



**REPÚBLICA DE CUBA
MINISTERIO DE EDUCACIÓN SUPERIOR
UNIVERSIDAD DE GRANMA
FACULTAD DE INFORMÁTICA**

**UNIVERSIDAD TÉCNICA DE COTOPAXI
UNIDAD ACADÉMICA DE CIENCIAS DE LA
INGENIERÍA Y APLICADAS**

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN INFORMÁTICA.**

**SISTEMA PARA LA GESTIÓN Y CONTROL DE LOS
MEDIOS BÁSICOS DE LA FACULTAD DE
INFORMÁTICA DE LA UNIVERSIDAD DE GRANMA.**

AUTOR: MARCO VINICIO CUYO SIGCHA.

**TUTORA: ING. YUDI CASTRO BLANCO.
CO-TUTOR: ING. RAMÓN OSMANY RAMÍREZ TASÉ.**

**GRANMA, CUBA
ENERO, 2011**





**REPÚBLICA DE CUBA
MINISTERIO DE EDUCACIÓN SUPERIOR
UNIVERSIDAD DE GRANMA
FACULTAD DE INFORMÁTICA**

**UNIVERSIDAD TÉCNICA DE COTOPAXI
UNIDAD ACADÉMICA DE CIENCIAS DE LA
INGENIERÍA Y APLICADAS**

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN INFORMÁTICA.**

**SISTEMA PARA LA GESTIÓN Y CONTROL DE LOS
MEDIOS BÁSICOS DE LA FACULTAD DE
INFORMÁTICA DE LA UNIVERSIDAD DE GRANMA.**

AUTOR: MARCO VINICIO CUYO SIGCHA.

**TUTORA: ING. YUDI CASTRO BLANCO.
CO-TUTOR: ING. RAMÓN OSMANY RAMÍREZ TASÉ.**

**GRANMA, CUBA
ENERO, 2011**



AGRADECIMIENTOS

- ❧ *A Dios por toda la fortaleza que imprimió en mi persona, y permitir que sus manantiales de sabiduría llegaran a mí.*
- ❧ *A mí, madre que con su amor y dedicación se entregó por completo a mis estudios, apoyándome en todo momento; por ser esa mano tierna que siempre me brindó su ayuda.*
- ❧ *A mí padre por su confianza, apoyo, consejos y por ser un fiel amigo.*
- ❧ *A mi abuela por ser lo que más quiero en la vida.*
- ❧ *A mi hermano Widinsón, es lo mejor que me pudo haber pasado en la vida.*
- ❧ *A mis 9 hermanos por el apoyo incondicional que me han brindado.*
- ❧ *A mi novia por darme su amor, su fuerza, su constancia y apoyo; por enseñarme lo bello de amar y ser amado.*
- ❧ *A mí cuñado Ing. Juan Huachisaca por estar cuando más lo necesité.*
- ❧ *A demás familiares, por constituir una unidad indisoluble en especial: a la memoria del Técnico. Francisco Umajinga, Ing. Luis Sigcha, Cbs. Luis Sigcha, Héctor Sigcha, José Sigcha.*
- ❧ *A mis tutores por toda la ayuda que me han brindado, por sus orientaciones certeras y su paciencia.*
- ❧ *A mis amigos y en especial aquellos que estuvieron cerca de mí en toda la carrera: Lic. Moisés Gallardo, Sr. Chango, Lic. Jhon Pacheco, Darío Jiménez, Guillermo Ruíz, David Mullulema, Julio Romero, Wilmer Calo, Lic. Silvia Bonilla, Lic. Zoila Albear.*
- ❧ *A los profesores por estar cerca y darme la mano cuando tuve necesidad y en especial: Ing. Ítalo Serrano, Ing. Mayra Álbán, Profesor Jaime Nuñez, Profesora Nely Enríquez, Ing. Ringo López, Dra. C. Yolanda Soler Pelliser, Ing. Camilo Trujillo.*
- ❧ *A todos los que de una u otra forma han influido en mi formación profesional.*

Gracias eternamente.

DEDICATORIA

☞ *A quién le debo todo lo bello y bueno que hay en mi vida.*

Jesús mi fiel amigo.

☞ *No existen palabras justas para describir el amor que en innumerables formas de ti recibo: porque a ti me debo, te dedico este trabajo, Mamá.*

María Rosa Sigcha Montaguano.

☞ *A mi padre, confianza y oportuno consejo de un verdadero padre y amigo.*

Manuel Patricio Cuyo Umajinga.

☞ *A mi abuela, por ser mi guía apreciada.*

Ramona Montaguano.

☞ *A los dos sentimientos sublimes que lo hicieron posible: el Amor y la Amistad, a mi novia, inspiración de todos mis proyectos.*

Mi Amor.

☞ *A mi hermano, por ser lo que más quiero en la vida.*

Widinsón René Cuyo Sigcha.

☞ *A mis 9 hermanos, faro de mis pasos y mis sueños.*

Todos por igual.

Este trabajo es para ustedes....

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor del trabajo de diploma titulado: **Sistema para la gestión y control de los Medios Básicos de la Facultad de Informática de la Universidad de Granma**, y que el mismo pertenece a la Facultad de Informática para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmo la presente a los 10 días del mes de Febrero del 2011.

Firma del Autor
Marco Vinicio Cuyo Sigcha.

OPINIÓN DEL USUARIO DEL TRABAJO DE DIPLOMA

El trabajo de diploma titulado: **Sistema para la gestión y control de los medios básicos de la Facultad de Informática de la Universidad de Granma**, fue realizado para la facultad de Informática de la Universidad de Granma. Esta entidad considera que en correspondencia con los objetivos trazados el trabajo de diploma le satisface.

- Totalmente
- Parcialmente en un _____ %

Los resultados de este trabajo de diploma le reportan a la entidad los beneficios siguientes:

Y para que así conste se firma el presente a los ____ días del mes de _____ del año ____.

Nombre del representante de la Entidad

Cargo

Firma y Cuño

RESUMEN

En el departamento de administración de la Facultad de Informática de la Universidad de Granma, se registra y controla la información, los movimientos y servicios relacionados con los medios básicos existentes en cada uno de los locales de la facultad.

Todo este proceso se realiza de forma manual, requiriéndose para ello mucho tiempo, debido al gran volumen de información proveniente de fuentes diversas y dispersas que es necesario controlar. Actualmente, para conocer el estado de los medios básicos es necesario consultar informes aislados que en ocasiones contienen errores y están sensibles a pérdidas. Esto indica que una recopilación y análisis sistemático es importante para mantener la dinámica efectiva del proceso.

Teniendo en cuenta la problemática existente se hace necesario desarrollar una herramienta informática para facilitar el registro, control y procesamiento de la información de los medios básicos en la Facultad de Informática de la Universidad de Granma de forma rápida, eficiente y segura.

El proceso de desarrollo del producto “*Sistema para la gestión y control de la información relacionada con los Medios Básicos de la Facultad de Informática de la Universidad de Granma*” se realiza sobre la base de la arquitectura cliente/servidor, guiado su desarrollo por la metodología *Rational Unified Process (RUP)* y para el modelamiento visual *Unified Modeling Language (UML)*, como tecnología para la programación de páginas dinámicas el lenguaje *PHP versión 5.2.8*, con soporte de base de datos en *MySQL versión 5.1.30*, como servidor Web el *Apache versión 2.2.11* y como herramienta de desarrollo de aplicaciones Web el *Dreamweaver 8*.

Summary

In the administration department of the Ability of Computer science of the Granma University, he/she registers and it controls the information, the movements and services related with the existent basic means in each one of the local of the ability.

This process is carried out in a manual way, being required for it a lot of time, due to the great volume of information coming from diverse sources and you disperse that is necessary to control. At the moment, to know the state of the basic means it is necessary to consult isolated reports that contain errors in occasions and they are sensitive to losses. This indicates that a summary and systematic analysis is important to maintain the effective dynamics of the process.

Keeping in mind the existent problem becomes necessary to develop a computer tool to facilitate the registration, control and prosecution of the information of the basic means in the Ability of Computer science of the Granma University, in a quick, efficient and sure way.

The process of development of the product "Computer System for the administration and control of the information related with the Basic Means of the Ability of Computer science of the Granma University". Will be on the base of the architecture cliente/servidor, guided its development for the methodology Rational Unified Process (RUP) and for the visual modelling Unified Modeling Language (UML), as technology for the programming of dynamic pages the language PHP version 5.2.8, with database support in MySQL version 5.1.30, as servant Web the Apache version 2.2.11 and as tool of web's applications the Dreamweaver 8.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1. FUNDAMENTOS PARA LA GESTIÓN Y CONTROL DE LOS MEDIOS BÁSICOS.....	7
1.1 INTRODUCCIÓN.....	7
1.2 PROCESO DE AUDITORÍA Y CONTROL INTERNO.....	7
1.2.1 El Control Interno.....	8
1.2.2 Evaluación del Control Interno en Cuba.....	9
1.2.3 Control de los recursos, a disposición de la entidad.....	10
1.2.4 Control Interno en la Universidad de Granma.....	12
1.3 SISTEMAS PARA LA GESTIÓN DE PROCESOS CONTABLES.....	13
1.3.1 Sistema Manual.....	13
1.3.2 Herramienta Informática.....	13
1.4 HERRAMIENTAS PARA EL DESARROLLO DEL PRODUCTO INFORMÁTICO.....	14
1.4.1 APLICACIONES WEB.....	14
1.4.1.1 El modelo cliente/servidor.....	18
1.4.2 LENGUAJES DE PROGRAMACIÓN.....	19
1.4.2.1 Tecnologías de programación del lado del cliente.....	19
1.4.2.2 Tecnologías de programación del lado del servidor.....	22
1.4.3 SERVIDOR WEB.....	25
1.4.3.1 El servidor Web Internet Information Server (IIS).....	26
1.4.3.2 El servidor Web Apache.....	27
1.4.4 DISEÑO DE INTERFAZ.....	27
1.4.4.1 Zend Studio.....	28
1.4.4.2 Macromedia Dreamweaver 8.....	28
1.4.5 SISTEMAS GESTORES DE BASE DE DATOS (SGDB).....	30
1.4.5.1 SQL Server.....	31
1.4.5.2 MySQL.....	31
1.4.5.3 PostgreSQL.....	32
1.5 METODOLOGÍA DE MODELADO.....	33
1.5.1 Rational Unified Process (RUP).....	33
1.5.2 Lenguaje de Modelaje Unificado (UML).....	34
1.6 CONCLUSIONES DEL CAPÍTULO.....	35

CAPÍTULO 2. DESCRIPCIÓN Y CONSTRUCCIÓN DE LA SOLUCIÓN

PROPUESTA.	36
2.1 INTRODUCCIÓN.	36
2.2. MODELO DEL DOMINIO.	36
2.3 REQUERIMIENTOS DEL SISTEMA.	38
2.3.1 Requerimientos funcionales.	38
2.3.2 Requerimientos no funcionales.	40
2.4 FACTIBILIDAD.	42
2.4.1 Introducción.	42
2.4.2 Planificación basada en casos de uso.	42
2.5 MODELO DEL SISTEMA.	45
2.5.1 Actores del sistema.	45
2.5.2 Paquetes del Sistema.	46
2.5.3 Diagramas de casos de uso del sistema.	47
2.6 MODELO DEL ANÁLISIS.	47
2.7 MODELO DEL DISEÑO.	48
2.7.1 Patrones de Diseño.	49
2.8 MAPA DE NAVEGACIÓN.	50
2.9 DISEÑO DE LA BASE DE DATOS.	50
2.9.1 Diagrama de clases persistentes.	51
2.9.2 Modelo de Datos.	51
2.10 MODELO DE IMPLEMENTACIÓN.	51
2.11 DIAGRAMA DE DESPLIEGUE.	52
2.12 TRATAMIENTO DE ERRORES.	53
2.13 CONCLUSIONES DEL CAPÍTULO.	53
CONCLUSIONES	54
RECOMENDACIONES	55
BIBLIOGRAFÍA	56
ANEXOS	60

ÍNDICE DE FIGURAS

Figura 1.1. Esquema de representación del PHP.....	23
Figura 2.1 Diagrama de Clases del Dominio.....	38
Figura 2.2 Relación entre Actores del Sistema.....	46
Figura 2.3 Diagrama de Paquetes.....	46
Figura 2.4 Diagrama de Casos de Uso del Sistema: Paquete de Control de Medios Básicos.	47
Figura 2.5 Diagrama de Colaboración. Caso de Uso: Visualizar Medios Básicos	48
Figura 2.6 Diagrama de Despliegue.....	52
Figura 2.7 Aviso de una Acción Indebida.....	53

ÍNDICE DE TABLAS

Tabla 2.1 Criterio Estadístico de la Distribución del Esfuerzo entre las diferentes actividades de un Proyecto.	44
Tabla 2.2 Cálculo de Esfuerzo de las diferentes actividades del Proyecto.....	45
Tabla 2.3 Descripciones de los Actores del Sistema.	46

ÍNDICE DE ANEXOS

Anexo # 1. Factor de Peso de los Actores y Casos de Uso sin Ajustar.	60
Anexo # 2. Factor de Complejidad Técnica (TCF) y Factor Ambiente (EF).....	62
Anexo # 3. Diagrama de Caso de Uso del Sistema (CUS).	64
Anexo # 4. Descripción Textual de los Casos de Uso del Sistema.....	65
Anexo # 5. Diagramas de Colaboración.....	69
Anexo # 6. Diagramas de Clase del Diseño.....	72
Anexo # 7. Pantallas Representativas del Sistema.....	77
Anexo # 8. Mapa de Navegación.	79
Anexo # 9. Diagramas de Clases Persistentes.	80
Anexo # 10. Modelo de Datos.	81
Anexo # 11. Diagramas de Componentes.....	82

INTRODUCCIÓN

Los sistemas de información surgen aun cuando el hombre no tenía conocimientos de que lo estaba utilizando para su comunicación y para el aprendizaje. Mientras recolectaba objetos sin conocer exactamente su fin (Padrón, 2010). Con los constantes cambios que sufre el mundo de hoy, los sistemas de información han evolucionado significativamente debido al desarrollo tecnológico.

El uso de las Tecnologías de la Información y las Comunicaciones (TIC) unido a los servicios Web aporta numerosas ventajas que vienen dadas de acuerdo a sus características fundamentales, como son: agilizar los procesos, mejorar la productividad, aumentar la eficacia, permite la interactividad, instantaneidad, innovación, digitalización, interconexión y diversidad.

El proceso de informatización de la universidad cubana ha planteado el desarrollo de sistemas informáticos que faciliten la gestión de información, así como la informatización de procesos para lograr un mejor funcionamiento y rendimiento de estas instituciones de acuerdo a las necesidades de la actualidad. Entre los procesos preponderantes por la importancia que se le otorga cuenta la gestión y control de los recursos financieros.

El mundo contemporáneo exige que cada día exista una mejor gestión de los recursos financieros que posea cualquier organización, de forma que la triada del triunfo financiero (eficiencia, eficacia y calidad) generen la riqueza que satisfaga las necesidades crecientes del pueblo. En el caso muy particular de Cuba se afana por el perfeccionamiento de su modelo económico, el cual demanda que los organismos implicados de la fiscalización y control estén debidamente capacitados, para que puedan realizar su labor y que reine una sociedad sin corrupción e ilegalidades (Pedraza, 2009).

El entorno de control marca la pauta del funcionamiento de una empresa o institución e influye en la conciencia de sus empleados. Es la base de todos los demás componentes del control interno, aportando disciplina y estructura. Se puede afirmar que el Control Interno ha sido preocupación de la mayoría de las entidades, aunque con diferentes enfoques y terminologías, lo cual se puede

evidenciar al consultar los libros de texto de auditoría, los artículos publicados por organizaciones profesionales, universidades y autores individuales (Hernández, 2003).

La gestión de la información es el proceso de analizar y utilizar la información que se ha recabado y registrado para permitir a los administradores (de todos los niveles) tomar decisiones documentadas. Las universidades cubanas se convierten cada vez más en centros de investigación donde la gestión informativa constituye una palabra de orden en la nuevas misiones del estudiante universitario, en la actualidad más del 50% de la investigación que se desarrolla en Cuba se realiza en las Instituciones de Educación Superior (IES), en particular en las IES adscritas al Ministerio de Educación Superior (MES). Todo esto se verifica con la existencia de varios sistemas de gestión de información que forman parte del funcionamiento universitario a nivel nacional. (MES, 2010).

La Universidad de Granma, no se queda atrás en este aspecto, está contribuyendo a elevar la ciencia y tecnología en la sociedad cubana y a apoyar los programas de la revolución en la actualidad. La carrera de Ingeniería Informática prepara profesionales integrales comprometidos con la Revolución, cuya función es desarrollar los procesos relacionados con los sistemas informáticos en las organizaciones, con el propósito de obtener un incremento en la eficacia y la eficiencia de su funcionamiento con técnicas que le permiten analizar el entorno para delimitar los procesos computacionales, la información a procesar y las interrelaciones correspondientes; así como la gestión de proyectos informáticos con un alto nivel de profesionalidad (MES, 2010).

Con el objetivo de contribuir a la informatización de la universidad cubana y en especial a la Universidad de Granma y con el compromiso de perfeccionar el proceso de gestión y control de los recursos financieros, se ha optado en desarrollar un sistema para la gestión y control de los medios básicos de la facultad de Informática.

En el departamento de administración de la Facultad de Informática de la Universidad de Granma, se registra y controla toda la información relacionada con

los medios básicos existentes en cada uno de los locales de dicha facultad; así como los movimientos y servicios que se prestan a dichos medios básicos.

El administrador de la Facultad de Informática es el encargado de controlar todos los medios básicos, cuando el departamento de administración económica realiza la auditoría, este presenta su control interno mensual que es un valor representativo al 10% de los medios básicos, esta información se controla a través de un modelo de control.

Todo este proceso se realiza de forma manual presentando algunas limitantes:

- Para mantener actualizada la información se requiere de un gran número de horas, debido al gran volumen de información proveniente de fuentes diversas y dispersas.
- Para conocer el estado de los medios básicos es necesario consultar informes aislados que en ocasiones contienen errores, pueden sufrir deterioro y están sensibles a pérdidas.
- No existe seguridad de acceso a los documentos, estos pueden ser modificados o sustraídos, lo que atenta contra la validez y veracidad de la información.
- Se requiere de gran cantidad de papeles para el control de la información.

Luego de analizar la situación existente en el departamento de administración de medios básicos de la Facultad de Informática de la Universidad de Granma, surge el siguiente **problema de investigación**: ¿Cómo favorecer el proceso de gestión y control de la información relacionada con los medios básicos en la Facultad de Informática de la Universidad de Granma?

Este problema se enmarca en el **objeto de la investigación**: Proceso de gestión y control de la información de los medios básicos de la Facultad de Informática de la Universidad de Granma.

Para contribuir a resolver el problema se propone el siguiente **objetivo**: Desarrollar un sistema informático para la gestión, control y procesamiento de la información de los medios básicos de la Facultad de Informática de la Universidad de Granma.

El objetivo delimita el **campo de acción**, que es, sistema de gestión y control de la información de los medios básicos en la Facultad de Informática de la Universidad de Granma.

La investigación transcurrirá a través de las siguientes **tareas**:

- Analizar la situación existente del objeto de estudio para identificar mejoras, en el proceso de control de los medios básicos en la Facultad de Informática.
- Elaborar los fundamentos teóricos de la investigación.
- Desarrollar el análisis y diseño de la aplicación.
- Seleccionar las herramientas más usadas en la implementación de la aplicación Web a partir del análisis realizado.
- Realizar el diseño de la aplicación Web utilizando la metodología del Proceso Unificado de Desarrollo de Software (RUP) junto con el Lenguaje Unificado de Modelado (UML).
- Diseñar y normalizar la base de datos.
- Implementar la herramienta que de solución al problema planteado según el diseño realizado.

Para realizar las tareas se emplearán los siguientes **métodos**:

Métodos empíricos:

- *Revisión de documentos* para la recopilación de la información y la observación, lo que permitió conocer con claridad los datos de interés, para así poderlos procesar de una forma correcta.
- *La observación* se utilizó para ver la funcionalidad del personal administrativo responsable de la gestión de medios básicos y determinar las situaciones problemáticas.

Métodos teóricos:

- *El Método Causal* para estudiar los factores que provocan la necesidad de un proyecto informático de este tipo en la Facultad de Informática de la Universidad de Granma.

- *Análisis y síntesis* para la recopilación y procesamiento de la información, y arribar a las conclusiones de la investigación, la obtención de conocimiento y resumir la información a procesar.
- *La modelación sistemática* permitió realizar un análisis del objeto de la investigación y su posterior modelación, usando las herramientas informáticas de Ingeniería de Software y las técnicas adecuadas de programación de sistemas.

Técnicas

- *Entrevista* a la administradora y empleados de la Facultad de Informática, con el objetivo de conocer los aspectos fundamentales que deben tomarse, para la gestión de la información relacionada con los medios básicos, y obtener información sobre cómo es el flujo de información, cómo la procesan y lo que necesitan.

Aporte

Esta investigación facilitará el registro, control y procesamiento de la información relacionada con los medios básicos, lo que permite la realización de un mejor trabajo por parte del administrador a la hora procesar dicha información, favoreciendo la obtención de registros válidos, confiables y así emitir la información correcta en el momento oportuno.

Novedad

Se propone una aplicación Web que apoye el trabajo que se realiza en el departamento de administración de la Facultad de Informática de la Universidad de Granma relacionada con el registro, control y procesamiento de la información de los medios básicos, el cual permitirá una mejor rápida, eficiente y segura de la información.

En el diseño de la propuesta de solución se utiliza la metodología *Proceso Unificado de Desarrollo (RUP)*, la cual permite el desarrollo del software con las ventajas del *Lenguaje de Modelado Unificado (UML)*. Para la implementación de la misma se emplea como tecnología para la programación de páginas dinámicas el lenguaje *PHP versión 5.2.8*, con soporte de base de datos en *MySQL versión*

5.1.30, como servidor Web el *Apache versión 2.2.11* y como herramienta de desarrollo de aplicaciones Web el *Dreamweaver 8*.

