Masaya
05	Carazo
06	Granada
07	Rivas
08	Boaco
09	Nueva Segovia
10	Madriz
11	Estelí
12	Chontales
13	Jinotega
14	Matagalpa
15	Rio San Juan
16	RAAN
17	RAAS

Municipios de los Departamentos

Departamento **Chinandega**

Código	Nombre de Municipio
01-01	El Viejo
01-02	Puerto Morazán
01-03	Somotillo
01-04	Santo Tomás del Norte
01-05	Cinco Pinos
01-06	San Pedro del Norte
01-07	San Francisco del Norte
01-08	Villanueva
01-09	Chinandega
01-10	El Realejo
01-11	Corinto
01-12	Chichigalpa
01-13	Posoltega

Departamento **León**

Código	Nombre de Municipio
02-01	León
02-02	Quezalguaque
02-03	Telica
02-04	Larreynaga
02-05	El Sauce
02-06	Achuapa
02-07	Santa Rosa del Peñón
02-08	El Jicaral
02-09	La Paz Centro
02-10	Nagarote

Departamento **Managua**

Código	Nombre de Municipio
03-01	San Francisco Libre
03-02	Tipitapa
03-03	Mateare
03-04	Villa Carlos Fonseca
03-05	San Rafael del Sur
03-06	Managua
03-07	Ticuantepe
03-08	El Crucero
03-09	Ciudad Sandino
03-10	Sabana Grande

Departamento **Masaya**

Código	Nombre de Municipio
04-01	La Concepción
04-02	Nindiri
04-03	Masaya
04-04	Tisma
04-05	Masatepe
04-06	Nandasmo
04-07	Catarina
04-08	Niquinohomo
04-09	San Juan de Oriente

Departamento **Granada**

Código	Nombre de Municipio
06-01	Granada
06-02	Diría
06-03	Diriomo
06-04	Nandaime

Departamento **Carazo**

Código	Nombre de Municipio
05-01	San Marcos
05-02	Diriamba
05-03	Dolores
05-04	Jinotepe
05-05	El Rosario
05-06	La Paz de Carazo
05-07	Santa Teresa
05-08	La Conquista

Departamento **Rivas**

Código	Nombre de Municipio
07-01	Tola
07-02	Belén
07-03	Potosi
07-04	Buenos Aires
07-05	Rivas
07-06	San Jorge
07-07	San Juan del Sur
07-08	Cardenas
07-09	Moyogalpa
07-10	Altagracia

Departamento **Boaco**

Código	Nombre de Municipio
08-01	San José de los Remates
08-02	Teustepe
08-03	Santa Lucia
08-04	Boaco
08-05	Camoapa
08-06	San Lorenzo

Departamento **Nueva Segovia**

Código	Nombre de Municipio
09-01	Santa Maria
09-02	Macuelizo
09-03	Dipilto
09-04	Ocotal
09-05	Mozonte
09-06	San Fernando
09-07	Ciudad Antigua
09-08	El Jícaro
09-09	Jalapa
09-10	Murra
09-11	Quilalí

Departamento **Madriz**

Código	Nombre de Municipio
10-01	Somoto
10-02	Totogalpa
10-03	Telpaneca
10-04	San Juan de Rio Coco
10-05	Yalaguina
10-06	Palacaguina
10-07	San Lucas
10-08	Las Sabanas
10-09	San José de Cusmapa

Departamento **Estelí**

Código	Nombre de Municipio
11-01	Pueblo Nuevo
11-02	Condega
11-03	San Juan de Limay
11-04	Estelí
11-05	La Trinidad
11-06	San Nicolás

Departamento **Chontales**

Código	Nombre de Municipio
12-01	Comalapa
12-02	Juigalpa
12-03	La Libertad
12-04	Santo Domingo
12-05	San Pedro de Lóvago
12-06	Santo Tomas
12-07	Acoyapa
12-08	Villa Sandino
12-09	Cuapa
12-10	El Coral

