

ESCUELA POLITÉCNICA DEL EJÉRCITO

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

**ANÁLISIS, DISEÑO E IMPLANTACIÓN DEL SISTEMA DE
ADMINISTRACIÓN ACADÉMICA Y FINANCIERA DEL
PROYECTO “LA ESPE COMO ACADEMIA REGIONAL DE
CISCO”.**

**Previa a la obtención del Título de:
INGENIERO EN SISTEMAS E INFORMÁTICA**

POR:

JUAN PABLO QUEREMBÁS ALTAMIRANO

SANGOLQUÍ 12 de Noviembre del 2008

CERTIFICACIÓN

Certifico que el presente trabajo fue realizado en su totalidad por el Sr. JUAN PABLO QUEREMBÁS ALTAMIRANO como requerimiento parcial a la obtención del Título de INGENIEROS EN SISTEMAS E INFORMÁTICA.

12 de Noviembre del 2008

Ing. Mauricio Campaña, Director de Tesis

DEDICATORIA

El presente trabajo lo dedico a mi Padre Manuel que con su ejemplo y dedicación a sido mi aliento día a día, a mi Madre Emmita que con su incansable labor ha sido mi apoyo y refugio en los momentos más difíciles de mi vida, a mi Esposa Gabriela mi compañera y amiga incondicional, a mis Hijos Lesly Stefania y Pablo Andrés por ser mis mas grandes amigos y compañeros, son todos a quienes les debo el poder culminar mi carrera.

No me queda más que decirles que Dios les pague.

Juan P.

AGRADECIMIENTO

Agradezco a Dios por darme la oportunidad de culminar una etapa mas de mi vida y empezar otra con la mejor preparación recibida durante todos estos años en lamedor institución del Ecuador.

Agradezco a mi querida Escuela Politécnica del Ejército, en especial a la Facultad de Ingeniería en Sistemas, facultad que auspició el desarrollo del proyecto.

A todas las personas que de una u otra manera han contribuido a la culminación de este proyecto, en especial al Ing. Mauricio Campaña.

Juan P.

ÍNDICE DE CONTENIDOS

Introducción.....	1
CAPITULO I.....	3
Generalidades.....	4
1.1 Antecedentes.....	4
1.2 Situación Actual.....	5
1.3 Justificación.....	5
1.4 Alcance.....	6
1.5 Objetivos.....	7
1.5.1 Objetivo General.....	7
1.5.2 Objetivos Específicos.....	8
1.6 Factibilidad.....	8
1.6.1 Factibilidad Técnica.....	8
1.6.2 Factibilidad Operativa.....	9
1.6.3 Factibilidad Económica.....	9
CAPITULO II.....	11
Marco Teórico.....	12
2.1 Paradigmas de la I.S.....	13
2.1.1 Modelo en Cascada.....	13
2.1.2 Modelo de Construcción de Prototipos.....	17
2.1.3 Modelo en Espiral.....	19
2.2 Metodología Orientado a Objetos.....	21
2.2.1 Metodología de Desarrollo OMT.....	24
2.2.1.1 Fases de la Metodología OMT.....	25
2.3 Técnica de Análisis y Diseño.....	25
2.3.1 Lenguaje Unificado de Modelado UML.....	25
2.3.1.1 Tipos de Diagramas UML.....	27
2.3.1.2 Elementos UML.....	30
2.3.1.3 Conectores UML.....	32
2.4 Herramientas Case.....	33
2.4.1 Componentes de una Herramienta Case.....	35
2.4.2 Características de una Herramienta Case.....	36
2.4.3 Herramientas Case CISCO – ESPE.....	37
2.4.3.1 Sybase.....	38
2.4.3.2 Power Designer.....	38
2.4.3.3 Power Designer 6.1.....	39
2.4.3.4 Power Buildes.....	40
CAPITULO III.....	41
Desarrollo de la Aplicación.....	42
3.1 Introducción.....	42
3.2 Obtención de Requerimientos.....	42
3.2.1 Definiciones.....	43
3.3 Requisitos de Interfaces Externas.....	44
3.3.1 Interfaces de Usuario.....	44
3.3.2 Interfaces de Hardware.....	45

3.3.3 Interfaces de Software.....	45
3.3.4 Interfaces de Comunicaciones.....	46
3.4 Análisis.....	47
3.4.1 Introducción.....	47
3.4.2 Definiciones.....	48
3.4.3 Especificación de Requerimientos Funcionales.....	49
Subsistema Académico.....	50
Subsistema Financiero.....	59
Subsistema de Seguridades.....	64
3.4.4 Seguridad.....	67
3.4.5 Mantenimiento.....	67
3.5 Diseño del Sistema CISCO – ESPE.....	68
3.5.1 Introducción.....	68
3.5.2 Mapa de Procesos.....	68
3.5.3 DFD'S.....	71
3.5.4 Diagramas Conceptuales.....	74
3.5.5 Casos de Uso.....	77
3.5.5.1 Especificación de Actores.....	77
3.5.5.2 Estructura de Casos de Uso.....	81
3.5.5.3 Destalle de Casos de Uso.....	89
3.5.5.4 Diagramas de Caso de Uso.....	99
3.5.5.4.1 Administración de Usuarios.....	99
3.5.5.4.2 Validación de Usuarios.....	100
3.5.5.4.3 Ingreso Academia Regional.....	100
3.5.5.4.4 Ingreso Academia Local.....	101
3.5.5.4.5 Ingreso Proyectos.....	101
3.5.5.4.6 Ingreso de Ciudadanos.....	102
3.5.5.4.7 Ingreso de Docentes.....	102
3.5.5.4.8 Ingreso de Estudiantes.....	103
3.5.5.4.9 Ingreso de Proveedores.....	103
3.5.5.4.10 Registro de Cursos.....	104
3.5.6 Diccionario de Datos Procesos.....	105
3.5.7 Modelo de Datos de Nivel Lógico.....	105
3.5.8 Diccionario de datos Modelo Lógico.....	105
3.5.9 Modelo de datos del Nivel Físico.....	105
3.5.10 Diccionario de Datos Modelo Físico.....	105
3.5.11 Scrip de creación de la Base de Datos.....	105
3.5.12 Base de Datos.....	105
3.5.13 Manual Técnico.....	105
3.6 Implementación.....	106
Creando una Aplicación en 3 Capas.....	106
Asignación de la clase al Modulo en P.W.....	112
Creación del EASERVER Profile de Jaguar.....	113
Instalación de un Paquete en el Servidor.....	118
Creación de un Componente en el Servidor.....	120
Creación de un Componente en P.W.....	123
Creando una Aplicación en P.W.....	137

Creando un Objeto.....	143
Creando la Interfaz de la Aplicación Cliente.....	148
Creación del Menú Principal del Sistema.....	154
Creación de Eventos.....	154
Creación de Ventanas.....	155
Funciones para el manejo de Datawindows.....	156
Creación de Ventanas de Datos.....	158
Selección de Tipos y Presentación.....	158
Modos de Trabajo.....	159
Tipos de Presentación.....	159
Funciones de Datawindows.....	160
Creación de una pagina WEB con Componentes.....	163
3.7 Manual de Usuario.....	169
3.7.1 Introducción.....	169
3.7.2 Requerimientos Técnicos.....	169
3.7.3 Interfaces de Usuario.....	170
3.7.4 Interfaces de Hardware.....	170
3.7.5 Interfaces de Software.....	170
3.7.6 Interfaces de Comunicación.....	171
3.7.7 Ambiente Cliente – Servidor.....	171
3.7.8 Pasos para el ingreso del sistema.....	172
CAPITULO IV.....	194
Pruebas y Evaluación de Resultados.....	195
4.1 Pruebas de Funcionamiento.....	195
4.2 Evaluación Técnica.....	196
4.3 Evaluación Económica.....	197
4.3.1 Materiales Directos.....	198
4.3.1.1 Herramientas de Análisis y Diseño.....	198
4.3.1.2 Herramientas de Desarrollo.....	199
4.3.2 Mano de Obra Directa.....	199
4.3.3 Costos Indirectos.....	200
4.4 Análisis de Resultados.....	200
Conclusiones y Recomendaciones.....	201
CAPITULO V.....	203
Biografía.....	204
5.1 Libros.....	204
5.2 Documentos.....	204
5.3 Enlaces.....	204
Hoja de legalización de Firmas.....	206

ÍNDICE DE TABLAS

Tabla 1.1 Factibilidad Económica.....	10
Tabla 3.1 Ambiente Cliente Servidor.....	46
Tabla 3.2. Mapa de Procesos.....	70
Tabla 3.3 Definición de Herramientas.....	136
Tabla 3.4 Descripción de Eventos.....	155
Tabla 3.5 Definición de Ventanas.....	156
Tabla 3.6 Modos de Trabajo.....	159
Tabla 3.7 Tipos de Presentación.....	160
Tabla 3.8 Funciones de Datawindows.....	162
Tabla 3.9 Ambiente Cliente – Servidor.....	172
Tabla 3.10 Descripción de Eventos.....	178

ÍNDICE DE FIGURAS

Figura 2.1. Modelo en Cascada.....	14
Figura 2.2 Modelo de Construcción de Prototipos.....	18
Figura 2.3 Modelo en Espiral.....	20
Figura 2.4 Tipos de Diagramas.....	28
Figura 2.5 Diagramas de Comportamiento.....	29
Figura 2.6 Representación de una Clase.....	30
Figura 2.7 Representación de un Objeto.....	30
Figura 2.8 Representación de una Interfaz.....	30
Figura 2.9 Representación de un Caso de Uso.....	31
Figura 2.10 Representación de un Componente.....	31
Figura 2.11 Representación de un Nodo.....	31
Figura 2.12 Conectores de Diagramas de Comportamiento.....	33
Figura 3.1 Modelo de Negocio.....	71
Figura 3.2 Modelo de Negocio.....	72
Figura 3.3 Modelo de Negocio.....	73
Figura 3.4 Diagrama Conceptual Académico.....	74
Figura 3.5 Diagrama Conceptual Financiero.....	75
Figura 3.6 Diagrama Conceptual Seguridades.....	76
Figura 3.7 Diagrama de Caso de Uso Usuario.....	81
Figura 3.9 Diagrama de Caso de Uso Supervisor.....	83
Figura 3.11 Diagrama de Caso de Uso Administrador.....	85
Figura 3. 15Administración de Usuarios.....	99
Figura 3.16 Validación de Usuarios.....	100
Figura 3.17 Ingreso Academia Regional.....	100
Figura 3.18 Ingreso Academia Loca.....	101
Figura 3.19 Ingreso de Proyectos.....	101
Figura 3.20 Ingreso de Ciudadanos.....	102
Figura 3.21 Ingreso de Docentes.....	102
Figura 3.22 Ingreso de Estudiantes.....	103
Figura 3.23 Ingreso de Proveedores.....	103
Figura 3.24 Ingreso de Estudiantes.....	104
Figura 3.25 Figuras de Implementación del Sistema.....	106
Figura 3.107 Creación del Menú Principal del Sistema.....	154
Figura 3.118 Manual de Usuario.....	172

INTRODUCCIÓN

El desarrollo de software constituye hoy en día un elemento muy importante en el proceso de implantar sistemas de información, es por esta razón que países como Estados Unidos, Francia o Inglaterra se han posicionado como potencias mundiales debido a la eficiencia y eficacia con las que procesan los diferentes tipos de información, dando de esta manera un alto estándar de vida para la sociedad, quienes se sienten satisfechos a la hora de ser atendidos en cualquier institución pública o privada.

El Ecuador en su afán de ser un país competitivo posee instituciones de gran prestigio educativo tal es el caso de la Escuela Politécnica del Ejército que en conjunto de sus diferentes sedes y facultades han formado profesionales e investigadores de excelencia, con capacidad de liderazgo y alta conciencia ciudadana, razón por la cual, la Facultad de Sistemas e Informática (FISI) consciente de la importancia de su misión de Desarrollo hacia el País y las Fuerza Armadas, ha emprendido un proceso de capacitación que permitirá cumplir con nuevos programas educativos en beneficio de la comunidad politécnica y público en general.

El manejo correcto de la educación facilitará el cumplimiento de la misión y objetivos de la ESPE, razón por lo cual se ha visto empeñada en calificar como Academia Regional de CISCO, y se encuentra en un proceso de desarrollo del Sistema de Administración Académica y Financiera del Proyecto, el mismo que a mediano plazo permitirá satisfacer los requerimientos de información y logrará una toma de decisiones ágil, oportuna y con un alto grado de certeza.

El tema de Tesis ha desarrollarse previo a obtener el título de “Ingeniero de Sistemas e Informática” es: “Análisis, Diseño e Implantación del Sistema de Administración Académica y Financiera del Proyecto ESPE como Academia Regional de CISCO”, la estructura del proyecto responde al formato general de la

Facultad. En términos generales, se definen los antecedentes teóricos del tema, la situación actual y la Justificación del trabajo. Seguidamente se establecen los objetivos y las metas, así como la estructura tentativa del temario. De igual manera, se describe la metodología de trabajo, las actividades y la Factibilidad del mismo.

CAPITULO I

GENERALIDADES

GENERALIDADES

1.1. ANTECEDENTES

La Escuela Politécnica del Ejército fue creada en 1922 y tiene como objetivos básicos la modernización de la administración, **la innovación del sistema educativo**, el impulso de la investigación científica, el desarrollo integral humano, y la vinculación con la comunidad. A partir de Junio del 2004 la ESPE se establecerá como Academia Regional autorizada por Cisco System.

El Programa de Cisco Networking Academy es una alianza entre Cisco Systems e Instituciones de educación, negocios, gobiernos y organizaciones alrededor del mundo.

El Currículo ofrecido por la academia de redes CISCO - ESPE se enfoca en la enseñanza a estudiantes de cómo diseñar, construir y mantener redes de Computadoras. Este programa prepara a estudiantes para el mercado laboral del siglo XXI, mientras sirve además como un valioso modelo de e-learning (Estudios en Línea).

Este avance tecnológico obliga a la Escuela Politécnica del Ejército estar siempre un paso adelante y para cumplir con dicha responsabilidad es necesario implantar un sistema que permita Administrar los recursos académicos y financieros del proyecto.

La concepción de este sistema brindará a la Facultad y al estudiante un nivel óptimo de respuesta a través de la integración de políticas administrativas, bases de datos e Internet, de tal forma que se convierta en un factor diferencial de competitividad ante otras instituciones educativas.

1.2. SITUACIÓN ACTUAL

La Escuela Politécnica del Ejército ostenta un claro liderazgo en la automatización de sistemas de administración, asentados sobre la calidad y confiabilidad de sus distintas variantes de hardware, facilidad de uso y potencia de su software y, muy especialmente, experiencia en aplicaciones orientadas al WEB.

Actualmente no se cuenta con un sistema que permita administrar los recursos académicos y financieros del proyecto antes mencionado, debido a que el mismo empezará en Junio del 2004, y es necesario desarrollar el sistema que permita a la ESPE estar preparada para satisfacer los servicios relacionados con el programa.

1.3. JUSTIFICACIÓN

La automatización de sistemas mediante la capacidad del procesamiento por computadoras, apunta a automatizar determinadas funciones y procesos manuales, ya sean entrantes o salientes; o sea transformar en acciones automáticas las realizadas por personas. Lo que hace eficiente, rápido y seguro el trabajo de operadores informáticos para reducir costos o aumentar el rendimiento diario y mejorar la calidad de atención; además, sustituye el trabajo de personas en la ejecución de ciertas tareas que una máquina puede resolver en forma efectiva y económica. Se emplean para ello soluciones que combinan transmisión de datos (como cifras, nombres, fechas, etc.

El objetivo de la automatización de sistemas es justamente evitar que las personas se conviertan en robots a causa de las exigencias de rendimiento y de la repetitividad de sus tareas.

Por otra parte, las bases de datos están creando una nueva generación avanzada de software que combina la informática y las comunicaciones de una

forma revolucionaria, ofreciendo herramientas para aspectos asociados a la experiencia informática.