El documento está estructurado en introducción, dos capítulos, conclusiones, recomendaciones, bibliografía y anexos.

El *capítulo 1* abarca temas relacionados con el objeto de estudio y el campo de acción de la investigación, así como un estudio bibliográfico actualizado sobre las principales tendencias y tecnologías actuales, incluyendo las herramientas usadas en el desarrollo de aplicaciones Web.

En el *capítulo 2* se profundiza en el entorno donde se desarrolla el sistema propuesto, los requerimientos funcionales y no funcionales que deben satisfacerse en el sistema, la modelación de los casos de uso del sistema y por último se hace una descripción del diseño e implementación que se ha seguido para el desarrollo del sistema.

CAPÍTULO 1. FUNDAMENTOS PARA LA GESTIÓN Y CONTROL DE LOS MEDIOS BÁSICOS.

1.1 Introducción.

En el presente capítulo se hace un análisis del objeto de estudio y el campo de acción de la investigación para la selección de los procesos a informatizar. Se abordan las tendencias y tecnologías actuales en cuanto a lenguajes de programación Web, servicios Web, diseño de interfaz, sistemas gestores de bases de datos y metodologías para el desarrollo de sistemas informáticos; haciendo énfasis en las que se emplean en la presente investigación.

1.2 Proceso de Auditoría y Control Interno.

La auditoría es la evaluación de planes, programas, proyectos y operaciones de una organización o entidad pública, a fin de medir e informar sobre el logro de los objetivos previstos, la utilización de los recursos públicos en forma económica, eficiente y la fidelidad con que los responsables cumplen con las normas jurídicas involucradas en cada caso. Además, los conceptos de Eficacia y Eficiencia están indisolublemente asociados a esta clase de auditoría (Jerez, 2006).

Jerez (2006), plantea que por eficacia se entiende el grado de cumplimiento de una meta, la que puede estar expresada en términos de cantidad, calidad, tiempo, costo, entre otros. Es fundamental, por lo tanto, que la organización cuente con una planificación detallada, con sistemas de información e instrumentos que permitan conocer en forma confiable y oportuna la situación en un momento determinado y los desvíos respecto a las metas proyectadas.

Este mismo autor considera que la eficacia se refiere a la relación entre los bienes o servicios producidos y los recursos utilizados para producirlos. Una operación eficiente produce el máximo de producto para una cantidad dada de insumos o requiere del mínimo de insumos para una calidad y cantidad de producto determinada. El objetivo es incrementar la productividad.

Objetivos de la auditoría de gestión.

La auditoría de gestión, está conformada por normas razonables que permitan comparar y evaluar condiciones existentes y tiene como objetivos:

- Establecer el grado en que el ente y sus servidores han cumplido adecuadamente los deberes y atribuciones que les han sido asignados.
- Determinar si tales funciones se han ejecutado de manera económica, eficiente y eficaz.
- Definir si los objetivos y metas propuestas han sido logrados.
- Precisar si se están llevando a cabo, exclusivamente, documentos de un control teniendo en cuenta la planificación establecida (Bejerano, 2010).

1.2.1 El Control Interno.

Es la base donde se gestionan las actividades y operaciones de una entidad, es decir, que las actividades de producción, distribución, financiamiento, administración, entre otras son regidas por el Control Interno. Es un instrumento de eficiencia y no un plan que proporciona un reglamento, el sistema de Control Interno, es aquel que no daña las relaciones de empresa a clientes y mantiene en un nivel de alta dignidad humana las relaciones de dirigentes y subordinados; su función es aplicable a todas las áreas de las operaciones de los negocios, de su efectividad depende que la administración obtenga la información necesaria para seleccionar de las alternativas, los intereses mejores de la entidad (Blanco, 2009).

El Control Interno es el sistema por el cual se da efecto a la administración de una entidad económica. La administración se emplea para designar el conjunto de actividades necesarias para lograr el objeto de la entidad económica. Por lo tanto abarca, las actividades de dirección, financiamiento, promoción, distribución y gestión de una empresa; sus relaciones públicas, privadas y la vigilancia general sobre su patrimonio y sobre aquellos de quien depende su conservación y crecimiento.

El Control Interno es el conjunto de planes, organización, métodos, procedimientos y otras medidas de una institución, tendientes a ofrecer una garantía razonable de que se cumplan los siguientes objetivos principales:

- Promover operaciones metódicas, económicas, eficientes y eficaces, así como productos y servicios de la calidad esperada.
- Preservar al patrimonio de pérdidas por mala gestión, errores, fraudes o irregularidades.

- Respetar las leyes y reglamentaciones, como también las directivas y estimular al mismo tiempo la adhesión de los integrantes de la organización a las políticas y objetivos de la misma.
- Obtener datos financieros, de gestión completos y confiables, presentados a través de informes oportunos (ONU, 2008).

1.2.2 Evaluación del Control Interno en Cuba.

Los antecedentes del Decreto Ley 107/09, posteriormente, por el Acuerdo 2914 del Comité Ejecutivo del Consejo de Ministros, del 30 de mayo de 1995, se creó la Oficina Nacional de Auditoría (ONA), adscrita al Ministerio de Finanzas y Precios, la que se constituyó como Órgano de Fiscalización Superior del país.

El 8 de junio de 1995 se dictó el Decreto Ley No. 159 “De la Auditoría”, el que declaró como tarea fundamental del Estado, el fortalecimiento del control económico y administrativo de las entidades, dirigida a prevenir el uso indebido de los recursos, rescatar la disciplina y la responsabilidad, así como lograr la máxima transparencia en la información económica-contable e, igualmente, verificar la honestidad de personas naturales y jurídicas, en la administración de los recursos del Estado (Pedraza, 2004).

En el año 2001, respondiendo al desarrollo alcanzado por el país en el control económico administrativo, mediante el Decreto Ley No. 219 del 25 de Abril del 2001, fue creado el Ministerio de Auditoría y Control, como el organismo de la Administración Central del Estado encargado de dirigir, ejecutar y controlar la aplicación de la política del Estado y del Gobierno en materia de Auditoría Gubernamental, Fiscalización y Control Gubernamental; así como para regular, organizar, dirigir y controlar, metodológicamente, el Sistema Nacional de Auditoría, teniendo como base de soporte la anterior Oficina Nacional de Auditoría (ONA) (Antecedente del Decreto Ley 107/09).

Según Bejerano (2010), en el sector estatal y en las relaciones entre las empresas y entidades, aún es insuficiente el establecimiento y la observación de las medidas de control interno, y continúan detectándose un conjunto de deficiencias e irregularidades en el funcionamiento de las direcciones administrativas, que se concentran fundamentalmente en la ejecución del

presupuesto, el control de los recursos materiales y financieros, en los portadores energéticos, obligaciones contractuales, inversiones, nóminas, efectivo, y tecnología de la información, que posibilitan las indisciplinas, ilegalidades y hechos de corrupción.

1.2.3 Control de los recursos, a disposición de la entidad.

Tener en cuenta que el control de los recursos es una de las bases elementales del Control Interno, y no sólo porque tribute a la toma de medidas en la ocurrencia de cada hecho, sino porque han de crearse todos los mecanismos necesarios para garantizar el control preventivo de los mismos. En consecuencia, la entidad deberá crear en cada uno de sus procedimientos la base del control de los recursos, estableciendo mecanismos donde tanto el trabajador como los funcionarios logren interiorizar el nivel de responsabilidad que les corresponde en cada caso (Francois, 2006).

El Control Interno puede ayudar a que una entidad consiga sus objetivos, a prevenir las pérdidas de recursos y a la obtención de la información más oportuna y eficaz. También puede reforzar la confianza de que la entidad cumple con las leyes y normas aplicables, evitando efectos perjudiciales para su reputación y otras consecuencias. En resumen, puede ayudar a que una entidad llegue adonde quiere ir y evite peligros y sorpresas en el camino (De Castro y García del Junco, 2001; Quirós, 2005).

El Control Interno es de mucha importancia, porque permite:

- El uso eficiente de los recursos humanos, materiales y financieros, asegurando su integridad, custodia y registro oportuno, con base en sistemas integrados de administración y finanzas.
- Contar con la información administrativa y financiera, oportuna, correcta y confiable para la toma de decisiones y la rendición de cuentas de la gestión de los funcionarios públicos.
- Detectar los registros de errores e irregularidades como base para identificar sus causas y promover acciones para eliminar las debilidades de control existentes.

- Promover un grado razonable de efectividad, eficiencia y economía en la administración y uso de los recursos públicos.
- Promover el cumplimiento de las políticas dictadas por la máxima autoridad, así como las leyes y reglamentos aplicables para alcanzar las metas y objetivos programados (Zavaro, 2000).

Son muchos los beneficios que para la entidad cubana tiene la implementación del nuevo concepto de Control Interno a través de sus componentes y con variados matices: un cambio en la mentalidad de los directivos y todos los trabajadores en cuanto al alcance, el enfoque y los objetivos del Control Interno y las responsabilidades de todos para con el sistema; mejor organización del trabajo al quedar establecidos los manuales de organización y el diseño de cada puesto de trabajo; poseer indicadores para evaluar el desempeño de cada área funcional y cada puesto de trabajo; contribuir al autocontrol; y fortalecer la autoridad y el liderazgo (Chiavenatto, 2005).

En Cuba se realiza todos los años entre los meses de abril y mayo la Comprobación Nacional al Sistema de Control Interno en las entidades donde participan todos los auditores del Sistema de Auditoría con los objetivos principales de:

- Evaluar las exigencias y efectividad de los resultados de la aplicación de la Guía de Autocontrol del Sistema de Control Interno. Verificar si contribuyó al perfeccionamiento e implementación del sistema (Estrada, 2004).
- Comprobar la efectividad del Sistema de Control Interno a partir del cumplimiento de los objetivos de control interno establecidos en la Resolución No. 297/03 del Ministerio de Finanzas y Precios, que regula las “Definiciones de Control Interno y el Contenido de los Componentes y Normas”.
- Verificar el cumplimiento de la Resolución No. 13/2006 del extinto Ministerio de Auditoría y Control, que establece las “Indicaciones para la elaboración y sistemático control del plan de medidas para la prevención de indisciplinas, ilegalidades y manifestaciones de corrupción”, para corroborar la efectividad de las medidas previstas (Hernández, 2003).

- Conciliar los resultados de la auditoría con los puntos vulnerables identificados en el Plan de Prevención, para corroborar el cumplimiento de las medidas previstas (Castillo Castillo, 2010).

1.2.4 Control Interno en la Universidad de Granma.

En la Universidad de Granma el control interno tiene un espacio importante dentro del proceso de control económico pues se hace necesario inspeccionar los recursos con los que se cuenta; para ello, en el departamento de administración económica existen dos técnicos en gestión económica que entre sus funciones cuenta la de controlar el trabajo que realiza cada administrador de las Áreas o Facultades de la Universidad y periódicamente efectúan auditoría en aras de detectar deficiencias o infracciones que entorpezcan el correcto mecanismo de control de los recursos del estado.

En el proceso de auditoría, cada administrador debe presenciarse con su control interno mensual que es un valor representativo al 10% de los medios básicos, esta información se controla a través de un modelo de control.

En el control interno se controlan los medios básicos que entran, salen y los que se encuentran en la Universidad. La Universidad puede adquirir Medios Básicos mediante compra, donación, asignación del Ministerio de Educación Superior ó por convenio de proyecto. Una vez que el Medio Básico se encuentra en la Universidad se asigna a un Área o Facultad según necesidad. Cuando un Medio Básico sale del Local al que pertenece puede ser trasladado a otro Local, otra Área o Facultad ó fuera de la Universidad pero siempre en beneficio del proceso docente-educativo, llamándosele al modelo que recoge esta información *Movimiento de Medio Básico*. Una vez que los Medios Básicos están en las diferentes Áreas o Facultades periódicamente se le realizan mantenimientos, el mantenimiento puede ser por rotura o porque le corresponda al Medio Básico según calendario de mantenimiento, el modelo que recoge la información del servicio prestado al Medio Básico se nombra *AT-5*.

El departamento de administración económica de la Universidad procesa toda la información del control interno mediante un sistema contable llamado ASSETS,

mientras que los administradores de las Áreas o Facultades lo procesan manualmente o apoyándose en el tabulador electrónico Microsoft Excel.

1.3 Sistemas para la Gestión de Procesos Contables.

1.3.1 Sistema Manual.

Los procesos contables se controlaban de forma manual anteriormente, la información se almacenaba en libros, y todo el proceso de gestión se realizaba utilizando un sin número de materiales, como hojas de modelo de movimiento, bolígrafos, calculadoras, entre otros. El personal administrativo aplicaba una disciplina contable, para imponer un orden lógico a las diferentes transacciones hasta donde fuera posible, clasificaba para registrar y documentar aquellas transacciones con el objetivo de llevar un control óptimo (Puerres, 2007).

1.3.2 Herramienta Informática

Dentro de los Sistemas de Gestión para las aplicaciones económico-contable, se encuentra el **ASSETS**, sistema para la Gestión Integral estándar y parametrizado, es un sistema italiano que controla la actividad económica, contable y de Recursos Humanos, perteneciente al Ing. Informático Marco de Lucas con dos versiones; la primera sobre Microsoft Access y la segunda versión conocida como “Nueva Solución” usa el modelo cliente-servidor, como gestor de base de datos Microsoft SQL Server 2000 y Crystal Reports 8.5 para la generación de reportes de salidas.(ASSETS, 2006; Cano Hidalgo, 2006).

El ASSETS se encuentra bastante generalizado en el país, está presente en los Ministerios de Educación Superior, de Justicia y de Auditoría y Control, la Aduana General de la República, el Consejo de Estado y el Palacio de las Convenciones existiendo muchos clientes, incluyendo la Universidad de Granma como se expresa en el epígrafe 1.2.4.

El ASSETS es un software propietario que no posibilita el acceso o publicación de un gran número de informaciones útiles para el personal no especializado en Recursos Humanos, es un sistema orientado estrictamente a los locales de trabajo de los especialistas en la materia, careciendo por lo tanto de una interfaz que permita un flujo de informaciones haciendo nula la comunicación externa.

Aunque este sistema es un apoyo importante en el Control de Interno específicamente en el control de Inventario, después de analizar sus características y estudiar la experiencia de su empleo en la Universidad de Granma, se considera que existe una serie de limitaciones asociadas a su uso, las fundamentales se argumentan a continuación:

1. *Portabilidad.* Al ser una aplicación desarrollada para Windows, sólo podrá ser ejecutada desde estaciones de trabajo con este sistema operativo (SO) instalado, lo que limita grandemente la posibilidad de utilizar otros SO.
2. *Accesibilidad.* Es preciso tener la aplicación cliente instalada para acceder al sistema.
3. *Distancia del proveedor.* El proveedor oficial del sistema en Cuba (INFOMASTER), radica en La Habana, esto condiciona que la rapidez con que puede dar respuesta a los problemas operativos que se presentan en la institución e involucren de alguna manera al sistema es muy limitada.
4. *Propiedad.* al ser un “software propietario” es imposible para la Universidad de Granma, aún cuando cuente con los recursos humanos para hacerlo, realizarle modificaciones que lo adapten a sus requerimientos particulares.

1.4 Herramientas para el desarrollo del producto informático.

Para el desarrollo del sistema se realizó un estudio sobre las posibles herramientas a utilizar en su construcción, teniéndose en cuenta la tendencia actual y las novedades de cada una de ellas.

1.4.1 Aplicaciones Web.

Una Aplicación Web es un conjunto de páginas Web enlazadas que visualizan diferentes partes de la información que se quiere mostrar a través de ella. Constituye una de las mejores herramientas para divulgar, gestionar y compartir la información por lo que trae consigo un aumento de la eficiencia en cuanto a la manipulación de gran cantidad de información. Permiten modificar la lógica del negocio a través de un navegador. (Hernán Ruiz, 2006; Cinca, 2008).

Surgimiento de las Aplicaciones Web.

La World Wide Web: o simplemente "la Web", puede definirse básicamente como tres cosas: hipertexto, que es un sistema de enlaces que permite saltar de unos lugares a otros; multimedia, que hace referencia al tipo de contenidos que puede manejar (texto, gráficos, vídeo, sonido y otros) e Internet, las base sobre las que se transmite la información (Furgeta, 2002).

La WWW permite visualizar en la pantalla del usuario páginas con información alojadas en computadoras remotas (llamadas genéricamente sitios). En la actualidad, existen dos tipos de sitios Web: los que se comportan como magazines, donde solo se puede leer la información que en él esté escrita; y los que se comportan a manera de software, donde se ejecutan una series de tareas específicas. Estos últimos son las llamadas aplicaciones Web. (Pressman, 1996; Sanz, 2002; Cinca, 2008).

La arquitectura de un sitio Web es simple. Contiene como componentes principales: el Servidor Web, una Red y un Navegador o cliente. La arquitectura de una aplicación Web además incluye la aplicación en el Servidor, que es la que permite al sistema manejar lógica de negocio y tener un estado (Bellman, 2007).

Con el uso de las aplicaciones Web en cualquier entidad se logra, entre otros beneficios, una alta eficiencia que se alcanza en el área que haga uso de los mismos. Debido a una serie de características que ellos poseen, se mejoran los mecanismos de intercambio de información en la entidad, disminuyendo en un alto grado cualquier dificultad que exista para difundir la misma (Villa, 2006).

Entre las principales comodidades que una aplicación Web pueden traer se encuentran las siguientes:

- **Compatibilidad multiplataforma.** Las aplicaciones Web tienen un camino mucho más sencillo para la compatibilidad multiplataforma que las aplicaciones de software descargables. Varias tecnologías incluyendo Java, Flash, ASP y Ajax permiten un desarrollo efectivo de programas soportando todos los sistemas operativos principales (Colombo, Demetrescu, I. y Laura, 2003).

- **Actualización.** Las aplicaciones basadas en Web están siempre actualizadas con el último lanzamiento sin requerir que el usuario tome acciones pro-activas, y sin necesitar llamar la atención del usuario o interferir con sus hábitos de trabajo con la esperanza de que va a iniciar nuevas descargas y procedimientos de instalación (algunas veces imposible cuando usted está trabajando dentro de grandes organizaciones) (Oliver, 2000).
- **Inmediatez de acceso.** Las aplicaciones basadas en Web no necesitan ser descargadas, instaladas y configuradas. Usted accede a su cuenta online y priistas están listas para trabajar sin importar cuál es su configuración o su hardware.
- **Facilidad de prueba.** Finalmente no habrá más obstáculos para permitir pruebas sencillas y efectivas de herramientas y aplicaciones antes de cargar su tarjeta de crédito. Actualmente, especialmente cuando hablamos de software costoso, hay todavía una gran cantidad de funcionalidades y pequeños detalles que no pueden ser totalmente probados descubiertos antes de comprometer dinero en alguna compra total.
- **Menos requerimientos de memoria.** Las aplicaciones basadas en Web tienen muchas más razonables demandas de memoria RAM de parte del usuario final que los programas instalados localmente. Al residir y correr en los servidores del proveedor, a esas aplicaciones basadas en Web usa en muchos casos la memoria de las computadoras que ellos corren, dejando más espacio para correr múltiples aplicaciones del mismo tiempo sin incurrir en frustrantes deterioros en el rendimiento.
- **Menos Bugs.** Las aplicaciones basadas en Web, deberían ser menos propensas a colgarse y crear problemas técnicos debido a software o conflictos de hardware con otras aplicaciones existentes, protocolos o software personal interno. Con aplicaciones basadas en Web, todos utilizan la misma versión, y todos los bugs pueden ser corregidos tan pronto como son descubiertos. Esta es la razón por la cual las aplicaciones basadas en Web deberían tener mucho menos bugs que el software de escritorio descargable tradicional (Zelkovitz, Shaw y Gannon, 2009).

- **Precio.** Las aplicaciones basadas en Web no requieren la infraestructura de distribución, soporte técnico y marketing requerido por el software descargable tradicional. Esto permite que las aplicaciones online cuesten una fracción de sus contrapartes descargables si no totalmente gratuitas, mientras que ofrecen componentes adicionales y servicios Premium como una opción.
- **Múltiples usuarios concurrentes.** Las aplicaciones basadas en Web puedan realmente ser utilizada por múltiples usuarios al mismo tiempo. No hay más necesidad de compartir pantallas o enviar instantáneas cuando múltiples usuarios pueden ver e incluso editar el mismo documento de manera conjunta. Las compañías de conferencia Web y colaboración online están involucradas algunas transformaciones claves y los usuarios necesitan explorar que significa realmente trabajar efectivamente y co-editar documentos juntos (Brusilovsky et al., 2007).
- **Desarrollar aplicaciones en el lenguaje que usted quiera.** Una vez que las aplicaciones han sido separadas de computadoras locales y sistemas operativos específicos esos pueden también ser escritas en prácticamente cualquier lenguaje de programación. Ya que las aplicaciones Web son esencialmente una colección de programas más que un simple programa, ellas podrían ser escritas en cualquier lenguaje de programación que esté de aplicación Web. Mientras que para software escritorio usted está limitado a usar el mismo lenguaje que el sistema operativo subyacente este no es el caso cuando la aplicación de software. (Brusilovsky, Gilles, Frasson y VanLehn, 2007).

Usando la Web, se tiene acceso a millones de páginas de información, la exploración se realiza por medio de un software especial denominado Browser o Explorador. Las aplicaciones Web son populares debido a la practicidad del navegador Web como cliente ligero (Prentzas, Hatzilygeroudis y Garofalakis, 2002; Conallen, 2006).

Internet

Internet es un conjunto de redes de ordenadores y equipos físicamente unidos mediante cables que conectan puntos de todo el mundo. Se difumina en

ocasiones porque los datos pueden transmitirse vía satélite, o a través de servicios como la telefonía celular, o porque a veces no se sabe muy bien a dónde está conectada. Internet es el mayor conjunto que existe de información, personas, ordenadores y software funcionando de forma cooperativa, publicando y organizando información, e interactuando a nivel global.

