



Universidad de Pinar del Río

Facultad de Ciencia Técnicas

Departamento de Informática

Trabajo de Diploma.

Título: Sistema Automatizado para el Control de Piezas de Computadora (SACPIC).

(Tesis en opción al título de Ingeniero en Informática)

Autor: José María Otacoma Jacho

Pinar del Río-2011



Universidad de Pinar del Río

Facultad de Ciencia Técnicas

Departamento de Informática

Trabajo de Diploma.

Título: Sistema Automatizado para el Control de Piezas de Computadora (SACPIC).

(Tesis en opción al título de Ingeniero en Informática)

Autor: José María Otacoma Jacho.

Tutora: Ing. Maite de la Caridad Hernández Soberao.

Asesor: Msc. Alberto Serrano.

Pinar del Río-2011

Proyecta lo difícil partiendo de donde aún es fácil.

Realiza lo grande partiendo de donde aún es
pequeño.

Todo lo difícil comienza fácil.

Todo lo grande comienza pequeño.

Lao Tzu. Tao Te King. Versículo 63

PÁGINA DE ACEPTACIÓN

Presidente del Tribunal

Secretario

Vocal

Ciudad y fecha:

DECLARACIÓN DE AUTORIDAD

Declaro que soy autor de este Trabajo de Diploma y que autorizo a la Universidad de Pinar del Río, a hacer uso del mismo, con la finalidad que estime conveniente.

Firma: _____

José María Otacoma Jacho

Cheyo_quince@hotmail.com

José María Otacoma Jacho autoriza la divulgación del presente trabajo de diploma bajo licencia Creative Commons de tipo **Reconocimiento No Comercial Sin Obra Derivada**, se permite su copia y distribución por cualquier medio siempre que mantenga el reconocimiento de sus autores, no haga uso comercial de las obras y no realice ninguna modificación de ellas. La licencia completa puede consultarse en: <http://creativecommons.org/licenses/by-ncnd/2.5/ar/legalcode>

José María Otacoma Jacho autoriza al Departamento de Informática adscrito a la Universidad de Pinar del Río a distribuir el presente trabajo de diploma en formato digital bajo la licencia Creative Commons descrita anteriormente y a conservarlo por tiempo indefinido, según los requerimientos de la institución, en el repositorio de materiales didácticos disponible en: <http://repointo.upr.edu.cu>

José María Otacoma Jacho autoriza al Departamento de Informática adscrito a la Universidad de Pinar del Río a distribuir el presente trabajo de diploma en formato digital bajo la licencia Creative Commons descrita anteriormente y a conservarlo por tiempo indefinido, según los requerimientos de la institución, en el repositorio de tesis disponible en: <http://revistas.mes.edu.cu>

AGRADECIMIENTOS

La vida es lo más valioso y hermoso que nos dio nuestro señor Dios y que cada día hay que vivirlo intensamente conjuntamente con nuestros seres queridos, encaminándoles por el buen sendero del amor como es la familia. Para llevar a cabo cada tarea, cada ejercicio cotidiano contamos con muchas personas que nos ofrecen ayuda, nos apoyan en los momentos que más necesitamos y nos hacen tener confianza en nosotros, y de esta manera plasmar nuestro objetivo y alcanzar el triunfo, a esas personas que me han ayudado les agradezco de todo corazón por haber estado a la lejanía de una palabra para decirme "aquí estoy yo".

Gracias a mi Tutora la Ing. Maite de la Caridad Hernández Soberao por su dedicación en el cumplimiento de su tarea encomendada.

Gracias a mi Asesor Msc. Alberto Serrano por ser parte fundamental en el desarrollo del trabajo de Investigación y el don de persona.

Gracias a mi lindísima esposa y mis bellos hijos por ser todo para mí y siempre entregarme su amor y confianza a ellos les debo el honor de ser hoy quien soy.

DEDICATORIA

Dedico este trabajo de diploma a todas aquellas personas que de una forma u otra me han apoyado en la realización del mismo.

- A mi lindísima esposa Wilma por todo su cariño y paciencia que me ha tenido en los momentos más difíciles de nuestras vidas.
- A mis bellos hijos Fernando, William, Darío, Sheily, Anahi por su amor su tiempo y ayuda siempre que los necesité y por ser la fuerza que me inspira a seguir adelante.

A mi queridísima madre Carmen y mis hermanos por su apoyo incondicional los cuales estuvieron presentes en los momentos que más necesite de ellos, mil gracias.

A mis compañeros de promoción Mayra, Paul, Víctor, Juan Carlos, Álvaro por demostrar sus dones de compañerismos y ser amigos de buen corazón que siempre me estuvieron alentando a continuar con mis estudios.

RESUMEN

A la Universidad Pinar del Río son asignadas para cada facultad una serie de computadoras, los cuales cumplen una función específica para las tareas diarias de los profesores y estudiantes. Actualmente el proceso de controlar las piezas de cada computadora, de cada local, perteneciente a cada facultad (en el caso que se realice) se hace de forma manual, permitiendo posibles pérdidas de piezas por no llevar un control exacto de las mismas. Analizando esta situación se hace evidente que estamos en presencia de un problema: Inexistencia de un sistema que permita llevar a cabo el control de las piezas de las computadoras existentes en la Universidad de Pinar del Río de manera eficiente.

El propósito de éste trabajo radica en el diseño e implementación de un Sistema Automatizado para el Control de Piezas de Computadoras que permita gestionar la información necesaria para tener un control exacto de las Piezas. El sistema fue implementado sobre el lenguaje de programación Java y empleando como Sistema Gestor de Base de Datos PostgreSQL.

En la Base de Datos se encontrarán almacenados tanto los datos correspondientes a las Piezas de las Computadoras, así como la ubicación de las mismas. La interfaz gráfica desarrollada es una aplicación de escritorio mediante la cual el usuario podrá gestionar toda la información residente en la Base de Datos.

Como resultado se obtuvo un sistema como herramienta para llevar a cabo el proceso de gestión de las piezas en computadoras, contribuyendo a un mejor control de las mismas.

SUMMARY

There are series of computer which are assigned to each Department at the University "Pinar Del Río" do a specific function for the professors, daily task and students. Actually the process of controlling the pieces of each computer, of each local, belonging to each ability (in the case that is carried out) it is made in a manual way, allowing possible losses of pieces for not taking an exact control of the same ones. Analyzing this situation, it becomes evident that we are in presence of a problem: Nonexistence of a system that allows to carry out the control of the pieces of the existent computers at the University of Pinar del Río in an efficient way.

The purpose of this work resides in the design and implementation of an Automated System for the Control of Pieces of Computers that allows to negotiate the necessary information to have an exact control of the Pieces. The system was implemented on the programming language Java and using as System database Agent PostgreSQL.

In the database they will be stored the data corresponding to the Pieces of the Computers so much, as well as the location of the same ones. The developed graphic interfaz is a desk application by means of which the user will be able to negotiate all the residing in information in the database.

As a result it was obtained a system like tool to carry out the process of administration of the pieces in computers, contributing to a better control of the same ones.

TABLA DE CONTENIDO

INTRODUCCIÓN	1
CAPÍTULO 1. CARACTERIZACIÓN DEL PROBLEMA	5
1.1 Computadora, su impacto e importancia para la sociedad	5
1.2 Análisis de sistemas dedicados al control de piezas de Computadoras	7
1.2.1 Everest	8
1.3 Modelo Conceptual del SACPIC	8
1.4 Análisis de Factibilidad de SACPIC	13
1.4.1. Cálculo de Casos de Uso y Actores sin ajustar	13
1.4.2. Cálculo de Puntos de Casos de Uso ajustados	17
1.4.2.1. Cálculo del Factor de Complejidad Técnica	17
1.4.2.2 Cálculo del Factor de Ambiente	20
1.4.3. De los Puntos de Casos de Uso a la Estimación del Esfuerzo	22
1.4.4. Tiempo de desarrollo	23
1.4.5. Costo Total del Proyecto	24
CAPÍTULO 2. DISEÑO DEL SACPIC	25
2.1 Herramientas utilizadas para el diseño del SACPIC	25
2.1.2 Proceso Unificado de Desarrollo de Software (RUP)	26
2.1.3 Lenguaje de Modelado Unificado (UML)	27
2.1.4 Enterprise Architect (EA)	28
2.2 Modelo de Datos del SACPIC	29
2.3 Diseño de la Interfaz de Usuario del SACPIC	31
2.3.1 Requerimientos Funcionales	31
2.3.2 Requerimientos No Funcionales	37

2.3.3	Actores del sistema.....	38
2.3.4	Diagrama de Casos de Uso.....	39
2.3.4.1	Descripción.....	40
CAPÍTULO 3.	IMPLEMENTACIÓN DEL SACPIC.....	47
3.1.	Implementación de la Base de Datos del SACPIC.	47
3.1.1.	Microsoft Access	48
3.1.2.	MySQL	48
3.1.3.	PostgreSQL.....	50
3.1.4.	Selección de PostgreSQL.....	50
3.2.	Implementación de la Interfaz de Usuario del SACPIC.....	53
3.2.1.	Lenguajes de programación.	53
3.2.1.1.	C++.....	53
3.2.1.2.	C #.....	54
3.2.1.3.	Java	55
3.2.1.4.	Selección de Java como lenguaje para ser utilizado	55
3.2.1	NetBeans como entorno de desarrollo para lenguaje seleccionado.	60
3.3	Implementación de la Ayuda.	62
3.4	Modelo de Componentes del Sistema.	64
3.5	Seguridad del SACPIC.	65
CONCLUSIONES.....		67
RECOMENDACIONES.....		68
REFERENCIAS BIBLIOGRAFICAS.....		69
ANEXOS		72
ANEXO 1	Descripción del CU “Gestionar datos del Disco Duro”.....	72
ANEXO 2	Diagrama de Clases del Análisis para el CU “Gestionar datos Computadora”.....	78

ANEXO 3	Diagramas de Secuencia para el CU “Gestionar datos de Computadora”	79
ANEXO 4	Diagrama de Clases del Diseño para el CU “Gestionar datos de Computadora”	82
ANEXO 5	Diagrama de Clases del Análisis para el CU “Gestionar datos del Disco Duro”	83
ANEXO 6	Diagramas de Secuencia para el CU “Gestionar datos del Disco Duro”	84
ANEXO 7	Diagrama de Clases del Diseño para el CU “Gestionar datos del Disco Duro”	87

INTRODUCCIÓN

El desarrollo científico - técnico comenzó a tener un papel significativo en las economías mundiales a principio de la década del 80, siendo hoy en día la clave para el adelanto de las economías de los países. Paralelo a ello el mundo desarrollado a dictado medidas radicales, las cuales se han acentuado con las políticas neoliberales.

Las Nuevas tecnologías de la información y el conocimiento (NTIC), han sido identificadas como elementos claves de la sociedad para lograr ventajas competitivas en un período de tiempo determinado. La modelación de la gestión de las TIC se han desarrollado para servir de guía en un sin número de organizaciones del mundo, fundamentalmente en países desarrollados, también las concepciones sobre los recursos humanos deben impulsar la expansión del conocimiento a todas las esferas de la sociedad [1].

Debido a la actual crisis económica de nuestro país, se está dando pasos para lograr un control más estricto de todos los recursos donde la falta del mismo afecta a nuestras empresas e instituciones.

Las nuevas tecnologías de la información y las comunicaciones, ponen al desnudo los problemas más agravantes de nuestra sociedad y como enfrentarlos de forma eficiente con la ayuda de herramientas que permiten realizar estudios científicos en cada caso [1].

A la Universidad Pinar del Río son asignadas para cada facultad una serie de computadoras los cuales cumplen una función específica para las tareas diarias de los profesores y estudiantes.

Se está en las puertas de un fenómeno de posibles pérdidas de medios básicos como son los medios informáticos, eslabón principal en el desarrollo y la calidad del proceso que se realiza en cada unas de sus entidades.

Actualmente el proceso de controlar las piezas de cada computadora de cada local perteneciente a cada facultad (en el caso que se realice) se hace de forma

manual, permitiendo posibles pérdidas de piezas por no llevar un control exacto de las mismas.

Analizando esta situación se hace evidente que estamos en presencia de un **problema**: Inexistencia de un sistema que permita llevar a cabo el control de las piezas de las computadoras existentes en la Universidad de Pinar del Río de manera eficiente.

Hipotéticamente, si se diseña una aplicación informática para la organización y almacenamiento de datos, se estará contribuyendo a mantener el control de las piezas de cada computadora evitando así las pérdidas de las mismas.

Por ese motivo, el **objetivo general** de esta investigación es elaborar una aplicación informática para el control de las piezas de las Computadora (desarrollada en Java) con una Base de Datos incluida (basada en PostgreSQL) que gestione la información relacionada de manera eficiente.

Del objetivo general de esta investigación se derivan los siguientes **objetivos específicos**:

- Revisión de la información relacionada con las partes de la Computadora, así como la importancia del control de piezas de las mismas.
- Diseñar una Base de Datos que permita almacenar la información relacionada con las computadoras y sus piezas.
- Diseñar e implementar una aplicación informática que sea capaz de gestionar y consultar la información almacenada y permita:
 1. Inserción, modificación, eliminación y búsqueda de la información.
 2. Muestra de Reportes relacionados con las piezas de las computadoras así como su estado técnico.
 3. Ayuda en línea.
 4. Seguridad de la Base de Datos y el sistema mediante niveles de usuarios.

El **objeto** de la presente investigación está constituido por los datos de las piezas de las computadoras y como **campo de acción** la gestión de la información de dichos datos.

Las **tareas** a desarrollar para darle cumplimiento a los objetivos son:

- Revisión de la información necesaria que sirva de punto de partida para la investigación.
- Seleccionar y estudiar la Plataforma, Lenguaje y Sistema Gestor de Base de Datos (SGBD) a utilizar en el desarrollo de la aplicación.
- Diseñar e implementar la Base de Datos con el SGBD seleccionado, que almacene toda la información necesaria para ser gestionada.
- Desarrollar una aplicación con el uso de la plataforma y lenguaje seleccionado que gestione la Base de Datos.
- Realizar pruebas al software con datos reales.

Los métodos científicos utilizados en esta investigación fueron:

Métodos Teóricos

Histórico: Utilizado en la revisión de los textos relacionados con la información.

Inducción – Deducción: Utilizado en los procesos de captura de los requerimientos, análisis, diseño e implementación del software.

Análisis – Síntesis: Aplicado en los procesos de captura de los requerimientos, análisis y desarrollo del software.

Modelación: Aplicado en la modelación de los datos necesarios para el diseño de la Base de Datos así como en su gestión con uso del software.

Métodos Empíricos

Entrevista: Empleado en la captación de los requerimientos a cumplir por el software.

Revisión bibliográfica: Utilizado en el estudio de los documentos existentes.

El presente trabajo se estructuró en 3 capítulos, a continuación se brinda una breve descripción de los mismos:

CAPÍTULO 1. CARACTERIZACIÓN DEL PROBLEMA.

En el capítulo 1 se realiza una caracterización de las computadoras, su impacto e importancia para la sociedad así como la necesidad del control de sus piezas. Se hace un análisis crítico de la existencia de sistemas afines con la actual investigación. Se presenta el Modelo Conceptual del SACPIC, el cual permite ver los conceptos o entidades empleados y las relaciones existentes entre estos. Y finalmente se brinda una estimación del costo en el que se incurre al diseñar y desarrollar el software.

CAPÍTULO 2. DISEÑO DEL SACPIC.

En este capítulo se trata todo lo referente al diseño del SACPIC. Primeramente se realiza una caracterización de la metodología utilizada, el Lenguaje de Modelado y la herramienta CASE seleccionada para realizar el diseño. Además se analiza el modelo de datos del sistema, se definen Requerimientos Funcionales y No Funcionales del mismo, Actores, Casos de Usos y el Diagrama de Casos de Uso así como la descripción de uno de los Casos de Uso más importantes dentro del sistema.

CAPÍTULO 3. IMPLEMENTACIÓN DEL SACPIC.

Para el desarrollo del presente capítulo se realizó una caracterización de algunos sistemas gestores de Base de Datos existentes y se explica la selección del sistema gestor utilizado para la implementación de la Base de datos del SGCTP. Además se realiza un análisis de algunos lenguajes de programación existente y las características más importantes justificando la selección del lenguaje para el desarrollo de la aplicación. Se muestra el Modelo de Componentes resultante del SACPIC. Se aborda lo referido a la implementación de la ayuda y finalmente se describe cómo fue implementada la seguridad del sistema

CAPÍTULO 1. CARACTERIZACIÓN DEL PROBLEMA

En el presente capítulo se ofrece una caracterización del concepto de Computadora, los avances que ha traído la misma a la humanidad, así como la utilización de la computadora en los lugares de trabajo.

En el epígrafe 2 se analiza un software que de manera general es afín con la actual investigación, se realiza una caracterización del mismo exponiendo las razones por las que no resuelve la problemática presente en SACPIC.

El tercer epígrafe muestra el Modelo Conceptual del SACPIC, el cual permite ver los conceptos o entidades empleados y las relaciones existentes entre estos.

Finalmente se realiza una estimación del costo en el que se incurre al desarrollar el sistema propuesto. La estimación realizada fue mediante el método de Puntos de Caso de Uso.

1.1 Computadora, su impacto e importancia para la sociedad.

Nuestros días se caracterizan por un explosivo, colosal y omnipresente desarrollo de la técnica, y su aplicación cada vez más extensa a todos los ámbitos de la vida humana. Sin duda alguna, el más significativo desarrollo tecnológico durante el último siglo, ha sido la construcción de computadoras de finalidades generales, capaces de hacer cosas que en el hombre se consideran como comportamiento inteligente. El desarrollo tecnológico alcanzado por el hombre a finales del siglo XIX y principios del XX, los conocimientos acumulados en las distintas ciencias como la Matemática, favoreció el surgimiento teórico de la construcción de las computadoras, materializado en la primera mitad de este siglo.[2]

La computadora es una máquina capaz de realizar una gran cantidad de cálculos aritméticos y procesos de control a una gran velocidad (procesos repetitivos y

tediosos que desgastan al hombre). La computadora es una de las herramientas más poderosas de la sociedad actual. [3]

La computación es el método idóneo para facilitar el registro, la elaboración y procesamiento de la información, así como los cálculos matemáticos para su análisis y para lograr la adopción de decisiones. En todos los sentidos la computación constituye una herramienta que ayuda a resolver los problemas que se presentan y esta ayuda no puede ni debe ser subestimada [3].

Las nuevas tecnologías de la información han permitido la rápida difusión de los conocimientos científicos, contribuyendo sin lugar a dudas a la introducción de nuevas técnicas en el desarrollo de la producción material y los servicios. Las grandes redes de computadoras y las novedosas tecnologías en las telecomunicaciones permiten hoy en día que la información pueda circundar el mundo a altas velocidades.

El progreso técnico ha conducido al ser humano a elaborar sus formas de producción, de consumo y de ocio. A partir del siglo XVIII los desarrollos técnicos han comenzado a plantear dificultades de control efectivo. Asimismo, la cantidad de información relativa a personas, datos técnicos, estadísticas, documentación, ha ido creciendo considerablemente [2].

Tanto el control de las máquinas, como la ordenación y el acceso directo a toda esta información han requerido la invención de un aparato que sea capaz de reproducir algunos aspectos característicos de la capacidad mental humana, y que gracias a ello pueda actuar como auxiliar del hombre. Este aparato es lo que conocemos con el nombre de computadora [3].