Esta iniciativa, va a hacer posible que instituciones educativas se desarrollen con tecnología de punta permitiendo la creación de servicios Web verdaderamente distribuidos, operados por sectores centralizados, que se integran entre sí y colaboran, con carácter complementario, para que los clientes disfruten de innovadores servicios unificados y coherentes.

1.4. ALCANCE

Con el advenimiento masivo de las comunicaciones y el amplio desarrollo de las instituciones educativas, se hizo necesaria la creación de sistemas de administración Académica y Financiera que permitan acceder a la información en forma ágil, segura y especialmente con muy alto rendimiento; es por este motivo, que se propone la creación e implantación del sistema de administración del proyecto CISCO – ESPE, orientándolo al Web con la ayuda de herramientas tecnológicas como son Oracle, Sybase – Sql server y Power Builder.

Se creará e implantará un sistema automatizado y centralizado, el cual permitirá la administración de procesos Académicos y Financieros del Proyecto la ESPE como Academia Regional de CISCO. El sistema estará en capacidad de procesar la información a través de sus diversos módulos de ingreso como el Académico, Financiero y de Seguridades.

El sistema estará en la capacidad de crear diferentes tipos de usuarios como pueden ser:

Administrador.- El Administrador es la persona que tendrá el control absoluto del Sistema.

Supervisor.- El supervisor es la persona encargada de controlar que los procesos del sistema funcionen correctamente.

Usuario.- El Usuario del Sistema es la persona encargada del ingreso de la información ha ser procesada por el sistema.

Este proyecto se desarrollará en diferentes fases que pretenden alcanzar un sistema informático con las mejores características de versatilidad, eficiencia y operación, todo esto combinado con un alto grado de precisión, además se deberá formular la documentación del sistema como: manuales técnicos y del usuario, de tal manera que se visualice el correcto funcionamiento del tema propuesto.

La fase de análisis abarca la recopilación de toda la información necesaria para el modelamiento de datos mediante la fase de diseño, en la fase de desarrollo se incluye toda la programación y codificación de los procesos que intervendrán en este sistema.

La fase de pruebas y evaluación del sistema es la más importante, ya que permiten analizar las capacidades y posibilidades del trabajo desde el punto de vista académico y financiero, que es la base central de este tema.

1.5. OBJETIVOS

1.5.1. Objetivo General.

Crear y poner en funcionamiento el sistema de administración Académica y Financiera del Proyecto “La ESPE como Academia Regional de CISCO”, utilizando herramientas de desarrollo como Oracle, Sybase – Sql server, Power Builder, e Internet, para optimizar la gestión y operación del programa.

1.5.2. Objetivos Específicos.

- ✓ Establecer fundamentos científicos
- ✓ Analizar el entorno que rodea al sistema
- ✓ Establecer las bases y especificaciones técnicas del sistema
- ✓ Establecer el inventario de recursos para el funcionamiento del sistema
- ✓ Determinar el diseño del sistema
- ✓ Construir, realizar pruebas y poner en funcionamiento el sistema
- ✓ Elaborar la documentación de respaldo (Manuales técnico y de usuario)

1.6. FACTIBILIDAD

1.6.1. Factibilidad Técnica

Para el desarrollo del sistema se cuenta con el equipamiento personal del desarrollador, además se cuenta con el Auspicio y apoyo de la Facultad de Ingeniería en Sistemas y el departamento CTT encargado del proyecto la ESPE como Academia Regional CISCO.

La Escuela Politécnica del Ejército a través de sus diferentes facultades ha logrado convenios con la empresa privada y obtenido licenciamiento Educativo Gratuito por lo que permite la viabilidad del sistema a muy bajos costos.

Las características del equipamiento del desarrollador cumplen con los requerimientos básicos para la instalación del software para el desarrollo del sistema, en tal virtud se posee un computador portátil HP Intel Centrino de 1.8, 40 GB en Disco Duro y 512 Mb en RAM, un computador portátil Toshiba Intel Celaron de 1.2, 40 GB en Dico Duro y 128 Mb en RAM.

Por último el desarrollador durante el periodo de estudios de la carrera de Ingeniería en Sistemas e Informática adquirió los conocimientos suficientes de las herramientas de desarrollo y junto con los Ingenieros involucrados permitirá la construcción del sistema cumpliendo con los objetivos específicos del proyecto.

1.6.2. Factibilidad Operativa

Con el advenimiento de nuevos retos en la Educación Universitaria la ESPE tomó la sabia decisión de impartir los conocimientos de las Academias CISCO en el Ecuador, por tal motivo nace la necesidad de desarrollar un sistema que permita la administración Académica y Financiera del proyecto CISCO – ESPE, por tal motivo se tiene una buena aceptación de la solución por parte de los usuarios.

Al finalizar el proyecto estará en la capacidad de facilitar el tratamiento, manejo y seguimiento de la información.

1.6.3. Factibilidad Económica

CATEGORÍA	DESCRIPCIÓN	COSTO
HARDWARE	Servidor para la instalación de la base de datos, posee el departamento del CTT. Máquina que actuara como Cliente computador portátil o de escritorio posee el desarrollador	\$ 0.00
SOFTWARE	Para el desarrollo del sistema se escogió Power Builder 9.0 i, Oracle 9.0 i, Power	\$ 0.00

	Designer 9.0 i, posee la Facultad de Sistemas	
DESARROLLADOR	Un desarrollador a un costo de \$350 dólares por doce meses	\$ 4.200
DOCUMENTACIÓN	Archivos Consultados en el Internet, Documentación de Biblioteca, impresiones de avance de la tesis y tesis final	\$ 1.000
TOTAL		\$ 5.200

Tabla 1.1. Factibilidad Económica

CAPITULO II

MARCO TEÓRICO

MARCO TEÓRICO

El avance disciplinado que ha tenido la Ingeniería de Software permite utilizar las computadoras hoy en día en innumerables campos de aplicación del conocimiento Humano, por esta razón el software que se crea es tan variado como las ideas que se forman en las mentes de las personas. Aún así, todos los proyectos de creación de software siguen un proceso muy parecido para llegar desde la idea original hasta un producto terminado y funcional. Durante el análisis y diseño de un sistema se presentan una serie de alternativas que de alguna u otra forma permiten cumplir con la finalidad planteada, el objetivo fundamental del presente capítulo es el de aportar con los conceptos necesarios de la Ingeniería de Software para el desarrollo adecuado del sistema que permita administrar los recursos académicos y financieros del proyecto la ESPE como academia regional de CISCO, así como los subsistemas que lo constituyen y las diferentes partes involucradas en su funcionamiento.

El proceso de desarrollo de software requiere de un conjunto de notaciones, técnicas, herramientas, y un lenguaje propio, este proceso es conocido también como el **“Ciclo de Vida del Software”** que es tan solo una idea general o un paradigma tras cualquier proceso particular de desarrollo, existen varios modelos para lograr cumplir con los pasos necesarios de este ciclo, cada uno con un enfoque distinto y una forma particular de aplicación. Actualmente estos modelos apuntan a la metodología de desarrollo de sistemas Orientada a Objetos, ya que primero estuvo la metodología de Desarrollo Convencional, luego la metodología de Desarrollo Estructurado, y actualmente la metodología de desarrollo Orientado a Objetos, que permite desarrollar aplicaciones en tres capas, o lo que es lo mismo aplicaciones orientadas al WEB. Todos estos modelos ayudan a realizar la función para la que fueron diseñados: El desarrollo de Software mediante Técnicas de Ingeniería Orientado a Objetos.

2.1. Paradigmas de la Ingeniería de Software

Los paradigmas también conocidos como modelos representan una filosofía particular de la realidad para la construcción del software, cada uno tiene sus ventajas y desventajas, y es importante saber identificarlos correctamente para escoger el más apropiado a la hora del desarrollo, entre los más importantes se encuentran:

- ✓ Modelo en Cascada;
- ✓ Modelo de Construcción de Prototipos;
- ✓ Modelos Evolutivos;
 - Modelo Incremental;
 - Modelo de Desarrollo Concurrente, y;
 - Modelo en Espiral.
- ✓ Modelo de Ensamblado de Componentes.

2.1.1. Modelo en Cascada

El modelo en cascada, también conocido como el modelo del ciclo de vida clásico, es una metodología de desarrollo de software con un criterio sistemático y secuencial, que permite tener un buen seguimiento, y crear una documentación que posteriormente facilita llevar a cabo actualizaciones claras y concisas, así como también procesos de mantenimiento que no conlleven más tiempo de lo que en realidad se necesite.

Este modelo consta de las siguientes fases del desarrollo de software:

- ✓ Ingeniería del Sistemas (Definición de Requerimientos);
- ✓ Análisis ;
- ✓ Diseño;
- ✓ Codificación;
- ✓ Pruebas, y;
- ✓ Operación y Mantenimiento.

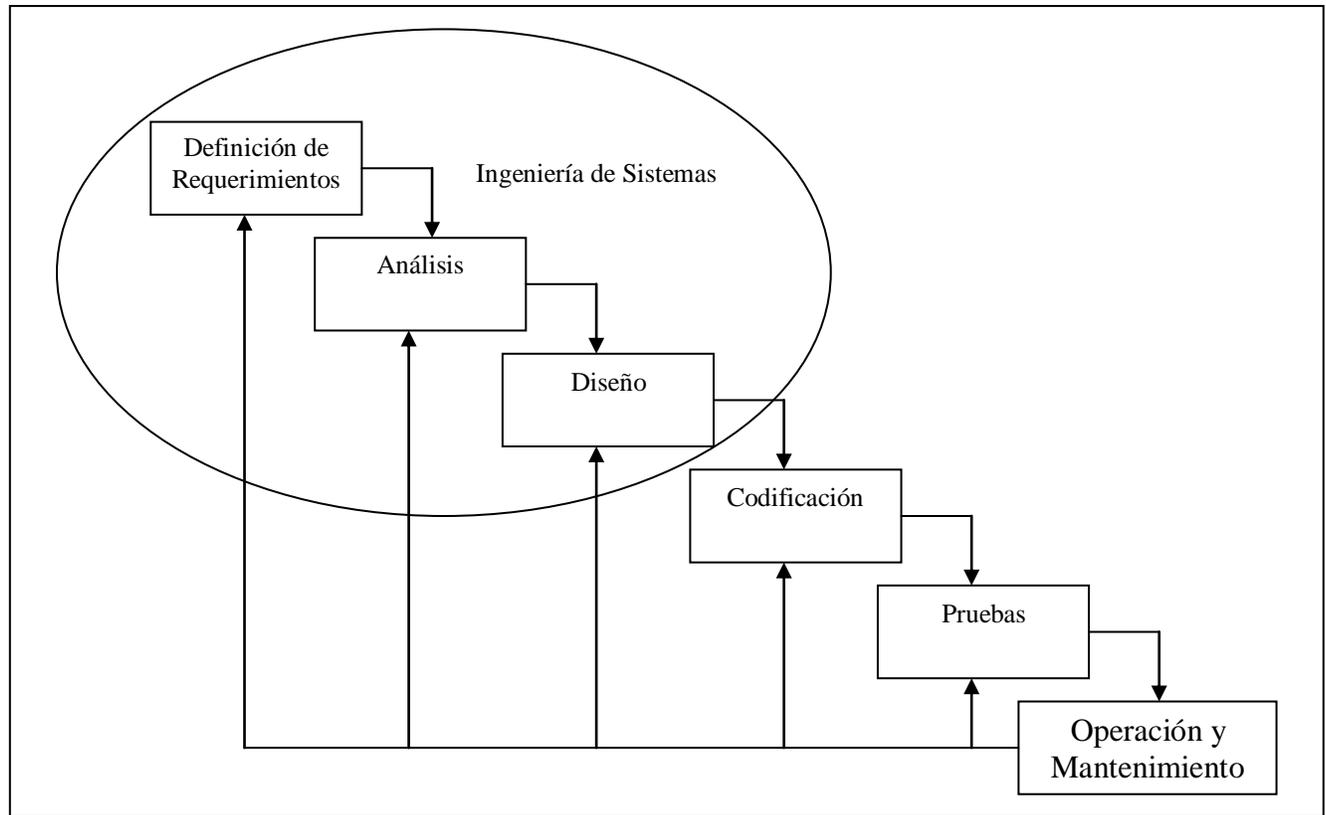


Figura 2.1. Modelo en Cascada

Ingeniería del sistema (Especificación).- Esta es una de las etapas mas importantes, ya que tiene por objetivo lograr un entendimiento claro de las necesidades de la organización y del ambiente en que operará el sistema a implantarse.

Análisis.- La etapa de análisis toma y verifica los descubrimientos de la etapa de especificación de requerimientos y expande estos en suficiente detalle para asegurar la precisión de los modelos del sistema, posibilitando un fundamento sólido para el diseño, dentro del alcance del mismo y tomando en cuenta sistemas existentes; el producto que se genera en esta etapa es el modelado del sistema.

Diseño.- La etapa de diseño toma los requerimientos y el modelado de la etapa de análisis y determina la mejor manera de satisfacerlos, es decir que del

diseño conceptual se pasa al diseño final que será utilizado para la implementación del sistema, e integra tres diseños, el de la base de datos, el de la aplicación y el de la red además se elaboran los planes de prueba y de transición y se realizan los diseños de los sistemas de auditoria, control, de respaldos y recuperación. Los resultados de esta etapa lo constituyen, la arquitectura del sistema, el diseño de la base de datos, la especificación de los programas, la especificación de los manuales de procedimientos.

Codificación.- A partir del diseño final generado en la anterior etapa, en esta se codificarán y probarán los nuevos programas, usando herramientas apropiadas. Esta etapa involucra planeación, diseño de la estructura del sistema, codificación de abajo hacia arriba (prueba de unidades y enlaces), pruebas de arriba a abajo (prueba del sistema) y un enfoque disciplinado en la realización del trabajo y en el control de versiones del sistema y pruebas. Los resultados de esta etapa son los programas probados y la base de datos afinada.

Uno de los productos fundamentales para un uso y un mantenimiento efectivo y eficiente de los sistemas programados son los manuales. Dentro de la codificación se incluye una etapa dedicada a esta actividad tan importante y hace hincapié para que en su elaboración se consideren el estilo de trabajo y las necesidades propias de los usuarios que utilizarán y mantendrán el sistema.

Pruebas.- Una vez que se tiene el código de la aplicación, esta debe ser probada con datos reales, analizando el funcionamiento y lógica interna del programa, verificando que a cada entrada que se ha dado, esta genere los datos esperados.

Mantenimiento.- Una vez que un sistema es terminado, si se ha diseñado e implementado de manera adecuada, funcionará y permitirá que sea aceptable por un buen tiempo, pero a lo largo de este y con el crecimiento de las exigencias del mundo real, se hacen necesarias modificaciones y en muchos de los casos se

requieren modificaciones por errores encontrados y con esto se cae nuevamente a la aplicación del paradigma.

En si esta es la etapa mas larga del modelo del Ciclo de Vida Clásico ya que muchas de las ocasiones se aplica de por vida al sistema generado y conlleva a iteraciones dentro del modelo.

Ventajas

- ✓ Planea un proyecto antes de embarcarse en él;
- ✓ Define el comportamiento externo deseado del sistema antes de diseñar su arquitectura interna;
- ✓ Documenta los resultados de cada actividad;
- ✓ Diseña un sistema antes de codificarlo;
- ✓ Testea un sistema después de construirlo;
- ✓ Flujo secuencial entre las fases;
- ✓ Cada etapa tiene una entrada y una salida;
- ✓ El proceso de desarrollo es confiable, y seguro;
- ✓ Es un método muy estructurado que permite su entendimiento, y;
- ✓ Provee estabilidad en los requerimientos

Desventajas

- ✓ A menudo es difícil que el cliente exponga explícitamente todos los requisitos. El modelo lo requiere y tiene dificultades a la hora de acomodar la incertidumbre natural al comienzo de muchos proyectos.
- ✓ El cliente debe tener paciencia. Una versión de trabajo de los programas no estará disponible hasta que el proyecto esté muy avanzado. Un error grave puede ser desastroso si no se detecta hasta la etapa de pruebas.