Departamento **Jinotega**

Código	Nombre de Municipio
13-01	Wiwili
13-02	Cuá Bocay
13-03	San Sebastian de Yali
13-04	La Concordia
13-05	San Rafael del Norte
13-06	Santa Maria de Pantasma
13-07	Jinotega

Departamento **Rio San Juan**

Código	Nombre de Municipio
15-01	Morrito
15-02	El Almendro
14-03	San Miguelito
14-04	San Carlos
14-05	El Castillo
14-06	San Juan del Norte

Departamento **RAAS**

Código	Nombre de Municipio
17-01	Bocana de Paiwas
17-02	Tortuguero
17-03	La Cruz de Rio Grande
17-04	Laguna de Perlas
17-05	Kukarahill
17-06	El Rama
17-07	Muelle de los Bueyes
17-08	Nueva Guinea
17-09	Bluefields
17-10	Corns Island

Departamento **Matagalpa**

Código	Nombre de Municipio
14-01	San Isidro
14-02	Sébaco
14-03	Ciudad Dario
14-04	Terrabona
14-05	San Dionisio
14-06	Esquipulas
14-07	Muy Muy
14-08	Matagalpa
14-09	San Ramón
14-10	El Tuma - La Dalia
14-11	Rancho Grande
14-12	Matiguas
14-13	Rio Blanco

Departamento **RAAN**

Código	Nombre de Municipio
16-01	Waspán
16-02	Bonanza
16-03	Rosita
16-04	Puerto Cabeza
16-05	Waslala
16-06	Siuna
16-07	Prinzapolka

Anexo 3 – Plan de estudios del programa de maestría en computación

Los cursos están organizados en 4 módulos que a continuación se describen:

MODULO	CODIGO	CURSO	H.T	H.P	T.H	T.C	REQUISITOS
I	MC-101	Negocios Electrónicos	40	56	96	4	
I	MC-102	Ingeniería del Software	40	56	96	4	
I	MC-103	Inteligencia de Negocios	40	56	96	4	
I	MC-104	Bases de Datos Avanzadas	40	56	96	4	
		Total	**160**	**224**	**384**	**16**	
II	MC-105	Auditoria de Sistemas Computacionales	40	56	96	4	
II	MC-106	Metodología Orientada a Objetos	40	56	96	4	
II	MC-107	Administración de Proyectos I	40	56	96	4	
II	MC-108	Sistemas para la Toma de Decisiones	40	56	96	4	
		Total	**160**	**224**	**384**	**16**	
III	MC-109	Electiva	40	56	96	4	
III	MC-110	Programa Orientado a Objetos	40	56	96	4	MC-106
III	MC-111	Administración de Proyectos II	40	56	96	4	MC-107
III	MC-112	Arquitectura de Sistemas de Información en Web	40	56	96	4	
		Total	**160**	**224**	**384**	**16**	
IV	MC-113	Planificación Estratégica de TI	40	16	56	3	MC-111
IV	MC-114	Control de Calidad de Software	40	16	56	3	
IV	MC-115	Comunicación Científica	40	16	56	3	
IV	MC-116	Tesis de Graduación para Maestría			160	4	
		Total	120	48	328	13	
		Defensa			80	2	
		Total			80	2	
		TOTAL	600	720	1560	63	

Nota: El Curso de Electiva es un curso que el estudiante puede llevar de acuerdo a los contenidos que sean de su interés.

Anexo 4 – Cotizaciones

COMTECH
Tecnología Computarizada

DATOS DEL CLIENTE

Señores : UNAN MANAGUA
Atención: GERALD ORTEGA

DATOS DE LA PROFORMA

Nº. Ref. : **AR-EB031210**
Fecha : 03 de diciembre de 2010
Asunto : Presentación de oferta

COMTECH tiene el gusto de ofertarle, los accesorios y equipos de computación con las especificaciones técnicas solicitadas a precios económicos.