En las últimas décadas, las computadoras y la tecnología de la computación se han introducido más y más en los lugares de trabajo, a menudo transformando considerablemente los tipos de trabajos que las personas realizan y la manera en que los hacen. Algunas personas respondieron a estos cambios con entusiasmo, recibiendo con agrado la oportunidad de aprender nuevas técnicas y sintiendo

emoción por las eficiencias que las nuevas tecnologías hicieron posibles. Otros respondieron de manera diferente, se encontraban a gusto con sus empleos como eran, resistieron la necesidad de volver a ser capacitados y sintieron temor, a menudo con razón, de que las nuevas tecnologías podrían eliminar sus empleos por completo [3].

Actualmente los sistemas computacionales son más fáciles de utilizar, y ahora es clara su importancia en el campo de trabajo. Pero la tecnología de la computación sigue desarrollándose a un paso increíblemente rápido, y no es posible saber con certeza cuáles serán todas las aplicaciones de esta tecnología en el futuro, porque hay que mantenerse al día con los nuevos adelantos en el campo de la computación.

Actualmente en la Universidad de Pinar del Río existe una serie de computadoras que han sido asignadas para facilitar el trabajo diario tanto de los profesores como de los estudiantes y que es necesario llevar sobre estas un estricto control para evitar cualquier pérdida de alguna pieza de las mismas.

El control descrito anteriormente, en la actualidad, se realiza de forma manual, lo que implica falta de la agilidad y la necesidad de la presencia de una aplicación que de forma eficiente permita gestionar la información relacionada con las piezas de las computadoras.

1.2 Análisis de sistemas dedicados al control de piezas de Computadoras.

Durante la búsqueda de sistemas encaminados al control de piezas de computadora se obtuvieron resultados relacionados en su mayoría con las partes internas de las mismas. En dichos sistemas se brindan reportes de la mayoría de las características internas de cada pieza de una computadora. A continuación se muestran las características de uno de estos sistemas y se brindan las razones por las cuales el mismo no resuelve la problemática que se presenta en esta investigación.

1.2.1 Everest

El objetivo del programa Everest es precisamente identificar el hardware de las computadoras, permitiendo buscar los drivers de los dispositivos, por ejemplo: tarjeta de sonido, tarjeta gráfica, tarjeta de red, placa base, etc. Everest permite generar un informe con la descripción de los dispositivos del hardware del PC. Este sistema controla los datos internos de los componentes de la computadora, no los atributos que se quieren controlar en esta investigación, relacionados con el inventario de cada máquina a la hora de ser incorporada a algún local de las facultades de la Universidad de Pinar del Río. Además el Everest, controla una gran cantidad de información innecesaria para el sistema que se pretende desarrollar. [4]

En resumen, actualmente no existe una herramienta que sea capaz de resolver de manera eficiente la problemática presente en esta investigación. Es por esta razón que se decidió llevar a cabo el desarrollo de un sistema capaz de almacenar y gestionar las piezas de las computadoras de todas las facultades de la Universidad de Pinar del Río y que permita además un mejor control de las mismas, facilitando el trabajo al personal que se encarga de esta tarea.

1.3 Modelo Conceptual del SACPIC

A continuación se realiza un análisis de los conceptos fundamentales que se tratan en el sistema a desarrollar, con el objetivo de una mejor comprensión del mismo, se presenta además un modelo conceptual como se puede observar en la figura 1.1, el cual muestra los conceptos empleados, contextualizados hacia el negocio.

Los conceptos son:

Facultad: Área de la Universidad que agrupa carreras con contenidos afines. Tiene atributos de interés como: *nombre*.

Local: Lugar donde son ubicadas las computadoras para ser utilizadas por el personal que trabaja en el mismo. Tiene atributos de interés como son: *nombre y descripción, así como el nombre de la facultad a la que pertenece.*

Responsable: Persona encargada del cuidado de un local, el mismo deberá dar cuenta de cada computadora ubicada en ese local, respondiendo por la integridad de la misma. Tiene atributos de interés como son: *nombre, apellido uno, apellido dos y carnet de identidad, así como el nombre del local del cuál es responsable.*

Computadora: La computadora es una máquina capaz de realizar una gran cantidad de cálculos aritméticos y procesos de control a una gran velocidad. Es una valiosa herramienta para realizar y simplificar muchas de las actividades de los seres humanos. En sí es un dispositivo electrónico capaz de interpretar y ejecutar los comandos programados para realizar en forma general las funciones de operaciones de entrada al ser receptora de información, operaciones de cálculo, lógica y almacenamiento, etc. Tiene atributos de interés como son: *tipo de equipo, descripción, marca, no sello garantía y no sello batalla, así como el nombre del local donde esta ubicada [5].*

Computadora Rota: Computadora que ha dejado de funcionar por alguna rotura. Tiene atributos de interés como son: *identificador, fecha de rotura y pieza rota, así como el identificador de la computadora que ha dejado de funcionar.*

Disco Duro: El disco duro es un sistema de grabación magnética digital, es donde en la mayoría de los casos reside el Sistema operativo de la computadora. En los discos duros se almacenan los datos del usuario. Tiene atributos de interés como son: *fabricante, capacidad y número de serie, así como el identificador de la computadora a la que pertenece [6].*

Unidad Central de Procesamiento (CPU): El CPU es una de las partes fundamentales del *Hardware*. Contiene los circuitos, los procesadores y las memorias que ejecutan las transferencias de información. La unidad central de proceso (CPU), es un conjunto de circuitos electrónicos digitales encargados de

recibir la información de los dispositivos de entrada/salida, procesarla y enviarla de nuevo a los dispositivos de entrada/salida, constituyéndose en la parte más importante del computador. Tiene atributos de interés como son: *tipo, fpo, velocidad y número de serie, así como el identificador de la computadora a la que pertenece* [7].

Lector CD: Esta unidad sirve para leer los discos compactos. La velocidad de esta unidad depende de dos factores: la tasa de transferencia de datos y el tiempo de acceso. Tiene atributos de interés como son: *fabricante, velocidad y número de serie, así como el identificador de la computadora a la que pertenece.* [6]

Lector Floppy: Un disquete o disco flexible (en inglés floppy disk o diskette) es un medio o soporte de almacenamiento de datos formado por una pieza circular de material magnético, fina y flexible (de ahí su denominación) encerrada en una cubierta de plástico cuadrada o rectangular. Los disquetes se leen y se escriben mediante un dispositivo llamado disquetera (o FDD, del inglés Floppy Disk Drive). En algunos casos es un disco menor que el CD. La disquetera es el dispositivo o unidad lectora/grabadora de disquetes, y ayuda a introducirlo para guardar la información. Tiene atributos de interés como son: *fabricante y número de serie, así como el identificador de la computadora a la que pertenece.* [6]

Fuente Interna: Además de convertir la tensión alterna del tomacorriente en continua, que es la que utilizan los circuitos de la computadora, la fuente también asegura que la tensión entregada al motherboard sea la correcta, es decir: no permite que la PC encienda si no están presentes todos los valores adecuados de tensión. Finalmente, distribuye la energía al resto de los componentes internos. Tiene atributos de interés como son: *fabricante, capacidad y número de serie, así como el identificador de la computadora a la que pertenece.* [6]

UPS: Proviene de la siglas de ("Uninterruptible Power Supply") ó respaldo de energía ininterrumpible. Sin embargo el nombre más utilizado es "No Brake" que significa sin interrupciones. Es un dispositivo que se conecta al enchufe doméstico, integra una circuitería especial que permite alimentar un juego de

baterías recargables internas mientras suministra energía eléctrica a la computadora. En caso de que se dé un corte de energía en el suministro de la red doméstica, las baterías automáticamente continúan alimentando a la computadora por un cierto periodo de tiempo, evitando pérdida de información. Tiene atributos de interés como son: *fabricante, modelo y número de serie, así como el identificador de la computadora a la que pertenece.* [8]

MotherBoard: Su función no es sólo de soporte físico (conectar mecánicamente placas, conectores, microprocesador y memorias), sino lograr que todos estos elementos, con sus características y señales diferentes, se puedan poner de acuerdo e intercambiar datos. Porque luego de la instalación mecánica de los componentes, para que el sistema funcione, es necesario que estén conectados físicamente entre sí y que alguien regule el tráfico de información y actúe de mediador entre las diferentes características que tienen las señales provenientes de cada elemento. Tiene atributos de interés como son: *fabricante, modelo y número de serie, así como el identificador de la computadora a la que pertenece.* [6]

Memoria RAM: Es la memoria de acceso aleatorio, en la que se guardan instrucciones y datos de los programas para que la CPU pueda acceder a ellos directamente a través del bus de datos externo de alta velocidad. Tiene atributos de interés como son: *tipo, capacidad, N/S o Lote o Chip, lote y número de serie, así como el identificador de la computadora a la que pertenece.* [6]

Teclado: Un *teclado de computadora* es un periférico, físico o virtual, utilizado para la introducción de órdenes y datos en una computadora. Tiene atributos de interés como son: *fabricante y número de serie, así como el identificador de la computadora a la que pertenece.* [6]

Tarjeta Gráfica: Una tarjeta gráfica, tarjeta de vídeo, placa de vídeo, tarjeta aceleradora de gráficos o adaptador de pantalla, es una tarjeta de expansión para una computadora u ordenador, encargada de procesar los datos provenientes de la CPU y transformarlos en información comprensible y representable en un

dispositivo de salida, como un monitor o televisor. Tiene atributos de interés como son: *fabricante y número de serie*, así como el *identificador de la computadora a la que pertenece*. [6]

Bocinas: Las bocinas se utilizan para escuchar los sonidos emitidos por el computador, tales como música, sonidos de errores, conferencias, etc. Tiene atributos de interés como son: *fabricante y número de serie*, así como el *identificador de la computadora a la que pertenece*. [6]

Impresora: Una impresora es un periférico de computadora que permite producir una copia permanente de textos o gráficos de documentos almacenados en formato electrónico, imprimiendo en papel de lustre los datos en medios físicos, normalmente en papel o transparencias, utilizando cartuchos de tinta o tecnología láser. Tiene atributos de interés como son: *fabricante, modelo y número de serie*, así como el *identificador de la computadora a la que pertenece*. [6]

Monitor: El monitor o pantalla de computadora, es un dispositivo de salida que, mediante una interfaz, muestra los resultados, o los gráficos del procesamiento de una computadora. Tiene atributos de interés como son: *fabricante, modelo y número de serie*, así como el *identificador de la computadora a la que pertenece*. [6]

En la siguiente figura se pueden observar las relaciones que tienen cada uno de los conceptos antes mencionados.

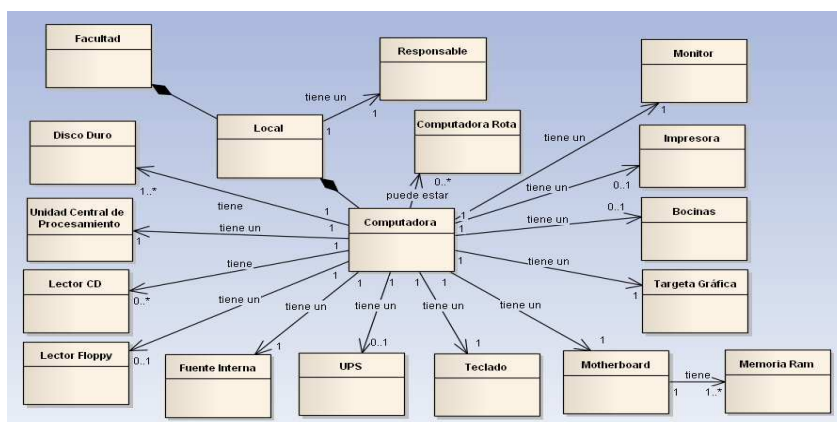


Figura 1.1. Modelo de Dominio del SACPIC.

1.4 Análisis de Factibilidad de SACPIC

Realizar la estimación de un proyecto es la tarea de calcular el tiempo y esfuerzo necesario para construir y desplegar una determinada situación.

La estimación mediante el análisis de Puntos de Casos de Uso es un método propuesto originalmente por Gustav Karner de Objectory AB, y posteriormente refinado por muchos otros autores. Se trata de un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de “pesos” a un cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de esos factores. [9]

Previamente a la implementación de SACPIC se realizó una valoración de los costos, para realizar esta valoración se utilizó el método expuesto anteriormente. A continuación se muestran los cálculos para la estimación del costo.

1.4.1. Cálculo de Casos de Uso y Actores sin ajustar.

Para realizar un correcto análisis tanto de la cantidad de actores que tiene el sistema a desarrollar y la complejidad de cada uno de ellos es preciso en primer lugar, conocer si el actor es una persona o sistema, y en segundo lugar la forma en que el actor interactúa con el mismo, en cuanto a la complejidad de los Casos de Uso, se establece teniendo en cuenta la cantidad de transacciones efectuadas en el mismo.

A partir de lo antes mencionado se establecen una serie de criterios como se puede observar en la en la tabla 1.1

TABLA 1.1: Criterios de los actores y Casos de Uso.

Criterios	Actores	Casos de Uso
Simple	Si el actor es otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación y tiene asignado como factor de peso 1.	El CU contiene de 1 a 3 transacciones. Tiene como factor de peso 5.
Medio	Si el actor es otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto y su factor de peso 2.	El CU tiene de 4 a 7 transacciones y su factor de peso es 10.
Complejo	Si el actor es una persona que interactúa con el sistema mediante una interfaz gráfica y su factor de peso asociado es 3.	El CU tiene más de 8 transacciones y su factor de peso es 15.

Para realizar el cálculo de Puntos de Casos de Uso sin ajustar es necesaria la siguiente ecuación [9].

$$\mathbf{PCU = FPA + FPCU, \quad (1.1)}$$

Donde,

PCU: Puntos de Casos de Uso sin ajustar.

FPA: Factor de Peso de los Actores sin ajustar.

FPCU: Factor de Peso de los Casos de Uso sin ajustar.

A continuación es mostrado los actores y CU determinados en SACPIC teniendo en cuenta los criterios antes mostrados. La Tabla 1.2 muestra los actores y su factor de peso asociado y en la tabla 1.3, los CU.

TABLA 1.2: Actores y sus factores de pesos.

Actores	Criterio	Factor de Peso
Administrador	Simple	3
Usuario	Simple	3
Administrador por Facultad	Simple	3
Total		9

TABLA 1.3: CU y sus factores de pesos.

Casos de Uso	Criterio	Factor de Peso
Gestionar datos de Computadora	Medio	10
Gestionar datos de Computadora Rota	Simple	5
Gestionar datos de Disco Duro	Simple	5
Gestionar datos de Unidad Central de Procesamiento.	Simple	5
Gestionar datos de Lector CD	Simple	5
Gestionar datos de Lector Floppy	Simple	5

Gestionar datos de Fuente Interna	Simple	5
Gestionar datos de UPS	Simple	5
Gestionar datos de Memoria Ram	Simple	5
Gestionar datos de Mother Board	Simple	5
Gestionar datos de Teclado	Simple	5
Gestionar datos de Tarjeta Grafica	Simple	5
Gestionar datos de Bocinas	Simple	5
Gestionar datos de Impresora	Simple	5
Gestionar datos de Monitor	Simple	5
Gestionar datos del Local	Simple	5
Gestionar datos de la Facultad	Simple	5
Gestionar datos del Responsable	Simple	5
Gestionar Usuario	Simple	5
Autenticación	Simple	5
Gestionar Contraseña de Usuario	Simple	5
Mostar Reportes	Medio	10
Mostar Ayuda	Simple	5
Total		125

Sustituyendo en la ecuación 1.1, el Factor de Peso de los Actores sin ajustar resulta:

FPA = 3 · 3 = 9, y el Factor de Peso de los Casos de Uso sin ajustar sería:

FPCU = 125

Luego

PCU = 9 + 125= 134.

1.4.2. Cálculo de Puntos de Casos de Uso ajustados

Una vez que se tienen los Puntos de Casos de Uso sin ajustar, se debe ajustar este valor mediante la siguiente ecuación [9].

$$\text{PCUA} = \text{PCU} \cdot \text{FCT} \cdot \text{FA}, \quad (1.2)$$

Donde,

PCUA: Puntos de Casos de Uso Ajustados

FCT: Factor de Complejidad Técnica

FA: Factor de Ambiente

1.4.2.1. Cálculo del Factor de Complejidad Técnica.

El Factor de Complejidad Técnica se estima mediante la cuantificación de factores que determinan la complejidad técnica del sistema [9]. 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, como se muestra en la Tabla 1.4.

TABLA 1.4. Factor de Complejidad Técnica.

Factor	Descripción	Peso	Valor	Peso · Valor	Comentario
FCT 01	Sistema Distribuido.	2	1	2.0	Aplicación Cliente servidor
FCT 02	Objetivos de	1	3	3.0	La velocidad es

	performance o tiempo de respuesta.				limitada por la entrada de datos por el operador.
FCT 03	Eficiencia del usuario final	1	5	5.0	Eficiente.
FCT 04	Procesamiento Interno Complejo.	1	1	1.0	No hay cálculos Complejos.
FCT 05	El código debe ser reutilizable	1	1	1.0	Deseable
FCT 06	Facilidad de Instalación.	0.5	4	2.0	Niveles medio de facilidad de instalación
FCT 07	Facilidad de Uso	0.5	3	1.5	Normal
FCT 08	Portabilidad	2	0	0.0	No se requiere
FCT 09	Facilidad de cambio	1	4	4.0	Fácil
FCT 10	Concurrencia	1	5	5.0	Multiusuario

FCT 11	Incluye objetivos especiales de Seguridad.	1	5	5.0	Restricciones de acceso por niveles de usuarios
FCT 12	Proporciona acceso directo a terceras partes	1	2	2.0	No acceso a terceras partes
FCT 13	Se requieren facilidades especiales de entrenamiento a usuarios.	1	5	5.0	Necesidad de entrenamiento para los usuarios
Total				36.5	

La ecuación para calcular el FCT es [9]:

$$\mathbf{FCT} = 0.6 + 0.01 \cdot \sum_i (\mathit{Peso}_i \cdot \mathit{Valor}_i) , \quad (1.3)$$

por lo que el FCT del presente proyecto sería:

$$\mathbf{FCT} = 0.6 + 0.01 \cdot 36.5$$

$$\mathbf{FCT} = 0.965$$

1.4.2.2 Cálculo del Factor de Ambiente.

En el caso del Factor de Ambiente, son las habilidades y el entrenamiento del grupo involucrado en el desarrollo de los factores contemplados. Dicho factor se calcula mediante la siguiente ecuación [9]:

$$FA = 1.4 - 0.03 \cdot \sum_i (Peso_i \cdot ValorAsignado_i) \quad (1.4)$$

En la Tabla 1.5 se muestra el significado y el peso de cada uno de estos factores. Sustituyendo el valor resultado de la sumatoria del peso por el valor de cada FA en la ecuación (1.4)

El resultado del FA sería:

$$FA = 1.4 - 0.03 \cdot 16.50$$

$$FA = 0.905.$$

TABLA 1.5. Factor Ambiental.

