

Ministerio de Educación Superior

Universidad de Camagüey

Facultad de Informática



Trabajo de Diploma

SACAP

Sistema Automatizado para el Control de la

Actividad de Postgrado

Autores

Genoveva Cevallos Masapanta

Maribel Uribe Campaña

Tutores

Msc. Rigre Garciandía Sónora

Lcda. Magalys Palomino Palomino

“AÑO DE LA ALTERNATIVA BOLIVARIANA PARA LAS AMÉRICAS”

2004-2005

***UNIVERSIDAD TÉCNICA DE
COTOPAXI***



CARRERA:

***CIENCIAS DE LA INGENIERÍA Y
APLICADAS***

ESPECIALIDAD:

***INGENIERÍA EN INFORMÁTICA Y
SISTEMAS COMPUTACIONALES***

Dedicatoria

Dedicamos este trabajo principalmente a nuestros padres y hermanos, los cuales han sido el pilar fundamental a lo largo de nuestra vida, siendo el faro y guía del camino aunado.

Ellos con su amor, esfuerzos y enseñanzas se han convertido en la fuerza que nos impulsa a alcanzar nuestros objetivos y hacer realidad cada uno de nuestros sueños, nos han formado como verdaderas hijas, buenas estudiantes y ahora grandes profesionales.

A nuestros familiares y amigos, que a pesar de la distancia nos apoyaron en esta experiencia inolvidable.

A todos los que han hecho suyos nuestros desvelos y esfuerzos, y que hoy comparten nuestros logros y satisfacciones en la culminación de una etapa mas en nuestras vidas.

Este es el fruto de nuestro esfuerzo Genoveva y Maribel.

Agradecimientos.

Queremos expresar nuestros mas sinceros agradecimientos a nuestros tutores, por su amistad, sabiduría y apoyo incondicional, durante todo el tiempo que compartimos junto a ellos, sabiendo ser una guía principal para el desarrollo y culminación del presente trabajo.

A todas las autoridades que intervinieron en la cristalización del convenio entre la Universidad Técnica de Cotopaxi y la Universidad de Camaguey y así hacer realidad la idea de conocer un país hermano, su pueblo, su cultura y su educación.

A todos y cada uno de los profesores de la carrera de Ciencias de la Ingeniería y Aplicadas de la Universidad Técnica de Cotopaxi; así como también de la Facultad de Informática de la Universidad de Camaguey, quienes han formado parte de nuestra vida estudiantil e hicieron posible este bello sueño. Extendemos este agradecimiento también a los profesores y autoridades de las dos universidades en general de las cuales nos sentimos orgullosas de pertenecer.

A nuestros familiares por sus consejos, apoyo, confianza y comprensión; principalmente a nuestros padres que han dado todo por hacer posible la realización de una de nuestras mayores aspiraciones.

A todas las personas, compañeros y amigos con quienes hemos compartido momentos inolvidables.

A todos muchas gracias.

Pensamiento

“Educar es depositar en cada hombre toda la obra humana que le ha antecedido: es hacer a cada hombre resumen del mundo viviente, hasta el día en que vive: es ponerlo a nivel de su tiempo, para que flote sobre él, y no dejarlo debajo de su tiempo, con lo que no podrá salir a flote; es preparar al hombre para la vida”

José Martí

Resumen

Dentro de los principales objetivos en el trabajo de la Educación Superior en Cuba esta la preparación de postgrado, de la que forman parte los procesos formativos y de desarrollo tales como: enseñanza-aprendizaje, investigación, innovación, creación artística, entre otros; articulados armónicamente en una propuesta docente - educativa pertinente a este nivel. La Universidad de Camagüey (UC) cuenta con un Centro de Postgrado Internacional (CEPI), donde se controlan las actividades de Superación Profesional y Formación Académica correspondientes a la preparación de postgrado.

Toda la actividad de dicho centro se la realiza a partir de la recepción de información emitida por las diferentes Facultades y el proceso de la misma actualmente posee retardo en la emisión de certificaciones, menor aprovechamiento de recursos humanos, materiales y tecnológicos y a su vez una demora innecesaria.

Después de analizar la dificultad existente en el tratamiento de la información para el proceso de certificación y control de la actividad de postgrado, el presente trabajo comprende, como objetivo principal el diseño, creación e implementación de un Sistema Automatizado que permita mejorar, fortalecer y modernizar el control y la gestión de la Actividad de Postgrado del CEPI en la Universidad de Camagüey, para mejorar el uso racional de los recursos, mayor consistencia, fiabilidad y seguridad de la información así como facilitar el acceso y manipulación de la misma.

Summary

Inside the main objectives in the work of the Superior Education in Cuba is the graduate degree preparation, in which the formative and development processes are included such as: teaching-learning, investigation, innovation, artistic creation, and others; they are articulated harmoniously in an educational proposal - educational pertinent at this level. The University of Camagüey (UC) it has a Center of International Graduate degree (CEPI), where the activities of Superation Professional and Academic Formation corresponding to the graduate degree preparation are controlled.

The whole activity of this center is carried out starting from the reception of information emitted by the different specialities and the process of it at the moment possesses retard in the emission of certifications, less use of human, materials and technological resources and in addition an unnecessary delay.

After analyzing the existent difficulty in the treatment of the information for the certification process and control of the graduate degree activity, the present work has, as main objective the design, creation and implementation of an Automated System that it allows to improve, to strengthen and to modernize the control and the administration of the Activity of Graduate degree of the CEPI in the University of Camagüey in order, to improve the rational use of the resources, bigger consistency, reliability and security of the information as well as to facilitate the access and manipulation of the same one.

Índice

Introducción.....	1
Capítulo I. Fundamentación Teórica.....	6
1.1. Metodologías.	6
1.2. Arquitectura Cliente - Servidor.	9
1.3. Lenguajes de Programación de Alto Nivel.....	12
1.4 Selección del Leguaje de Programación a utilizar.....	18
1.5. Sistemas Gestores de Bases de Datos (SGBD).	21
1.6. Selección del Sistema Gestor de Base de Datos.....	25
Resumen.	27
Capítulo II. Fundamentación de la Propuesta.....	28
2.1. Descripción del Objeto de Estudio.	28
2.2. Objeto de Automatización.	29
2.3. Propuesta del Sistema.....	31
2.4. Requisitos de la Aplicación.	34
Resumen	40
Capítulo III. Análisis y Diseño General del Sistema.	41
3.1 Definición de Casos de Uso.	42
3.2 Definición de Actores.....	63
3.3. Definición del Modelo Conceptual.....	63
3.4. Diseño de la Base de Datos.	64
3.5. Diseño Físico de la Base de los Datos.....	64
3.6. Diagramas de Secuencia.....	65
3.7. Diagramas de Clases.....	65
3.8. Diagramas de Componentes.	65
3.9. Diagramas de Despliegue.	66
3.10. Patrones en el Desarrollo del Sistema	66
Resumen	74

Capitulo IV. Prototipo del Sistema	75
4.1. Principios generales del Diseño de las Interfaces de Usuario.	75
4.2. Principios Generales del Sistema SACAP.....	76
4.3. Principios Generales del tratamiento de los Errores.....	88
4.4. Principios Generales del Sistema de Ayuda	91
4.5. Concepción General del Sistema de Seguridad y Protección.....	92
Resumen	93
Conclusiones.....	94
Recomendaciones.....	96
Referencias Bibliográficas	97
Bibliografía.....	100
Anexos.....

Capítulo I. Fundamentación Teórica

La rápida difusión y el gran interés en el mundo de la informática han revolucionado los diferentes procesos que permiten la creación de aplicaciones de última generación mucho más simples, eficaces y eficientes; esto a su vez provoca un cambio en la forma de vida del ser humano. Todo esto indiscutiblemente establece un cambio en la organización, nuevas herramientas de trabajo, cambios en la metodología y en los recursos de forma cuantitativa y cualitativa.

En este capítulo se realizará un análisis del estado actual de las tecnologías que serán necesarias para la construcción del sistema que se pretende desarrollar. De la misma forma se definirán conceptos básicos para un mayor entendimiento del sistema. Se analizarán diferentes modelos de confección de aplicaciones, tales como el modelo cliente-servidor, y modelos de programación por capas. También se realiza un análisis comparativo entre diferentes lenguajes de programación, con el objetivo de hacer una selección de aquel que se adapte lo mejor posible a las condiciones del sistema que se desea implementar, otro de los aspectos que será analizado en el transcurso de este capítulo será la selección del sistema de gestión de bases de datos adecuado.

1.1. Metodologías.

La necesidad de crear un orden en el proceso de construcción de un software conlleva a hacer uso de una metodología que permita cumplir con las expectativas deseadas. La experiencia ha demostrado que la clave del éxito de un proyecto de software es la elección correcta de la metodología para así guiar al programador a desarrollar un sistema de software adecuado. Todo esto al mismo tiempo ayuda a elegir las mejores y

más potentes herramientas. Dentro de la lógica del mercado mundial la exigencia al desarrollar productos de software en menor tiempo y costo, ha incrementado considerablemente, es por eso que se vuelve imprescindible profesionalizar la labor de desarrollo al aplicar un proceso de ingeniería adecuado.

Dentro de las metodologías que se puede citar para la construcción del sistema están.

La Metodología RUP (Proceso Unificado de Rational) *“es el conjunto de actividades necesarias para transformar los requerimientos del usuario en un sistema informático”* [Addison-Wesley, 2000]; es decir considera que un proceso define quién está haciendo qué, cuándo y cómo alcanzar un determinado objetivo. Su estructura es bidimensional, en donde en el eje vertical están las distintas etapas del desarrollo en cascada (flujos de trabajo), y en el eje horizontal la evolución en el tiempo.

La Metodología ADOOSI Visual (Metodología de Análisis y Diseño orientado a objetos para Sistemas Informáticos en medios ambientes Visuales), *“cubre todas las etapas del ciclo de vida de un proyecto informático, garantizando el desarrollo de éste en forma interactiva (refinando los productos de etapas y fases antes hechas), incremental (construyendo el proyecto por partes que se integran) y en paralelo (realizando el proyecto por subgrupos de analistas)”* [Álvarez, 1997]; por lo que el ciclo de vida del proyecto no es en cascada como se ha tratado tradicionalmente.

1.1.1. Metodología de Análisis y Diseño Orientado a Objetos para Sistemas Informáticos en medios ambientes Visuales (ADOOSI Visual).

La metodología ADOOSI Visual, *“se utiliza para la realización de todo el ciclo de vida (estudio preliminar, análisis, diseño, desarrollo, prueba e implantación) de un proyecto informático basado la tecnología orientada a objetos. [Álvarez, 1997]; en la cual se maneja el principio cliente-servidor y dentro de éste más específicamente en el principio de diseño, manejado por responsabilidades; es por eso los conceptos utilizados son: cliente, servidor, contratos, colaboración, etc.*

Dicha metodología *“tiene en cuenta las características del desarrollo de la aplicación en medios ambientes de programación visual como Visual FoxPro y Delphi [Álvarez, 1997]; esta filosofía de trabajo se fundamenta en crear un prototipo (software rudimentario del sistema a elaborar), que permite al usuario examinar, ejecutar y de acuerdo a su criterio refinar en forma interactiva y dinámica con el objetivo de llegar al producto final.*

1.1.2. Selección de la Metodología a utilizar.

Las dos metodologías analizadas cumplen con los requisitos exigidos por la aplicación que se desea elaborar ya que presentan todos los aspectos necesarios para elaborar trabajos similares al que se esta diseñando, para el control del proyecto se seleccionó la metodología ADOOSI ya que cumple con los requerimientos exigidos por la aplicación que se desea realizar y además es una metodología cubana y de fácil uso. No obstante se usaron herramientas de modelación incluidas en los paquetes ofrecidos por Rational.

Para los diagramas de modelación se utilizará el Lenguaje de Modelación Unificado (UML).

1.2. Arquitectura Cliente - Servidor.

Dos conceptos destacan la conexión entre el mundo de los negocios y el mundo de las computadoras, las reglas y los requerimientos del negocio todo tipo de aplicación trata de reflejar parte del funcionamiento del mundo real. *“La arquitectura Cliente – Servidor es utilizada por todas las aplicaciones de Internet e Intranet, en donde el servidor contiene una multitud de información correspondiente a los clientes”* [Castaño y Piattini, 2003]. Las reglas del negocio no son mas que restricciones que tiene que cumplir el sistema, que de otro modo serian llevadas a cabo de modo menos eficiente o bien no podrían realizarse, para ello es necesario que cada aplicación refleje las restricciones que existen, de modo que nunca sea posible llevar a cabo acciones no validas a las reglas que debe seguir la aplicación. Los requerimientos del negocio por su parte son restricciones que usualmente son expuestas por la propia compañía o entidad, que estas utilizan para colaborar en su ambiente particular de negocio, estos definen una línea para los requerimientos de la aplicación y proporcionan una guía para los desarrolladores en términos prácticos, estos requerimientos se traducen en metas que los desarrolladores se esfuerzan por conocer para el desarrollo de sus aplicaciones

Juntos a *“las reglas y requerimientos del negocio proporcionan un entorno para desarrollar aplicaciones que puedan ayudar a la compañía a alcanzar sus metas”* [Agulló, 1997], en la actualidad se cuenta con técnicas y arquitecturas de desarrollo de sistemas que facilitan la modelación de estos requerimientos y reglas de negocio. Entre las mas utilizadas se encuentran los sistemas con arquitectura cliente - servidor los que

normalmente se divide en capas encapsulando en cada de ellas diferentes funcionalidades que permiten presentarle al usuario un producto único que cumpla con sus expectativas.

Según [DataPro, 1994]; la arquitectura cliente servidor es un modelo de computación en que el procesamiento requerido para ejecutar una aplicación o conjunto de aplicaciones, relacionadas se dividen entre 2 o mas procesos que cooperan entre si.

1.2.1. Componentes de la Arquitectura Cliente - Servidor.

El Cliente es quien interactúa con el usuario usualmente en forma grafica, se comunican con procesos auxiliares que se encargan de establecer la conexión con el servidor, envía el pedido, recibir las respuestas, manejar las fallas y realizar actividades de sincronización y de seguridad.

El Servidor proporciona un servicio al cliente y devuelve los resultados, en algunos casos existen procesos auxiliares que se encargan de recibir las solicitudes del cliente, verificar la protección, activar un proceso servidor para satisfacer el pedido, recibir sus respuestas y enviarlas al cliente.

La infraestructura de comunicaciones es la que proporciona diferentes tipos de mecanismos que hacen posible la comunicación entre el cliente y el servidor.

1.2.2. Características de la Arquitectura Cliente - Servidor.

Dada la importancia de esta arquitectura veremos a continuación algunas de sus características:

- Usualmente la mayoría del trabajo pesado se hace en el proceso llamado servidor (s) y el o los procesos cliente (s) solo se encargan de la interacción con el usuario.
- El servidor posee su información instalada localmente.
- El servidor se puede encontrar en cualquier lugar de la red.
- Esta arquitectura trabaja sobre una red de ordenadores, en donde el ordenador cliente es el que inicia la consulta y el ordenador servidor, es quien atiende dicha consulta.
- Estas peticiones y respuestas son transferencias de datos que cada ordenador cliente se encarga de mostrar por pantalla, presentar informes tabulados, imprimir, guardar, etc. dejando al servidor libre.