- ✓ Los responsables del desarrollo del software siempre se retrasan innecesariamente. La naturaleza lineal lleva a estados de bloqueo, en el que algunos miembros del equipo deben esperar a otros miembros para completar tareas pendientes.

A pesar de sus desventajas, sigue siendo el modelo más extensamente utilizado en la Ingeniería de Software, existen algunas variantes en que se ha tratado de combinar sus ventajas con otros esquemas que reduzcan sus limitaciones, para que de esta manera se puedan construir modelos híbridos, combinando varios modelos con un solo objetivo, el desarrollo adecuado del software.

2.1.2. Modelo de Construcción de Prototipos

El desarrollo de prototipos comienza con la definición de requerimientos y objetivos globales por el desarrollador y el usuario. Se utiliza un diseño rápido, que se centra en una representación de esos aspectos del software que serán visibles para el usuario. El diseño rápido lleva a la construcción de un prototipo que lo evalúa el usuario y utiliza para refinar los requisitos de software a desarrollar, la interacción ocurre cuando el prototipo satisface las necesidades del usuario, a la vez que permite que el desarrollador comprenda mejor lo que se necesita hacer.

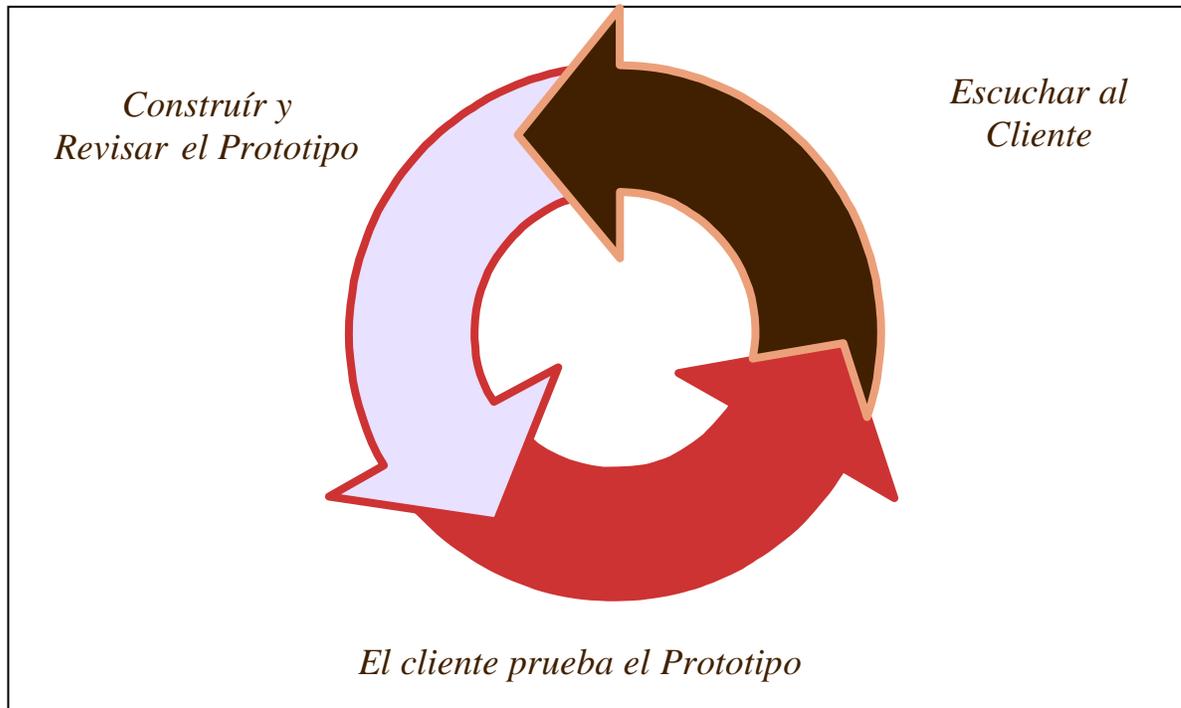


Figura 2.2. Modelo de Construcción de Prototipos

También es válida la construcción de prototipos para probar la eficiencia de algoritmos a implementar o para comprobar el rendimiento de un determinado componente del sistema, tal como una base de datos o el soporte de hardware.

La construcción de prototipos se caracteriza por:

- ✓ Un alto grado de interacción;
- ✓ Un alto grado de participación del usuario

Ventajas

- ✓ Útiles cuando los requerimientos son cambiantes.
- ✓ Cuando no se conoce bien la aplicación.
- ✓ Cuando el usuario no se quiere comprometer con los requerimientos.
- ✓ Cuando se quiere probar una arquitectura o tecnología.
- ✓ Cuando se requiere rapidez en el desarrollo.

Desventajas

- ✓ No se conoce cuando se tendrá un producto aceptable.
- ✓ No se sabe cuantas interacciones serán necesarias.
- ✓ Da una falsa ilusión al usuario sobre la velocidad del desarrollo.

A pesar de los problemas que pueden producirse, la realización de prototipos es aplicada en varios modelos como parte del conocimiento de las necesidades de los usuarios, la clave es definir las reglas al inicio: el usuario y el equipo de desarrollo deben acordar su realización y su uso como mecanismo de definición de requisitos.

2.1.3. Modelo en Espiral

El modelo en espiral para la ingeniería del software es actualmente el paradigma más realista para el desarrollo de sistemas a gran escala. Utiliza un enfoque “evolutivo” para la ingeniería del software, permitiendo al desarrollador y al usuario entender y reaccionar a los riesgos en reducción del mismo, pero, lo que es más importante, permite a quien lo desarrolla aplicar el enfoque de creación de prototipos en cualquier etapa de la evolución del producto. Mantiene el enfoque sistemático correspondiente a los pasos sugeridos por el ciclo de vida clásico, pero incorporándola dentro de un marco de trabajo interactivo que refleja de forma más realista el mundo real. El modelo en espiral demanda una consideración directa de riesgos técnicos, y si se aplica adecuadamente, debe reducirlos en todas las etapas del proyecto antes de que se conviertan en problemáticos.

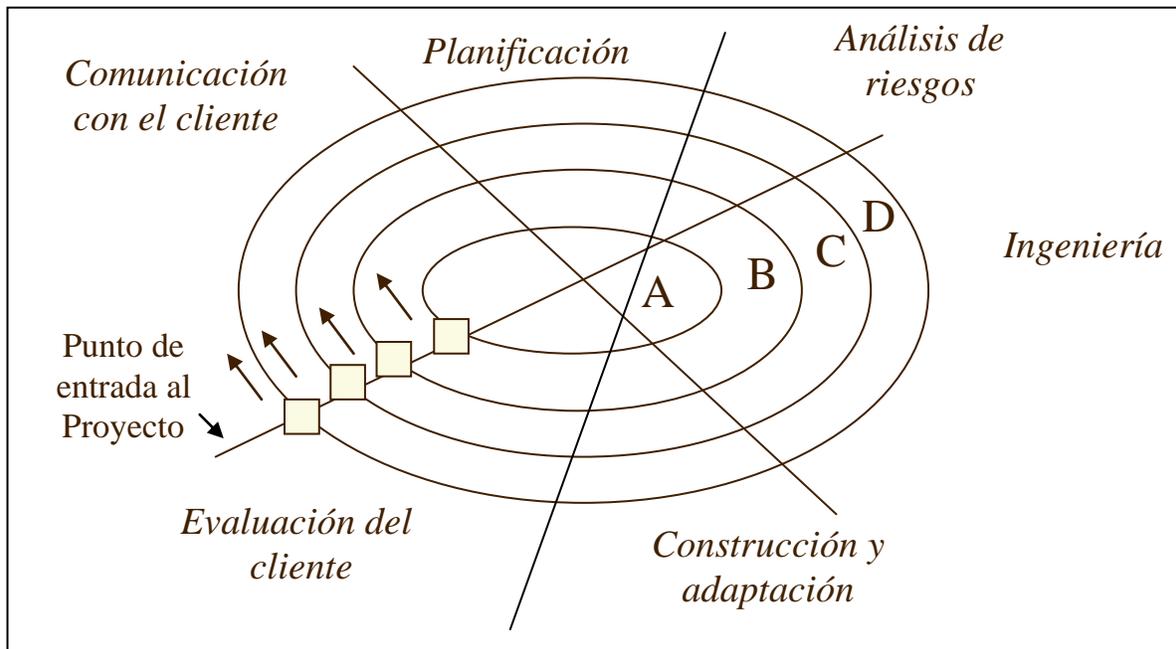


Figura 2.3. Modelo en Espiral

- | | |
|----------------------------|------------------------------|
| A: Desarrollo de Conceptos | B: Desarrollo del Sistema |
| C: Mejora del Sistema | D: Mantenimiento del Sistema |

El software se desarrolla en una serie de versiones incrementales. Cada giro de la espiral representa una fase del proceso de software, tal como fue definido por la dirección del proyecto. El centro del modelo concierne a la factibilidad del sistema. El siguiente a la definición de los Requerimientos, el siguiente al Diseño del Sistema y así sucesivamente.

Ventajas

- ✓ El modelo en espiral utiliza la construcción de prototipos como un mecanismo de reducción de riesgos pero, lo que es más importante, permite a quien lo desarrolla aplicar el enfoque de construcción de prototipos en cualquier etapa de evolución del producto.
- ✓ Mantiene el enfoque sistemático de los pasos sugeridos por el ciclo de vida clásico, pero lo incorpora a un marco de trabajo interactivo que refleja en forma más realista el mundo real.

- ✓ Si se aplica adecuadamente debe reducir los riesgos antes de que se conviertan en problemáticos.

Desventajas

- ✓ Resulta difícil convencer a grandes clientes de que el enfoque evolutivo es controlable.
- ✓ Requiere una considerable habilidad para la evaluación del riesgo y cuenta con esa habilidad para su éxito. Si un riesgo no es descubierto y gestionado surgirán problemas.
- ✓ Es un esquema relativamente nuevo en relación con esquemas lineales o de construcción de prototipos.

2.2. Metodología Orientado a Objetos

La Programación Orientada a Objetos (OOP por sus siglas en inglés de Object Oriented Programming) como paradigma, "es una forma de pensar, una filosofía, de la cual surge una cultura nueva que incorpora técnicas y metodologías diferentes. La OOP es una postura ontológica: el universo computacional está poblado por objetos, cada uno responsabilizándose por sí mismo, y comunicándose con los demás por medio de mensajes" [Greiff 1994].

Se debe distinguir que la OOP como paradigma (es un enfoque o manera de visualizar la realidad) y como metodología (es una colección de características para la ingeniería de software) no es lo mismo. Sin embargo, la publicidad confunde asociando la OOP más a una metodología, que al paradigma. De aquí que "el interés en la OOP radica más en los mecanismos que aporta para la construcción de programas que en aprovechar un esquema alterno para el modelado de procesos computacionales" [Greiff 1994]. La Programación Orientada a Objetos desde el punto de vista computacional "es un método de implementación en el cuál los programas son organizados como grupos cooperativos de objetos, cada uno de los cuales representa una instancia de

alguna clase, y estas clases, son miembros de una jerarquía de clases unidas vía relaciones de herencia" [Greiff 1994]. El paradigma OO se basa en el concepto de objeto. Un objeto es aquello que tiene estado (propiedades más valores), comportamiento (acciones y reacciones a mensajes) e identidad (propiedad que lo distingue de los demás objetos). La estructura y comportamiento de objetos similares están definidos en su clase común; los términos instancia y objeto son intercambiables. Una clase es un conjunto de objetos que comparten una estructura y comportamiento común. La diferencia entre un objeto y una clase es que un objeto es una entidad concreta que existe en tiempo y espacio, mientras que una clase representa una abstracción, la "esencia" de un objeto, tal como son.

De aquí que un objeto no es una clase, sin embargo, una clase puede ser un objeto. En el enfoque OO las propiedades del objeto son claves. Los principios del modelo OO son: abstracción, encapsulación, modularidad y jerarquía, fundamentalmente, y en menor grado tipificación (typing), concurrencia, persistencia. [Booch 1986] dice que si un modelo que se dice OO no contiene alguno de los primeros cuatro elementos, entonces no es OO.

- ✓ **Abstracción.** Es una descripción simplificada o especificación de un sistema que enfatiza algunos de los detalles o propiedades del sistema, mientras suprime otros.
- ✓ **Encapsulación.** En el proceso de ocultar todos los detalles de un objeto que no contribuyen a sus características esenciales.
- ✓ **Modularidad.** Es la propiedad de un sistema que ha sido descompuesto en un conjunto de módulos coherentes e independientes.
- ✓ **Jerarquía o herencia.** Es el orden de las abstracciones organizado por niveles.
- ✓ **Tipificación.** Es la definición precisa de un objeto de tal forma que objetos de diferentes tipos no puedan ser intercambiados o, cuando mucho, puedan intercambiarse de manera muy restringida.
- ✓ **Concurrencia.** Es la propiedad que distingue un objeto que está activo de uno que no lo está.

- ✓ **Persistencia.** Es la propiedad de un objeto a través de la cual su existencia trasciende el tiempo (es decir, el objeto continua existiendo después de que su creador ha dejado de existir) y/o el espacio (es decir, la localización del objeto se mueve del espacio de dirección en que fue creado).

Los beneficios del enfoque OO son: [Booch 1986]

- ✓ El uso del modelo OO nos ayuda a explotar el poder expresivo de todos los lenguajes de programación basados en objetos y los orientados a objetos, como Smalltalk, Object Pascal, C++, CLOS, Ada, [y recientemente Java].
- ✓ El uso del modelo OO alienta el reuso no solo del software, sino de diseños completos.
- ✓ Produce sistemas que están contruidos en formas intermedias estables y por ello son más resistentes al cambio en especificaciones y tecnología.

El mismo autor considera que el principal beneficio del OOD es que da un mecanismo para formalizar el modelo de la realidad.

[Greiff 1994] comenta que el Análisis Orientado a Objetos (OOA por sus siglas en inglés de Object Oriented Analysis) "es un método de análisis que examina los requerimientos desde la perspectiva de las clases y objetos encontrados en el vocabulario del dominio del problema". Según [Booch 1986], el Diseño Orientado a Objetos "es un método de diseño abarcando el proceso de descomposición orientado a objetos y una notación para representar ambos modelos lógico y físico tal como los modelos estáticos y dinámicos del sistema bajo diseño".

En cuanto a las metodologías OO, hay un gran número de métodos orientado a objetos actualmente. Muchos de los métodos pueden ser clasificados como orientados a objetos porque soportan de manera central los conceptos de la orientación a objetos. Algunas de las metodologías más conocidas y estudiadas hasta antes del UML según [Jacobson 1992] son:

- ✓ Object-Oriented Design (OOD), Booch.
- ✓ Object Modeling Technique (OMT), Rumbaugh.
- ✓ Object Oriented Analysis (OOA), Coad/Yourdon.
- ✓ Hierarchical Object Oriented Design (HOOD), ESA.
- ✓ Object Oriented Structured Design (OOSD), Wasserman.
- ✓ Object Oriented Systems Analysis (OOSA), Shaler y Mellor.
- ✓ Responsibility Driven Design (RDD), Wirfs-Brock, entre otros.

Actualmente las metodologías más importantes de análisis y diseño de sistemas han confluído en lo que es el UML, bajo el respaldo del Object Management Group.

2.2.1. Metodología de Desarrollo OMT (Técnica de Modelamiento de Objetos).

Existen muchas aproximaciones de desarrollo de software que utilizan modelos orientado a objetos, pero que no tienen todos los soportes para desarrollo de aplicaciones de base de datos. Algunas aproximaciones carecen de suficientes abstracciones y tienen un bajo relacionamiento para detalles de implementación.

Otros métodos de programación orientados ponen un escaso énfasis en la estructura de datos y constantes, que son muy importantes para aplicaciones de base de datos.

OMT pone énfasis en la importancia del modelo y uso para lograr una abstracción, en el cual el análisis está enfocado en el mundo real para un nivel de diseño, también pone detalles particulares para modelado de recursos de la computadora. Esta Tecnología puede ser aplicada en varios aspectos de implementación incluyendo archivos, base de datos relacionales, base de datos orientados a objetos. OMT está construido alrededor de descripciones de estructura de datos, constantes, sistemas para procesos de transacciones.