Factor	Descripción	Peso	Valor	Peso · Valor	Comentario
FA01	Familiaridad con el modelo de proyecto utilizado.	1.5	3.0	4.5	No todo el grupo se encuentra familiarizado con este modelo.
FA02	Experiencia en la aplicación.	0.5	2.0	1.0	La mayoría no ha trabajado en este tipo de aplicación
FA03	Experiencia en	1.0	3.0	3.0	Algunos

	orientación a Objetos.				integrantes del grupo programan orientado a objetos
FA04	Capacidad del analista líder.	0.5	4.0	2.0	Se encuentra bien preparado
FA05	Motivación	1.0	5.0	5.0	Todo el equipo está muy motivado
FA06	Estabilidad de los requerimientos	2.0	3.0	6.0	Pueden ocurrir cambios en los requerimientos
FA07	Personal a tiempo compartido	-1.0	1.0	-1.0	Full-Time
FA08	Dificultad del lenguaje de programación.	-1.0	4.0	-4.0	Se utiliza java
Total	16.50				

Sustituyendo en la ecuación (1.2) los resultados de los cálculos realizados anteriormente quedaría:

$$\text{PCUA} = 134 \cdot 0.965 \cdot 0.905.$$

$$\text{PCUA} = 117.02$$

1.4.3. De los Puntos de Casos de Uso a la Estimación del Esfuerzo

El esfuerzo en horas-hombre viene dado por la siguiente ecuación [9]:

$$\text{E} = \text{PCUA} \cdot \text{FC}, \quad (1.5)$$

Donde,

E: Esfuerzo estimado en horas-hombre.

FC: Factor de conversión.

Siguiendo los criterios establecidos para determinar el factor de conversión, se determinó en este caso un valor de 20 horas-hombre/Punto de Casos de Uso.

Sustituyendo estos datos en la ecuación (1.5)

$$\text{E}_{\text{(Programación)}} = 117.02 \cdot 20 = 2340 \text{ Horas-Hombre.}$$

Para poder obtener una estimación 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 del software. Para ello se puede tener en cuenta el criterio que propone la distribución del esfuerzo entre las diferentes actividades de un proyecto, según la aproximación mostrada en la Tabla 1.6 [9].

Tabla 1.6. Distribución del esfuerzo por actividades.

Actividad	Porcentaje	Horas-Hombre
Análisis	10%	585
Diseño	20%	1170
Implementación	40%	2340
Pruebas	15%	877.5
Sobrecarga(otras actividades)	15%	877.5
Total	100%	5850

$$E_{(Total)} = 5850 \text{ horas- hombre.}$$

1.4.4. Tiempo de desarrollo.

A partir del esfuerzo total calculado anteriormente se pueden realizar las estimaciones del tiempo de desarrollo aproximado del proyecto y el costo total del mismo.

El tiempo de duración se obtiene dividiendo el esfuerzo total entre la cantidad de personas con que se cuenta [9]. La ecuación es la siguiente:

$$TDES_{(Total)} = E_{(Total)} / CH, \quad (1.6)$$

donde:

TDES _(Total): Tiempo de desarrollo

CH: Cantidad de hombres que desarrollan el proyecto.

Sustituyendo en la ecuación (1.6) quedaría:

$$\text{TDES}_{(Total)} = 5850 \text{ horas-hombre} / 2 \text{ hombre} = \mathbf{2925 \text{ horas.}}$$

1.4.5. Costo Total del Proyecto

Una vez calculado el tiempo de desarrollo del proyecto y conociendo la cantidad de desarrolladores del mismo se puede realizar una estimación de su costo total haciendo uso de la siguiente ecuación [9]:

donde:

$$C_{(Total)} = \text{TDES} \cdot \text{CHH}, \quad (1.7)$$

CHH : Costo por hombre hora

y se calcula como:

$$\text{CHH} = \text{K} \cdot \text{THP}, \quad (1.8)$$

donde:

K: Coeficiente que tiene en cuenta los costos indirectos (1.5)

THP: Tarifa Horaria Promedio.

El salario promedio de las personas que trabajan en el proyecto dividido entre 160 horas. Sustituyendo en la ecuación (1.8)

$$\text{CHH} = 1.5 \cdot 325 / 160 = 3.0469.$$

Luego en la ecuación (1.7)

$$\mathbf{C (Total) = 2925 \cdot 3.0469 = \$ 8912.18}$$

El costo total en el que implica la implementación de SACPIC es de **\$ 8912.1**

CAPÍTULO 2. DISEÑO DEL SACPIC.

El capítulo 2 se encuentra estructurado de la siguiente forma: En el epígrafe 1 se describen las herramientas utilizadas para el diseño del SACPIC, brindando sus características principales.

En el epígrafe 2 muestra las entidades y relaciones de la Base de Datos de SACPIC así como el modelo de datos del mismo.

En el tercer epígrafe se trata lo relacionado con el diseño de la Interfaz de Usuario del SACPIC definiéndose: Requerimientos Funcionales y No Funcionales del sistema, Actores, Casos de Usos, Diagramas de Casos de Uso, se describe un Caso de Uso textualmente sobre la base de su prototipo de Interfaz-Usuario.

2.1 Herramientas utilizadas para el diseño del SACPIC.

La sociedad actual está en continuo cambio y evolución, las nuevas tecnologías avanzan a pasos agigantados, el alto nivel de competitividad que existe hoy en la industria del software hace que las empresas dedicadas al desarrollo del mismo utilicen una disciplina mediante la cual controlen que los ingenieros usen de forma consistente los nuevos métodos. Obtener buena calidad deberá ser la meta y el camino de cada empresa, es por esto que una de sus razones principales será la de mejorar el proceso de desarrollo de software, por lo tanto, se requiere utilizar los métodos y procedimientos de la Ingeniería y Gestión de Software.

Durante el desarrollo de este sistema se ha utilizado la metodología RUP, UML como el lenguaje unificado de modelado y como herramienta CASE el Enterprise Architect.

2.1.2 Proceso Unificado de Desarrollo de Software (RUP)

El Proceso Unificado de Desarrollo Software o simplemente Proceso Unificado es un marco de desarrollo de software que se caracteriza por estar dirigido por casos de uso, centrado en la arquitectura y por ser iterativo e incremental. El refinamiento más conocido y documentado del Proceso Unificado es el Proceso Unificado de Rational o simplemente RUP. El objetivo que se persigue con esta metodología es producir software de alta calidad, es decir, que cumpla con los requerimientos de los usuarios dentro de una planificación y presupuesto establecidos [10].

CARACTERISTICAS PRINCIPALES

- **Iterativo e Incremental:** El Proceso Unificado es un marco de desarrollo iterativo e incremental compuesto de cuatro fases denominadas Inicio, Elaboración, Construcción y Transición. Cada una de estas fases es a su vez dividida en una serie de iteraciones (la de inicio sólo consta de varias iteraciones en proyectos grandes). Estas iteraciones ofrecen como resultado un incremento del producto desarrollado que añade o mejora las funcionalidades del sistema en desarrollo. Cada una de estas iteraciones se divide a su vez en una serie de disciplinas que recuerdan a las definidas en el ciclo de vida clásico o en cascada: Análisis de requisitos, Diseño, Implementación y Prueba. Aunque todas las iteraciones suelen incluir trabajo en casi todas las disciplinas, el grado de esfuerzo dentro de cada una de ellas varía a lo largo del proyecto [10].
- **Dirigido por los casos de uso** En el Proceso Unificado los casos de uso se utilizan para capturar los requisitos funcionales y para definir los contenidos de las iteraciones. La idea es que cada iteración tome un conjunto de casos de uso o escenarios y desarrolle todo el camino a través de las distintas disciplinas: diseño, implementación, prueba, etc. el proceso dirigido por casos de uso es el rup. Nota: en UP se está Dirigido por

requisitos y riesgos de acuerdo con el Libro UML 2 de ARLOW, Jim que menciona el tema [10].

- **Centrado en la arquitectura** El Proceso Unificado asume que no existe un modelo único que cubra todos los aspectos del sistema. Por dicho motivo existen múltiples modelos y vistas que definen la arquitectura de software de un sistema. La analogía con la construcción es clara, cuando construyes un edificio existen diversos planos que incluyen los distintos servicios del mismo: electricidad, fontanería, etc [10].

2.1.3 Lenguaje de Modelado Unificado (UML)

UML es el Lenguaje de Modelado Unificado (Unified Modeling Language) es la sucesión de una serie de métodos de análisis y diseño orientadas a objetos que aparecen a fines de los 80 y principios de los 90. UML es llamado un lenguaje de modelado, no un método. El mismo incrementa la capacidad de lo que se puede hacer con otros métodos de análisis y diseño orientados a objetos [11].

El lenguaje de modelado es la notación (principalmente gráfica) que usan los métodos para expresar un diseño. El proceso indica los pasos que se deben seguir para llegar a un diseño [11].

La estandarización de un lenguaje de modelado es invaluable, ya que es la parte principal del proceso de comunicación que requieren todos los agentes involucrados en un proyecto informático. Si se quiere discutir un diseño con alguien más, ambos deben conocer el lenguaje de modelado y no así el proceso que se siguió para obtenerlo. El lenguaje para modelamiento unificado (UML), es un lenguaje para la especificación, visualización, construcción y documentación de los artefactos de un proceso de sistema intensivo. [11].

CARACTERISTICAS PRINCIPALES

- Proporcionar una notación y semánticas suficientes para poder alcanzar una gran cantidad de aspectos del modelado contemporáneo de una forma directa y económica [11].
- Proporcionar mecanismos de extensión de forma que aproximaciones de modelado futuras podrían desarrollarse encima del UML.
- Permitir el intercambio del modelo entre una gran variedad de herramientas.
- Proporcionar a los usuarios un lenguaje de modelado visual expresivo y utilizable para el desarrollo e intercambio de modelos significativos.
- Ser independiente del proceso de desarrollo y de los lenguajes de programación.
- Proporcionar una base formal para entender el lenguaje de modelado.

2.1.4 Enterprise Architect (EA).

Enterprise Architect es una herramienta comprensible de diseño y análisis UML, cubriendo el desarrollo de software desde el paso de los requerimientos a través de las etapas del análisis, modelos de diseño, pruebas y mantenimiento. EA es una herramienta multi-usuario, basada en Windows, diseñada para ayudar a construir software robusto y fácil de mantener. Ofrece salida de documentación flexible y de alta calidad. EA le ayuda a administrar la complejidad con herramientas para rastrear las dependencias, soporte para modelos muy grandes, interfaz intuitiva y de alto rendimiento con vista de proyecto como un "explorador" [12].

EA soporta generación e ingeniería inversa de código fuente para muchos lenguajes populares, incluyendo C++, C#, Java, Delphi, VB.Net, Visual Basic y PHP. Con un editor de código fuente con "resaltador de sintaxis" incorporado, EA le permite navegar y explorar su modelo de código fuente en el mismo ambiente. EA le ayuda a visualizar sus aplicaciones soportando ingeniería inversa de un

amplio rango de lenguajes de desarrollo de software y esquemas de repositorios de base de datos [12].

EA soporta transformaciones de Arquitectura avanzada dirigida por Modelos (MDA) usando plantillas de transformaciones de desarrollo y fáciles de usar. Con transformaciones incorporadas para DDL, C#, Java, EJB y XSD, Ud. puede rápidamente desarrollar soluciones complejas desde los simples "modelos independientes de plataforma" (MIP) que son el objetivo en "modelos específicos de plataforma" (MEP). Un MIP se puede usar para generar y sincronizar múltiples MIP's - proveyendo un aumento de productividad significativo [12].

2.2 Modelo de Datos del SACPIC.

Un modelo de datos es un conjunto de conceptos que sirven para describir la estructura de una Base de Datos. Para la obtención del modelo de datos del sistema se utilizaron conceptos de Base de Datos como entidades y relaciones, llaves primarias y extranjeras, etc.

Teniendo en cuenta los mismos, la Base de Datos del SACPIC quedó definida de la siguiente forma:

Entidades: Se definen como Entidades: Usuario, Local, Responsable, Facultad, Computadora, Computadora Rota, Disco Duro, Unidad Central de Procesamiento, Lector CD, Lector Floppy, Fuente Interna, UPS, Memoria Ram, Mother Board, Teclado, Tarjeta Gráfica, Bocinas, Impresora y Monitor.

Relaciones: En SACPIC quedaron definidas 6 relaciones de uno a muchos, cumpliendo con la regla que expresa que la llave de la tabla que tiene asociado "el uno" se pasa como llave extranjera a la que tiene asociado "el mucho" , y 12 de uno a uno .

Para una mejor comprensión del modelo de datos del SACPIC en la figura 2.1 se pueden observar las tablas con sus atributos y las relaciones entre ellas.

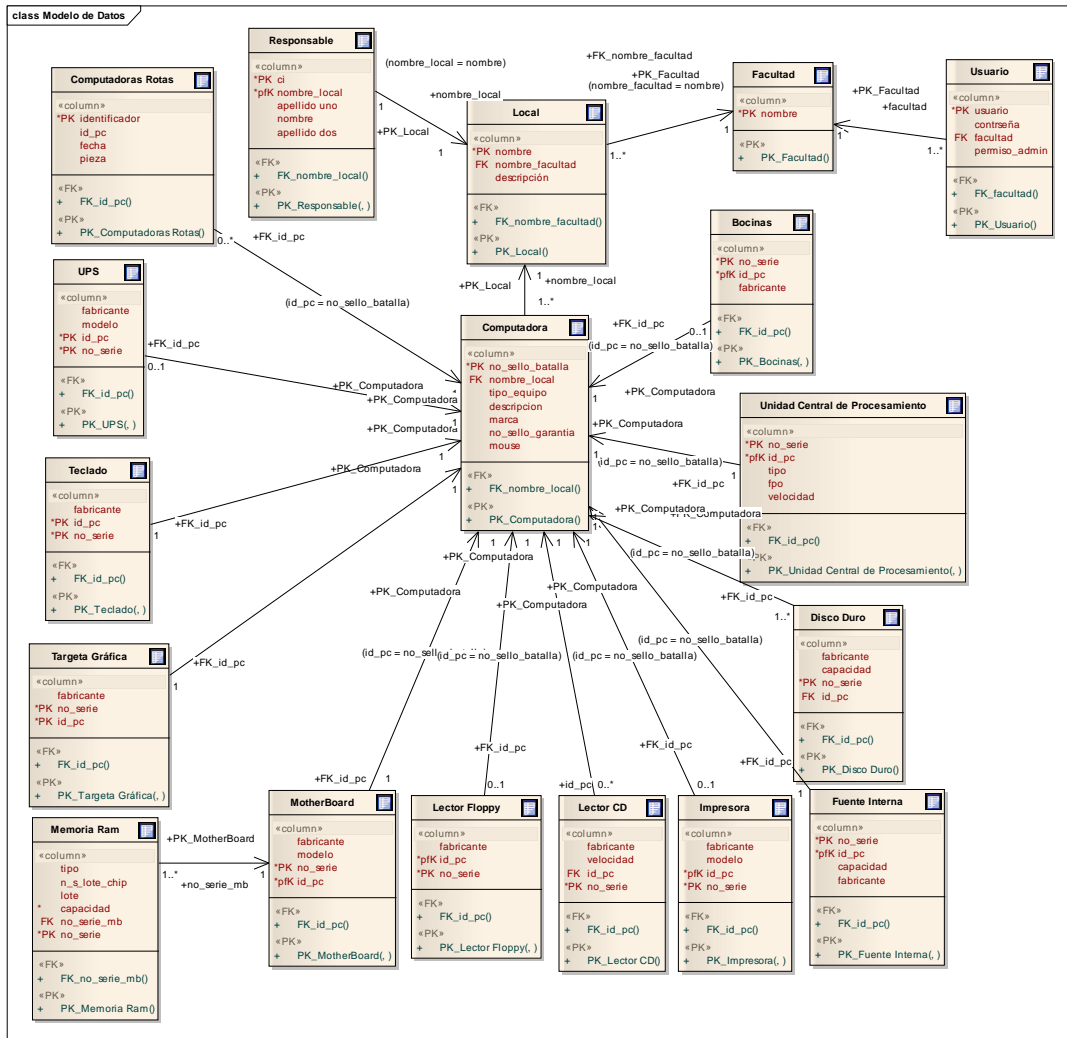


Figura 2.1: Modelo de Datos de SACPIC.

2.3 Diseño de la Interfaz de Usuario del SACPIC.

El objetivo principal de todo desarrollador de software es lograr un producto de buena calidad y que satisfaga completamente al cliente para el cual esta siendo desarrollado. Para obtener estos resultados se hace necesaria la recopilación de los requerimientos funcionales como no funcionales del sistema.

La Interfaz de Usuario debe cumplir en su totalidad con estas funcionalidades ya que las mismas son aspectos importantes a tener en cuenta para satisfacer al cliente.

2.3.1 Requerimientos Funcionales.

Los requerimientos funcionales propuesto por el cliente en este sistema son:

RF1. Gestionar datos de Computadora

RF1.1 Insertar datos de Computadora

RF1.2 Modificar datos de Computadora

RF1.3 Eliminar datos de Computadora

RF1.4 Buscar datos de Computadora

RF1.5 Mostrar datos de Computadora

RF2. Gestionar datos de Computadora Rotas

RF2.1 Insertar datos de Computadora Rotas

RF2.2 Modificar datos de Computadora Rotas

RF2.3 Eliminar datos de Computadora Rotas

RF2.4 Buscar datos de Computadora Rotas

RF2.5 Mostrar datos de Computadora Rotas

RF3. Gestionar datos del Disco Duro

RF3.1 Insertar datos del Disco Duro

RF3.2 Modificar datos del Disco Duro

RF3.3 Eliminar datos del Disco Duro

RF3.4 Buscar datos del Disco Duro

RF3.5 Mostrar datos del Disco Duro

RF4. Gestionar datos de la Unidad Central de Procesamiento

RF4.1 Insertar datos de la Unidad Central de Procesamiento

RF4.2 Modificar datos de la Unidad Central de Procesamiento

RF4.3 Eliminar datos de la Unidad Central de Procesamiento

RF4.4 Buscar datos de la Unidad Central de Procesamiento

RF4.5 Mostrar datos de la Unidad Central de Procesamiento

RF5. Gestionar datos del Lector CD

RF5.1 Insertar datos del Lector CD

RF5.2 Modificar datos del Lector CD

RF5.3 Eliminar datos del Lector CD

RF5.4 Buscar datos del Lector CD

RF5.5 Mostrar datos del Lector CD

RF6. Gestionar datos del Lector Floppy

RF6.1 Insertar datos del Lector Floppy

RF6.2 Modificar datos del Lector Floppy

RF6.3 Eliminar datos del Lector Floppy

RF6.4 Buscar datos del Lector Floppy

RF6.5 Mostrar datos del Lector Floppy

RF7. Gestionar datos de la Fuente Interna

RF7.1 Insertar datos de la Fuente Interna

RF7.2 Modificar datos de la Fuente Interna

RF7.3 Eliminar datos de la Fuente Interna

RF7.4 Buscar datos de la Fuente Interna

RF7.5 Mostrar datos de la Fuente Interna

RF8. Gestionar datos de la UPS

RF8.1 Insertar datos de la UPS

RF8.2 Modificar datos de la UPS

RF8.3 Eliminar datos de la UPS

RF8.4 Buscar datos de la UPS

RF8.5 Mostrar datos de la UPS

RF9. Gestionar datos de la Memoria RAM

RF9.1 Insertar datos de la Memoria RAM

RF9.2 Modificar datos de la Memoria RAM

RF9.3 Eliminar datos de la Memoria RAM

RF9.4 Buscar datos de la Memoria RAM

RF9.5 Mostrar datos de la Memoria RAM

RF10. Gestionar datos de la Mother Board

RF10.1 Insertar datos de la Mother Board

RF10.2 Modificar datos de la Mother Board

RF10.3 Eliminar datos de la Mother Board

RF10.4 Buscar datos de la Mother Board

RF10.5 Mostrar datos de la Mother Board

RF11. Gestionar datos del Teclado

RF11.1 Insertar datos del Teclado

RF11.2 Modificar datos del Teclado

RF11.3 Eliminar datos del Teclado

RF11.4 Buscar datos del Teclado

RF11.5 Mostrar datos del Teclado

RF12. Gestionar datos de la Tarjeta Gráfica

RF12.1 Insertar datos de la Tarjeta Gráfica

RF12.2 Modificar datos de la Tarjeta Gráfica

RF12.3 Eliminar datos de la Tarjeta Gráfica

RF12.4 Buscar datos de la Tarjeta Gráfica

RF12.5 Mostrar datos de la Tarjeta Gráfica

RF13. Gestionar datos de la Bocina

RF13.1 Insertar datos de la Bocina

RF13.2 Modificar datos de la Bocina

RF13.3 Eliminar datos de la Bocina

RF13.4 Buscar datos de la Bocina

RF13.5 Mostrar datos de la Bocina

RF14. Gestionar datos de la Impresora

RF14.1 Insertar datos de la Impresora

RF14.2 Modificar datos de la Impresora

RF14.3 Eliminar datos de la Impresora

RF14.4 Buscar datos de la Impresora

RF14.5 Mostrar datos de la Impresora

RF15. Gestionar datos del Monitor

RF15.1 Insertar datos del Monitor

RF15.2 Modificar datos del Monitor

RF15.3 Eliminar datos del Monitor

RF15.4 Buscar datos del Monitor

RF15.5 Mostrar datos del Monitor

RF16. Gestionar datos del Local

RF16.1 Insertar datos del Local

RF16.2 Modificar datos del Local

RF16.3 Eliminar datos del Local

RF16.4 Buscar datos del Local

RF16.5 Mostrar datos del Local

RF17. Gestionar datos de la Facultad

RF17.1 Insertar datos de la Facultad

RF17.2 Modificar datos de la Facultad

RF17.3 Eliminar datos de la Facultad

RF17.4 Buscar datos de la Facultad

RF17.5 Mostrar datos de la Facultad

RF18. Gestionar datos del Responsable

RF18.1 Insertar datos del Responsable

RF18.2 Modificar datos del Responsable

RF18.3 Eliminar datos del Responsable

RF18.4 Buscar datos del Responsable

RF18.5 Mostrar datos del Responsable

RF19. Gestionar Usuario

RF19.1 Insertar Usuario

RF19.2 Eliminar Usuario

RF19.3 Buscar Usuario

RF19.4 Mostrar Usuario

RF20. Autenticación

RF20.1 Gestionar Autenticación

RF20.1 Actualizar Fichero de Configuración

RF21. Gestionar Contraseña de Usuario

RF21.1 Cambiar Contraseña de Usuario

RF21.2 Cambiar de Nivel de Acceso

RF22. Mostar Reportes

RF22.1 Mostrar respotes

RF22.2 Guardar respotes

RF22.3 Imprimir respotes

RF23. Mostar Ayuda

RF23.1 Consultar Ayuda

2.3.2 Requerimientos No Funcionales.

Los requerimientos no funcionales, especifican propiedades y características que debe tener el sistema.

En el SACPIC se determinaron los siguientes requerimientos no funcionales:

RNF1. Hardware: Para la implantación del sistema debe contarse con una computadora con suficiente potencia y capacidad para procesar y almacenar información, velocidad del microprocesador superior a 850 MHz, que tenga una capacidad de disco duro de 20 GB en lo adelante y como mínimo, 512 MB de memoria RAM para garantizar un buen desempeño.

RNF2. Software: Para la implantación del sistema debe contarse con un servidor de postgresql, ya sea en la propia máquina o en cualquier otra computadora conectada a la red. Además debe tener instalado un interprete de java, osea la máquina virtual de java.

RNF3. Confidencialidad: Cada usuario accederá a la aplicación en dependencia de sus privilegios o niveles de acceso.

RNF4. Interfaz Externa: Debe ser una interfaz sencilla, legible, fácil de manipular, amigable y agradable para el usuario; debe ser diseñada con colores agradables a la vista de los usuarios, así como textos legibles.

RNF5. Usabilidad: Debe tener una consistencia en sus interfaces, lo que facilitará su uso y comprensión. El usuario debe sentirse guiado y orientado en su uso.

2.3.3 Actores del sistema.

Un Actor es aquel usuario que interactúa con el sistema, con el propósito de lograr un beneficio.

En SACPIC se definieron tres actores: El Administrador, el Usuario y el Administrador por Facultad, los mismo son beneficiados con las funcionalidades del sistema, cada unos de ellos tiene nivel de acceso dentro de la aplicación, desempeñando su rol en esta, en la tabla 2.1 se puede observar los actores y sus roles dentro del sistema a desarrollar.

TABLA 2.1 Actores y sus Roles dentro del SACPIC.

Actor	Rol
Administrador	Es el encargado de la inserción, modificación y eliminación de la información relacionada con los datos de las computadoras, así como de las piezas pertenecientes a cada una de ellas, entre otros. Pudiendo gestionar la información almacenada en la Base de Datos, y teniendo permisos para acceder a todos los servicios que

	brinda el sistema.
Usuario	Solo podrá consultar la información almacenada en la Base de Datos que es mostrada en los reportes del sistema. Además podrá acceder a los servicios de la ayuda y a modificar su contraseña de autenticación.
Administrador por Facultad	Tiene permiso para crear los usuarios de sus facultades, así como consultar la información almacenada en la base de datos que corresponda con su facultad, cambiar su contraseña de acceso al sistema, y consultar la ayuda.

2.3.4 Diagrama de Casos de Uso.

Los CU son fragmentos de funcionalidad que el sistema ofrece para aportar un resultado de valor para sus actores, cuyo objetivo final es satisfacer las necesidades del cliente. El diagrama de casos de uso representa la forma en como un Cliente (Actor) opera con el sistema en desarrollo, además de la forma, tipo y orden en como los elementos interactúan.

En SACPIC se definieron 23 CU garantizando así un análisis del sistema. Mediante la herramienta CASE utilizada se modeló el diagrama de Casos de Uso del sistema como se observa en la figura 2.2. En el mismo es mostrada la relación de cada CU representando el flujo de la información entre ellos.

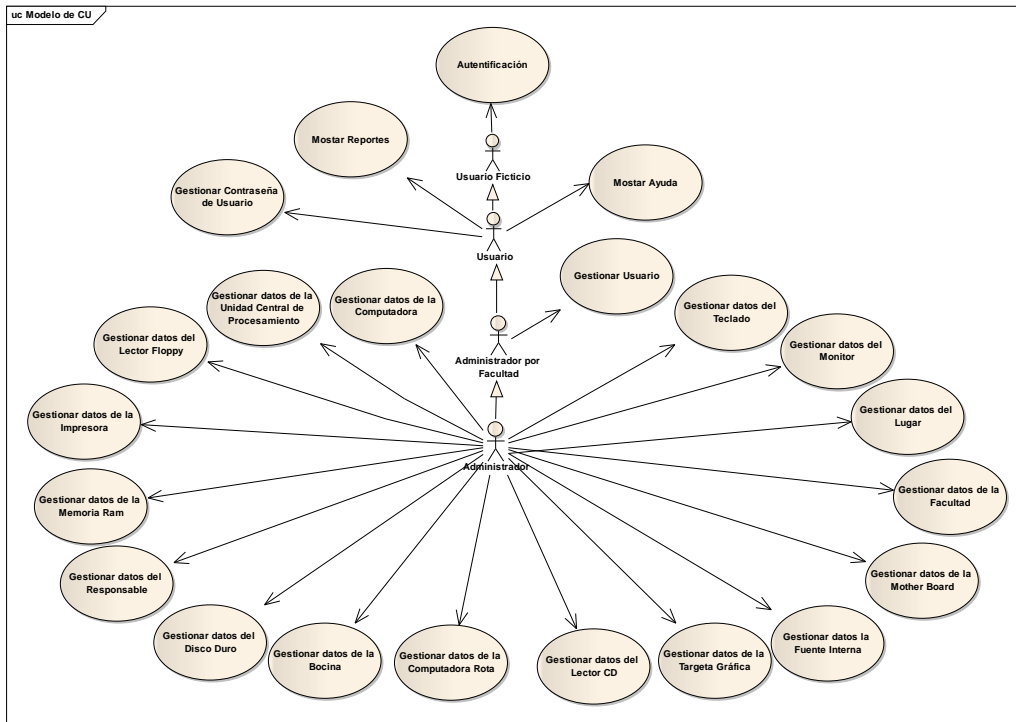


Figura 2.2: Diagrama de Casos de Usos con los Actores del Sistema.

2.3.4.1 Descripción.

Toda sistema a desarrollar contiene CU que por su contenido tiene mas peso e importancia dentro del sistema, en SACPIC uno de los CU más significativos, ya que de él7 dependen otros es “Gestionar datos de la Computadora” en la tabla 2.2 es descrito el mismo y para más información en el Anexo 2 se muestra el diagrama de clases del análisis de dicho caso de uso, además de sus diagramas de secuencia y de clases del diseño en el Anexo 3 y 4 respectivamente.

Tabla 2.2: Descripción del CU “Gestionar datos de Computadora”.

Caso de uso	Gestionar datos de Computadora
Actor	Administrador
Resumen	El Caso de Uso se inicia cuando el Administrador necesita insertar, modificar o eliminar una Computadora en el sistema, una vez realizada su gestión concluye el Caso de Uso.
Referencia	RF1
Precondiciones	El usuario ingresó al sistema como Administrador.
Poscondiciones	Queda actualizado el listado de las computadoras.
Pantallas asociadas	

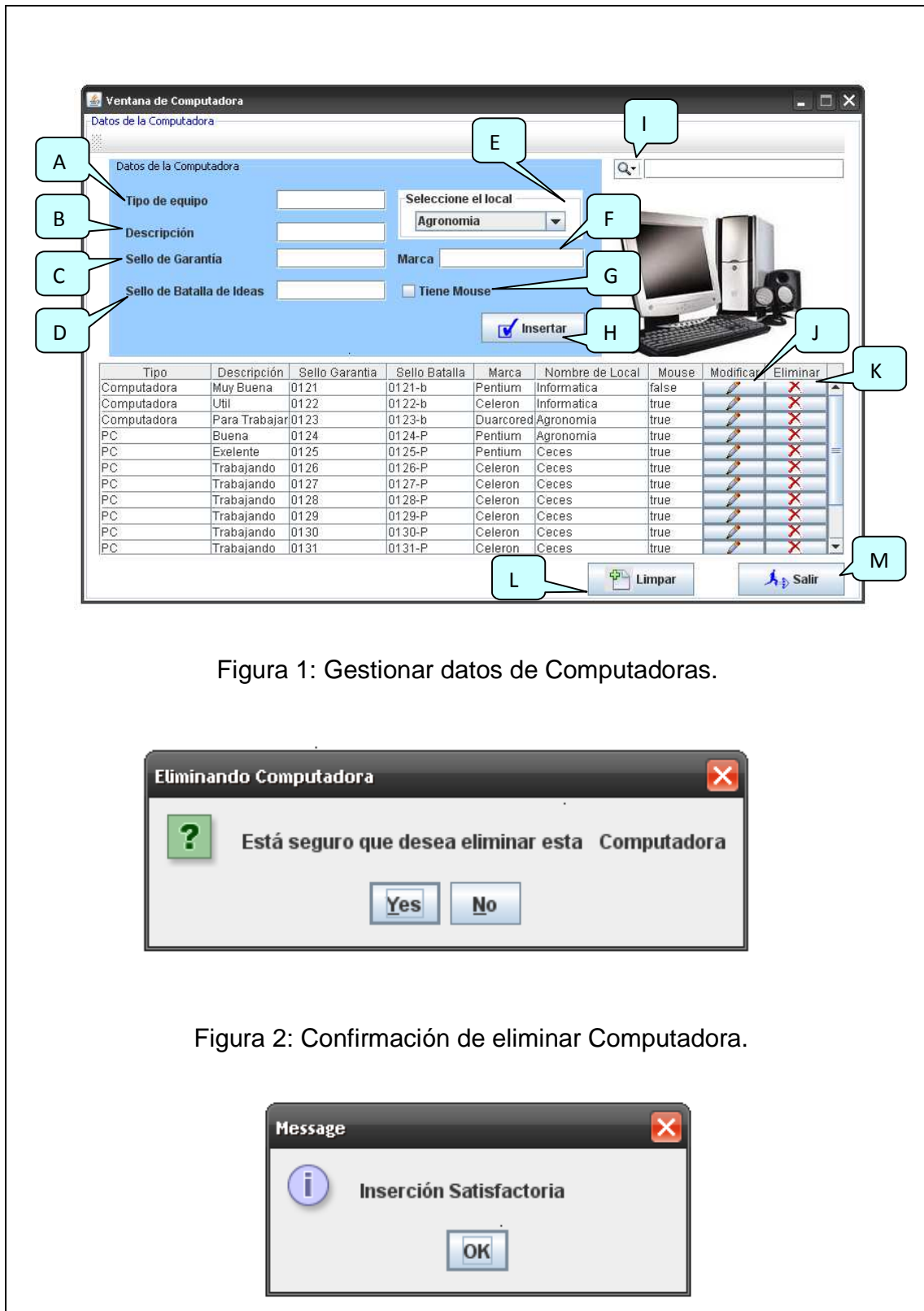


Figura 1: Gestionar datos de Computadoras.

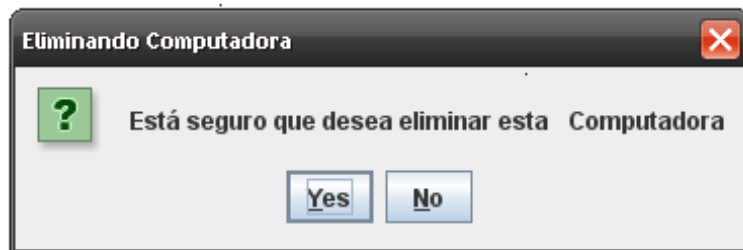


Figura 2: Confirmación de eliminar Computadora.



Figura 3: Confirmación de Inserción Correcta.



Figura 4: Confirmación de Modificación Correcta.

Pantallas Asociadas para cursos alternos

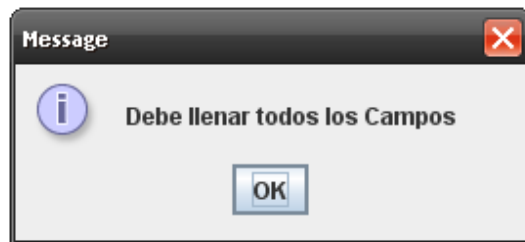


Figura 5: Confirmación de Campos Vacíos.



Figura 6: Confirmación de Existencia de Computadora en la Base de Datos.

Sección Principal	
Acciones de los Actores	Respuestas del Sistema
<p>1- El Administrador selecciona la opción (H) en la Pantalla 1 si lo que desea es insertar, la opción (J) en la Pantalla 1 si lo que desea es modificar o la opción de (K) en la Pantalla 1 si lo que desea es eliminar.</p>	<p>2- Si fue la opción insertar muestra la Pantalla 1, ver sección insertar, si fue modificar muestra la Pantalla 1, ver sección modificar, si fue eliminar muestra la Pantalla 2, ver sección eliminar.</p>
Sección Insertar	
<p>1-El Administrador del sistema desea insertar una nueva computadora por lo que introduce los datos de la misma, desde el elemento A hasta el G de la Pantalla 1, luego presiona el botón Insertar opción H de esta misma pantalla.</p>	<p>2-El sistema verifica la información introducida, de ser correcta la registra, y actualiza el lista mostrada en la</p>

	parte inferior de la pantalla 1 y muestra un mensaje de “Inserción correcta” como se observa en pantalla 3.
Sección Modificar	
<p>1. El Administrador del sistema desea modificar los datos de una Computadora existente por lo que presiona la opción [J] de la pantalla 1.</p> <p>3. El Administrador modifica los datos desde el elemento A hasta el G y presiona la opción [H] .</p>	<p>2. El sistema muestra en la parte superior de la pantalla 1 los datos correspondientes a la fila de la tabla que el usuario ha seleccionado para ser modificado y cambia el texto en la opción [H] de “Insertar” a “Modificar”.</p> <p>4. El sistema verifica la información registrada, de ser correcta la modifica en la base de datos, actualiza el lista mostrada en la parte inferior de la pantalla 1 y envía un mensaje de “Modificación correcta” como se observa en pantalla 4.</p>
Sección Eliminar	
1. El administrador del sistema desea eliminar una computadora por lo que presiona la opción [K] de la	

<p>pantalla 1.</p> <p>3. El administrador selecciona la opción deseada. Yes o No</p>	<p>2. El sistema envía una confirmación de si está seguro que desea eliminar la computadora correspondiente a la fila de la tabla que ha sido seleccionada por el administrador, como se muestra en la pantalla 2.</p> <p>4. El sistema elimina o no la Computadora según la acción deseada por el administrador y es actualizada la tabla de la parte inferior de la pantalla 1.</p>
<p>Cursos Alternos</p>	
<p>Sección: Insertar, Modificar y Eliminar.</p>	
<p>1. Si el administrador ha dejado algún campo sin introducir o los caracteres escritos son no válidos, el sistema mostrará un mensaje como se puede observar en la pantalla 5.</p> <p>2. Si el parámetro correspondiente a la opción [D] de la pantalla 1 es igual al de alguna computadora introducida se mostrará un mensaje como se observar en la pantalla 6.</p>	

CAPÍTULO 3. IMPLEMENTACIÓN DEL SACPIC.

En este capítulo se realiza un análisis de diferentes sistemas gestores de Base de Datos y se brindan las principales características del sistema gestor seleccionado para la implementación de la base de datos de SACPIC.

En el epígrafe 2 se caracterizan algunos lenguajes de programación, exponiendo posteriormente las razones de la selección de java como lenguaje, se describe además a NetBeans como entorno de desarrollo para el lenguaje seleccionado.

En el epígrafe 3 se muestra el modelo de componente resultante del SACPIC.

En el cuarto epígrafe se aborda lo referido a la implementación de la ayuda, se realiza una breve caracterización de la biblioteca de Java utilizada y se muestra un ejemplo del resultado de dicha implementación.

En el quinto y último epígrafe se describe cómo fue implementada la seguridad del SACPIC.

3.1. Implementación de la Base de Datos del SACPIC.

Los sistemas de gestión de bases de datos (SGBD) (en inglés *database management system*, abreviado *DBMS*) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. El propósito general de los sistemas de gestión de bases de datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante para una organización [13].

Algunas de las ventajas que traen los SGBD son:

- Proveen facilidades para la manipulación de grandes volúmenes de datos.
Entre éstas:
 - Simplifican la programación de equipos de consistencia.

- Manejando las políticas de respaldo adecuadas, garantizan que los cambios de la base serán siempre consistentes sin importar si hay errores correctamente, etc.
- Organizan los datos con un impacto mínimo en el código de los programas.
- Disminuyen drásticamente los tiempos de desarrollo y aumentan la calidad del sistema desarrollado si son bien explotados por los desarrolladores. [13]
- Usualmente, proveen interfaces y lenguajes de consulta que simplifican la recuperación de los datos.

