

*Universidad de Pinar del Río*  
*“Hermanos Saíz Montes de Oca”*



**Facultad de Informática**  
**Departamento de Informática**

*PROYECTO DE GRADO*

**Título: “Sistema para la Gestión de los Documentos de Tesis”.**

**SisGeTe**

*Autor:* Ángel Israel Bustillos Toaquiza.

*Tutor:* MSc. Manuel Jesús López Vázquez.

*-Pinar del Río, 2008-*

## DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de esta tesis y que autorizo al Departamento de Informática de la Universidad de Pinar del Río “Hermanos Saíz Montes de Oca” para que hagan el uso que estimen pertinente con este trabajo.

---

Angel Israel Bustillos Toaquiza

Autor

---

Msc. Manuel J. López Vázquez

Tutor

# *DEDICATORIA*

# *DEDICATORIA*

*A mis amados padres, hermanos, familiares y amigos por todo su apoyo, comprensión y amor que me han brindado en todo este tiempo. Además a mis queridos enemigos.*

*Gracias*

*Angel Israel Bustillos Toaquiza*

# *AGRADECIMIENTOS*

## AGRADECIMIENTOS

*A Dios por darme salud y vida, por guiarme en mi caminar para llegar a cumplir mi gran sueño de llegar a ser un profesional.*

*A Rosa, Alfonso mis amados padres, por su amor, confianza y esfuerzo que depositaron en mi, gracias por su apoyo incondicional tanto económicamente como emocionalmente que me han brindado durante todo este largo caminar hasta llegar a concluir mi Carrera Universitaria.*

*A mis adorables hermanos, Adriana, Mayra, Adrián y Paola, por todas las palabras de ánimo que me llenaron de mucha emoción, valentía y coraje para seguir en este trajín de la vida.*

*A mis abuelitos, tíos (as), primos (as), amigos (as) los cuales me han apoyado de la mejor manera que es emocionalmente.*

*A mis panas del Alma, Paúl, Fernando, Mauro, Pablo, Vinicio, José, con quienes compartí mi vida universitaria gracias por su paciencia hermanos.*

*A mi tutor Msc. Manuel J. López Vázquez, quien me supo brindar su amistad y conocimientos profesionales para el desarrollo de mi trabajo de Diploma.*

*A la Universidad Técnica de Cotopaxi, por llenarme de conocimientos y permitirme crecer éticamente y profesionalmente. Y la Universidad de Pinar del Río, Hnos. Saíz Montes de Oca, por abrirme las puertas para llegar a concluir mi Carrera, en la elaboración de la tesis de diploma.*

*En fin a todas aquellas personas que contribuyeron con un granito de arena, para guiarme en este trayecto de mi vida.*

# *RESUMEN*

**Autor:** Angel Israel Bustillos Toaquiza

**e-mail:** [Israel\\_2t@yahoo.es](mailto:Israel_2t@yahoo.es)

**Tutor:** Msc. Manuel López Vázquez

**Centro de Trabajo:** Departamento de Informática. Universidad de Pinar del Río.

**e-mail:** [malova@infomail.upr.edu.cu](mailto:malova@infomail.upr.edu.cu)

El presente trabajo surge ante la necesidad del Departamento de Informática de tener un mejor registro y manejo de los trabajos de Tesis de Diploma y Maestría que realizan los estudiantes de la Carrera de Informática y maestrantes. Estos trabajos de Tesis son registrados y consultados en forma manual y guardados su documento en formato pdf, provocando así un inadecuado registro de la información, su pérdida e ineficiencia en su tratamiento al realizar búsquedas y consultas.

Como solución al problema se creó un Sistema automatizado (SisGeTe) capaz de gestionar una Base de Datos normalizada que contiene de los trabajos de Diplomas y Tesis la información necesaria para lograr la eficiencia adecuada en el registro, búsquedas y consultas.

El Sistema creado fue desarrollado tomando como gestor MSAccess y como lenguaje de programación Visual Basic para Aplicaciones empleando como entorno de desarrollo el mismo ofrecido por el gestor de Base de Datos elegido.

# *SUMMARY*

**SUMMARY**

The present work arises in the face of the necessity of Computer science's Department of having a better registration and handling of the works of Thesis of Diploma and Master that the students of Computer science's Career carry out. These works of Thesis are registered and consulted in form manual and kept their document in format pdf, causing this way an inadequate registration of the information, their lost one and in efficiency in their treatment when carrying out searches and consultations.

As solution to the problem one believes an automated System (SisGeTe) able to negotiate a normalized Database that it contains of the works of Diplomas and Thesis the necessary information to achieve the efficiency adapted in the registration, searches and consultations.

The created System was developed taking as agent MSAccess and I eat language of Visual programming Basic for Applications using like development environment the same one offered by the elected agent of Database.

# *ÍNDICE*

**INDICE**

INTRODUCCIÓN.....	2
INTRODUCCIÓN.....	7
I.1.- EL DEPARTAMENTO DE INFORMÁTICA DE LA UPR.....	9
I.2.- LA GESTIÓN DE LAS TESIS DE DIPLOMAS Y DE MAESTRÍA EN EL DEPARTAMENTO DE INFORMÁTICA.....	12
I.3.- ESTIMACIÓN DEL COSTO DE LA PROPUESTA DE SOLUCIÓN.....	16
INTRODUCCION.....	24
II.1.- EL MODELO DE LOS DATOS.....	26
II.2.- CONCEPTOS DE BASE DE DATOS UTILIZADOS.....	26
II.3.- CARACTERIZACIÓN DEL GESTOR DE BASE DE DATOS UTILIZADO.....	33
III.4.- IMPLEMENTACIÓN DEL MODELO DE DATOS.....	39
INTRODUCCION.....	40
III.1.- EL DISEÑO DE LAS INTERFASES DE USUARIOS.....	42
III.1.1.- EL LENGUAJE UNIFICADO DE MODELADO (UML).....	52
III.1.2.- CARACTERIZACIÓN DE LA HERRAMIENTA CASE UTILIZADA.....	57
III. 2.- DESARROLLO DE SISGETE.....	60
III.2.1.- EL LEGUAJE DE PROGRAMACIÓN “VISUAL BASIC PARA APLICACIONES”.....	61
III.2.2.- EL ENTORNO DE DESARROLLO DE MSACCESS.....	64
INTRODUCCION.....	67
IV.1.- LA SEGURIDAD DE SISGETE Y SU IMPLEMENTACIÓN.....	69
IV.2.- HERRAMIENTAS DEL SISTEMA.....	80
TABLAS DE SISGETE Y SU ESTRUCTURA.....	88

# INTRODUCCIÓN

**INTRODUCCIÓN**

La Educación Superior en Cuba tiene como misión principal la formación de profesionales con alta capacidad técnica y una excelente preparación moral, que respondan de manera incondicional a los intereses de la revolución. El Ministerio de Educación Superior (MES) de Cuba tiene una estructura con la cual garantiza el logro de estos objetivos, teniendo para ello diversos Centros de Educación Superior (CES) que cubren todo el país.

Los CES permiten a todas las personas interesadas, en obtener un nivel de educación superior poderlo lograr. Estas posibilidades se han incrementado con la universalización de la enseñanza universitaria en los municipios de cada provincia mediante las Sedes Universitarias Municipales (SUM) subordinadas a los CES. Uno de estos CES es la Universidad Pinar del Río (UPR), la misma se compone de seis Facultades: Facultad de Agronomía de Montaña, Facultad de Ciencias Económicas, Facultad de Ciencias Sociales y Humanísticas, Facultad de Forestal y Agronomía, Facultad de Geología–Mecánica y la Facultad de Informática Y Telecomunicaciones. A su vez estas facultades tienen subordinados Departamentos, en los cuales se materializa la formación instructiva-educativa del futuro profesional. Para culminar los estudiantes sus estudios universitarios realizan una tesis de Diploma la que es defendida ante un tribunal. A su vez estos departamentos también responden por la superior de los profesionales de la provincia mediante la educación de postgrado ya sea en: cursos, diplomados, maestrías y doctorados. Para obtener los grados de master o de doctor los aspirantes presentan también una tesis. Tanto las tesis de Diplomas como las de maestría y doctorados son necesarias tenerlas recogidas de forma que faciliten su consulta por los interesados, para lo cual es ineludible poder distinguir de estos aspectos como: su temática, su campo de aplicación, resumen, etc., que permita rápidamente su localización y valoración. Esta necesidad se incrementa a medida que por año académico crece el volumen de tesis defendidas.

En el caso del Departamento de Informática la gestión del registro y consulta de estas tesis se realiza de forma manual, bien por los documentos impresos o consultando los correspondientes ficheros, en formato pdf, de estos documentos. Lo anterior provoca:

- ❖ Registro inadecuado de la información que no facilita el proceso de consulta al no tenerse guardada en una Base de Datos.
- ❖ Empleo grande de tiempo para buscar los documentos de interés.
- ❖ Difícil búsqueda y consulta de la información.
- ❖ No seguridad de la información, al estar esta esparcida por diferentes soportes magnéticos (CD).

Teniendo como problema: “La ineficiente gestión de los documentos de tesis”. Esto hizo que el objeto de investigación fuese: “La gestión de los documentos de tesis del Departamento de Informática de la UPR”. El autor se planteo que de poseer una Base de Datos capaz de recoger la información deseada de los documentos de tesis y de un software que la gestione entonces se podrá realizar de forma eficiente la búsqueda y consultas deseadas de estos documentos.

El objetivo de la investigación se dirigió a Diseñar e Implementar una Base de Datos que guarde la información necesaria de los documentos de tesis realizados en el Departamento de Informática de la UPR, para lo que se empleó el gestor de Base de Datos MSAccess, acompañado de un sistema capaz de gestionar esta Base de Datos, el que se desarrollo en el lenguaje Visual Basic para Aplicaciones, utilizando el entorno de desarrollo brindado por el gestor escogido.

A continuación se muestra como se organizó la investigación para cumplir con los objetivos trazados. Esta organización se corresponde con los capítulos en que se estructuró el presente documento.

**Capítulo I.- “Caracterización de la gestión de las Tesis en el Departamento de Informática”.**

En el capítulo se realiza una caracterización del objeto de estudio “La Gestión de las Tesis de Diploma y de Maestría en el Departamento de Informática de la UPR”. Se comienza definiendo el rol social de este departamento, su ubicación en el organigrama de la UPR y las principales actividades por él realizadas.

Se muestra un Modelo del Dominio donde es reflejado los objetos necesarios a utilizar para poder realizar la gestión de los documentos de tesis así como la relación existente entre estos objetos.

El capítulo concluye con una estimación del costo aproximado para desarrollar un software capaz de gestionar estos documentos y cumplir además con las exigencias de los usuarios.

**Capítulo II.- “Análisis de los Datos del Sistema y su Implementación”.**

Para desarrollar un Sistema es necesario analizar:

- Los Datos.
- Sus funciones.
- Seguridad.

En el capítulo se realiza el primero de estos análisis partiendo de la caracterización realizada del objeto de investigación en el capítulo previo, se obtiene el Modelo de los Datos en el cual se mostrada las entidades de interés, las relaciones existentes entre estas y los elementos de datos considerados en cada entidad. Son presentados los conceptos más importantes utilizados sobre Base de Datos para obtener el Modelo de Datos concluyendo con una caracterización de los Sistemas de Gestión de Base de Datos (SGBD) particularizando al caso de MSAccess, SGBD elegido para implementar el Modelo de Datos, exponiendo las razones tenidas para su elección.

**Capítulo III.- “Análisis de las Funcionalidades del Sistema y su desarrollo”.**

El capítulo aborda el análisis funcional del sistema comenzando este por la captura de los requerimientos que el sistema (SisGeTe) debe cumplir. Se determina los actores (usuarios que interactúan con el Sistema para obtener un beneficio de este), sus roles o papeles que estos desempeñan ante el Sistema y los Casos de Uso (funcionalidades brindadas por el Sistema a sus usuarios) con que estos se relacionan. En el se caracteriza el Lenguaje Unificado de Modelado (UML) del que se empleo los artefactos de: Actor, Caso de Uso y Diagramas de Casos de Uso y del CASE Power Designer, herramienta empleada para confeccionar estos artefactos. Se muestra el diseño de las interfases usuarios de algunos Casos de Uso, así como las reglas del negocio a tener en cuenta en el funcionamiento de estas. Por ultimo es caracterizado el Lenguaje de Programación Visual Basic para Aplicaciones, lenguaje embebido en todas las aplicaciones de Office (incluyendo MSAccess), con el que se implemento las Interfases de Usuarios diseñadas, caracterizando el entorno de desarrollo empleado para crear las interfases de usuarios de SisGeTe diseñadas.

**Capítulo IV.- “Seguridad del Sistema y sus herramientas”.**

En el capítulo se presenta el diseño de la seguridad de la Base de Datos del sistema, la que se implemento usando el gestor MSAccess. Este diseño muestra los grupos de usuarios y cuentas considerados y los derechos que fueron otorgados a estos. A nivel de la interfaz usuario es señalado las funcionalidades del sistema que las que podrán acceder cada usuario teniéndose también una seguridad a nivel de software al limitarse las funcionalidades brindadas según el usuario.

El capítulo concluye con la descripción de las herramientas con que dispone el Sistema para resguardar, compactar y reparar los datos y mantener actualizado el registro de cuentas de usuario y contraseñas.

Los métodos de investigación científica utilizados durante el desarrollo de la investigación fueron:

- 1. Histórico–Lógico.-** Empleados en la revisión de los documentos del departamento de Informática de la UPR relacionados con la actividad de la gestión de tesis.
- 2. Análisis–Síntesis.** Empleados en los procesos de captura de los requerimientos y del análisis y desarrollo del Sistema y Base de Datos.
- 3. Inductivo–Deductivo.-** Empleados en los procesos de captura de los requerimientos y del análisis y desarrollo del Sistema.
- 4. Modelación.-** Aplicado en la obtención del Modelo de Datos y de la Interfaz-Usuario del Sistema.
- 5. Entrevista.-** Aplicado en la captura de los requerimientos a cumplir por el software.

# CAPÍTULO I

## *Caracterización de la gestión de las Tesis en el Departamento de Informática*

**INTRODUCCIÓN**

En el capítulo se realiza una caracterización del objeto de estudio “**Las Tesis de Diploma y de Maestría del Departamento de informática de la UPR**”, realizadas por sus egresados. Se comienza definiendo el rol social de este departamento, su ubicación en el organigrama de la UPR y las principales actividades por él realizadas.

Se hace un análisis de cómo se gestiona los Documentos de Tesis con vista a consultarlos para la realización de investigaciones y de otros documentos de diplomas y de maestrías, señalando las dificultades e insuficiencias de esta gestión. Se muestra un Modelo del Dominio donde es reflejado los objetos necesarios a utilizar para poder realizar la gestión así como la relación existente entre estos.

El capítulo concluye con una estimación del costo aproximado para desarrollar un software capaz de gestionar estos documentos y cumplir además con las exigencias de los usuarios. Esta estimación se realiza usando el Modelo de Diseño Temprano de COCOMO II. (**C**onstructive **C**oste **M**odel) empleando la métrica de los puntos de función y el software “CoCoMo “ de la Universidad de Carolina para realizar los cálculos del modelo.

**I.1.- El Departamento de Informática de la UPR.**

El actual Departamento de Informática, surgió en el año 1987, asociado al Departamento de Matemática subordinado entonces a la entonces Facultad de Economía, no es hasta 1990 que se independiza como Departamento de Servicios a las carreras entonces existentes en la UPR pasando a subordinándose a la Vice Rectoría de Investigaciones y Postgrado. En el año 2000 pasa a ser un Departamento de Servicios de la Facultad de Ciencias Técnicas (actualmente no existente). En ese año se comienza a preparar el expediente para solicitar al Ministerio de Educación Superior (MES) la apertura de la Carrera de Ingeniería en Informática en la UPR la que comienza en el curso 2001-2002 creándose la Facultad de Informática y Telecomunicaciones a la cual es subordina el Departamento con el nombre de Informática pasando a ser un Departamento Carrera teniendo la responsabilidad de la carrera creada además de seguir brindándole servicios a las demás carreras.

Desde su creación este departamento ha tenido la responsabilidad de orientar y dirigir la enseñanza de Computación en las Carreras de la UPR, tanto de los cursos regulares, como los de para trabajadores y los relacionados con la universalización de la enseñanza universitaria en los municipios, así como la superación de postgrado en esta temática de la propia Universidad y del territorio. A lo anterior se le añade, en el curso 2001-2002, la de responder por la formación de los futuros Ingenieros en Informática formados en la UPR.

La misión del mismo es la de Contribuir al desarrollo y aplicación de las tecnologías de la información y las comunicaciones en el territorio mediante la formación integral de profesionales de la Informática, así como su superación postgraduada y al avance de la ciencia y la técnica en correspondencia con las necesidades y expectativas de la informatización de la Sociedad Pinareña y nuestro Estado Revolucionario.

La ubicación actual de este departamento en la estructura de la UPR puede ser vista en la figura I.1.1 tomada del Proyecto de Diploma de Santiago Javier Ortiz

y de Juan Carlos Lezcano [ORTIZ, 2007]. En este Diploma se señala los departamentos subordinados a cada una de las facultades existentes en la UPR teniéndose por facultades:

**FACULTAD DE AGRONOMÍA DE MONTAÑA.**

- Departamento Carrera de Agronomía de Montaña

**FACULTAD DE CIENCIAS ECONÓMICAS**

- Departamento Carrera de Contabilidad y Finanzas
- Departamento Carrera de Economía.

**FACULTAD DE CIENCIAS SOCIALES Y HUMANÍSTICAS.**

- Departamento Servicios de Idiomas
- Departamento Servicios de Marxismo Leninismo
- Departamento Carrera de Humanidades
- Departamento Carrera de Derecho

**FACULTAD DE FORESTAL Y AGRONOMIA**

- Departamento Servicios de Biología
- Departamento Servicios de Química
- Departamento Carrera de Agronomía
- Departamento Carrera de Forestal.

**FACULTAD DE GEOLOGIA – MECANICA**

- Departamento Servicios de Física.
- Departamento Carrera de Geología
- Departamento Carrera de Mecánica.

## FACULTAD DE INFORMATICA Y TELECOMUNICACIONES

- Departamento Servicios de Matemática.
- **Departamento Carrera de Informática.**
- Departamento Carrera de Telecomunicaciones.

Destaquemos que hay Departamentos Carreras que a su vez brindan servicios a otras carreras, y en carreras presentes en los Municipios.

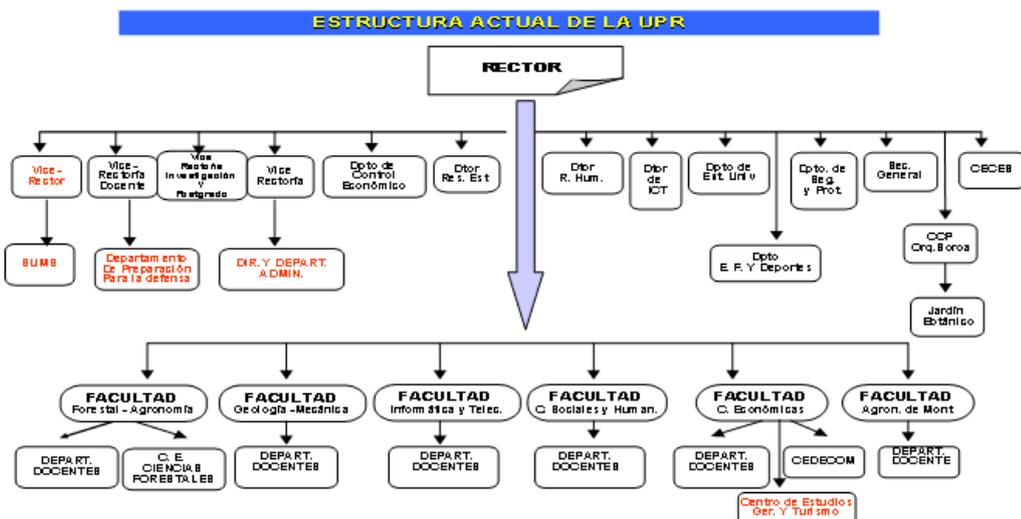
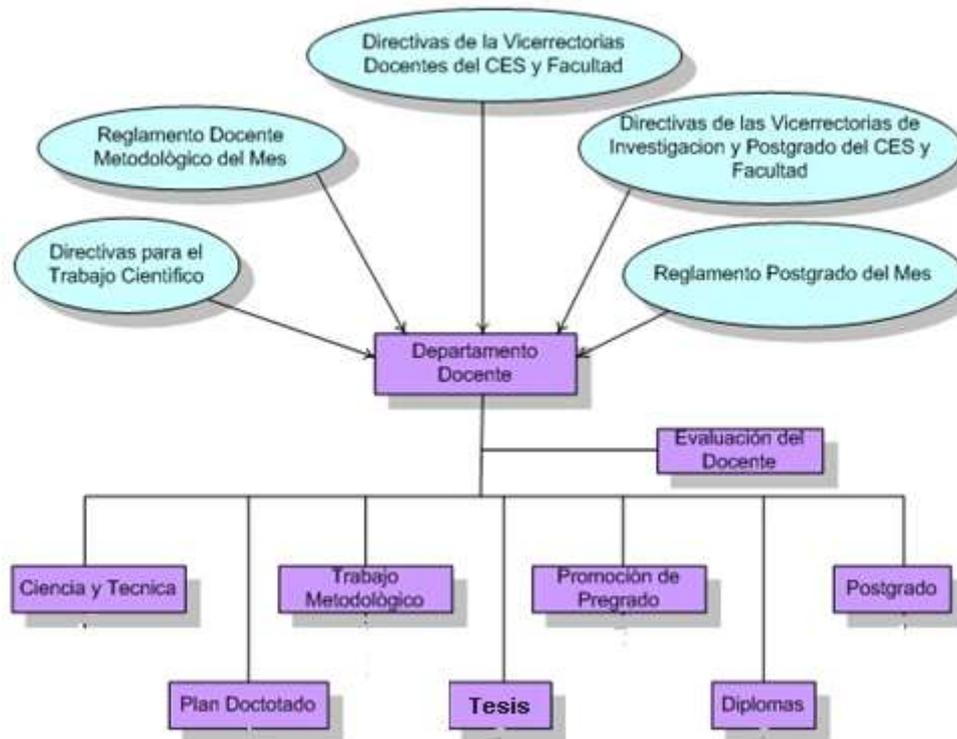


Figura I.1.1.- Estructura de la UPR.