2.2.1.1. Fases de la Metodología OMT.

Análisis. Su objetivo es desarrollar un modelo de lo que va a hacer el sistema. El modelo se expresa en términos de objetos y de relaciones entre ellos, flujo dinámico de control y las transformaciones funcionales.

Diseño del sistema. Se define la arquitectura del sistema y se toman las decisiones estratégicas.

Diseño de objetos. Su objetivo es refinar el modelo del análisis y proporcionar una base detallada para la implementación tomando en cuenta el ambiente en que se implementará.

Mantenimiento. La documentación del desarrollo y seguimiento de los modelos a través del código facilita el posterior mantenimiento.

La metodología OMT soporta múltiples estilos de desarrollo. Se puede usar OMT para conseguir un alto performance en la fase de análisis y diseño e implementación con una estricta secuencia de pasos, también adopta una estrategia de desarrollo iterativa.

2.3. Técnica de Análisis y Diseño

2.3.1. Leguaje Unificado de Modelado UML (Unmified Modeling Language)

UML (Unified Modeling Language), que en español significa Lenguaje Unificado de Modelado, es un lenguaje que permite a los desarrolladores especificar, modelar, construir y documentar los elementos que un sistema de software establece para su composición; es importante mencionar que es un modelado que permite trabajar con sistemas que están orientados a objetos, estableciendo una notación y semántica común; por tal motivo se ha convertido en un estándar que establece una notación y semántica común.

UML es el sucesor de la oleada de métodos de análisis y diseño orientado a objetos (OOA&D) que surgió a finales de la década de 1980 y principios de la siguiente. El UML unifica, sobre todo, los métodos de Booch (“Object-Oriented Analysis and Design with Applications” OOD), Rumbaugh (“Object-Oriented Modeling and Design” OMT) y Jacobson (“Object-Oriented Software Engineering”), se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real. UML ofrece nueve diagramas en los cuales modelar sistemas.

- ✓ Diagramas de Casos de Uso para modelar los procesos 'business'.
- ✓ Diagramas de Secuencia para modelar el paso de mensajes entre objetos.
- ✓ Diagramas de Colaboración para modelar interacciones entre objetos.
- ✓ Diagramas de Estado para modelar el comportamiento de los objetos en el sistema.
- ✓ Diagramas de Actividad para modelar el comportamiento de los Casos de Uso, objetos u operaciones.
- ✓ Diagramas de Clases para modelar la estructura estática de las clases en el sistema.
- ✓ Diagramas de Objetos para modelar la estructura estática de los objetos en el sistema.
- ✓ Diagramas de Componentes para modelar componentes.
- ✓ Diagramas de Implementación para modelar la distribución del sistema.

Esta notación ha sido ampliamente aceptada debido al prestigio de sus creadores y a que incorpora las principales ventajas de cada uno de los métodos particulares en los que se basa: Booch, OMT y OOSE. UML ha puesto fin a las llamadas “guerras de métodos” que se han mantenido a lo largo de los 90, en las que los principales métodos sacaban nuevas versiones que incorporaban las técnicas de los demás. Con UML se fusiona la notación de estas técnicas para formar una herramienta compartida entre todos los ingenieros software que trabajan en el desarrollo orientado a objetos.

El objetivo principal cuando se empezó a gestar UML era posibilitar el intercambio de modelos entre las distintas herramientas CASE orientadas a objetos del mercado. Para ello era necesario definir una notación y semántica común. Hay que tener en cuenta que el estándar UML no define un proceso de desarrollo específico, tan solo se trata de una notación.

2.3.1.1. Tipos de Diagramas UML

Dentro de los diagramas UML existen dos grupos mayoritarios de diagramas UML:

- ✓ Diagramas Estructurales, los cuales muestran una vista estática del modelo, y;
- ✓ Diagramas de Comportamiento, que muestran una vista dinámica del modelo.

Diagramas Estructurales

Los diagramas estructurales representan elementos componiendo un sistema o una función. Estos diagramas pueden reflejar las relaciones estáticas de una estructura, como lo hacen los diagramas de clases, de paquetes, o arquitecturas en tiempo de ejecución.

Los diagramas estructurales incluyen:

- ✓ Diagramas de Clases;
- ✓ Diagramas de Estructuras Compuestas;
- ✓ Diagramas de Componentes;
- ✓ Diagramas de Despliegue
- ✓ Diagramas de Objetos, y;
- ✓ Diagramas de Paquetes.

En la siguiente figura podemos encontrar un ejemplo de cada uno de los diagramas mencionados anteriormente.

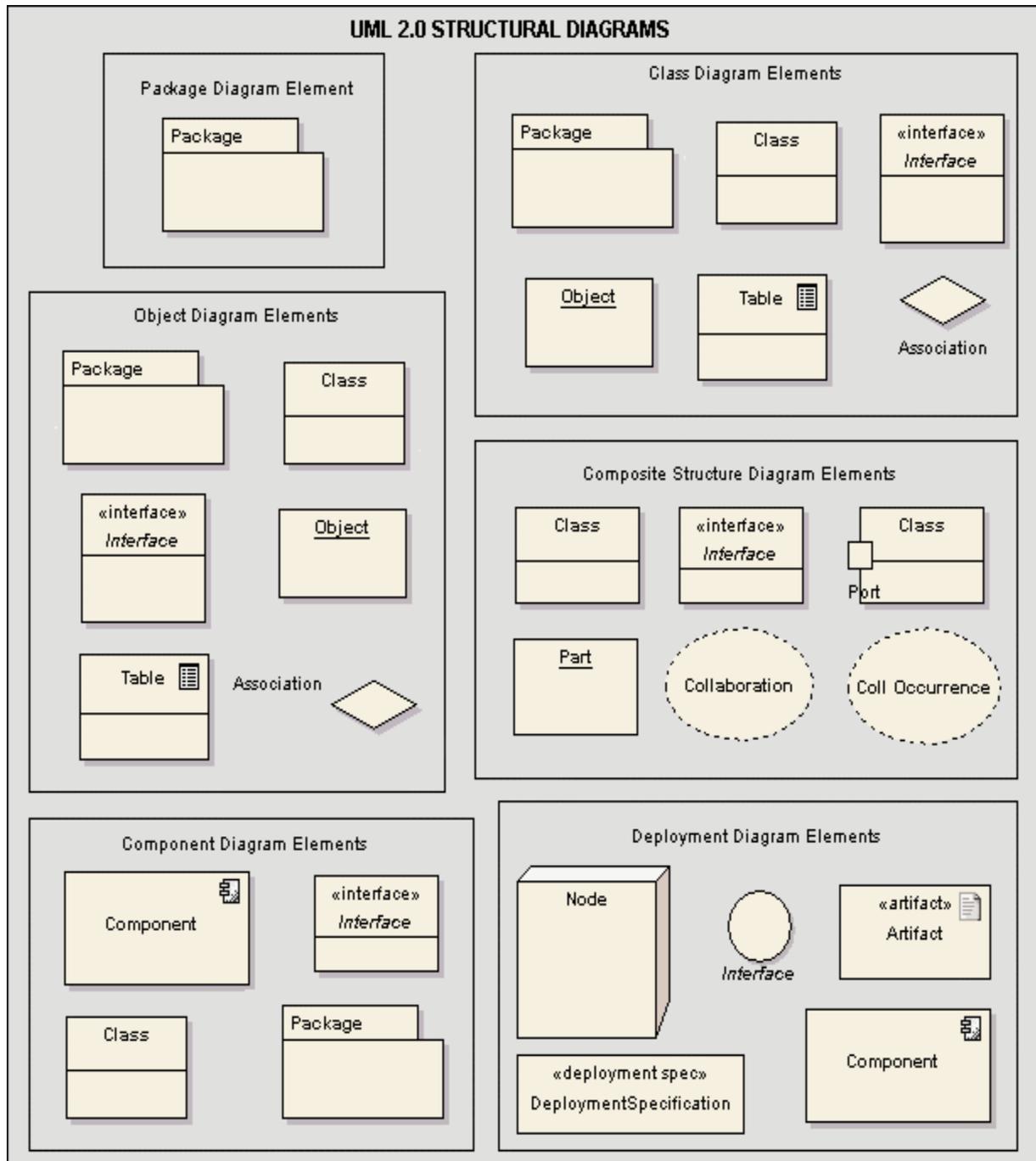
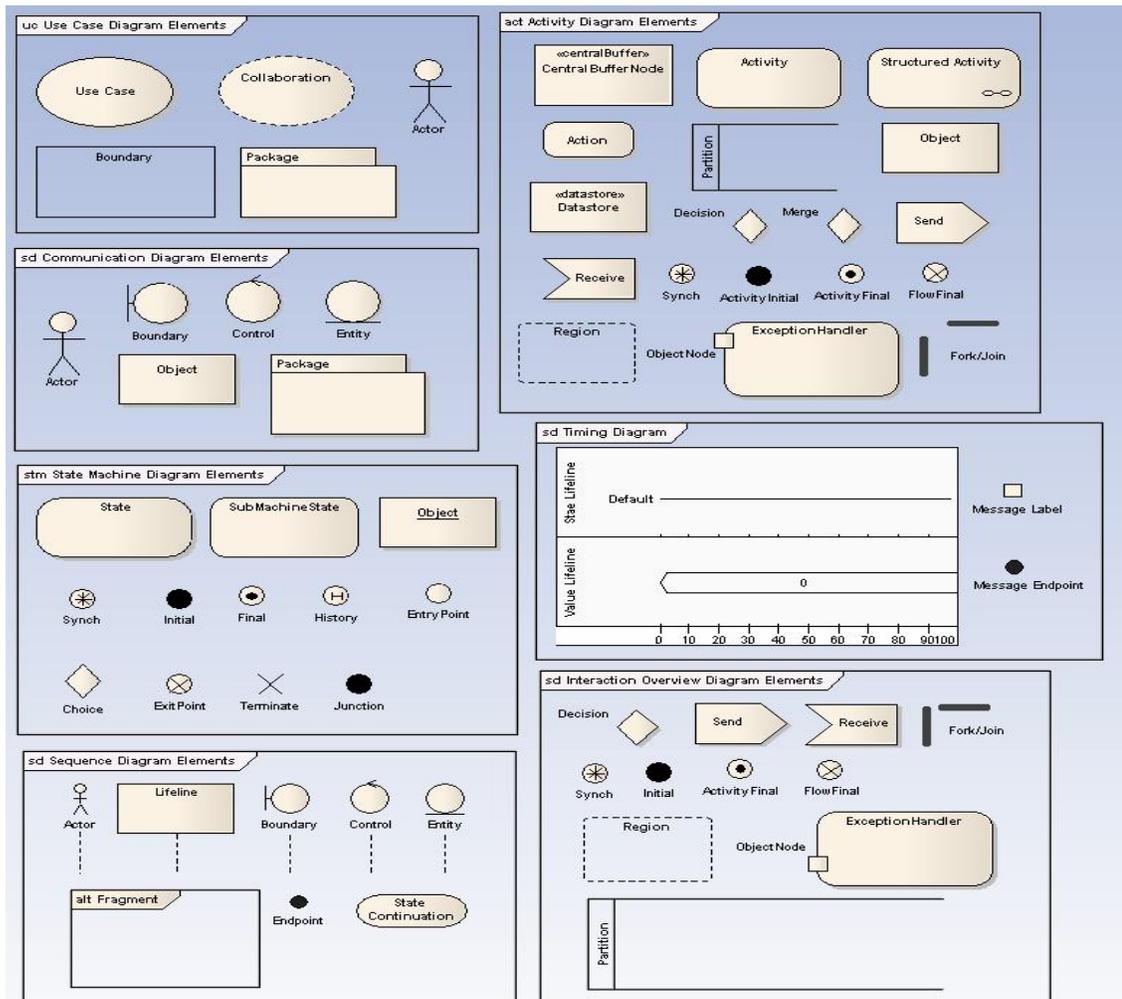


Figura 2.4. Tipos de Diagramas¹

¹ <http://www.sparxsystem.com.ar/download/ayuda/index.html?umlelements.htm>

Los diagramas de Comportamiento como su nombre lo indica, describe el comportamiento de un sistema o procesos de negocios. Los diagramas de comportamiento incluyen:

- ✓ Diagramas de Actividades;
- ✓ Diagramas de Máquina de estados;
- ✓ Diagramas de Comunicaciones;
- ✓ Diagramas de la Descripción de la Interconexión;
- ✓ Diagramas de Secuencias;
- ✓ Diagramas Tiempos, y;
- ✓ Diagramas de Casos de Uso.



2

Figura 2.5. Diagrama de Comportamiento²

² <http://www.sparxsystem.com.ar/download/ayuda/index.html?umlelements.htm>

Los modelos se constituyen en UML a partir de elementos. Los elementos de modelado se usan en distintas etapas del proceso de diseño para diferentes propósitos, los elementos incluyen: Clases, Objetos, Interfaces, Casos de Uso, Componentes, Nodos.

Clases.- Una Clase es una representación de uno o más objetos, que refleja su estructura y comportamiento en el sistema. Es una plantilla desde la cual se crean las instancias actualmente en ejecución, una clase puede tener atributos (Datos), y métodos (operaciones o comportamiento), una clase se representa de la siguiente manera:



Figura 2.6. Representación de una Clase.

Objetos.- Un objeto es un instancia de una clase en tiempo de ejecución, en las etapas tempranas del análisis, los objetos se pueden utilizar para capturar rápidamente todas las cosas que son de relevancia en el dominio del sistema, un objeto se representa de la siguiente manera:

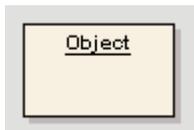


Figura 2.7. Representación de un Objeto.

Interfaces.- Una interfaz es una especificación de comportamiento que los implementadores acordaron (Contrato), una interfaz se representa de la siguiente manera:



Figura 2.8. Representación de una Interfaz.

) de modelado

UML que describe como un usuario del sistema propuesto interactuará con el sistema para llevar a cavo una unidad discreta de trabajo, y se lo representa de la siguiente manera:

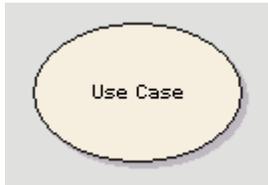


Figura 2.9. Representación de un Caso de Uso.

Componentes.- Un componente es una parte modular del sistema, cuyo comportamiento es definido por sus interfaces proporcionadas y requeridas. Un componente se puede componer de cases múltiples, o de componentes ensamblados, un componente se representa de la siguiente manera:

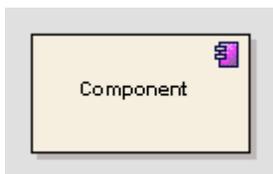


Figura 2.10. Representación de un Componente.

Nodos.- Un nodo es una pieza física de equipamiento sobre el que se desplegará el sistema, un nodo usualmente aloja componentes y otras piezas ejecutables de código, los cuales a su vez pueden ser vinculados a procesos particulares o a espacios de ejecución, se representa de la siguiente manera:

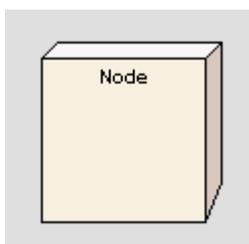


Figura 2.11. Representación de un Nodo.