Existen diversos SGBD, entre los que podemos mencionar, Microsoft Access, MySQL, PostgreSQL, etc. A continuación es mostrada una breve caracterización de cada uno de ellos, así como las razones por las que fue seleccionado PostgreSQL como SGBD para la implementación de la base de datos del SACPIC.

3.1.1. Microsoft Access

Microsoft Access es un sistema de gestión de bases de datos relacionales para los sistemas operativos Microsoft Windows, desarrollado por Microsoft y orientado a ser usado en un entorno personal o en pequeñas organizaciones. Es un componente de la suite ofimática Microsoft Office. Permite crear ficheros de bases de datos relacionales que pueden ser fácilmente gestionadas por una interfaz gráfica simple. Además, estas bases de datos pueden ser consultadas por otros programas. Este programa permite manipular los datos en forma de tablas (formadas por filas y columnas), crear relaciones entre tablas, consultas, formularios para introducir datos e informes para presentar la información. [14]

3.1.2. MySQL

MySQL es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU. Su diseño multihilo le permite soportar una gran carga de forma

muy eficiente. Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración. MySQL surgió como una necesidad de un grupo de personas sobre un gestor de bases de datos rápido, por lo que sus desarrolladores fueron implementando únicamente lo que precisaban, intentando hacerlo funcionar de forma óptima. Es por ello que, aunque MySQL se incluye en el grupo de sistemas de bases de datos relacionales, carece de algunas de sus principales características [15]:

1. Subconsultas: tal vez ésta sea una de las características que más se echan en falta, aunque gran parte de las veces que se necesitan, es posible reescribirlas de manera que no sean necesarias.
2. SELECT INTO TABLE: Esta característica propia de Oracle, todavía no está implementada.
3. Triggers y Procedures: Se tiene pensado incluir el uso de procedures almacenados en la base de datos, pero no el de triggers, ya que los triggers reducen de forma significativa el rendimiento de la base de datos, incluso en aquellas consultas que no los activan.
4. Transacciones: a partir de las últimas versiones ya hay soporte para transacciones, aunque no por defecto (se ha de activar un modo especial).
5. Integridad referencial: aunque sí que admite la declaración de claves ajenas en la creación tablas, internamente no las trata de forma diferente al resto de campos.

Los desarrolladores comentan en la documentación que todas estas carencias no les resultaba un problema, ya que era lo que ellos necesitaban. De hecho, MySQL fue diseñada con estas características, debido a que lo que buscaban era un gestor de bases de datos con una gran rapidez de respuesta. [15]

3.1.3. PostgreSQL

PostgreSQL es un sistema de dirección de base de datos objeto-correlativo basado en POSTGRES, desarrollada en la Universidad de California al Berkeley Departamento Informática. POSTGRES abrió camino muchos conceptos que sólo se pusieron disponibles muy después en algunos sistemas de la base de datos comerciales. También, PostgreSQL puede extenderse por el usuario de muchas maneras, por ejemplo agregando nuevo tipos de datos, funciones ,operadores ,funciones del agregado ,ponga en un índice los métodos ,los idiomas procesales ,y debido a la licencia liberal, PostgreSQL puede usarse, modificado, y distribuyó por todos gratis para cualquier propósito, sea él privado, comercial, o académico. [16]

3.1.4. Selección de PostgreSQL.

PostgreSQL ofrece muchas ventajas respecto a otros sistemas de bases de datos [17]:

Instalación ilimitada: Es frecuente que las bases de datos comerciales sean instaladas en más servidores de lo que permite la licencia. Algunos proveedores comerciales consideran a esto la principal fuente de incumplimiento de licencia. Con PostgreSQL, nadie puede demandarlo por violar acuerdos de licencia, puesto que no hay costo asociado a la licencia del software.

Estabilidad y confiabilidad legendarias: En contraste a muchos sistemas de bases de datos comerciales, es extremadamente común que compañías reporten que PostgreSQL nunca ha presentado caídas en varios años de operación de alta actividad. Ni una sola vez. Simplemente funciona.

Extensible: El código fuente está disponible para todos sin costo. Si su equipo necesita extender o personalizar PostgreSQL de alguna manera, pueden hacerlo con un mínimo esfuerzo, sin costos adicionales. Esto es complementado por la

comunidad de profesionales y entusiastas de PostgreSQL alrededor del mundo que también extienden PostgreSQL todos los días.

Multiplataforma: PostgreSQL está disponible en casi cualquier Unix (34 plataformas en la última versión estable).

Diseñado para ambientes de alto volumen: PostgreSQL usa una estrategia de almacenamiento de filas llamada MVCC para conseguir una mejor respuesta en ambientes de grandes volúmenes.

A continuación es mostrada una serie de características técnicas que tiene PostgreSQL como sistema gestor y que fueron utilizadas para el desarrollo de la base de datos del SACPIC.

- **Cliente/Servidor** PostgreSQL usa una arquitectura proceso-por-usuario cliente/servidor. Esta es similar al método del Apache 1.3.x para manejar procesos. Hay un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intente conectar a PostgreSQL [17].

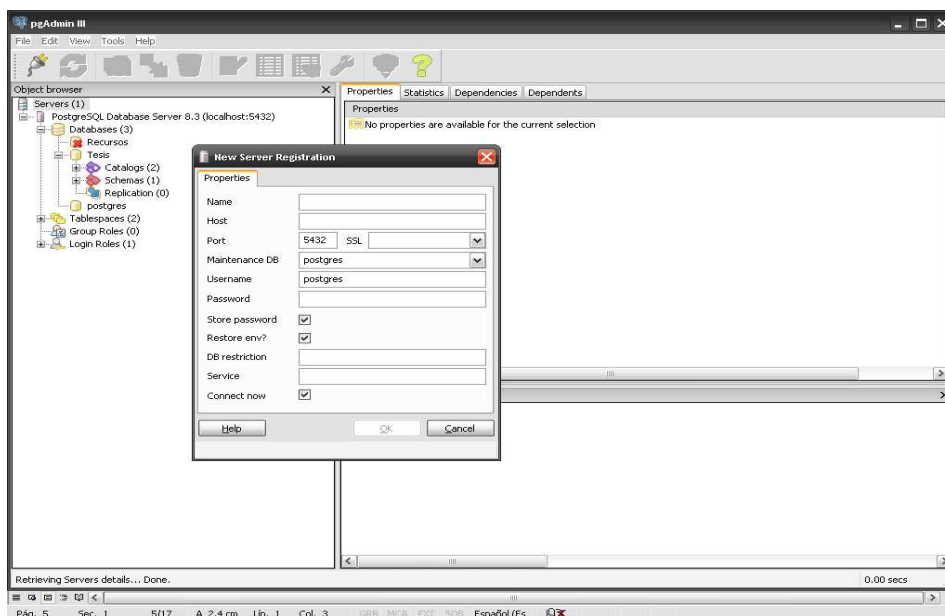


Figura 3.1 Características de PostgreSQL.

- **Altamente Extensible** Soporta operadores, funciones, métodos de acceso y tipos de datos definidos por el usuario.
- **Integridad Referencial** Soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la Base de Datos [18].

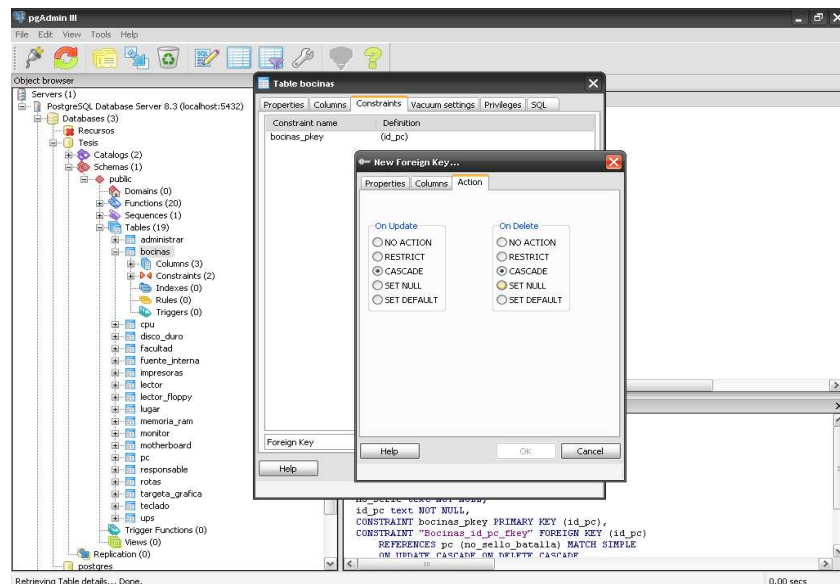


Figura 3.2. Integridad Referencial de PostgreSQL.

En la siguiente figura es mostrado como fue creado a través de PostgreSQL una tabla de la Base de datos de SACPIC, en este caso la de Computadora.

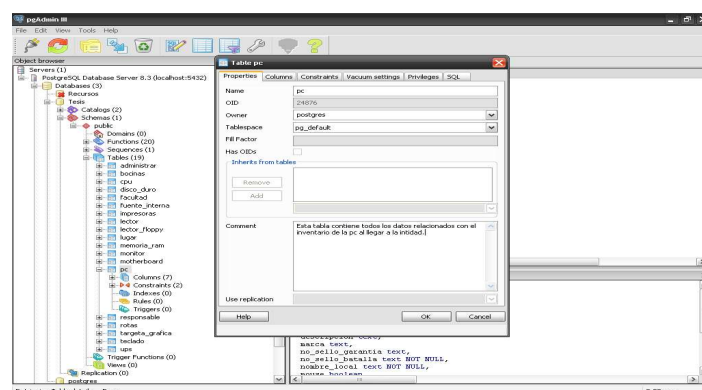


Figura 3.3. Creando Tabla de PC en PostgreSQL.

3.2. Implementación de la Interfaz de Usuario del SACPIC.

Una importante y delicada selección debe ser la de las herramientas a usar a la hora de implementar el sistema, esto finalmente permitirá cumplir con los requerimientos del cliente (listados en el capítulo II) y obtener un software en gran forma exitoso.

En esta investigación se utilizó como lenguaje de programación Java y como entorno de desarrollo el proporcionado por la plataforma de desarrollo NetBeans IDE en su versión 6.5. A continuación es mostrado un análisis de los lenguajes de programación y finalmente se describe las razones que se tuvieron en cuenta para la selección de java como lenguaje de programación.

3.2.1. Lenguajes de programación.

Un lenguaje de programación es un idioma artificial diseñado para expresar computaciones que pueden ser llevadas a cabo por máquinas como las computadoras. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana. Está formado por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. [18]

Actualmente existen diversos lenguajes en el mundo, a continuación se brinda brevemente características de algunos de ellos y se muestra la razón de la selección del lenguaje utilizado para programar SACPIC.

3.2.1.1. C++

C++ es un lenguaje de programación diseñado a mediados de los años 1980 por Bjarne Stroustrup. La intención de su creación fue el extender al exitoso lenguaje de programación C con mecanismos que permitan la manipulación de objetos. En

ese sentido, desde el punto de vista de los lenguajes orientados a objetos, el C++ es un lenguaje híbrido.

Posteriormente se añadieron facilidades de programación genérica, que se sumó a los otros dos paradigmas que ya estaban admitidos (programación estructurada y la programación orientada a objetos). Por esto se suele decir que el C++ es un lenguaje de programación multiparadigma.[19]

Una particularidad del C++ es la posibilidad de redefinir los (operadores), y de poder crear nuevos tipos que se comporten como tipos fundamentales.

3.2.1.2. C #

El C# es un lenguaje de programación orientado a objetos. Con este nuevo lenguaje se quiso mejorar con respecto de los dos lenguajes anteriores de los que deriva el C, y el C++.

Con el C# se pretendió que incorporase las ventajas o mejoras que tiene el lenguaje JAVA. Así se consiguió que tuviese las ventajas del C, del C++, pero además la productividad que posee el lenguaje JAVA y se le denominó C#. [20]

Algunas de las características del lenguaje de programación C# son: Su código se puede tratar íntegramente como un objeto. Su sintaxis es muy similar a la del JAVA. Es un lenguaje orientado a objetos y a componentes. Armoniza la productividad del Visual Basic con el poder y la flexibilidad del C++. Ahorramos tiempo en la programación ya que tiene una librería de clases muy completa y bien diseñada. [20]

Las desventajas que se derivan del uso de este lenguaje de programación son que en primer lugar se tiene que conseguir una versión reciente de Visual Studio .NET, por otra parte se tiene que tener algunos requerimientos mínimos del sistema para poder trabajar adecuadamente tales como contar con Windows NT 4 o superior, tener alrededor de 4 gigas de espacio libre para la pura instalación, etc. [20]

3.2.1.3. Java

Java surgió en 1991 cuando un grupo de ingenieros de Sun Microsystems trataron de diseñar un nuevo lenguaje de programación destinado a electrodomésticos. La reducida potencia de cálculo y memoria de los electrodomésticos llevó a desarrollar un lenguaje sencillo capaz de generar código de tamaño muy reducido. [21]

El lenguaje Java puede considerarse como una evolución del C++. La sintaxis es parecida a la de este lenguaje. A pesar de que puede considerarse como una evolución del C++ no acarrea los inconvenientes del mismo, ya que Java fue diseñado “partiendo de cero”, es decir, no necesitaba ser compatible con versiones anteriores de ningún lenguaje como ocurre con C++ y C. [21]

Gracias a que fue diseñado “partiendo de cero” ha conseguido convertirse en un lenguaje orientado a objetos puro, limpio y práctico. No permite programar mediante otra técnica que no sea la programación orientada a objetos y, una vez superado el aprendizaje de la programación orientada a objetos, es realmente sencillo aprender Java. [21]

3.2.1.4. Selección de Java como lenguaje para ser utilizado

Java es un lenguaje de desarrollo de propósito general, y como tal es válido para realizar todo tipo de aplicaciones profesionales. Incluye una combinación de características que lo hacen único y está siendo adoptado por multitud de fabricantes como herramienta básica para el desarrollo de aplicaciones comerciales de gran repercusión. [22]

A continuación se listan una serie de características que los hacen distinto de los demás lenguajes:

1. **Simple.** Java, como todo lenguaje de programación de propósito general, no es sencillo de usar y menos de dominar. Por tanto debemos definir su simplicidad en función de algo. Si nos basamos en la dificultad de programar en C++ (lenguaje tomado como origen por los diseñadores de Java), entonces sí podemos afirmar

que el lenguaje es simple. Las principales simplificaciones consisten en no necesitar tener ficheros de cabeceras, la ausencia de punteros, no tener que gestionar la memoria y disponer de un extenso surtido de librerías que nos facilita la faena cuando tenemos que enfrentarnos a programas de cierta complejidad. También tenemos muchas menos estructuras de programación y habitualmente más sencillas.[22]

2. **Orientado a objetos.** Java es, efectivamente, un lenguaje plenamente orientado a objetos. Respecto a C++, las mayores diferencias son la forma diferente de ligar invocaciones polimórficas a métodos y la manera distinta de resolver la herencia múltiple.

3. **Distribuido.** Una de las líneas principales que marcaron el diseño de Java fue su orientación a usos relacionados con Internet. Cualquiera que haya intentado programar alguna aplicación Internet en otro lenguaje se dará cuenta de la simplicidad de Java en estos menesteres.

4. **Robusto.** La robustez de los programas Java también se ve beneficiada gracias a que muchos errores que normalmente encontramos en tiempo de ejecución Java los detecta en tiempo de compilación. También podemos recoger los errores que se produzcan en tiempo de ejecución gracias al mecanismo de excepciones, donde podremos dar soluciones alternativas a los problemas que puedan suceder en la ejecución de nuestro programa. [22]

5. **Seguro.** Puesto que se pretende usar Java en entornos en red y distribuidos se han cuidado mucho los aspectos relacionados con seguridad. Parece difícil que una aplicación Internet hecha con Java pueda dañar la integridad del sistema cliente accediendo a zonas de memoria o del sistema de ficheros no permitidas, pero este es un punto en el que siempre pueden surgir agujeros.

6. **Arquitectura neutral.** El compilador genera un fichero objeto con un formato independiente de la arquitectura. Esto permite que el código compilado sea ejecutable en todos los procesadores para los que exista un run time system Java

(llamado Máquina Virtual Java). El compilador logra esto generando instrucciones *bytecode*, que no son de ninguna arquitectura en particular, sino que están pensadas para ser fácilmente traducidas *on the fly* a cualquier lenguaje máquina existente, y por tanto interpretadas. Esto supone que un programa hecho en Java y compilado a *bytecode* puede funcionar en diferentes plataformas *hardware/software* y comportarse exactamente de la misma forma. [22]

7. **Portable.** Además de generarse un código independiente de la arquitectura se han cuidado al máximo todos los aspectos relacionados con portabilidad. Los tipos de datos ocupan estrictamente lo mismo en cualquier plataforma (a diferencia de C/C++ donde un int puede tener 16 o 32 bits según la implementación), y se codifica todo en *big endian*.

8. **Intepretado.** Java no genera un ejecutable diferente cada vez que compilamos en una plataforma distinta sino que, independientemente de la plataforma, se generará el mismo código intermedio (*bytecode*), que después será ejecutado en cualquier plataforma por un intérprete específico para la plataforma. [22]

9. **Alto rendimiento.** Un lenguaje interpretado acostumbra a ser entre 20 y 100 veces más lento que el mismo programa compilado y ejecutado. Teniendo esto en cuenta, el rendimiento de Java es más que aceptable, pero en valor absoluto, hablar de “alto rendimiento” es una ligera imprecisión.

10. **Multiflujo.** La programación multiflujo (*multithreading*) en Java ofrece unos resultados muy buenos, especialmente si los comparamos con otros lenguajes. En parte es debido a que ofrece unas librerías específicas que nos simplifican mucho el trabajo. Se aprovechan también las ventajas de los sistemas multiprocesadores.

11. **Dinámico.** Java está diseñado para adaptarse a entornos en evolución. En este sentido las librerías pueden añadir nuevos métodos y atributos a sus clases sin afectar a los clientes. Es posible cargar nuevas clases en tiempo de ejecución y consultar los métodos que estas tienen disponibles.

Para la implementación de la interfaz de usuario se utilizaron una serie de propiedades con las que cuenta java como lenguaje. A continuación se muestra la utilización de las mismas durante el desarrollo de SACPIC.

- Mecanismo para captura de excepciones, mediante sentencias try y catch: Para capturar una excepción en particular hay que colocar el código que puede lanzar dicha excepción dentro de un bloque try, entonces se crea una lista de bloques catch adyacentes, uno para cada posible excepción que se quiera capturar. El código de un bloque catch se ejecuta cuando la excepción generada coincide en tipo con la del bloque catch.[21]

```
try {
    if(jButton1.getText().equals("Insertar")) {
        if (!impresora.Buscar(compu)) {

            impresora.Insertar();
            JOptionPane.showMessageDialog(null,"Inserción Satisfactoria");
            jTextField2.setText("");
            jTextField3.setText("");
            jTextField4.setText("");
            Object[] vector1 = {compu, no_serie,modelo,fabricante};
            defaultTableModel.addRow(vector1);
            jTable1.setModel(defaultTableModel);
        } else
            JOptionPane.showMessageDialog(null,"Esta Impresora ya existe en la Base de Datos");
    }
} catch (SQLException ex) {
    Logger.getLogger(V_Computadora.class.getName()).log(Level.SEVERE, null, ex);
} catch (ClassNotFoundException ex) {
    Logger.getLogger(V_Computadora.class.getName()).log(Level.SEVERE, null, ex);
}
```

Figura 3.4 Tratamiento de excepciones en SACPIC.