1.2.3. Diferentes tipos de Arquitectura Cliente - Servidor.

Dentro de los Sistemas Cliente - Servidor podemos citar algunos, dependiendo del tipo de aplicaciones que el servidor pone a disposición de los clientes.

- Servidores de Impresión, mediante el cual los usuarios comparten impresoras.
- Servidores de Archivos, con el cual los clientes comparten discos duros.
- Servidores de Bases de Datos, donde existe una única base de datos.
- Servidores de Lotus Notes, que permite el trabajo simultáneo de distintos clientes con los mismos datos, documentos o modelos.

1. 3. Lenguajes de Programación de Alto Nivel.

Incontables veces, un determinado lenguaje es seleccionado debido al grado de familiaridad que los programadores tengan con él, cuestión que es razonable. Otras veces, es seleccionado aquel lenguaje que se ha convertido en una herramienta de mercado y ha generado una gran publicidad debido a su reciente puesta en marcha o a las grandes promesas de sus creadores.

Actualmente existe una gran cantidad de lenguajes de programación (C, C++, Microsoft Visual Basic, Delphi, C#, Java, entre otros) con diversas características disponibles para que el programador valore cual de ellos es la opción más adecuada al momento de desarrollar una aplicación y que permita un mejor desempeño de sus objetivos.

En la elección del lenguaje a utilizar deben ser analizados, entre otros factores, como el conocimiento que posea el personal en el trabajo, disponibilidades en el mercado, fortalezas y debilidades del mismo para desarrollar una determinada tarea. *“El problema a resolver debería determinar el lenguaje de programación a utilizar”* [Radeck, 2003].

A continuación se planteará una breve comparación entre algunos lenguajes de programación más conocidos. Dicho análisis se realiza con el objetivo de seleccionar la herramienta de construcción más adecuada para la solución del problema.

1.3.1 C / C++

C y C++ tienen un soporte elevado en cualquier plataforma o sistema operativo que el programador utilice. Puede utilizarse para crear aplicaciones de grandes proporciones. Dichos lenguajes son básicos en el desarrollo de cualquier programador, ya que estos

les darán los conocimientos necesarios para poder ampliar sus capacidades y de esta forma hacer uso de lenguajes más avanzados.

“Un programa escrito en C o C++ podrá ser copiado sin dificultad para otra computadora con un entorno diferente, y luego, podrá ser ejecutado con solo algunos cambios” [Oualline, 1995]. La cita anterior hace referencia a una de las características de los lenguajes antes mencionados, la cual permite al programador escribir los programas tan solo una vez y luego poder reutilizarlos en programas diferentes tan solo con algunas modificaciones y ser capaces de ejecutarse en sistemas operativos o plataformas diferentes.

C / C++ *“brindan el poder y la flexibilidad de manipular la memoria y acceder directamente al hardware”* [Oualline, 1995]. Esto no solo incrementa la posibilidad de que un error se encuentre alojado en su código, sino que incrementa también la cantidad de tiempo necesaria para depurar el programa con el objetivo de lograr su correcto funcionamiento.

Muchos de los programas comerciales para Linux, Windows y Macintosh, entre otros, son escritos en C / C++ para aprovechar la velocidad, eficiencia y portabilidad de éstos; además estos lenguajes tienen la ventaja de contar con una inmensa comunidad de programadores alrededor del mundo entero. Con tantas personas aprendiendo y programando, un programador no tendrá dificultad alguna para encontrar ayuda y recursos disponibles en Internet.

A pesar de todas las ventajas mencionadas un gran número de programadores actualmente dan la espalda a estos lenguajes debido a la existencia de nuevos lenguajes

que facilitan de manera considerable las tareas y son más atractivos para el programador que los hacen perder popularidad en nuestros días.

1.3.2 Visual Basic.

Visual Basic es uno de los lenguajes de programación más populares para el desarrollo de software. Un programador puede aprender Visual Basic y comenzar a utilizarlo mucho más rápido que si eligiera otro lenguaje de programación. La curva de aprendizaje de Visual Basic es extremadamente pequeña. Mientras más rápido usted pueda aprender un lenguaje, más rápido podrá implementar una solución a su problema. Los programas escritos en Visual Basic tienden a correr mucho más lentamente que los programas equivalentes desarrollados en otros lenguajes de programación como C/C++. Además, tienden a consumir grandes cantidades de espacio en disco aun por aplicaciones bien simples.

Según [Smith, 1998], “*Visual Basic es fácil de aprender debido a que aísla al programador de los detalles técnicos de la programación de una computadora*”, pero a la misma vez, esto imposibilita que un programador pueda tomar el control del ordenador, lo cual trae consigo una limitación en el poder de las aplicaciones.

Por último, y no lo menos importante, Visual Basic corre solamente sobre la familia de sistemas operativos Windows, lo cual significa que un programa escrito en Visual Basic nunca correrá en otro tipo de sistema operativo. Esto significa una gran desventaja en cuanto a portabilidad.

1.3.3 Java.

Java es uno de los lenguajes que ha ganado renombre en la programación moderna. Al igual que C y C++, cuenta con una inmensa comunidad de programadores alrededor del mundo. Con Java pueden ser creadas poderosas aplicaciones, sitios web o pequeños programas, conocidos como *applets*, que pueden ser corridos en Internet.

Dicho lenguaje de programación es completamente portátil, por lo que *“un programa escrito en Java podrá, teóricamente, correr en cualquier computadora o sistema operativo sin que sea necesaria modificación alguna”* [Deitel, 2002].

Este lenguaje utiliza un mecanismo llamado recolector de basura (*garbage collector*) para eliminar los objetos creados por el programador en la memoria de la computadora. Esto trae consigo que disminuyan considerablemente los desbordes de memoria, como en C y C++, y que el código sea mucho más limpio y fácil de escribir.

De la misma forma que C y C++, Java se encuentra diseñado con el objetivo de brindar una plataforma de libre distribución para la creación de aplicaciones. Todo lo contrario a Visual Basic, por el que hay que pagar determinada cantidad de dinero para su adquisición. En Internet puede encontrarse un gran número de entornos de programación libre de impuestos para el desarrollo de programas en Java.

A pesar de sus grandes ventajas, Java también cuenta con un número de desventajas que deberán ser analizadas en el momento de tomar una decisión. Por ejemplo, los programas en Java tienden a ser mucho más lentos que sus equivalentes escritos en otros lenguajes de programación.

1.3.4 C#

El lenguaje de programación C#, es uno de los últimos lenguajes utilizados en el mercado; la base de este lenguaje ha sido creada bajo la concepción de C y C++. Cuenta con un compilador y herramientas necesarias de forma gratuita, su plataforma .NET brinda un entorno visual para la escritura que resulta ser más atrayente, además se integra de manera correcta con el desarrollo de aplicaciones Web XML y muchas de las tecnologías emergentes, así mismo uno sus objetivos está el de ser portable a cualquier sistema operativo, ya que aprovecha el nuevo entorno .NET FRAMEWORK, eliminando la dificultad de trabajar con punteros como sus predecesores.

Posee un entorno común de ejecución (CLR), un conjunto de bibliotecas de clases lo cual establece una diferencia con otros productos similares, *“C# es un lenguaje completamente orientado a objetos con una gran cantidad de características y mejoras sobre sus predecesores”* [Radeck, 2003]. Al igual que Java, trabaja en un entorno manipulado de memoria, aislando al programador de los engorrosos detalles del hardware. A pesar de esto, *“el programador tendrá la posibilidad de renunciar a esta característica y lidiar por sí mismo con la manipulación de la memoria”* [Robinson, 2001].

Considerando el poco tiempo de ser utilizado dentro del mercado C#, aún no ha consolidado una comunidad de programadores lo suficientemente poderosa, pero hoy en día gracias a la lluvia de información a la que se puede acceder en el Internet, próximamente podremos contar con ella.

A causa de la gran cantidad de características adicionales que presenta el C#, resulta un lenguaje complejo de aprender, pero nunca tan complejo como resulta el C y C++. No obstante, debido a las similitudes en el código, programadores de Java y C++ encontrarán muy fácil esta tarea.

1.4 Selección del Leguaje de Programación a utilizar.

Hasta el momento, se han analizado las características fundamentales de los lenguajes de programación candidatos para la implementación del objetivo propuesto en este trabajo. Para la selección del lenguaje más apropiado, es necesario realizar una comparación teniendo en cuenta un grupo de características que influirán directamente en el ambiente donde se desea realizar la implementación propuesta.

A partir del análisis realizado anteriormente, se seleccionó el siguiente conjunto de criterios para realizar la evaluación final y con ésta decidir el lenguaje a utilizar para la realización de este trabajo:

Características Multiplataformas: En este punto podemos encontrar que el Java y el C/C++ son los dos únicos candidatos, dado que Visual Basic y C# son solamente soportados en plataforma Windows.

Curva de Aprendizaje: La curva de aprendizaje de los lenguajes analizados puede ser organizada del más simple al más difícil de la siguiente forma: Visual Basic, C#, Java, C/C++.

Velocidad de ejecución: La velocidad de ejecución es mayor en los lenguajes C y C++, seguido por C#, Visual Basic y por último Java, el cual es el más lento de todos.

Disponibilidad de Recursos: En cuanto a la disponibilidad de recursos en Internet, existe una gran competencia entre los lenguajes C, C++ y Java, siendo estos los mayormente utilizados para la creación y publicación de artículos, códigos de ejemplo y bibliotecas de funcionalidad. C# y Visual Basic son menos utilizados, por lo que existe una menor cantidad de recursos en Internet.

Comunidad: Al igual que la disponibilidad de recursos, Java, C y C++ son los lenguajes de mayor comunidad de programadores. Visual Basic tiene una comunidad bien establecida que ha tendido a disminuir con la aparición de la plataforma .NET, todo lo contrario a la comunidad de C# que ha ido solidificándose e incrementándose cada vez más.

Familiarización con el Lenguaje: El grupo de desarrollo encargado de realizar la implementación propuesta está familiarizado con los siguientes lenguajes en orden de prioridad: C, C++ y Visual Basic, resultando este último prácticamente desconocido para sus miembros.

Luego de realizado esta comparación, podemos concluir que, dada las características analizadas, los lenguajes C# y Java, se encuentran favorecidos en una gran parte de los aspectos tratados. Por otra parte la plataforma utilizada en la Universidad de Camagüey

es Windows, y la plataforma .NET esta preparada para ser 100% compatible con este sistema operativo por lo que las aplicaciones confeccionadas en la misma pueden hacer un uso eficiente de los recursos que presenten las maquinas donde estén corriendo.

Debido a que la aplicación estará corriendo en plataforma Windows se decidió que la implementación de la misma debería ser en algún lenguaje que se adaptase lo mejor posible a las características de este sistema operativo, teniendo en cuenta este aspecto quedan tres lenguajes que pueden ser utilizados en la confección de la misma, estos lenguajes son: C++, Visual Basic y C#. Hay que tener en cuenta que Microsoft dejo de producir Visual Basic a partir del surgimiento del .NET, por lo que no es aconsejable iniciar la producción de un software en este lenguaje, ya que no se sabe a ciencia cierta cual será el futuro de éste, además el lenguaje C++ presenta una curva de aprendizaje extremadamente elevada y la herramienta para desarrollo en este lenguaje con que se cuenta (Microsoft Visual C++) es poco productiva, lo que puede dar al traste con los tiempos en los cuales se exige la entrega de una aplicación estable, y que cumpla con las normas de calidad requeridas.

Teniendo en cuenta todo esto, se seleccionó el lenguaje C# para la confección de la aplicación dado a que es un lenguaje que genera códigos que se integran perfectamente la plataforma Windows, es un sistema de una alta productividad y además no presenta una curva de aprendizaje tan elevada como el C++ que fue uno de los candidatos más cercanos para ser elegidos.

1.5. Sistemas Gestores de Bases de Datos (SGBD).

Tomando como referencia que un SGBD *“es un conjunto de programas, procedimientos, lenguajes que permiten la implantación, acceso y mantenimiento de la base de datos; los mismos que suministran a los distintos tipos de usuarios los medios necesarios para describir, manipular, recuperar o actualizar los datos almacenados en la base”* [Castaño y Piattini, 2003]; facilitando el mantenimiento de la seguridad (confidencialidad, disponibilidad e integridad) del conjunto de datos. Los SGBD *“admiten varios lenguajes para la manipulación de datos como: Cobol, Ensamblador, Fortran, PL/I, Pascal, C, etc. y además los lenguajes de cuarta generación como el SQL”* [Alegre Gutiérrez, 2003].

A continuación se planteará una breve comparación entre algunos gestores de base de datos más conocidos. Dicho análisis se realiza con el objetivo de seleccionar la herramienta de construcción más adecuada para la solución del problema.

1.5.1 Microsoft Access.

Con Access se puede crear una aplicación multiusuario haciendo uso de la facilidad que brinda el mismo de trabajar compartiendo la base de datos en un servidor de ficheros. Esta aplicación basada en un servidor de ficheros puede ser una solución tradicional basada en formularios e informes, una solución basada en Web o una combinación de ambas. En el mismo se pueden definir reglas de integridad y la creación de un programa con el mismo se hace de una forma muy sencilla y amigable.

A pesar de sus grandes ventajas, Access también cuenta con un número de desventajas que deberán ser analizadas en el momento de tomar una decisión. Por ejemplo, los programas en Access tienden a ser mucho más lentos, y a ser utilizados en el desarrollo de aplicaciones que no exijan mayor complejidad ya que no cuenta con componentes especializados, también las aplicaciones multiusuarios creadas en este sistema de gestión tienden a no ser seguras ya que los usuarios, para poder hacer uso de las mismas, tendrán que acceder directamente a el fichero del proyecto, lo cual podría ser la causa de una violación de seguridad en el caso de que haya usuarios inexpertos, o usuarios mal intencionados.

1.5.2. Oracle.

Oracle es en estos momentos uno de los sistemas de gestión mas robustos que existe, este servidor de base de datos cuenta una de las implementaciones mas completas de todas las características que puede tener una base de datos hoy en día, el mismo cuenta con los aspectos tradicionales que debe tener un servidor de bases de datos hoy en día, tales como restricciones de integridad, procedimientos almacenados, triggers, reglas, etc, aparte de eso presenta muchas características implementadas para la gestión de almacenes de datos.

Oracle trabaja sobre la arquitectura cliente – servidor, en donde no se guarda la información en la parte del cliente ya que quien se encarga de esta función es el servidor por medio de tablas. A pesar de que Oracle es uno de los sistemas de gestión mas completos que existe, también presenta desventajas sobre todo en sistemas de carácter pequeño como el que se quiere implementar en este trabajo, este sistema de gestión

necesita de maquinas potentes en el lado del servidor y por ende la implementación de un sistema haciendo uso de mismo trae consigo grandes gastos en lo que a equipamiento se refiere.

1.5.3. My SQL Server.

My SQL Server es el mas popular Open Source servidor de bases de datos que soporta un gran número de librerías, programas clientes, herramientas administrativas, y una gran cantidad de aplicaciones programadas interfaces (APIs) por medio de librerías multi- tareas en la cuales se puede desarrollar proyectos de mediana y considerable amplitud. Open Source significa que cualquier tipo de usuario puede hacer uso de él, sin costo alguno, evitando la necesidad de poseer una licencia comercial.