2.3.1.3. Conectores UML

Un conector es una relación lógica o funcional entre elementos del modelo. Hay varios tipos diferentes de conexiones, cada una con propósitos y sintaxis particulares, los conectores juntos con los elementos UML forman la base de los modelos UML, así tenemos:

Conectores de los diagramas de Comportamiento	Conectores de los diagramas Estructurales	Conectores incorporados y extendidos
Diagrama de actividades Flujo de control Flujo de objeto Flujo de interrupción	Diagrama de estructura compuesta Conector Ensamble Delegar Enlace de Roles Representa Ocurrencia	Diagrama de análisis Flujo de información Flujo de objeto Asociar Realización Representation
Diagrama de Casos de uso Uso Asociar Generalización Inclusión Extensión Realización Invoca Precede	Diagramas de paquetes, clases y objetos Asociar Generalización Componer Agregar Clase Asociación Ensamble	Conectores comunes Dependencia Realización Traza Flujo de información Vínculo de nota
Diagrama de estados Transición Flujo de objeto	Dependencia Realización Flujo de información Anidamiento Combinación del Paquete Importación de Paquetes	Perfil Extension Generalización Aplicación Valor etiquetado Redefinición
Diagrama de tiempos Mensaje	Diagrama de componentes Ensamble Delegar Asociar Realización Generalización	Meta modelo Generalización Asociar Componer Agregar
Diagrama de secuencia Mensaje Auto mensaje Recursión Llamada		Personalizado Asociar
Diagrama de comunicación		



Figura 2.12. Conectores de Diagramas de Comportamiento³

2.4. Herramientas CASE

Para el desarrollar software es necesario contar con otros tipos de software, es decir, se necesita de herramientas de desarrollo adecuadas que faciliten la construcción de verdaderos sistemas de cómputo. Los medios sistematizados que se utilizaron por mucho tiempo estaban limitados a los tradicionales editores de texto para la codificación, y los compiladores del lenguaje respectivo. Fuera de éstos era poco el soporte que un programador o desarrollador de sistemas obtenía por parte de su ambiente de trabajo. Debido a esta escasez de herramientas adecuadas para el desarrollo de sistemas surgió la lógica necesidad de crear sistemas que se pudieran utilizar verdaderamente como herramientas de soporte en la construcción de software. De ahí surge la Ingeniería de Software Asistida por Computadora, o en inglés, Computer-Aided Software Engineering (CASE). Así, una herramienta CASE es un producto computacional enfocado a apoyar una o mas técnicas dentro de un método de desarrollo de software [Jarzabek, 98].

³ <http://www.sparxsystem.com.ar/download/ayuda/index.html?umlelements.htm>

A pesar de que las herramientas CASE no tienen una historia extremadamente larga, pues empiezan a surgir a partir de principios de la década de los ochenta, ya se han extendido a la mayor parte de las fases y actividades involucradas en el desarrollo de software. Existen diversas taxonomías de las herramientas CASE, que utilizan varios criterios para su clasificación. Una clasificación por función se divide en dos grandes áreas: CASE superiores (U-CASE) y CASE inferiores (L-CASE). Los U-CASE abarcan las etapas de planeación, análisis y diseño, mientras que los L-CASE comprenden las de codificación, pruebas y mantenimiento. De esta manera se cubren las grandes áreas del desarrollo de software.

Las herramientas CASE individuales pueden estar enfocadas a un área de ingeniería de software más específica, como lo puede ser la ingeniería de información, el modelado de procesos, planificación y administración de proyectos, análisis de riesgos, seguimiento de requisitos, métricas, documentación, control de calidad, gestión de bases de datos, de desarrollo de interfaz o de generación de prototipos entre otros [Pressman, 98], contribuyendo a mejorar la calidad y la productividad en el desarrollo de sistemas de información y se plantean los siguientes objetivos:

- Permitir la aplicación práctica de metodologías estructuradas.
- Facilitar la realización de prototipos y el desarrollo conjunto de aplicaciones.
- Simplificar el mantenimiento de los programas automatizando el chequeo de errores.
- Automatizar, Estandarizar y Mejorar la documentación.
- Automatizar la generación de código y aumentar la portabilidad de las aplicaciones.
- Facilitar la reutilización de componentes software.
- Permitir un desarrollo y un refinamiento visual de las aplicaciones, mediante la utilización de nuevas interfaces de usuarios.

El tipo específico de herramienta que se utilice depende de los requerimientos tanto del sistema a implementar como de los desarrolladores.

2.4.1. Componentes de una Herramienta CASE

Una Herramienta CASE se compone de los siguientes elementos:

- Repositorio (diccionario) donde se almacenan los elementos definidos o creados por la herramienta, y cuya gestión se realiza mediante el apoyo de un Sistema de Gestión de Base de Datos (SGBD) o de un sistema de gestión de ficheros.
- Meta modelo (no siempre visible), que constituye el marco para la definición de las técnicas y metodologías soportadas por la herramienta.
- Carga o descarga de datos, son facilidades que permiten cargar el repertorio de la herramienta CASE con datos provenientes de otros sistemas, o bien generar a partir de la propia herramienta esquemas de base de datos, programas, etc. que pueden, a su vez, alimentar otros sistemas. Este elemento proporciona así un medio de comunicación con otras herramientas.
- Comprobación de errores, facilidades que permiten llevar a cabo un análisis de la exactitud, integridad y consistencia de los esquemas generados por la herramienta.
- Interfaz de usuario, que constará de editores de texto y herramientas de diseño gráfico que permitan, mediante la utilización de un sistema de ventanas, iconos y menús, con la ayuda del ratón, definir los diagramas, matrices, etc. que incluyen las distintas metodologías.

2.4.2. Características de una Herramienta Case

Una Herramienta Case debe ser capaz de:

- **Proporcionar Aplicaciones Portátiles.-** La herramienta debe generar código para Windows, OS/ 2, Macintosh, Unix y todas las plataformas de servidores conocidas.
- **Control de Versión.-** La herramienta debe reconocer las versiones de códigos que se ejecutan en los clientes y servidores, y asegurarse que sean consistentes. También debe ser capaz de controlar un gran número de tipos de objetos incluyendo texto, gráficos, mapas de bits, documentos complejos y objetos únicos, tales como definiciones de pantallas y de informes, archivos de objetos y datos de prueba y resultados. Debe mantener versiones de objetos con niveles arbitrarios de granularidad; por ejemplo, una única definición de datos o una agrupación de módulos.
- **Creación de código compilado en el servidor.-** La herramienta debe ser capaz de compilar automáticamente código 4GL en el servidor para obtener el máximo performance.
- **Trabajar con una variedad de Administradores de Recursos.-** La herramienta debe adaptarse ella misma a los administradores de recurso que existen en varios servidores de la red; su interacción con los administradores de recurso debería ser negociable a tiempo de ejecución.
- **Trabajar con una variedad de software intermedio.-** La herramienta debe adaptar sus comunicaciones cliente / servidor al software intermedio existente. Como mínimo la herramienta debería ajustar los temporizadores basándose en, si el tráfico se está moviendo en una LAN o WAN.

- **Soporte multiusuarios.**-La herramienta debe permitir que varios diseñadores trabajen en una aplicación simultáneamente. Debe gestionarse los accesos concurrentes a la base de datos por diferentes usuarios, mediante el arbitrio y bloqueos de accesos a nivel de archivo o registro.
- **Seguridad.**- La herramienta proporciona mecanismos para controlar el acceso y las modificaciones a los que contiene. La herramienta debe, al menos, mantener contraseñas y permisos de acceso en distintos niveles para cada usuario. También debe facilitar la realización automática de copias de seguridad y recuperaciones de las mismas, así como el almacenamiento de grupos de información determinados.
- **Desarrollo en equipo, repositorio de librerías compartidas.**- Debe permitir que grupos de programadores trabajen en un proyecto común; debe proveer facilidades de check-in/ check-out registrar formas, widgets, controles, campos, objetos de negocio, DLL, etc.; debe proporcionar un mecanismo para compartir las librerías entre distintos realizadores y múltiples herramientas; Gestiona y controla el acceso multiusuario a los datos y bloquea los objetos para evitar que se pierdan modificaciones inadvertidamente cuando se realizan simultáneamente.

2.4.3. Herramientas CASE escogidas para el Desarrollo del Sistema Cisco - Espe

Actualmente existe varias herramientas CASE para la mayor parte de los procesos del ciclo de vida del software, las cuales se presentan con las metodologías de ingeniería de software más conocidas. Las herramientas escogidas para el desarrollo del sistema CISCO – ESPE permitirán aplicar los conceptos y metodologías anteriormente mencionadas, es por esta razón que se

necesita de herramientas para cada fase del ciclo de vida del software, en tal virtud se tiene que para el análisis y diseño de software se escoge Power Designer, herramienta de desarrollo que permite modelar los datos aplicando metodologías Orientadas a Objetos como son OMT y UML, para la etapa de codificación, pruebas y mantenimiento del sistema se divide en dos partes: para el desarrollo de la base de datos o Back End, se escoge la herramienta de desarrollo Oracle 9i, que permite tener una base de datos consistente y adecuada, y para el desarrollo de la interfaz del usuario se escoge Power Builder, herramienta de desarrollo que tiene gran compatibilidad con la base de datos descrita anteriormente. La razón para incluir herramientas que también cuenten con la fase de diseño, orientado a objetos, es debido a que el proyecto se propone como la primera parte de un ambiente integrado.

2.4.3.1. SYBASE

Aporta la tecnología necesaria para que las organizaciones desarrollen soluciones que les permitan distribuir la información ahí donde es necesaria en el momento preciso. SYBASE ha centrado su estrategia en bases de datos y productos de conectividad y herramientas de desarrollo para ofrecer soluciones a sus clientes.

2.4.3.2. POWER DESIGNER

Es la herramienta para el análisis, diseño inteligente y construcción sólida de una base de datos y un desarrollo orientado a modelos de datos a nivel físico y conceptual, que da a los desarrolladores Cliente/Servidor la más firme base para aplicaciones de alto rendimiento.

Ofrece un acercamiento de diseño para optimizar las estructuras de las bases de datos capturando el flujo de datos de su organización, puede crear un modelo conceptual y físico de la base de datos. La técnica de diseño a dos niveles permite separar lo que se desea diseñar de lo que se desea implementar.

2.4.3.3. POWER DESIGNER 6.1

Los equipos de proyectos comparten un diccionario central construido sobre una base de datos SQL. Puede asegurar la consistencia de los datos accediendo al modelado de información con una hojeada al poderoso diccionario browser. Para seguridad e integridad de los datos, puede también administrar los derechos de usuarios y bloquear el acceso.

- **PowerDesigner 6.1** Es una suite de aplicaciones de Powersoft para la construcción, diseño y modelado de datos a través de diversas aplicaciones. Esta suite cuenta con los siguientes productos:
- **PowerDesigner ProcessAnalyst.** Permite analizar el flujo de datos de toda la empresa, a través de los departamentos hasta el usuario final.
- **PowerDesigner DataArchitect.** Provee a los diseñadores de las bases de datos una manera eficiente para la creación inteligente, depuración e ingeniería de reversa del modelado tanto conceptual como físico de los datos.
- **PowerDesigner AppModeler.** Permite el diseño y ajuste de los componentes de objetos y datos en aplicaciones de uso común como PowerBuilder, Power++, Visual Basic y Delphi ajustando el modelo de base de datos. Junto con la aplicación de servidor PowerDynamo (incluido) se pueden publicar las bases de datos en Internet/Intranet directamente del modelo de base de datos. Esta herramienta también puede generar páginas de servidor activas para MS Internet Information Server v.3.0.
- **PowerDesigner WarehouseArchitect.** Provee un poderoso datawarehousing para el diseño e implementación de una base de datos. Cuenta con soporte para bases de datos tradicionales DBMS y bases de datos en plataformas de sistemas analíticos usando modelados dimensionales, esquemas de "estrella" y "nieve", particionamiento y agregación. También cuenta con un alto desempeño en el indexamiento de esquemas.

- **PowerDesigner MetaWorks.** Permite fácilmente ver y compartir la información del modelado de datos con una definición constante de objetos. También puede comparar y mezclar dos modelos de datos paso a paso.
- **PowerDesigner Viewer.** Crea reportes de los modelos físicos, conceptuales y procesos del modelado de la base de datos. También permite generar reportes para Internet en HTML. Este producto cuenta con demos directos de sitio de Sybase en Internet para su evaluación.

2.4.3.4. POWER BUILDER

PowerBuilder es una herramienta de desarrollo de aplicaciones para el ambiente Windows, como tal utiliza las características de este ambiente. Primero debemos entender que el ambiente Windows permite estar bajo el concepto "Lo que Ud. ve es lo que puede realizar", bajo este contexto la presentación cobra mucha importancia.

PowerBuilder fue uno de los primeros ambientes de programación de aplicaciones Cliente/Servidor en plataformas gráficas en salir al mercado y quizás debido a ello, es una de las más difundidas. Permite definir y usar muy fácilmente objetos de la interfaz Windows. Adicionalmente, permite crear objetos nuevos a partir de objetos preexistentes, sin embargo, no tiene tanta flexibilidad para elaboración de pantallas como la existente en otras herramientas del mundo Windows, como Visual Basic o Delphi.

CAPITULO III
DESARROLLO DE LA APLICACION

DESARROLLO DE LA APLICACIÓN

3.1. INTRODUCCIÓN

Este capítulo permite centrar el estudio en los procesos esenciales del Sistema de Administración Académica y financiera del proyecto la ESPE COMO ACADEMIA REGIONAL DE CISCO, pero para el desarrollo de nuestro proyecto se evito considerar aquellos que pertenecen a otros subsistemas existentes en el Sistema Integrado ESPE, eliminando así la duplicidad de actividades en su contexto general.

Además se determina el alcance, limitaciones y restricciones del Sistema para aportar con información confiable y oportuna, el correcto control de asistencia tanto de docentes como de personal administrativo.

3.2. OBTENCIÓN DE REQUERIMIENTOS

Los requerimientos son las necesidades sobre la función de un producto o servicio, los mismos se los extrae mediante entrevistas, talleres, encuestas, lectura de documentación, y otras formas que posibiliten la obtención de la información.

Mediante la planificación se puede realizar todas las actividades de cualquier institución, el sistema de Administración Académica y Financiera del proyecto la ESPE COMO ACADEMIA REGIONAL DE CISCO, espera ser un aporte medular para la matriculación del alumnado, tomando en cuenta el personal de instructores de la Academia, el tipo de alumno, la carga horaria, horario, además se podrá obtener datos valiosos al momento de emitir la facturación. El área Académica podrá contar con lineamientos más claros y concisos para realizar la Planificación de los procesos a su cargo. Igualmente el área Administrativa y Financiera manejarán información más fiable acerca de los empleados y de esta manera optimizarán sus funciones.

Dentro de los subsistemas identificados tenemos varios procesos que se van a automatizar, se detallan los respectivos procedimientos y actividades de esos procesos dentro del Anexo B: “Mapa de Procesos del Sistema de Administración Académica y Financiera del Proyecto CISCO – ESPE”.

3.2.1. Definiciones

Base de Datos: Conjunto de datos pertenecientes al un mismo contexto y almacenados sistemáticamente para su posterior uso.

Interfaz: Medio de comunicación entre el sistema y el usuario.

Entrada: Datos que recopila el sistema para procesarlos.

Salida: Resultados que el sistema entrega después de procesar los datos ingresados.

Actor: Alguien o algo que interactúa con el sistema; es quien utiliza el sistema.

Rol: Función que alguien o algo cumple.

Validación: Proceso de comprobar la precisión de los datos; conjunto de reglas que se pueden aplicar a un control para especificar el tipo y el intervalo de datos que los usuarios pueden especificar.

Usuario: Persona que utiliza o trabaja con el sistema.

3.3. REQUISITOS DE INTERFACES EXTERNAS

3.3.1. Interfaces de usuario

El usuario utilizará como ambiente gráfico el Sistema Operativo Windows 2000 o XP y la interfaz gráfica que nos provee la herramienta Power Builder. Todas las aplicaciones funcionarán en las estaciones cliente bajo Windows 2000 o Windows XP, los cuales posibilitan que el usuario se desenvuelva en un ambiente visual, para el desarrollo se utiliza Power Builder 9.0i mismo que soporta ambientes gráficos y ofrece una tecnología moderna para poder crear interfaces (GUI) amigables.

3.3.2. Interfaces Hardware

Se recomienda que para el mejor desempeño deberá ser instalado en máquinas Pentium IV, de 128 MB de RAM y por lo menos 20 GB en disco duro, con tarjeta de red.

El Motor de Base de datos debe estar instalado en un servidor de mejores características a las estaciones de trabajo.