Las actividades realizadas por los Departamentos Carreras durante el desempeño de sus funciones pueden ser resumidas en el gráfico presentado en la figura I.1.2 el que fue tomada de la Proyecto de Diploma de Ortiz y Lezcano [ORTIZ, 2007].



**Figura. I.1.2.- Actividades realizadas por los Departamentos Carreras.**

De estas actividades nuestro interés es el análisis de cómo se hace la gestión de las Tesis (maestrías y doctorados) y los diplomas por el Departamento de Informática de la Universidad de Pinar del Río.

### **I.2.- La gestión de las Tesis de Diplomas y de Maestría en el Departamento de Informática.**

En el Departamento Carrera de Informática anualmente se realizan alrededor de 30 tesis de Diplomas correspondientes a los alumnos que culminan sus estudios en la carrera de Ingeniería en Informática. La primera promoción de esta carrera se realizó en el curso 2005-2006, existiendo por ende, sin tener en cuenta los que egresaran de este curso 2007-2008, dos promociones, es decir alrededor de 60 diplomas.

Además de los diplomas de los estudiantes cubanos, desde el curso 2004-2005 se ha atendido por el departamento estudiantes de Universidades extranjeras con vista a que culminen sus estudios con el trabajo de diploma, teniéndose en hasta el presente alrededor de 40 diplomas defendidos.

En relación con las tesis de maestría el Departamento de Informática realizó una maestría de Informática Aplicada (dos versiones) de las que concluyeron alrededor de 10 maestrantes en cada una, así como una masiva de Nuevas Tecnologías para la Educación (NTE) con 90 maestrantes defendidos aproximadamente, de la cual se comenzará una nueva versión.

Los documentos de diplomas y tesis son guardados en el departamento, en formato digital (pdf), sin tenerse un registro en una Base de Datos. El documento pdf contiene como información:

- Título del Diploma.
- Datos de los autores.
- Datos de los tutores.
- Resumen.
- Palabras claves (no existente en todos los casos).
- Introducción.
- Conclusiones.
- Recomendaciones.
- Referencias Bibliográfica.

Además de contener el desarrollo del trabajo realizado. Sin embargo estos documentos no se encuentran clasificados por temática, rama de aplicación, etc.

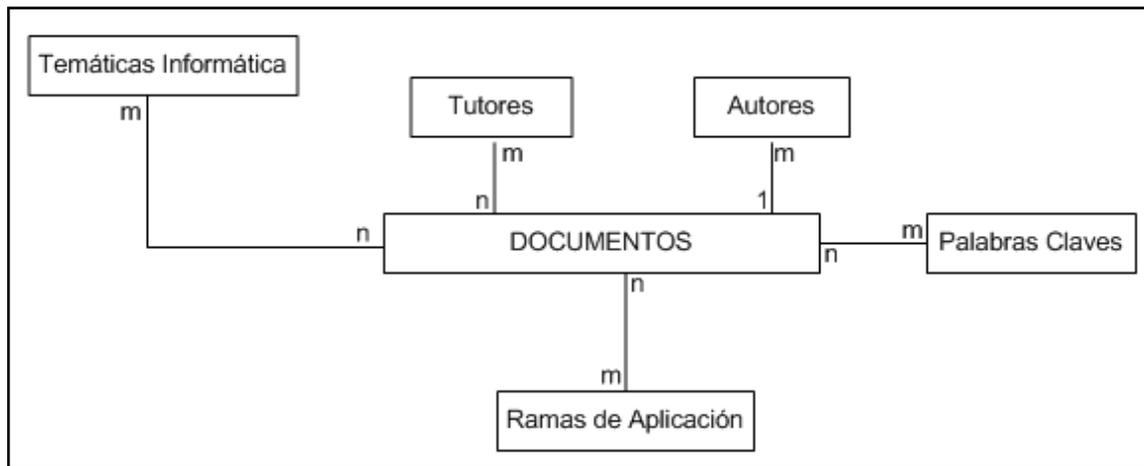
Al solo tenerse en el departamento el documento digital (pdf) de los diplomas y tesis se hace engorrosa la consulta de estos para tomarlos como referencia al realizar trabajos de investigación o de nuevas tesis. Se infiere de lo planteado que la gestión de estos documentos es ineficiente presentándose problemas como:

- No poseerse un registro, en una Base de Datos, de los Diplomas y Tesis realizados, solamente la existencia de los documentos digitalizados en formato pdf en un CD.
- No poder filtrar los documentos a consultar según temática y/o palabras claves u otro criterio que facilite su localización.
- Pérdida de tiempo al tener que revisar uno a uno los documentos digitalizados, lo cual se irá incrementando por año al irse aumentando el volumen de diplomas y de tesis defendidos.

En resumen existe una inadecuada gestión de los diplomas y tesis en el departamento de informática de la UPR, no existiendo una Base de Datos que recoja los elementos necesarios de estos documentos que permita su gestión eficiente.

Para dar solución al problema presentado el autor del trabajo se propuso como objetivo **“Crear un Sistema automatizado que permita gestionar los Diplomas y Tesis del Departamento de Informática con uso de una Base de Datos donde se recoja los datos de interés de estos documentos que facilite su búsqueda y consulta”**

Los objetos necesarios manejar para realizar la gestión de los Documentos de Diploma y de Tesis son mostrados en la figura I.2.1. En la misma puede ver la relación existente entre estos objetos. Este Modelo del Dominio servirá de base en la obtención del Modelo de Datos, aspecto tratado en el Capítulo II.



**Figura I.2.1- Modelo del Dominio.**

Los objetos considerados en el Modelo de la figura I.2.1 son:

- **Documentos.-** Son documentos producto de una investigación. Estos puede ser una tesis de maestría o de Diploma, siendo de interés guardar: Título, mes y año de defensa, tipo de documento (Diploma o Tesis), localización digital del documento (pdf), del resumen, de la introducción, de las conclusiones y de las recomendaciones..
- **Autores.-** Son los estudiantes de pregrado o de posgrado que confeccionaron el documento. De ellos es de interés conocer: Nombre y dos apellidos. En el caso de los maestrantes es necesario conocer además: centro laboral donde se desempeñan como profesionales y el correo electrónico.
- **Tutores.-** Son los profesionales que dirigieron el desarrollo de la investigación realizada. De ellos se desea conocer: Nombre y dos apellidos, centro donde se desempeñan como profesionales, dirección de correo electrónico, si es profesor del departamento y la categoría de investigativa que poseen.

- **Temáticas.-** Son diferentes temáticas consideradas para clasificar los documentos de tesis y de diplomas. De ellas se conoce el nombre de la temática.
- **Ramas de Aplicaciones.-** Son diferentes ramas de aplicación consideradas para clasificar los documentos de tesis y de diplomas. De ellas es conocido el nombre de la rama.
- **Palabras Claves.-** Son aquellas palabras consideradas para clasificar los documentos de tesis y de diplomas.

### **I.3.- Estimación del costo de la propuesta de solución**

Antes de desarrollar el Sistema se realizó una valoración aproximada de su costo y tiempo de desarrollo con uso del Modelo de Diseño Temprano de COCOMO II (**C**onstructive **C**ost **M**odel), comparando el mismo con los beneficios obtenidos al emplear el sistema para determinar la decisión a tomar.

Al realizar esta estimación se tuvo en cuenta lo señalado en [**COCOMO, 1999**] y [**BOHEM, 2000**].

**Entradas Externas (EI):** Entradas que proporciona al software diferentes datos orientados a la aplicación.

<b>Nombre</b>	<b>Cantidad de ficheros</b>	<b>Cantidad de Elementos de datos</b>	<b>Complejidad</b>
Registrar Tesis	10	13	Alta

**Tabla I.3.1.- Entradas Externas de los Trabajos Diploma**

**Salidas Externas (EO):** Salidas que proporciona al usuario información orientada al tema de aplicación. En este contexto la “salida” se refiere a informes, resúmenes, etc.

Nombre	Cantidad de ficheros	Cantidad de Elementos de datos	Complejidad
Consultar Tesis	10	20	Alto

**Tabla I.3.2.- Salidas Externas de los Trabajos Diploma**

**Peticiones (EQ):** Entradas interactivas que resultan de la generación de algún tipo de respuesta en forma de salida interactiva.

Nombre	Cantidad de ficheros	Cantidad de Elementos de datos	Complejidad
Autenticarse	0	2	Baja
Cambiar Contraseña	0	2	Baja

**Tabla I.3.3.- Peticiones de los Trabajos Diploma**

**Ficheros internos (ILF):** son archivos (tablas) maestros lógicos (o sea una agrupación lógica de datos que puede ser una parte de una base de datos o un archivo independiente).

Nombre	Cantidad de Record	Cantidad de Elementos de datos	Complejidad
Tesis	200	9	Medio
Autores	230	3	Medio
Autores Maestría y Doctorados	160	3	Medio
Tutores	230	3	Medio
Tutorías	250	2	Medio
Ramas Aplicación X Tesis	600	2	Medio
Palabras Claves X Tesis	600	2	Medio
Temáticas X Tesis	600	2	Medio
Ramas de Aplicación	50	1	Medio
Temáticas	20	1	Medio
Palabras Claves	800	1	Medio

**Tabla I.3.4.- Ficheros Internos (tablas) de los Trabajos Diploma**

Según los datos anteriores se obtuvieron, mediante la métrica de los puntos de función, las líneas de código fuentes mostradas en la parte inferior de la figura I.3.1.

Function Type	# of Function Points			SubTotal
	Low	Average	High	
Internal Logical Files	0	11	0	110
External Interface Files	0	0	0	0
External Inputs	0	0	1	6
External Outputs	0	0	1	7
External Inquiries	2	0	0	6
Total Unadjusted Function Points				129
Equivalent Total in SLOC				4902

**Figura I.3.1.- Líneas de código.**

Se consideró como Lenguaje de Programación “Visual Basic para Aplicaciones” (Access) tomándose como promedio 38 líneas código en este lenguaje por punto de función, obteniéndose así **4,902** instrucciones (líneas de código) fuentes para construir un total de **129** puntos de función.

Los valores considerados de los Multiplicadores de esfuerzo (EM) para el Modelo de Diseño Temprano fueron:

Factores	Valor	Justificación
RCPX	Baja	Base de Datos complejidad media.
RUSE	Baja	El nivel de reutilizabilidad es a través del programa.
PDIF	Bajo	El tiempo y la memoria estimada para el proyecto son de baja complejidad.
PREX	Normal	Los especialistas tienen cierta experiencia en el uso de las de tecnologías.
FCIL	Normal	Se han utilizado herramientas de alto nivel de desarrollo como el Visual Basic para Aplicaciones, entorno de desarrollo de MSAccess, CASE Power Designer
SCED	Normal	Los requerimientos de cumplimiento de cronograma son normales.
PERS	Normal	La experiencia del personal de desarrollo es normal, tienen una buena capacidad.

Tabla I.3.5.- Valores de los EM

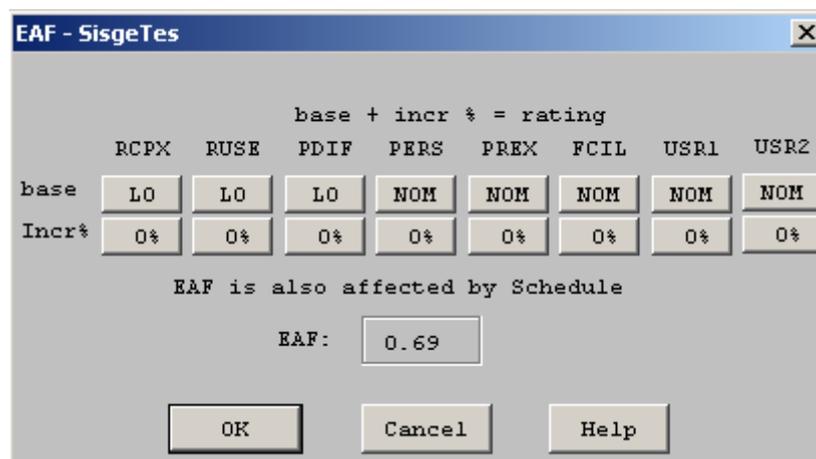
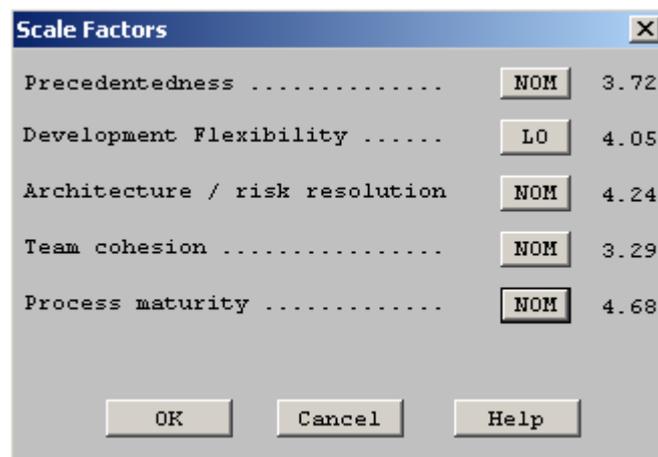


Figura I.3.2.- Valores de EM introducidos en el software.

Los valores considerados de los **Factores de Escala (SF)** fueron:

<b>Factores</b>	<b>Valor</b>	<b>Justificación</b>
PREC	Normal	Se posee una comprensión de los objetivos del producto, no se tiene experiencia en realizar de software de este tipo.
FLEX	Baja	Debe haber considerable cumplimiento de los requerimientos del sistema.
TEAM	Normal	El equipo que va desarrollar el software es cooperativo.
RESL	Normal	Se cuenta con un plan definido.
PMAT	Medio	Se encuentra en el nivel 3 (medio).

**Tabla I.3.6.- Valores de los SF**



**Figura I.3.3: Factores de Escala introducidos en el software.**

Se consideró un salario promedio de \$ 300 obteniendo los resultados reflejados en la Figura I.3.4.

The screenshot shows the USC-COCOMO II.2000.0 software interface. The project name is 'SisgeTes'. The development model is 'Early Design'. The main table displays the following data:

X	Module Name	Module Size	LABOR Rate (\$/month)	EAF	Language	NOM Effort DEV	EST Effort DEV	PROD	COST	INST COST	Staff	RISK
	SisgeTes	F:4902	300.00	0.69	Access	17.2	11.8	416.4	3531.55	0.7	1.5	0.0

Summary statistics for the project:

Estimated	Effort	Sched	PROD	COST	INST	Staff	RISK
Optimistic	7.9	7.1	621.5	2366.14	0.5	1.1	
Most Likely	11.8	8.1	416.4	3531.55	0.7	1.5	0.0
Pessimistic	17.7	9.2	277.6	5297.33	1.1	1.9	

Additional information: Total Lines of Code: 4902.

**Figura I.3.4.- Resultados de los Cálculos de COCOMO II mediante el software.**

Utilizando los valores mostrados en la parte inferior de la figura I.3.4 se obtiene:

### Esfuerzo (DM).

$$DM = (\text{Valor Optimista} + 4 \times (\text{Valor Esperado}) + \text{Valor Pesimista}) / 6$$

$$DM = (7.9 + 4 * 11.0 + 17.7) / 6 = \mathbf{24.76 = 25 \text{ Hombres/Mes}}$$

### Tiempo (TDev).

$$TDev = (\text{Valor Opt} + 4 \times (\text{Valor Esperado}) + \text{Valor Pesimista}) / 6$$

$$TDev = (7.1 + 4 * 8.1 + 9.2) / 6 = \mathbf{16.51 = 17 \text{ Meses.}}$$

### Cantidad de hombres (CH):

$$CH = DM / TDev$$

$$CH = 25 / 17$$

$$CH = \mathbf{1.47 = 2 \text{ hombres}}$$

### Costo de la fuerza de trabajo

$$CFT = (\text{Valor Optimista} + 4 \times (\text{Valor Esperado}) + \text{Valor Pesimista}) / 6$$

$$CFT = (2366.14 + 4 * 3531.55 + 5297.33) / 6 = \mathbf{\$3,631.61}$$

Agregándole a este el **Costo de los Medios Técnicos**, compuesto este por los costos de depreciación (el que no fue considerado), de mantenimiento y de gasto de energía, y el **Costo en Gasto en Materiales**, el que no fue significativo, se obtiene como costo total del proyecto:

$$\text{CTP} = \$ 4,000.00$$

El software está dirigido a gestionar las Tesis de Diploma y de Maestría”, realizadas por egresados del departamento de Informática de la Universidad de Pinar del Río (UPR). Esta aplicación constituye una nueva herramienta que no solo contribuye a gestionar la información antes señalada sino también facilita todo el proceso de actualizar los datos dentro del sistema propiciando aumentar la eficiencia en la gestión contribuyendo con ello a mejorar el control de la actividad así como ahorrar un considerable tiempo. **SisGeTe** aporta los s beneficios de:

- Poder registrar, de manera eficiente, los documentos de tesis de diploma y maestría manejadas por el Departamento de Informática.
- Poder consultar con facilidad y rapidez los documentos de tesis de Diploma y de Maestría según diversos criterios de búsqueda.
- Proteger y guardar de forma eficiente toda la información de las Tesis de Diplomas y Maestría.

Por el peso que tienen los beneficios reportados al emplear el software, además de tener en cuenta que su desarrollo se realizara como un trabajo de Diploma, lo que evita tener que desembolsar su costo, se tomo la decisión de realizar su diseño y desarrollo.

En el diseño y desarrollo del sistema **SisGeTe** fue empleado:

### Recursos Humanos:

- ❖ Dos personas para el análisis, diseño y desarrollo del sistema:

**Tutor:** Msc. Manuel Jesús López Vázquez.

**Autor:** Angel Israel Bustillos Toaquiza

**Recursos Técnicos:**❖ **Hardware** para su diseño y desarrollo:

Procesador	Pentium Dual - Core 1.73 GHz
Memoria	512 MB
Disco Duro	80 GB 7200 RPM
Unidad de Respaldo	DVD / CD-ROM
Monitor	Resolución SVGA (1024 x 768) píxeles

❖ **Software:**

Sistema Windows Xp.

Microsoft Access 2003.

Power Designer.

USC COCOMO II.