El manejo de los sistemas de las base de datos se cumple mediante el almacenamiento de los datos en tablas separadas, su manejo es sumamente rápido y fácil de usar. Su velocidad, flexibilidad, conectividad y seguridad lo hacen verdaderamente útil para aplicaciones en el Internet., debido a que la manipulación de las bases de datos es superior a otros gestores comerciales existentes.

1.5.4. Microsoft SQL Server.

Microsoft SQL Server trabaja sobre la arquitectura cliente – servidor, en donde no se guarda la información en la parte del cliente ya que quien se encarga de esta función es el servidor por medio de tablas, todo este proceso lo realiza mediante la utilización de componentes como: el LDD Lenguaje de definición de datos que se encarga de la

creación de todos los objetos y datos que se utilizan normalmente (tablas, vistas, restricciones, etc.) así como las directivas de seguridad y el LMD Lenguaje de manipulación de datos el cual esta orientado principalmente a las consultas, inserción, modificación, eliminación y actualización de los datos.

SQL Server es un sistema de gestión robusto y presenta la mayoría de las características que debe tener un servidor de base de datos en la actualidad. Además de esto, este sistema de gestión esta preparado para correr de forma optima en ambientes Windows y presenta un conjunto de bibliotecas nativas en las herramientas de Microsoft para el acceso a los datos y objetos definidos en el este, por lo que las aplicaciones desarrolladas con las herramientas de Microsoft logran altos desempeños y una gran compatibilidad entre sí logrando herramientas robustas y con un alto grado de seguridad y estabilidad.

1.6. Selección del Sistema Gestor de Base de Datos.

Luego de realizar el análisis de los diferentes (SGBD), fundamentado en las características de mayor relevancia, se ha seleccionado a Microsoft SQL Server, para que sea una de las herramientas que formaran parte en el desarrollo de la aplicación, debido a que este maneja la concepción del trabajo cliente – servidor, con un gran dominio de base de datos relacionales, su plataforma brinda al programador diversas características que él puede aprovechar para el desarrollo de sus aplicaciones de modo rápido, confiable, robusto, seguro y fácil de usar tanto para volúmenes de datos grandes como pequeños. Entre una de sus funciones es permitir el trabajo en red y que varios usuarios puedan acceder al sistema a ala vez (conurrencia). Posibilita también a que otras aplicaciones puedan hacer uso de la información ya existente, algo que constituye

un elemento de gran importancia para el desarrollo informático, ya que no es necesario tener que procesar nuevamente la información.

Uno de los principales procesos utilizados en el presente trabajo, es la utilización de procedimientos almacenados que son funciones y procesos por lotes de software pre-compilado, que el gestor de datos almacena, procesa y ejecuta dentro de su misma fuente, sin tener que transferir los datos a través de la red, dicha función se la realiza por medio de la llamada de su nombre, su creación y utilidad beneficia la:

- Funcionalidad que es la extracción de información útil del servidor incluso si no se conoce la estructura de las tablas de las que se extrae.
- Mejora del rendimiento de red que evita la sobrecarga en el momento que el usuario ejecute un procedimiento.
- Seguridad e integridad de los datos.

Resumen.

En el presente capítulo se realizó un análisis de las tecnologías que serán utilizadas para la construcción del sistema propuesto. De la misma forma se definieron conceptos básicos para un mayor entendimiento del resto del trabajo.

Se estudio los diferentes conceptos de modelos de confección de aplicaciones, tales como el modelo cliente-servidor, y modelos de programación por capas. Además el concepto de la metodología ADOOSI Visual; los mismos que se han definido como los mas apropiados a usar.

Se realizó un análisis comparativo entre los diferentes lenguajes de programación C/C++, Visual Basic, Java y C# llegándose a la conclusión de que el más apropiado para utilizar es el C#. Otro de los aspectos analizados y comparados fue los sistemas gestores de bases de datos dentro de los cuales se tuvieron en cuenta Microsoft Access, Oracle, My SQL Server y SQL Server, llegando a la conclusión que SQL Server es el mas apropiado.

Capítulo II. Fundamentación de la Propuesta.

En el capítulo anterior se realizó un análisis de las posibles herramientas para la creación del sistema automatizado para el control de la actividad de postgrado. Además se analizaron las ventajas que proporcionarían las herramientas seleccionadas para la creación de la aplicación así como también las limitaciones existentes en aquellas aplicaciones que manejan los conceptos abordados en este trabajo.

Dentro de este capítulo se explicará de manera clara el desarrollo del sistema propuesto, las dificultades que posee el centro actualmente, la definición del problema, la propuesta del sistema a implementarse, el objeto de automatización, y los requisitos funcionales y no funcionales. A partir de estas especificaciones, podrá ser implementado el sistema propuesto, teniendo en cuenta cada una de las características necesarias para el cumplimiento de su cometido.

2.1. Descripción del Objeto de Estudio.

2.1.1. Caracterización del Problema a Resolver.

Las facultades de la Universidad de Camagüey no cuentan con un sistema automatizado para el proceso de control de la actividad de Postgrado, esta situación provoca que la elaboración de las certificaciones emitidas para los postgrados y diplomados sean más propensas a errores asociados con el retraso en la entrega de las mismas, ya que el encargado de esta actividad tiene que llenar distintos tipos de modelos como: Certificado de Postgrado, Reconocimiento, Hago Constar y el Certificado de Diplomado por cada postgrado o diplomado que se imparta, modelos que

posteriormente deben enviarse al CEPI, entre otras informaciones.

Además dicho proceso carece de un control estricto sobre la creación de las certificaciones de manera legal lo que lo hace vulnerable a falsificaciones de los documentos emitidos. Los errores en el proceso mencionado son de suma importancia pues los resultados de la información recibida se ven reflejados en los reportes estadísticos que son tomados en cuenta para la proyección del siguiente curso académico. Todas estas dificultades analizadas originan imprecisión en el control que realiza el centro, perjudicando la visión de la universidad en este aspecto. Todas estas dificultades en el tratamiento de la información podrán evitarse a partir de la implementación del sistema que en el presente trabajo se propone.

2.2. Objeto de Automatización.

Una de las principales tareas del país hoy en día es la Universalización de la educación posgraduada, que significa llevar a cada comunidad la superación de postgrado para satisfacer las necesidades propias a través de los profesionales del territorio. Este proceso conseguirá multiplicar los niveles de profesionales superados en pocos años, así como la variedad y dinámica de las ofertas de superación profesional; esto exige de la automatización de la gestión en cuanto a tramitación de certificaciones de postgrado se refiere.

Las formas de la educación de postgrado en Cuba son: la superación profesional y la formación académica. Las modalidades fundamentales de la superación profesional están las conferencias, seminarios, cursos de postgrado, entrenamientos y diplomados. Por su parte la formación académica se logra a través de Especialidades, Maestrías y

Doctorados.

Las facultades a las que se ofertan estas dos formas de educación, dentro de la Universidad de Camagüey, son las siguientes: Ciencias Técnicas (Ingeniería Química, Mecánica, Eléctrica, Civil, Arquitectura, Informática y Farmacia), Ciencias Agropecuarias (Agronomía y Medicina Veterinaria), Ciencias Económicas (Contabilidad y Economía) y Ciencias Sociales y Humanísticas (Derecho, Estudios Socioculturales, Historia, Psicología, Comunicación Social e Información Científico-Técnica).

Una parte importante de la gestión del centro se dedica a la tramitación de las certificaciones de actividades de superación profesional (cursos de postgrado, entrenamientos y diplomados) por ser esta, la vía fundamental a través de la cual se superan cada año más de 11 mil profesionales.

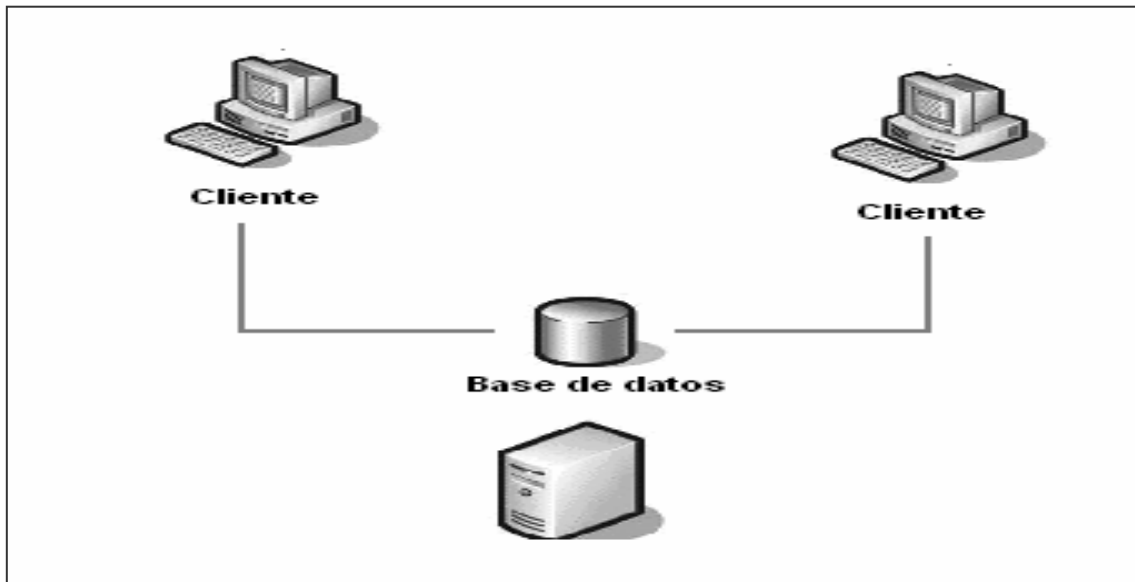
Cada facultad tiene que hacer llegar, a través del responsable designado, a la secretaría del CEPI, el modelo M-1 establecido como registro primario para la tramitación de la certificación (Anexo I). El modelo se acompaña de los certificados individuales de los profesionales (participantes y profesores) que formaron parte de la modalidad. En este modelo se controlan los datos generales de los cursos, los participantes, profesores, los módulos de los cursos tipo diplomado, la evaluación que se le asigna a cada participante de un curso y la cantidad de horas que imparte un profesor en un curso; además el responsable de cada una de esas facultades es el que tiene que controlar la entrega de certificados.

2.3. Propuesta del Sistema.

El sistema propuesto debe contar con una interfaz amigable que será utilizada por los usuarios autorizados para la manipulación de la información, de forma tal, que cumpla con los requisitos, necesidades y funciones que realiza el CEPI.

El sistema será confeccionado utilizando la arquitectura cliente servidor de dos capas. Para la confección del servidor de base de datos se utilizará el Microsoft SQL Server. Para la creación de la interfaz visual de la aplicación (aplicación cliente) se utilizo la herramienta Visual Studio .NET utilizando el lenguaje de programación C#.

A La arquitectura en la parte cliente de la aplicación estará compuesta por 5 capas utilizando la arquitectura propuesta por Microsoft, estas 5 capas son UI, Facade, DataAccess, Framework y Common, cada una de las cuales cumple con un rol dentro de la aplicación.



Arquitectura Propuesta en el Servidor

UI (Interfaz de Usuario): Implementa la lógica de comunicación de la aplicación con los usuarios del sistema e interactúa entre la capa facade y el usuario.

Facade: Como su nombre lo indica es una fachada entre la interfaz de usuario y la lógica de la aplicación, la misma permite aislar la interfaz de usuario de las reglas de negocio definidas en el sistema, esto trae como ventaja que si el día de mañana se desea implementar el sistema utilizando una interfaz distinta (Ej: Web), no habrá que reprogramar el sistema completo sino que solamente habrá que implementar la lógica de comunicación con los usuarios del sistema. Es en donde se desarrolla la lógica de la aplicación o las reglas del negocio (operaciones) es el enlace entre el UI y el Data Access.

Manager: Es la lógica de acceso a los datos (Data Access); es decir es una acción sobre la base de datos mediante procedimientos almacenados.

Framework: El framework define la infraestructura básica de la aplicación, en el mismo está implementado el acceso a los ficheros de configuración y las reglas generales de control de excepciones. En el cual se implemento un mecanismo de sistema de traza Postgrado.Log el cual permite tirar a un fichero los errores que genera la aplicación. Dentro de los archivos de configuración tenemos el fichero App. Config en el cual se especifica los detalles de la conexión con la base de datos.

Common: Son las partes comunes a todas las capas de la aplicación, en la misma están definidas las clases comunes del sistema los cuales son conjuntos de datos (DataSet) y las constantes que son utilizadas por todas las capas.

Además la arquitectura implementada presenta varias innovaciones e implementaciones de última generación generando una aplicación de óptima calidad, dentro de estas innovaciones podemos describir que posee:

Fichero de Recursos: Es en donde se especifican todos los mensajes, advertencia que genera la aplicación y el mismo soporta la norma de Internacionalización con respecto al formato Standard I. 18N, esto significa que si el día de mañana se desea implementar el sistema en cualquier otro idioma, tan solo se requerirá modificar este archivo.

Fichero de Constantes: Es un archivo en donde se almacenan todas las declaraciones de las constantes que serán utilizadas en la aplicación.

2.4. Requisitos de la Aplicación.

Los requerimientos definen necesidades que requiere el cliente del sistema que se este confeccionando.

2.4.1 Definición de Requerimientos Funcionales.

Después de haber analizado el panorama general, las necesidades del cliente y objetivos propuestos, se han identificado los siguientes requisitos funcionales para el desarrollo del sistema:

1. Establecer conexión con la base de datos.
2. Autenticación de usuarios de la aplicación.
3. Ingresar los datos generales para el registro de un nuevo Usuario.
4. Eliminar registros de los Usuarios.
5. Cambio de contraseña de Usuarios.
6. Cambio de sesión de Usuarios.
7. Ingresar los datos generales para el registro de una nueva Facultad.
8. Modificar el nombre de la Facultad.
9. Eliminar registros de Facultades.
10. Ingresar los datos generales para el registro de un nuevo Profesor.
11. Modificar datos generales de un Profesor.
12. Eliminar un Profesor de la base de datos.
13. Adicionar a Participantes.
14. Modificar datos generales de un Participantes.

15. Eliminar Participantes.
16. Adicionar postgrados
17. Modificar postgrados
18. Eliminar un postgrados
19. Definir participantes de un postgrado
20. Modificar datos de participantes de un postgrado
21. Eliminar datos de participantes de un postgrado
22. Definir evaluación de cada uno de los participantes de un postgrado
23. Definir el número de horas que impartirá un profesor en un postgrado.
24. Eliminar el número de horas impartidas por un profesor en un postgrado.
25. Asentar en libros la información de participantes evaluados de un postgrado.
26. Adicionar los datos de un Diplomado
27. Modificar los datos de un Diplomado
28. Eliminar un Diplomado
29. Definir participantes de un diplomado
30. Modificar datos de participantes de un diplomado
31. Eliminar datos de participantes de un diplomado
32. Definir los datos de un postgrado que pertenezca a un diplomado.
33. Modificar datos de un postgrado que pertenezca a un diplomado.
34. Eliminar un postgrado que pertenezca a un diplomado.
35. Definir evaluación de cada uno de los participantes de un diplomado.