COTIZACION Nº. AR-EB031210			
CANT	DESCRIPCION	P. UNIT U$	TOTAL U$
01	**SERVIDOR** **MARCA HP** **MODELO PROLIANT ML350 G6 -/487930-001** Processor: 1 x Intel Xeon E5520 / 2.26 GHz (Quad-Core) Cache Memory: 8 MB L3 cache Cache Per Processor: 8 MB **RAM**: **6 GB (installed)** / 192 GB (max) - DDR3 SDRAM - 1066 MHz - PC3-10600 Storage Controller: RAID (Serial ATA-150 / SAS) - PCI Express x4 (Smart Array P410i) Server Storage Bays: Hot-swap 2.5" Optical Storage: DVD-ROM Disco de Duro: 500GB Video Memory: 64 MB Networking: Network adapter - PCI Express x4 - Ethernet, Fast Ethernet, Gigabit Ethernet - Ethernet Ports : 2 x Gigabit Ethernet Power: AC 120/230 V (50/60 Hz) **Tiempo de entrega inmediata** **Garantía 1 años**	2,168.00	2,168.00
		SUB TOTAL	2,168.00
		IVA	325.20
		TOTAL	2,493.20

- **Tiempo de Entrega: 6 HORAS HABILES DESPUES DE SU FACTURACION**
- Forma de pago: Contado y/o ck a/n de COMTECH, Somos grandes contribuyentes
- **GARANTIA DE 1 AÑO**
- Precios cotizados en CORDOBAS/ Tipo de cambio paralelo 22.05

Muy atentamente:

ERICA BOLAÑOS
ASESOR DE VENTAS
Tecnología Computarizada-COMTECH/Telf. 267-4012 ext. #115

COMTECH
Tecnologia Computarizada

DATOS DEL CLIENTE

Señores: UNAN MANAGUA /DPTO DE COMPUTACION
ATENCION :

DATOS DE LA PROFORMA

Nº. Ref. : AR-EB231110
Fecha : 23 de noviembre de 2010
Asunto : Presentación de oferta

COMTECH tiene el gusto de ofertarle, los accesorios y equipos de computación con las especificaciones técnicas solicitadas a precios económicos.

COTIZACIÓN Nº. AR-EB231110			
CANT	DESCRIPCION	P. UNIT U$	TOTAL U$
01	COMPUTADOAR DE ESCRITORIO CLON Case ATX Minitorre Tarjeta Madre INTEL 775 Procesador **INTEL CORE 2 DUO 2.9/MHz** Memoria RAM DDR2 3GB Kingston **Disco Duro de 320GB SATA 7200 RPM** **QUEMADOR DE DVD Y CD MUTI-SENCILLO** 2 PS/2 para mouse y teclado, 4 puertos USB **TECLADO Y MOUSE,ALMOHADILLA Y PARLANTES** **GARANTIA 1 AÑO** **TEIMPO DE ENTREGA INMEDIATA**	420.00	420.00
02	COMPUTADOAR DE ESCRITORIO CLON Case ATX Minitorre Tarjeta Madre INTEL 775 Procesador **INTEL CORE 2 DUO 2.9/MHz** Memoria RAM DDR2 2GB Kingston **Disco Duro de 320GB SATA 7200 RPM** **QUEMADOR DE DVD Y CD MUTI-SENCILLO** 2 PS/2 para mouse y teclado, 4 puertos USB **TECLADO Y MOUSE,ALMOHADILLA Y PARLANTES** **MONITOR LCD 17" PANTALLA PLANA** **GARANTIA 1 AÑO** **TEIMPO DE ENTREGA INMEDIATA**	505.00	1,010.00
02	IMPRESORA HP 2000 DESKJET INKJET /CH390C#AKY	44.00	88.00
02	MUEBLE P/COMPUTADORA / AM100GEN08	19.50	39.00
02	SILLA CON BRAZO/COLOR NEGRO/AM160GEN76	34.00	68.00
		SUB TOTAL	1.625.00
		IVA	243.75
		TOTAL	1,868.75