## *CAPÍTULO II*

*Análisis de los Datos del Sistema y su Implementación”.*

*Para desarrollar un Sistema es necesario analizar*

**INTRODUCCION**

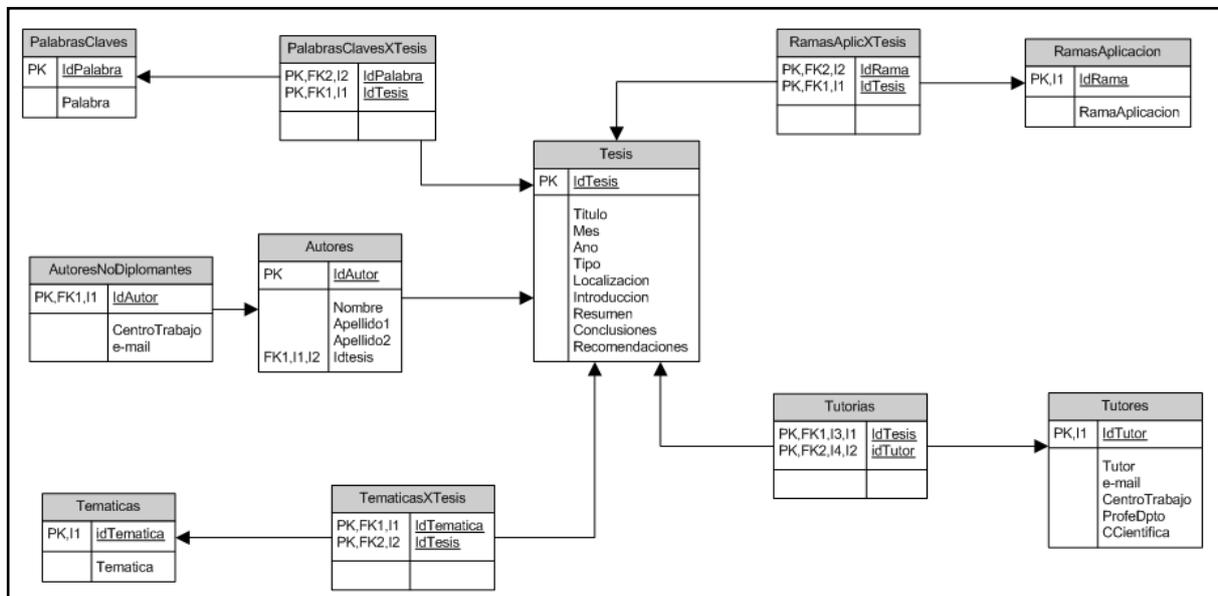
Un componente importante al abordar el desarrollo de un software es el análisis de datos, estando este muy relacionado con el funcional. En el presente capítulo se realiza el primero de estos análisis partiendo del Modelo del Dominio presentado en el capítulo precedente.

En el primer epígrafe se establece el Modelo de Datos utilizado para gestionar los documentos de tesis, se parte de la caracterización realizada, en el primer capítulo, del objeto de investigación. En este Modelo puede verse las entidades de interés, las relaciones existentes entre ellas y los elementos de datos considerados de cada una.

El segundo epígrafe presenta los conceptos más importantes sobre Base de Datos empleados para obtener el Modelo de Datos del epígrafe anterior y poder implementarlo en el gestor escogido, ejemplificando estos conceptos con uso del Modelo de Datos obtenido.

El capítulo concluye con un epígrafe donde son caracterizados los Sistema de Gestión de Base de Datos (SGBD) particularizándose al caso de MSAccess por ser el elegido para implementar el Modelo de Datos, exponiendo las razones tenidas para su elección y presentando la implementación de la Base de Datos en este gestor.

### II.1.- El Modelo de los Datos.



**Figura II.1.1.- Modelo de los Datos**

Este Modelo muestra las entidades de interés para gestionar los documentos de Tesis del Departamento de Informática, las relaciones entre estas entidades, sus llaves primarias (PK) y foráneas (FK) y los elementos de datos considerados de cada una.

### II.2.- Conceptos de Base de Datos utilizados

Para obtener el Modelo de Datos e implementarlo en el gestor escogido se utilizó algunos de los conceptos examinados en [DATE, 1994], [LOPEZ, 2001], [CASTRO, 2006], [CLAUDIO, 1999-2004], [HERRERA, 2004] y [SYBASE, 2006], los que a continuación son expuestos.

**Base de Datos:**

Conjunto de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información de una empresa o negocio en particular, es una modelación dinámica de la realidad en el mundo de los datos. En nuestro caso la organización que explotará el Modelo de Datos presentado es el Departamento de Informática de la UPR para gestionar los documentos de Tesis.

**Modelo de Datos**

Se compone de las **Entidades** que se necesita recoger información, de los elementos de datos (atributos) de interés registrar de estas entidades, de las relaciones existentes entre ellas y de las restricciones que se imponen a los valores posibles a tomar por sus elementos de datos.

Para obtener el Modelo de Datos es necesario realizar las siguientes tareas:

- Identificación de las Entidades y de sus elementos de datos.
- Relaciones existentes entre las Entidades
- Dominio de Valores de los elementos de datos

**Entidad**

Una entidad es un objeto que existe y se distingue de otros objetos de acuerdo a sus características llamadas atributos (elementos de datos). Las entidades pueden ser concretas como: Tesis, Autores, Tutores, Ramas de Aplicación, Temáticas, Palabras Claves o abstractas como: Tutorías, Temáticas por Tesis, Ramas de Aplicaciones por Tesis y Palabras Claves por Tesis.

**Tipos de Entidades**

**Entidad Fuerte.**- Se determinan por si solas, ejemplo de estas son: Tesis, Autores, Tutores, Ramas de Aplicación, Temáticas, Palabras Claves

**Entidad Débil.-** Son aquellas cuya existencia depende de la existencia de otra entidad. En nuestra problemática no tenemos presencia de estas.

**Agregación.-** Es una Entidad que se origina de la relación existente entre otras entidades. El objetivo de crearlas es el poder relacionarlas con otras Entidades o captar información de ellas. En nuestra problemática pudiéramos considerar como agregaciones las entidades: Tutorías, Temáticas por Tesis, Ramas Aplicación por Tesis y Palabras Claves por Tesis, aunque ellas no se justifica al no tener estas entidades relaciones con otras entidades, ni atributos propios.

**Generalización/Especialización.-** La especialización son entidades tipos de la generalizada. Ejemplo de este tipo de entidad “Autores no Diplomantes” que es una especialización de la entidad generalizada “Autores”.

**Elemento de Dato (Atributo):**

Representan las propiedades básicas de las entidades y relaciones, es decir, es la menor información posible a tomarse de una entidad o relación.

**Dominio**

Define los valores posibles que puede tomar un elemento de dato (última columna de las tablas que se muestran en el anexo 1 “Tablas de SisGeTe y su Estructura”).

**Tabla Base**

Es una relación autónoma a diferencia de las vistas (consultas). Toda consulta se compone de una o varias Tablas Bases.

**Llave Candidata**

Elemento de Dato o conjunto de Elementos de Datos que sirven para identificar sin ambigüedad a un registro de una Entidad, es decir, no hay dos registros en la Entidad con el mismo valor de la llave candidata. Según el número de elementos de datos que la componen estas pueden ser:

- **Simples:** Formada por un solo Elemento de Datos, ejemplo de estas son: **IdTesis** o **Nombre Tesis** en la Entidad Tesis, **IdTutor** o **Nombre Tutor** en la Entidad Tutores, etc.
- **Compuestas:** Formada por varios elementos de datos. Ejemplo de estas son: **IdTesis;IdTutor** en la Entidad Tutorías, **IdTesis; IdTematicas** en la Entidad Temáticas X Tesis, etc.

**Llave Primaria (Llave Principal o Maestra):** Es aquella llave candidata tomada como llave de la Entidad. Como ejemplo de Llave Primaria tenemos: **IdTesis** en la Entidad Tesis, **IdTutor** en la Entidad Tutores, **IdTesis; IdTematica** en la Entidad Temática X Tesis, etc.

**Llave Foránea (Llave extranjera):** Son aquellos elementos de datos de la Entidad que sus valores deben concordar con los de la Llave Primaria de otra Entidad, es decir, son Llave Primaria de otra Entidad. Ejemplo de Llaves Foráneas tenemos: **IdTesis** y **IdTutor** en la Entidad Tutorías, la primera es Llave Principal en la Entidad Tesis, mientras que la segunda lo es de la Entidad Tutores.

**Propiedades del Modelo Relacional****➤ Integridad de Llave**

Toda Entidad tiene una Llave Principal por la cual que puede identificarse sin ambigüedad sus registros. Luego esto exige que

una Llave primaria tenga valor único y no pueda ser este valor nulo.

➤ **Integridad Referencial**

Toda entidad que tenga una Llave Extranjera (foránea) los valores que tome deben ser valores que existan en la entidad donde ese elemento de datos es Llave Primaria.

➤ **No hay un orden preestablecido en los elementos de datos:**

Esta propiedad se sigue del hecho de que la estructura de una tabla (o relación) esta definido como un conjunto de elementos de datos sin tener en cuenta su orden.

➤ **Todos los valores de los elementos de datos son atómicos:**

El elemento de datos no puede estar compuesto a su vez como concatenación de otros elementos.

**RELACIONES:**

Es la correspondencia o asociación entre dos o más entidades. Las mismas son representadas mediante líneas que unen a las Entidades relacionadas, señalándose en sus extremos la cardinalidad. Los posibles tipos de relaciones que puede haber entre Entidades son:

➤ **Relaciones 1-1.** : Una instancia de la entidad A se relaciona con una y solamente una de la entidad B y viceversa. Ejemplo de este tipo de relación es la relación entre las entidades: **Autores** y **Autores no Diplomantes**. (vea esta en el Modelo de Datos presentado en la Figura II.1.1).

➤ **Relaciones 1-n:** Cada instancia de la entidad A se relaciona con varias instancias de la entidad B. Ejemplo de este tipo de relación son las relaciones entre las entidades: **Tesis** y **Autores**. Donde el extremo uno corresponde a Tesis y el extremo n a la entidad Autores (Vea esta en el Modelo de Datos presentado en las Figura II.1.1).

- **Relaciones n-m:** Varias instancia de la entidad A se relaciona con varias instancias de la entidad B y viceversa. Ejemplo de este tipo de relación lo tenemos en las relaciones entre las entidades:
  - **Tesis y Tutores** (Tutorías)
  - **Tesis y Temáticas** (Temáticas X Tesis)
  - **Tesis y Palabras Claves** (Palabras Claves X Tesis), etc.

(Vea estas en el Modelo de Datos presentado en las Figuras II.1.1).

Una relación n-m da como resultado una tabla adicional al ser implementado el Modelo de Datos (vea figura II.4.1). Esta tabla por lo general consta con las llaves principales de cada tabla con las cuales tiene relación, pudiendo o no contar además atributos propios, algunos de estos es posible que formen parte de su Llave Primaria. En los ejemplos puestos, entre paréntesis aparece el nombre de las tablas originadas por las relaciones de n-m. Ninguna de estas tablas, en nuestro caso tiene atributos propios. (Vea el Modelo de Datos implementado, figura II.4.1).

## **NORMALIZACIÓN**

Es un proceso que permite obtener una estructura de datos eficientes con la que se eviten las anomalías en su actualización. Fue introducido por Codd para aplicar a los Modelos Relacionales y resulta una expresión formal del modo de realizar un buen diseño de la Base de Datos. A pesar de lo anterior es importante tener en cuenta, al realizar el diseño de la Base de Datos, escoger aquel que mejor se adapta a nuestras necesidades aunque no se encuentre totalmente normalizado. Con la Normalización se logra que:

- El espacio requerido para almacenar los datos sea el menor posible.

- Evitar anomalías al actualizar la Base de Datos, logrando con ello garantizar la integridad de los datos.

Normalizar no garantiza por si solo tener la mejor representación en el mundo de los datos de la problemática, esto solo se logra con la experiencia del diseñador y la buena comprensión de la semántica del problema.

### **PRIMERA FORMA NORMAL (1FN)**

Se dice que una relación está en **1FN** si:

- Todos los elementos de datos son atómicos.
- No existan grupos repetitivos.

### **SEGUNDA FORMA NORMAL (2FN)**

Una relación está en 2FN si:

- Esta en 1FN
- Todos sus elementos de datos secundarios (no llaves) dependen totalmente de la Llave Primaria.

### **TERCERA FORMA NORMAL (3FN)**

Una relación está en 3FN si:

- Si está en 2FN.
- No exista dependencia entre sus elementos de datos secundarios.

Todas las tablas obtenidas del Modelo de Datos se encuentran en Tercera Forma Normal (3FN), lo que puede comprobarse tomando cualquiera de ellas y viendo que cumple con los requerimientos antes señalados. La implementación del Modelo de Datos presentado en la figura II.1.1 puede ser visto en la figura II.4.1.

**II.3.- Caracterización del Gestor de Base de Datos utilizado.**

Para facilitar las tareas de administración de los datos y acelerar el desarrollo de la aplicación se hace necesario realizar una selección adecuada del Sistema Gestor de Bases de Datos (SGBD), estos son un tipo de software muy específico dedicado a servir de interfaz entre las Bases de Datos y las aplicaciones que la utilizan. Resumiendo lo planteado en la literatura referente al tema [DATE, 1994], [LOPEZ, 2001], podemos decir que entre las funciones mínimas que estos deben tener se encuentran las de:

- Permitir introducir, almacenar y recuperar datos.
- Crear y mantener la estructura de los datos (esquema de la BD, tablas, campos, índices).
- Actualizar los datos (altas, bajas, modificaciones).
- Presentar la información (vistas o consultas).
- Facilitar el desarrollo de aplicaciones (interfases de usuario).

Al mismo tiempo permiten otra serie de funciones que hacen de ellos herramientas superiores a los métodos tradicionales de almacenamiento de datos: archivadores, carpetas, etc., entre las cuales se encuentran:

- **Abstracción de la información.**  
Ahorran a los usuarios detalles acerca del almacenamiento físico de los datos. Da lo mismo si una Base de Datos ocupa uno o cientos de archivos, este hecho se hace transparente al usuario, existiendo varios niveles de abstracción que no constituyen objetivo su tratamiento aquí.
- **Facilidad en la manipulación y consulta de los Datos.**  
Poseen un Lenguaje de Manipulación de los datos con el cual facilite esta tarea a los usuarios. Existen otros tipos de Lenguajes como es el Lenguaje de Definición de Datos (SQL)

➤ **Redundancia mínima.**

Permiten poder realizar un buen diseño de una Base de Datos evitando la aparición de información repetida o redundante. Lo ideal es lograr una redundancia nula; no obstante, en algunos casos la complejidad de los cálculos hace necesaria la aparición de redundancias. También es necesario en el Modelo de Datos Relacional tener repeticiones de algunos elementos de datos con los cuales se garantiza la relación entre las tablas del Modelo (Llaves Foráneas).

➤ **Consistencia.**

Permiten en aquellos casos en los que no se ha logrado esta redundancia nula, vigilar que la información que aparece repetida se actualice de forma coherente, es decir, que todos los datos repetidos se actualicen de forma simultánea (integridad referencial).

➤ **Seguridad.**

La información almacenada en una Base de Datos puede llegar a tener un gran valor. Los SGBD garantizan que esta información se encuentra asegurada frente a usuarios malintencionados, que intenten leer o actualizar información no autorizada; frente a ataques que deseen manipular o destruir la información; o simplemente ante las torpezas de algún usuario autorizado pero despistado. Normalmente, los SGBD disponen de un complejo sistema de permisos a usuarios y grupos de usuarios por objetos, que permiten otorgar diversas categorías de permisos.

➤ **Respaldo y recuperación.**

Proporcionan una forma eficiente al realizar copias de seguridad de la información almacenada en ellos.

A continuación caracterizaremos el gestor de Base de Datos de Microsoft con el cual se implementó el Modelo de Datos presentado en la figura II.1.1, para lo

cual se consulto: [FORTE,1999], [SCOUT,1999], [MICROSOFT CORPORATION,1995], [MICROSOFT,1995], [MSDN,2003].

En la bibliografía referida se destaca que MSAccess es una buena opción para la pequeña y mediana empresa. Es muy visual, posee poderosos asistentes (Wizard) con los cuales se pueden crear interfaces para la entrada, modificación y consulta de los datos de una tabla, algo mucho menos engorroso que la introducción de sentencias SQL.

Se enfatizan características específicas, las cuales algunas de ellas fueron utilizadas en la implementación de la Base de Datos como:

- Permite el ingreso de datos de tipos: Numéricos, Texto, Fecha, Sí/No, OLE, Moneda, Memo, etc. con los cuales se satisfacen los requerimientos de SisGeTe.

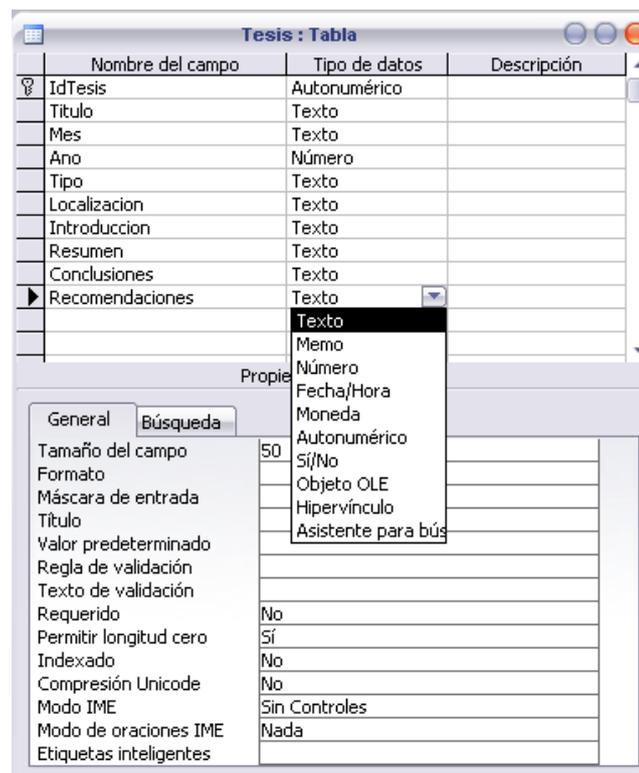
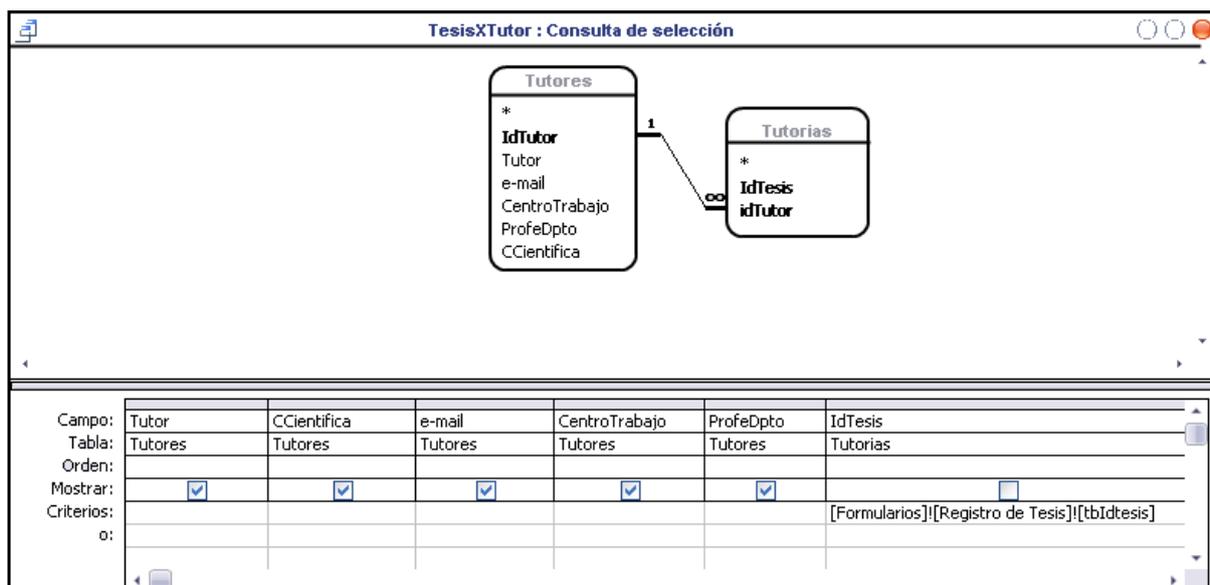


Fig. II.3.1. - Ejemplo de Tipos de Datos posibles a utilizar.

- Posibilita realizar consultas directas a las tablas contenidas mediante instrucciones SQL o mediante su potente generador de consultas que construye las mismas de manera visual con solo arrastrar las tablas a usar y determinar los campos a utilizar de estas tablas. Brindando la posibilidad de utilización de gran variedad de consultas como las de selección (figura II.3.2), actualización de datos anexados, de eliminación, etc.