36. Definir el número de horas que impartirá un profesor en un postgrado de un diplomado.
37. Eliminar el número de horas impartidas por un profesor en un postgrado de un diplomado.
38. Asentar en libros la información de participantes evaluados de un diplomado.
39. Generar certificados de Postgrado.
40. Generar certificados de Diplomados.
41. Generar certificados de Reconocimiento.
42. Generar certificados de Hago Constar.
43. Generar reportes de Asentamiento en Libros.

2.4.2 Definición de Requerimientos no Funcionales.

Las características, propiedades o cualidades de un producto de software conforman los requerimientos no funcionales, los mismos que determinan los detalles y restricciones que de una u otra forma puedan limitar al sistema, en la construcción del sistema se detectaron los siguientes requisitos funcionales:

1. Apariencia o interfaz externa: El sistema debe tener una interfaz amigable, legible, interactiva, fácil de usar, clara y sencilla.

2. Usabilidad: El sistema está concebido para ser usado por el encargado de llevar el control de la actividad de superación profesional en las dependencias de la Universidad

de Camagüey, por lo que es necesario que cuente con un diseño de interfaz fácil dado que todas estas personas no tienen porqué tener gran experiencia en el manejo de computadoras.

3. Requerimientos de Seguridad: Debe tener confidencialidad, integridad y disponibilidad.

3.1 Confidencialidad: La información manejada por el sistema deberá estar protegida de acceso no autorizado y divulgación. Los usuarios accederán a la información correspondiente cada uno.

3.2 Integridad: La información manejada por el sistema será objeto de cuidadosa protección contra estados inconsistentes.

3.3 Disponibilidad: A los usuarios autorizados se les deberá garantizar el acceso a la información solicitada en todo momento.

4. Requerimientos de Rendimiento: Debe ser eficiente, rápido y preciso. El tiempo de respuesta no debe ser mayor de 10 segundos.

5. Portabilidad: El producto será usado en plataforma Windows 98 o superior.

6. Requerimientos de Confiabilidad: El sistema debe estar disponible todo el tiempo para el personal autorizado. Las salidas del sistema tienen que ser 100% veraces. Y debe ser capaz de recuperarse rápidamente de fallas.

7. Legales: El software debe cumplir con las leyes vigentes.

8. Soporte: Debe ser de fácil instalación, adaptable a numerosas plataformas y de fácil mantenimiento.

9. Ayuda y documentación en línea: Elaborar una ayuda sencilla donde se explique el funcionamiento del software.

10. Plataformas del Sistema Operativo:

Cliente: Microsoft Windows 98 o superior.

Servidor: Microsoft Windows 2000 o superior.

11. Software Adicional:

Tanto para los clientes, como para el servidor.

SQL-Server 2000.

Microsoft Visual Estudio.NET 2003

12. Hardware:

CPU Pentium a 100MHz o superior.

32 MBytes RAM o superior.

13. Tiempo de respuesta: El sistema concibe una restricción de frontera para emitir los reportes de un tiempo estimado de 10 segundos.

Resumen.

Dentro de este capítulo se ha realizado un análisis acerca de la propuesta para dar solución al problema planteado a lo largo de este trabajo. De la misma forma, han sido analizados un grupo de conceptos importantes alrededor del dominio que abarca la implementación del sistema para el control de actividad de postgrado.

Se realizó un análisis del objeto de estudio, objeto de automatización y de la propuesta del sistema. Como resultado de este análisis se tiene que el sistema propuesto contara con una arquitectura cliente-servidor de dos capas. Para la confección del servidor de base de datos se utilizará el Microsoft SQL Server. La arquitectura en la parte cliente de la aplicación estará compuesta por 5 capas utilizando la arquitectura propuesta por Microsoft, estas 5 capas son: UI, Facade, DataAccess, Framework y Common, cada una de las cuales cumple con un rol específico dentro de la aplicación.

Para finalizar este capítulo fueron enumerados cada uno de los requisitos funcionales y no funcionales necesarios para llevar a cabo con éxito el producto final propuesto en este trabajo.

Capítulo III. Análisis y Diseño General del Sistema.

En los capítulos anteriores fue brindada una idea clara de los conceptos involucrados y definiciones de las herramientas seleccionadas para dar cumplimiento al objetivo deseado.

En este capítulo será analizado y definido el sistema propuesto para su posterior implementación. Además de la metodología ADOOSI – UML [Álvarez, 2000], que se utilizó en el desarrollo de la aplicación; aquí abordaremos aquellos aspectos que se consideraron imprescindibles para lograr una buena comprensión de la estructura del sistema.

Para mostrar las funciones del sistema, sus entornos y las relaciones existentes entre ambos, serán definidos y documentados cada uno de los actores, casos de uso encontrados, modelo conceptual, diseño físico de la base de datos. De la misma forma, serán representados los diagramas de secuencia y de clases, completando el diseño global de la arquitectura propuesta.

Por último serán identificados y analizados algunos de los patrones utilizados en el desarrollo de la aplicación. Cada uno de los patrones detectados serán debidamente documentados en esta sección.

3.1 Definición de Casos de Uso.

El comportamiento de un sistema en desarrollo es documentado en un diagrama de Casos de Uso que ilustrará las funciones del sistema (Casos de Uso), sus entornos (Actores) y las relaciones entre los Casos de Uso y los Actores (Diagramas de Caso de Uso). “*Un Caso de Uso es una secuencia de acciones que obtienen resultados de valor para un Actor, y un Actor representa cualquier cosa que interactúa con el sistema que puede ser un humano, software o hardware*” [Álvarez, 2000]. El rol más importante de un modelo de Casos de Uso es el de la comunicación. “*Un modelo de Casos de uso provee un vehículo usado por los clientes o usuarios finales y desarrolladores para discutir el comportamiento y la funcionalidad del sistema*” [Quatrani, 2000].

A continuación, serán identificados y definidos cada uno de los Actores que participan en el sistema, los Casos de Uso involucrados, y las relaciones existentes entre ambos. Finalmente, todo el modelo será resumido a modo de esquema haciendo uso del UML a través de un Diagrama de Casos de Uso que podrá ser encontrado en el Anexo II.

3.1.1. Especificación de Casos de Uso existentes en el Sistema.

En el análisis realizado del sistema propuesto fueron identificados los Casos de Uso que a continuación son enumerados:

1. Autenticar el ingreso de usuarios del Sistema.
2. Manipular Usuarios.
3. Manipular Facultades.
4. Manipular Profesores.
5. Manipular Participantes.
6. Manipular Postgrados.
7. Manipular Diplomados.
8. Generar Certificados de Postgrado.
9. Generar Certificados de Diplomados.
10. Generar Certificados de Reconocimiento.
11. Generar Certificados Hago Constar
12. Generar Reportes de Asentamiento en Libros.

13. A continuación, serán descritos cada uno de estos Casos de Uso.

Caso de Uso	Autenticación de Usuarios
Actores	Usuario
Objetivos	Es un conjunto de características que tiene como objetivo el acceso de un usuario autorizado, hacer uso del manejo de la información que contiene el sistema.
Precondiciones	Ser usuario del sistema, registrado anteriormente en la base de datos.
Flujo Principal	El caso de uso iniciará cuando se reciba una solicitud del sistema cliente para autenticar al usuario (E1), esta requerirá el login y contraseña. En cada caso, el flujo correspondiente será ejecutado.
Flujos Alternativos	Ninguno.
Excepciones	E1: Si el nombre de usuario y la contraseña no coinciden para la autenticación, el sistema elevará una excepción y el caso de uso finalizará su ejecución.
Referencias	1, 2

Caso de Uso	Manipular Usuarios
Actores	Usuario
Objetivos	Es un conjunto de características que tiene como objetivos la inserción, cambio de contraseña y eliminación de usuarios del sistema.

Precondiciones	Ninguna
Flujo Principal	El caso de uso iniciará cuando se reciba una solicitud del sistema cliente para la inserción, modificación de contraseña y eliminación de uno o varios usuarios. En cada caso, el flujo alternativo correspondiente será ejecutado.
Flujos Alternativos	S1: Insertar
	El sistema cliente especificará el nombre, primer apellido, segundo apellido, nombre de usuario y contraseña del nuevo usuario que se desea insertar. El nuevo objeto será creado y almacenado en el medio correspondiente (E1). El caso de uso terminará su ejecución.
	S2: Cambiar Contraseña
	El sistema cliente especificará la contraseña actual, la nueva contraseña y la confirmación de la misma. El sistema actualizará la información del objeto correspondiente (E2 , E3) y culminará la ejecución del caso de uso.
	S3: Eliminar
	El sistema cliente seleccionará el o los usuarios que se desea eliminar. El sistema verificará la existencia del objeto. En caso de que exista, será eliminado y se ejecutará el flujo principal alternativo. En caso de no existir, no se ejecutará acción y terminará el caso de uso.
Excepciones	E1: Si en el medio de almacenamiento correspondiente existe un usuario con el mismo nombre de usuario, se elevará una excepción y el caso de uso finalizará su

	ejecución.
	E2: Si la contraseña actual no coincide con la contraseña anteriormente almacenada se elevara una excepción y el caso de uso finalizará su ejecución.
	E3: La nueva contraseña ingresada no coincide con la confirmación de la misma, se elevara una excepción y el caso de uso finalizará su ejecución
Referencias	3, 4, 5, 6

Caso de Uso	Manipular Facultades
Actores	Usuario
Objetivos	Es un conjunto de características que tiene como objetivos la inserción, modificación y eliminación de facultades del sistema.
Precondiciones	Ninguna
Flujo Principal	El caso de uso iniciará cuando se reciba una solicitud del sistema cliente para insertar, modificar y eliminar una o varias facultades. En cada caso, el flujo alternativo correspondiente será ejecutado.
Flujos Alternativos	S1: Insertar
	El sistema cliente especificará el nombre de la nueva facultad que se desea insertar. El nuevo objeto será creado y almacenado en el medio correspondiente (E1). El caso de

	uso terminará su ejecución.,
	S2: Modificar
	El sistema cliente seleccionara la facultad que se desea modificar sus datos generales. El sistema actualizará la información de la facultad correspondiente y culminará la ejecución del caso de uso.
	S3: Eliminar
	El sistema cliente seleccionara las facultades que se desea eliminar.
Excepciones	E1: Si en el medio de almacenamiento correspondiente no existe el nombre de la facultad especificado y existe una facultad con el mismo nombre, se elevará una excepción y el caso de uso terminará su ejecución.
Referencias	7, 8, 9

Caso de Uso	Manipular Profesores
Actores	Usuario
Objetivos	Es un conjunto de características que tiene como objetivos la inserción, modificación y eliminación de profesores del sistema.
Precondiciones	Ninguna
Flujo Principal	El caso de uso iniciará cuando se reciba una solicitud del sistema cliente para insertar, modificar y eliminar uno o varios profesores. En cada caso, el flujo alternativo

	correspondiente será ejecutado.
Flujos Alternativos	S1: Insertar
	El sistema cliente especificará el número de identificación, el nombre, primer apellido, segundo apellido, categoría docente, categoría científica, sexo, trabajador del Centro de Educación Superior (CES) del nuevo profesor que se desea insertar. El nuevo objeto será creado y almacenado en el medio correspondiente (E1). El caso de uso terminará su ejecución.,
	S2: Modificar
	El sistema cliente seleccionara el profesor que se desea modificar sus datos generales. El sistema actualizará la información del profesor correspondiente y culminará la ejecución del caso de uso.
	S3: Eliminar
	El sistema cliente seleccionara el o los profesores que se desea eliminar. El sistema verificará la existencia del objeto. En caso de que exista, será eliminado y se ejecutará el flujo alternativo (E2). En caso de no existir, no se ejecutará ninguna acción y terminará el caso de uso.
Excepciones	E1: Si en el medio de almacenamiento existe un profesor con el mismo número de identidad, se elevará una excepción y el caso de uso terminará su ejecución.
Referencias	10, 11, 12

Caso de Uso	Manipular Participantes
Actores	Usuario
Objetivos	Es un conjunto de características que tiene como objetivos la inserción, modificación y eliminación de participantes nacionales y extranjeros dentro del sistema.
Precondiciones	Ninguno.
Flujo Principal	El caso de uso iniciará cuando se reciba una solicitud del sistema cliente para insertar, modificar y eliminar uno o varios participantes. En cada caso, el flujo alternativo correspondiente será ejecutado.
Flujos Alternativos	S1: Insertar
	El sistema cliente seleccionara el tipo de participante que se desea ingresar con sus datos generales (E1, E2, E3, E4). El nuevo objeto será creado y almacenado en el medio correspondiente. El caso de uso terminará su ejecución.
	S2: Modificar
	El sistema cliente seleccionara el participante que se desea modificar sus datos generales. El sistema actualizará la información del participante correspondiente y culminará la ejecución del caso de uso.
	S3: Eliminar
	El sistema cliente seleccionara el o los participantes que se desea eliminar. El sistema verificará la existencia del objeto. En caso de que exista, será eliminado y se ejecutará el flujo alternativo. En caso de no existir, no se ejecutará ninguna

	acción y terminará el caso de uso.
Excepciones	E1: Si el participante es nacional se requiere los datos de carné de identidad, nombre, primer apellido, segundo apellido, sexo, provincia, municipio, sector, tipo, se realiza su almacenamiento y el caso de uso terminará su ejecución.
	E2: Si el participante es extranjero se requiere los datos de número de pasaporte, nombre, primer apellido, segundo apellido, sexo, país y nivel, se realiza su almacenamiento y el caso de uso terminará su ejecución
	E3: Si el participante es cubano y existe el carné de identidad en el medio de almacenamiento se eleva una excepción.
	E4: Si el participante es extranjero y existe el número de pasaporte en el medio de almacenamiento se eleva una excepción.
Referencias	13, 14, 15

Caso de Uso	Manipular Postgrados.
Actores	Usuario
Objetivos	Es un conjunto de características que tiene como objetivos la inserción, modificación, eliminación, definir participantes, asentar postgrados dentro del sistema.
Precondiciones	El sistema cliente debe definir previamente los participantes para los postgrados.
Flujo Principal	El caso de uso iniciará cuando se reciba una solicitud del sistema cliente para insertar, modificar, eliminar, definir participantes, asentar uno o varios postgrados. En cada caso, el flujo alternativo correspondiente será ejecutado.
Flujos Alternativos	S1: Insertar Postgrado
	El sistema cliente especificará el nombre del curso, la facultad, fecha de inicio, fecha de culminación, número de horas y el tipo de planificación del nuevo postgrado que se desea insertar (E1, E5) y el caso de uso terminará su ejecución.
	S2: Agregar profesor a el postgrado
	El sistema cliente seleccionará el nombre del profesor y especificará la cantidad de horas en dicho postgrado para su ingreso y el caso de uso terminará su ejecución.
	S3: Eliminar profesor del postgrado
	El sistema cliente seleccionará el profesor que se desea eliminar. El sistema verificará la existencia del objeto.