- Java es un lenguaje fuertemente tipificado, lo que quiere decir toda variable debe ser declarada de un tipo. De los ocho tipos primitivos, hay seis numéricos (cuatro enteros y dos en coma flotante), otro es el carácter y el último es el booleano. La portabilidad de Java garantiza todos los tipos tendrán el mismo tamaño independientemente de la plataforma.[22]

```

public class C_Computadora {
    private String tipo_equipo;
    private String descripcion;
    private int marca;
    private float no_sello_garantia;
    private int no_sello_batalla;
    private String nombre_local;
    private boolean mouse;
}

```

Figura 3.5. Tipo de datos usados en SACPIC.

- Java define operadores aritméticos, relacionales, lógicos de manipulación de bits, de conversión de tipo, de clase, de selección y de asignación.

```

public void center() {
    Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();
    Dimension frameSize = this.getSize();
    if (frameSize.height > screenSize.height)
    {
        frameSize.height = screenSize.height;
    }
    if (frameSize.width > screenSize.width)
    {
        frameSize.width = screenSize.width;
    }
    this.setLocation((screenSize.width - frameSize.width) / 2, (screenSize.height - frameSize.height) / 2);
}

```

Figura 3.6 Usando operadores en SACPIC.

- Uso de clases ya existentes: Java permite instanciar objetos de una clase ya existente e invocar métodos sobre los mismos.

```

String source="admin\\admin123\\nbesarrollo\\ncd080";

Vector guardar = new Vector();
FileReader entrada = null;
BufferedReader br = null;
File f = new File("Conexion_BD.txt");

if (!f.exists()) {
    FileWriter fout = null;
    try {
        fout = new FileWriter("Conexion_BD.txt");
        fout.write(source);
        Conexion.usaz = "postgres";
        Conexion.password = "1234/";
        Conexion.base_datos = "Control_pc";
        Conexion.host = "127.0.0.1";
    } catch (IOException ex) {
    } finally {
        try {
            fout.close();
        } catch (IOException ex) {
        }
    }
}

```

Figura 3.7. Utilizando clases de java en SACPIC.

3.2.1 NetBeans como entorno de desarrollo para lenguaje seleccionado.

NetBeans es un entorno que está desarrollado en Java, lo que hace posible su uso en Windows, GNU/Linux, Mac OS X y Solaris y, virtualmente, sobre cualquier otro sistema operativo para el que exista un JRE (*Java Runtime Environment*). Es un producto basado en código libre y con licencia GPLv2. Ofrece un impresionante conjunto de posibilidades, en especial para todos los tipos de proyectos basados en Java. El diseñador de interfaces de usuario Swing simplifica una tarea que siempre ha planteado ciertas dificultades, al ocuparse de la correcta colocación, espaciado y distribución de los componentes sin tener que especificar todos los detalles manualmente.[23]

La construcción de los programas clásicos basados en ventanas y que se ejecutan directamente en el ordenador del usuario como es el caso de SACPIC resulta especialmente cómoda y rápida en NetBeans gracias a los diseñadores que ofrece para trabajar con componentes Java Swing. [23]

NetBeans proporciona una serie de facilidades a la hora de diseñar e implementar la interfaz de usuario, dichas facilidades han sido utilizadas para el desarrollo de SACPIC, y son mostradas a continuación.

- **Diseñar la interfaz de usuario:** Es una tarea que se lleva a cabo en un entorno con un contenedor, el formulario, una paleta de componentes prefabricados y una lista de propiedades que facilita la personalización de estos. Mediante la técnica de arrastrar y soltar, los controles se colocan en el contenedor, tras lo cual es posible modificar sus atributos.[23]

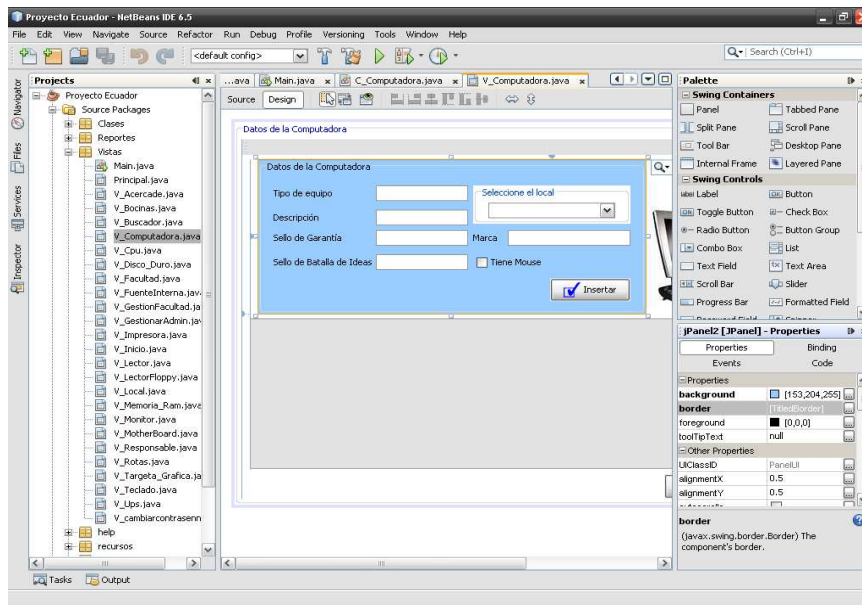


Figura 3.8 Diseñando la interfaz de usuario en la Herramienta.

- **La escritura de código:** Se reduce al mínimo, ya que incluso las consultas a bases de datos y conexión de los componentes se efectúan mediante asistentes. El editor de código cuenta con todo tipo de ayudas: listas de parámetros al invocar a un método, listas de miembros de un objeto, ventajas emergentes con la documentación de las propiedades y métodos, etc.

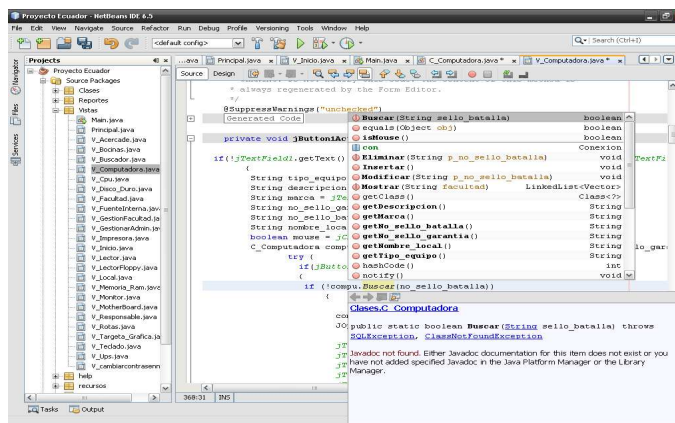


Figura 3.9. Ejemplo de código en SACPIC.

- **Realiza verificaciones:** En busca de problemas tanto en tiempo de compilación como en tiempo de ejecución en NetBeans se detecten errores lo antes posible, normalmente en el ciclo de desarrollo. El código Java pasa muchos tests antes de ejecutarse en una máquina. Pasa a través de un verificador de código de byte que comprueba el formato de los fragmentos de código y aplica un probador de teoremas para detectar fragmentos de código ilegal, código que falsea punteros, viola derechos de acceso sobre objetos o intenta cambiar el tipo o clase de un objeto [23].

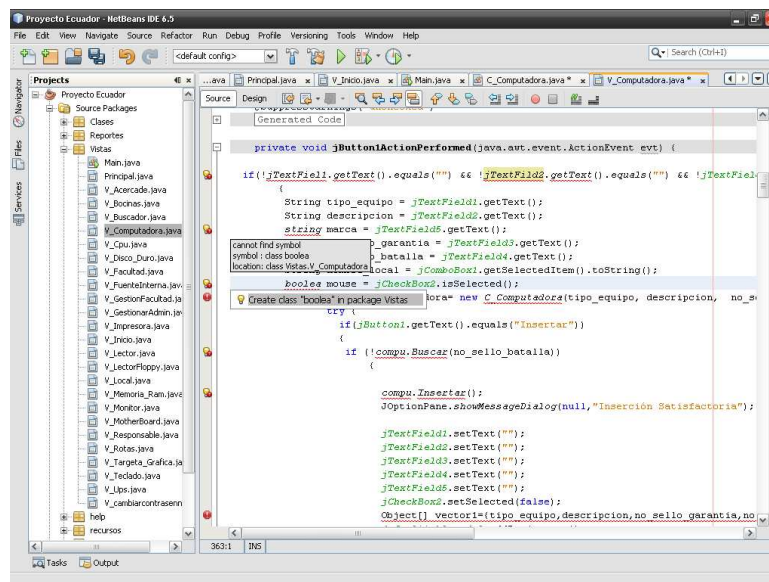


Figura 3.10 Ejemplo de corrección de errores de NetBeans.

3.3 Implementación de la Ayuda.

Para el diseño e implementación de la ayuda de SACPIC se hizo uso de la biblioteca Java Help. La misma es JavaHelp es una expansión de Java que facilita la programación de las ventanas de ayuda en las aplicaciones java.

Las ventanas de ayuda de JavaHelp se configuran por medio de varios ficheros en formato XML. Los textos de ayuda que se quieran mostrar se escribirán en ficheros con formato HTML. Para el desarrollo de los ficheros en el formato HTML se utilizó la herramienta Macromedia Dreamweaver el cual es un editor de código HTML profesional para el diseño visual y la administración de sitios y páginas Web. [24]

Los ficheros en el formato HTML son las páginas de la ayuda que serán mostradas a los usuarios del sistema y los ficheros en formato XML deben configurar para que estén relacionados con la información de las páginas HTML. [24]

A continuación es mostrado un ejemplo de cómo se creó una página de la ayuda del sistema en HTML.

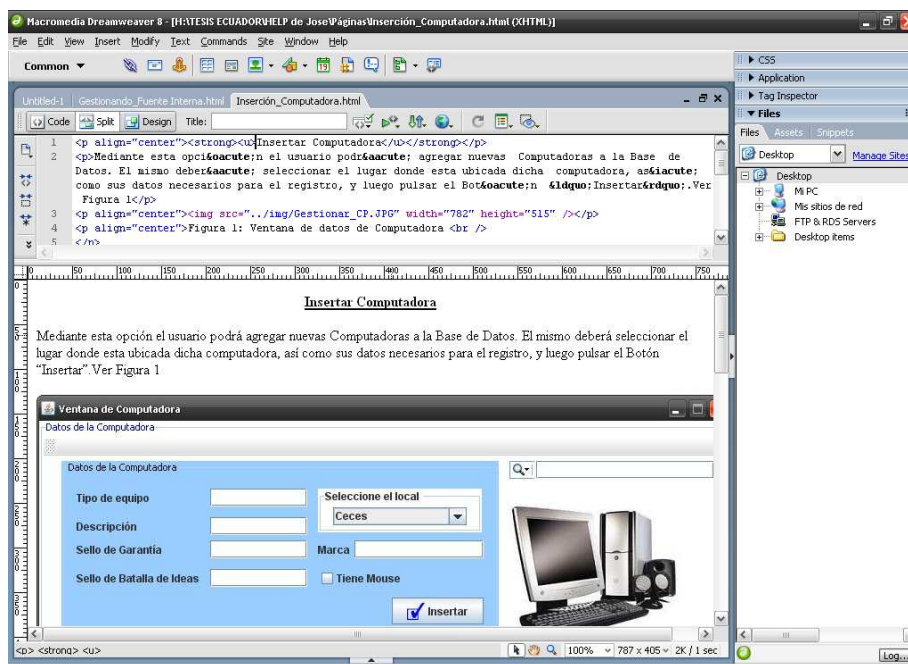


Figura 3.11. Diseño de página de ayuda en Dreamweaver.

Una vez terminadas las páginas fueron actualizados los ficheros XML para conectar dichas páginas a la ventana de Java. Finalmente es mostrada la ayuda del sistema.



Figura 3.12. Ayuda de SACPIC.

3.4 Modelo de Componentes del Sistema.

El modelo de componentes ilustra los componentes de software que se usarán para construir el sistema. Se pueden construir a partir del modelo de clases y escribir desde cero para el nuevo sistema o se pueden importar de otros proyectos y de productos de terceros.

Los estereotipos estándar de componentes se listan a continuación [25]:

- <<executable>> es un programa que puede ser ejecutado.
- <<file>> es un fichero que contiene código fuente o datos.
- <<library>> es una librería estática o dinámica.
- <<table>> es una tabla de Base de Datos.
- <<document>> es un documento.

El diagrama de componentes muestra la relación entre componentes de software, sus dependencias, su comunicación y otras condiciones.

A Continuación es mostrado el diagrama de componentes de SACPIC.

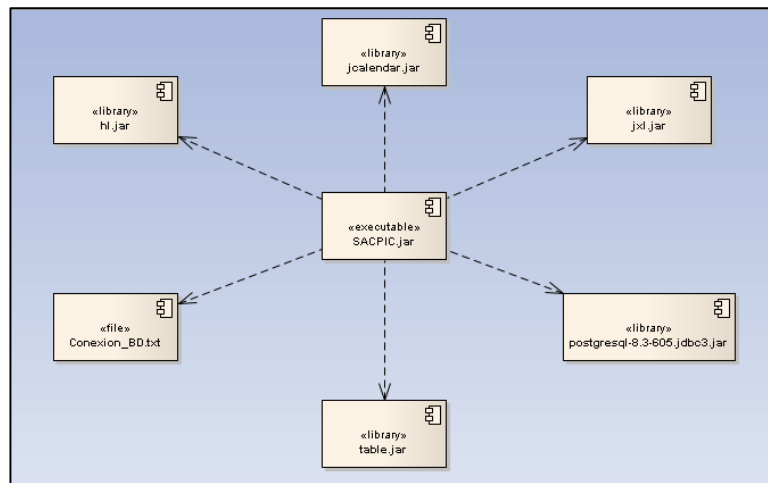


Figura 3.13. Diagrama de Componentes de SACPIC.

3.5 Seguridad del SACPIC.

La seguridad de la información involucra la implementación de estrategias que cubran los procesos en donde la información es el activo primordial. Estas estrategias deben tener como punto principal el establecimiento de políticas, controles de seguridad, tecnologías y procedimientos para detectar amenazas que puedan explotar vulnerabilidades y que pongan en riesgo dicho activo, es decir,

que ayuden a proteger y salvaguardar tanto información como los sistemas que la almacenan y administran.

La implementación de niveles de usuario es requerida en SACPIC ya que en el mismo se gestionan datos relacionados con el control de las piezas de las computadoras de la Universidad de Pinar del Río.

En la aplicación existen dos niveles de usuario: El Administrador y el Usuario.

El Administrador tiene permisos para acceder todas las funcionalidades dentro del sistema, en el caso del Usuario solo podrá consultar la información almacenada en la base de datos mostrada a través de reportes.

La seguridad del sistema está en que cada persona para poder acceder a la aplicación necesita autenticarse con usuario y contraseña y dependiendo de sus datos el sistema verifica si tiene o no permisos administrativos. Ver Figura 3.14.



Figura 3.14. Autenticación en SACPIC.

CONCLUSIONES

Con el desarrollo de esta investigación se cumplieron los objetivos propuestos, al obtenerse una aplicación informática con una Base de Datos que gestiona la información relacionada con las piezas de las computadoras, lo cual hace disponible una herramienta para el almacenamiento y muestra de reportes de todos los datos, permitiendo tener un mejor control sobre los componentes de las computadoras.

Se lograron los siguientes resultados:

1. Se diseñó e implementó una Base de Datos que almacena toda la Información relacionada con las piezas de las computadoras.
2. Se desarrolló una aplicación informática que es capaz de gestionar y consultar toda la información almacenada en la Base de Datos, brindando un mayor control de la información al personal dedicado a esta tarea.
3. Se creó una Ayuda de Usuario que facilita la explotación del sistema desarrollado.

RECOMENDACIONES

Se recomienda:

1. Capacitar al personal que trabajará con el sistema para garantizar un adecuado uso de la herramienta.
2. Introducir el uso del SACPIC en universidad de Pinar del Río evitando posibles pérdidas de piezas por no llevar un control exacto de las mismas.
3. Mantener actualizados los datos que manipula el sistema para asegurar un correcto control de las piezas de las computadoras en cada local.
4. Realizar el mantenimiento del SACPIC para solucionar los posibles problemas que puedan aparecer a medida que se utilice el sistema.

REFERENCIAS BIBLIOGRAFICAS

[1] HEISIG, P. (1999). Gestión del conocimiento: mejores prácticas en Europa. Curso Como implantar la gestión del conocimiento en su empresa. Madrid: Club gestión de calidad.

[2] "Sistemas de Información" Disponible en <http://www.monografias.com/trabajos27/sistemas-informatica/sistemas-informatica.shtml> . Consultado 10-10-2011.

[3] "La Computadora" Disponible en <http://colposfesz.galeon.com/libro/menus.htm> Consultado 10-10-2011.

[4] "Cómo utilizar el programa Everest". Disponible en <http://es.kioskea.net/faq/1683-como-utilizar-el-programa-everest> Consultado 10-10-2011.

[5] "Computadoras" Disponible en <http://es.wikipedia.org/wiki/Computadora>. Consultado 11-12-2011.

[6]"Componentes de las PC" Disponible en <http://www.mailxmail.com/curso-componentes-pc-s/funciones-pc> Consultado 11-10-2011.

[7] "Unidad Central de Proceso o CPU" Disponible en <http://www.mitecnologico.com/Main/UnidadCentralDeProceso> Consultado 05-10-2011.

[8] "UPS" Disponible en <http://www.informaticamoderna.com/UPS.htm> Consultado 05-10-2011.

[9] M. Peralta, "Estimación del esfuerzo basada en casos de uso", Instituto Tecnológico de Buenos Aires, 2001.

[10] G. Jacobson, I. Booch, J. Rumbaugh, "El Proceso Unificado de Desarrollo de Software".2000.

[11] G. Jacobson, I. Booch, J. Rumbaugh; "El Lenguaje Unificado de modelado". 2000.

[12] "Enterprise Architect - Herramienta de diseño UML" Disponible en <http://www.sparxsystems.com.ar/products/ea.html> Consultado 05-10-2011

[13] "Sistemas de Gestión de Base de Datos" Disponible en http://www.um.es/geograf/sigmur/sigpdf/temario_9.pdf
Consultado 17-11-2011.

[14] "Microsoft Access Tutorial" Disponible en http://www.quackit.com/microsoft_access/tutorial/ Consultado 17-11-2011.

[15] "Curso de MySQL, Creación de Base de datos" Disponible en <http://mysql.conclase.net/curso/?cap=007a> Consultado 17-11-2011

[16] "SQL avanzado y PHP", Curso Base de Datos PostgreSQL. Disponible en <http://www.usabilidadweb.com.ar/postgre.php>. Consultado 05-10-2011

[17] "Ventajas y Desventajas de PostgreSQL" Disponible en <http://www.google.com/cu/search?q=ventajas-y-desventajas-de-postgresql&btnG=Buscar&hl=es&site=&btnK> . Consultado 20-12-2011.

[18] "Lenguaje_de_Programación" Disponible en http://www.ecured.cu/index.php/Lenguaje_de_Programaci%C3%B3n Consultado 20-12-2011.