COMTECH, S.A.
Ing. Ericka Bolaños
Asesora de Ventas
Tel:2525767 Fax:2706224

- **INSTALE LICENCIAS ORIGINALES (COMTECH no se hace responsable x daños en**
- **Tiempo de Entrega: 6 HORAS HABILES DESPUES DE SU FACTURACION**
- Forma de pago: Contado y/o ck a/n de COMTECH, Somos grandes contribuyentes
- **GARANTIA DE 1 AÑO**
- Precios cotizados en DOLARES/ Tipo de cambio paralelo 21.99

Avenida principal Altamira D'Este Nº 589 - Ferreteria SINSA, 25 vrs. arriba - Managua, Nicaragua
PBX: (505) 267-4012 Fax: (505) 270-6224 - E-mail: comtech@cablenet.com.ni - www.comtech.com.ni

COMTECH
Tecnología Computarizada

- Esta oferta es válida por 8 días

COMTECH, S.A.
Erica Bolaños
Asesora de Ventas
Tel:2525767 Fax:2706224

Muy atentamente:
ERICA BOLAÑOS
ASESOR DE VENTAS
Tecnología Computarizada-COMTECH/Telf. 267-4012 ext. #115

Avenida principal Altamira D'Este Nº 589 - Ferretería SINSA, 25 vrs. arriba - Managua, Nicaragua
PBX: (505) 267-4012 Fax: (505) 270-6224 - E-mail: comtech@cablenet.com.ni - www.comtech.com.ni

Anexo 5 – Hoja de Inscripción a la Maestría (Formulario)

Universidad Nacional Autónoma de Nicaragua
Recinto Universitario Rubén Darío
Facultad de Ciencias
Departamento de Computación, Pabellón # 16
Teléfonos: 2278 -6764 / 69 Ext 128, Fax: 2278-2950
E-mail: pmc@unan.edu.ni

Fecha
/ /

Datos Personales

Nº de cedula

Fecha Nacimiento
/ /

Foto

Apellidos

Nombres

Sexo M ◯ F ◯

Estado Civil Soltero ◯ Casado ◯

Teléfono

Móvil

Dirección

País

Departamento

Municipio

Dirección domiciliar

Datos Académicos

Universidad

Tipo Estatal ◯ Privada ◯

Carrera

Fecha
/ /

Título Otorgado

Modalidad Graduación

Monografía ◯ Seminario ◯ Examen ◯

Nombre de tesis de graduación

Datos Profesionales

Dirección

Nombre institución

Dirección institución

Teléfono

Fax

Tipo de contratación

Cargo laboral

Correo electrónico

Conocimiento de Idiomas

Idioma

Dominio

Habla	Lectura	Escritura
O Bien	O Bien	O Bien
O Regular	O Regular	O Regular
O Mal	O Mal	O Mal

En caso de ejercer la docencia complete los siguientes datos

Datos de Docencia

Categoría Docente

- O Profesor Titular
- O Profesor (II)
- O Profesor (III)
- O Profesor (IV)

Universidad

Publicaciones

Nombre publicación	Año de publicación		
	DD	MM	AAAA
Nombre publicación	DD	MM	AAAA

Investigaciones

Nombre investigación	Año de publicación		
	DD	MM	AAAA
Nombre investigación	DD	MM	AAAA

Referencias

Nº de Cedula

Fecha de Nacimiento

/ /

Apellidos

Nombres

Nombres

Correo electrónico

Teléfono

Ocupación

Dirección

País

Departamento

Municipio

Dirección domiciliar

Referencias

Nº de Cedula

Fecha de Nacimiento

/ /

Apellidos

Nombres

Nombres

Correo electrónico

Teléfono

Ocupación

Dirección

País

Departamento

Municipio

Dirección domiciliar

Referencias

Nº de Cedula

Fecha de Nacimiento

/ /

Apellidos

Nombres

Correo electrónico

Teléfono

Ocupación

Dirección

País

Departamento

Municipio

Dirección domiciliar

Información adicional

Motivación

Medio de publicidad por el que se enteró del programa

☐ Internet ☐ Publicidad en la vía pública ☐ Otro

Otra información de interés

Firma del Solicitante