	S4: Modificar Postgrado
	El sistema cliente seleccionara el nombre del postgrado que se desea modificar sus datos generales. El sistema actualizará la información del postgrado correspondiente y culminará la ejecución del caso de uso.
	S5: Eliminar Postgrado
	El cliente seleccionara los postgrados que se desea eliminar. El sistema verificará la existencia del objeto. En caso de que exista, será eliminado. En caso de no existir, no se ejecutará ninguna acción y terminará el caso de uso (E6).
	S6: Definir Participantes
	El sistema cliente definirá todos los participantes que conformarán parte de un postgrado determinado por medio de los flujos alternativos (S7), (S8), (S9), (S10) y terminará el caso de uso.
	S7: Agregar Participantes
	El sistema cliente seleccionará de una lista de participantes inscriptos a aquellos que formarán parte de un postgrado determinado y culminará la ejecución del caso de uso.
	S8: Agregar Nuevo Participante
	El sistema cliente tiene esta opción para usarla cuando un participante no haya sido definido anteriormente; para ello debe ingresar toda la información necesaria y el caso de uso finalizará.
	S9: Quitar Participante

	<p>El sistema cliente seleccionará el participante ya incluido en un postgrado determinado y lo eliminará.</p>
	<p>S10: Definir Evaluación</p>
	<p>El sistema cliente seleccionará el participante al cual se desea registrar la evaluación obtenida en dicho postgrado y el caso de uso finalizará (E4).</p>
	<p>S11: Asentar Postgrado</p>
	<p>El sistema cliente seleccionara el postgrado que desea asentar (E3) y caso de uso finalizará su ejecución.</p>
Excepciones	<p>E1: Si en el medio de almacenamiento correspondiente no existen todos los datos necesarios para crear el nuevo postgrado especificado, se elevará una excepción y el caso de uso terminará su ejecución.</p>
	<p>E2: Si el participante seleccionado existe se elevará una excepción para que el sistema cliente confirme su eliminación y el caso de uso finalizará su ejecución.</p>
	<p>E3: Si el postgrado seleccionado no presenta usuarios evaluados dentro se produce una excepción y el caso se uso termina.</p>
	<p>E4: La evaluación debe ser un número entre 2 y 5.</p>
	<p>E5: La suma de la cantidad de horas impartidas por cada uno de los profesores en un postgrado debe ser mayor que 20</p>

	E6: No se pueden eliminar postgrados que hayan sido asentados en el libro
Referencias	16, 17, 18, 19, 20, 21, 22, 23, 24,25

Caso de Uso	Manipular Diplomados.
Actores	Usuario
Objetivos	Es un conjunto de características que tiene como objetivos la inserción, modificación, eliminación, definición de participantes, definición de los postgrados que componen el diplomado y el asentamiento de diplomados dentro del sistema.
Precondiciones	El sistema cliente debe definir primeramente los postgrados para un determinado diplomado y luego los participantes del mismo.
Flujo Principal	El caso de uso iniciará cuando se reciba una solicitud del sistema cliente para insertar, modificar, eliminar, definir participantes, definir postgrados y asentar uno o varios diplomados. En cada caso, el flujo alternativo correspondiente será ejecutado.
Flujos Alternativos	S1: Insertar Diplomado
	El sistema cliente especificará el nombre del diplomado, fecha de inicio, fecha de culminación y el tipo de planificación para insertar, este nuevo objeto será creado y almacenado en el medio correspondiente (E1, E6) y el caso

	de uso terminará su ejecución.
	S2: Modificar Diplomado
	El sistema cliente seleccionara el nombre del diplomado que se desea modificar. El sistema actualizará la información del diplomado correspondiente (E6) y culminará la ejecución del caso de uso.
	S3: Eliminar Diplomado
	El sistema cliente seleccionara los diplomados que se desea eliminar. El sistema verificará la existencia del objeto. En caso de que exista, será eliminado (E3). En caso de no existir, no se ejecutará ninguna acción y terminará el caso de uso.
	S4: Definir Postgrados
	El sistema cliente definirá todos los postgrados que conformarán parte de un diplomado determinado por medio de los flujos alternativos (S11), (S12), (S13), (S14) y terminará el caso de uso.
	S5: Asentar Diplomado
	El sistema cliente seleccionara el diplomado que desea asentar (E4) y el caso de uso terminará su ejecución.
	S6: Definir Participantes
	El sistema cliente seleccionará el diplomado para el cual se desee definir los participantes, se ejecutará los flujos alternativos (S7), (S8), (S9), (S10) y terminará el caso de

	uso.
	S7: Agregar Participantes
	El sistema cliente seleccionará de una lista de participantes inscriptos todos lo que formarán parte de un diplomado determinado y culminará la ejecución del caso de uso.
	S8: Agregar Nuevo Participante
	El sistema cliente tiene esta opción para usarla cuando un participante no haya sido definido anteriormente; para ello debe ingresar toda la información necesaria, registrarla y el caso de uso finalizará.
	S9: Quitar Participante
	El sistema cliente seleccionará el participante ya incluido en un diplomado determinado y lo eliminará (E3).
	S10: Definir Evaluación
	El sistema cliente seleccionará el participante al cual se desea registrar la evaluación obtenida en dicho diplomado y el caso de uso finalizará (E3, E2).
	S11: Agregar
	El sistema cliente seleccionará el nombre del profesor y especificará la cantidad de horas de dicho curso de postgrado para su ingreso y el caso de uso terminará su ejecución (E3).
	S12: Eliminar
	El sistema cliente seleccionará el o los profesores que se desea eliminar. El sistema verificará la existencia del objeto.

	<p>En caso de que exista, se elevará una excepción para que el sistema cliente confirme su eliminación y el caso de uso finalizará su ejecución (E3).</p>
	<p>S13: Agregar Postgrado</p>
	<p>El sistema cliente seleccionará esta opción después de definir los datos generales para el nuevo postgrado del diplomado seleccionado (E5, E3) y el caso de uso finalizara.</p>
	<p>S14: Eliminar Postgrado</p>
	<p>El sistema cliente seleccionará el o los postgrados que se desea eliminar. El sistema verificará la existencia del objeto.</p> <p>En caso de que exista, se elevará una excepción para que el sistema cliente confirme su eliminación y el caso de uso finalizará su ejecución.</p>
Excepciones	<p>E1: Si en el medio de almacenamiento correspondiente no existen todos los datos necesarios para crear el nuevo diplomado especificado se elevará una excepción y el caso de uso terminará su ejecución.</p>
	<p>E2: La evaluación de un participante debe ser un número entre 2 y 5, en caso contrario se produce un error</p>
	<p>E3: No se pueden eliminar ni modificar diplomados que estén asentados en el libro de registro.</p>
	<p>E4: Si el diplomado no presenta participantes registrados con sus respectivas notas se produce un error</p>
	<p>E5: Si las fechas del postgrado que forma parte de un diplomado se encuentran fuera del marco de las fechas de</p>

	<p>inicio y culminación de un diplomado y la suma de la cantidad de horas de cada postgrado no es igual a mínimo 200 horas el sistema elevará una excepción y el caso de uso culminará su ejecución</p>
	<p>E6: La suma de las horas de los postgrados de un diplomado debe ser superior a 200 horas, en caso contrario se produce un error.</p>
Referencias	26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37,38

Caso de Uso	Generar Certificados de Postgrado
Actores	Usuario
Objetivos	Es un conjunto de operaciones que tiene como objetivo la generación de cada una de las certificaciones de los postgrados aprobados por parte de los participantes por medio de la creación de reportes.
Precondiciones	Que exista algún postgrado registrado en el libro
Flujo Principal	El caso de uso iniciará cuando se reciba una solicitud del sistema cliente por medio de la selección del nombre del postgrado y el nombre del participante, para la creación e impresión de la correspondiente Certificación de Postgrado y la finalización del caso de uso.
Referencias	39

Caso de Uso	Generar Certificados de Diplomado
-------------	--

Actores	Usuario
Objetivos	Es un conjunto de operaciones que tiene como objetivo la generación de cada una de las certificaciones de los diplomados aprobados por parte de los participantes por medio de la creación de reportes.
Precondiciones	Que exista algún diplomado registrado en el libro
Flujo Principal	El caso de uso iniciará cuando se reciba una solicitud del sistema cliente por medio de la selección del nombre del diplomado y el nombre del participante, para la creación e impresión de la correspondiente Certificación de Diplomado y la finalización del caso de uso.
Referencias	40

Caso de Uso	Generar Certificados de Reconocimiento
Actores	Usuario
Objetivos	Es un conjunto de operaciones que tiene como objetivo la generación de cada una de las certificaciones de los profesores que impartieron dichos postgrados por medio de la creación de reportes.
Precondiciones	Que exista algún postgrado o diplomado registrado en el libro.
Flujo Principal	El caso de uso iniciará cuando se reciba una solicitud del sistema cliente por medio de la selección del nombre del profesor y el nombre del postgrado que impartió, para la

	creación e impresión de la correspondiente Certificación de Reconocimiento y la finalización del caso de uso.
Referencias	41

Caso de Uso	Generar Certificados Hago Constar
Actores	Usuario
Objetivos	Es un conjunto de operaciones que tiene como objetivo la generación de cada una de las certificaciones de los participantes no profesionales que aprobaron un postgrado por medio de la creación de reportes.
Precondiciones	Que exista algún postgrado o diplomado registrado en el libro.
Flujo Principal	El caso de uso iniciará cuando se reciba una solicitud del sistema cliente por medio de la selección del nombre del postgrado y el nombre del participante técnico, para la creación e impresión de la correspondiente Certificación Hago Constar y la finalización del caso de uso.
Referencias	12

Caso de Uso	Generar Reportes de Asentamiento en Libros.
Actores	Usuario
Objetivos	Es un conjunto de operaciones que tiene como objetivo el asentamiento de los postgrados y diplomados aprobados por los participantes en los diferentes libros de asentamiento.
Precondiciones	Que todos los postgrados y diplomados se encuentren debidamente ingresados.
Flujo Principal	El caso de uso iniciará cuando se reciba una solicitud del sistema cliente por medio de la selección del tipo de libro y

	el número de libro y folio para la creación e impresión registró en libros y el caso de uso terminará.
Referencias	43

Caso de Uso	Generar Reportes de Asentamiento en Libros.
Actores	Usuario
Objetivos	Es un conjunto de operaciones que tiene como objetivo el asentamiento de los postgrados y diplomados aprobados por los participantes en los diferentes libros de asentamiento.
Precondiciones	Que todos los postgrados y diplomados se encuentren debidamente ingresados.
Flujo Principal	El caso de uso iniciará cuando se reciba una solicitud del sistema cliente por medio de la selección del tipo de libro y el número de libro y folio para la creación e impresión registró en libros y el caso de uso terminará.
Referencias	43

3.2 Definición de Actores.

Dado que la aplicación que se pretende desarrollar es un Sistema Automatizado, puede ser identificado un solo Actor, el cual, llamaremos en lo sucesivo **Usuario**. Dicho usuario, será la única entidad externa que interactúe con el sistema.

Actor	Justificación.
--------------	-----------------------

Usuario	Encargado de mantener el control, seguridad y actualización de la información sobre la actividad de postgrado de cada facultad o dependencia de la UC.
---------	--

3.3. Definición del Modelo Conceptual.

Dado que un **modelo conceptual** es una descripción de conceptos (objetos), atributos y asociaciones del dominio de un problema real, no es una descripción del diseño del software, el cual se puede desarrollar en paralelo con el desarrollo de los casos de uso. [Larman, 2004].

Las informaciones que fluyen entre las actividades que conforman un proceso de negocio representan entidades o conceptos del dominio y por tanto son una buena base para crear el modelo conceptual del sistema. A partir de la lista de informaciones que se han obtenido a lo largo del análisis del sistema, se construyó el modelo conceptual del sistema propuesto. Este modelo incluirá las clases del dominio y las relaciones entre ellas. Su representación aparece en el Anexo III.

3.4. Diseño de la Base de Datos.

La base de datos del sistema contiene toda la información que es procesada cumpliendo con todos los objetivos a tener cuenta en cualquier sistema SGBD como son: la independencia de datos, los programas de aplicación, la minimización, la redundancia, la integración, la sincronización, la integridad de los datos, la seguridad, la facilidad de manipulación y el control centralizado de los datos. Posteriormente, de obtener la información a procesar, se permite la creación de un modelo lógico global de datos a

partir del cual se estructurará el Diagrama Entidad – Relación. El DER puede ser encontrado en el Anexo IV.

3.5. Diseño Físico de la Base de los Datos.

Es de gran importancia en el sistema propuesto, ya que es aquí donde se aplican las técnicas de implementación para lograr una información bien organizada y que cumpla con los objetivos de cualquier SGDB mencionados anteriormente, a través del diseño físico se realizan las consultas que el sistema brindará a los directivos y a los especialistas en el tema, y partiendo de allí, se inicia con la implementación y desarrollo de la aplicación cliente, mediante la cual el usuario explota el sistema. Como resultado del diseño físico se obtuvieron las siguientes tablas que pueden ser analizadas en el Anexo V.

3.6. Diagramas de Secuencia.

Con el objetivo de mostrar las interacciones entre los objetos organizados en una secuencia de tiempo y comprender mejor el funcionamiento del sistema, se elaboraron los diagramas de secuencia correspondientes. Estos describen los objetos y las clases que interactúan en el escenario, así como la secuencia de mensajes intercambiados entre clases y objetos para llevar a cabo la funcionalidad de éste. Los diagramas de secuencia del sistema podrán ser encontrados en el Anexo VI.

3.7. Diagramas de Clases.

A partir del modelo conceptual y los diagramas de secuencia vistos anteriormente, se construyeron los diagramas de clases que aparecen en el Anexo VII. En este diagrama son mostradas las clases que han sido identificadas en los diagramas de secuencia así como las relaciones entre las mismas. El comportamiento de las clases que aparece en los diagramas de secuencia ha sido extraído incluyendo los métodos que aparecen en ellos, y refinando las asociaciones de multiplicidad, herencia, navegabilidad y agregación.

3.8. Diagrama de Componentes

En el diagrama de componente mostrado en el Anexo VIII, se muestra básicamente la estructura del sistema en cuanto a la distribución y relación de los componentes de éste.

3.9. Diagramas de Despliegue

En el diagrama de despliegue mostrado en el Anexo IX, se describe la situación de los componentes en una posible implantación del sistema propuesto de acuerdo a los requisitos iniciales. Con este diagrama se pretende describir la topología del sistema, mostrando la ejecución de los componentes que residen en ellos.

Para la realización de este diagrama no se utilizó la notación UML, sino que se busco una representación gráficamente más expresiva, para de esta forma lograr que la distribución de los componentes del sistema quede representada lo más claro posible.

3.10. Patrones utilizados en el Desarrollo del Proyecto.

La utilización de patrones en el diseño e implementación de software es extremadamente importante de acuerdo con las ventajas que brindan los mismos. En el desarrollo de la plataforma para la implementación del sistema fue necesario el uso y aplicación de un conjunto de patrones de diseño dado la afinidad de los mismos con el dominio y contexto desarrollado.

A continuación, serán relacionados y documentados cada uno de los patrones que fueron utilizados en el desarrollo de la plataforma.