[19]" C++ con Clase - Programación C++" " Disponible en <http://c.conclase.net/> Consultado 20-12-2011.

[20]"Curso de iniciación a C#" Disponible en <http://www.elguille.info/NET/cursoCSharpErik/index.htm> Consultado 25-10-2011.

[21] "Aprender Java". Disponible en <http://www.elrincondelprogramador.com>.

Consultado 25-10-2011.

[22] I. Párraga García,"Curso de Java", Febrero 2003. Disponible en

<http://www.lawebdelprogramador.com/cursos/Java/index1.html>.

Consultado 25-10-2011.

[23] NetBeans-65-un-entorno-polifacético".

Disponible en <http://www.netbeans.org>. Consultado 25-10-2011.

[24] "Java Help, Módulo de ayuda para las aplicaciones desktop".

Disponible en <http://javahelp.dev.java.net/> Consultado 25-10-2011.

[25] "Modelo de Componentes". Disponible en http://component_model/
Consultado 25-10-2011.

ANEXOS

ANEXO 1 Descripción del CU “Gestionar datos del Disco Duro”.

Caso de uso	Gestionar datos del Disco Duro
Actor	Administrador
Resumen	El Caso de Uso se inicia cuando el Administrador necesita insertar, modificar o eliminar un Disco Duro en el sistema, una vez realizada su gestión concluye el Caso de Uso.
Referencia	RF3
Precondiciones	El usuario ingresó al sistema como Administrador.
Poscondiciones	Queda actualizado el listado de los Discos Duros.
Pantallas asociadas	

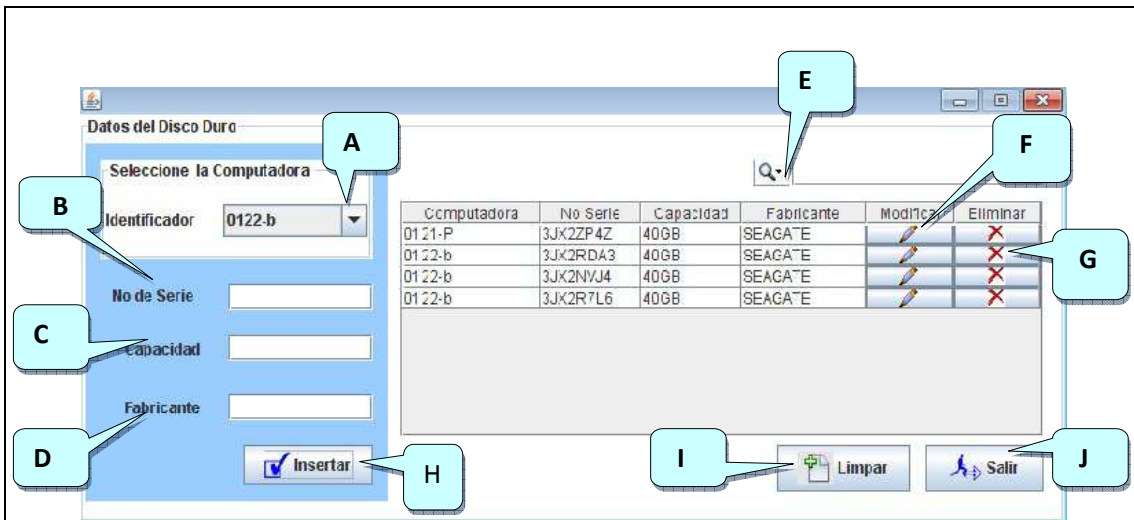


Figura 1: Gestionar datos del Disco Duro.

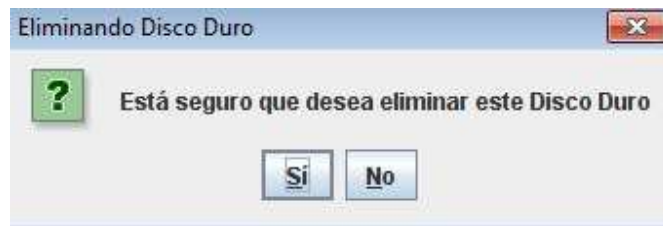


Figura 2: Confirmación de eliminar el Disco Duro.

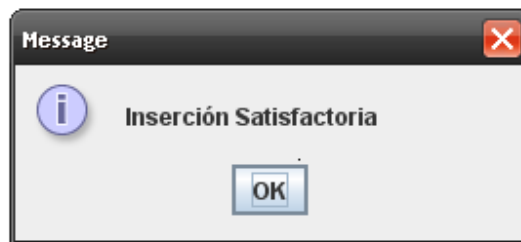


Figura 3: Confirmación de Inserción Correcta.



Figura 4: Confirmación de Modificación Correcta.

Pantallas Asociadas para cursos alternos

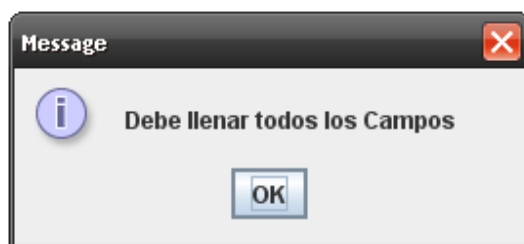


Figura 5: Confirmación de Campos Vacíos.



Figura 6: Confirmación de Existencia del Disco Duro en la Base de Datos.

Sección Principal	
Acciones de los Actores	Respuestas del Sistema
<p>1- El Administrador selecciona la opción (H) en la Pantalla 1 si lo que desea es insertar, la opción (F) en la Pantalla 1 si lo que desea es modificar o la opción de (G) en la Pantalla 1 si lo que desea es eliminar.</p>	<p>2- Si fue la opción insertar muestra la Pantalla 1, ver sección insertar, si fue modificar muestra la Pantalla 1, ver sección modificar, si fue eliminar muestra la Pantalla 2, ver sección eliminar.</p>
Sección Insertar	
<p>1-El Administrador del sistema desea insertar un nuevo Disco Duro por lo que introduce los datos de la misma, desde el elemento A hasta el D de la Pantalla 1, luego presiona el botón Insertar opción H de esta misma pantalla.</p>	<p>2-El sistema verifica la información introducida, de ser correcta la registra, y actualiza la lista mostrada en la parte inferior de la pantalla 1 y muestra un mensaje de "Inserción correcta" como se observa en pantalla 3.</p>

<p>eliminar un Disco Duro por lo que presiona la opción [G] de la pantalla 1.</p> <p>3. El administrador selecciona la opción deseada. Yes o No</p>	<p>2. El sistema envía una confirmación de si está seguro que desea eliminar el Disco Duro correspondiente a la fila de la tabla que ha sido seleccionada por el administrador, como se muestra en la pantalla 2.</p> <p>4. El sistema elimina o no el Disco Duro según la acción deseada por el administrador y es actualizada la tabla de la parte inferior de la pantalla 1.</p>
<p>Cursos Alternos</p>	
<p>Sección: Insertar, Modificar y Eliminar.</p>	
<p>1. Si el administrador ha dejado algún campo sin introducir o los caracteres escritos son no válidos, el sistema mostrará un mensaje como se puede observar en la pantalla 5.</p> <p>2. Si el parámetro correspondiente a la opción [D] de la pantalla 1 es igual al de algún Disco Duro introducido se mostrará un mensaje como se observar en la pantalla 6.</p>	

ANEXO 2 Diagrama de Clases del Análisis para el CU “Gestionar datos Computadora”.

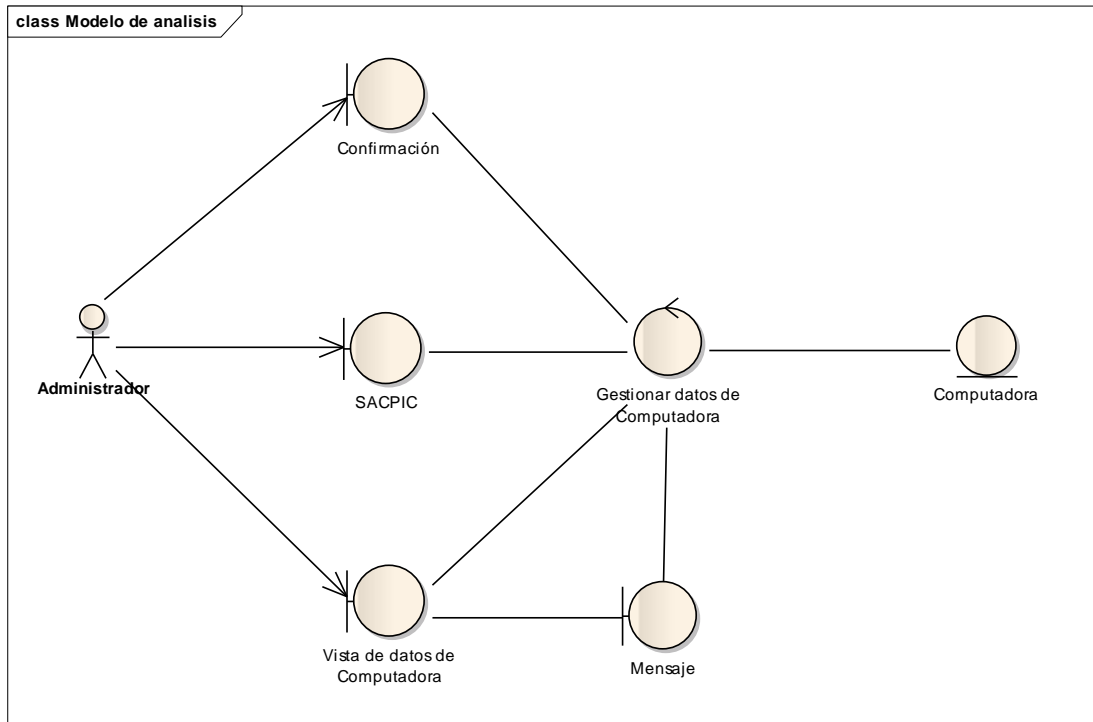


Figura 2.1. Clases del Análisis.

ANEXO 3 Diagramas de Secuencia para el CU “Gestionar datos de Computadora”

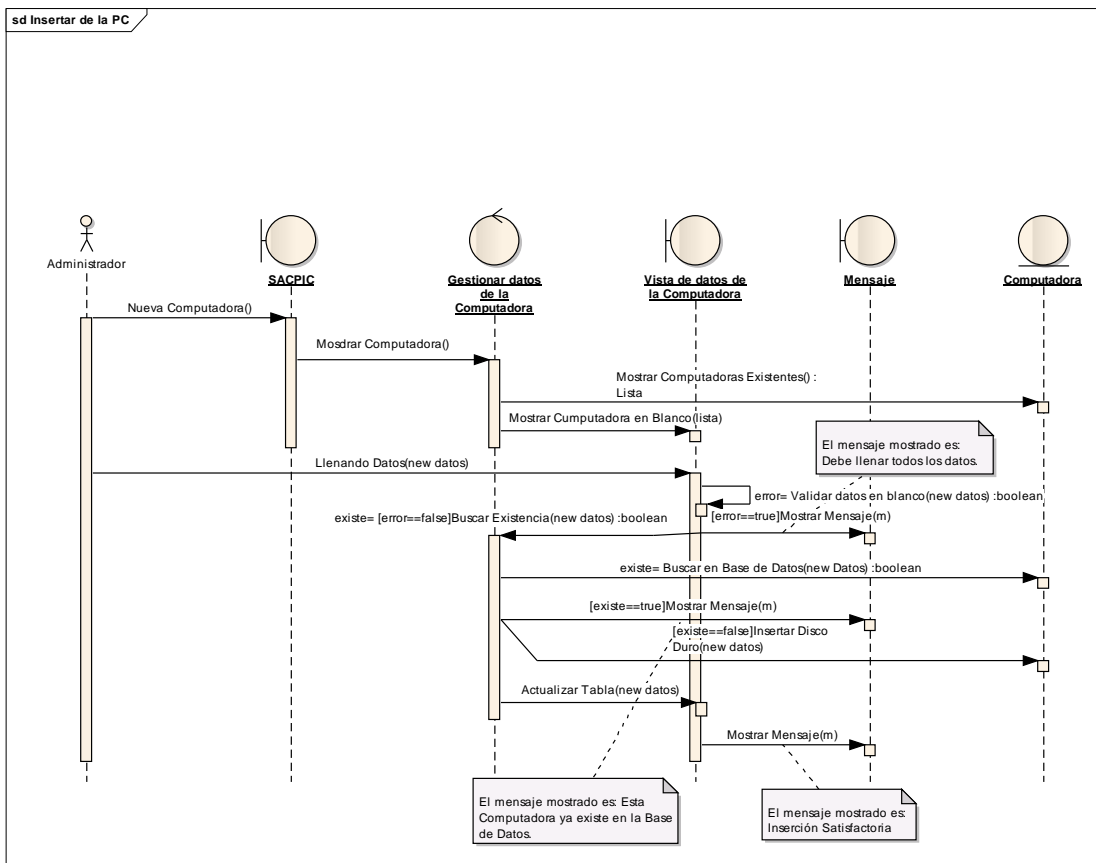


Figura 3.1. Sección Adicionar.

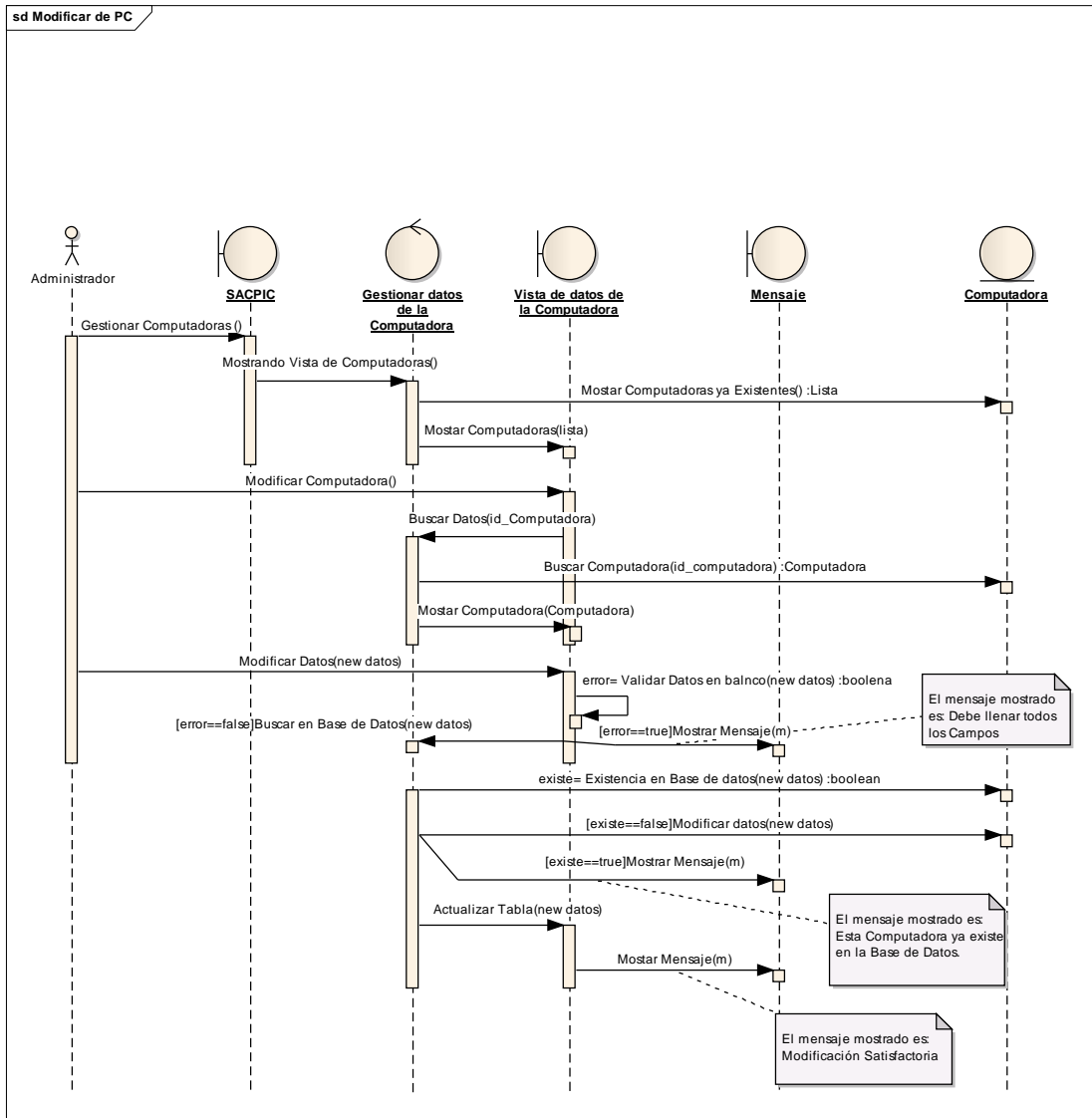


Figura 3.2. Sección Modificar.

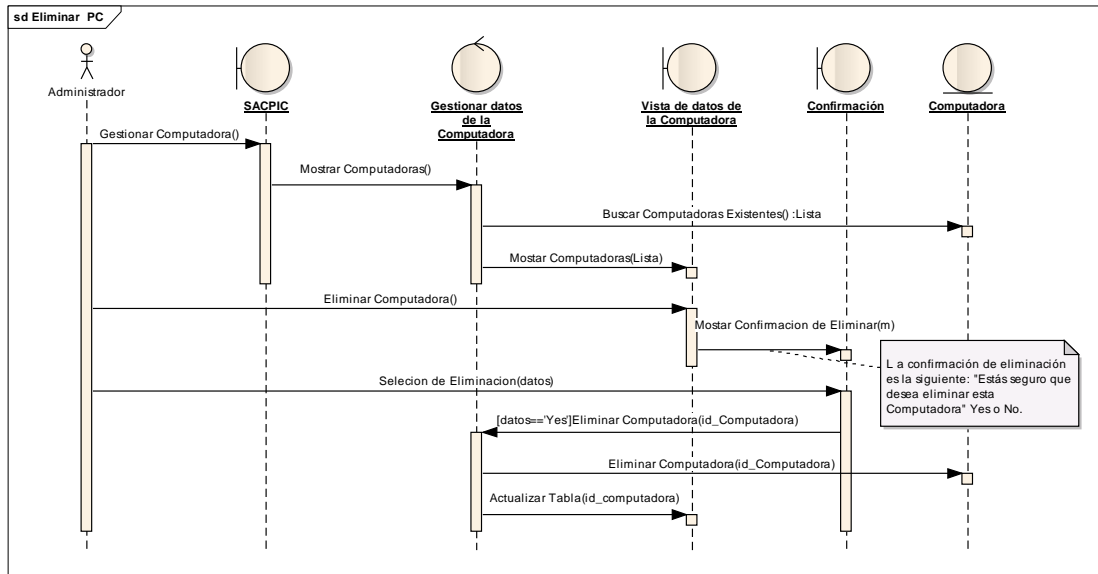


Figura 3.3. Sección Eliminar

ANEXO 4 Diagrama de Clases del Diseño para el CU “Gestionar datos de Computadora”.