El desarrollo de Internet no solo se traduce en beneficio para los usuarios, sino que también lo es para los organismos, las instituciones, las empresas etc. Internet se ha convertido en uno de los medios de comunicación más extendido en toda la historia de la humanidad. Está bien llamada "Red de redes", permite establecer la cooperación y colaboración entre gran número de comunidades y grupos de interés por temas específicos, distribuidos por todo el planeta (Lynch, 2003).

Intranet

Es una infraestructura basada en los estándares y tecnologías de Internet que soporta el compartir información dentro de un grupo bien definido y limitado. Las Intranets son Internets internas basadas en particular en el sistema WWW, pero adaptándolo a los límites físicos y a las características particulares de cada organización (Herrera, Sandoval y Quijada, 2009).

Una Intranet es una alternativa para las empresas que necesiten compartir la información que se genera en ella. Es utilizar y distribuir la información dentro de la empresa pero a la manera de Internet. Es una red de comunicaciones dentro de una empresa, que permite integrar una serie importante de recursos, pudiendo así obtener la máxima eficiencia.

1.4.1.1 El modelo cliente/servidor.

Las aplicaciones Web siguen el modelo cliente-servidor que es un sistema de "inteligencia repartida" en el que el cliente establece una conexión con el servidor, para recurrir a los servicios que ofrece. Algunas aplicaciones de servidor pueden servir al mismo tiempo a varios clientes, ahorrando a cada uno de ellos el problema de tener la información instalada y almacenada localmente. (Wallace, Aufgang y Raggett, 2002).

El cliente es la aplicación que establece conexiones con los servidores para aprovechar los servicios correspondientes. El servidor toma conexiones de los clientes y pone a disposición sus servicios.

En el modelo Cliente-Servidor el servidor Web aporta ventajas adicionales en aspectos de relevancia como son:

- Muestran información de forma sencilla a través de un hipervínculo.
- Es posible visualizar datos, documentos, multimedia por medio de los estándares de Internet.
- Debido a la facilidad con que se puede acceder al servidor Web, centralizando la información, se puede obtener de forma inmediata (King, 2007).

1.4.2 Lenguajes de Programación.

En la programación de la Web se habla de dos tipos de lenguajes: los lenguajes del *lado servidor* que son aquellos lenguajes que se caracterizan por desarrollar la lógica de negocio dentro del servidor. Además de ser los encargados del acceso a base de datos, tratamiento de la información, etc. Por otro lado, los lenguajes de *lado cliente*, que son los encargados de aportar dinamismo a la aplicación en los navegadores (Furgeta, 2002).

Entre los lenguajes del lado del servidor se pueden encontrar entre los más sobresalientes al Active Server Pages (ASP), Personal Home Pages (PHP), Java Server Pages (JSP). Del lado del cliente se encuentra el HTML, Java y JavaScript los cuales son simplemente incluidos en el código HTML (Aldana, 2009).

1.4.2.1 Tecnologías de programación del lado del cliente.

Hypertext Markup Language (HTML).

HTML es el lenguaje de la Web, estas siglas significan Hiper Text Markup Language (Lenguaje de Marcas de Hipertextos), en donde hipertexto hace referencia a la capacidad del lenguaje para la utilización de los conocidos hipervínculos. Se basa en el uso de etiquetas o marcas, gracias a ellas es posible definir el formato del texto, los distintos elementos que conforman la página, sus propiedades y disposición. Este lenguaje es interpretado por los navegadores, procesado pasados y convertidos en una página Web (Málaga, 2008).

Una página Web o documento HTML no es otra cosa que un archivo de texto que se puede crear con cualquier editor; desde el bloc de notas Windows hasta cualquiera de DOS o LINUX.

Fue creado en 1986 por el físico nuclear Tim Berners-Lee; el cual tomo dos herramientas preexistentes: El concepto de Hipertexto (Conocido también como link o ancla) el cual permite conectar dos elementos entre si y el SGML (Lenguaje Estándar de Marcación General) el cual sirve para colocar etiquetas o marcas en un texto que indique como debe verse. HTML no es propiamente un lenguaje de programación como C++, Visual Basic, etc., sino un sistema de etiquetas. HTML no presenta ningún compilador, por lo tanto algún error de sintaxis que se presente éste no lo detectará y se visualizara en la forma como éste lo entienda (Hernán Ruiz, 2006).

Java Script

JavaScript es un lenguaje de scripts desarrollado por Netscape para incrementar las funcionalidades del lenguaje HTML. Se utiliza embebido en el código HTML, entre las tags `<script>` y `</script>`. Sus características más importantes son: [JS-a] [JS-b]

Es un lenguaje interpretado basado en guiones, es decir, no requiere compilación. El navegador del usuario se encarga de interpretar las sentencias Java Script contenidas en una página HTML y ejecutarlas adecuadamente (Colombo et al., 2003).

Java Script es un lenguaje orientado a objetos. El modelo de objetos de Java Script está reducido y simplificado, pero incluye los elementos necesarios para que los Scripts puedan acceder a la información de una página y puedan actuar sobre la interfaz del navegador.

Cuando un usuario selecciona un enlace o mueve el puntero sobre una imagen se produce un evento. Mediante JavaScript se pueden desarrollar Scripts que ejecuten acciones en respuesta a estos eventos. En este caso actúa como un lenguaje orientado a eventos (Málaga, 2008).

Es además un lenguaje de alto nivel multiplataforma que admite programación estructurada y no necesita ningún kit o entorno de desarrollo.

Java

Java es un lenguaje de programación clásico en cuanto a que requiere un proceso de compilación. El código compilado puede ser integrado en la página Web para ser ejecutado por el cliente.

Su nacimiento data de enero de 1996, tras el lanzamiento por la empresa creadora Sun Microsystem del JDK 1.0 (Java Development Kit). Es un entorno que puede obtenerse de forma gratuita e incorpora las herramientas básicas para la creación de una aplicación en Java (JDK, 2009).

Con Java se pueden crear dos tipos de programas:

- Applets: programas que se integran en las páginas Web y que, residiendo en el servidor, son ejecutados por el cliente. La ejecución necesita de la interpretación del código compilado por el software cliente (ASKALL ORG, 2007).
- Aplicaciones: Programas autónomos que se pueden ejecutar en cualquier equipo. En este último caso puede optarse por generar código compilado similar al de los applets y que para su ejecución necesita de un intérprete o código compilado ejecutable directamente como en cualquier otro lenguaje de programación.

En el caso de applets, el código fuente no se incrusta directamente en el documento HTML, si no al que se lo añade es un código binario resultado de la compilación, el denominado JBC (Java Byte Code). Esto permite proteger el código fuente, aunque hasta cierto punto, ya que las particularidades de este código compilado hacen que sea factible el proceso inverso, es decir, la decompilación, recuperar el código fuente a partir del compilado. En la propia Internet pueden encontrarse programas capaces de hacerlo. La razón de todo esto está en el hecho de que para conseguir la portabilidad de los programas el código compilado es un código que se encuentra entre el código fuente y un código objeto fuertemente dependiente de una plataforma. Es por ello que se suele decir

de Java que es un lenguaje que combina la flexibilidad de los lenguajes interpretados y el poder de los compilados (JBC, 2009).

VBScript

VBScript es, al igual que JavaScript, un lenguaje basado en guiones que permite integrar programas directamente en el código HTML. Admite un doble uso, por un lado como lenguaje del lado del cliente, pero también como lenguaje del lado del servidor para la generación de páginas ASP.

Es un lenguaje desarrollado por Microsoft tomando como referente de sintaxis el VBA, Visual Basic para aplicaciones. Por supuesto, no ofrece todas las funcionalidades de un entorno de desarrollo visual como Visual Basic, pero sí se presenta como una herramienta poderosa y de fácil uso para generar páginas Web interactivas (ASKALL ORG, 2007).

1.4.2.2 Tecnologías de programación del lado del servidor.

Es aquella en que los lenguajes son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él.

Professional Home Page Tools (PHP)

PHP (Profesional Home Page Tools) es un lenguaje de programación el cual se ejecuta en los servidores Web y que permite crear contenido dinámico en las páginas HTML, con un lenguaje propietario derivado del Perl (ANTARESX, 2004).

PHP fue creado por Rasmus Lerdorf a finales de 1994, aunque no hubo una versión utilizable por otros usuarios hasta principios de 1995. Esta primera versión se llamó, Personal Home Page Tools.

Al principio, PHP sólo estaba compuesto por algunas macros que facilitaban el trabajo a la hora de crear una página Web. Hacia mediados de 1995 se creó el analizador sintáctico y se llamó PHP/F1 Versión 2, y sólo reconocía el texto HTML y algunas directivas de MySQL. A partir de este momento, la contribución al código fue pública. El crecimiento de PHP desde entonces ha sido exponencial, y han surgido versiones nuevas como los actuales, PHP3, PHP4 y PHP5.

Dispone de múltiples herramientas que permiten acceder a bases de datos de forma sencilla, por lo que es ideal para crear aplicaciones para Internet (Cobo, Gómez, Pérez y Rocha, 2010).

Es multiplataforma, funciona tanto para Unix (con Apache) como para Windows (con Microsoft Internet Information Server) de forma que el código que se haya creado para una de ellas no tiene porqué modificarse al pasar a la otra.

El lenguaje PHP es un lenguaje de programación de estilo clásico, con variables, sentencias condicionales, bucles, funciones, entre otras. La sintaxis que utiliza, la toma de otros lenguajes muy extendidos como C y Perl (Figura 1.1).



Figura 1.1. Esquema de representación del PHP.

El funcionamiento del PHP se puede describir a través de los pasos siguientes:

- Escribir en las páginas HTML pero con el código PHP dentro.
- Guardar la página en el servidor Web.
- Un navegador solicita una página al servidor.
- El servidor interpreta el código PHP.
- El servidor envía el resultado del conjunto de código HTML y el resultado del código PHP que también es HTML (Cobo et al., 2010).

En ningún caso se envía código PHP al navegador, por lo que todas las operaciones realizadas son transparentes al usuario, el código PHP es ejecutado en el servidor y el resultado enviado al navegador. El resultado es normalmente una página HTML. Por lo que al usuario le parecerá que está visitando una página HTML que cualquier navegador puede interpretar.

Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que el navegador lo soporte, es independiente del navegador, pero sin embargo para que sus páginas PHP funcionen, el servidor donde están alojadas debe soportar PHP (Gracia, 2006; Álvarez, 2007).

Active Server Page (ASP)

ASP (*Active Server Pages*), Páginas Activas en el Servidor, es una tecnología creada por Microsoft, destinada a la creación de sitios Web. No se trata de un lenguaje de programación en sí mismo (ya que las ASP se pueden programar en VBScript, JavaScript, PerlScript), sino de un marco sobre el cual construir aplicaciones basadas en Internet (Morell, 2006).

ASP añade otra alternativa en sus posibles opciones para el desarrollo de las funcionalidades del lado del servidor. ASP le permite combinar HTML y código Script en el servidor para crear páginas Web dinámicas y altamente interactivas. (Villa, 2008).

El paradigma de desarrollo de ASP difiere en gran medida de la programación Script del lado del cliente, ya que en esta última, el Script se incrusta dentro de la página que es enviada al usuario, este a su vez, es ejecutado por el navegador, que por supuesto debe soportar el uso del lenguaje Script particular para poder ejecutarlo. Si el navegador no reconoce el lenguaje del Script, entonces ignorará el código. Por el contrario, con ASP, todos los Scripts son procesados en el servidor y los resultados son retornados al cliente en formato HTML estándar, reconocible por cualquier navegador (Colombo et al., 2003; Málaga, 2008).

La mayor desventaja que presenta este lenguaje es que solo se puede implementar en los servidores Web de su desarrollador: Microsoft. Actualmente se ha presentado ya la segunda versión de ASP: el ASP.NET, que comprende algunas mejoras en cuanto a posibilidades del lenguaje y rapidez con la que funciona. ASP.NET tiene algunas diferencias en cuanto a sintaxis con el ASP, de modo que se ha de tratar de distinta manera uno de otro. Para implantarlo es necesario montar en el servidor la plataforma (Wallace et al., 2002; Morell, 2006).

Comparación entre el PHP y ASP

Luego de hacer el análisis entre el PHP y el ASP, se decide utilizar *PHP* porque:

- Está soportado en la mayoría de las plataformas de Sistemas Operativos, mientras que con ASP por ser propiedad de Microsoft no es multiplataforma.
- PHP es un lenguaje interpretado del lado del servidor que surge dentro de la corriente denominada código abierto (*open source*). Se caracteriza por su potencia, versatilidad, robustez y modularidad. Al igual que ocurre con tecnologías similares, los programas son integrados directamente dentro del código HTML (Málaga, 2008).
- El PHP no tiene costo oculto, o sea que cuando se adquiere incluye un sinnúmero de bibliotecas que proporcionan el soporte para la mayoría de las aplicaciones Web, por ejemplo e-mail, generación de ficheros PDF y otros. En caso de que no se tengan las bibliotecas están se pueden encontrar gratis en Internet. En el caso de ASP forma parte del Internet Information Server que viene integrado en Windows Server con su elevado costo de adquisición. (ANTARESX, 2004).
- PHP y ASP son parecidos en cuanto a la forma de utilización, pero PHP es más rápido, gratuito y multiplataforma (Cobo et al., 2010).
- La principal ventaja de PHP es su carácter multiplataforma. Por otro lado, los programas en ASP resultan más lentos y pesados, y también menos estables.
- PHP, es uno de los lenguajes más utilizados actualmente en el desarrollo de las aplicaciones Web y viene experimentado un constante crecimiento en su nivel de utilización en internet.

1.4.3 Servidor Web.

Un servidor Web es un programa que implementa el protocolo HTTP. Este protocolo está diseñado para transferir lo que se llama hipertextos, páginas Web o páginas HTML: textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de sonido (Valles, 2008).

Un servidor Web se encarga de mantenerse a la espera de peticiones HTTP llevada a cabo por un cliente HTTP o navegador. El navegador realiza una

petición al servidor y este le responde con el contenido que el cliente solicita. Al solicitar una dirección de una página Web en un navegador, este realiza una petición HTTP al servidor de dicha dirección. El servidor responde al cliente enviando el código HTML de la página. El cliente, una vez recibido el código, lo interpreta y lo muestra en pantalla. Entre los servidores Web más importantes se encuentra el Apache e Internet Information Server (IIS). Sobre ambos de amplia a continuación.

1.4.3.1 El servidor Web Internet Information Server (IIS).

El servidor Web Internet Information Server (IIS), es un poderoso servidor Web altamente confiable y fácil de manejar, es el más indicado para trabajar con páginas ASP. Su configuración es muy sencilla y se realiza siempre mediante una interfaz visual (Microsoft, 2009).

Resulta ideal para Intranets ya que puede ser un servidor Web de gran potencia en el hardware ya existente en las empresas. Está integrado en el sistema operativo Windows a partir de la versión Windows 2000 y aprovecha sus características de seguridad y capacidades de rendimiento.

IIS sólo funciona bajo servidores Windows y debe ser usado bajo licencia, o sea, hay que pagar para utilizarlos, además la mayoría de funcionalidad extra debe ser comprada. Esto no es provechoso para Cuba, ni para nuestra Universidad, ya que serían muchos gastos adicionales, teniendo otra opción muy buena y poderosa que es gratis como Apache (Aldana, 2009).

Ante problemas de seguridad, que son usuales en IIS, la compañía Microsoft suele ser lenta en corregirlos e incluso ineficaz a la hora de resolverlos, mientras que los desarrolladores de Apache, resuelven los problemas mucho más rápido. IIS tiene una interfaz muy intuitiva con cuadros de diálogos y ventanas, pero a costa de una integración total en el sistema, que hace difícil distinguir qué opciones afectan al servidor y cuáles al Sistema Operativo. Además utiliza más espacio en el disco duro. Para IIS lo ideal es usar ASP, no PHP, ya que el PHP es mejor integrarlo con Apache.

1.4.3.2 El servidor Web Apache.

Apache es el servidor Web más difundido y utilizado en Internet. Según las estadísticas de servidores de Netcraft, el 63.7 % de los sitios activos de Internet utilizan Apache, contra un 27.39 % que utilizan el servidor de Microsoft y el resto repartido entre otros (Aldana, 2009).

Tal vez lo que hace más atractivo a Apache es su alta estabilidad, seguridad y facilidad de expansión, aparte de su costo, ya que es Open Source, lo que significa que su código es libre y gratuito. Esto último significa que se distribuye por medio de una licencia gratuita y de código abierto.

Apache se encuentra disponible para una gran cantidad de sistemas operativos, entre los que se encuentran Linux, BeOS, FreeBSD, HP UNIX, Mac OS X, OS/ 2, QNX, Solaris, Rhapsody, Sun OS, UNIXware y Windows, entre otros.

Su nombre se debe a que originalmente Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. Era, en inglés, a patchy server (un servidor parcheado). Se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation (Aldana, 2009).

Apache presenta entre otras características, mensajes de error altamente configurables, BD de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

Tiene capacidad para servir páginas tanto de contenido estático, para lo que solo se necesitaría un viejo ordenador 486, como de contenido dinámico a través de otras herramientas soportadas que facilitan la actualización de los contenidos mediante BD, ficheros u otras fuentes de información.

1.4.4 Diseño de interfaz

Las páginas que conforman las aplicaciones con tecnología Web pueden ser diseñadas y programadas por diversos editores. Esto facilita su desarrollo y minimiza el tiempo de trabajo. Algunos de estos editores son el Zend Studio y el Macromedia Dreamweaver 8. El Dreamweaver permite diseñar de forma visual, el Zend Studio se utiliza para la programación de las páginas de las aplicaciones. Sobre estos editores se amplía a continuación.

1.4.4.1 Zend Studio.

El Zend Studio es una potente plataforma para desarrollar y darle mantenimiento a aplicaciones programadas en PHP. Ofrece muchas mejoras para los programadores de aplicaciones Web. Hace más fácil entender la codificación utilizada en la aplicación, tiene una buena integración con el navegador, ya que desde este se puede acceder al Zend para seguir el flujo de la ejecución de la página. El programa, además de servir de editor de texto para páginas PHP, proporciona una serie de ayudas que pasan desde la creación y gestión de proyectos hasta la depuración de código.

Lo más destacable es que contiene una ayuda contextual con todas las librerías de funciones del lenguaje que asiste en todo momento ofreciendo nombres de las funciones y parámetros que deben recibir. Aunque esta ayuda contextual no solo se queda en las funciones definidas en el lenguaje, sino que también reporta ayudas con las funciones que se vayan creando, incluso en páginas que se tengan incluidas.

1.4.4.2 Macromedia Dreamweaver 8

Sin lugar a dudas, una de las herramientas más utilizadas por los Webmasters que trabajan con aplicaciones visuales es Dreamweaver, de Macromedia. El programa se adapta increíblemente a las necesidades de todo tipo de profesional del diseño Web, tanto para los que prefieren programar el código directamente en el editor de texto como para los que gustan de una metodología de trabajo totalmente visual (Taylor, 2009).

Macromedia Dreamweaver 8 es un editor de texto común y corriente para trabajar en el diseño de páginas Web, pero las alternativas que brinda hace que los Webmaster puedan olvidarse de las partes más tediosas del diseño, como la disposición de capas, tablas, formularios y demás elementos.

Las funciones de edición visual de Dreamweaver permiten crear páginas Web de forma rápida, sin escribir una sola línea de código. Puede ver todos los elementos o activos del sitio y arrastrarlos desde un panel fácil de usar directamente hasta un documento. Puede agilizar el flujo de trabajo de desarrollo mediante la creación y edición de imágenes en Macromedia Fireworks o en otra aplicación de

gráficos y su posterior importación directa a Dreamweaver. Dreamweaver también contiene herramientas que facilitan la adición de activos de Flash a las páginas Web (Taylor, 2009).

Además de las funciones de arrastrar y soltar que le ayudan a crear páginas Web, Dreamweaver le ofrece un entorno de codificación con todas las funciones, que incluye herramientas para la edición de código (tales como coloreado de código, terminación automática de etiquetas, barra de herramientas para codificación y contracción de código) y material de referencia para lenguajes sobre hojas de estilos en cascada (CSS), JavaScript y ColdFusion Markup Language (CFML) entre otros. La tecnología Roundtrip HTML de Macromedia importa los documentos con código manual HTML sin modificar el formato del código. Posteriormente, si lo desea, puede formatear el código con el estilo que prefiera.

Además, Dreamweaver le permite crear aplicaciones Web dinámicas basadas en bases de datos empleando tecnologías de servidor como CFML, ASP.NET, ASP, JSP y PHP. Si prefiere trabajar con datos en XML, Dreamweaver incorpora herramientas que le permiten crear fácilmente páginas XSLT, adjuntar archivos XML y mostrar datos XML en sus páginas (Taylor, 2009).

Dreamweaver se puede personalizar totalmente. Puede crear sus propios objetos y comandos, modificar métodos abreviados de teclado e incluso escribir código JavaScript para ampliar las posibilidades que ofrece Dreamweaver con nuevos comportamientos, inspectores de propiedades e informes de sitios.

Dreamweaver 8 incorpora muchas funciones nuevas que le ayudarán a diseñar páginas Web y aplicaciones con un mínimo de tiempo y esfuerzo. Dreamweaver simplifica las tecnologías más complejas y las hace accesibles, ayudándole a conseguir más en menos tiempo.

Funciones de Dreamweaver 8.

- Herramienta Zoom y guías.
- Vinculación de datos visual de XML.
- Nuevo panel de estilos CSS.
- Visualización de diseño CSS.
- Contracción de código.

- Barra de herramientas de codificación.
- Transferencia de archivos en segundo plano.
- Inserción de comando de Flash Video (Taylor, 2009).

1.4.5 Sistemas Gestores de Base de Datos (SGDB)

Un Sistema de Gestión de Bases de Datos (SGBD) consiste en un conjunto de programas, procedimientos y lenguajes que nos permiten, proporcionándonos las herramientas necesarias, definir, manipular y utilizar la información que contienen las bases de datos; realizar todas las tareas de administración necesarias para mantenerlas operativas, mantener su integridad, confidencialidad y seguridad. Incorporar una serie de funciones que permita definir los registros, sus campos, sus relaciones, insertar, suprimir, modificar y consultar los datos (I. M. Lockhart, 2009).