Diseño y Organización Preliminar

Para comenzar todo proyecto, debe realizarse un estudio por el equipo de desarrollo del mismo, para sentar las bases y organizar todo el proceso de diseño e implementación del sistema a desarrollar. Según [Brown, 1998], en esta etapa preliminar *“un cuidadoso análisis trae consigo una etapa de implementación mucho menos costosa y de mayor eficacia”*.

Existen definidos una serie de patrones que ayudan en la dirección y la toma de decisiones para el desarrollo de esta primera etapa. Estos podrán ser aplicados en dependencia de las características del grupo de desarrollo y del sistema a desarrollar. A continuación son enumerados todos aquellos patrones que fueron utilizados en el diseño y organización de la plataforma para la implementación del Sistema Automatizado del Control de la Actividad de Postgrado.

Code Ownership

El patrón *Code Ownership* [Coplien, 1996] o Propiedad de Código establece que un desarrollador no puede tener un ritmo de acción uniforme en un código que constantemente esté cambiando sin su conocimiento. Esto ocurre en un contexto donde las responsabilidades no se encuentran correctamente establecidas o donde varios desarrolladores modifican los mismos módulos. Para lidiar con esto es necesario que cada módulo en el sistema sea responsabilidad de un solo desarrollador. Excepto en circunstancias excepcionales y específicas, el código debe ser modificado solamente por su propietario.

Engage QA

El patrón *Engage QA* [Coplien, 1996] o Conseguir Certidumbre de Calidad trata la manera de garantizar la calidad del producto que se encuentra desarrollándose. El éxito de un producto depende sin lugar a dudas de la calidad del mismo. Para lograrlo es necesario hacer de la Certidumbre de Calidad (*Quality Assurance* - QA) un rol central en el desarrollo del proyecto. Es necesario acoplarla con el desarrollo desde el primer momento que exista algo que probar. La aplicación de control y pruebas al código fuente debe realizarse en paralelo con el proceso de implementación. Como resultado, la calidad del producto crecerá rápidamente.

Group Validation

El patrón *Group Validation* [Coplien, 1996] o Validación de Grupo asegura la calidad del producto. Este patrón se encuentra estrechamente relacionado con el patrón *Engage QA*. Antes que este último sea aplicado, el equipo de desarrollo – incluyendo los clientes – deberá validar el diseño. Podrán ser utilizadas técnicas como las tarjetas CRC y depuración en grupo para socializar y solucionar los problemas. El objetivo es lograr

un sistema de mucha más calidad, los errores serán resueltos mucho antes que se conviertan en verdaderos problemas.

Scenarios Define Problem

El patrón *Scenarios Define Problem* [Coplien, 1996] o Escenarios Definen el Problema plasma que los documentos de diseño son frecuentemente inefectivos como vehículos para comunicar la visión del cliente de cómo el sistema debe funcionar. Esta comunicación es crucial para el éxito del proyecto. La solución no es otra que capturar los requerimientos funcionales del sistema como casos de usos. El problema será entonces definido y bien ejemplificado, siendo esta documentación de gran valor para el cliente del proyecto debido a la poca utilización de lenguaje técnico e incomprendibilidad del mismo en los casos de usos.

Implementación

En la implementación de la plataforma fueron identificados y aplicados un número de patrones de diseño con el objetivo de facilitar el desarrollo de la misma. Estos patrones son clasificados en las categorías de Patrones de Creación, Patrones Estructurales y Patrones de Comportamiento. Los primeros atienden lo concerniente al proceso de creación de objetos, los segundos lidian con la composición de clases y objetos, y los terceros caracterizan la forma en que las clases u objetos interactúan y distribuyen sus responsabilidades. A continuación, enumeramos y fundamentamos dichos patrones de diseño.

Singleton

El patrón *Singleton* [Townsend, 2002] o Ocurrencia Simple asegura que una clase tenga solamente una sola instancia creada y provee un punto global para acceder a ésta. Una clase basada en este patrón, protege su constructor de forma tal que solamente pueda ser llamado desde la propia clase. Debe ser expuesto un método que permita a los clientes crear una instancia de la clase en caso de que ésta no haya sido creada previamente. Si tiene sentido que una clase tenga solamente una instancia, este patrón debe ser implementado para evitar posibles errores en la utilización del código.

Facade

El patrón *Facade* [Gamma, 1995] o Fachada provee una interfase unificada a conjunto de interfases en un subsistema específico. El patrón Fachada define una interfase de alto nivel que logra una utilización del subsistema mucho más sencilla ya que evita que el cliente tenga que lidiar con clases mucho más complejas. En todos los casos, el código ocultado por la Fachada puede ser modificado sin necesidad de cambios en el código cliente.

Iterator

El patrón *Iterator* [Gamma, 1995] o Iterador provee una forma de acceder secuencialmente a los elementos de un objeto sin exponer su representación interna. El lenguaje de programación C# implementa esta funcionalidad en la mayoría de las colecciones con las que cuenta. Este patrón es una vía de separar el recorrido a través de un conjunto de nodos, de los nodos mismos. La programación genérica es una buena forma de implementar iteradores.

Control y Prueba del Código Fuente

La realización de pruebas luego de culminado un sistema es hoy en día toda una ciencia. “Para lograr la producción de una aplicación lo más confiable posible, la realización de controles al código fuente es de vital importancia” [DeLano, 1996]. Existe un grupo de patrones relacionados con la aplicación de dichos controles a sistemas luego de finalizada la implementación del mismo. A continuación relacionamos aquellos que fueron utilizados en el control y prueba de la plataforma de catálogos de información.

Problem Área

El patrón Problem Área [DeLano, 1996] o Área del Problema especifica las áreas del sistema que deberán recibir especial atención en la realización de pruebas y control de código fuente. La solución es mantener una lista de las áreas con problemas persistentes y los casos de pruebas para estas. Deben ser realizadas pruebas sistemáticas a estas áreas aunque no sean realizadas modificaciones en el código comprendido. Como resultado podrán ser controlados errores que ocurren con determinada frecuencia, lo que trae consigo un sistema mucho más confiable y libre de desaciertos.

Strange Behavior

El patrón Strange Behavior [DeLano, 1996] o Comportamiento Raro trata sobre las decisiones a tomar cuando una característica del sistema implementado se encuentra funcionando pero no de la forma esperada. Debe tomarse cualquier comportamiento inusual como un posible problema y seguir su pista. Esto debe ser realizado aún cuando el problema identificado no se encuentre relacionado con la prueba en cuestión. Es necesario estar alerta cuando las pruebas arrojen resultados aceptables pero diferentes de lo esperado.

Problem Cluster

El patrón Problem Cluster [DeLano, 1996] o Cluster del Problema trata sobre la estrategia a seguir para la selección del próximo punto de código a probar. La solución es concentrar las pruebas en aquellas áreas donde controles anteriores identificaron errores. Como resultado, las áreas problemáticas serán examinadas lo antes posible y los problemas podrán ser resueltos inmediatamente.

Killer Test

El patrón Killer Test [DeLano, 1996] o Control Asesino especifica la forma en que la calidad de un sistema en desarrollo puede ser determinada. Para esto debe ser construido un caso de prueba o test case que pueda ser ejecutado en cualquier momento bajo cualquier circunstancia. Este caso de prueba debe estar diseñado de forma tal que sea inminente su fallo. Todos los casos críticos y extremos deben ser incluidos. Mientras más complejo sea el caso de prueba, mejor podrá evaluarse la calidad del sistema en desarrollo. El resultado de este control será un buen termómetro para medir la estabilidad del sistema. Al ser encontrados y corregidos este tipo de problemas, la aplicación será gradualmente mucho más estable y eficiente.

Document the Problem

El patrón Document the Problem [DeLano, 1996] o Documentar el Problema explica la manera en que los problemas encontrados deben ser comunicados. Es necesario escribir un reporte con los problemas identificados. No puede confiarse en la memoria, no pueden ser aceptadas promesas de que todo será corregido, el problema debe quedar plasmado en documentos o sistemas de control de cambios. De esta forma, es mucho más seguro que el problema será corregido en el menor tiempo posible. Violar estas

consideraciones pudiera acarrear el olvido de un error que más tarde dará al traste con el buen funcionamiento de la aplicación.

Resumen.

En el presente capítulo fue analizado y definido el sistema propuesto para su posterior implementación, para ello se partió desde la definición y documentación de cada uno de los casos de uso y actores, para mostrar las funciones del sistema, sus entornos y las relaciones existentes entre ambos.

De la misma forma, se definió y construyó el modelo conceptual, el diseño de la base de datos representada por el diagrama Entidad-Relación, el diseño físico de la base de datos, representado por cada una de las tablas, los diagramas de secuencia, los diagramas de clases, los diagramas de componentes y los diagramas de despliegue, completando así el diseño global de la arquitectura propuesta. Por último se documentó cada uno de los patrones utilizados en el desarrollo del proyecto.

Capítulo IV. Prototipo del Sistema

En este Capítulo se describirá los principios generales que se siguieron para el diseño de las (UI Interfaces de Usuario), el tratamiento de errores, el sistema de ayuda y además se mostrará una breve guía ilustrada de las opciones que brinda el sistema.

4.1. Principios Generales del Diseño de las Interfaces de Usuario.

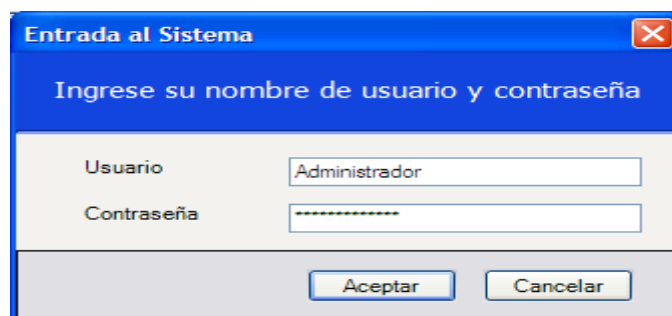
Las UI que aparecen en la aplicación toman las características de las ventanas de Windows XP gracias a la utilización del Catálogo de Recursos ARWEN. El diseño se realizó siguiendo un mismo estilo homogéneo en todas las formas, las formas poseen funciones de complejidad como el enlace presentado de la misma manera todo lo que

tenga funciones similares, ayudando así a que el usuario se familiarice rápidamente con la manipulación del sistema.

La aplicación presenta un menú principal en el borde superior de la pantalla. Su apariencia es similar a todos los de Windows, con el nombre de la aplicación en la parte superior y las principales funciones del mismo. Las pantallas se diseñaron de forma sencilla, partiendo de las principales propiedades (color de fondo, tamaño de las letras, tamaño de la pantalla, etc.) manteniendo un diseño uniforme en estas. Se hace uso de controles para hacer más fácil la selección por parte del usuario.

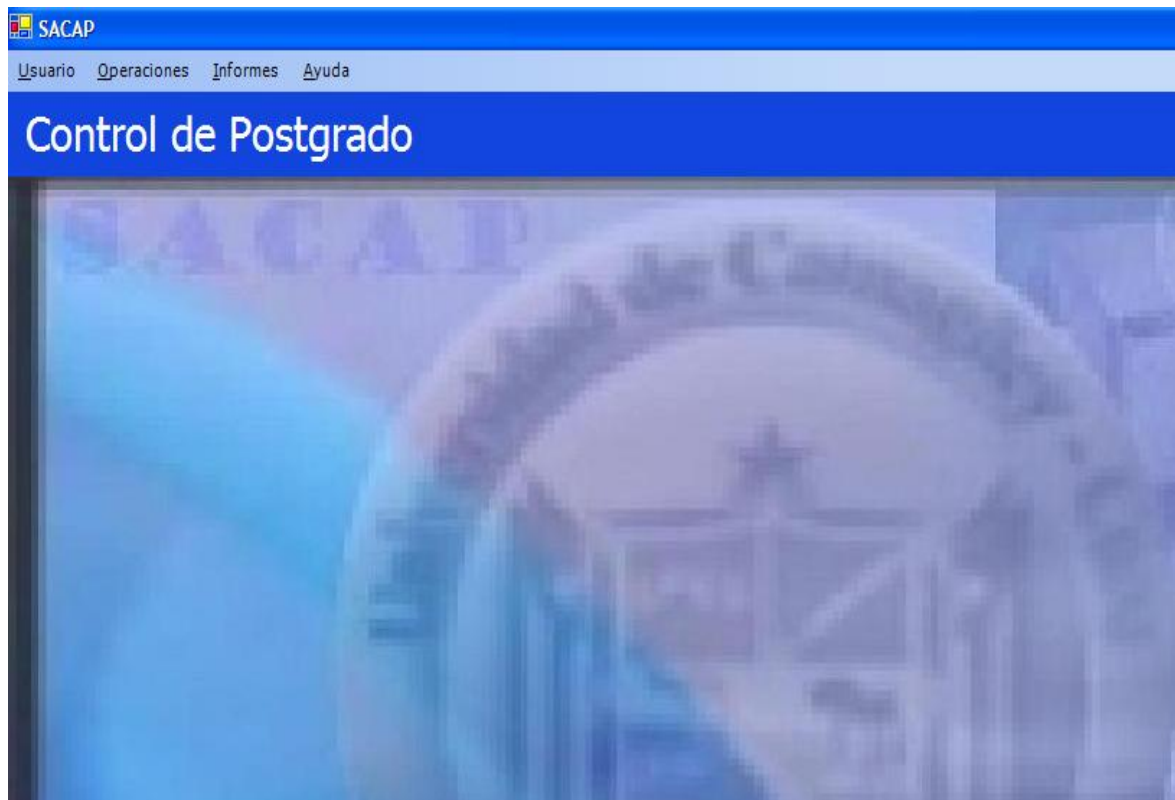
4.2. Principios Generales del Sistema SACAP.

Cuando se ejecuta la aplicación se desplegará la interfaz de usuario de autenticación la cuál requiere el ingreso del nombre del usuario y su respectiva contraseña, como se muestra en la figura.



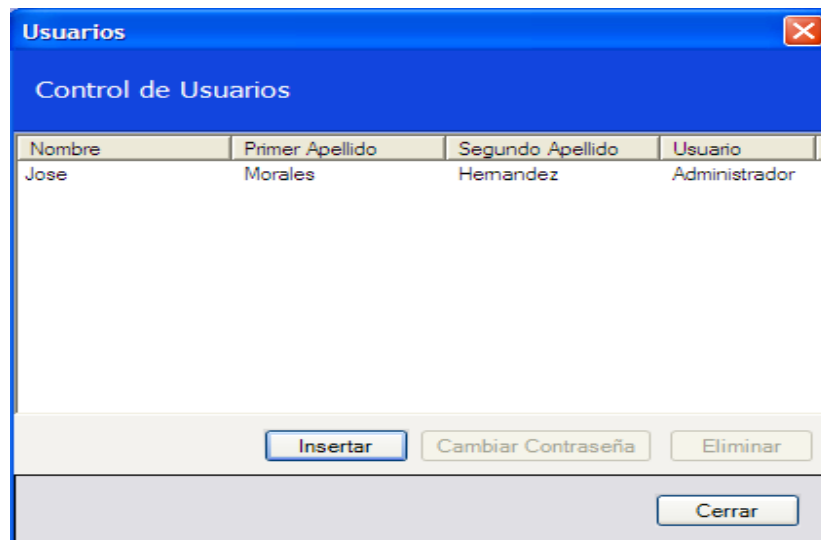
The image shows a Windows-style dialog box titled "Entrada al Sistema" (System Entry). The dialog has a blue title bar with a close button (X) in the top right corner. Below the title bar, the text "Ingrese su nombre de usuario y contraseña" (Enter your username and password) is displayed in blue. The main area of the dialog is white and contains two input fields. The first field is labeled "Usuario" (Username) and contains the text "Administrador". The second field is labeled "Contraseña" (Password) and contains a series of asterisks. At the bottom of the dialog, there are two buttons: "Aceptar" (Accept) and "Cancelar" (Cancel).