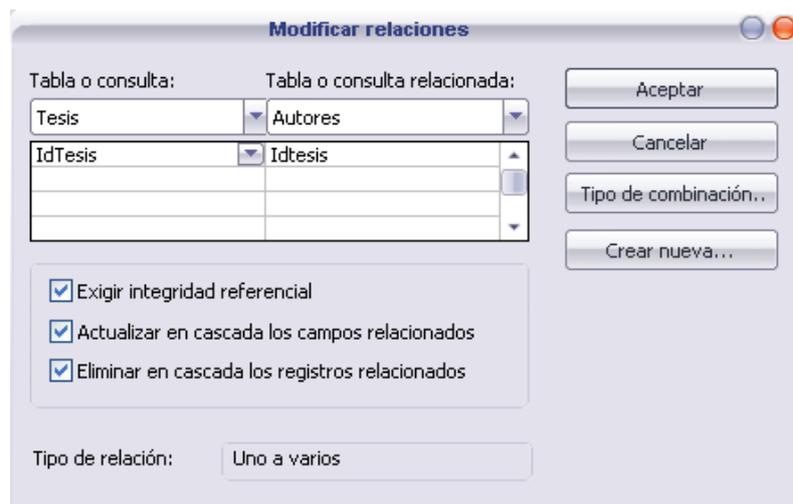


**Fig. II.3.2.- Ejemplo de Consulta de Selección construida con el generador**

```
SELECT Tutores.Tutor, Tutores.CCientifica, Tutores.[e-mail], Tutores.CentroTrabajo, Tutores.ProfeDpto
FROM Tutores INNER JOIN Tutorias ON Tutores.IdTutor = Tutorias.idTutor
WHERE (((Tutorias.IdTesis)=[Formularios]![Registro de Tesis]![tbIdtesis]));
```

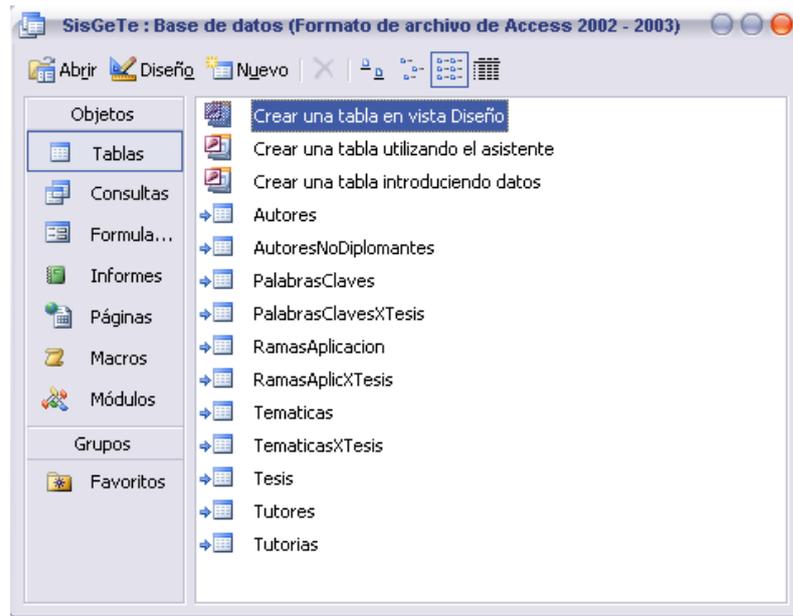
**Fig. II.3.3.- Vista en SQL de la consulta de Selección anterior.**

- Garantiza por si mismo las integridades de llave y referencial así como las operaciones de actualizado y borrado en cascada, facilitándose así mucho más el trabajo al no tener que controlar esto por código.



**Fig. II.3.4.- Ejemplo de exigencia de la Integridad Referencial y de las operaciones de actualizado y eliminado en cascada.**

- Una Base de Datos de MSAccess contiene tanto las tablas como los demás objetos utilizados en la interfaz de usuario: formularios, reportes, módulos, etc. Sin embargo es posible separar los datos (tablas) de los usados en la interfaz (Formularios, Reportes, Módulos, Macros), con objetivo de lograr independencia entre estos y así facilitar la tarea de mantenimiento de la aplicación, como también lograr mayor eficiencia en el uso de la aplicación en un entorno de red al tener cada usuario en su ordenador solo la interfaz, encontrándose los datos en un servidor (o disponer de una réplica de los mismos). Esto se logra con la creación de dos Bases de Datos, una de las cuales una contiene los objetos de la Interfaz (**SisGeTe**), y la otra las tablas, es decir, los datos, (**SisGeTe\_Datos**). La Base de Datos de la interfaz (SisGeTe) tiene una referencia (vínculo) a la Base de Datos "**SisGeTe\_Datos**" que contiene las tablas, esto es reflejado mediante una flecha a cada tabla vinculada en la Base de Datos "**SisGeTe**".



**Fig. II.3.5.- Tablas de la Base de Datos “SisGeTe\_Datos”  
vinculadas en la Base de Datos SisGeTe**

- MSACCESS brinda diferentes niveles y métodos de protección de los datos entre ellos el más potente es el de seguridad a nivel de usuario. Con el se puede determinar a que objeto se puede acceder y que acciones es posible realizar con el. Esto será tratado con profundidad en el capítulo IV donde será expuesta la seguridad del Sistema.
- MSAccess tiene capacidad de red que hace de Access un gestor muy apto para su empleo en Bases de Datos de pequeñas empresas, que no necesitan un número de accesos simultáneos muy alto. Además de esto ofrece el método de Replicar la Base de Datos donde SisGeTe\_Datos en los PC que harán uso de ella, así cada usuario tendrá en su PC una replica de la Base de Datos evitando con ello el tráfico en la red. Las replicas serán sincronizadas entre si en el momento que lo decida el administrador.
- MSAccess posee una excelente compatibilidad con su hermano mayor SQL Server, que permite emigrar a este con gran facilidad, brindando para ello un asistente.

Algunos críticos y especialistas en Diseño de software dirían que MSAccess no es aconsejable, por las dificultades que presentan al momento de trabajar con grandes volúmenes de información, o tal vez con un numero elevado de usuarios, sin embargo estas no son nuestras condiciones, en el caso que esto ocurra solo bastaría con emigrar a SQL Server la Base de Datos SisGeTec\_Datos. Por lo que MSAccess es el Gestor de Base de Datos ideal para desarrollar nuestro sistema al cumplir con todas las exigencias planteadas, a ello se le agrega poseer un buen dominio, y experiencia en su uso y una documentación abundante de este gestor relacional.

### III.4.- Implementación del Modelo de Datos.

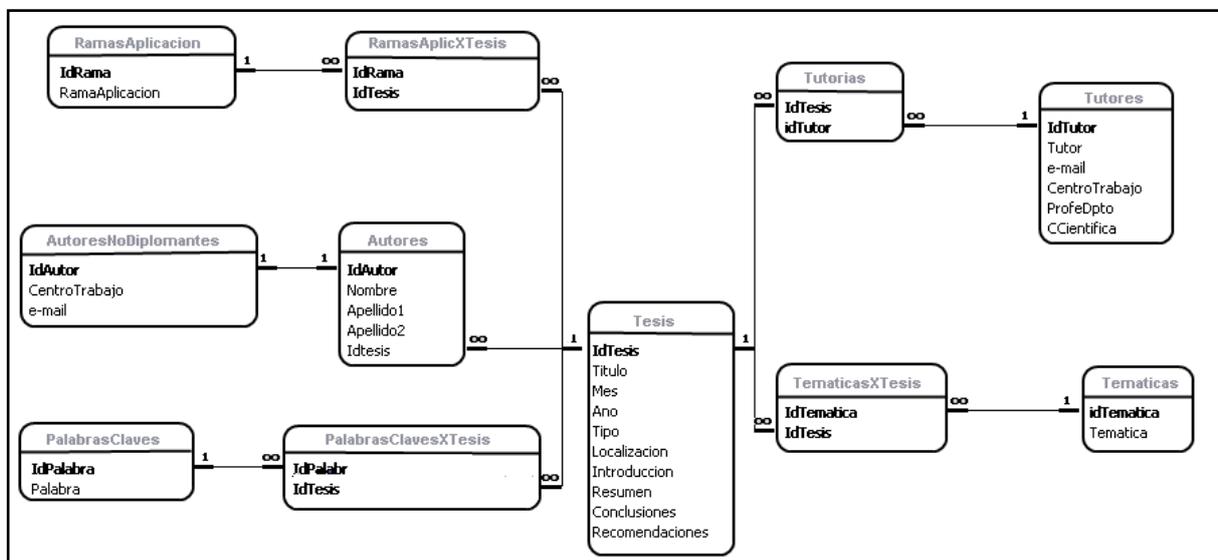


Fig. II.4.1.- Modelo de Datos implementado en MSAccess.

En el anexo1 “Tablas de SisGeTe y su Estructura” puede ver de cada tabla, sus llaves primarias y foráneas, elementos de datos y el tipo de dato utilizado para cada elemento, el que no se muestra en la implementación.

# *CAPÍTULO III*

## *Análisis de las Funcionalidades del Sistema y su desarrollo*

**INTRODUCCION**

El capítulo comienza con la captura de los requerimientos a cumplir por el Sistema (SisGeTe). Se establece sus actores (usuarios que interactúan con el Sistema para obtener un beneficio del mismo), los roles o papeles que estos desempeñan ante el Sistema y los Casos de Uso (funcionalidades brindadas por el Sistema a sus actores). Se muestra el diseño de las interfases de usuario de algunos de los Casos de Uso. Son caracterizados el Lenguaje Unificado de Modelado (UML), del que se empleó sus artefactos: Actor, Caso de Uso y Diagramas de Casos de Uso, y la herramienta CASE Power Designer, empleada para confeccionarlos.

En el segundo epígrafe se presenta el Lenguaje de Programación “Visual Basic para Aplicaciones”, lenguaje embebido en todas las aplicaciones de Office incluyendo MSAccess, con el que se implementó las Interfases de Usuarios diseñadas, y el entorno de desarrollo de MSAccess empleado al desarrollar la aplicación en el lenguaje seleccionado.

**III.1.- El diseño de las interfases de usuarios**

Para diseñar la Interfaz de Usuario es necesario conocer el Modelo de Datos, vea este en el Capítulo II, y los requerimientos funcionales que debe cumplir el sistema para satisfacer al usuario. La captura de estos requerimientos se realizó mediante entrevistas realizadas al usuario, teniendo identificados los requerimientos funcionales:

**R1- Autenticarse.****R2- Gestionar Cuentas de Usuario.****R3.- Gestionar Contraseña de Conexión.****R4- Gestionar Documentos de Tesis.**

R41 Registrar Tesis.

R42 Editar datos de Tesis.

R43 Eliminar Tesis.

**R5- Consultar Documentos de Tesis.**

R51 Realizar búsquedas de Documentos de Tesis por diversos criterios.

**R6- Utilidades de Datos.**

R61- Actualizar vinculo con los Datos

R62- Resguardar Datos

R63- Compactar Datos

R67- Reparar Datos

Con los que se cubren las necesidades de todos los usuarios que harán uso del sistema (actores). Estos se identifican con apoyo de su concepto. Según lo señalado en [BOOCH, 2000], [JACOBSON, 2000], [BERTAMI, 2007], [MAURILLO, 2006] se tiene que:

**Actor:** Son elementos que interactúan con la aplicación ya sean humanos, software o hardware para beneficiarse de alguna funcionalidad brindada por la esta. Los actores no forman parte del sistema, sólo interactúan con este. Al interactuar pueden que:

- Brinden información al sistema
- Reciban información del sistema
- De entrada y reciba información para y del sistema.

Típicamente, estos actores son encontrados en la declaración del problema y por las conversaciones con usuarios y expertos del dominio. En la literatura mencionada se señala que las preguntas siguientes pueden ayudar a identificar los mismos:

- ¿Quién está interesado en un cierto requisito?
- ¿En que organización el sistema es usado?
- ¿Quién se beneficiará del uso del sistema?
- ¿Quién proporcionará al sistema la información, quién usará esta información, y quién quitará esta información?
- ¿Quién apoyará y mantendrá actualizado el sistema?
- ¿El sistema usa un recurso externo?
- ¿Una persona juega varios papeles o roles diferentes ante el sistema?
- ¿Varias personas juegan el mismo papel?
- ¿El sistema actúa recíprocamente con un sistema heredado?

En el lenguaje Unificado de Modelado (UML), un actor se representa como un stickman, o muñeco, como muestra la Figura III.1.1



**Figura III.1.1 Representación de un actor en UML.**

Apoyándonos en lo antes dicho, los actores del Sistema (**SisGeTe**) y sus roles son:

Actor	Rol
<b>Consultante</b>	<ul style="list-style-type: none"> <li>• Consultar Documentos de Tesis.</li> <li>• Realizar búsquedas de los Documentos de Tesis según criterio.</li> <li>• Autenticarse.</li> </ul>
<b>Registrador</b>	<ul style="list-style-type: none"> <li>• Cambiar contraseña de conexión.</li> </ul> <p>Además de los roles del consultante se agregan los de:</p> <ul style="list-style-type: none"> <li>• Mantener actualizado los registros de información: Tesis, Autores, Tutores, Tutorías, Temáticas, Ramas de Aplicación, Palabras Claves e indicadores de búsqueda por tesis.</li> </ul>
<b>Analista</b>	<p>Además de los roles de los registradores podrá:</p> <ul style="list-style-type: none"> <li>• Utilizar las utilidades de Datos</li> <li>• Mantener actualizado el registro de cuentas usuario</li> <li>• Modificar los diseños de las tablas e interfaces.</li> </ul>

Figura III.1.2 Actores de SisGeTe y sus roles.

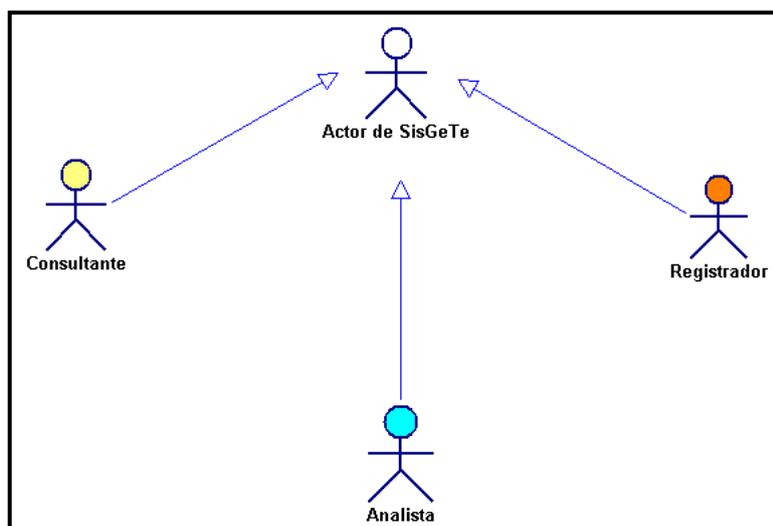


Figura III.1.3 Actores de SisGeTe.



**Figura III.1.4 Jerarquía de Actores en SisGeTe**

A continuación se identificaron los Casos de Uso del sistema con el apoyo de su concepto, usando para ello la literatura y diplomas ya mencionados. Se define que:

**Caso de Uso:** Es la agrupación de fragmentos de funcionalidad que el sistema ofrece para aportar un resultado de valor a un actor. Los Casos de Uso establecen el diálogo entre los actores y el sistema mediante una interfaz de usuario. La colección de casos de uso para un sistema constituye todas las maneras definidas por las que se puede hacer uso del sistema.

En la documentación mencionada se dice que las siguientes preguntas pueden ayudar a identificar los Casos de Uso del sistema:

- ¿Cuáles roles juegan cada actor del Sistema?
- ¿Cualquier actor creará, guardará, cambiará, quitará, o leerá la información en el sistema?
- ¿Qué funcionalidades creará, guardará, cambiará, quitará, o leerá esta información?
- ¿Cualquier actor necesitará información del sistema sobre cambios súbitos y externos?
- ¿Cualquier actor necesita ser informado sobre ciertas ocurrencias en el sistema?
- ¿Qué funcionalidades apoyarán y mantendrán el sistema?
- ¿Pueden todos los requisitos funcionales realizarse por los casos de uso identificados?

En el UML, un Caso de Uso se representa como un ovalo, como es mostrado en Figura III.1.5., donde en su interior aparece su nombre.



**Figura III.1.5.- Representación de un Caso de Uso en UML**

Teniendo en cuenta los requerimientos funcionales primarios o bases a cumplir por SisGeTe se tiene como Casos de Uso.

<b>Casos de Uso</b>	<b>Requerimiento Funcional</b>
<b>Autenticarse.</b>	R1.- Autenticarse
<b>Gestionar Cuentas de Usuario.</b>	R2.- Gestionar Cuentas de Usuario.
<b>Gestionar Contraseña de Conexión.</b>	R3.- Gestionar Contraseña Conexión
<b>Gestionar Documentos de Tesis.</b>	R4.- Gestionar Documentos Tesis
<b>Consultar Documentos de Tesis.</b>	R5.- Consultar Documentos tesis
<b>Utilidades de Datos</b>	R6.- Utilidades de Datos.

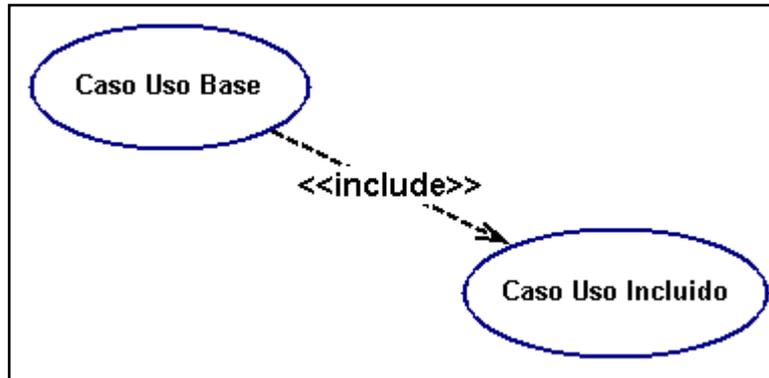
**Figura III.1.6.- Caso de Usos de SisGeTe y requerimientos funcionales asociados a ellos.**

Existen Casos de Usos que nos facilita la comprensión del Caso de Uso Base o es una extensión del comportamiento de este, estos Casos de Uso se denominan Casos de Uso Abstractos pudiendo ser de dos tipos:

**Casos de Uso Incluidos:** Se justifica su empleo cuando:

- Su contenido puede ser rehusado en otros Casos de Uso.
- Simplifica la comprensión del Caso de Uso Base.

Los mismos se ejecutan al ejecutarse el Caso de Uso Base. En el UML, un Caso de Uso Incluido de uno Base se representa como se muestra en Figura III.1.7

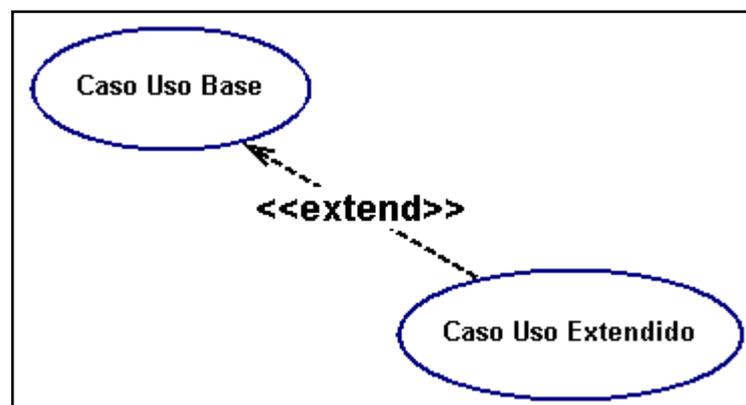


**Figura III.1.7.- Representación en UML de los Casos de Uso Incluido.**

**Casos de Uso Extendido:** Se justifica su empleo cuando:

- Existe una extensión del comportamiento del Caso de Uso Base.
- Existen comportamientos del Caso de Uso Base que se ejecutan de acuerdo a la selección que realice el actor.

A diferencia de los Incluidos estos no se ejecutan al ejecutarse el Caso de Uso Base. En el UML, un Caso de Uso Extendido de uno Base se representa como se muestra en Figura III.1.8



**Figura III.1.8.- Representación en UML de los Casos de Uso Extendido**

Ejemplos de estos tipos de Casos de Uso abstractos pueden verse en el Diagrama de Casos de Uso de SisGeTe.