Se puede considerar al SGBD como el interfaz entre el usuario y la base de datos. Para comunicarse con un SGBD, tanto para definir datos y estructuras como para hacer consultas sobre los datos se puede utilizar SQL (*Structured Query Language*), que no es más que un lenguaje de consultas estructurado compuesto por comandos, cláusulas, operadores y funciones de agregado. Estos elementos se combinan en grupos de instrucciones (Consultas) para actualizar y manipular las bases de datos.

Entre los SGBD comúnmente utilizados en el mundo tenemos Oracle, MySQL, Microsoft SQL Server, PostgreSQL, InterBase, entre otros. Todos estos presentan un enfoque relacional con un buen basamento matemático centrado en el Álgebra Relacional. A continuación se realiza un análisis de las principales características y facilidades de dos de estos SGBD para determinar cual utilizar.

Un SGBD tiene los siguientes objetivos específicos:

- Independencia de los datos y los programas de aplicación.
- Minimización de la redundancia.
- Integración y sincronización de las bases de datos.
- Integridad de los datos.
- Seguridad y protección de los datos.

- Facilidad de manipulación de la información.
- Control centralizado (I. M. Lockhart, 2009).

1.4.5.1 SQL Server

Microsoft SQL Server, propietario de Microsoft, pertenece a la familia de los sistemas de administración de base de datos, operando en una arquitectura cliente/servidor de gran rendimiento. Su desarrollo fue orientado para hacer posible manejar grandes volúmenes de información, y un elevado número de transacciones. SQL Server es una aplicación completa que realiza toda la gestión relacionada con los datos. El servidor sólo tiene que enviarle una cadena de caracteres (la sentencia SQL) y esperar a que le devuelvan los datos (SQL, 2007).

SQL Server permite la creación de procedimientos almacenados, los cuales consisten en instrucciones SQL que se almacenan dentro de una base de datos de SQL Server, realizados en lenguaje SQL, se trata de procedimientos que se guardan semicompilados en el servidor y que pueden ser invocados desde el cliente. Se ejecutan más rápido que instrucciones SQL independientes.

SQL Server puede manejar perfectamente bases de datos de TeraBytes con millones de registros y funciona sin problemas con miles de conexiones simultáneas a los datos, sólo depende de la potencia del hardware del equipo en el que esté instalado.

1.4.5.2 MySQL

MySQL es un servidor de base de datos muy rápido, robusto, multitarea y multiusuario. Tiene enfoque relacional y soporta AnsiSQL. Soporta clientes en C, C++, Eiffel, Java, Perl, PHP, Python y Tcl. Trabaja en diferentes plataformas además de soportar múltiples idiomas Completo y optimizado uso del SQL y contiene un MyODBC (Pérez, 2008).

MySQL es un servidor de Base de datos relacional, tiene interfaces para desarrollo de clientes en diversos lenguajes, está disponible en diferentes plataformas, es gratis, se puede disponer del código fuente.

MySQL no cumple con las propiedades ACID de los Sistemas de Gestión de Bases de Datos Relacionales (RDBMS por sus siglas en inglés). Inicialmente no incorporaba integridad referencial, con lo que la base de datos podían tener datos inconsistentes (por ejemplo empleados pertenecientes a departamentos inexistentes), esto se resuelve actualmente con las tablas InnoDB pero según muchos especialistas lo que hace MySQL es simular la integridad referencial. Carece además de transacciones incumpliendo otra vez las ACID. MySQL no es considerado como RDBMS, sino un sistema de ficheros con una interfaz SQL (Pérez, 2008). Además, realiza los bloqueos de escritura al nivel de tabla, limitando el acceso concurrente a las tablas de la base de datos.

MySQL guarda los Blobs (objetos grandes binarios) en la misma tabla que guarda los datos normales, o sea como un campo más en la misma fila, lo cual no es el mejor método para realizar esta tarea ya que hace caer el rendimiento del acceso a los otros datos.

1.4.5.3 PostgreSQL

Surge del paquete de Postgres desarrollado en la universidad de Berkeley en California. Con más de una década de desarrollo como respaldo, PostgreSQL es uno de los sistemas de BD libre avanzados que está disponible en el mundo hoy.

PostgreSQL tiene transacciones, integridad referencial, vistas, y multitud de funcionalidades, pero es lento y pesado que MySQL, aunque en las últimas versiones del mismo esto ha mejorado mucho (I. M. Lockhart, 2009).

En PostgreSQL se utilizan las vistas como tablas virtuales, como una tabla que no existe físicamente en la BD, pero aparece al usuario como si existiera. Cuando se habla de una tabla base, se refiere a que realmente hay un equivalente almacenado para cada fila en la tabla en algún sitio del almacenamiento físico.

Uno de los principios del modelo relacional es que los atributos de una relación son atómicos. PostgreSQL no contiene esta restricción; los atributos pueden contener subvalores a los que puede accederse desde el lenguaje de consulta (T. Lockhart, 2009).

1.5 Metodología de modelado.

El uso de una metodología para la elaboración de un producto informático, garantiza determinadas características en el mismo, dentro de ellas la calidad, factor clave tanto para el cliente como para el productor.

1.5.1 Rational Unified Process (RUP).

La metodología Proceso Unificado de Desarrollo de Software (RUP) se basa en la orientación a objetos, el desarrollo iterativo y el modelamiento visual usando el Lenguaje Unificado de Modelación (UML) para la ingeniería de sistemas y de software, lo que permite incorporar al proceso de desarrollo de software un mejor control de los requerimientos y cambios. Esta metodología proporciona al equipo del proyecto procedimientos y herramientas promoviendo las siguientes prácticas (Jacobson et al., 2000):

- Gestión de los Requisitos.
- Desarrollos iterativos.
- Uso de arquitecturas basadas en componentes.
- Desarrollo Visual del Software con UML.
- Verificación continua de la calidad del software.
- Gestión de los Cambios.

Según la metodología RUP el ciclo de vida de un proyecto se divide en las siguientes fases:

- Fase de concepción.
- Fase de elaboración.
- Fase de construcción.
- Fase de transición (Bowden, 2007).

A pesar de ser una metodología desarrollada directamente para el trabajo con clases y objetos brinda amplias posibilidades para el manejo eficiente del tiempo de diseño e implementación utilizando las extensiones para aplicaciones Web. (Furgeta, 2002).

1.5.2 Lenguaje de Modelaje Unificado (UML).

El UML, es el lenguaje gráfico de modelaje orientado a objetos estándar de la industria para especificar, visualizar, construir y documentar los elementos de los sistemas de software. UML proporciona una forma estándar de escribir los planos de un sistema, cubriendo tanto las cosas conceptuales, tales como procesos del negocio y funciones del sistema, como las cosas concretas, tales como las clases escritas en un lenguaje de programación específico, esquemas de bases de datos y componentes de software reutilizables. Simplifica el proceso complejo de análisis y diseño de software, facilitando un plano para la construcción.

El UML fue desarrollado en Rational Software Corporation por Grady Booch, James Rumbaugh e Ivar Jacobson con contribuciones de otros metodólogos líderes, vendedores de software y muchos usuarios; el UML está basado en el uso extensivo del método Booch, OMT y Jacobson; es decir, el UML es la evolución de éstas y otras aproximaciones para modelado de procesos de negocios, objetos, y el modelado de componentes (Jacobson et al., 2000; Pressman, 2006).

El desarrollo de UML empezó en el 1994 de octubre cuando Grady Booch y Jim Rumbaugh en la Rational Software Corporation empezaron a trabajar para unificar el Booch y la OMT. Un proyecto versión 0.8 del Método Unificado (UML), como se llamó desde un comienzo, salió al público en octubre de 1995, mas tarde se integra Ivar Jacobson.

El Lenguaje Unificado de Modelación proporciona ventajas en la representación del ciclo de vida de un software y de los artefactos específicos del Proceso Unificado de Desarrollo de Software. Permite una comunicación sencilla y rápida entre desarrolladores y clientes del software que se desarrolla. No se necesitan conocimientos profundos de Ingeniería de Software para que los clientes comprendan lo que los desarrolladores muestran, de modo que rápidamente pueden expresar su conformidad con el producto o las nuevas mejoras que desean ver introducidas (Sanz, 2002).

Los creadores de UML han desarrollado una herramienta CASE para representarlo, Rational Rose, que cubre el ciclo de vida de un proyecto. Esta

herramienta permite documentar y llegar a la solución del sistema atravesando todas las etapas del proyecto.

1.6 Conclusiones del capítulo.

Al finalizar el capítulo se ha expuesto todo lo referente a la base teórica, que fundamenta la presente investigación. Se tiene una valoración sobre el por qué de las elecciones referentes al lenguaje de programación a usar, el sistema gestor de bases de datos, la herramienta de desarrollo, el servidor Web y la metodología de ingeniería del software.

Luego de un análisis de algunas propuestas se escogió *PHP versión 5.2.8* como lenguaje de programación, integrado con el servidor Web *Apache versión 2.2.11* y con el servidor de Bases de Datos *MySQL versión 5.1.30*. La herramienta de desarrollo *Dreamweaver 8* fue elegida para crear y editar las páginas Web del Sistema y para guiar el desarrollo la metodología *Rational Unified Process (RUP)* utilizándose en el modelamiento visual *Unified Modeling Language (UML)*.

CAPÍTULO 2. Descripción y Construcción de la solución propuesta.

2.1 Introducción.

Teniéndose como base la metodología seleccionada en el capítulo anterior, se estructuran los tópicos de este capítulo por las disciplinas que la metodología Rational Unified Process (RUP) define.

Se realiza una descripción de la propuesta de solución de este trabajo. Se listan los requerimientos tanto funcionales como no funcionales que debe tener el sistema propuesto, lo que permite hacer una concepción general del mismo e identificar mediante un Diagrama de Casos de Uso las relaciones de los actores que interactúan con el sistema y las secuencias de acciones que realizan. Así como la construcción de la propuesta desarrollando los diagramas de clases de diseño, los diagramas de componentes en la implementación, además de valorar los principios de diseño que se tratan en la aplicación.

2.2. Modelo del dominio.

Para capturar los requisitos correctos y construir el sistema correcto se requiere un firme conocimiento del contexto donde se emplazará el sistema. Para esto se propone un modelo del dominio con el objetivo de definir los tipos más importantes de objetos que representan las cosas que existen o eventos que suceden en el entorno en el que trabaja el sistema (Jacobson, Booch y Rumbaugh, 2004)

¿Por qué un modelo del dominio?

Hay por lo menos dos aproximaciones para expresar el contexto del sistema en una forma utilizable para el desarrollador: modelado del dominio y modelado del negocio.

El sistema que se propone desarrollar no da soporte a un negocio que pueda definirse, sino que surge por una necesidad específica existente que se refuerza con el incremento de la cantidad de medios y las estrategias informáticas actuales sobre el control.

En la *figura 2.1* se puede ver el modelo del dominio, pero antes se identifican los conceptos principales que se utilizan en el diagrama, mediante un glosario de términos de las clases conceptuales:

- Ministerio de Educación Superior: Representa el ministerio de educación al que pertenece y se subordina la Universidad de Granma. Es el encargado de asignar Medios Básicos a la Universidad de Granma.
- Universidad de Granma: Representa la universidad que se beneficia de los Medios Básicos que adquiere por compra, asignación del Ministerio de Educación Superior o por donación.
- Compra de Medio Básico: Representa la(s) compra(s) que realiza la Universidad de Granma según necesidad o presupuesto para beneficiarse de dichos medios básicos.
- Proyecto: Representa la coordinación que existe entre la Universidad de Granma y otra institución para el desarrollo de proyectos investigativos o de producción; es además quién según necesidad y acuerdo dona medios básicos a la Universidad de Granma para el desarrollo de dichos proyectos.
- Facultad o áreas: Representa cada una de las facultades o áreas por las que está organizada la Universidad de Granma, es donde se asignan los medios básicos que obtiene la Universidad por compra, asignación del Ministerio de Educación Superior o por donación.
- Local: Representa el(los) local(es) que pertenecen a un área o facultad de la Universidad de Granma. Es donde son distribuidos los Medios Básicos que obtiene la Universidad.
- Medio Básico: Son recursos materiales que obtiene la Universidad de Granma por compra, asignación del Ministerio de Educación Superior o donación.
- Movimiento: Representa el traslado de un lugar a otro que se realiza a un medio básico con un fin determinado.
- Control de Servicio: Es el proceso de controlar el (los) servicio(s) de reparación que se realiza a un medios básico por rotura o manteniendo programando.

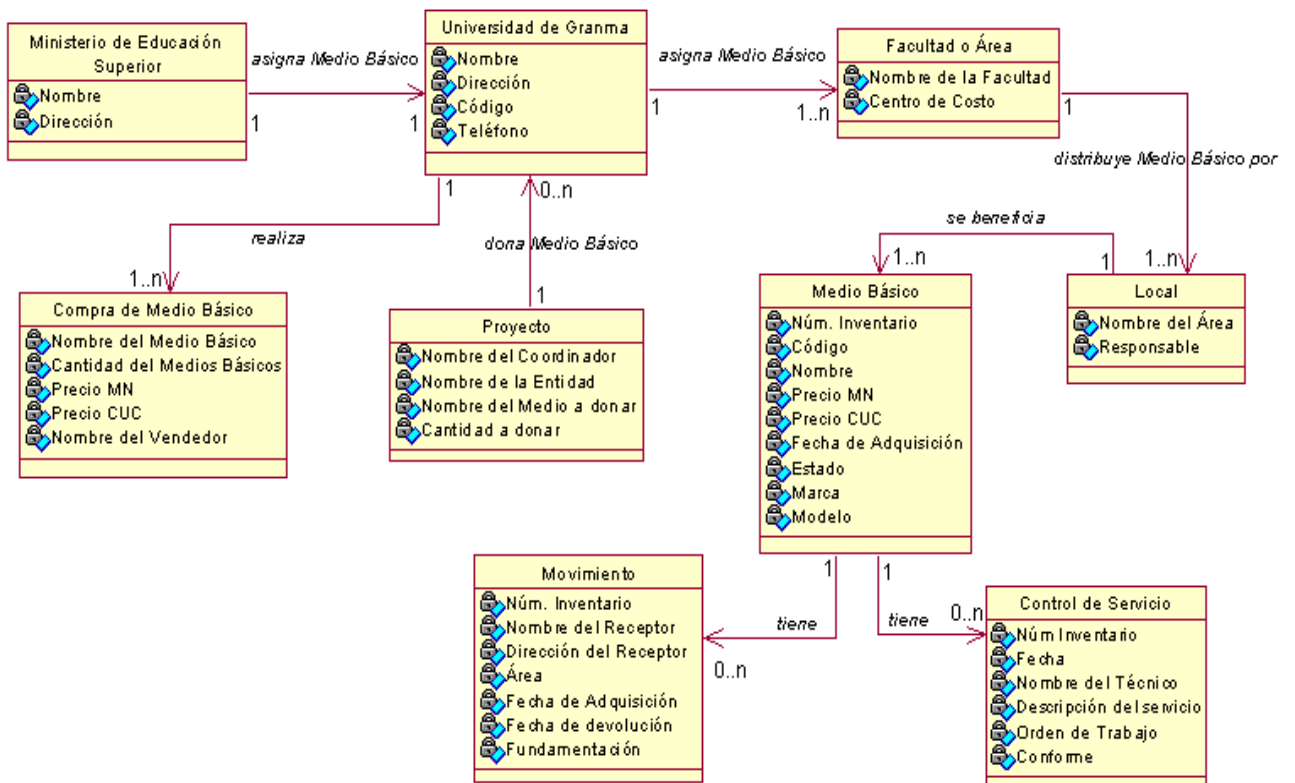


Figura 2.1 Diagrama de clases del dominio.

2.3 Requerimientos del sistema.

Conocidos los conceptos que rodean al objeto de estudio, se analizan las funcionalidades del producto a desarrollar, o sea, ¿qué debe hacer sistema para que se cumplan los objetivos planteados?, para ello se enumeran, a través de requerimientos funcionales, las acciones que el sistema deberá ser capaz de realizar. El propósito de su gestión es establecer un entendimiento común entre el usuario y el desarrollador de software. Los requerimientos se clasifican en requerimientos funcionales y no funcionales (Pressman, 2007).

2.3.1 Requerimientos funcionales.

Los requerimientos funcionales especifican el comportamiento de entrada y salida del sistema y surgen de la razón fundamental de la existencia del producto. Especifican también acciones que el sistema debe ser capaz de realizar, sin tomar en consideración ningún tipo de restricción física.

Para una mejor comprensión, los requerimientos funcionales se han agrupado en tres paquetes que representan el sistema y agrupan las funcionalidades de este, los mismos se muestran a continuación (Pressman, 2008).

Control de Medios Básicos

- R1. Insertar medio básico.
- R2. Modificar medio básico.
- R3. Eliminar medio básico.
- R4. Insertar movimiento de medios básicos.
- R5. Modificar movimiento de medios básicos.
- R6. Eliminar movimiento de medios básicos.
- R7. Insertar control de servicio.
- R8. Modificar control de servicio.
- R9. Eliminar control de servicio.
- R10. Insertar local.
- R11. Modificar local.
- R12. Eliminar local.
- R13. Insertar área o facultad.
- R14. Modificar área o facultad.
- R15. Eliminar área o facultad.
- R16. Insertar responsable de local.
- R17. Modificar responsable de local.
- R18. Eliminar responsable de local.
- R19. Insertar receptor de medio básico.
- R20. Modificar receptor de medio básico.
- R21. Eliminar receptor de medio básico.
- R22. Insertar tipo de movimiento de medios básicos.
- R23. Modificar tipo de movimiento de medios básicos.
- R24. Eliminar tipo de movimiento de medios básicos.

Visualización

- R25. Visualizar medios básicos.
- R26. Visualizar movimientos de medios básicos.
- R27. Visualizar tarjeta de control de servicio (modelo AT-5).

- R28. Visualizar locales.
- R29. Visualizar áreas o facultades.
- R30. Visualizar responsables de locales.
- R31. Visualizar receptores de medio básico.
- R32. Visualizar tipos de movimientos de medios básicos.
- R33. Imprimir medios básicos.
- R34. Imprimir movimientos de medios básicos.
- R35. Imprimir tarjeta de control de servicio (modelo AT-5).

Seguridad.

- R36. Insertar usuario.
- R37. Modificar usuario.
- R38. Eliminar usuario.
- R39. Establecer nivel de acceso.
- R40. Visualizar usuario.
- R41. Autenticarse.
- R42. Validar usuario.
- R43. Finalizar sesión.

2.3.2 Requerimientos no funcionales.

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener y que de una u otra forma puedan limitar el sistema. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. Normalmente están vinculados a requerimientos funcionales.

Apariencia o interfaz externa

- Diseño sencillo, orientado al entorno de trabajo del cliente para que se sienta identificado con la aplicación.
- Los colores deben ser claros y agradables a la vista del usuario.
- El sistema debe tener una interfaz cómoda, facilitando la navegación y evitando la utilización en exceso de imágenes.
- El vocabulario utilizado será en idioma español exclusivamente.

Usabilidad.

- El sistema debe estar funcionando durante el horario laboral.
- El sistema será utilizado por profesional (autorizado del departamento de Administración de medios básicos de la facultad de Informática de la Universidad de Granma).

Software.

- Las computadoras clientes deben contar con Internet Explorer 5.0 o superior.
- El sistema debe funcionar principalmente en el sistema operativo Windows.
- La máquina computadora servidor debe tener Windows 2000 o superior, Servidor Web Apache versión 2.2.11, MySQL versión 5.1.30 y PHP versión 5.2.8.

Hardware.

Para la explotación de la aplicación se necesita un servidor para procesar y almacenar la información con las siguientes características (condiciones mínimas):

- Microprocesador: Pentium III o superior.
- Memoria RAM: 128Mb o superior.
- Disco duro: 10 Gb o más
- Periféricos: Teclado, mouse, tarjeta de Red, entre otros periféricos estándares.
- Debe estar conectada a la red.

Los requerimientos mínimos de las máquinas clientes deben ser:

- Microprocesador: Pentium I o superior.
- Memoria RAM: 64 MB o superior.
- Velocidad en la Red: 10 Mbit/seg o mayor.
- Deben estar conectadas a la red.

Seguridad.

- Solo los usuarios autorizados podrán acceder a determinada información para garantizar la confidencialidad de la información y que cada usuario sólo pueda realizar las operaciones que le corresponden de acuerdo al rol que

desempeña dentro del mismo. Para garantizar esto se cuenta con 2 niveles de acceso: Administrador y usuario.

Los requerimientos funcionales generados por los requerimientos no funcionales de seguridad son R36 - R43.

Confiabilidad.

- La información almacenada en el sistema debe ser estrictamente confidencial.
- La información almacenada en el sistema debe corresponderse con el resultado del proceso de registro y control de la información referente a los medios básicos de la facultad de Informática de la Universidad de Granma.

2.4 Factibilidad.

2.4.1 Introducción.

Debido a la necesidad en todo proyecto de software de analizar su factibilidad, se obtiene el tiempo de desarrollo y el costo total para la aplicación propuesta. Se ha tenido en cuenta para la valoración de los parámetros necesarios en el método de estimación basado en casos de uso, los elementos brindados por Gustav Karner en su document "Estimation Using Use Case Points" (Karner, 2006; Peralta, 2007).

2.4.2 Planificación basada en casos de uso.

Paso 1. Cálculo de los puntos de casos de uso sin ajustar (UUCP).

A partir de los requerimientos funcionales se identificaron las características del sistema a desarrollar en términos de actores (Factor de Peso de los Actores sin ajustar) y casos de uso (Factor de Peso de los Casos de Uso sin ajustar), se clasificaron y se realizó la ponderación aplicando pesos para de esta forma obtener el Factor de Peso de los Actores sin ajustar (UAW) y Factor de Peso de los Casos de Uso sin ajustar (UUCW). La tabla que contiene esta información puede ser consultadas en el *Anexo 1*.

Puntos de Casos de Uso sin ajustar (UUCP) = UAW + UUCW

Puntos de Casos de Uso sin ajustar (UUCP) = 6 + 165 = **171**

Paso 2. Cálculo de los puntos de caso de uso ajustados (UCP).