Posteriormente, de ingresados correctamente los datos de autenticación el sistema, mostrará al usuario la siguiente ventana principal con las siguientes opciones como son: Usuarios, Operaciones Reportes y Ayuda como se muestra en la figura.



4.2.1 Uso del Menú Usuarios.

Como primera opción el sistema brinda el submenú usuarios con las siguientes opciones:



- Usuarios. Que permite el ingreso de un nuevo usuario, el cambio de contraseña y la eliminación del mismo, como se muestra en la figura.
- Cambio de usuario. Que permite el cerrar una sesión de un usuario y abrir la sesión de otro usuario autorizado.
- Salir. Permite la desconectar el sistema con la base de datos y cerrar la aplicación.

4.2.2. Uso del Menú Operaciones.

Como primera opción de este submenú se brinda las siguientes opciones: facultades, profesores, postgrado, participantes y diplomado.

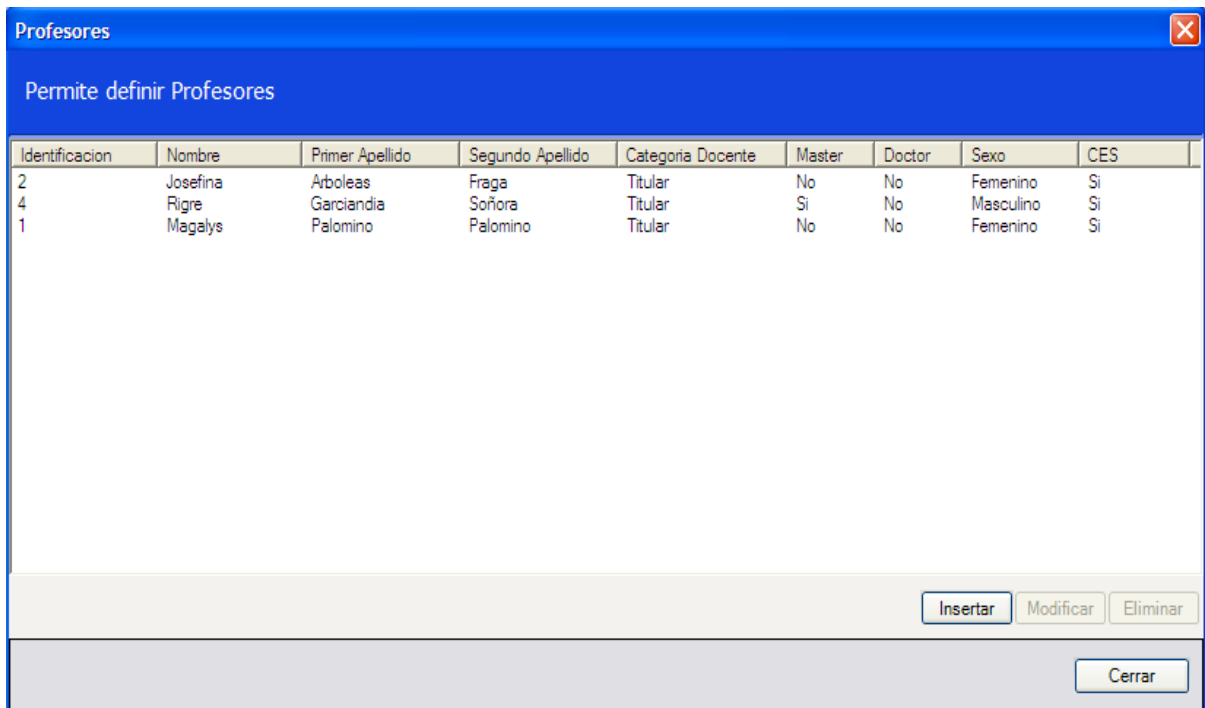
Tomando como referencia que todas las pantallas poseen la misma concepción en su diseño a continuación se mostrará una síntesis de cada una de ellas.

Uso del Submenú Facultades.

Este desplegará un listado de todas las facultades y las diferentes opciones de manipular esta información (insertar, modificar y eliminar).

Uso del Submenú Profesores.

Este desplegará un listado de todas las facultades y las diferentes opciones de manipular esta información (insertar, modificar y eliminar), como se muestra en la figura.



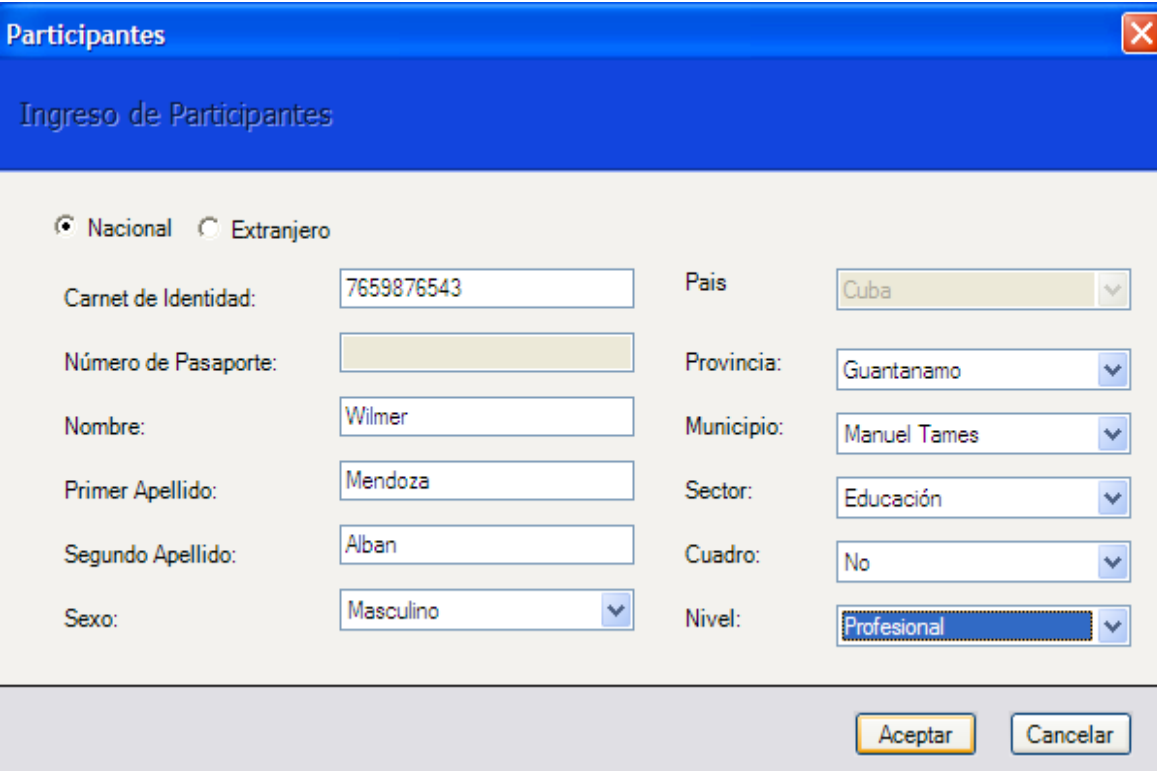
The screenshot shows a software window titled "Profesores" with a subtitle "Permite definir Profesores". It contains a table with the following data:

Identificacion	Nombre	Primer Apellido	Segundo Apellido	Categoria Docente	Master	Doctor	Sexo	CES
2	Josefina	Arboleas	Fraga	Titular	No	No	Femenino	Si
4	Rigre	Garcandia	Soñora	Titular	Si	No	Masculino	Si
1	Magalys	Palomino	Palomino	Titular	No	No	Femenino	Si

At the bottom of the window, there are four buttons: "Insertar", "Modificar", "Eliminar", and "Cerrar".

Uso del Submenú Participantes.

Este desplegará un listado de todos los participantes nacionales y extranjeros y las diferentes opciones de manipular esta información (insertar, modificar y eliminar), aquí se mostrarán específicamente la forma de cómo la aplicación realiza el proceso de inserción de los datos, como se muestra en la figura.



The image shows a software window titled "Participantes" with a close button in the top right corner. Below the title bar, the subtitle "Ingreso de Participantes" is displayed. The form contains two radio buttons: "Nacional" (selected) and "Extranjero". Below these are two columns of input fields. The left column includes: "Carnet de Identidad:" with the value "7659876543"; "Número de Pasaporte:" with an empty field; "Nombre:" with the value "Wilmer"; "Primer Apellido:" with the value "Mendoza"; "Segundo Apellido:" with the value "Alban"; and "Sexo:" with a dropdown menu set to "Masculino". The right column includes: "Pais:" with a dropdown menu set to "Cuba"; "Provincia:" with a dropdown menu set to "Guantanamo"; "Municipio:" with a dropdown menu set to "Manuel Tames"; "Sector:" with a dropdown menu set to "Educación"; "Cuadro:" with a dropdown menu set to "No"; and "Nivel:" with a dropdown menu set to "Profesional". At the bottom right of the form are two buttons: "Aceptar" and "Cancelar".

Uso del Submenú Postgrado

Este desplegará un listado de todos los postgrados y las diferentes opciones de manipular esta información (insertar, modificar y eliminar), aquí se mostrarán específicamente la forma de cómo la aplicación realiza el proceso de modificación de los datos, como se muestra en la figura.

Postgrado [X]

Insertar Postgrado

Nombre del Curso:

Facultad:

Fecha de Inicio:

Fecha Culminación:

Número de Horas:


Tipo de Planificación:

Nombre	Primer Apellido	Segundo apellido	Horas
Rigre	Garcandia	Soñora	20

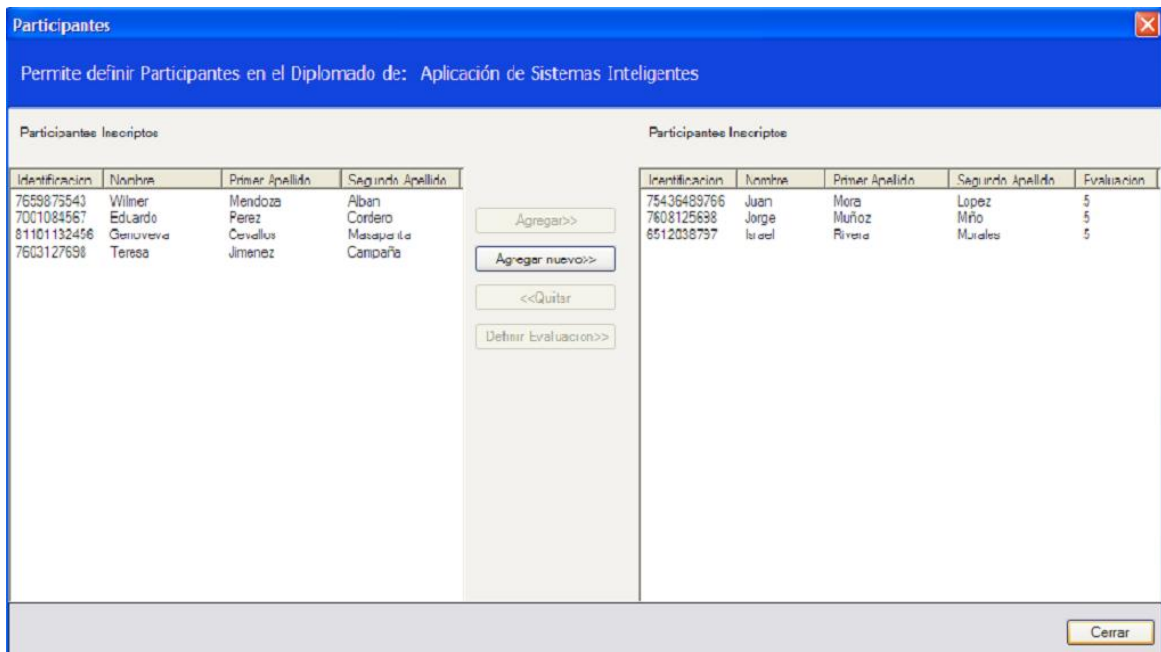
Uso del Submenú Diplomado

Este desplegará un listado de todos los diplomados y las diferentes opciones de manipular esta información (insertar, modificar y eliminar), aquí se mostrarán específicamente la forma de cómo la aplicación realiza el proceso de eliminación de los datos, como se muestra en la figura.

Advertencia [X]

 Usted esa a punto de eliminar algunos Diplomados. ¿Esta usted seguro?.

Ventanas de Ingreso de Datos Generales: Se utilizará para realizar cualquier operación, de las que el software ofrece, a los datos de un diplomado, al seleccionar esta opción se mostrará la ventana en la que se seleccionara, agregara y definirá la evaluación de un participante, como se muestra en la figura:



Para definir los postgrados de un diplomado, el sistema muestra la siguiente ventana.

Nombre	Facultad	Fecha Inicio	Fecha Culminación
Redes Neuronales	Infomatica	15/04/2005	29/04/2005

Nombre	Primer Apellido	Segundo Apellido	Horas
Josefina	Arboles	Fraga	50

Para registrar la evaluación de un participante, el sistema muestra la siguiente ventana.