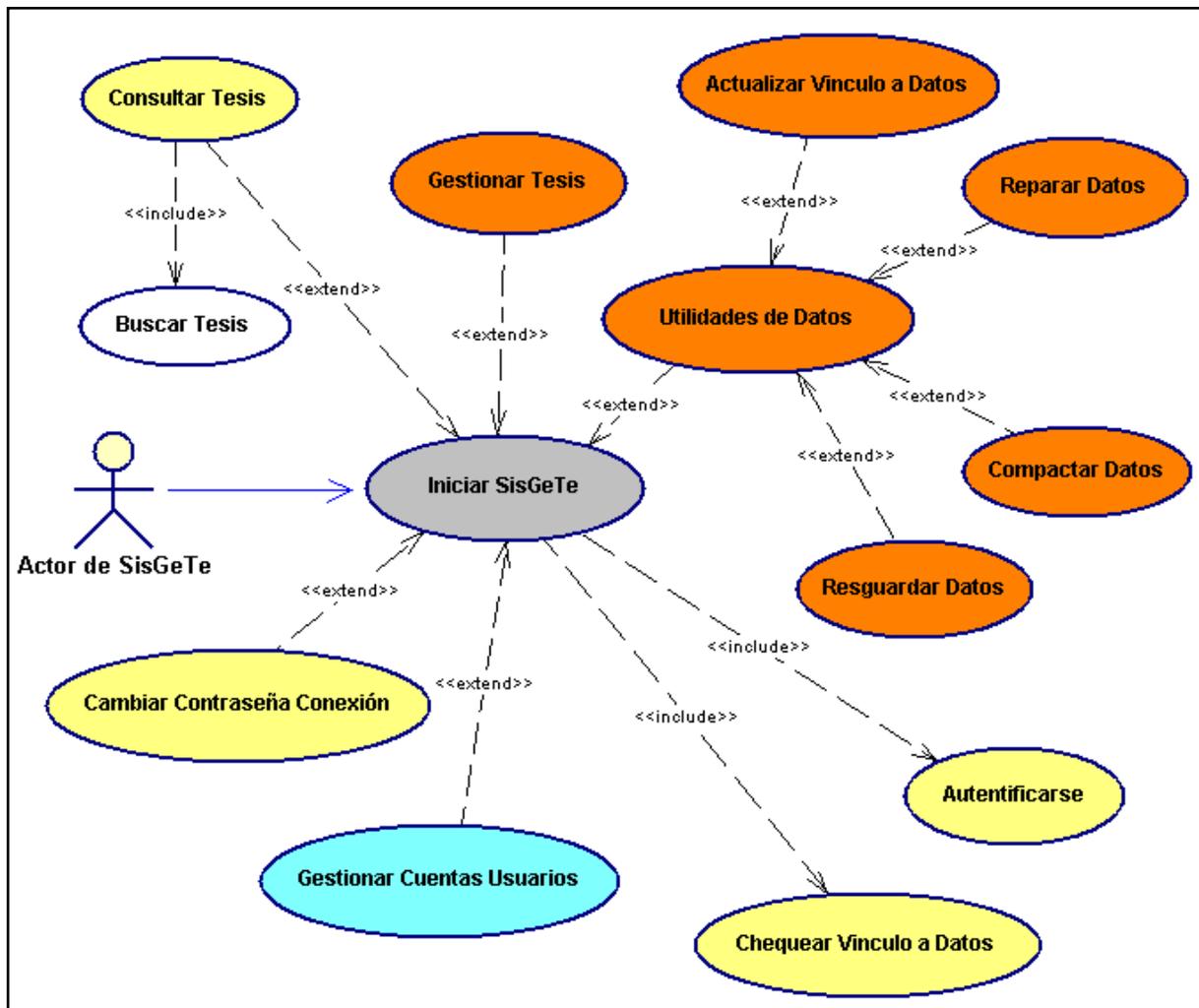


Figura III.1.9.- Diagrama de Casos de Uso de SisGeTe.

Seguidamente se muestra el diseño de las Interfases de Usuario correspondiente a cada uno de los Casos de Uso principales del sistema.

➤ Interfases del Caso de Uso “Gestionar Tesis”

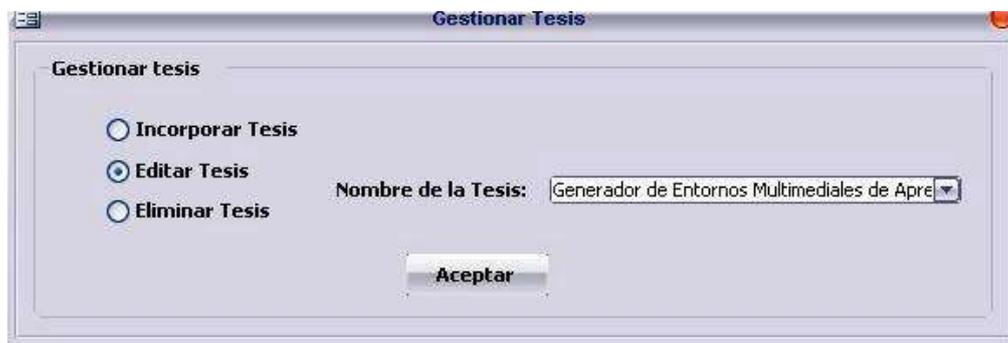


Figura III.1.10.- Interfaz principal del CU “Gestionar Tesis”

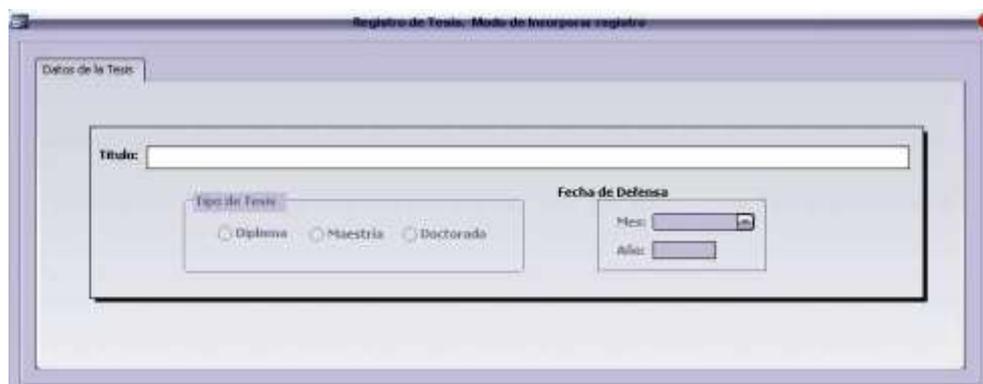


Figura III.1.10a.- Interfaz del CU “Incorporar Tesis”

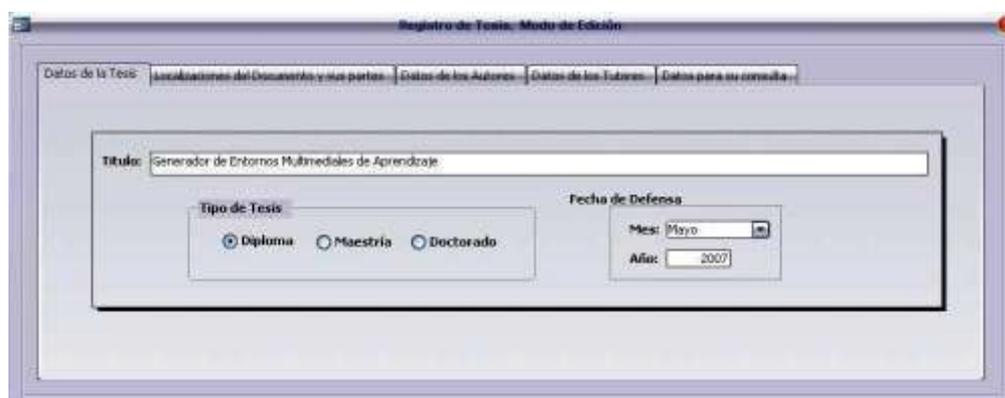


Figura III.1.10b.- Interfaz del CU “Editar Tesis”

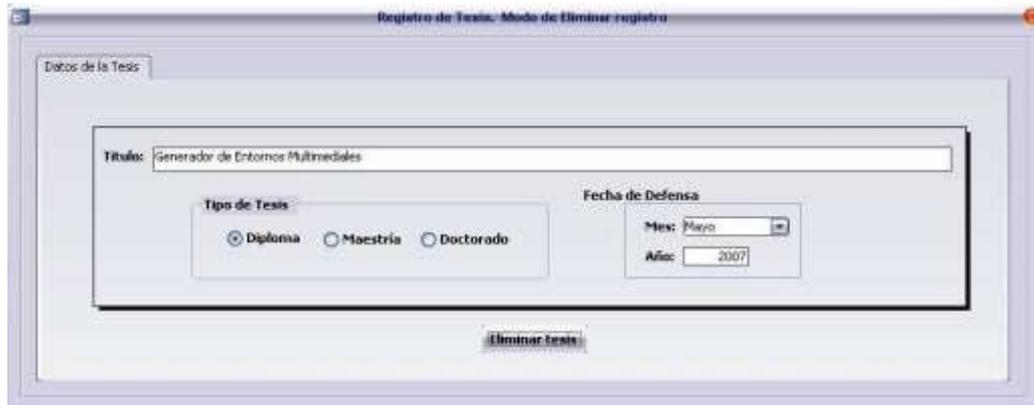


Figura III.1.10c.- Interfaz del CU “Eliminar Tesis”

➤ Interfases del caso de Uso “Consultar Tesis”.

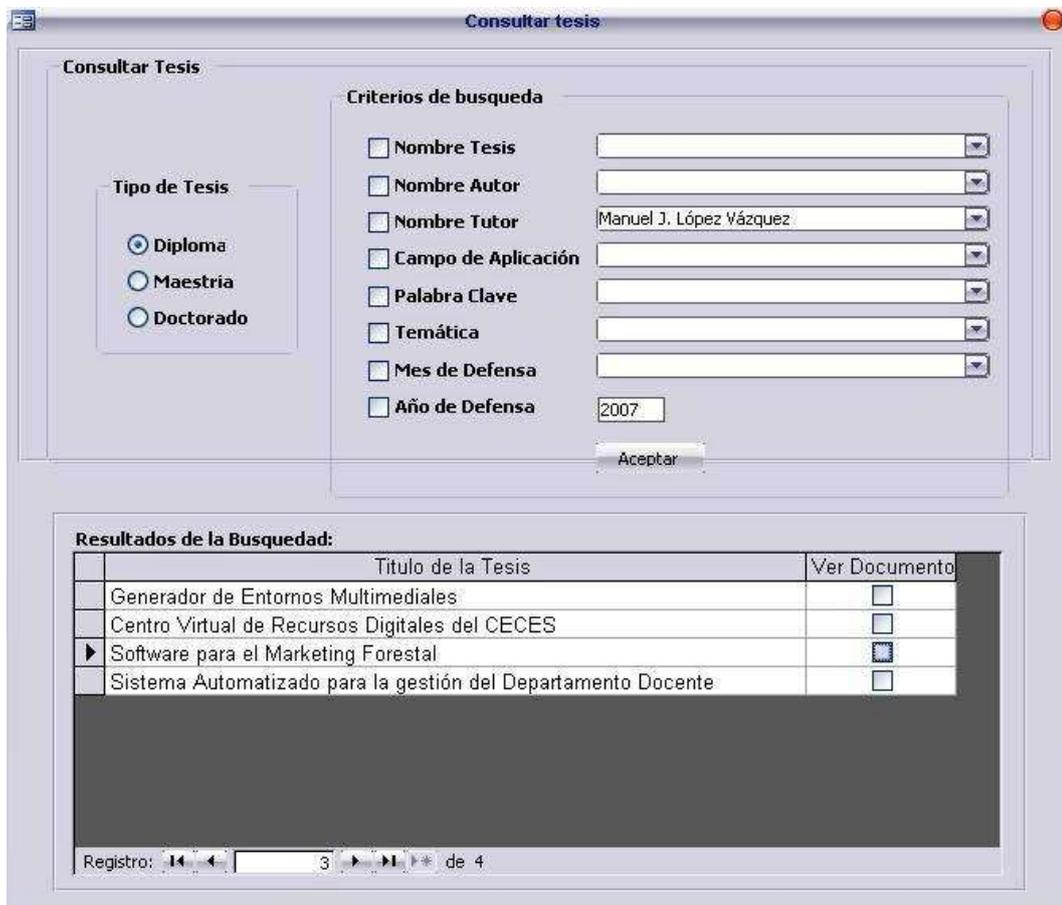


Figura III.1.11.- Interfaz del CU “Consultar Tesis”

Interfaz del CU “Utilidades de Datos”.

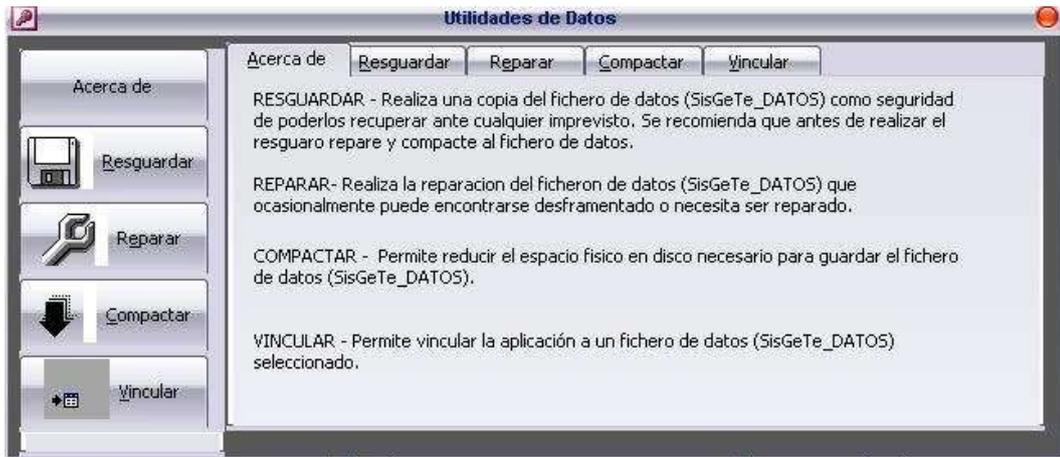


Figura III.1.12.- Interfaz del CU “Utilidades de Datos”

➤ Interfaz del CU “Autenticarse”

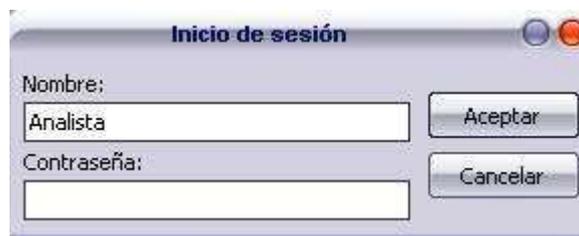


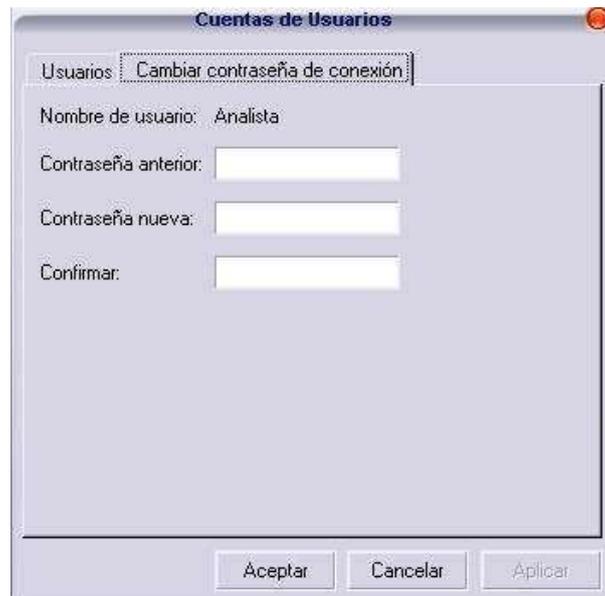
Figura III.1.13.- Interfaz del CU “Autenticarse”

➤ Interfaz del CU “Gestionar Cuentas de Usuarios”



Figura III.1.14.- Interfaz del CU “Gestionar cuentas de Usuario”

Interfaz del CU “Cambiar contraseña de conexión”



The image shows a Windows-style dialog box titled "Cuentas de Usuarios". It has a tabbed interface with the active tab labeled "Cambiar contraseña de conexión". Inside the dialog, there are four text input fields: "Nombre de usuario" (containing "Analista"), "Contraseña anterior", "Contraseña nueva", and "Confirmar". At the bottom of the dialog, there are three buttons: "Aceptar", "Cancelar", and "Aplicar".

**Figura III.1.15.- Interfaz del CU “Cambiar contraseña de conexión”**

### III.1.1.- El lenguaje Unificado de Modelado (UML).

Los artefactos del Lenguaje Unificado de Modelado (UML) empleados para diseñar **SisGeTe** fueron: Actor, Casos de Uso, Casos de Usos Abstractos y Diagramas de Casos de Usos, además de estos artefactos este lenguaje presenta muchos más que posibilitan tener diferentes vistas del Sistema. Seguidamente haremos una breve caracterización del mismo con apoyo de la bibliografía referenciada al respecto, [JACOBSON, 2000], [BOOCH, 2000] y los diplomas [BERTAMI, 2007]. Comencemos respondiendo a la pregunta, ¿que es UML?

**UML** es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra a una gran cantidad de software, este se caracteriza por:

- Ser un lenguaje gráfico con una semántica bien definida que estandariza la modelación durante el proceso de desarrollo del software para que sea legible por todo el equipo de proyecto.
- Permite construir modelos precisos, no ambiguos y completos
- No es un lenguaje de programación, pero sus modelos pueden transformarse en código fuente, tablas o almacenamiento de objetos (Generación directa del código)
- Permite describir requerimientos, la arquitectura y modelar las pruebas a través de artefactos que permiten documentar el proceso.

Es importante recalcar que UML no es una guía para realizar el análisis y diseño orientado a objetos, es decir, no es un proceso. UML es un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos. **[BOOCH, 2000]**.

Los posibles artefactos que brinda este lenguaje son::

### **Diagrama de Casos de Uso** (Vea figura III.1.9)

Es una vista gráfica de algunos o todos los actores, Casos de Uso, y sus interacciones identificadas por un sistema. Cada sistema tiene típicamente un Diagrama de Caso de Uso Principal que es una imagen de la frontera del sistema (actores) y la mayor funcionalidad proporcionada por el sistema (casos de uso). Con un diagrama de caso de uso, usted ve de forma instantánea la funcionalidad del sistema inmediatamente. Pueden agregarse detalles extensos después al diagrama si usted necesita elucidar puntos interesantes en la conducta del sistema.

### **Componentes de un Diagrama de Caso de Uso.**

- **Actor:** Un actor representa quien o que inicia una acción dentro del sistema, en otras palabras, es simplemente un rol que es llevado a cabo por una persona o cosa. Un Actor en un diagrama Uso-Caso es

representado por una figura en forma de persona.

- **Caso de Uso:** Es representado por un ovalo que describe la funcionalidad a *grosso modo* que se requiere por el sistema.
- **Comunicación:** Este elemento representa la relación que existe entre un Caso de Uso y un Actor, dicho elemento es representado simplemente por una línea recta que se extiende de la figura del actor hacia el ovalo del Caso de Uso o viceversa.
- **Limite de Sistema (System Boundry):** Empleado para delimitar los límites del sistema, y representado por un rectángulo con color de fondo distintivo.
- **Generalización:** Una generalización indica que un Caso de Uso (ovalo) es un caso especial de otro caso, en otros términos, representa una relación padre-hijo, donde el hijo puede ser suplido directamente por el padre en cualquier momento. Este elemento es representado por una línea con flecha que se extiende del Caso de Uso hijo hacia el Caso de Uso padre (general).
- **Inclusión:** Una inclusión es utilizada para indicar que un Caso de Uso (ovalo) depende de otro caso, dicho de otra manera, significa que la funcionalidad de determinado caso se requiere para realizar las tareas de otro. Este elemento es representado por una línea punteada con flecha y comentario <<include>> que se extiende del Caso de Uso base hacia el Caso de Uso de inclusión.
- **Extensión:** Una extensión representa una variación de un Caso de Uso a otro, aunque similar a una generalización, una extensión representa una dependencia especifica, mientras una generalización no implica que los Casos de Uso dependen uno del otro. Este elemento es representado por una línea punteada con flecha y comentario <<extend>> que se origina del Caso de Uso extensión al Caso de Uso base.