Una vez que se tienen los Puntos de Casos de Uso sin ajustar, se debe ajustar éste valor para ello se debe calcular primero el factor de complejidad técnica (TCF) y el factor ambiente (EF).

El Factor de Complejidad Técnica (TCF) es un coeficiente que se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de los factores se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante. El Factor de complejidad técnica se calcula mediante la siguiente ecuación: $TCF = 0.6 + 0.01 \times \sum (\text{Peso}_i \times \text{Valor asignado}_i)$.

El Factor Ambiente (EF) son las habilidades y el entrenamiento del grupo involucrado en el desarrollo su cálculo es similar al cálculo del Factor de complejidad técnica, es decir, se trata de un conjunto de factores que se cuantifican con valores de 0 a 5. El Factor de ambiente se calcula mediante la siguiente ecuación: $EF = 1.4 - 0.03 \times \sum (\text{Peso}_i \times \text{Valor asignado}_i)$

Las tablas que contienen los resultados de la cuantificación de los factores de complejidad técnica y el factor ambiente pueden ser consultadas en el *Anexo 2*.

$$\text{Factor de complejidad técnica (TCF)} = 0.6 + 0.01 * 24 = \mathbf{0.84}$$

$$\text{Factor Ambiente (EF)} = 1.4 - 0.03 * 19.5 = \mathbf{0.815}$$

$$\text{Luego } UCP = UUCP * TCF * EF$$

$$UCP = \mathbf{171} * \mathbf{0.84} * \mathbf{0.815}$$

$$\mathbf{UCP = 117,06}$$

Paso 3. Cálculo de la estimación del esfuerzo estimado en horas-hombre (E)

Para calcular el esfuerzo estimado en horas-hombre primeramente se debe obtener el factor de conversión (CF) a utilizar según las características que del proyecto, para ello se contabilizan cuántos factores de los que afectan al Factor de ambiente (EF) están por *debajo* del valor medio (3), para los factores E1 a E6 y cuántos están por *encima* del valor medio (3), para los factores E7 y E8. Del resultado obtenido ver que condición se cumple:

- Si el total es 2 o menos, se utiliza el *factor de conversión 20 horas-hombre/Punto de Casos de Uso*, es decir, un Punto de Caso de Uso toma 20 horas-hombre.
- Si el total es 3 o 4, se utiliza el *factor de conversión 28 horas-hombre/Punto de Casos de Uso*, es decir, un Punto de Caso de Uso toma 28 horas-hombre.
- Si el total es mayor o igual que 5, se recomienda efectuar cambios en el proyecto, ya que se considera que el riesgo de fracaso del mismo es demasiado alto.

Como $EF < 2$ entonces $CF = 20$ horas-hombre

Luego: $E = UCP \times CF$

$E = 117,06 * 20$ horas-hombre

$E = 2341,2$ horas-hombre

Este método proporciona una estimación del esfuerzo en horas-hombre contemplando sólo el desarrollo (Implementación) de la funcionalidad especificada en los casos de uso. Para una estimación más completa de la duración total del proyecto, hay que agregar a la estimación del esfuerzo obtenida por los Puntos de Casos de Uso, las estimaciones de esfuerzo de las demás actividades relacionadas con el desarrollo de software.

Para ello se tuvo en cuenta el siguiente criterio, que estadísticamente se considera aceptable. El criterio plantea la distribución del esfuerzo entre las diferentes actividades de un proyecto, según la siguiente aproximación:

Actividad	Porcentaje
Análisis	10.00 %
Diseño	20.00 %
Programación	40.00 %
Pruebas	15.00 %
Sobrecarga (otras actividades)	15.00 %

Tabla 2.1 Criterio estadístico de la distribución del esfuerzo entre las diferentes actividades de un proyecto.

Con éste criterio, y tomando como entrada la estimación de esfuerzo de implementación calculada a partir de los Puntos de Casos de Uso, se calculan las demás estimaciones para obtener la duración total del proyecto.

Actividad	% esfuerzo	Valor del esfuerzo
Análisis	10%	585.3 horas-hombre
Diseño	20%	1170.6 horas-hombre
Implementación	40%	2341,2 horas-hombre
Prueba	15%	877.95 horas-hombre
Sobrecarga	15%	877.95 horas-hombre
Total	100%	5853 horas-hombre

Tabla 2.2 Cálculo de esfuerzo de las diferentes actividades del proyecto.

Paso 4. Cálculo de tiempo de desarrollo.

Suponiendo que se trabaja 8 horas por día, y un mes tiene como promedio 30 días; la cantidad de horas que puede trabajar una persona en un mes es 240 horas; como el esfuerzo total de desarrollo del proyecto es 5856 horas-hombre se calcula entonces el tiempo de desarrollo del proyecto al dividir el esfuerzo total entre la cantidad de horas que se laboran en un mes, obteniéndose:

$$5853 / 240 = 24.3875 \text{ mes-hombre}$$

Esto quiere decir que 1 persona puede realizar el proyecto en más o menos 24 meses y medio (24.3875). Como el equipo de trabajo lo constituyen 4 personas, estimándose que todas realicen el mismo esfuerzo el proyecto puede realizarse en aproximadamente **6 meses** (24.3875 mes-hombre / 4 hombres = 6.09 meses)

2.5 Modelo del Sistema.

Utilizando las facilidades que brinda el lenguaje de modelamiento visual (UML), los requerimientos funcionales del sistema son representados mediante un diagrama de casos de uso. Para ello se definen cuáles son los actores que van a interactuar con el sistema y los casos de uso que representan las funcionalidades (Larman, 2004).

2.5.1 Actores del sistema

Un actor no es parte del sistema, sino un rol que se juega dentro del mismo, que puede intercambiar información o puede ser un recipiente pasivo de información. Puede representar un ser humano, un *software* o una máquina que interactúa con

el sistema. En la *figura. 2.2* se muestran los actores del sistema y sus relaciones y en la *Tabla. 2.3* se encuentran sus descripciones.



Figura 2.2 Relación entre Actores del Sistema.

Actores	Descripción	Requerimientos asociados
Administrador	Es una especialización del actor usuario. Representa el rol que desempeña la administradora de los medios básicos de la Facultad de Informática de la Universidad de Granma. Es el rol encargado de controlar y procesar los datos relacionados con los movimientos y servicios realizados a estos.	R1- R24, R36 - R40
Usuario	Generalización del actor Administrador. Representa el rol de cualquier usuario que accede al sistema para consultar o adquirir cualquier información relacionada con el control de los medios básicos.	R25 - R35, R41 - R43

Tabla 2.3 Descripciones de los actores del sistema.

2.5.2 Paquetes del Sistema

Para su mejor comprensión y teniendo en cuenta sus funcionalidades, el sistema se dividió en 3 paquetes: Seguridad, Control de Medios Básicos y Visualización. En la *figura 2.3* se muestran estos paquetes.

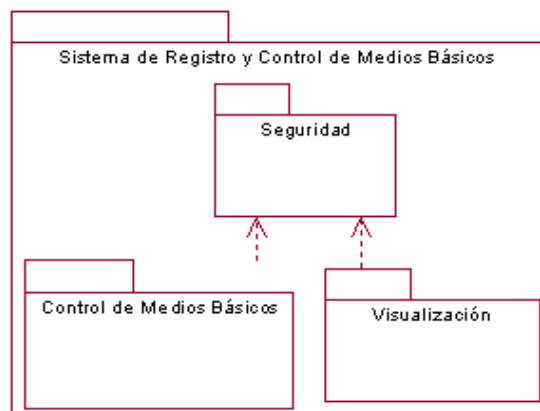


Figura. 2.3 Diagrama de Paquetes.

2.5.3 Diagramas de casos de uso del sistema.

En la *figura 2.4* se puede apreciar el diagrama de caso de uso del sistema correspondiente al paquete de *control de Medios Básicos*. Los diagramas de caso de uso de los paquetes *Visualización* y *Seguridad* pueden ser consultados en el *Anexo 3*. Para un mejor entendimiento de los casos de uso del sistema consultar sus descripciones textuales representadas en el *Anexo 4*.

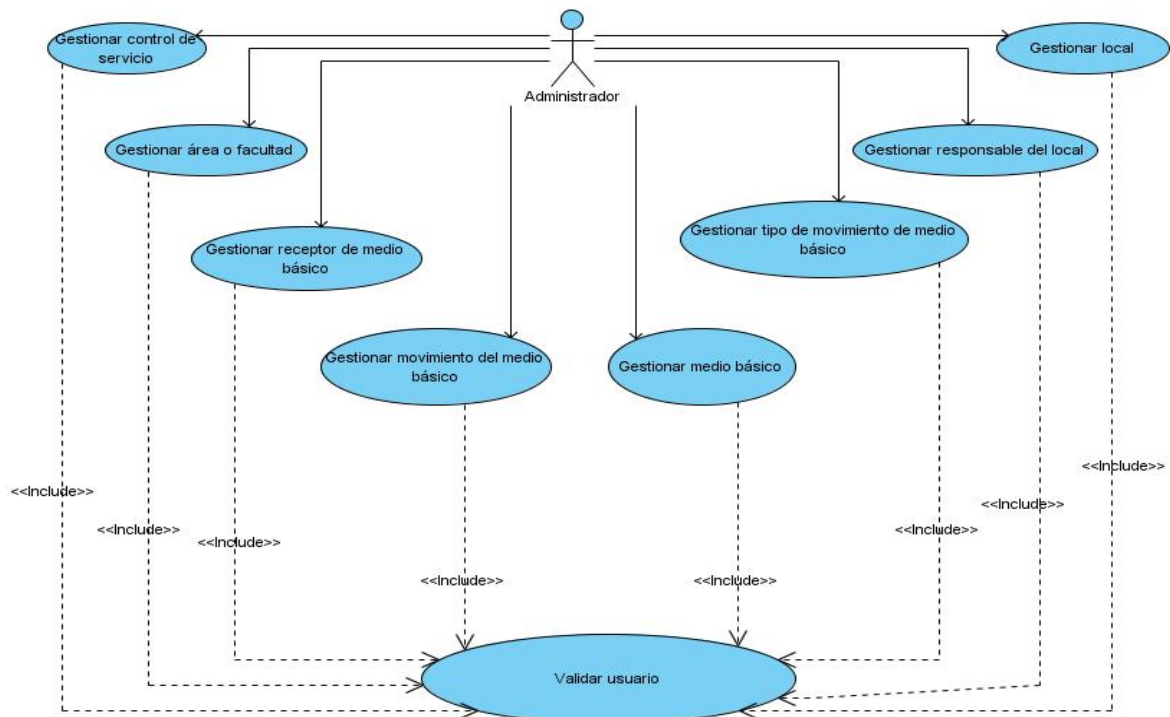


Figura. 2.4 Diagrama de casos de uso del sistema: *Paquete de control de Medios Básicos*.

2.6 Modelo del Análisis

El modelo del análisis es un refinamiento y cambio de estructuración de los requisitos con el fin de que se comprenda de manera más precisa el sistema que se desarrolla.

Este flujo reviste una gran importancia pues permite, de forma general:

- Describir requisitos de forma precisa.
- Estructurarlos de manera que se facilite su comprensión.
- Servir de punto de partida para dar forma al sistema durante su diseño e implementación, incluyendo su arquitectura.

- Utiliza clases de análisis para describir los casos de uso, esto se logra mediante la colaboración entre estas clases y se le denomina Realizaciones de Caso de Uso-Análisis.

Entre los artefactos del modelo del análisis se encuentran las clases del análisis, los diagramas de clases del análisis, los diagramas de interacción (secuencia o colaboración) y la descripción del flujo de sucesos-análisis que completa los diagramas de colaboración.

Para llevar a cabo este flujo en el desarrollo de la aplicación Web se realizaron los diagramas de colaboración, en la *figura 2.5* se puede apreciar el diagrama de colaboración del CUS *Visualizar Medios Básicos* los principales diagramas de colaboración pueden ser consultados en el *Anexo 5*.

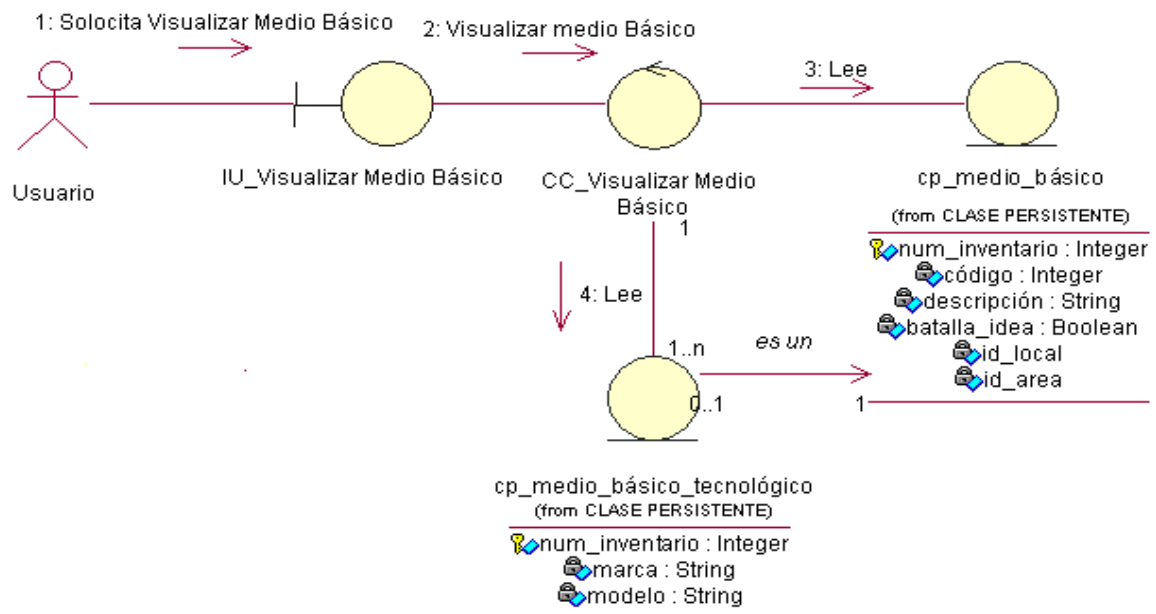


Figura. 2.5 Diagrama de colaboración. Caso de uso Visualizar Medios Básicos.

El resultado obtenido en el análisis constituye una entrada esencial en el diseño pues proporciona una comprensión detallada de los requisitos e impone una estructura de la aplicación que en el modelo del diseño se conserva lo más fielmente posible.

2.7 Modelo del Diseño

En la fase de diseño se detalla el modelo de análisis; su propósito es especificar una solución que funcione y pueda ser convertida a código fuente. Este flujo

describe la realización física de los casos de uso usando artefactos como diagramas de clases. Dentro de este modelo se realizan los artefactos: Diagrama de clases del diseño, Mapa de navegación, Diagrama de clases persistentes y Modelo de datos (Rational Rose, 2007).

Los elementos más comunes que conforman la tecnología Web son las páginas y los formularios. UML brinda varios estereotipos para extensiones Web, entre los más comunes se encuentran:

- <<Server Page>> Representa la página Web que tiene código que se ejecuta en el servidor.
- <<Client Page>> Una instancia de Página Cliente es una página Web, con formato HTML. Son interpretadas por el navegador.
- <<Html Form>> Colección de elementos de entrada que son parte de un página cliente. Sus atributos son los elementos de entrada del formulario.

En la aplicación, para ayudar a identificar las clases se le añadió a los nombres de las clases clientes, servidoras y formularios los prefijos Cl, Sr y Fr respectivamente.

La clase cliente Index representa la portada del sistema. Tiene enlaces a las clases servidoras que representan a cada uno de los módulos implementados. Una muestra representativa de los diagramas de clases del diseño que ilustran la estructura de la solución propuesta puede ser consultada en el *Anexo 6*.

2.7.1 Patrones de Diseño

Este sistema fue diseñado para facilitar el registro y control de los medios básicos en la Facultad de Informática de Universidad de Granma de forma dinámica y agradable al usuario. Para lograrlo se emplearon algunos principios de diseño visual en las páginas que la conforman.

El diseño visual define la apariencia del sistema y es de gran importancia para lograr que el usuario se sienta satisfecho con la información que obtiene y con la forma en que lo hace. Se eligieron los colores azul claro y oscuro, negro y blanco pues se consideró que estos colores contribuyen a construir una interfaz agradable a la vista. Se utilizó el azul claro para el fondo de las páginas, algunas

letras, y para el fondo de las tablas, el negro para las letras garantizando una lectura favorable de los textos.

Se escogió la letra Verdana para los textos de las páginas. Este tipo de letra permite una lectura rápida y cómoda. Es mínimo el uso de imágenes y nulo el de animaciones para evitar largos tiempos de espera a la hora de visualizarlas.

Una muestra de las pantallas de autenticación al sistema puede ser consultada en el *Anexo 7*.

2.8 Mapa de Navegación

El mapa de navegación define la estructura jerárquica de páginas lógicas de la aplicación, siendo cada página lógica candidata a convertirse en una interfaz de usuario. El modelo de datos de la aplicación se puede consultar en el *Anexo 8*.

2.9 Diseño de la Base de Datos

Una etapa muy importante en la realización de un software es el diseño de la Base de Datos (BD), el cual incluye la definición de las tablas, los índices, las vistas, y otras estructuras específicas necesarias para almacenar, recuperar y eliminar la información persistente. Teniendo como objetivo asegurar que los datos persistentes se almacenen de forma consistente y eficientemente, definiendo el comportamiento que debe ser implementado en la BD.

La base de datos que contiene la información persistente generada a partir del modelo de datos de la solución propuesta se encuentra almacenada en el servidor de base de datos *MySQL versión 5.1.30*, proporcionando grandes ventajas en cuanto a confiabilidad y rapidez en el acceso a los datos.

Para lograr la conexión desde las páginas dinámicas hasta la base de datos dentro del sistema se necesita una exitosa autenticación en el servidor. Esto se logra cuando las páginas dinámicas le proporcionan al servidor de base de datos el usuario y contraseña definidos en él. Una vez conectadas se envía información a la base de datos y se actualizan las tablas o se leen datos de éstas para conformar los reportes que los usuarios solicitan.

2.9.1 Diagrama de clases persistentes

Las clases persistentes representan la información que el sistema necesita almacenar a largo plazo en su base de datos, información que es identificada a partir del diagrama de clases del diseño.

Se definieron clases persistentes para almacenar toda la información necesaria de cada uno de los módulos del sistema, teniendo en cuenta que no hubiese redundancia de los datos para garantizar la integridad, consistencia y confiabilidad de éstos. El diagrama de clases persistentes se puede encontrar en el *Anexo 9*.

2.9.2 Modelo de Datos

El modelo de datos describe la estructura lógica de la información persistente almacenada por el sistema. Se creó a partir de las clases persistentes obtenidas previamente. Esto permitió mantener las estructuras que garantizaran la integridad, consistencia y confiabilidad de los datos tal y como se constató en las pruebas realizadas al sistema.

Este modelo fue empleado para definir la transformación de las clases persistentes en las estructuras de datos persistentes utilizadas en la aplicación Web. El modelo de datos obtenido del diagrama de clases persistentes puede ser consultado en el *Anexo 10*.

2.10 Modelo de Implementación

Durante este flujo de trabajo se desarrolla todo lo necesario para obtener un sistema ejecutable. El modelo de implementación está compuesto por componentes que incluyen todos los ejecutables.

El Diagrama de componentes se usa para modelar la estructura del software, incluyendo las dependencias entre los componentes de este, los del código binario y los ejecutables. Además se modelan componentes del sistema, a veces agrupados por paquetes, y las dependencias que existen entre componentes (y paquetes de componentes).

El Diagrama de Componentes se usa para modelar la estructura del sistema, modelando como bien su nombre indica los componentes del mismo, a veces agrupados por paquetes, y las dependencias que existen entre estos. En el

Anexo 11 pueden ser consultados los principales diagramas de componentes del sistema.

2.11 Diagrama de Despliegue

El diagrama de despliegue del Sistema para el registro y control de Medios Básicos tiene como propósito fundamental mostrar las conexiones que se establecen entre los nodos necesarios para formar el sistema. Se pueden observar en la *Figura. 2.6* los elementos de procesamiento que son utilizados por la aplicación.

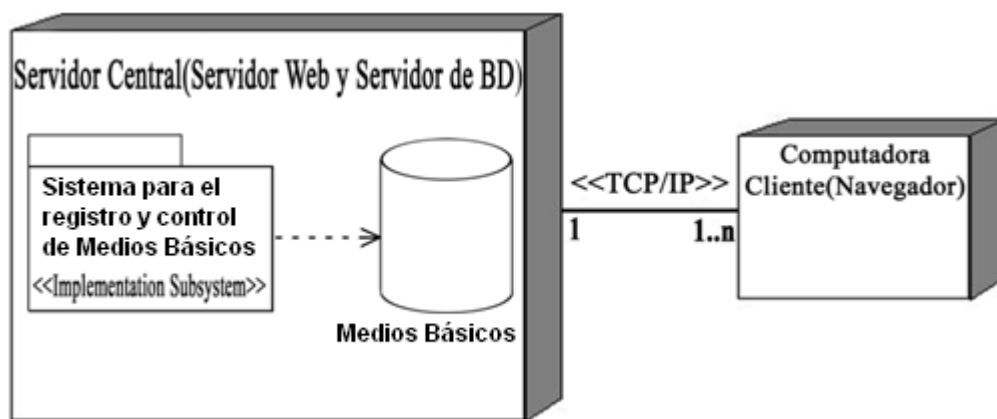


Figura 2.6 Diagrama de Despliegue.

El Servidor Central se encontrará en la Universidad de Granma donde se encuentran todos los componentes del sistema y como servidor de Base de Datos, donde se almacenan los datos que son actualizados y consultados por los usuarios del sistema.

La Computadora Cliente representa el grupo de computadoras a través de las cuales los usuarios pueden gestionar la información que se encuentra en el Servidor Central. Estas computadoras son las que se encuentran en los diferentes Locales de la Facultad de Informática. Para acceder a la aplicación, las Computadoras Clientes utilizarán una aplicación cliente, en este caso, el navegador Internet Explorer u otro navegador compatible.