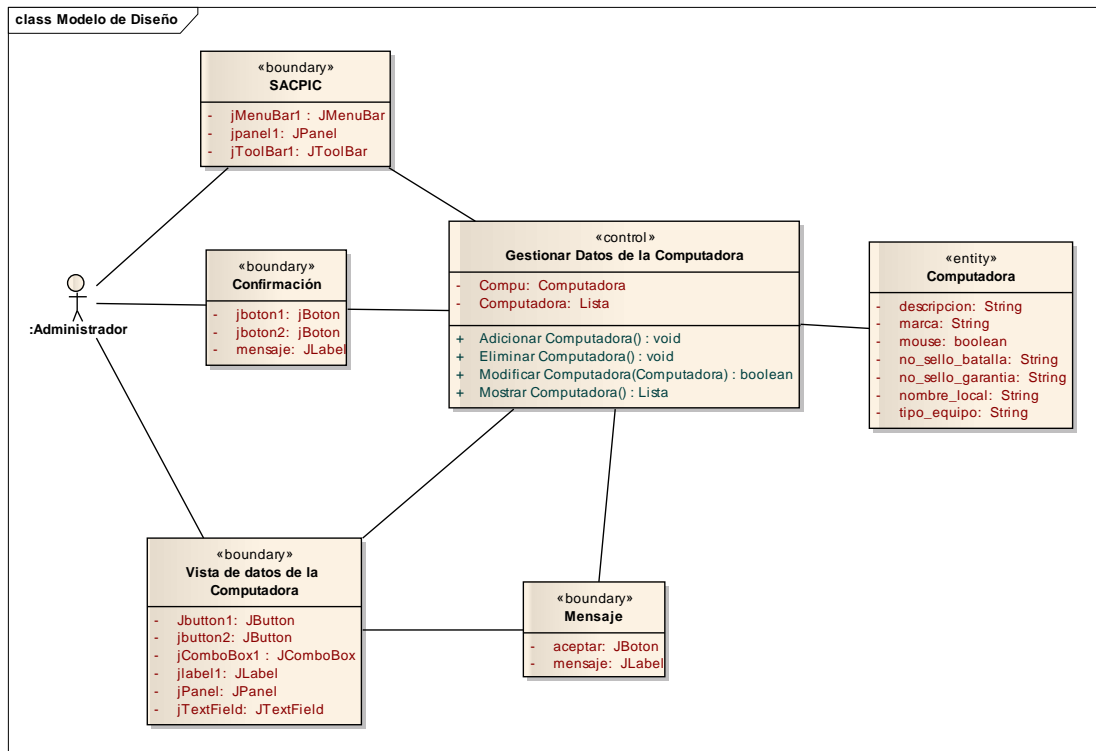


Figura 4.1. Clases del Diseño

ANEXO 5 Diagrama de Clases del Análisis para el CU “Gestionar datos del Disco Duro”.

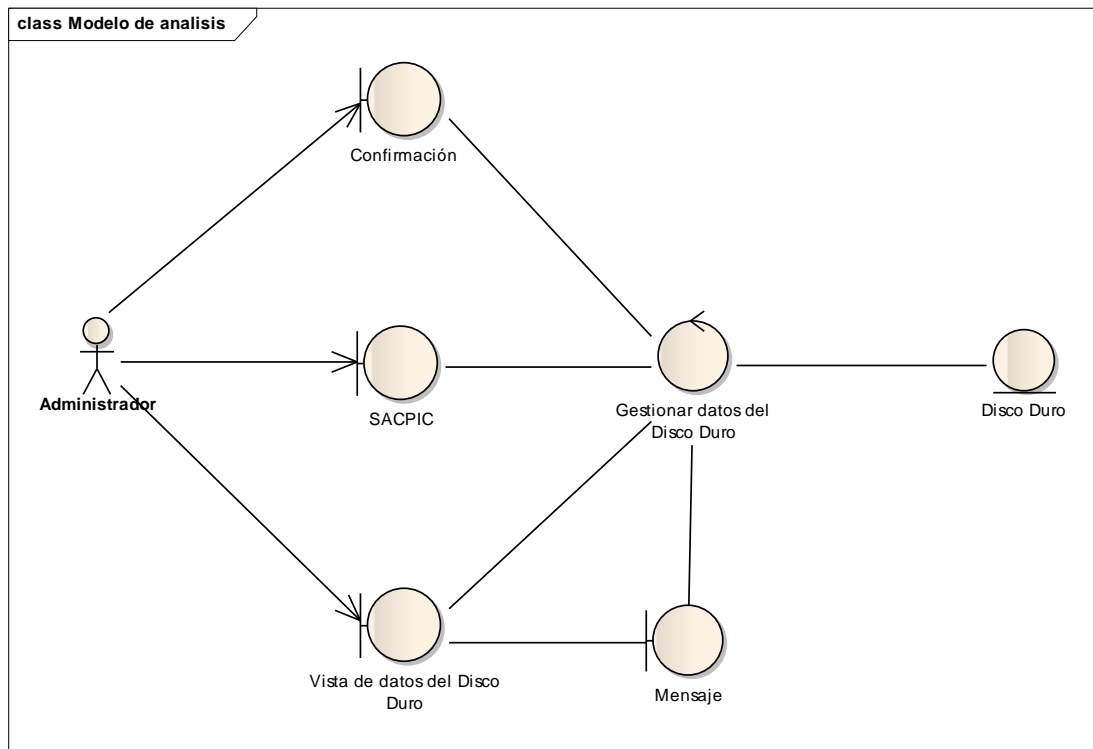


Figura 5.1. Clases del Análisis.

ANEXO 6 Diagramas de Secuencia para el CU “Gestionar datos del Disco Duro”.

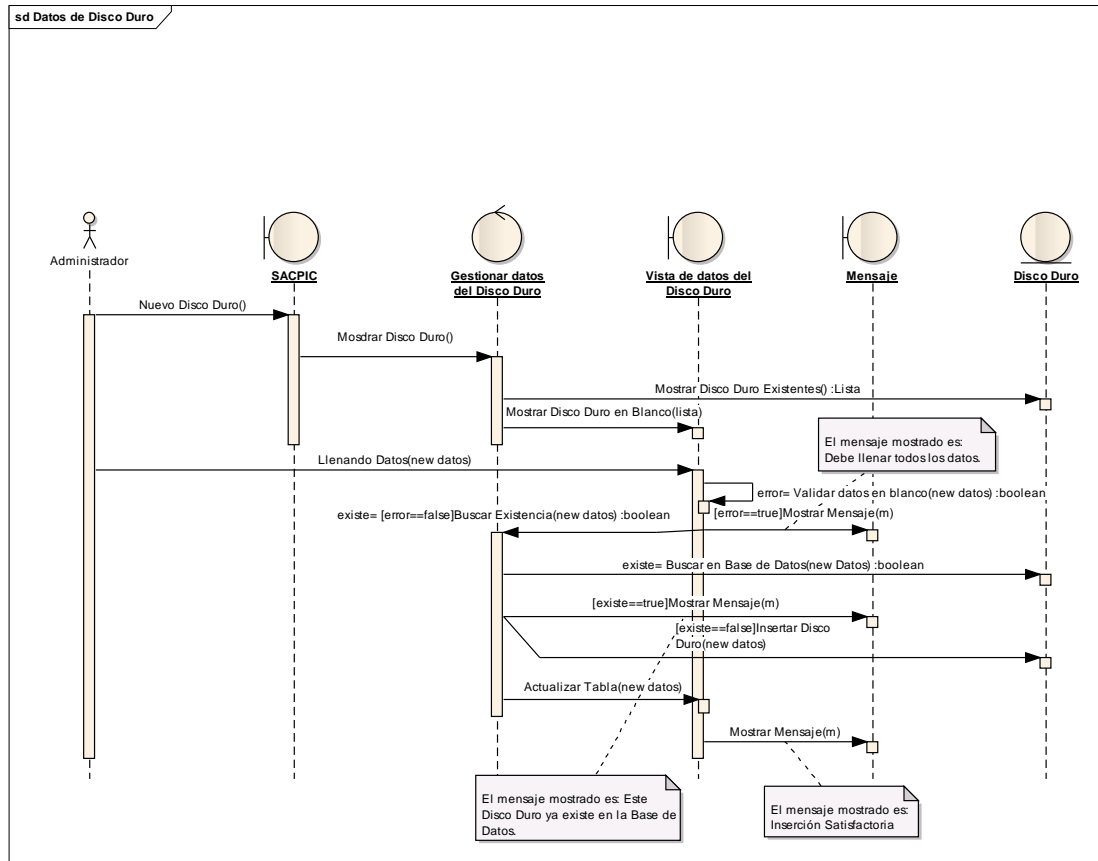


Figura 6.1. Sección Insertar

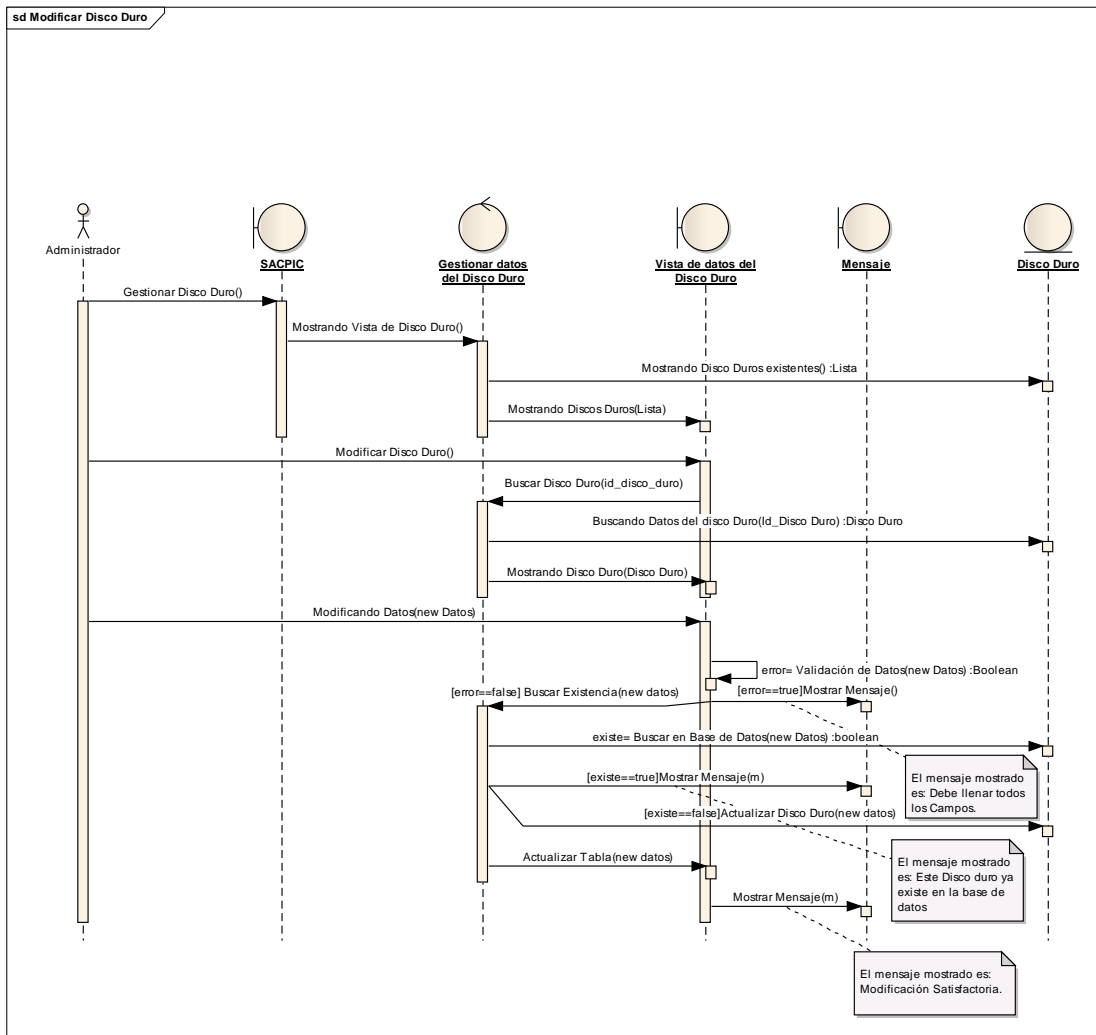


Figura 6.2. Sección Modificar

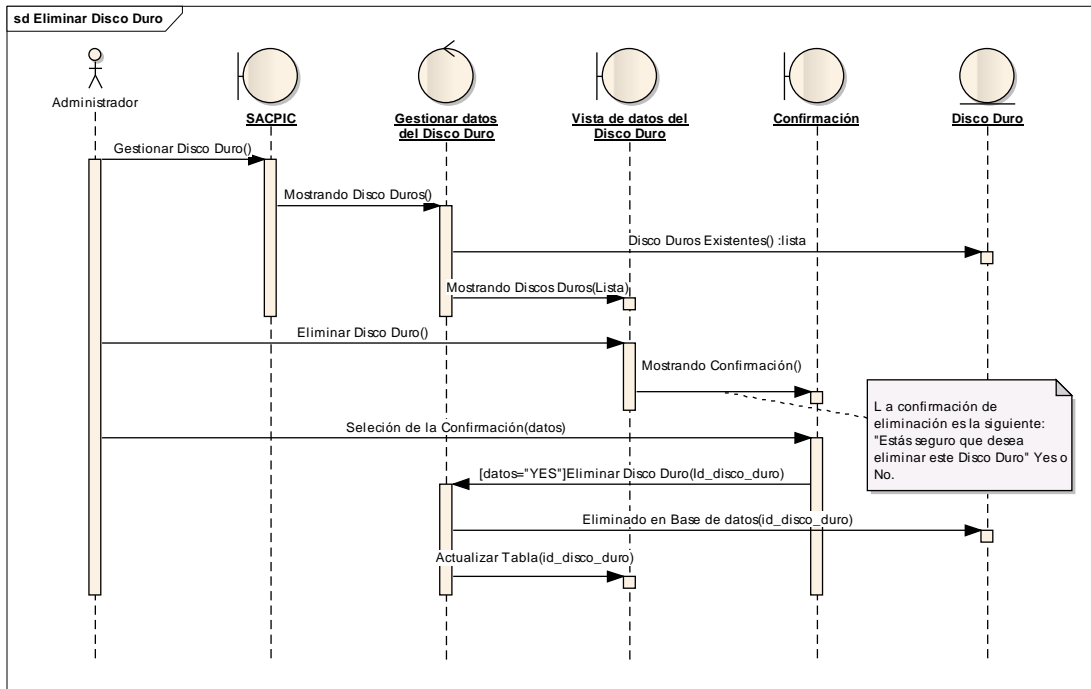


Figura 6.3. Sección Eliminar

ANEXO 7 Diagrama de Clases del Diseño para el CU “Gestionar datos del Disco Duro”.

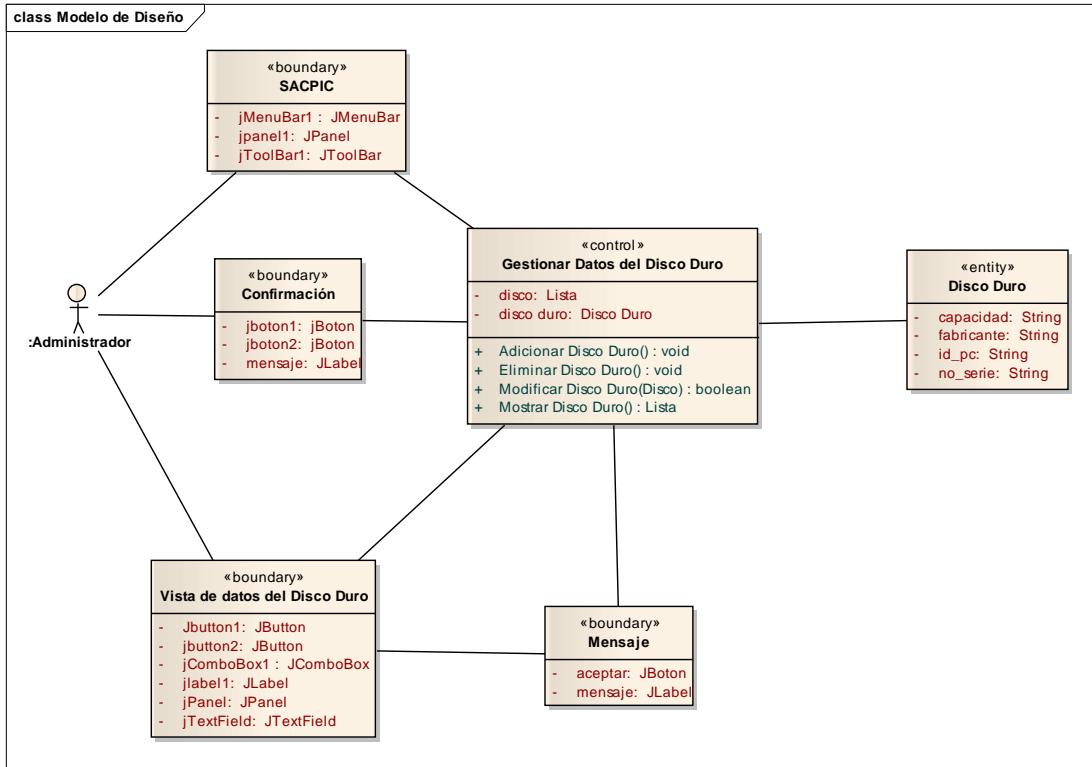


Figura 7.1. Clases del Diseño.

OPINIÓN DEL TUTOR.

El Trabajo de Diploma titulado “Sistema Automatizado para el Control de Piezas de Computadora (SACPIC)”, del estudiante José María Otacoma Jacho, constituye una propuesta muy completa de aplicación informática con una Base de Datos incluida que gestiona la información relacionada con los datos de las piezas de las computadoras, brindando un mayor control de la información al personal dedicado a esta tarea.

El software en concreto está compuesto por una base de datos realizada en PostgreSQL y una interfaz gráfica de usuario programada en el lenguaje de programación Java. En la base de datos se encuentran almacenados tanto los datos correspondientes a la facultad, los locales, así como las computadoras ubicadas en cada uno de ellos y las piezas de cada Computadora, entre otras informaciones necesarias. La interfaz gráfica es una aplicación de escritorio mediante la cual el usuario puede gestionar toda la información residente en la base de datos. Algunas de sus características son la inserción, modificación y eliminación de objetos; permite además la búsqueda de información de dichos datos ; también incluye la muestra de reportes, los cuales pueden ser impresos directamente desde la aplicación o almacenados en ficheros de extensión *.xls, y una ayuda, la cual brinda apoyo al trabajo con dicha interfaz. Por tanto, queda explícita la correspondencia del contenido del trabajo con los objetivos planteados.

En particular, el estudiante tuvo que perfeccionar su habilidad en el trabajo con un lenguaje de programación no habitual para él, en este caso, Java, y un gestor de base de datos también nuevo (PostgreSQL).

Durante el desarrollo del software tuvo que resolver diferentes problemas y retos que los requerimientos planteaban. Para realizar el trabajo, el estudiante no sólo tuvo que manejar conocimientos de diversas disciplinas del área de la informática, tales como Base de Datos, Programación e Ingeniería de Software, sino también de la aplicación en cuestión. Fue cumplimentado en tiempo y forma.

Finalmente, es necesario afirmar que el desarrollo de este Trabajo de Diploma por parte del estudiante José María Otacoma Jacho le ha permitido demostrar haber adquirido los conocimientos, las habilidades y los valores que a los profesionales de su especialidad la sociedad exige.

Ing. Maite de la Caridad Hernández Soberao.