Los diagramas que continuación se presentan, aunque no fueron empleados, permite tener una visión más completa de las potencialidades brindadas por este lenguaje.

### **Diagramas de Colaboración**

Un diagrama de colaboración es un camino alternativo de mostrar un escenario. Este tipo de diagrama muestra las interacciones de objetos organizadas alrededor de los objetos y sus enlaces a cada uno de los otros.

### **Diagrama de Secuencia.**

Un diagrama de secuencia muestra las interacciones de objetos organizadas en la secuencia de tiempo. Ellos representan los objetos y clases involucradas en el escenario y la sucesión de mensajes intercambiados entre los objetos necesarios para llevar a cabo la funcionalidad del escenario. Los diagramas de secuencia típicamente son asociados con una realización de casos de uso en la Vista Lógica del sistema bajo desarrollo.

### **Diagrama de Actividades.**

El diagrama de actividad es parte de un Modelo Objeto-orientado. Es uno de los diagramas disponibles en un OOM. Muestra los símbolos de los objetos definidos en el modelo. Los objetos principales son las actividades y las transiciones entre ellos. Estos diagramas representan la dinámica del sistema. Ellos son mapas de flujo que se usan para mostrar el flujo de trabajo de un sistema; es decir, ellos muestran el flujo de control desde una actividad a otra actividad en el sistema, que actividades pueden hacerse en paralelo, y cualquier camino alternativo a través del flujo.

### **Diagrama de Componente**

Es parte de un Modelo Objeto-orientado. Es uno de los diagramas disponibles en un OOM. Muestra los símbolos de componentes definidos en

el modelo incluso interfaces definidas en componentes y las relaciones entre ellos. Se dirige la vista de aplicación estática de un sistema, dibujada como un gráfico de componentes del software conectado por dependencias.

Los componentes también pueden ser conectados por relaciones de la generalización cuando un componente hereda de otro uno.

### **Diagrama de Despliegue.**

El diagrama del despliegue es parte del Idioma Modelado Unificado (UML) es un diagrama de aplicación que complementa el diagrama del componente dando detalles más exactos sobre la aplicación física y interacciones de componentes.

El diagrama del despliegue es una vista de nodos conectada por eslabones de comunicación. Le permite diseñar nodos, casos del componente que pueden contenerse dentro del símbolo del nodo, los objetos del archivo asociaron con nodos que se usan para el despliegue, y relaciones entre los nodos. Cuando el nodo contiene casos del componente, esto indica que el componente ejecuta en el nodo.

### **Diagrama de Clases.**

El diagrama de la clase es parte de un Modelo Objeto-orientado. Es uno de los diagramas disponibles en un OOM. Muestra los símbolos de los objetos definidos en el sistema. Los objetos son las clases, paquetes, y las relaciones entre ellos.

### **Diagrama de Objeto.**

Un diagrama de objeto muestra la existencia de objetos y sus relaciones en el plan lógico de un sistema. Un diagrama de objeto puede representar todos o parte de la estructura del objeto de un sistema, y principalmente ilustra la semántica de mecanismos en el plan lógico.

### **Diagrama de Estado.**

Un diagrama del statechart (estado) muestra el espacio estatal de una clase dada, los eventos que causan una transición de un estado a otro, y las acciones que son el resultado de un cambio estatal.

### **III.1.2.- Caracterización de la herramienta Case utilizada.**

Las Herramientas CASE (**C**omputer **A**ided **S**oftware **E**ngineering), tienen su propia historia la que trataremos de describir de la manera más sintética. En la década de los setenta el proyecto ISDOS desarrolló un lenguaje llamado "Problem Statement Language" (PSL) para la descripción de los problemas de usuarios y las necesidades de solución de un sistema de información en un diccionario computarizado. Problem Statement Analyzer (PSA) era un producto asociado que analizaba la relación de problemas y necesidades. Pero la primera herramienta CASE como hoy conocemos para PC fue "Excelerator" en 1984. Actualmente la oferta de herramientas CASE es muy amplia entre muchas otras están: Rational Rose, Erwin ER/STUDIO, SYSTEM ARCHITECT, y el **POWER DESIGNER** la que se seleccionó para crear los Diagramas de Caso de Uso.

En la actualidad las Herramientas Case son muy importantes en el campo de la Informática, ya que son de gran ayuda en las actividades de desarrollo de los grandes Sistemas de Información, que no es nuestro caso. Entre sus principales objetivos se encuentran:

- Aumentar la productividad de las áreas de desarrollo y mantenimiento de los sistemas informáticos.
- Mejorar la calidad del software desarrollado.
- Reducir tiempos y costos de desarrollo y mantenimiento del software.
- Mejorar la gestión y dominio sobre el proyecto en cuanto a su Planificación, Ejecución y Control.

- Mejorar el archivo de datos (enciclopedia) de conocimientos y sus facilidades de uso, reduciendo la dependencia de analistas y programadores.

A continuación caractericemos la Herramienta Case **Power Designer** apoyándonos lo que se dice de esta en los diplomas **[MAURILLO,2006]**, y **[ORTIZ,2007]**.

- Crea bases de datos y aplicaciones cliente/servidor basadas o no en Web.
- Permite a los diseñadores de aplicaciones complejas de cliente/servidor tener una descripción general de los procesos particulares para comprender mejor a la organización.
- Exporta información del modelo físico y extiende atributos al diccionario de 4GL. Importa atributos extendidos de PowerBuilder.
- Soporta definición de atributos extendidos para PowerBuilder, Progress, Uniface, PowerHouse, Axiant, y NS-DK.
- Cuenta con herramientas para la creación y control de diagramas como son:
  - Off-page Connector: que representa los flujos de entradas y salidas en un proceso.
  - Business Rules: Define las reglas de uso para Procesos de Almacenamiento de datos, Entidades externas, y Flujos de dato.
  - CRUD Matrix: Define el efecto de un proceso de datos en términos de Crear, Leer, Actualizar, y Borrar operaciones (CRUD).
- Posee una ayuda sensible al contexto.
- Data Architect proporciona capacidades de modelado de datos tradicional, incluyendo diseño de Bases de Datos, generación,

mantenimiento, ingeniería de reversa y documentación para arquitecturas de bases de datos.

- Permite que los diseñadores de Bases de Datos creen estructuras de datos flexibles, eficientes y efectivos para usar una ingeniería de aplicación de bases de datos.
- Proporciona un diseño conceptual de modelo de datos, generación automática de modelo de datos, diseño de normalización física, sistema de manejo de bases de datos múltiples (DBMS) y soporte de herramientas de desarrollo, y elementos de reportes con presentación y calidad.
- Mediante el incremento del modelo de la base de datos, AppModeler genera instantáneamente objetos, componentes data-ware, y hasta aplicaciones básicas listas para ejecutarse inmediatamente en PowerBuilder, Power++, Visual Basic, Delphi, y Web-based objects.
- El AppModeler permite a los desarrolladores: diseñar modelos de bases de datos físicas o crearlas instantáneamente a través de la ingeniería de reversa de bases de datos existentes, generar, documentar y mantener bases de datos, generar rápidamente objetos de aplicación y componentes de datos para PowerBuilder 4.0 y 5.0; Visual Basic 3.0, 4.0, y 5.0; Delphi 2.0; Power++; y el Web.
- Generación de objetos PowerBuilder. Soporta todas las ediciones de PowerBuilder 4.0 y 5.0. Genera objetos personalizables de PowerBuilder y componentes basados en modelos de bases de datos físicos y plantillas que se encuentran dentro de las librerías de clases de su elección. Genera objetos ventana y ventana de datos basadas en tablas, vistas y relaciones de llaves primarias-foráneas. Genera y hace ingeniería de reverso a los atributos. Incluye plantillas personalizables para la librería PowerBuilder Foundation Class (PFC).
- Generación de objetos en Visual Basic. Soporta todas las ediciones de Visual Basic 3.0, 4.0, y 5.0. Incluye add-in de Visual Basic para la fácil manipulación de plantillas predeterminadas personalizables. Genera formas basadas en tablas, vistas, y relaciones de llaves primarias-

secundarias. Genera proyectos basados en modelos de propiedades. Genera controles tales como menús, listas, etc.

- Generación de objetos Delphi. Soporta todas las ediciones de Delphi 2.0. Incluye add-in de Delphi para una manipulación de plantillas personalizables predefinidas.
- Genera aplicaciones y objetos (proyectos, formas, y controles) de tablas, columnas y referencias.
- Soporta Modelos Funcionales y Notaciones de Diagramas de Flujo Modelo Funcional de Objeto (OMT) Yourdon/DeMarco Gane & Sarson SSADM (Análisis de sistema estructurado y metodología de diseño, Structured System Analysis & Design Methodology).
- Creación flexible de reportes estructurados a través de plantilla de reportes.
- Estructura de árbol de elementos seleccionados para facilitar la organización.
- Posee Objetos drag-and-drop con estructura de árbol para facilitar los ajustes.
- Salva plantillas de reportes.
- Vista previa del reporte antes de imprimirlo.
- Selecciona un lenguaje por omisión para el reporte.
- Permite dirigir la impresión o exportarla a Microsoft Word, Word Perfect, PageMaker, etc.

### **III. 2.- Desarrollo de SisGeTe.**

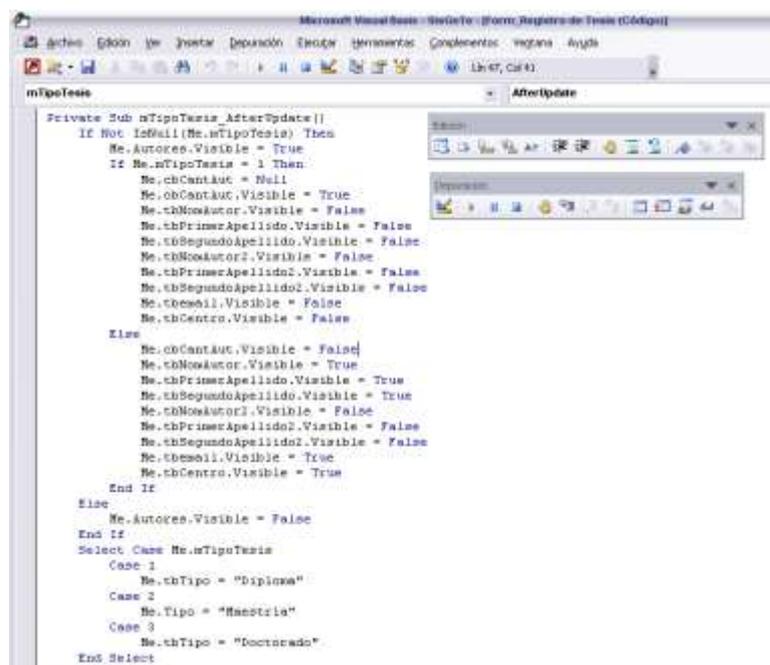
El diseño de las interfaces de usuario a emplear por los Casos de Usos de SisGeTe se implementaron mediante el lenguaje de Programación **Visual Basic para Aplicaciones**, usando como entorno de desarrollo el ofrecido por MSAccess, los cuales caracterizamos con apoyo de la bibliografía [MSDN, 2003 ], [MICROSOFT, 1995], [MICROSOFT CORPORATION, 1995], [PABLOPG3, 2000], [SANTANA, 2005], [NCHT, 2001], [ELGUILLE, 2002], [FORTE, 1999] y [SCOUT, 1999 ].

**III.2.1.- El lenguaje de programación “Visual Basic para Aplicaciones”**

Visual Basic es uno de los tantos lenguajes de programación que podemos encontrar hoy en día. Dicho lenguaje nace del BASIC (Beginner's All-purpose Symbolic Instruction Code) que fue creado en su versión original en el Dartmouth College, con el propósito de servir a aquellas personas que estaban interesadas en iniciarse en algún lenguaje de programación. Luego de sufrir varias modificaciones, en el año 1978 se estableció el BASIC estándar. La sencillez del lenguaje ganó el desprecio de los programadores avanzados por considerarlo "un lenguaje para principiantes".

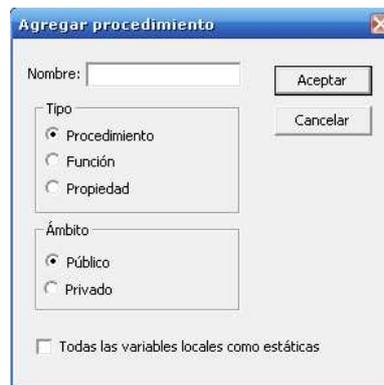
Primero fue GW-BASIC, luego se transformó en QuickBASIC y actualmente se lo conoce como Visual Basic y la versión más reciente es la 7 que se incluye en el paquete Visual Studio 7 de Microsoft. Esta versión combina la sencillez del BASIC con un poderoso lenguaje de programación Visual que juntos permiten desarrollar robustos programas de 32 bits para Windows. Esta fusión de sencillez y la estética permitió ampliar mucho más el monopolio de Microsoft. Visual Basic ya no es "un lenguaje para principiantes" sino que es una perfecta alternativa para los programadores de cualquier nivel que deseen desarrollar aplicaciones compatibles con Windows. Entre los aspectos a destacar aparece:

- Poseer un potente editor y depurador de errores



**Fig. III.2.1.1.- Ventana de Edición de Visual Basic y herramientas para la edición y depuración de código.**

- Permitir la creación de Sub y Function.

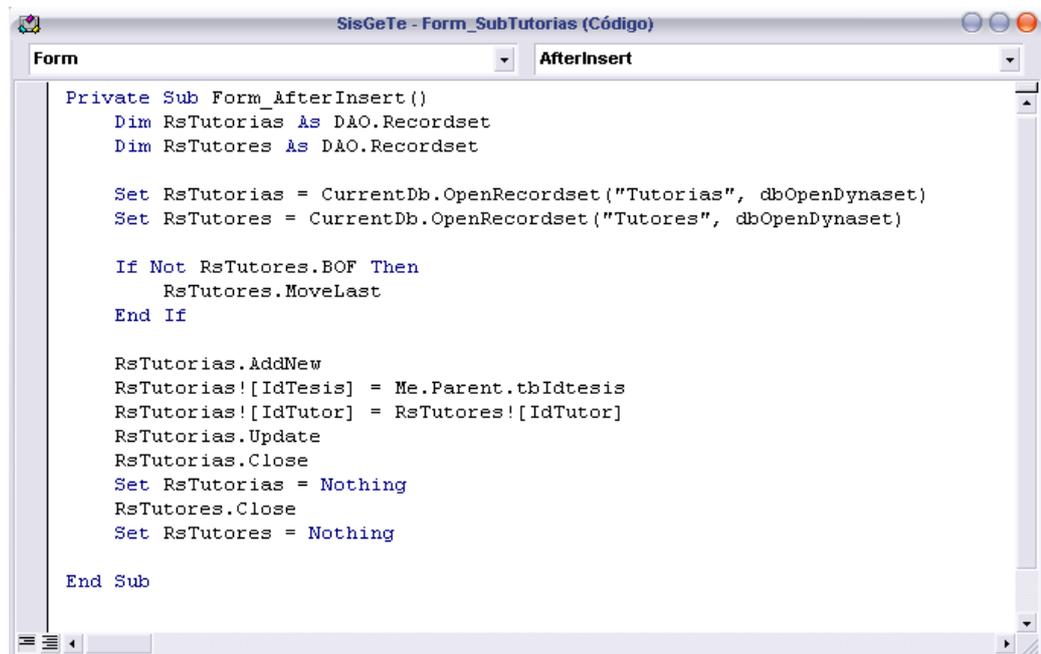


**Fig. III.2.1.2.- Ventana para insertar un procedimiento en un módulo.**

- Poseer estructuras de control, repetitivas y selectivas iguales a las de cualquier lenguaje moderno como:
  - IF ----- End If
  - Select ----- End Select
  - For ----- Next
  - Do ----- Loop
  - While ----- Whend

Ejemplo del uso de estas estructuras puede ser vistos en la figura III.2.1.1 en la que aparece el uso de las estructuras: If ----- End If y Select ----- End Select.

- Permite el trabajo con registros y campos utilizando los modelos: Objetos de Acceso de Datos (DAO) y ActiveX Data Objects (ADO).



```
Private Sub Form_AfterInsert()  
    Dim RsTutorias As DAO.Recordset  
    Dim RsTutores As DAO.Recordset  
  
    Set RsTutorias = CurrentDb.OpenRecordset("Tutorias", dbOpenDynaset)  
    Set RsTutores = CurrentDb.OpenRecordset("Tutores", dbOpenDynaset)  
  
    If Not RsTutores.EOF Then  
        RsTutores.MoveLast  
    End If  
  
    RsTutorias.AddNew  
    RsTutorias![IdTesis] = Me.Parent.tbIdtesis  
    RsTutorias![IdTutor] = RsTutores![IdTutor]  
    RsTutorias.Update  
    RsTutorias.Close  
    Set RsTutorias = Nothing  
    RsTutores.Close  
    Set RsTutores = Nothing  
  
End Sub
```

**Fig. III.2.1.3.- Ejemplo del Trabajo con registros y campos de objetos Recordset.**

En el ejemplo presentado en la figura como respuesta al evento después de insertar del Form SubTutorias (asociado con la consulta de selección TesisXTutor) es añadido un nuevo registro a la tabla Tutorías siendo el valor de la llave foránea IdTutor , el del tutor seleccionado.

- Permitir manipular los errores en tiempo de ejecución.
- Permite cambiar las propiedades de los objetos en tiempo de la ejecución.
- Permite personalizar los mensajes de MSAccess.

Por estas razones se eligió este lenguaje para desarrollar **SisGeTe**, agregándole a ellas las de poseer documentación amplia así como experiencia en su uso.

### **III.2.2.- El entorno de desarrollo de MSAccess.**

Microsoft ha embebido en todas las aplicaciones de la Suite de office el **Visual Basic para Aplicaciones** con igual potencialidad del Visual Basic, añadiendo a ello su perfecta integración con la aplicación en que se encuentra embebido, proporcionando para ello un excelente entorno de desarrollo de aplicaciones.

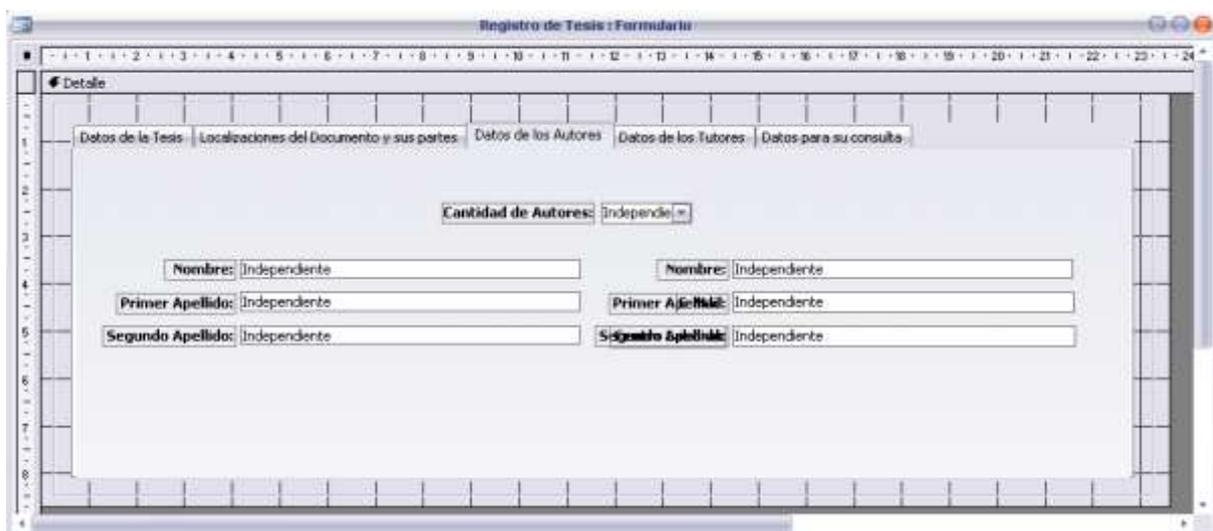
Este entorno proporciona una serie de objetos para esos fines como: formularios, Informes, páginas, macros y módulos (los dos últimos permiten dar respuestas a los eventos que ocurren en los primeros con el objetivo de integrar su trabajo y el de los controles situados sobre los mismos).