La comunicación entre las Computadoras Clientes y el Servidor Central se establece utilizando el conjunto de protocolos de comunicación TCP/IP.

2.12 Tratamiento de Errores

En todas las páginas del sistema se garantiza que cuando se produzca algún error el usuario sea advertido, lo que se realiza a través de alertas JavaScript. En las páginas de inserción y modificación de información se garantiza que el usuario no pueda escribir caracteres que no se correspondan con el tipo de dato en la base de datos, por ejemplo: puntos donde sólo se deben escribir números enteros, números donde sólo se debe escribir texto, etc. Cada nueva información a introducir en el sistema es chequeada que no exista en la base de datos para evitar la existencia de información repetida en la misma. Si se deja de introducir un dato (que sea obligatorio entrar), el sistema informa mediante una alerta y no permite la inserción de la información en la base de datos hasta que no se complete. En la *figura 2.7* se presenta una ventana de alerta que se genera cuando se trata de introducir un elemento que no es una letra donde debe ir.



Figura 2.7 Aviso de una acción indebida.

2.13 Conclusiones del Capítulo

El uso de los diferentes artefactos proporcionados por la metodología RUP para el desarrollo de software permitió alcanzar el objetivo propuesto en la investigación, o sea la concepción final de la aplicación Web.

A partir del análisis realizado se construyó la aplicación Web que introdujo una nueva vía para la gestión y control de la información de los medios básicos en la Facultad de Informática de la Universidad de Granma.

CONCLUSIONES

Con el desarrollo de la presente investigación, se da cumplimiento al objetivo propuesto, obteniendo un software para la gestión, control y procesamiento de la información referente a los medios básicos de la Facultad de Informática de la Universidad de Granma. Los resultados más relevantes se mencionan a continuación:

- La aplicación permite, utilizando la conectividad, un mejor flujo de información referida a los medios básicos de la Facultad de Informática de la Universidad de Granma.
- Es una herramienta de fácil manejo que agiliza el proceso de gestión y control de los medios básicos de la Facultad de Informática de la Universidad de Granma.

RECOMENDACIONES

Teniendo como base la investigación realizada se proponen las recomendaciones siguientes:

- Definir las políticas de seguridad para la implantación del sistema en la Facultad de Informática de la Universidad de Granma.
- Generalizar el Sistema a todas las áreas y facultades de la Universidad de Granma.
- Valorar la posibilidad de añadir nuevas utilidades al sistema como parte de futuras investigaciones, fundamentalmente el diseño de un método de encriptación para el control de usuarios y contraseñas.

BIBLIOGRAFÍA

- Aldana, C. R. (2009). Apache. El servidor Web más reconocido, 76.
- Álvarez, M. (2007). ¿Qué es ASP? Disponible en: <http://www.programación.com>.
- ANTARESX. (2004). Lenguajes de Programación Orientados a Objeto. 69.
- ASKALL ORG, S. A. (2007). Programando en Java. Standfor: Addison-Wesley Company, 69.
- ASSETS. (2006). ASSETS: Sistema de Gestión Integral. Disponible en: <http://www.assets.co.cu/> [2010, 13-12-2010].
- Bejerano, G. (2010). Control interno, Conferencia de Control Interno (pp. 34). Provincia Granma.
- Bellman, R. E. (2007). Web Programming (Vol. 2). Princenton Princenton University Press, 201.
- Blanco, L. (2009). Manual de trabajo. Su empleo en la gestión empresarial (pp. 38-41): Revista especializada MAC. Auditoría y Control.
- Bowden, D. (2007). Rational Rose Tutorials. Disponible en: <http://homepages.uel.ac.uk/D.Bowden/> [2010, 12-1-2010].
- Brusilovsky, P., Gilles, G., Frasson, C. y VanLehn, K. (2007). Course sequencing for Static Courses? Applying ITS Techniques in Large-Scale Web_Based Education (Springer Verlag ed.): Proceedings of Intelligent Tutoring Systems, 625-634.
- Callejas, A. (2002). Los Alcances del Control de Gestión. Dirección de la especialización de finanzas y Administración Pública, 9.
- Cano Hidalgo, I. (2006). Sistema de gestión Integral ASSETS-NS: garantía de seguridad total. Disponible en: http://www.cadenagramonte.cubaWeb.cu/ciencia/sistemas_gestion_integral.asp [2010, 12-12-2010].
- Castillo Castillo, T. (2010). Curso virtual de Sistema de control interno para los auditores del oriente cubano de la Contraloría General de la República. Universidad de Granma, Bayamo, 64.
- Cinca, T. C. (2008). Características principales de la Web. Web Estática y Web Dinámica. Aplicaciones para la gestión, 12(4), 9-14.

- Cobo, Á., Gómez, P., Pérez, D. y Rocha, R. (2010). PHP y MySQL Tecnologías para el desarrollo de aplicaciones Web. Ciencias de la información, 13.
- Colombo, B., Demetrescu, C., I., F. y Laura, L. (2003). A Java-based System for Building Animated Presentations over the Web. Elsevier Science of Computer Programming, 901-946.
- Conallen, J. (2006). Building Web Applications with UML, 119.
- Chiavenatto, A. (2005). Administración: Proceso Administrativo (Tercera Edición ed.). Bogotá: Makron Books Do Brasil Editora, LTDA, 56.
- De Castro, E. y García del Junco, J. (2001). Administración y Dirección: McGraw - Hill Interamericana de España, S.A, 121.
- Estrada, M. (2004). Metodología para documentar los sistemas de control interno. La Habana, 41-50.
- Francois, L. (2006). Manual de la gestión basada en resultados, 1-70.
- Furgeta, A. (2002). Introduction to Systems Engenieering. New York: MIT Press, 56
- Gracia, J. (2006). Munual de PHP. WebEstilo: Adisson-Wesley, 111.
- Hernán Ruiz, M. (2006). Programación Web avanzada: Soluciones rápidas y efectivas para desarrolladores de sitios. La Habana: Editorial Félix Varela, 296.
- Hernández, J. (2003). La lucha contra la corrupción y las ilegalidades. Res 13/03. La Habana, 13.
- Herrera, F., Sandoval, J. y Quijada, J. (2009). Incorporación de nuevas tecnologías. INTRANET. Disponible en: <http://WWW.rediris.es/rediris/boletín/41-42/ponencia13.html> [2010, 1-12-2010].
- Jacobson, I., Booch, G. y Rumbaugh, J. (2000). El Proceso Unificado de Desarrollo de Software (Addison-Wesley Company ed.). Stanford, 268.
- Jacobson, I., Booch, G. y Rumbaugh, J. (2004). El Proceso Unificado de Desarrollo de Software (Vol. I y II). La Habana: Félix Varela, 120.
- JBC. (2009). Manual de JBC (Java Byte Code), 211.
- JDK. (2009). Microsystem del JDK 1.0 (Java Development Kit).

- Jerez, R. (2006). El proceso de auditoría y control. Instituto Mexicano de Contadores Públicos. Disponible en: <http://www.monografias.com/apa.shtml> [2010, 21-11-2010].
- Karner, G. (2006). Estimation Using Use Case Points. Disponible en: <http://www.bfpug.com.br/Artigos/UCP/Damodaran> [2010, 1-12-2010].
- King, B. (2007). Better Designs in Half the Time: Implementing QFD Quality Function Deployment in America: Methuen, MA, GOAL/QPC, 98.
- Larman, C. (2004). UML y patrones. Introducción al análisis y diseño orientado a objetos (Vol. I y II). La Habana: Félix Varela, 258.
- Lockhart, I. M. (2009). Gestor de Base de Datos Postgre SQL. 23-33.
- Lockhart, T. (2009). Tutorial de PostgreSQL. 111-116.
- Lynch, E. (2003). Constructivism and Distance Education. Internet. Disponible en: <http://seamonkey.ed.asu.edu/~mcisaac/emc703old97/spring97/7/lynch7.htm> [2010, 12-6-2010].
- Málaga, D. (2008). Manual de HTML y JavaScript, 111.
- Modificaciones al Plan de Estudios D de la carrera Ingeniería Informática, (2010).
- Microsoft. (2009). Internet Information Server (IIS). 11-23.
- Morell, M. (2006). Programa en Active Server Page (ASP), 56.
- Oliver, K. (2000). Methods for developing constructivist learning on the Web. Educational Technology Research & Development, 6, 1-18.
- ONU. (2008). Seminario Internacional de Auditoría Gubernamental de Austria, bajo el patrocinio de la Organización de las Naciones Unidas (ONU). Disponible en: <http://www.monografias.com/trabajos71/activos-fijos-procedimientos-control-contabilizacion/activos-fijos-procedimientos-control-contabilizacion2.shtml> [2010, 1-12-2010].
- Pedraza, L. (2004). Discurso de la Ministra de Auditoría y Control. Auditoría y Control, 13-15.
- Peralta, M. (2007). Estimación del esfuerzo basada en casos de uso, 78.
- Pérez, J. M. (2008). ¿Qué es MySQL? , 12-21.
- Prentzas, J., Hatzilygeroudis, I. y Garofalakis, J. (2002). A Web-Based Intelligent Tutoring System Using Hybrid Rules. Intelligent Tutoring Systems- ITS 2002, 119-128.

- Pressman, R. (2008). Software Engineering. A practitioner's Approach (Fourth Edition ed. Vol. Tomo I), 234.
- Pressman, R. S. (1996). Ingeniería de Software: Un Enfoque Práctico (Addison-Wesley Company ed. Vol. 1). Standford, 94.
- Puerres, I. (2007). Módulo de auditoría financiera, Auditoría (pp. 46).
- Quirós, C. (2005). Auditoría Interna. Paper presented at the Congreso Latinoamericano de de Auditoría Interna. X CLAI.
- Rational Rose. (2007). Procedimientos básicos para desarrollar un proyecto. Disponible en: <http://www.vico.org/TallerRationalRose.pdf> [2010, 20-11-2010].
- Sanz, A. (2002). Introducción a UML. Disponible en: <http://www.yoprogramo.com/docs/umlintro.doc>. [2006, 22-04-2006].
- SQL. (2007). Mysql Disponible en: <http://ardilla.zapto.org/maestria/MySQL.PDF> [2011, 11-1-2011].
- Taylor, J. (2009). Nuevas funciones y ventajas de Dreamweaver 8. 12-17.
- Valles, J. G. (2008). El modelo Cliente-Servidor. 23-34.
- Villa, L. (2006). La importancia creciente de la accesibilidad de sitios Web. Disponible en: <http://www.alzado.org/articulo.php> [2011, 11-1-2011].
- Villa, T. M. (2008). Lenguajes mas utilizados para la realizacion de Webs dinámicas, 45.
- Wallace, D., Aufgang, J. y Raggett, I. (2002). Extreme Programming for Web Projects: Addison Wesley, 78.
- Zavaro, I. (2000). Auditoría a Distancia, un ojo informático para el control. Revista de Auditoría y Control, 1(2), 11-15.
- Zelkovitz, M. V., Shaw, A. C. y Gannon, J. D. (2009). Principles of Software Engineering and Design (MIT Press ed. Vol. 2), 366.

ANEXOS

Anexo # 1. Factor de Peso de los Actores y Casos de Uso sin ajustar.

Factor de Peso de los Actores sin ajustar (UAW).			
Tipo de Actor	Descripción	Factor de peso	Valor
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación (API, Application Programming Interface)	1	0
Medio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto.	2	0
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica.	3	2
Factor de Peso de los Actores sin ajustar (UAW) = $\sum(\text{actores} * \text{Peso})$		1*0 + 2*0 + 3*2 = 6	

No.	Caso de uso	Cantidad de Transacciones
1	Gestionar Medios Básicos	5
2	Gestionar Movimiento de Medio Básico	5
3	Gestionar Control de Servicio	5
4	Gestionar Local	5
5	Gestionar Área o Facultad	5
6	Gestionar Responsable del Local	5
7	Gestionar Receptor de Medio Básico	5
8	Gestionar Tipo de Movimiento de Medio Básico	5
9	Visualizar medios básicos	1
10	Visualizar movimientos de medios básicos	1
11	Visualizar tarjeta de control de servicio (modelo AT-5)	1
12	Visualizar locales	1
13	Visualizar áreas o facultades	1
14	Visualizar responsables de locales	1
15	Visualizar receptores de medio básico	1
16	Visualizar tipos de movimientos de medios básicos	1
17	Imprimir medios básicos	1
18	Imprimir movimientos de medios básicos	1
19	Imprimir tarjeta de control de servicio (modelo AT-5)	1
20	Gestionar usuario	5
21	Visualizar usuario	1
22	Autenticarse	1
23	Validar Usuario	1
24	Finalizar sesión	1

Factor de Peso de los Casos de Uso sin ajustar (UUCW)			
Tipo de caso de uso	Descripción	Factor de peso	Valor
Simple	El Caso de Uso contiene de 1 a 3 Transacciones.	5	15
Medio	El Caso de Uso contiene de 4 a 7 Transacciones.	10	9
Complejo	El Caso de Uso contiene más de 8 Transacciones.	15	0
Factor de Peso de los Casos de Uso sin ajustar (UUCW)= \sum CU*Peso		5*15+10*9+15*0=165	

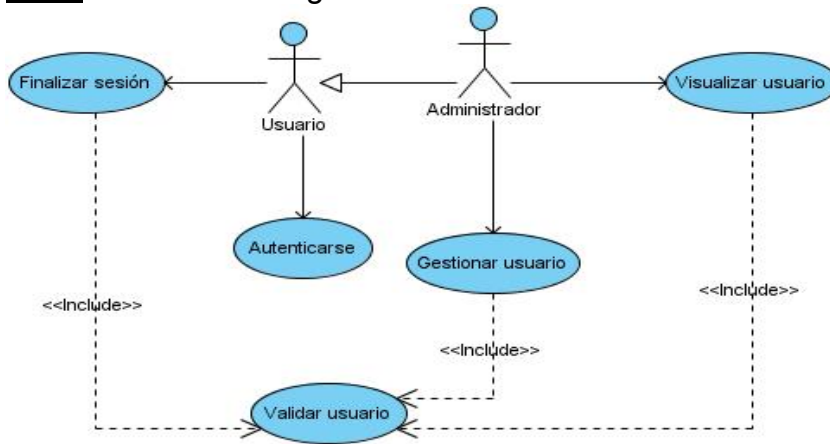
Anexo # 2. Factor de Complejidad Técnica (TCF) y Factor Ambiente (EF).

Factor de complejidad técnica (TCF)					
Factor	Descripción	Peso	Valor	Comentario	$\Sigma (\text{Peso}_i * \text{Valor}_i)$
T1	Sistema distribuido	2	0	El sistema es centralizado.	0
T2	Objetivos de performance o tiempo de respuesta	1	2	La velocidad es limitada por las entradas provistas por el usuario.	2
T3	Eficiencia del usuario final	1	3	Eficiencia media o presencia media.	3
T4	Procesamiento interno complejo	1	0	No hay cálculos Complejos.	0
T5	El código debe ser reutilizable	1	0	No se requiere que el código sea reutilizable.	0
T6	Facilidad de instalación	0.5	2	Eficiencia de requerimientos moderada.	1
T7	Facilidad de uso	0.5	4	Normal.	2
T8	Portabilidad	2	0	No se requiere que el sistema sea portable.	0
T9	Facilidad de cambio	1	3	Se requiere un costo moderado de mantenimiento	3
T10	Concurrencia	1	0	No hay concurrencia.	0
T11	Incluye objetivos especiales de seguridad	1	4	Seguridad significativa.	4
T12	Provee acceso directo a terceras partes	1	5	Los usuarios Web tienen acceso directo.	5
T13	Se requieren facilidades especiales de entrenamiento a los usuarios	1	4	El sistema es fácil de usar.	4
Total					24

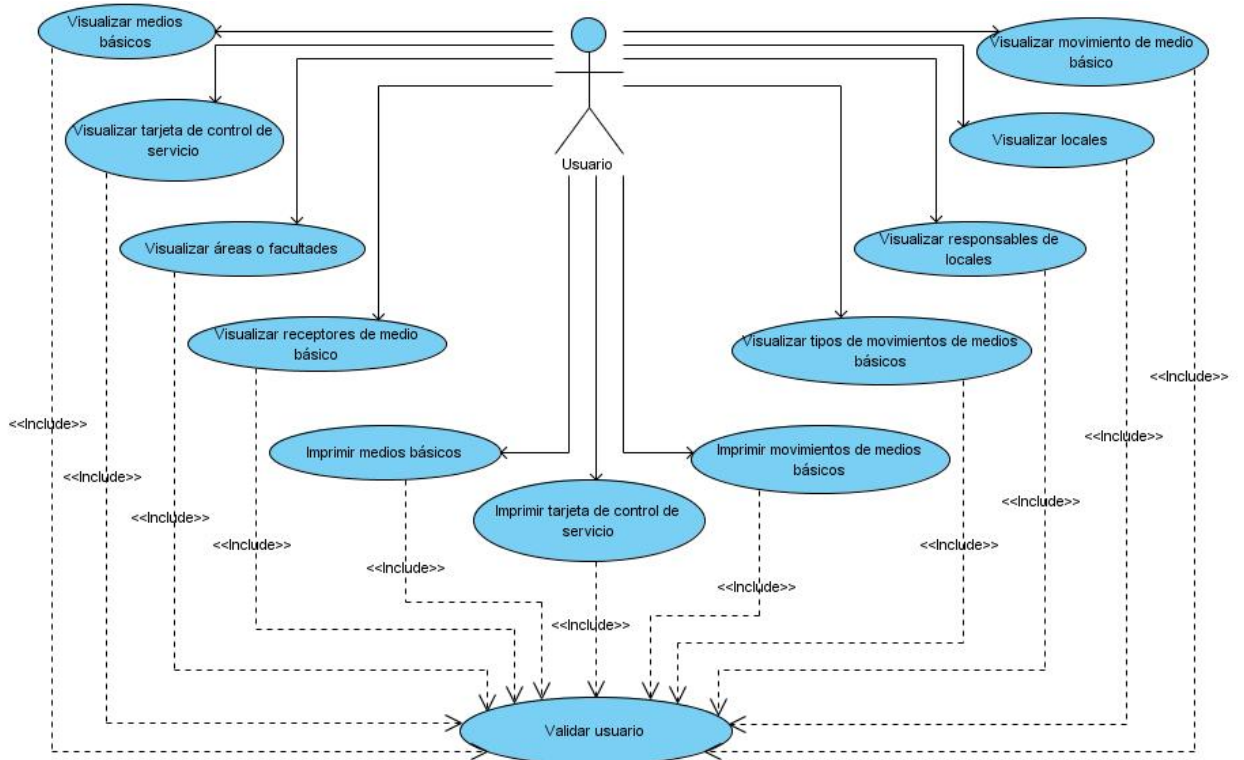
Factor Ambiente (EF)					
Factor	Descripción	Peso	Valor	Comentario	$\Sigma (\text{Peso}_i * \text{Valor}_i)$
E1	Familiaridad con el modelo de proyecto utilizado	1.5	3	El grupo está bastante familiarizado con el modelo	4.5
E2	Experiencia en la aplicación	0.5	4	La mayoría del grupo ha trabajado mucho tiempo en ésta aplicación	2
E3	Experiencia en orientación a objetos	1	4	La mayoría del grupo programa en objetos	4
E4	Capacidad del analista líder	0.5	4	Buena experiencia	2
E5	Motivación	1	5	El grupo está altamente motivado	5
E6	Estabilidad de los requerimientos	2	2	Se esperan cambios	4
E7	Personal part-time	-1	0	Todo el grupo es full-time	0
E8	Dificultad del lenguaje de programación	-1	2	Se usará lenguaje Php y JavaScript, de los cuales se tiene un nivel medio de conocimiento.	-2
Total					19.5

Anexo # 3. Diagrama de Caso de Uso del Sistema (CUS).

CUS: Subsistema Seguridad.



CUS: Subsistema Visualizar.



Anexo # 4. Descripción textual de los casos de uso del sistema.

Paquete Seguridad.

Caso de Uso	<u>Validar Usuario.</u>
Actor(es)	Este es un caso de uso incluido del resto de los casos de uso del sistema. El actor que lo inicia es el que inicie cualquiera de los casos de uso que lo incluyen.
Propósito	Comprobar que el usuario que intenta acceder a una opción restringida está autorizado.
Resumen	El caso de uso se inicia cuando se realiza alguna acción por parte del usuario actual. El sistema comprueba el nombre de usuario, contraseña y el nivel de acceso. Si su nombre de usuario y su contraseña son válidos el sistema comprueba sus privilegios y en dependencia de estos se le da entrada al lugar correspondiente. En caso contrario el sistema se va a la pantalla de iniciar sesión. El caso de uso finaliza cuando se accede a las opciones o se va a la pantalla de inicio.
Referencias	R42. Los casos de usos Gestionar medios básicos, Gestionar movimiento de medio básico, Gestionar control de servicio, Gestionar local, Gestionar área o facultad, Gestionar responsable de local, Gestionar receptor de medio básico, Gestionar tipo de movimiento de medio básico, Visualizar medios básicos, Visualizar movimientos de medios básicos, Visualizar tarjeta de control de servicio (modelo AT-5), Visualizar locales, Visualizar áreas o facultades, Visualizar responsables de locales, Visualizar receptores de medio básico, Visualizar tipos de movimientos de medios básicos, Imprimir medios básicos, Imprimir movimientos de medios básicos, Imprimir tarjeta de control de servicio (modelo AT-5), Gestionar usuario, Visualizar usuario, Finalizar sesión. Son casos de usos bases del caso de uso incluido Validar Usuarios.
Precondiciones	Usuario Autenticado.
Poscondiciones	El usuario queda validado.
Prioridad	Alta.

Caso de Uso	<u>Gestionar Usuario.</u>
Actor(es)	Administrador (inicia).
Propósito	Actualizar los usuarios del sistema.
Resumen	El caso de uso inicia cuando el Administrador accede a la opción de insertar, modificar o eliminar usuarios; inserta, modifica o elimina un usuario y finaliza cuando se cierra la aplicación o se accede a otras opciones.
Referencias	R36 – R39. Validar Usuario es un caso incluido de este caso de uso.
Precondiciones	El Administrador tiene que haberse autenticado satisfactoriamente. Para eliminar o modificar un usuario existente.
Poscondiciones	Queda actualizado el usuario.
Prioridad	Auxiliar.