3.3.3. Interfaces de software.

El sistema se desarrolla utilizando las siguientes herramientas:

- Diseño en Power Designer 6.1 de 32 Bits
- Desarrollo en Power Builder 9.0
- Windows 2000 (Servidor)
- Windows 2000 o Windows XP (Estaciones)
- Oracle 9.i (Servidor de Base de Datos)
- Personal Oracle 9.i (Clientes de Base de Datos)

3.3.4. Interfaces de comunicaciones

Para un mejor rendimiento del sistema, deberá ser instalado mediante una red, la misma que deberá tener un Switch, cableado estructurado, conectores RJ45, entre otros.

Los protocolos que deberán emplearse serán TCP/IP o NetBios, además de esto debe indicarse que la conexión hacia la Base de Datos será mediante el protocolo ODBC.

Para la implementación del sistema de Administración Académica y Financiera del Proyecto “La ESPE como Academia Regional Cisco, se requiere de las siguientes especificaciones de Hardware y Software:

AMBIENTE CLIENTE – SERVIDOR

✓ Sistema Operativo:	Windows 2000 P. o XP P.
✓ Comunicaciones:	TCP/IP Protocolos
✓ Base de Datos:	ORACLE
✓ Case de modelamiento de Procesos:	Interactúe con Oracle 9.i
✓ Case de modelamiento de Datos:	Interactúe con Oracle 9.i
✓ Software de Desarrollo:	Interactúe con Oracle 9.i (Power Builder 10)

Tabla 3.1. Ambiente Cliente Servidor

3.4. ANÁLISIS

3.4.1. Introducción

El análisis del sistema CISCO –ESPE es un conjunto de procedimientos o programas relacionados de manera que juntos formarán una sola unidad, es un conjunto de hechos, principios y reglas clasificadas y dispuestas de manera ordenada mostrando un plan lógico al momento de la integración de sus partes con el objetivo de alcanzar las metas predefinidas en el proceso de la información.

En esta etapa del desarrollo de software se lleva a cabo ciertos principios que se detalla a continuación:

- ✓ En el levantamiento de la información, debe presentarse y entenderse el dominio de la información de un problema;
- ✓ Se definen las funciones que debe de realizar el software;
- ✓ Se debe representar el comportamiento del software a consecuencia de los acontecimientos externos, y;
- ✓ Se divide en forma jerárquica los modelos que representan la información, funciones y comportamiento.

Los procesos deben partir desde la información esencial hasta el detalle de la Implementación.

El Sistema de Administración Académica y Financiera del Proyecto “La ESPE como Academia Regional Cisco”, está diseñado para garantizar la correcta asistencia de los alumnos a sus clases en los horarios establecidos, además de presentar información pertinente a las diferentes áreas interrelacionadas cuando sea necesario.

El Sistema tiene la Automatización de los diferentes procesos dentro de los subsistemas identificados.

- ✓ Administrar Academias Regionales;
- ✓ Administrar Academias Locales;
- ✓ Administrar Cursos;
- ✓ Administrar Alumnos
- ✓ Administrar Docentes
- ✓ Facturación
- ✓ Administrar Niveles de Seguridad

3.4.2. Definiciones

El análisis del sistema CISCO – ESPE nos permitirá desarrollar un producto que pueda usarse en beneficio de la Escuela Politécnica del Ejército. Para lograr este objetivo el sistema hace uso de los siguientes 6 elementos:

- ✓ **Software.-** Son programas de computadora con estructura de datos y su documentación que hace efectiva la logística y metodología o controles de requerimientos del sistema.

- ✓ **Hardware.-** Dispositivos electrónico y electromecánicos que proporcionan capacidad de calculo y funciones rápidas, exactas y efectivas que proporcionan una función externa dentro de los sistemas.
- ✓ **Personal.-** Son los operadores o usuarios directos de las herramientas del sistema.
- ✓ **Base de Datos.-** Es una gran colección de información organizada y enlazada al sistema a las que se accede a través del software.
- ✓ **Documentación.-** Manuales, formularios, y otra información descriptiva que detalla o da instrucciones sobre la operación del sistema.
- ✓ **Procedimientos.-** Son los pasos que definen el uso específico de cada uno de los elementos o componentes del sistema y las reglas de su manejo y mantenimiento.

3.4.3. ESPECIFICACIÓN DE REQUERIMIENTOS FUNCIONALES

La especificación de requerimientos funcionales que se detalla a continuación está definida en base al acuerdo entre los usuarios y el diseñador del sistema, con respecto a todo lo que se deberá manejar para apoyar la gestión Académica y Finanzas del proyecto la ESPE como Academia regional CISCO.

En la descripción de los requerimientos, se describe el nombre del proceso, del subproceso, las entradas, proceso y salidas, que corresponden a las actividades automatizables del Mapa de procesos.

1. Sistema de Administración Académica y Financiera del proyecto

“CISCO – ESPE”

1.1. Subsistema Académico

1.1.01 Proceso: Administración de Academias Regionales

Código	Actividad:
1.1.01.01	Mantenimiento de Academias Regionales.
Descripción:	<p>Esta actividad contempla varias opciones como:</p> <ul style="list-style-type: none"> • Recopilar la información necesaria • Registrar datos de la Academia Regional • Actualizar Datos de la Academia Regional • Eliminar Academia Regional <p>que permitirá al administrador del sistema el ingreso adecuado de la información a la base de datos correspondiente. Es importante que antes de realizar esta actividad se recolecten y organicen los datos de tal forma que puedan ser administrados correctamente.</p>
Entrada:	<p>Datos organizados de la Academia Regional como:</p> <p>Nombre, Dirección, Ubicación, Teléfonos, Representante Legal, etc.</p>
Proceso:	<p>Administración de la base de datos bajo el esquema de la recopilación, registro, actualización, y eliminación</p>

	datos de la Academia Regional.
Salida:	Registro, Actualización, y Eliminación de los datos de una Academia Regional.

Código	Actividad:
1.1.01.02	Consulta de Academias Regionales.
Descripción:	Esta actividad permite al administrador del sistema hacer consultas en la base de datos de la Academia Regional, bajo los siguientes parámetros: <ul style="list-style-type: none"> • Consulta Datos de la Academia Regional • Realizar Reportes por Academia Regional
Entrada:	Información requerida a ser consultada como: Nombre de la Academia, Dirección, Teléfonos, Numero de academias locales inscritas, etc.
Proceso:	Ingreso de los datos a ser consultados mediante un proceso automático del sistema. Consultar los datos ingresados de la Academia Regional. Elaboración del reporte preestablecido impreso o en pantalla de los datos consultados de la Academia Regional.
Salida:	Base de Datos de la Academia Regional Consultada Satisfactoriamente, en pantalla o impreso.

1.1.02 Proceso: Administración de Academias Locales

Código 1.1.02.01	Actividad: Mantenimiento de Academias Locales
Descripción:	<p>Esta actividad contempla varias opciones como:</p> <ul style="list-style-type: none"> • Recopilar la información necesaria • Registrar datos de la Academia Local • Actualizar Datos de la Academia Local • Eliminar Academia Local <p>que permitirá al administrador del sistema el ingreso adecuado de la información a la base de datos correspondiente. Es importante que antes de realizar esta actividad se recolecten y organicen los datos de tal forma que puedan ser administrados correctamente.</p>
Entrada:	Datos organizados de la Academia Local como: Nombre, Dirección, Ubicación, Teléfonos, Representante Legal, etc.
Proceso:	Administración de la base de datos bajo el esquema de la recopilación, registro, actualización, y eliminación de la Academia Local.
Salida:	Registro, Actualización, y Eliminación de los datos de una Academia Local.

Código 1.1.02.02	Actividad: Consulta de Academias Locales.
Descripción:	Esta actividad permite al administrador del sistema hacer consultas en la base de datos de la Academia Local, bajo los siguientes parámetros: <ul style="list-style-type: none"> • Consulta Datos de la Academia Local • Realizar Reportes por Academia Local.
Entrada:	Información requerida a ser consultada como: Nombre de la Academia, Dirección, Teléfonos, Numero de alumnos inscritos, docentes de la academia, etc.
Proceso:	Ingreso de los datos a ser consultados mediante un proceso automático del sistema. Consultar los datos ingresados de la Academia Regional. Elaboración del reporte preestablecido impreso o en pantalla de los datos consultados de la Academia Local.
Salida:	Base de Datos de la Academia Local Consultada Satisfactoriamente, en pantalla o impreso.

1.1.03 Proceso: Administración del Curso

Código 1.1.03.01	Actividad: Mantenimiento del Curso
Descripción:	Esta actividad contempla varias opciones como: <ul style="list-style-type: none"> • Recopilar la información necesaria

	<ul style="list-style-type: none"> • Registrar datos del curso • Actualizar Datos del Curso • Eliminar Curso <p>que permitirá al administrador del sistema el ingreso adecuado de la información a la base de datos correspondiente. Es importante que antes de realizar esta actividad se recolecten y organicen los datos de tal forma que puedan ser administrados correctamente.</p>
Entrada:	Datos organizados del Curso como: Horario, Aula, Fecha de Inicio, etc.
Proceso:	Administración de la base de datos bajo el esquema de la recopilación, registro, actualización, y eliminación del Curso.
Salida:	Registro, Actualización, y Eliminación de los datos de un Curso.

Código	Actividad:
1.1.03.02	Consulta del Curso.
Descripción:	<p>Esta actividad permite al administrador del sistema hacer consultas en la base de datos del Curso, bajo los siguientes parámetros:</p> <ul style="list-style-type: none"> • Consulta Datos del Curso • Realizar Reportes por Curso.

Entrada:	Información requerida a ser consultada como: Fecha de Inicio, Laboratorio, Numero de alumnos inscritos, Instructor Asignado, etc.
Proceso:	Ingreso de los datos a ser consultados mediante un proceso automático del sistema. Consultar los datos ingresados del Curso. Elaboración del reporte preestablecido impreso o en pantalla de los datos consultados del Curso.
Salida:	Base de Datos del Curso Consultada Satisfactoriamente, en pantalla o impreso.

1.1.04 Proceso: Administración de Alumnos

Código 1.1.04.01	Actividad: Mantenimiento de Alumnos
Descripción:	<p>Esta actividad contempla varias opciones como:</p> <ul style="list-style-type: none"> • Recopilar la información necesaria • Registrar datos del Alumno • Actualizar Datos del Alumno. • Eliminar Alumno. <p>que permitirá al administrador del sistema el ingreso adecuado de la información a la base de datos correspondiente. Es importante que antes de realizar</p>

	esta actividad se recolecten y organicen los datos de tal forma que puedan ser administrados correctamente.
Entrada:	Datos organizados del Alumno como: Tipo de Alumno, Nombre, Dirección, Teléfonos, Representante Legal, etc.
Proceso:	Administración de la base de datos bajo el esquema de la recopilación, registro, actualización, y eliminación del Alumno.
Salida:	Registro, Actualización, y Eliminación de los datos de un Alumno.

Código 1.1.04.02	Actividad: Consulta de Alumnos.
Descripción:	Esta actividad permite al administrador del sistema hacer consultas en la base de datos del Alumno, bajo los siguientes parámetros: <ul style="list-style-type: none"> • Consulta Datos del Alumno. • Realizar Reportes por Alumno.
Entrada:	Información requerida a ser consultada como: Tipo del Alumno, Nombre del Alumno, Dirección, Teléfonos, etc.
Proceso:	Ingreso de los datos a ser consultados mediante un proceso automático del sistema.

	<p>Consultar los datos ingresados del Alumno.</p> <p>Elaboración del reporte preestablecido impreso o en pantalla de los datos consultados del Alumno.</p>
Salida:	<p>Base de Datos del Alumno Consultada</p> <p>Satisfactoriamente, en pantalla o impreso.</p>

1.1.05 Proceso: Administración de Instructores

Código	Actividad:
1.1.05.01	Mantenimiento de Instructores
Descripción:	<p>Esta actividad contempla varias opciones como:</p> <ul style="list-style-type: none"> • Recopilar la información necesaria • Registrar datos de Instructores. • Actualizar Datos de Instructores. • Eliminar Instructores. <p>que permitirá al administrador del sistema el ingreso adecuado de la información a la base de datos correspondiente. Es importante que antes de realizar esta actividad se recolecten y organicen los datos de tal forma que puedan ser administrados correctamente.</p>
Entrada:	<p>Datos organizados del Instructor como:</p> <p>Nombre, Dirección, Teléfonos, etc.</p>
Proceso:	<p>Administración de la base de datos bajo el esquema de la recopilación, registro, actualización, y eliminación de</p>

	Instructores.
Salida:	Registro, Actualización, y Eliminación de los datos de un Instructor.

Código	Actividad:
1.1.05.02	Consulta de Instructores.
Descripción:	Esta actividad permite al administrador del sistema hacer consultas en la base de datos del Instructor, bajo los siguientes parámetros: <ul style="list-style-type: none"> • Consulta Datos del Instructor. • Realizar Reportes por Instructores.
Entrada:	Información requerida a ser consultada como: Nombre, Dirección, Teléfonos, academia, etc.
Proceso:	Ingreso de los datos a ser consultados mediante un proceso automático del sistema. Consultar los datos ingresados del Instructor Elaboración del reporte preestablecido impreso o en pantalla de los datos consultados del Instructor.
Salida:	Base de Datos del Instructor Consultada Satisfactoriamente, en pantalla o impreso.

1.2. Subsistema Financiero

1.2.01 Proceso: Administración de Alumnos

Código 1.2.01.01	Actividad: Mantenimiento de Alumnos.
Descripción:	Esta Actividad dentro del subsistema financiero corresponde al cliente que este caso es el alumno que se inscribirá en un curso que oferta la academia de Cisco en la ESPE, el alumno se escoge directamente de la base de datos Alumnos en el subsistema Académico, en el caso de que no conste al momento de la matricula, se lo podrá registrar mediante este modulo.
Entrada:	Datos organizados del Alumno como: Nombre, Dirección, Teléfonos, Estudios realizados, etc.
Proceso:	Registrar los Datos mediante el Sistema, y en la base de datos correspondiente.
Salida:	Datos registrados.

Código 1.2.01.02	Actividad: Consulta de Alumnos.
Descripción:	Realiza consultas de los datos ingresados en la base de datos del Alumno, con las opciones del sistema de: <ul style="list-style-type: none"> • Consultar Datos del Alumno

	<ul style="list-style-type: none"> Realizar Reporte por Alumno.
Entrada:	Datos a Consultar.
Proceso:	Realizar la búsqueda de los datos necesarios para la consulta.
Salida:	Datos Consultados Satisfactoriamente.

1.2.02 Proceso: Administración de Facturas

Código	Actividad:
1.2.02.01	Registrar el alumno que se inscribe en un curso
Descripción:	Esta actividad permite registrar los datos previamente ingresados del alumno, bajo las opciones de registrar al alumno que se inscribe en un curso, en la factura, esta actividad es importante, porque se necesita conocer el tipo de alumno que es para el descuento respectivo.
Entrada:	Datos organizados del Alumno como: Nombre, Dirección, Teléfonos, Estudios realizados, Curso en el que se inscribe, tipo de alumno que es, etc.
Proceso:	Registrar los Datos de alumno mediante el Sistema, organizando la información en la factura respectiva.
Salida:	Datos del alumno registrados en la factura.

Código 1.2.02.02	Actividad: Generación de Valores a ser Facturados
Descripción:	Esta actividad permite calcular los valores a ser facturados por alumno inscrito en un curso, el administrador del sistema podrá calcular los valores correspondientes bajo las siguientes opciones: <ul style="list-style-type: none"> • Calculo de Valores de descuento • Calculo del Valor total de la Factura • Registro de la forma de Pago.
Entrada:	Datos del tipo de alumno que se inscribe: Alumno Militar ESPE y Militar = 250 Alumno ESPE = 250 Egresado = 270 Alumno Particular = 290.
Proceso:	Registro de los Datos del tipo de alumno mediante el Sistema, organizando la información en la factura para el respectivo cálculo del descuento.
Salida:	Valores de la Factura calculados satisfactoriamente.