**Fig. III.2.2.1.- Entorno de desarrollo de MSAccess**

Los formularios y reportes son contenedores a su vez de controles con los cuales es posible captar y mostrar el contenido de la Base de Datos. Entre estos controles se encuentran:

- Etiqueta
- Cuadro de Texto
- Cuadro Combinado
- Botón de Comando
- Botón de Radio
- Grupo de Opciones
- Fichas
- SubFormularios
- SubReportes
- ActiveX, etc.



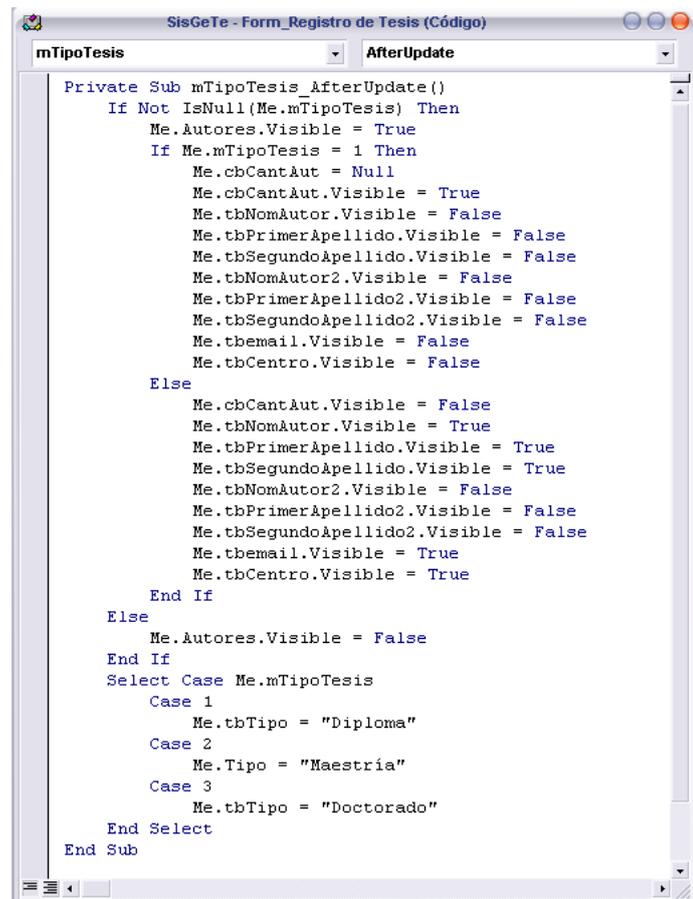
**Fig. III.2.2.2.- Ejemplo de controles contenidos en la Interfaz-Usuario del Caso de Uso “Editar Tesis”.**

En el ejemplo se puede ver el uso de controles: Ficha (que permite con sus páginas ingresar la información del documento de Tesis); Cuadro combinado (cantidad de autores), cuadro de texto (nombre, apellidos, etc.) y de etiquetas.

La coordinación del trabajo de los controles en un formulario o reporte se realiza dando respuesta a los eventos de interés ocurrido en ellos. Esta respuesta es posible darla mediante el uso:

- De código Visual Basic para Aplicaciones.

## ➤ De macros



```
SisGeTe - Form_Registro de Tesis (Código)
mTipoTesis AfterUpdate

Private Sub mTipoTesis_AfterUpdate()
    If Not IsNull(Me.mTipoTesis) Then
        Me.Autores.Visible = True
        If Me.mTipoTesis = 1 Then
            Me.cbCantAut = Null
            Me.cbCantAut.Visible = True
            Me.tbNomAutor.Visible = False
            Me.tbPrimerApellido.Visible = False
            Me.tbSegundoApellido.Visible = False
            Me.tbNomAutor2.Visible = False
            Me.tbPrimerApellido2.Visible = False
            Me.tbSegundoApellido2.Visible = False
            Me.tbemail.Visible = False
            Me.tbCentro.Visible = False
        Else
            Me.cbCantAut.Visible = False
            Me.tbNomAutor.Visible = True
            Me.tbPrimerApellido.Visible = True
            Me.tbSegundoApellido.Visible = True
            Me.tbNomAutor2.Visible = False
            Me.tbPrimerApellido2.Visible = False
            Me.tbSegundoApellido2.Visible = False
            Me.tbemail.Visible = True
            Me.tbCentro.Visible = True
        End If
    Else
        Me.Autores.Visible = False
    End If
    Select Case Me.mTipoTesis
        Case 1
            Me.tbTipo = "Diploma"
        Case 2
            Me.Tipo = "Maestría"
        Case 3
            Me.tbTipo = "Doctorado"
    End Select
End Sub
```

**Fig. III.2.2.3.- Respuesta, mediante código Visual Basic para Aplicaciones, al evento después de actualizar del control mTipoTesis del formulario “Registro de Tesis”.**

Como respuesta al evento “Después de Actualizar” del control mTipoTesis, si el valor no es nulo, entre otras acciones, se le asigna al control tbTipo el valor “Diploma”, o “Maestría” o “Doctorado” según el valor introducido en mTipoTesis.

# *CAPÍTULO IV*

## *Seguridad del Sistema y sus herramientas*

**INTRODUCCION**

En el capítulo se presenta el diseño de la seguridad de la Base de Datos del sistema, el que se implementó usando la seguridad a nivel de usuario brindada por el gestor MSAccess, se crearon los grupos y cuentas de usuario teniendo presente los roles jugados por los actores del sistema, definidos en el capítulo III, otorgándole los derechos sobre la Base de Datos y objetos de esta a los grupos. A su vez a nivel de la interfaz se limitan las acciones posibles a ejecutar por los usuarios según sea el grupo a que estos pertenecen, mediante el menú que se le muestra.

El capítulo concluye con una descripción de las herramientas que dispone el Sistema para resguardar, compactar y reparar los datos, y mantener actualizado el registro de cuentas de usuario y contraseñas.

### IV.1.- La seguridad de SisGeTe y su implementación.

El Sistema de Seguridad de **SisGeTe** se implementó usando la seguridad por usuario ofrecida por el gestor MSAccess. La manera más fácil es haciendo uso del “Asistente para Seguridad”, el cual se inicia como se muestra en la Figura IV.1.1

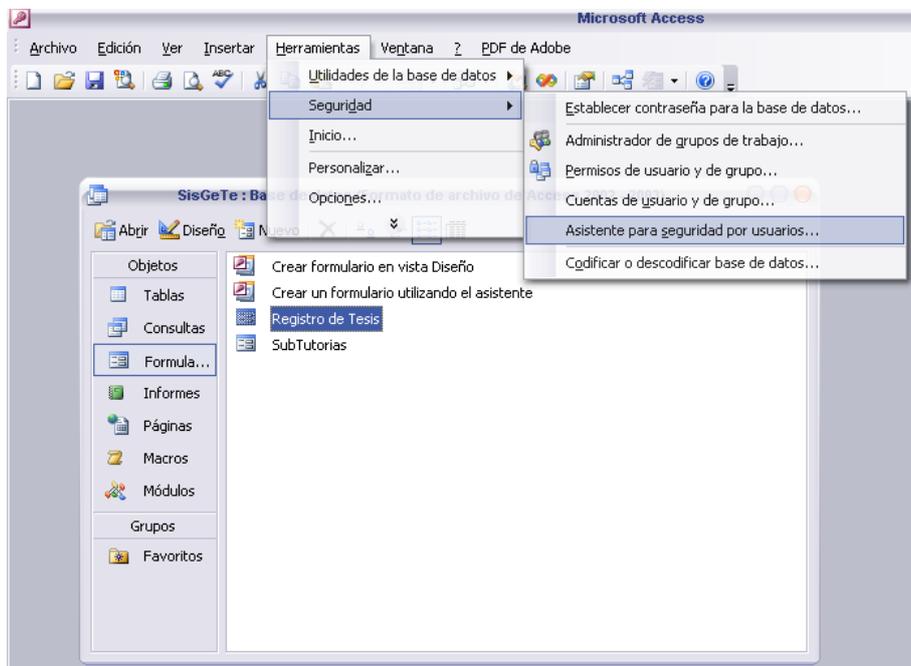
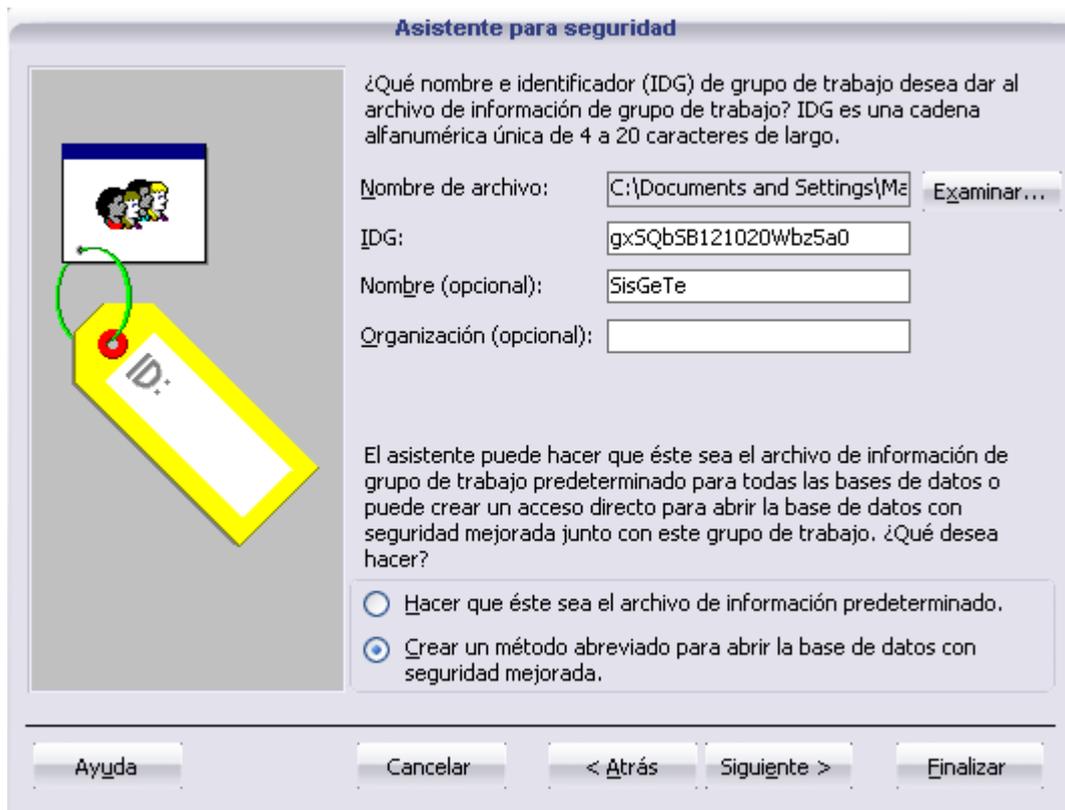


Figura IV.1.1.- Iniciar el Asistente para Seguridad.

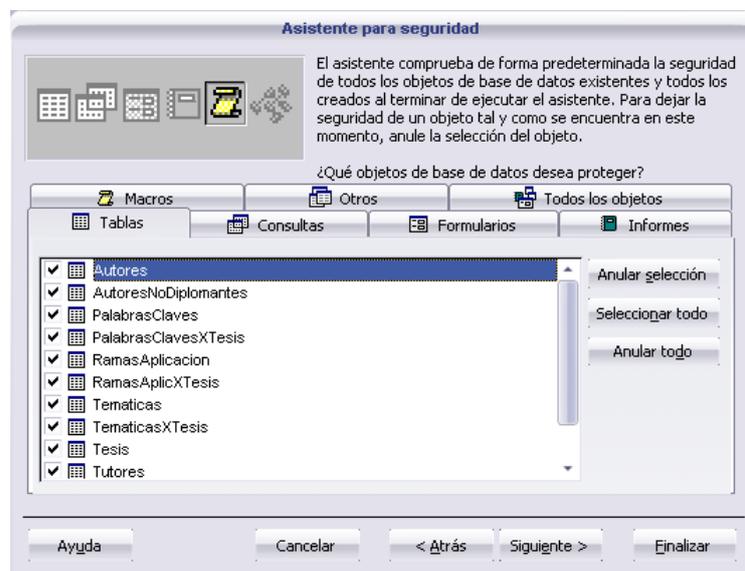
El trabajo del Asistente comienza captando los datos para crear el **Grupo de Trabajo**, vea figura IV1.2., este es un fichero con extensión **mdw** que contiene las cuentas de Grupos y de Usuarios establecidas para la Base de Datos.

El Grupo de Trabajo puede también ser creado, sin uso del asistente, seleccionando la opción “Administrador de Grupos de Trabajo” del menú de seguridad (vea figura IV.1.1).



**Figura IV.1.2.- Creando el Grupo de Trabajo de SisGeTe**

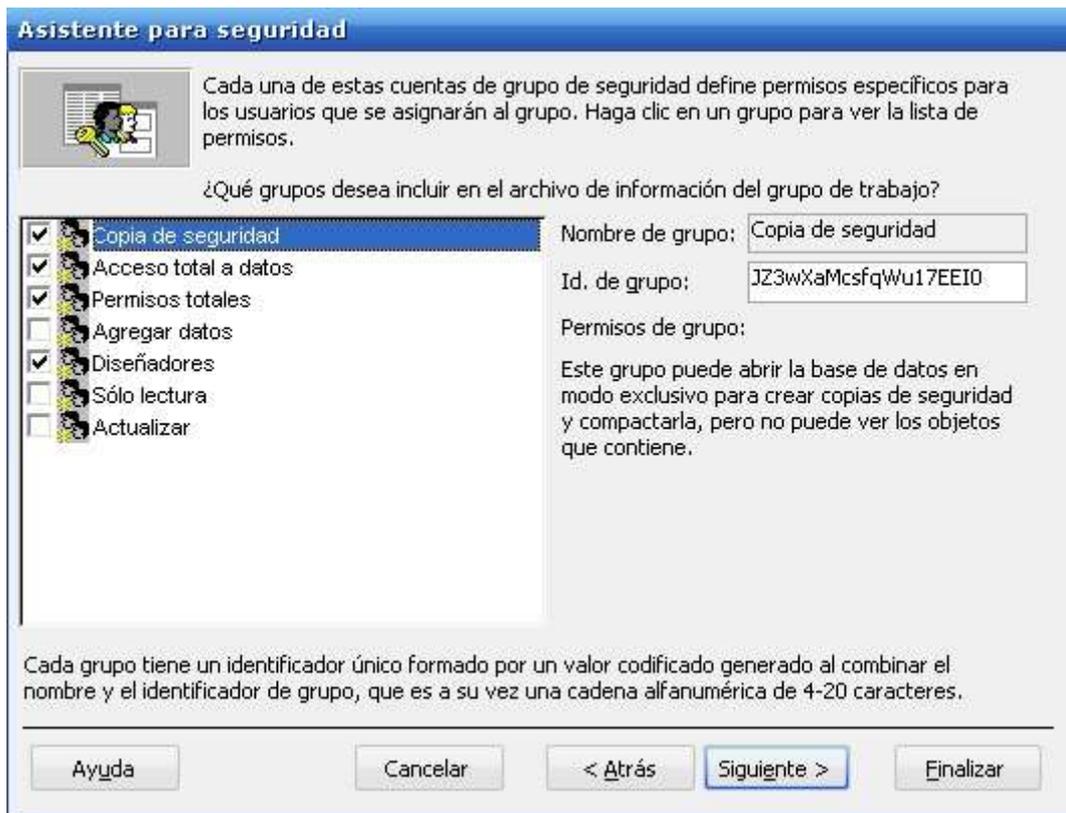
Usando la pantalla mostrada la figura IV.1.3 se selecciona los objetos de la Base de Datos que deseamos proteger.



**Figura IV.1.3.- Seleccionando los objetos que serán protegidos de la Base de Datos.**

A continuación es seleccionada las cuentas de Grupos ofrecidos por Access a incluir en el Grupo de Trabajo. Vea estas cuentas en la figura IV.1.4.

Es posible crear cuentas de Grupos propias, además de las brindadas por Access, empleando la opción "Cuentas de usuarios y de grupo" del menú Seguridad (vea figura IV.1.1), en el caso de SisGeTe por esta vía se crearon los grupos: Consultantes, Registradores y Analistas.



**Figura III.4.4.- Seleccionando las cuentas de Grupos de Seguridad de SisGeTe.**

Los permisos otorgados por Access a cada una de estas cuentas de Grupos, sobre los objetos y datos de la Base de Datos son mostrados en la tabla siguiente:

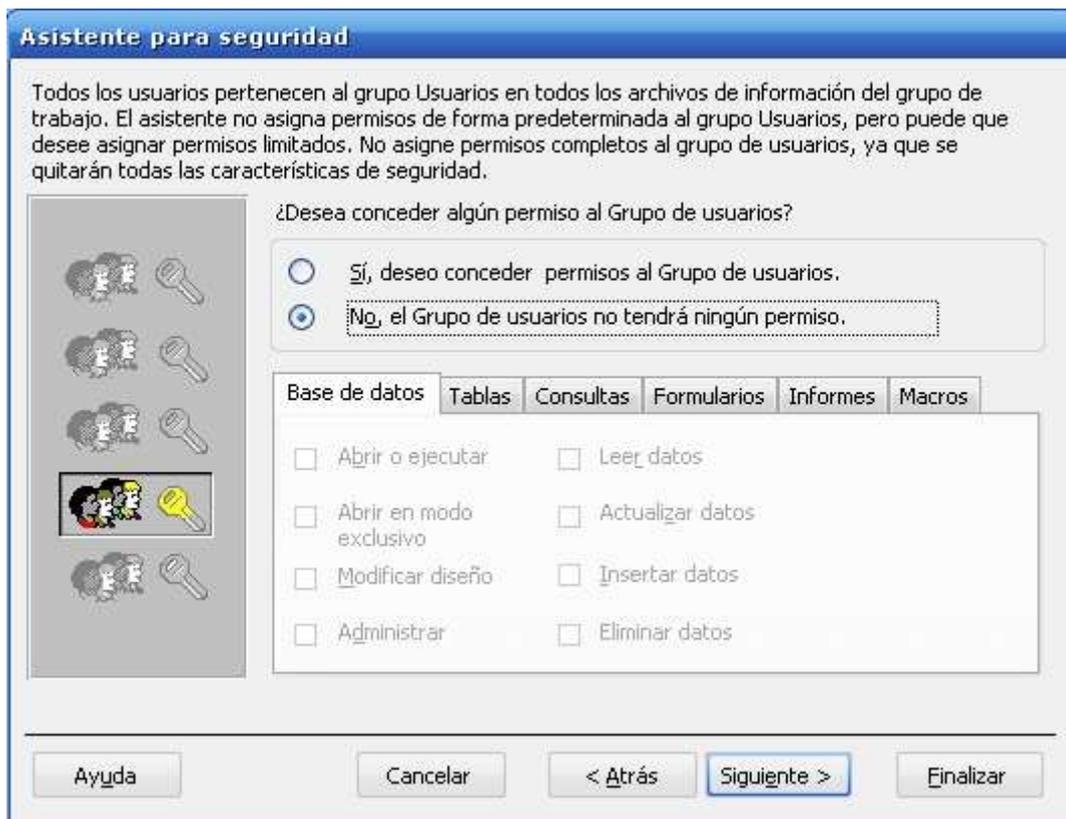
<b>Cuenta de Grupo de Seguridad</b>	<b>Permisos</b>
Copia de Seguridad	Abrir la base de Datos en Modo Exclusivo (Crear copias de Seguridad; Compactarla), pero no puede ver los objetos.
Acceso total a Datos	Acceso total a los datos, pero no puede cambiar el diseño de los objetos.
Permisos Totales	Permisos totales sobre los objetos de Datos, pero no puede asignar permisos a otros usuarios.
Agregar Datos	Puede leer e insertar datos, pero no puede cambiar diseño de ningún objeto ni eliminar y actualizar datos.
Diseñadores	Tiene permisos totales para modificar datos y todos los objetos, pero no tablas y relaciones.
Solo Lectura	Puede leer todos los datos pero no puede modificar datos ni el diseño de los objetos de la Base de Datos.
Actualizar	Puede leer y actualizar datos, pero no puede modificar el diseño de los objetos de la Base de Datos, ni insertar e eliminar datos.

En el caso que creamos nuestras propias cuentas de Grupos podemos otorgarle los permisos a estos usando la opción “Permisos de usuarios y de grupo” del menú Seguridad.