Nombre del Curso:	Sistemas Inteligentes I
Identificación:	75436489766
Evaluación:	5

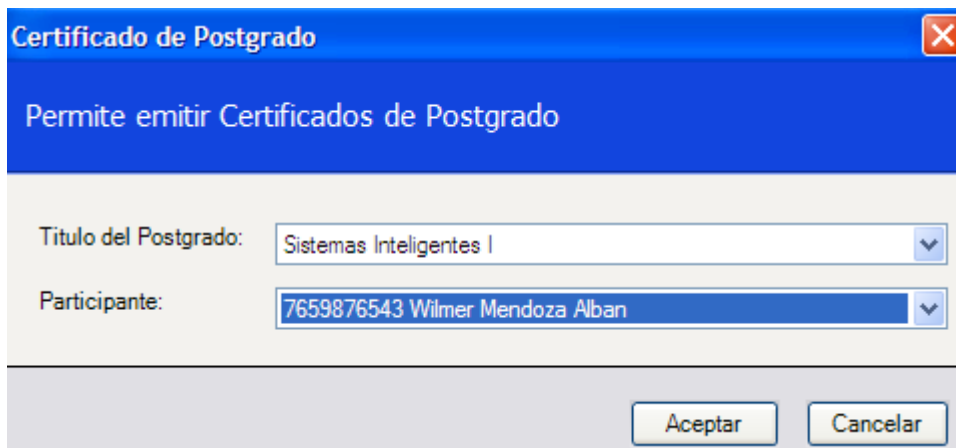
4.2.3. Uso del Menú Informes

Este despliega un listado de todos los reportes que el sistema genera, como se muestra en la figura:

- Certificado de Postgrado
- Certificado de Diplomado
- Certificado Hago Constar
- Modelo de Superación
- Certificado de Reconocimiento

Uso del Submenú Certificado de Postgrado

Este desplegará una ventana en la cual se elegirá los nombres del postgrado y del participante del cual se desea imprimir el certificado, como se muestra en la figura:



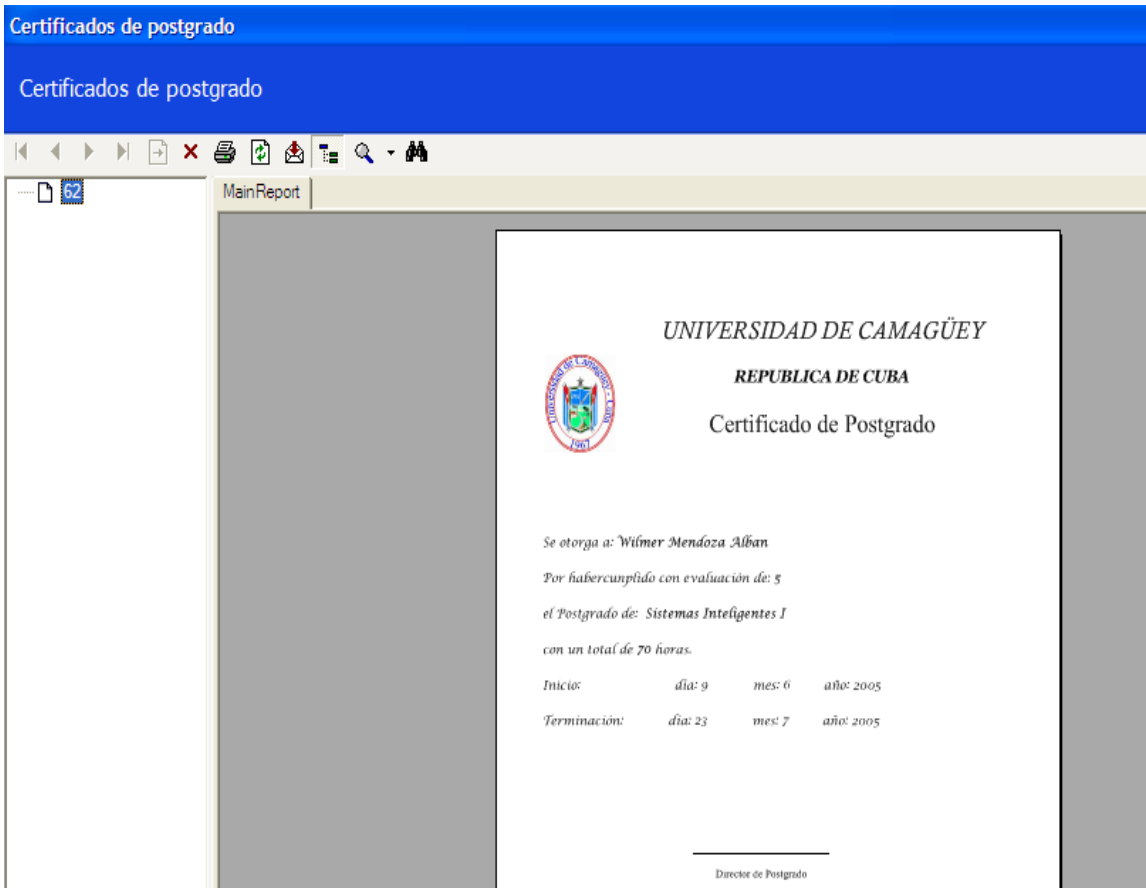
Certificado de Postgrado [X]

Permite emitir Certificados de Postgrado

Titulo del Postgrado:

Participante:

Aceptar Cancelar



4.2.4. Uso de los Botones.

Botón Insertar: Se utiliza para adicionar un nuevo registro a la base de datos. Al iniciar este proceso, se debe dar click sobre este botón, ingresar los datos deseados y presionar sobre el botón Aceptar para realizar la modificación física en la base de datos.

Botón Eliminar: Se utiliza para eliminar un registro de la base de datos.

Botón Agregar: Se utiliza para adicionar un participante inscripto a un determinado postgrado o un diplomado.

Botón Agregar Nuevo: Se utiliza para adicionar un participante nuevo a un determinado postgrado o un diplomado.

Botón Agregar dentro de la forma Postgrado Diplomado: Se utiliza para adicionar el nombre del profesor y las horas que impartió determinado postgrado o diplomado.

Botón Eliminar dentro de la forma Postgrado Diplomado: Se utiliza para eliminar el nombre del profesor y las horas que impartió determinado postgrado o diplomado.

Botón Agregar Postgrado dentro de la forma Postgrado Diplomado: Se utiliza para adicionar un postgrado a un diplomado.

Botón Eliminar Postgrado dentro de la forma Postgrado Diplomado: Se utiliza para eliminar un postgrado a un diplomado.

Botón Quitar: Se utiliza para eliminar un participante a un determinado postgrado o diplomado.

Botón Definir Evaluación: Se utiliza para definir la nota de un participante de un determinado postgrado o un diplomado.

Botón Cerrar: Se usa para salir de la forma.

Botón Aceptar: Si usted selecciona esta opción significa, que aceptó los datos y que desea que se ejecute la operación.

Botón Cancelar: Se usa para cancelar la operación y se cerrará la forma.

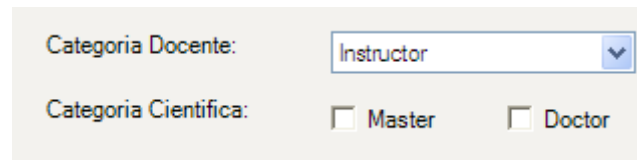
4.3. Principios Generales del tratamiento de los errores.

La interfaz gráfica de Windows en el proceso de ingreso de datos se ha diseñado de forma tal que el usuario tenga que teclear innecesariamente e inhabilitando aquellas operaciones que no puede realizar en determinados momentos, lo cual reduce la posibilidad de errores de escritura y al mismo tiempo facilita la validación.

El software se encuentra implementado de forma que brinde pantallas de advertencia al usuario de posibles errores o confirmando la realización de alguna operación. Estas UI están formadas por una pequeña ventana con un mensaje al usuario, en la barra de título se muestra: Error, Información o Confirmación.

Para la validación o verificación de los datos y para evitar posibles errores por parte del usuario en el instante de manipular la información se hace uso de las posibilidades que brinda el lenguaje por ejemplo:

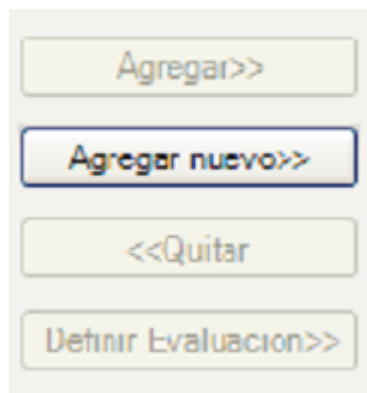
- Uso de Cuadros Combinados (ComboBox) y Botones de Chequeo (CheckBox), los que permiten selecciones prefijadas de posibilidades sin que el usuario se vea en la necesidad de teclearlas, como se muestra en la figura:



Categoria Docente:

Categoria Cientifica: Master Doctor

- Uso de la propiedad Activado (Enabled) de los objetos para permitir solamente el acceso en el momento adecuado, como se muestra en la figura:



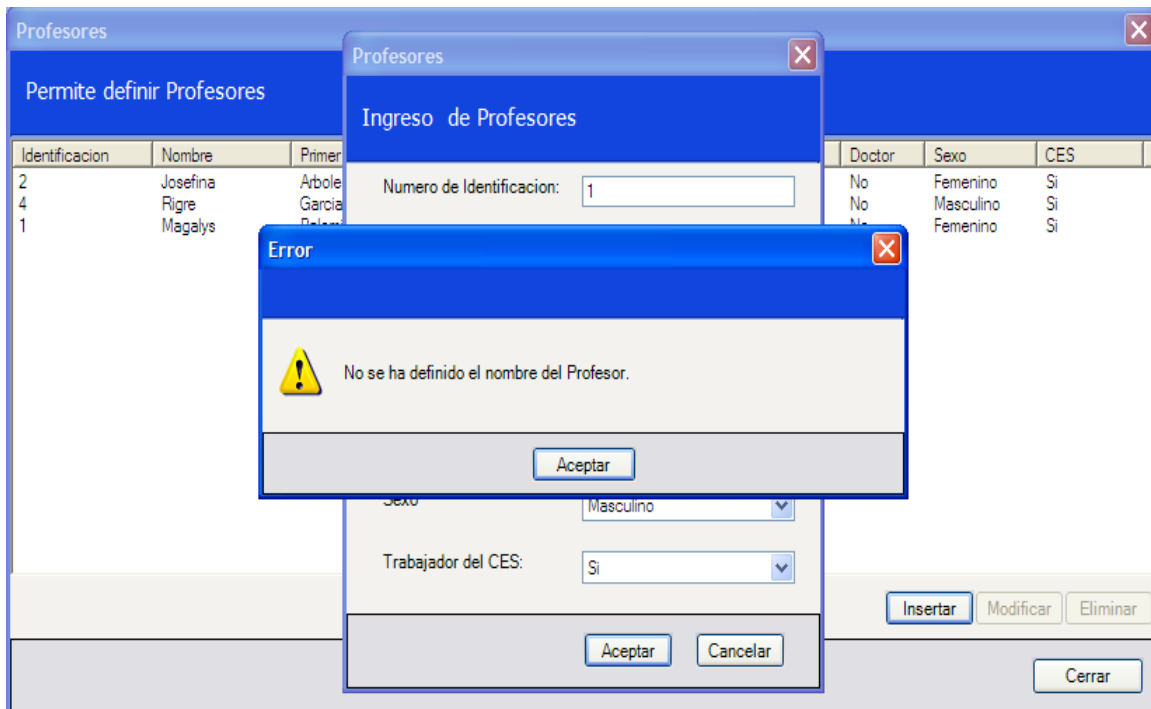
Agregar>>

Agregar nuevo>>

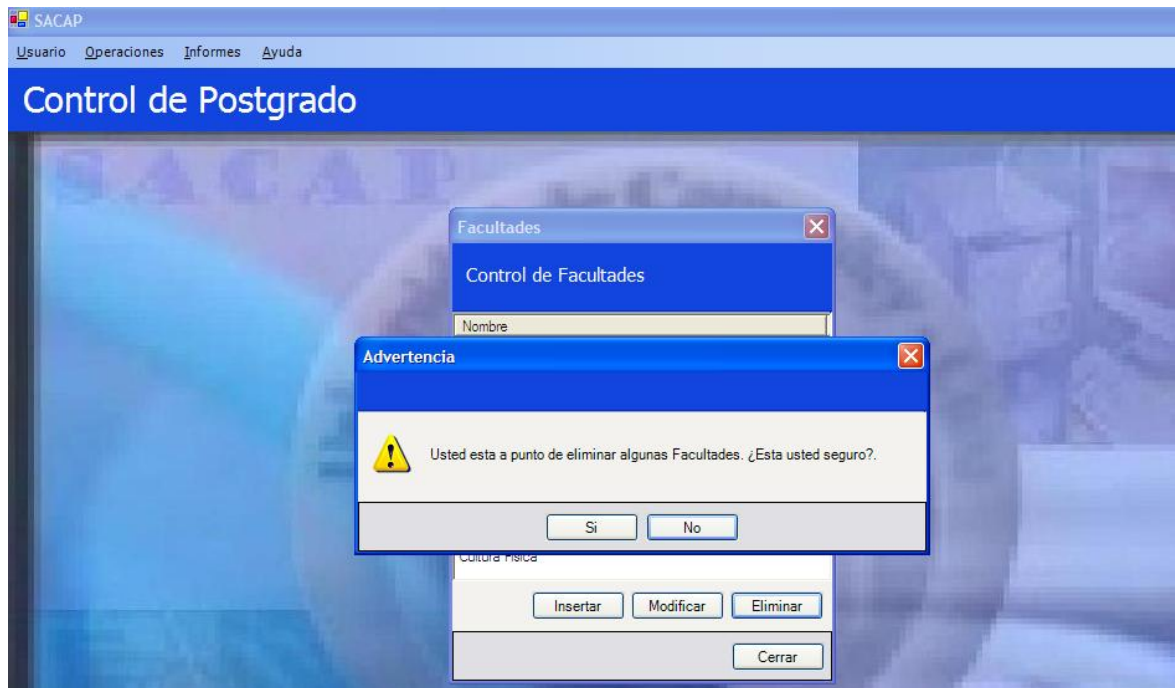
<<Quitar

Definir Evaluacion>>

- Uso de mensajes de advertencia cuando el usuario cometa un error equivocadamente, como se muestra en la figura:



- Uso de mensajes de confirmación cuando el usuario ingrese, confirme o actualice la información en la base de datos, como se muestra en la figura:



4.4. Principios Generales del Sistema de Ayuda

La ayuda de un sistema le permite al usuario brindarle un recurso de apoyo para que el, pueda despejar y solucionar sus dudas y dificultades en el caso de desconocimiento de alguna opción de las que brinda la aplicación. Esta aparece si se presiona en el Menú Ayuda, esta permite acceder a una descripción clara y detenida sobre el funcionamiento de cada proceso que se realiza. Es una ayuda típica de Windows y fue diseñada con el paquete utilitario Macromedia Dreamweaver.

4.5. Concepción General del sistema de seguridad y protección.

Debido a la importancia y sensibilidad de los datos que se manejan en las aplicaciones se hace necesario tener un sistema de seguridad y protección que garantice la calidad de los datos y el buen funcionamiento del software. Para ingresar al sistema siempre se requiere el usuario y contraseña que sean válidos en el sistema, de lo contrario no se permite que se pueda modificar la información almacenada en el mismo. Además el gestor de base de datos utilizado (SQL Server) presenta un control de seguridad estricto que permite que solo los administradores puedan ingresar a la base de datos y hacer modificaciones en la misma, por otro lado el propio SQL Server controla todo lo relacionado con la integridad de los datos posibilitando así que los datos almacenados sean 100% fiables.

Resumen.

En el transcurso del capítulo se expuso el prototipo del sistema, describiendo los principios generales para el diseño de las UI (Interfaces de Usuario), los principios generales del sistema SACAP, el tratamiento de errores, el sistema de ayuda, la concepción general del sistema de seguridad y protección y además se mostró una breve guía ilustrada de las opciones que brinda el sistema.

En el diseño de la interfaz la aplicación toman las características de las ventanas de Windows XP, en los principios generales del sistema SACAP la ventana principal presenta un menú con cuatro opciones, cada una de estas contiene un submenú que ayuda al usuario en el manejo. Para el tratamiento de errores el software se encuentra implementado de forma que brinde pantallas de advertencia al usuario de posibles errores o confirmando la realización de alguna operación. En el sistema de ayuda fue diseñado con el utilitario Macromedia Dreamweaver. Para la concepción del sistema de seguridad y protección, se cuenta con un control de seguridad del software diseñado y además con la del gestor de base de datos.