Código 1.2.02.03	Actividad: Registro del Curso
Descripción:	Esta actividad permitirá al usuario del sistema registrar al alumno en el curso y horario que el alumno matriculado

	<p>haya escogido, esto dependerá del número máximo y mínimo que necesita un curso para dictarse, el administrador del sistema contará con las siguientes opciones:</p> <ul style="list-style-type: none"> • Registro del alumno al Curso y Horario escogido • Asignación del Instructor al Curso • Asignación del Laboratorio al Curso.
Entrada:	<p>Horario del Curso</p> <p>Paralelo</p> <p>Laboratorio</p>
Proceso:	<p>Registro, Asignación de Instructor, y Laboratorio al Alumno al curso matriculado.</p>
Salida:	<p>Alumno registrado en el curso satisfactoriamente.</p>

1.2.03 Proceso: Facturación

Código	Actividad:
1.2.03.01	Mantenimiento de Facturas.
Descripción:	<p>Esta actividad permitirá al administrador, almacenar las facturas emitidas a los alumnos en el sistema de forma automática, a través de las siguientes opciones:</p> <ul style="list-style-type: none"> • Registrar Datos de la Factura • Actualizar Datos de la Factura

	<ul style="list-style-type: none"> Anulación de la Factura
Entrada:	Facturas emitidas. Numero de Facturas emitidas en un periodo.
Proceso:	Registrar, Actualizar y Anular las facturas emitidas en el sistema de un periodo determinado.
Salida:	Facturas registradas Satisfactoriamente.

Código	Actividad:
1.2.03.02	Consultas de facturas.
Descripción:	<p>Esta actividad permitirá al usuario del sistema consultar los datos de las facturas emitidas a los alumnos en el sistema de forma automática, a través de las siguientes opciones:</p> <ul style="list-style-type: none"> Consultar datos de la Factura Realizar reportes por Factura
Entrada:	Número de factura a consultar. Nombre o código del alumno dueño de la factura.
Proceso:	Realizar las consultas de acuerdo a un criterio escogido.
Salida:	Facturas consultadas Satisfactoriamente.

1.3. Subsistema de Seguridades

1.3.01 Proceso: Opciones de Acceso

Código 1.3.01.01	Actividad: Establecer opciones de Acceso.
Descripción:	Esta Actividad permitirá al usuario del sistema registrar o almacenar los niveles de acceso al sistema mediante los diferentes niveles de acceso del personal que manejará el sistema, esta activada permite al administrador las siguientes opciones: * Registrar nivel de Acceso.
Entrada:	Niveles de Acceso Perfil del usuario.
Proceso:	Registrar los niveles de acceso al sistema de acuerdo al perfil de cada usuario.
Salida:	Niveles de Acceso Registrados Satisfactoriamente.

1.3.02 Proceso: Perfiles de Usuario

Código 1.3.02.01	Actividad: Determinar perfiles de Usuario
Descripción:	Esta Actividad permitirá al administrador del sistema asignar el nivel de acceso a un usuario, luego de que se haya determinado el nivel de acceso correspondiente,

	<p>esta actividad permite al administrador del sistema las siguientes opciones:</p> <ul style="list-style-type: none"> • Asignar nivel de Acceso • Actualizar perfiles de Usuarios • Registrar Perfiles de Usuario. <p>Nota: el administrador del sistema tendrá acceso a todo.</p>
Entrada:	<p>Identificar Usuario</p> <p>Identificar el nivel de acceso.</p>
Proceso:	<p>Registrar los niveles de acceso al sistema de acuerdo al perfil de cada usuario.</p>
Salida:	<p>Nivel de acceso asignado Satisfactoriamente.</p>

Código	Actividad:
1.3.03.01	Creación de Usuario
Descripción:	<p>Esta actividad permitirá al administrador del sistema almacenar los datos del usuario del sistema entendiéndose que usuario es la persona que maneja el sistema y al cual se le debe dar el acceso correspondiente, el administrador cuenta con las siguientes opciones:</p> <ul style="list-style-type: none"> • Registrar Usuario en el Sistema • Asignar un Login y un Password al Usuario • Actualizar la base de Datos

	<ul style="list-style-type: none"> • Habilitar aplicación correspondiente al Perfil
Entrada:	<p>Datos del usuario</p> <p>Identificación del Usuario</p> <p>Perfil Especificado.</p> <p>Nivel de acceso.</p>
Proceso:	<p>Registrar al usuario en el sistema de acuerdo al perfil y nivel de acceso asignado.</p>
Salida:	<p>Usuario registrado Satisfactoriamente.</p>

Código	Actividad:
1.3.03.02	Mantenimiento de Usuarios
Descripción:	<p>Esta actividad permitirá al administrador del sistema hacer cambios en la base de datos del usuario. Es importante aclarar que antes de hacer esta actividad se necesita de una autorización correspondiente, el administrador del sistema cuenta con las siguientes opciones:</p> <ul style="list-style-type: none"> • Realizar cambios solicitados al Usuario • Actualizar tabla de Usuarios • Realizar reportes por usuario.
Entrada:	<p>Autorización correspondiente</p> <p>Nombre del usuario</p> <p>Datos a cambiarse</p>

Proceso:	Identificación del usuario. Identificación de los datos a cambiarse.
Salida:	Datos cambiados del usuario Satisfactoriamente.

3.4.4. SEGURIDAD

La seguridad del sistema estará dada por claves de acceso según los perfiles de usuarios para garantizar el uso de la información de forma segura.

Se contará con la administración adecuada de perfiles de usuario para garantizar el uso de la información en forma segura.

La aplicación permitirá sacar respaldos de la información periódicamente.

3.4.5. MANTENIMIENTO

El mantenimiento del sistema se realizará de una forma fácil de manejar con interfaces en cada subsistema para que cada usuario sea dueño que la información que necesita.

Las pantallas de mantenimientos siguen un estándar después detallado que sirven al usuario para realizar óptimamente sus procesos.

Las pantallas de mantenimiento que son comunes para todo el sistema se recopilarán en un solo menú para que sea usada y mantenida por el Departamento de Organización y Sistemas.

3.5. DISEÑO DEL SISTEMA

3.5.1. Introducción

El diseño del sistema se define como el proceso de aplicar ciertas técnicas y principios con el propósito de definir un dispositivo, un proceso o un sistema con suficientes detalles como para permitir su interpretación y realización física.

El diseño del sistema encierra cuatro etapas:

- ✓ El diseño de Datos
- ✓ El diseño Arquitectónico
- ✓ El diseño de la Interfaz
- ✓ El diseño de Procedimientos

El diseño del sistema es la implementación de todo el levantamiento de la información en la etapa de análisis y se ve plasmado en la arquitectura, plataforma, y componentes.

3.5.2 Mapa de Procesos (Anexo B)

El mapa de procesos es el resultado del análisis de los datos a los diferentes actores interesados en el desarrollo del Sistema CISCO – ESPE, a continuación se detalla el mapa de procesos del sistema:

MAPA DE PROCESOS DEL SISTEMA DE ADMINISTRACION ACADEMICA Y FINANCIERA DEL PROYECTO

"CISCO - ESPE"

1. SISTEMA DE ADMINISTRACION ACADEMICA Y FINANCIERA

SUBSISTEMA	PROCESO	ACTIVIDADES	AUTOMATIZABLE	
1,1 A C A D E M I C O	1.1.01	1.1.01.01	MANTENIMIENTO DE ACADEMIAS REGIONALES	
			Recopilar la información necesaria	No
			Registrar Datos de la Academia Regional	Si
			Actualizar Datos de la Academia Regional	Si
		Eliminar Academia Regional	Si	
		1.1.01.02	CONSULTA DE ACADEMIAS REGIONALES	
	Consultar Datos de la Academia Regional	Si		
	Realizar Reportes por Academia Regional	Si		
	1.1.02	1.1.02.01	MANTENIMIENTO DE ACADEMIAS LOCALES	
			Recopilar la Información Necesaria	No
			Registrar Datos de la Academia Local	Si
			Actualizar Datos de la Academia Local	Si
		Eliminar Academia Local	Si	
		1.1.02.02	CONSULTA DE ACADEMIAS LOCALES	
	Consultar Datos de la Academia Local	Si		
	Realizar Reportes por Academia Local	Si		
	1.1.03	1.1.03.01	MANTENIMIENTO DE L CURSO	
			Recopilar la Información Necesaria	No
			Registrar Datos del Curso	Si
			Actualizar Datos del Curso	Si
		Eliminar Curso	Si	
		1.1.03.02	CONSULTA DEL CURSO	
	Consultar Datos del Curso	Si		
	Realizar Reportes por Curso	Si		
1.1.04	1.1.04.01	MANTENIMIENTO DE ALUMNOS		
		Recopilar la Información Necesaria	No	
		Registrar Datos del Alumno	Si	
		Actualizar Datos del Alumno	Si	
	Eliminar Alumno	Si		
	1.1.04.02	CONSULTA DE ALUMNOS		
Consultar Datos del Alumno	Si			
Realizar Reportes por Alumno	Si			
1.1.05	1.1.05.01	MANTENIMIENTO INSTRUCTORES		
		Recopilar la Información Necesaria	No	
		Registrar Datos del Instructor	Si	
		Actualizar Datos del Instructor	Si	
	Eliminar Instructor	Si		
	1.1.05.02	CONSULTAS INSTRUCTORES		
Consultar Datos del Instructor	Si			
Realizar Reportes por Instructor	Si			
1,2 F I N A N	1.2.01	1.2.01.01	MANTENIMIENTO DE ALUMNOS	
			Recopilar la Información Necesaria	No
			Registrar Datos del Alumno	Si
			Actualizar Datos del Alumno	Si
		Eliminar Alumno	Si	
		1.2.01.02	CONSULTA DE ALUMNOS	
	Consultar Datos del Alumno	Si		
	Realizar Reportes por Alumno	Si		
	1.2.02	1.2.02.01	INGRESAR FACTURA	
			Registrar el Alumno que se Inscribe en un Curso	Si
		Registrar el Curso en el que se Inscribe	Si	
		1.2.02.02	GENERAR VALORES	
Calcular Valor de descuento de acuerdo al tipo de alumno que se inscribe en un curso			Si	
Calcular Valor Total de la Factura a Cobrar			Si	
Registrar Forma de Pago de la Factura	Si			

C I E R O			1.2.02.03	REGISTRO DEL CURSO	
				Registro del Alumno al Curso y Horario Escogido	Si
				Asignacion del Instructor al Curso.	Si
				Asignacion del Laboratorio al Curso	Si
	1.2.03	FACTURACIÓN	1.2.03.01	MANTENIMIENTO DE FACTURAS	
				Registrar Datos de la Factura	Si
				Receptar Solicitud de Mantenimiento de Factura	No
				Verificacion de Datos a Modificar	No
				Actualizar Datos de la Factura	Si
				Receptar Solicitud para la anulacion de la Factura	No
			1.2.03.02	CONSULTAS DE FACTURAS	
				Consultar Datos de la Factura	Si
				Realizar Reportes por Factura	Si
Archivar Copia de la Factura				No	
1,3 S E G U R I D A D E S	1.3.01	OPCIONES DE ACCESO	1.3.01.01	ESTABLECER OPCIONES DE ACCESO	
				Identificar Aplicaciones del Sistema	No
				Identificar Módulos	No
				Identificar Areas de Aplicación	No
				Relacionar Aplicaciones, Módulos y Areas	No
				Asignar Conjunto de Opciones	No
				Determinar Nivel de Acceso	No
	Registrar Nivel de Acceso	Si			
	1.3.02	PERFILES DE USUARIO	1.3.02.01	DETERMINAR PERFILES DE USUARIO	
				Identificar Usuarios	No
				Agrupar Usuarios	No
				Especificar Perfiles	No
				Determinar Funciones por Perfil	No
1.3.03	ADMINISTRACION DE USUARIOS	1.3.03.01	CREACION DE USUARIO		
			Receptar Solicitud de Creación de Usuarios	No	
			Verificar Datos	No	
			Determinar Perfil	No	
		1.3.03.02	MANTENIMIENTO DE USUARIOS		
			Registrar Usuario en el Sistema	Si	
			Asignar un Login y un Password al Usuario	Si	
			Actualizar la base de Datos	Si	
				Habilitar Aplicación correspondiente al Perfil	Si
				Receptar Solicitud de Mantenimiento de Usuarios	No
				Realizar Cambios Solicitados al Usuario	Si
				Actualizar Tabla de Usuarios	Si
				Realizar Reportes por usuario	Si