Paquete Visualización.

Caso de Uso	<u>Visualizar Medios Básicos.</u>
Actor(es)	Usuario (inicia).
Propósito	Visualizar el registro de los medios básicos, que se encuentre insertado.
Resumen	El caso de uso inicia cuando el usuario selecciona la opción de visualizar el medio básico y finaliza cuando se cierra la aplicación o se accede a otras opciones.
Referencias	R25. Validar Usuario es un caso incluido de este caso de uso.
Precondiciones	Visualizador tiene que haberse autenticado satisfactoriamente.
Poscondiciones	El registro de medios básicos queda visualizado.
Prioridad	Auxiliar.

Caso de Uso	<u>Visualizar Tarjeta de Control de Servicio (modelo AT-5).</u>
Actor(es)	Usuario (inicia).
Propósito	Permitir visualizar la tarjeta de control de servicio (modelo AT-5), existente.
Resumen	El caso de uso inicia cuando usuario selecciona la opción de visualizar la tarjeta de control de servicio (modelo AT-5). Finaliza cuando se cierra la aplicación o se accede a otras opciones.
Referencias	R27. Validar Usuario es un caso incluido de este caso de uso.
Precondiciones	El usuario tiene que haberse autenticado satisfactoriamente. Para visualizar la tarjeta de control de servicio (modelo AT-5), y este tiene que existir el registro de al menos un movimiento.
Poscondiciones	La Tarjeta de control de servicio queda visualizada.
Prioridad	Auxiliar.

Caso de Uso	<u>Imprimir Tarjeta de Control de Servicio (modelo AT-5).</u>
Actor(es)	Usuario (inicia).
Propósito	Imprimir la tarjeta de control de servicio (modelo at-5) existente.
Resumen	El caso de uso inicia cuando el administrador selecciona la opción de imprimir la tarjeta de control de servicio (modelo at-5) existente, y finaliza cuando se cierra la aplicación o se accede a otras opciones.
Referencias	R35. Validar Usuario es un caso incluido de este caso de uso.
Precondiciones	Administrador tiene que haberse autenticado satisfactoriamente.
Poscondiciones	La tarjeta de control de servicio (modelo at-5) queda impreso.
Prioridad	Alta.

Paquete Control de Medios Básicos.

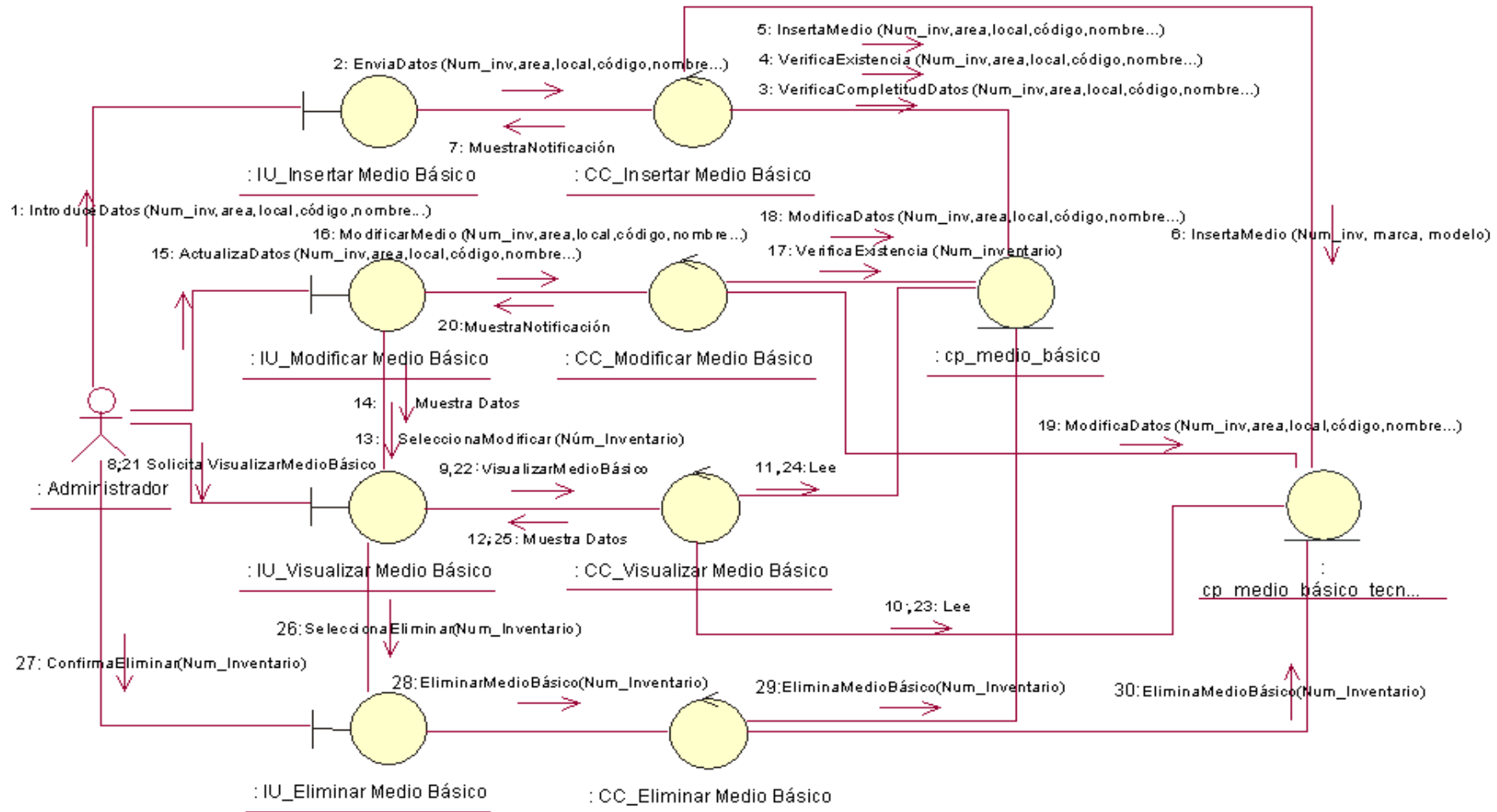
Caso de Uso	<u>Gestionar Medios Básicos.</u>
Actor(es)	Administrador (inicia).
Propósito	Permitir Insertar, modificar o eliminar un medio básico en el sistema.
Resumen	El caso de uso inicia cuando el administrador del sistema desea insertar un nuevo medio básico, modificar o eliminar un medio básico existente en el sistema. El caso de uso finaliza cuando accede a otras opciones o sale del sistema.
Referencias	R1 – R3. Validar Usuario es un caso incluido de este caso de uso.
Precondiciones	El administrador debe haber autenticado satisfactoriamente. Para modificar o eliminar un medio básico este debe existir.
Poscondiciones	Queda creado, modificado o eliminado un medio básico.
Prioridad	Alta.

Caso de Uso	<u>Gestionar Movimiento de Medio Básico.</u>
Actor(es)	Administrador (inicia).
Propósito	Permitir Insertar, modificar o eliminar un movimiento de medio básico en el sistema.
Resumen	El caso de uso inicia cuando el administrador del sistema desea insertar un nuevo movimiento de un medio básico, modificar o eliminar un movimiento de medio básico existente en el sistema. El caso de uso finaliza cuando accede a otras opciones o sale del sistema.
Referencias	R4 – R6. Validar Usuario es un caso incluido de este caso de uso.
Precondiciones	El administrador debe haber autenticado satisfactoriamente. Para modificar o eliminar un medio básico este debe existir.
Poscondiciones	Queda creado, modificado o eliminado un movimiento de medio básico.
Prioridad	Alta.

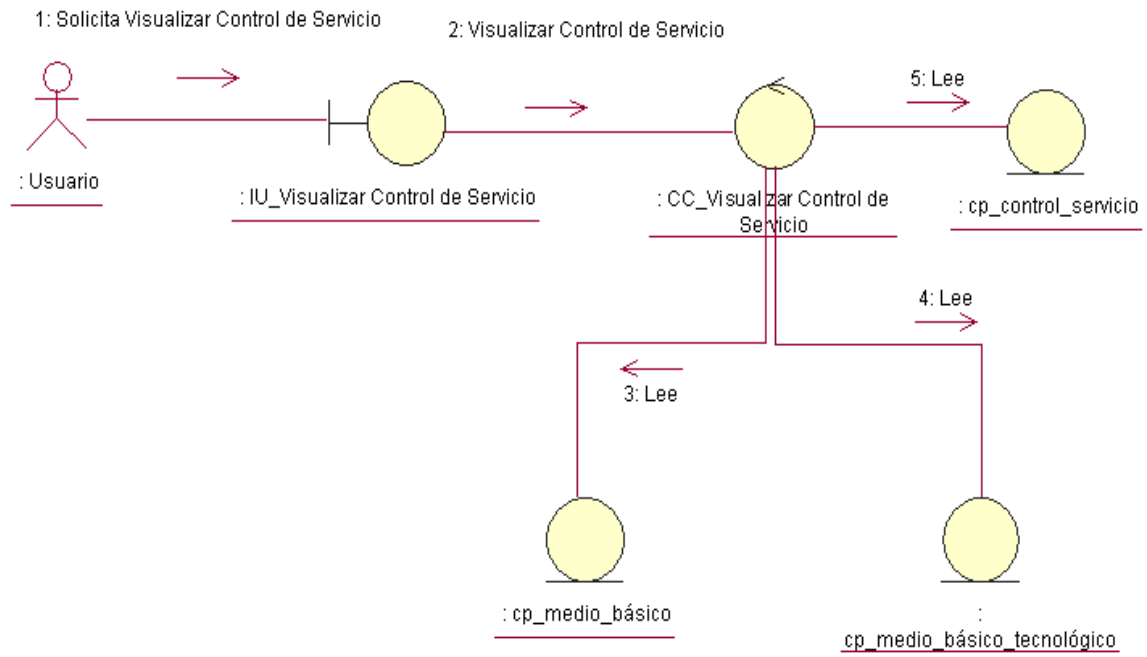
Caso de Uso	<u>Gestionar Control de Servicio.</u>
Actor(es)	Administrador (inicia).
Propósito	Permitir Insertar, modificar o eliminar un control de servicio en el sistema.
Resumen	El caso de uso inicia cuando el administrador del sistema desea insertar un nuevo control de servicio, modificar o eliminar un control de servicio existente en el sistema. El caso de uso finaliza cuando accede a otras opciones o sale del sistema.
Referencias	R7 - R9. Validar Usuario es un caso incluido de este caso de uso.
Precondiciones	El administrador debe haber autenticado satisfactoriamente. Para Modificar o Eliminar un control de servicio este debe existir.
Poscondiciones	Queda creado, modificado o eliminado un control de servicio.
Prioridad	Alta.

Anexo # 5. Diagramas de Colaboración.

Actualizar Medio Básico

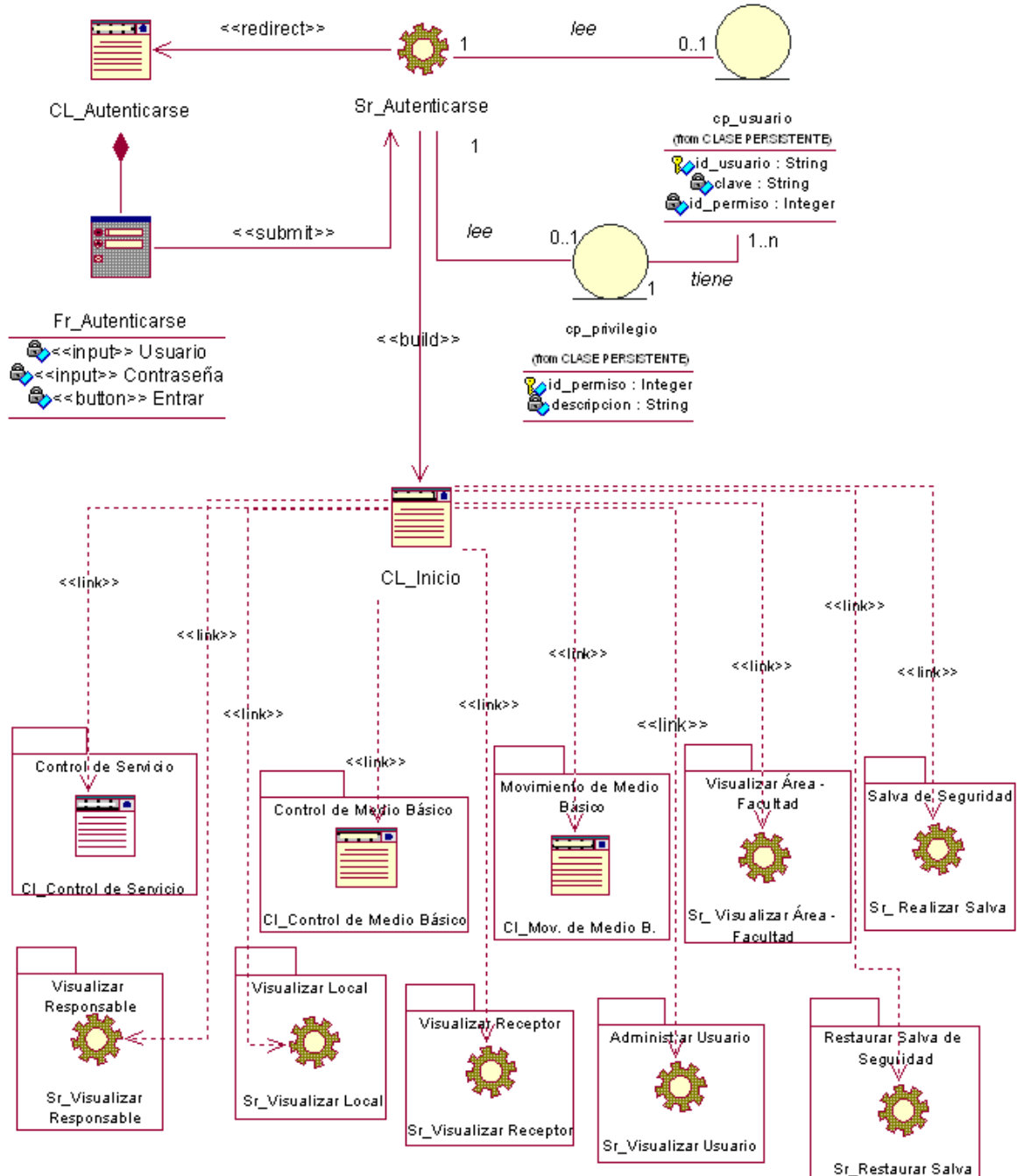


Visualizar Control de Servicio

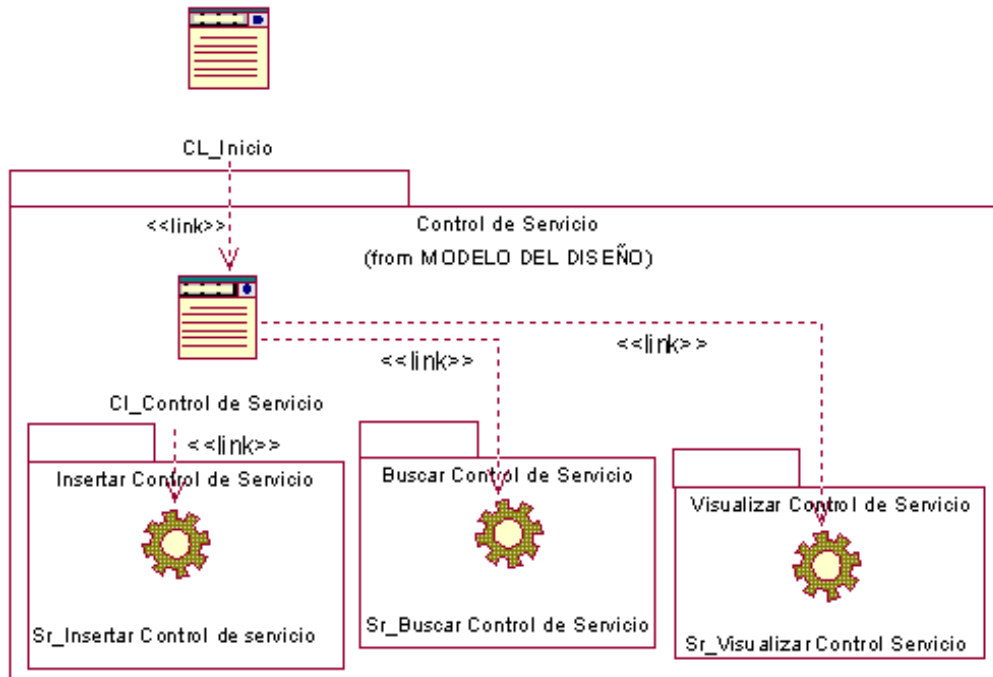


Anexo # 6. Diagramas de Clase del Diseño.

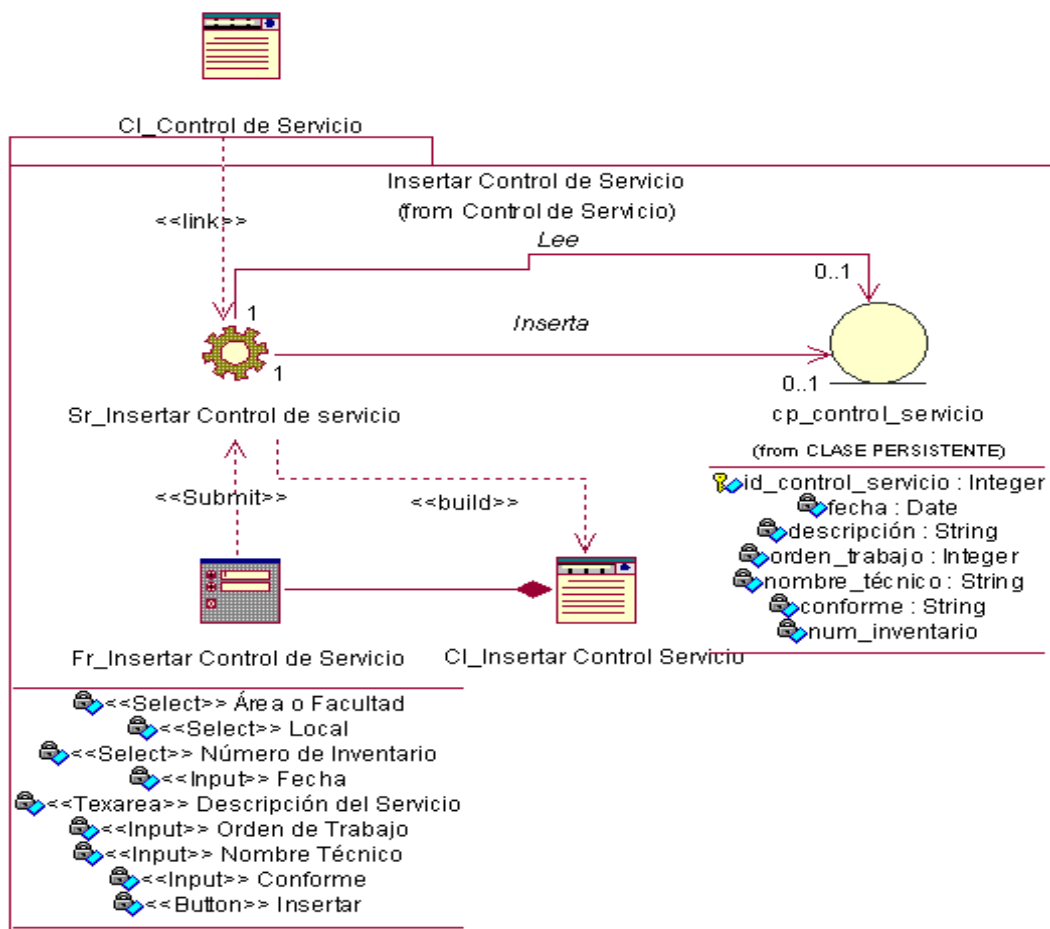
General del sistema.