Todas las Bases de Datos de Access poseen las cuentas de Grupo:

- **Administradores.-** Sus cuentas tendrán derecho a administrar tanto la Base de Datos como a sus objetos pudiendo cambiar y asignar permisos sobre estos a otros usuarios así como crear y eliminar cuentas de Grupos y de Usuarios.

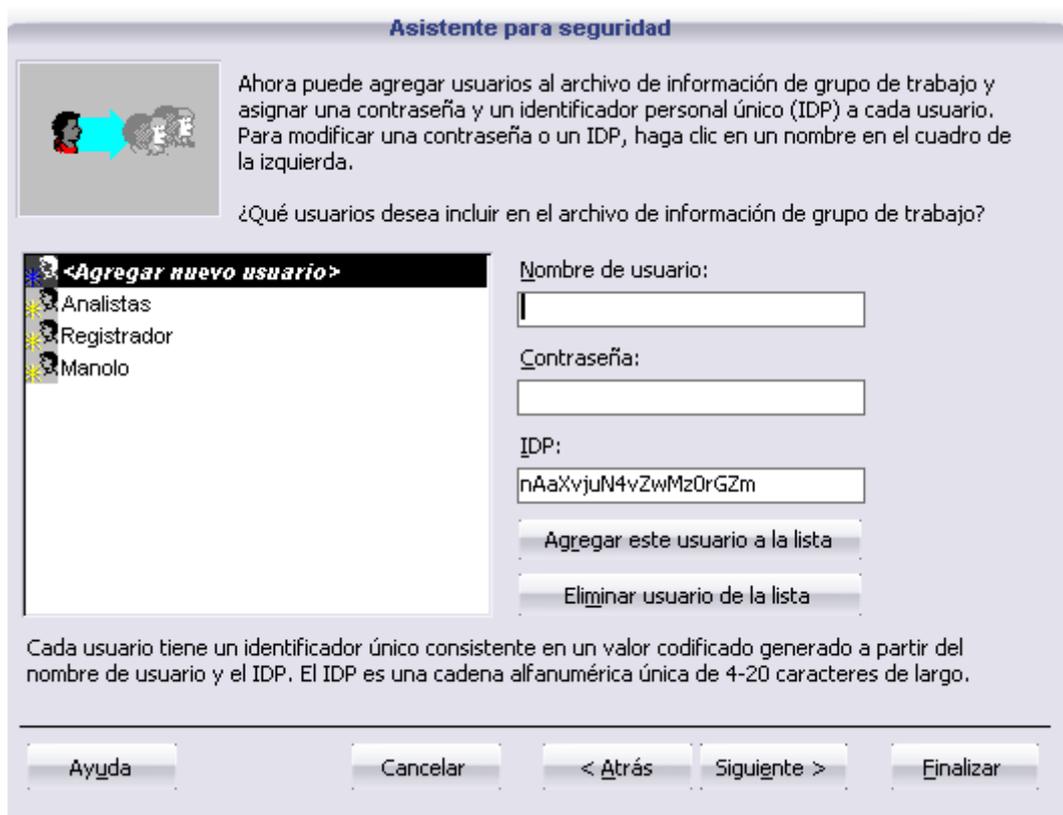
- **Usuarios.-** A este grupo pertenecen todas las cuentas de Usuarios existentes en la Base de Datos. Por defecto Access otorga a este grupo todos los permisos sobre la base de Datos y sus objetos, luego por lógica será necesario a este grupo quitarle todos los permisos pues de lo contrario toda cuenta disfrutara de todos los derechos. Vea en la siguiente figura como desde el asistente le es quitado los permisos el grupo “Usuarios”.



**Figura IV.1.5.- Quitando los permisos al Grupo “Usuarios”.**

El siguiente paso del Asistente para Seguridad es definir las cuentas de Usuario a incluir en el Grupo de Trabajo. En la figura IV.1.6 aparece la pantalla que el Asistente ofrece para estos fines.

Las cuentas de Usuario pueden crearse sin el asistente empleando la opción “Cuentas de usuarios y de grupo” del menú Seguridad (vea figura IV.1.1).

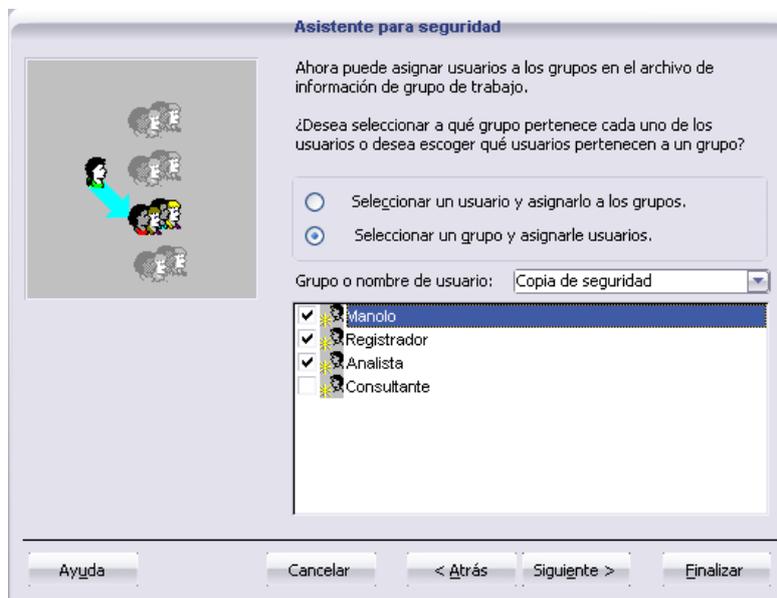


**Figura IV.1.6.- Cuentas de Usuarios de SisGeTe.**

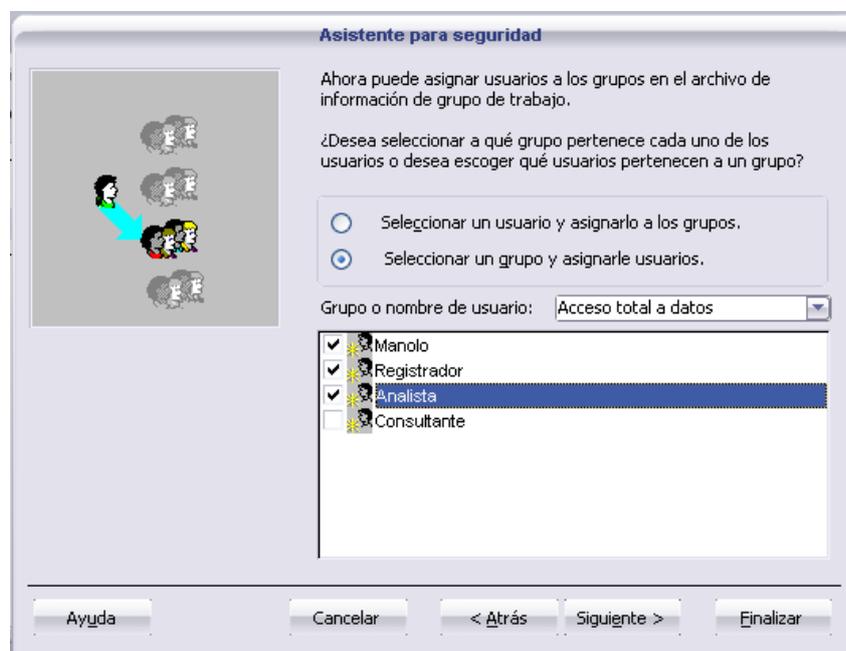
Todas las Bases de Datos de Access posee la Cuenta de Usuario **Administrador** la que pertenece a los Grupos Administradores y Usuarios, no teniendo la misma contraseña. Con esta cuenta es que se entra por defecto en las Bases de Datos de Access al no tenerse aun implementada la seguridad por usuario. Como es una cuenta común para todas las Bases de Datos de Access es conveniente quitarle todos los derechos y en su lugar crear una cuenta nueva que tenga como rol el de administrar o sea que pertenezca al grupo administradores.

El Asistente para Seguridad, al implementar la seguridad por usuario, le asigna a la cuenta Administrador una contraseña. Esto también puede hacerse empleando la opción "Cuentas de usuarios y de grupo" del menú Seguridad (vea figura IV.1.1).

Los permisos a dar a las cuentas de usuarios es más cómodo realizarlo haciendo pertenecer estas cuentas a un Grupos de Seguridad, de esta forma tomaran los permisos que tengan los grupos a los que pertenecen, bastando con ello otorgar los permiso al grupo y no a cada una de las cuentas. Podemos ver en las figuras siguientes los grupos a los que pertenecen cada cuenta de usuario de SisGeTe.



**Figura IV.1.7.- Cuentas de Usuarios pertenecientes al grupo “Copia de Seguridad”.**



**Figura IV.1.8.- Cuentas de Usuarios pertenecientes al grupo “Acceso total a datos”**



Figura IV.1.9.- Cuentas de Usuarios del Grupo de Seguridad “Permisos Totales”



Figura IV.1.10.- Cuentas de Usuarios pertenecientes al grupo “Diseñadores”



**Figura IV.1.11.- Cuentas de Usuarios pertenecientes al grupo “Administradores”**

La asignación de las cuentas de usuario a las de Grupos puede ser realizada sin el asistente empleando la opción “Cuentas de usuarios y de grupo” del menú Seguridad (vea figura IV.1.1).

También es posible con uso de la opción “Permisos de usuarios y de grupo” del menú Seguridad otorgar permisos sobre la Base de Datos y sus objetos a las Cuentas de Usuarios.

El asistente crea, si se desea, una copia de resguardo de la Base de Datos sin proteger como se muestra en la figura IV.1.12.



**Figura IV.1.12.- Creando una copia de seguridad desprotegida de la Base de Datos.**

Finalmente se genera un Informe de Seguridad en el cual aparece toda la información necesaria para volver a crear al archivo de Grupo de trabajo, en caso que este se elimine por descuido, y así poder recuperar el acceso a la Base de Datos protegida. En el contenido de este informe aparece:

- Localización de la Base de Datos desprotegida (copia de resguardo creada).
- Localización de la Base de Datos protegida.
- Localización del Grupo de Trabajo y su Identificador.
- Objetos de la Base de Datos protegida.
- Cuentas de Grupo de Seguridad con su Identificador y las cuentas de Usuario que pertenecen al grupo.
- Cuentas de Usuario con su Identificador, contraseña y cuentas de Grupos de Seguridad a las que pertenece.

Implementada la seguridad en la Base de Datos al esta ser iniciada aparece la ventana de inicio de sesión que solicitará el usuario y su contraseña de conexión, como se muestra en la figura IV.1.13.

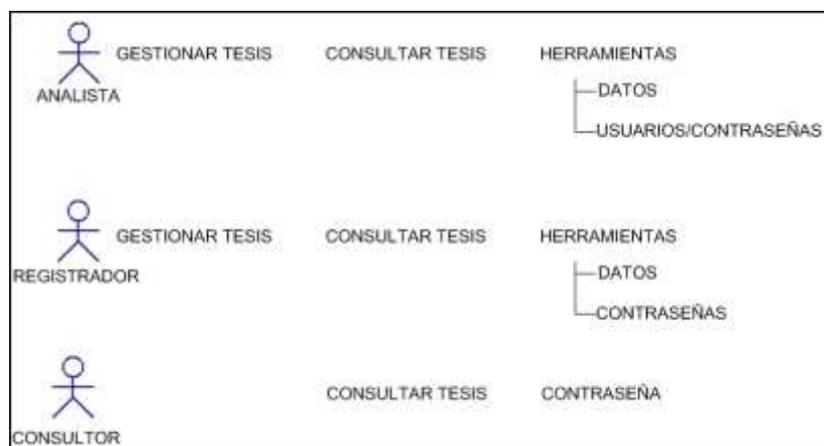


**Figura IV.1.13- Ventana Inicio de Sesión.**

A las cuentas de Grupos tomadas ofrecidas por Access, se añadieron las de:

- Analistas
- Registradores
- Consultores

Estas fueron creadas con la opción “Cuentas de usuario y de Grupo” del menú Seguridad (vea figura IV.1.1). Su objetivo fue el de identificar el usuario que inicio SisGeTe con vista establecer su comportamiento, mediante código, en las interfases mostrándole un menú acorde al rol por el jugado.



**Figura IV.1.14- Menú mostrado según usuario que inicio la sección**

**IV.2.- Herramientas del sistema.**

Estas se dividen en dos:

- Las orientadas a los datos.
- Las orientadas a los registros de cuentas de usuarios y contraseñas.

La primera va dirigida a garantizar la protección de los datos de la Base de Datos SisGeTe\_Datos, permitiendo:

- Realizar copias de resguardo de la Base de Datos.
- Vincularse a la Base de Datos deseada (recuperar una copia de resguardo)
- Compactar la Base de Datos evitando así crecimiento indeseado de esta al eliminar los espacios quedados en blancos después de realizarse diversas operaciones de actualización de los datos.
- Recuperar la Base de Datos dañado al no ser cerrada adecuadamente por corte de fluido eléctrico.

La interfaz utilizada para realizar estas operaciones sobre los datos es la que se mostró en el capítulo III (figura III.1.12).

La herramienta para gestionar las cuentas de usuario y sus contraseñas permite:

- Crear y eliminar cuentas de usuarios de la aplicación. Otorgando los derechos haciendo pertenecer la cuenta a los grupos creados: Analistas, Registradores, Consultantes y Administradores.
- Cambiar la contraseña de inicio de sección por cada usuario.
- Borrar la contraseña de cualquier usuario, si la cuenta pertenece al grupo administradores.

Las interfaces ofrecidas para realizar estas acciones fueron mostradas en el capítulo III (figuras III.1.14 y III.1.15).

# CONCLUSIONES

Con la investigación se logró:

- Obtener una Base de Datos la cual contiene la información necesaria para realizar las búsquedas y consultas de los documentos de tesis gestionados por el Departamento de Informática de la UPR.
  
- Desarrollar la aplicación **SisGeTe** que permite al usuario gestionar con eficiencia la Base de Datos teniendo en cuenta las reglas del negocio y satisfaciendo sus requerimientos.

# RECOMENDACIONES

- Darle el mantenimiento a la aplicación con vista a corregir los posibles errores que puedan surgir durante su explotación.
  
- Hacer uso de la aplicación SisGeTe para gestionar las tesis del departamento de Informática incorporando para ello la información de los documentos de tesis existentes.
  
- Recomendar el uso de SisGeTe a otros departamentos de la UPR:

*REFERENCIAS  
BIBLIOGRÁFICAS*

1. **[ORTIZ,2007]** Ortiz, Amores, Santiago J., Lascano, Tapia, Juan C., “Sistema para la Gestión del Departamento Docente (SisGeDeDoc)”. Proyecto de Diploma presentado en opción al título de Ingeniero en Informática, Universidad Pinar del Río, 2007.
2. **[COCOMO,1999]** Ruiz, González, Francisco, de la Fuente Moya Antonio COCOMO V2 Modelo de Estimación de Costos para Proyectos de Software. Universidad de Castilla – La Mancha, Mayo 1999.
3. **[BOHEM,2000]** Bohem, B.W. “SW Const Estimation with COCOMO II”, Prentice Hall, 2000.
4. **[DATE,1994]** Date, C, An Introduction to Database System, Addison Wesley, 6ta Edición, 1994, 839 páginas.
5. **[LOPEZ,2001]** Vázquez, López, Manuel, Base de Datos, Universidad Politécnica de Valencia, 2001, 112 páginas.
6. **[CASTRO,2006]** Castro, Jesús Antonio. BASES DE DATOS I, Unidad II: MODELO ENTIDAD RELACION, Instituto Tecnológico de la Paz.  
[http://www.itlp.edu.mx/publica/tutoriales/basedat2/hdos\\_1.htm](http://www.itlp.edu.mx/publica/tutoriales/basedat2/hdos_1.htm).  
Fecha consulta: Febrero 2008
7. **[CLAUDIO,1999-2004]** Claudio, Casares, Base de Datos  
[http://www.itlp.edu.mx/publica/tutoriales/basedat1/tema5\\_1.htm](http://www.itlp.edu.mx/publica/tutoriales/basedat1/tema5_1.htm)  
Fecha consulta: Febrero 2008.
8. **[HERRERA,2004]** Herrera, Alvaro. EL MODELO ENTIDAD RELACION, 2004  
<http://alvherre.atentus.cl/modBasico/node3.html>.  
Fecha consulta: Febrero 2008.
9. **[MURILLO,2006]** *Herrera, Gabriela, Murillo, Mirian, Sistema Automatizado de Información para la Comunidad Estudiantil Universitaria de la UPR (SAICEU), Proyecto de Diploma en opción al título de Ingeniero en informática, Universidad de Pinar del Río, 2006*
10. **[SYBASE, 2006]**  
<http://www.sybase.com/products/developmentegration/pawerdesigner>  
Fecha: Febrero 2008.

11. **[BOOCH,2000]** BOOCH, Grady, RUMBAUGH, James, JACOBSON, Ivar; 2000. "El Lenguaje Unificado de Modelado". Addison Wesley, 2000,
12. **[JACOBSON, 2000]** Jacobson, I.; Booch, G. y Rumbaugh, J.; "El Proceso Unificado de Desarrollo de software", Addison-Wesley, 2000.
13. **[PRESSMAN,2002]** Presuman, Roger, 2002. Ingeniería de Software. Un enfoque práctico, McGraw.Hill/Interamericana de España, 824 páginas.
14. **[BERTAMI,2007]** Bertamí Hernández K, "Software para el Marketing Forestal", Proyecto de Diploma presentado en opción al título de Ingeniero en Informática, Universidad de Pinar del Río, 2007
15. **[FORTE,1999]** Forte, Stephen, Howe, Tom, Ralston, James, Access 2000 Development, Editorial SAM, 1999, 877 páginas.
16. **[SCOUT,1999]** Barrer, Scout, F, Access 2000 Power Programming, Editorial SAM, 1999, 1333 páginas.
17. **[Microsoft,1995]** Microsoft Corporation, Access 7 Language Reference, 1995, 953 páginas.
18. **[Microsoft Corporation,1995]** Microsoft Corporation, Creación de Aplicaciones con Microsoft Access para Windows 95, 571 páginas.
19. **[Pablog3, 2000]** Evaluación del producto Microsoft Visual Basic 6.0 por pablog3 Junio, 2000.
20. **[SANTANA,2005]** Evaluación del producto Microsoft Visual Basic 6.0 por Santana82 17.09.2005
21. **[NACHT,2001]** Evaluación del producto Microsoft Visual Basic 6.0 por Nacht Octubre, 2007
22. **[EIGUILLE,2002]** La Web de Visual Basic, C#, Net y más...  
<http://www.elguille.info>  
Fecha consulta: Marzo 2008
23. **[MSDN,2003]** MSDN Library de Microsoft, 2003

# *ANEXOS*

## Tablas de SisGeTe y su Estructura

No	Entidades	Elemento de Datos (Atributos)	Llave primaria	Llave Foránea	Dominio del Elemento de Datos. (Tipo de Dato)
01	Autores	Id_Autor Nombre Apellido1 Apellido2	Id_Autor		Doble Texto Texto Texto
02	AutoresNoDiplomados	Id_Autor Centro de Trabajo e-mail	Id_Autor	Id_Autor	Doble Texto Texto
03	Palabras Claves	Id_Palabra Palabra	Id_Palabra	Id_Palabra	Doble Texto
04	PalabrasClavesXTesis	Id_Palabra Id_Tesis	Id_Palabra	Id_Palabra Id_Tesis	Autonumérico Entero Largo
05	Ramas Aplicación	Id_Rama RamaAplicación	Id_Rama		Doble Texto
06	RamasAplicXTesis	Id_Rama Id_Tesis	Id_Rama	Id_Tesis	Autonumérico Entero Largo
07	Temáticas	Id_Tematica Temática	Id_Tematica		Entero Largo Texto (250)
08	TemáticasXTesis	Id_Tematica Id_Tesis	Id_Tematica	Id_Tesis	Entero largo Entero largo
09	Tesis	Id_Tesis Titulo Mes Año Tipo Localización Introducción Resumen Conclusiones Recomendaciones	Id_Tesis		Entero largo Texto Texto Texto Texto Texto Texto Texto Texto Texto
10	Tutores	Id_Tutor Tutor E-mail	Id_Tutor		Entero largo Texto Texto

		Centro de Trabajo ProfeDpto CCientífica			Texto Texto Texto
11	Tutorías	Id_Tesis Id_Tutor	Id_Tesis Id_Tutor	Id_Tesis Id_Tutor	Entero largo Entero largo