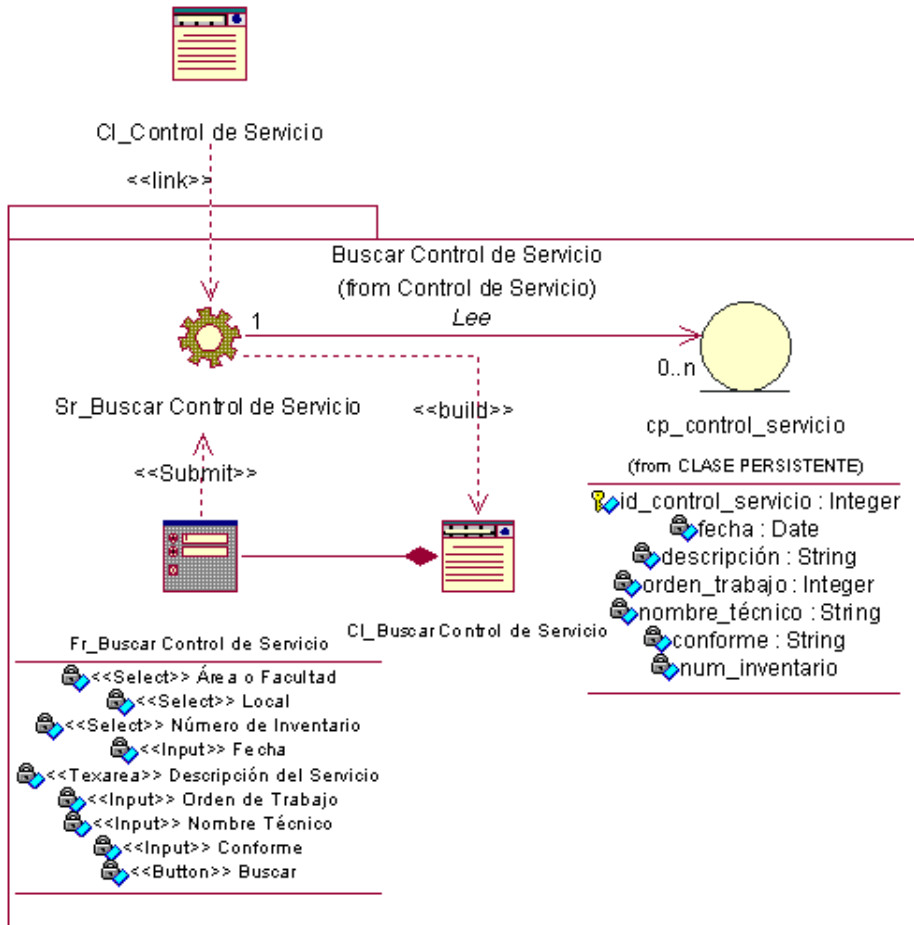
Control de Servicio



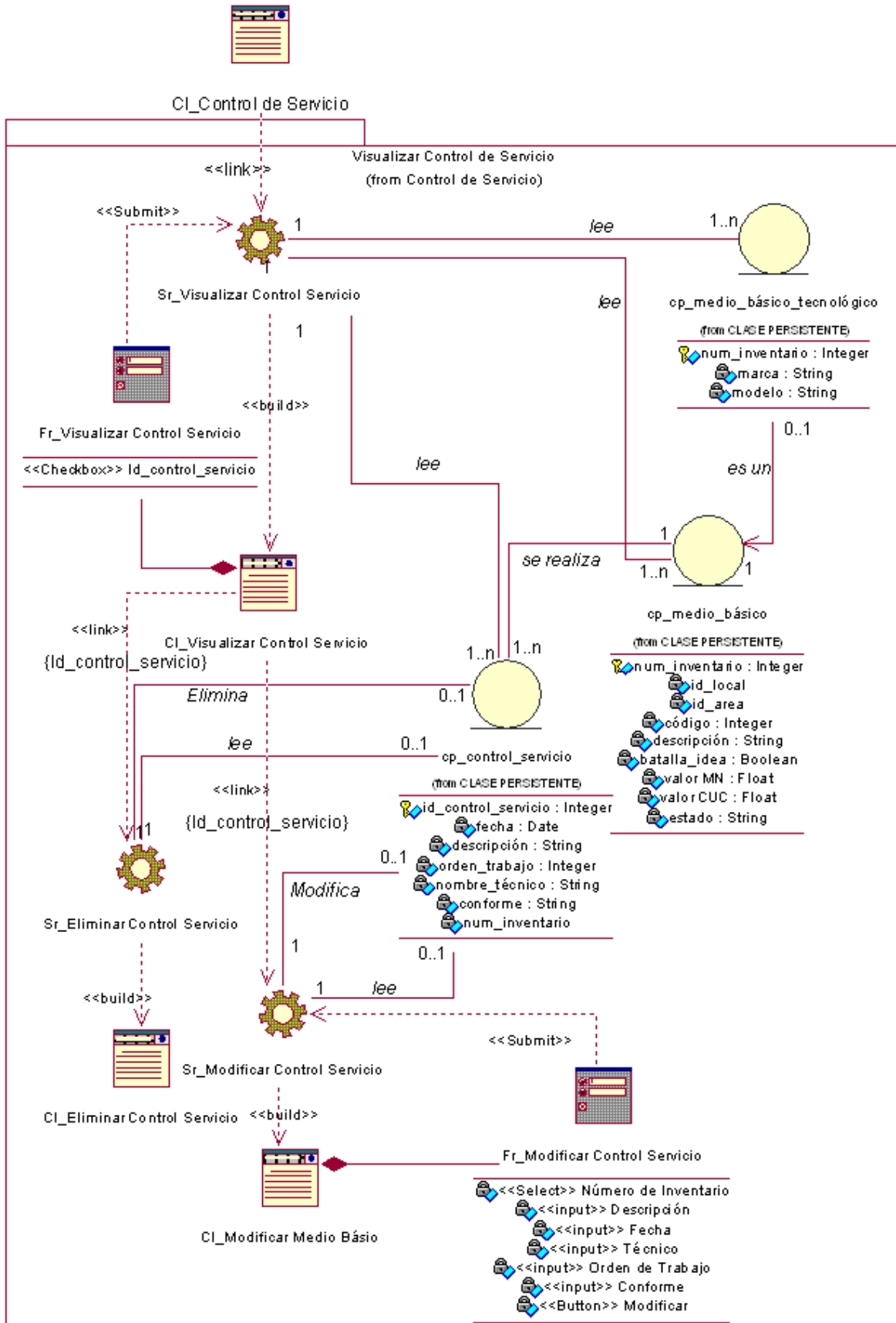
Insertar Control de Servicio



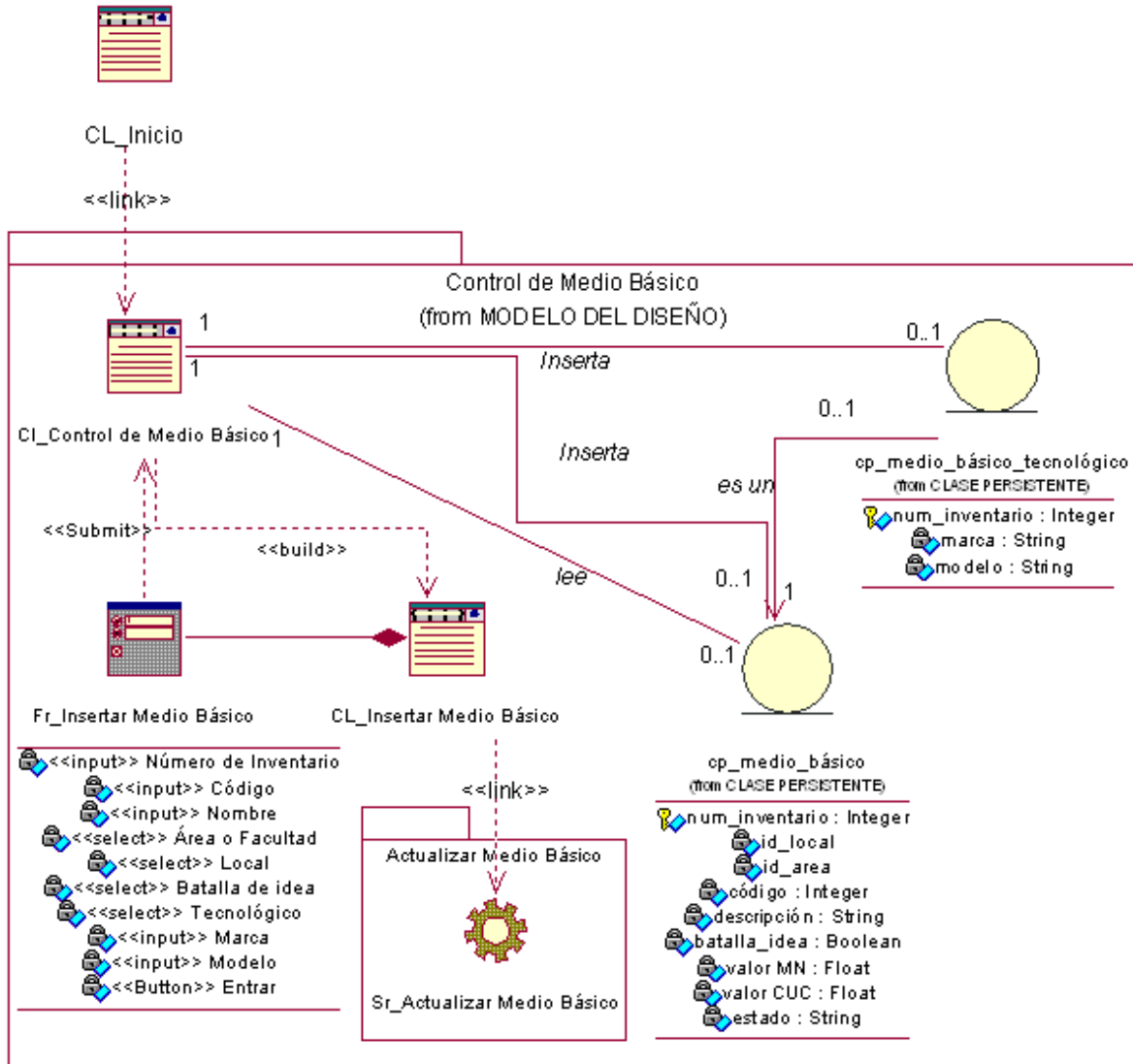
Buscar Control de Servicio



Visualizar Control de Servicio



Control de Medio Básico



Anexo # 7. Pantallas representativas del sistema.

Pantalla de autenticación del sistema.



Universidad de Granma

Jueves, 10 de Febrero del 2011

Medios Básicos
Sistema de Gestión

Ayuda Intranet Biblioteca

Si es usuario de este sistema deberá introducir sus datos para acceder al mismo.

Autenticación

Usuario:

Contraseña:

Entrar

Bienvenido al Sistema de Gestión de Información para el Control de Medios Básicos de la Facultad de Informática de la Universidad de Granma.

El sistema brinda un conjunto de utilidades para la gestión y control de los medios básicos de la Facultad de Informática aunque por sus características puede ser extendido a otras áreas sin modificación alguna. La interfaz de usuario lo hace un software de fácil manejo por lo que no se requiere de grandes conocimientos de informática, puede ser manejado con facilidad por personal no especialista sin dificultad alguna.

Este sistema ha sido diseñado e implementado solo para uso de la Universidad de Granma, el autor no se hace responsable de la pérdida o daño de la información por uso indebido del mismo.

Facultad de Informática | Universidad de Granma
© Copyright 2010 | Todos los derechos reservados

Medios Básicos

Pantalla de bienvenida.

Universidad de Granma

Jueves, 10 de Febrero del 2011

Inicio Ayuda Intranet Biblioteca Salir

Medios Básicos
Sistema de Gestión

Medios Básicos

Usuario activo: mayra

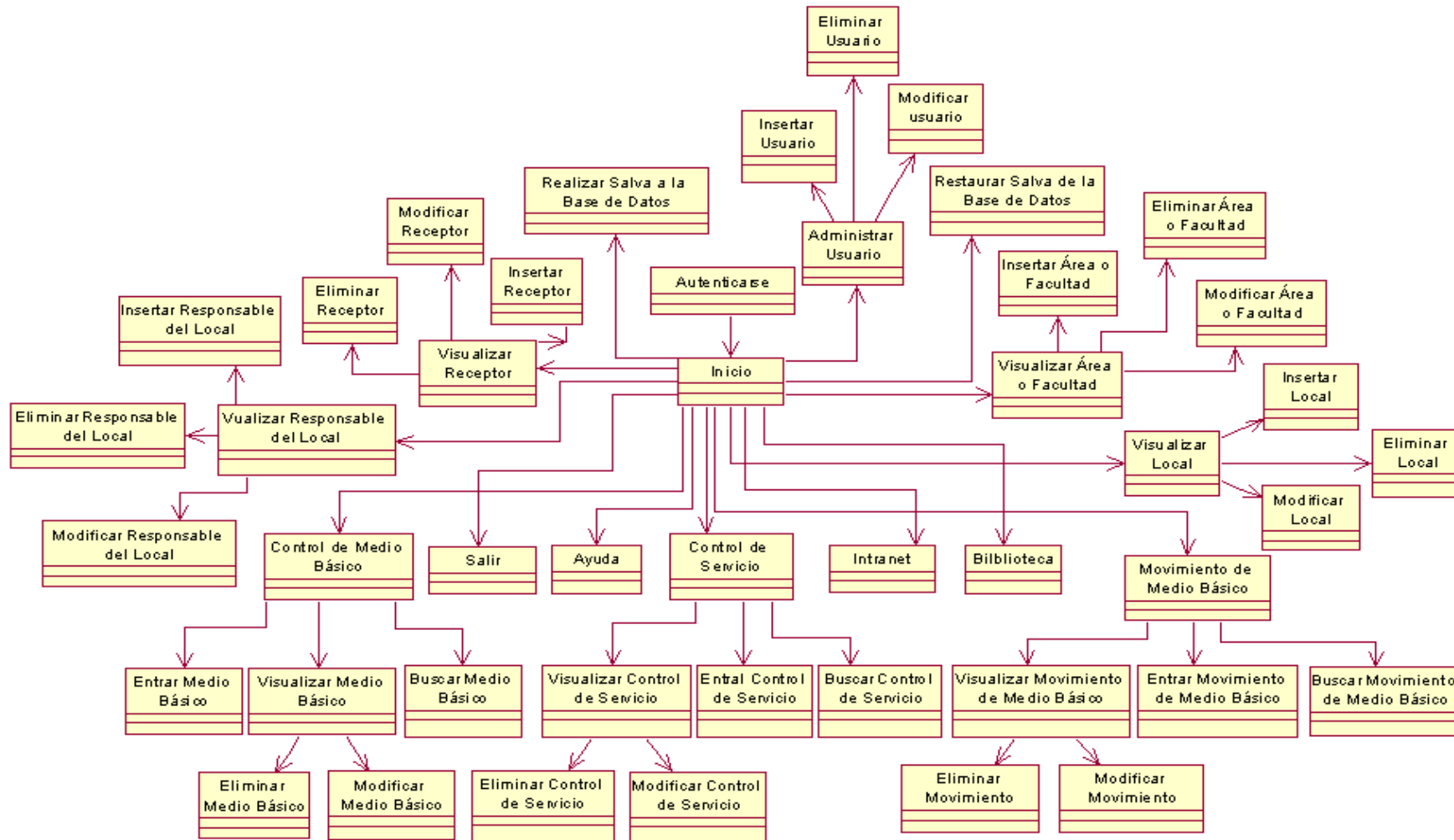
Medio Básico	
Control de Medio Básico	▶
Control de Servicio	▶
Movimiento de Medio Básico	▶
Configuración	
Área-Facultad	
Local	
Responsable de Local	
Receptor de Movimiento	
Administración	
Administrar Usuario	
Salva de Seguridad	
Restaurar Salva	

Bienvenido al módulo de administración del Sistema de Gestión y Control de los Medios Básicos de la Facultad de Informática de la Universidad de Granma.

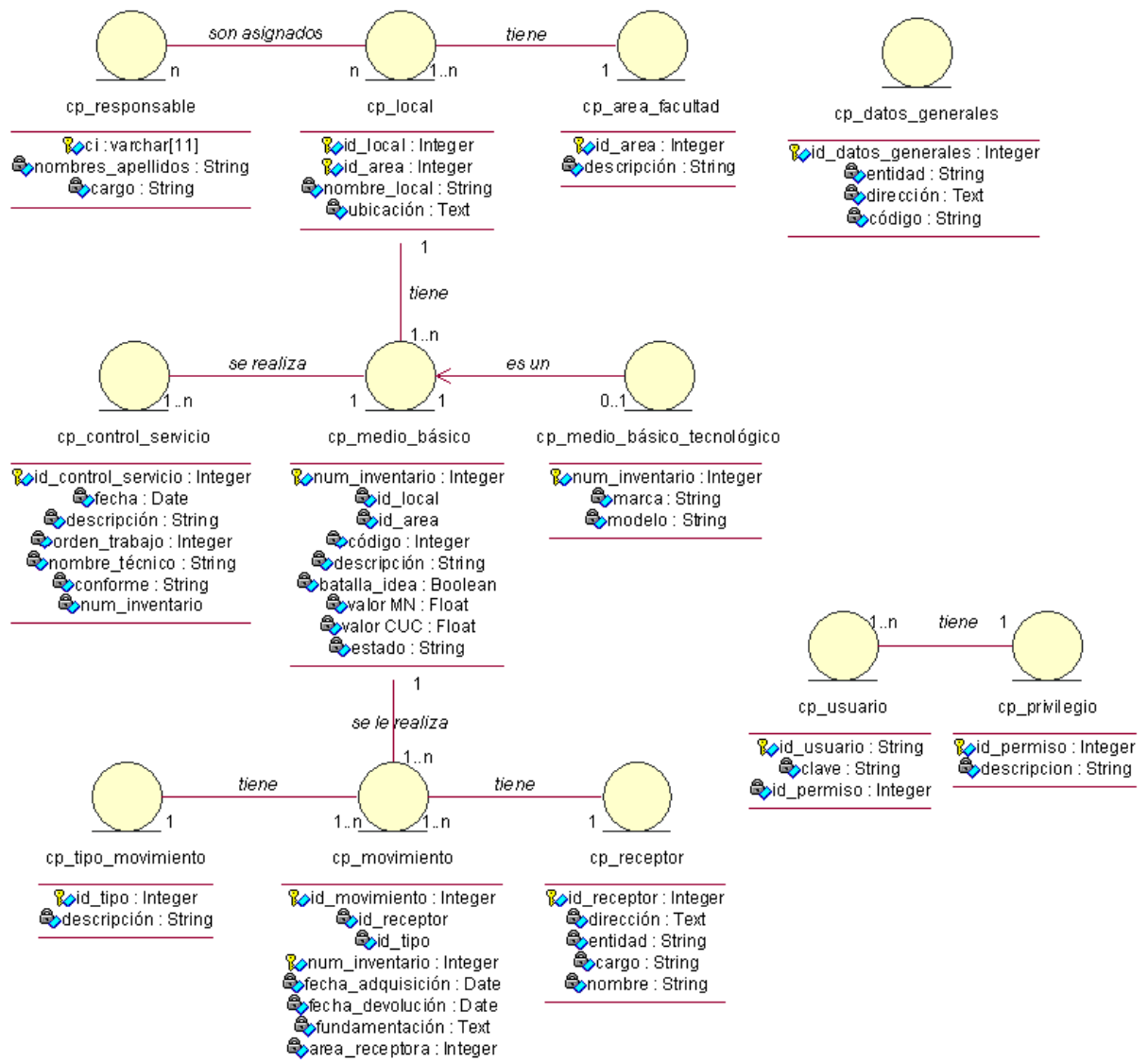
Usted además de consultar información y generar reportes sobre los medios básicos existentes, podrá dar de alta , eliminar o mover medios básicos. Tambien en este módulo podrá controlar los servicios prestados a dichos medios.

Facultad de Informática | Universidad de Granma
© Copyrigh 2010 | Todos los derechos reservados

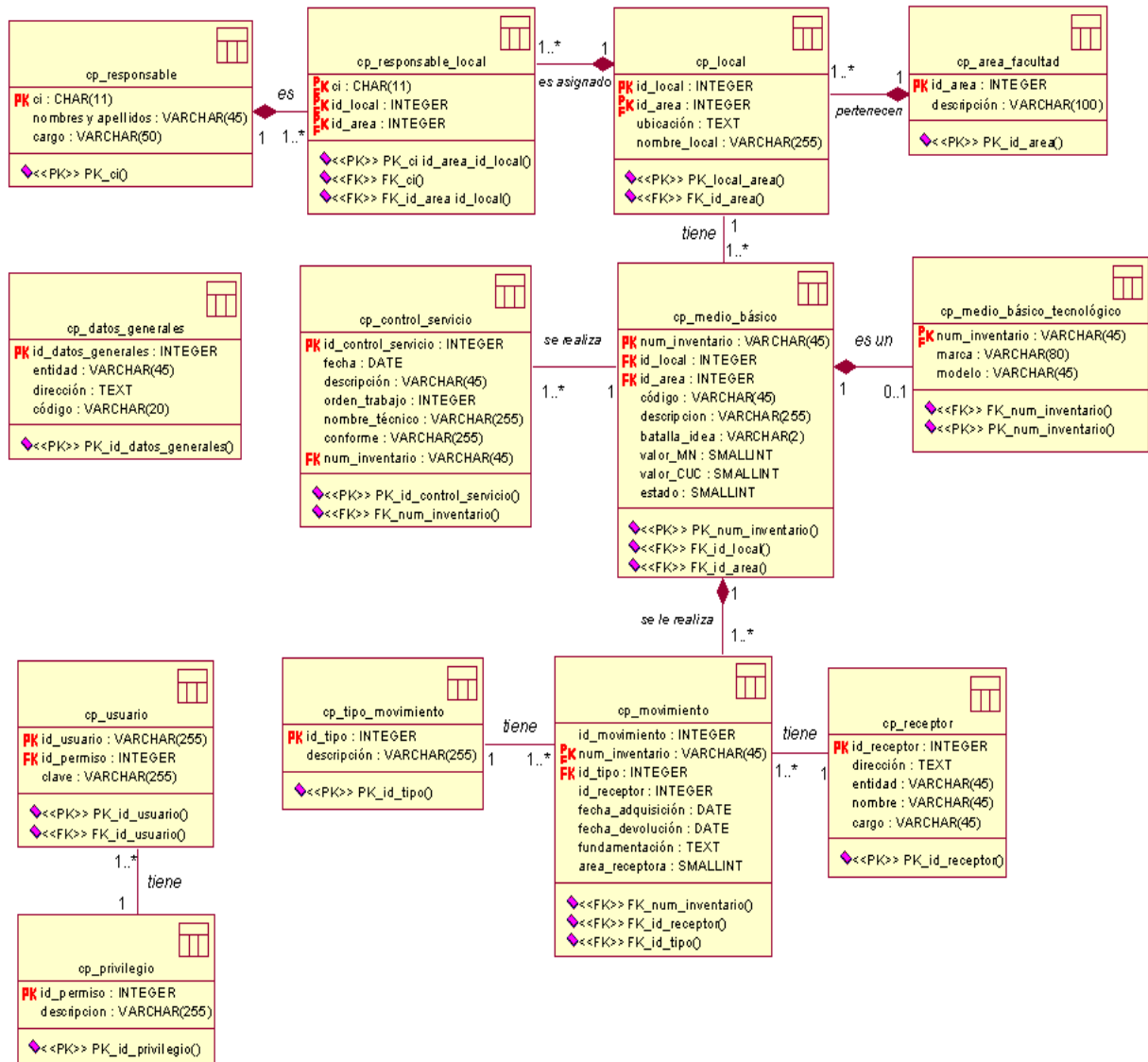
Anexo # 8. Mapa de Navegación.



Anexo # 9. Diagramas de Clases Persistentes.



Anexo # 10. Modelo de Datos.



Anexo # 11. Diagramas de Componentes.