Tabla 3.2. Mapa de Procesos

3.5.3 DFD'S (Anexo C)

MODELO DEL NEGOCIO
SISTEMA CISCO ESPE

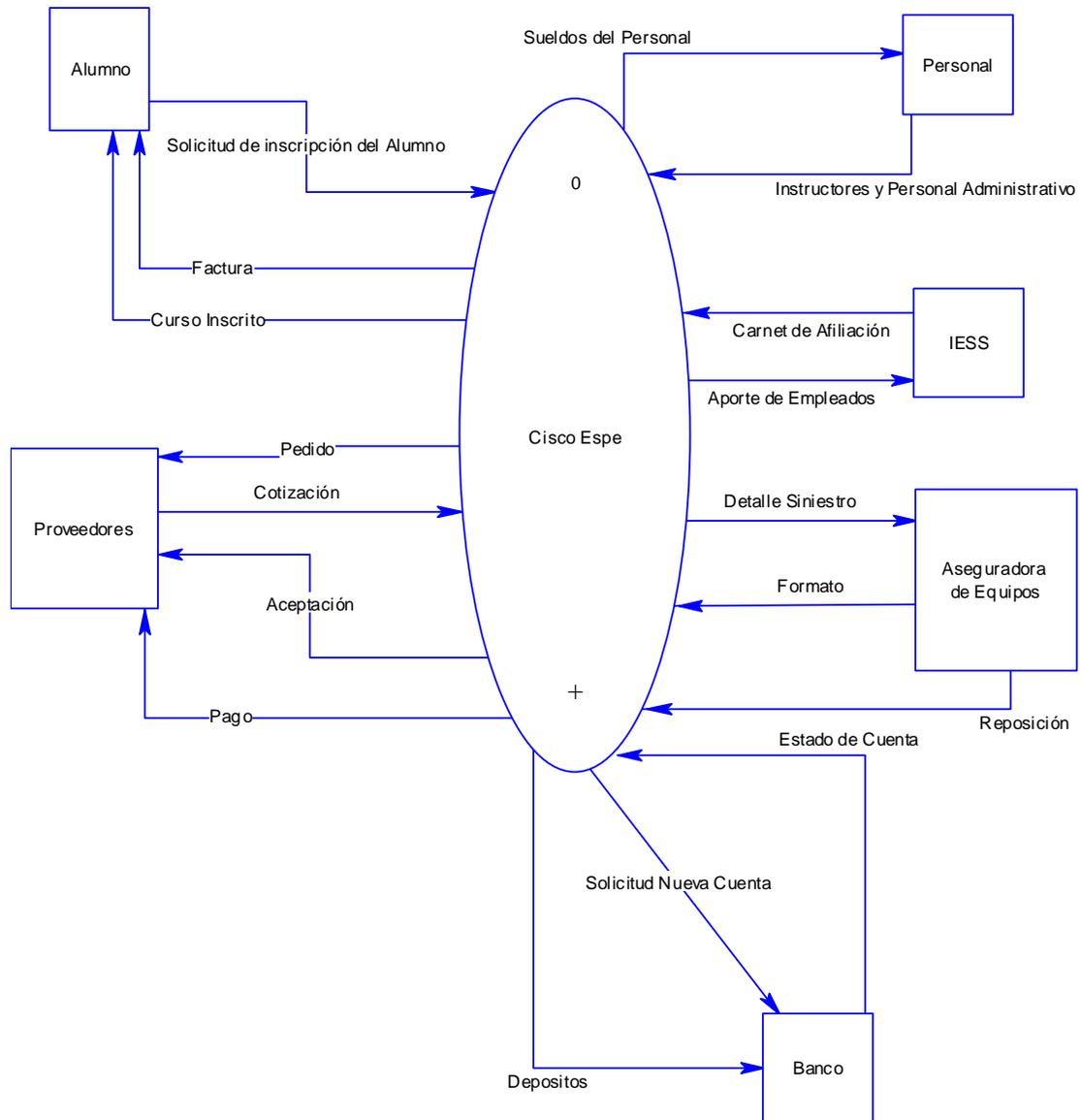


Figura 3.1. Modelo del Negocio

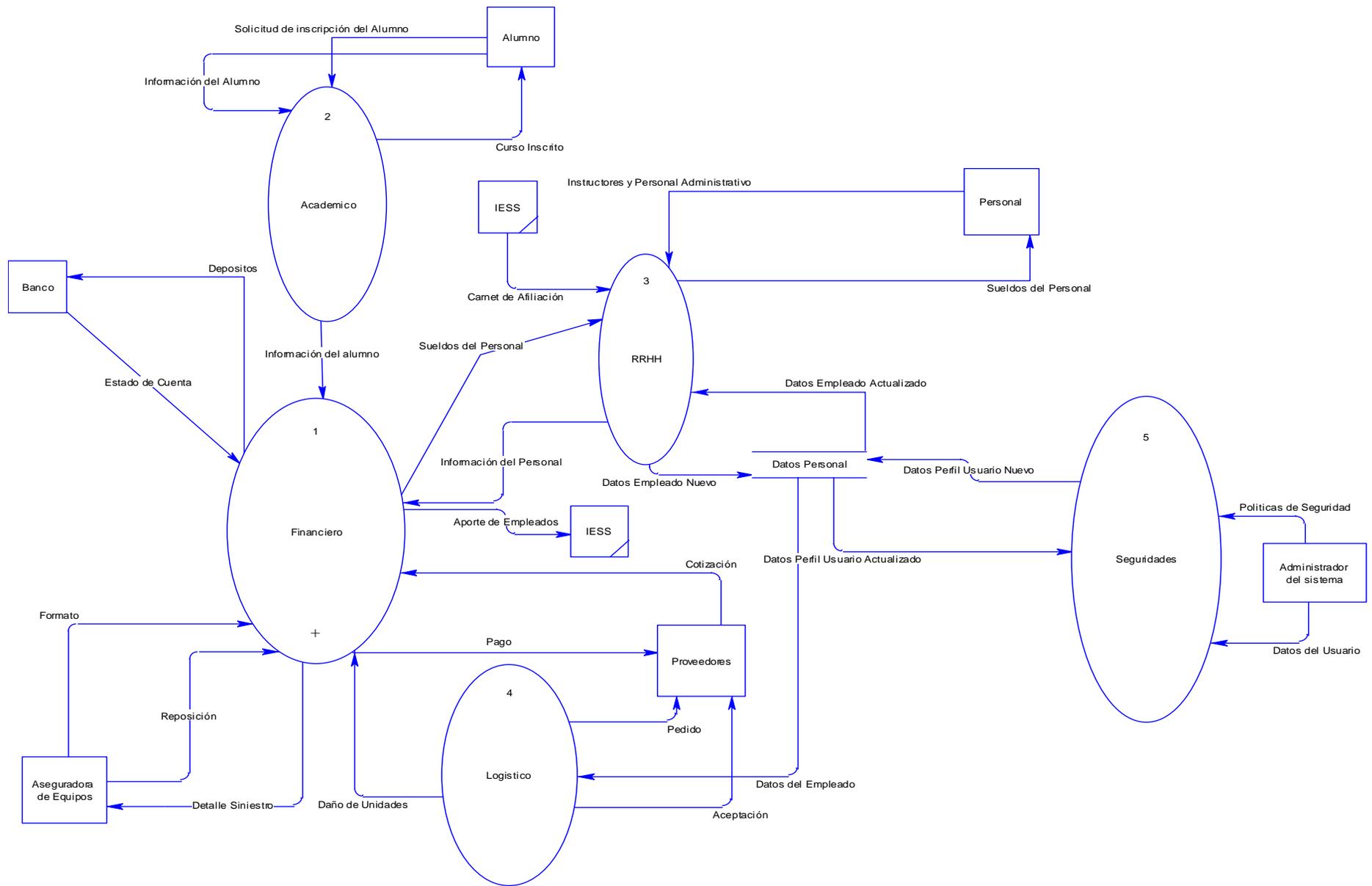


Figura 3.2. Modelo del Negocio

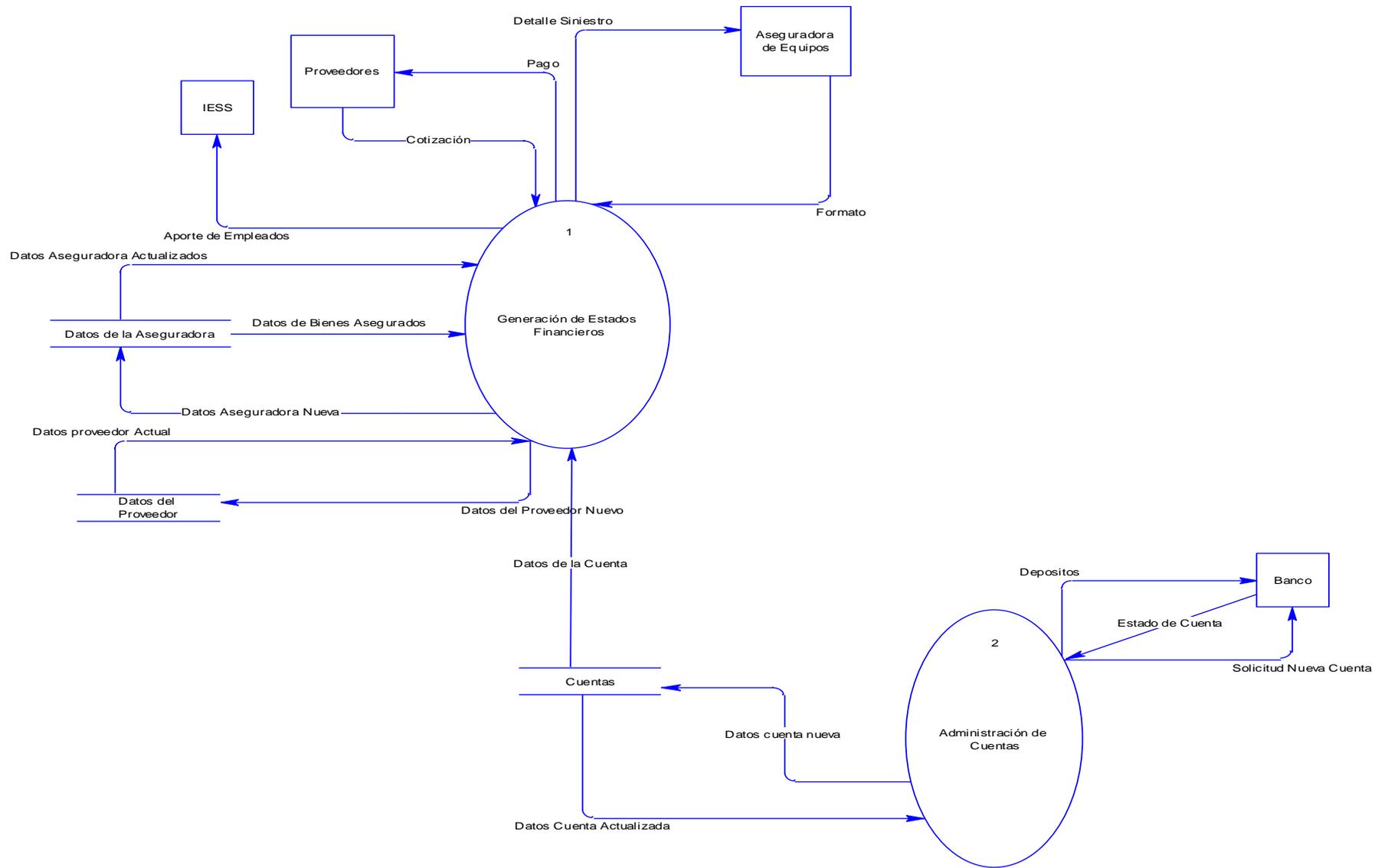


Figura 3.3. Modelo del Negocio

3.5.4. Diagramas Conceptuales (Anexo D)

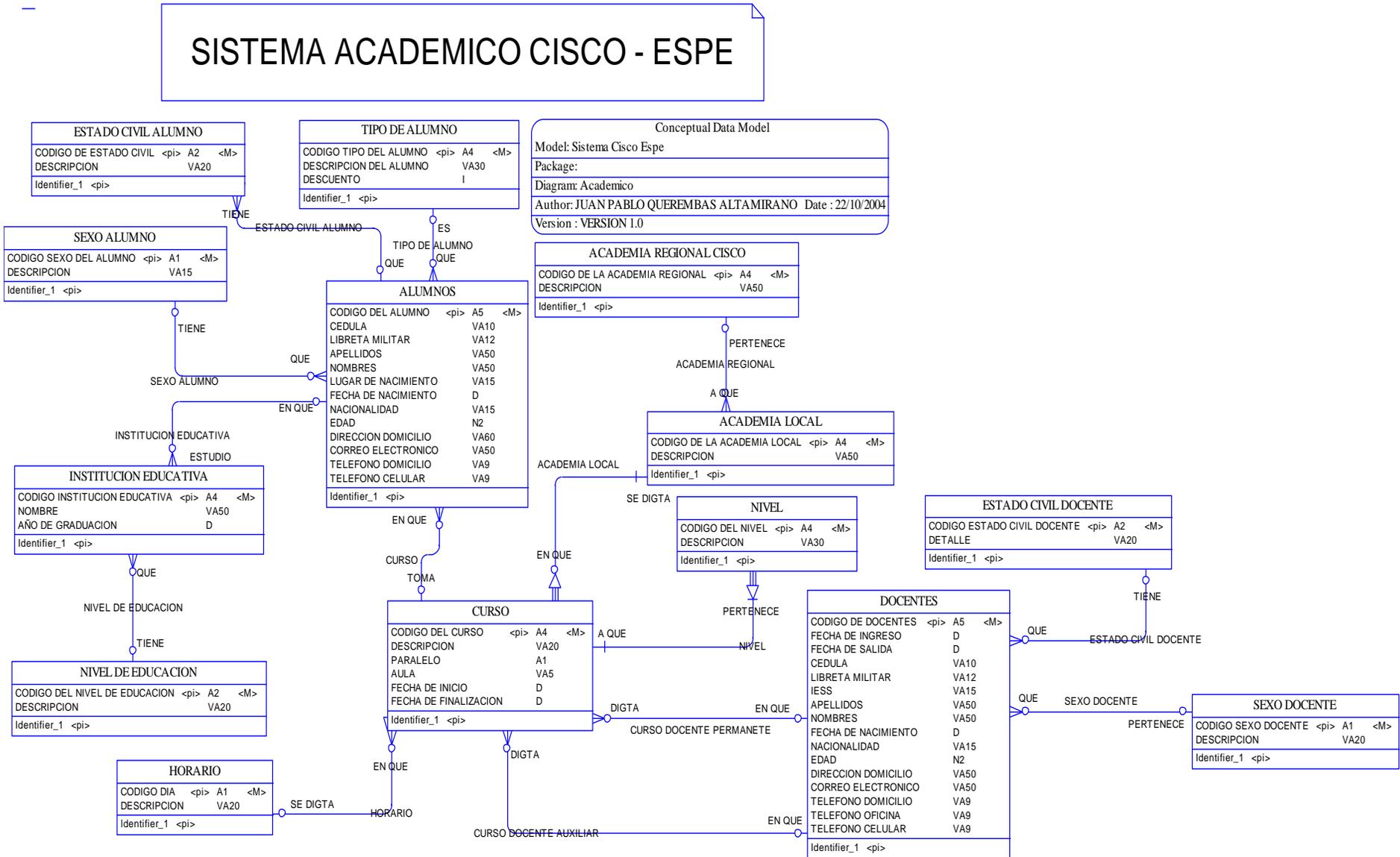


Figura 3.4. Diagrama Conceptual Académico

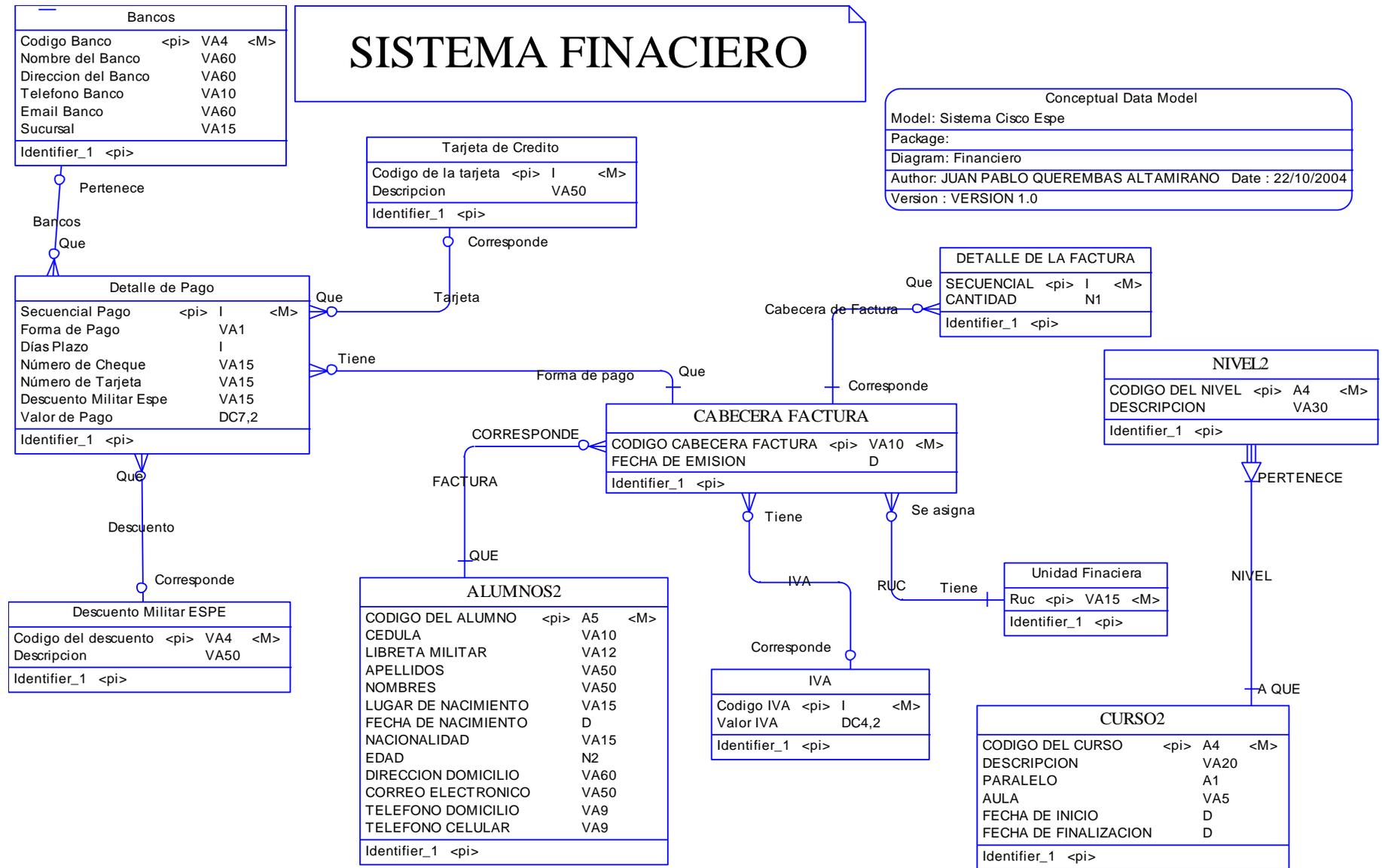


Figura 3.5. Diagrama Conceptual Financiero

SISTEMA DE SEGURIDADES

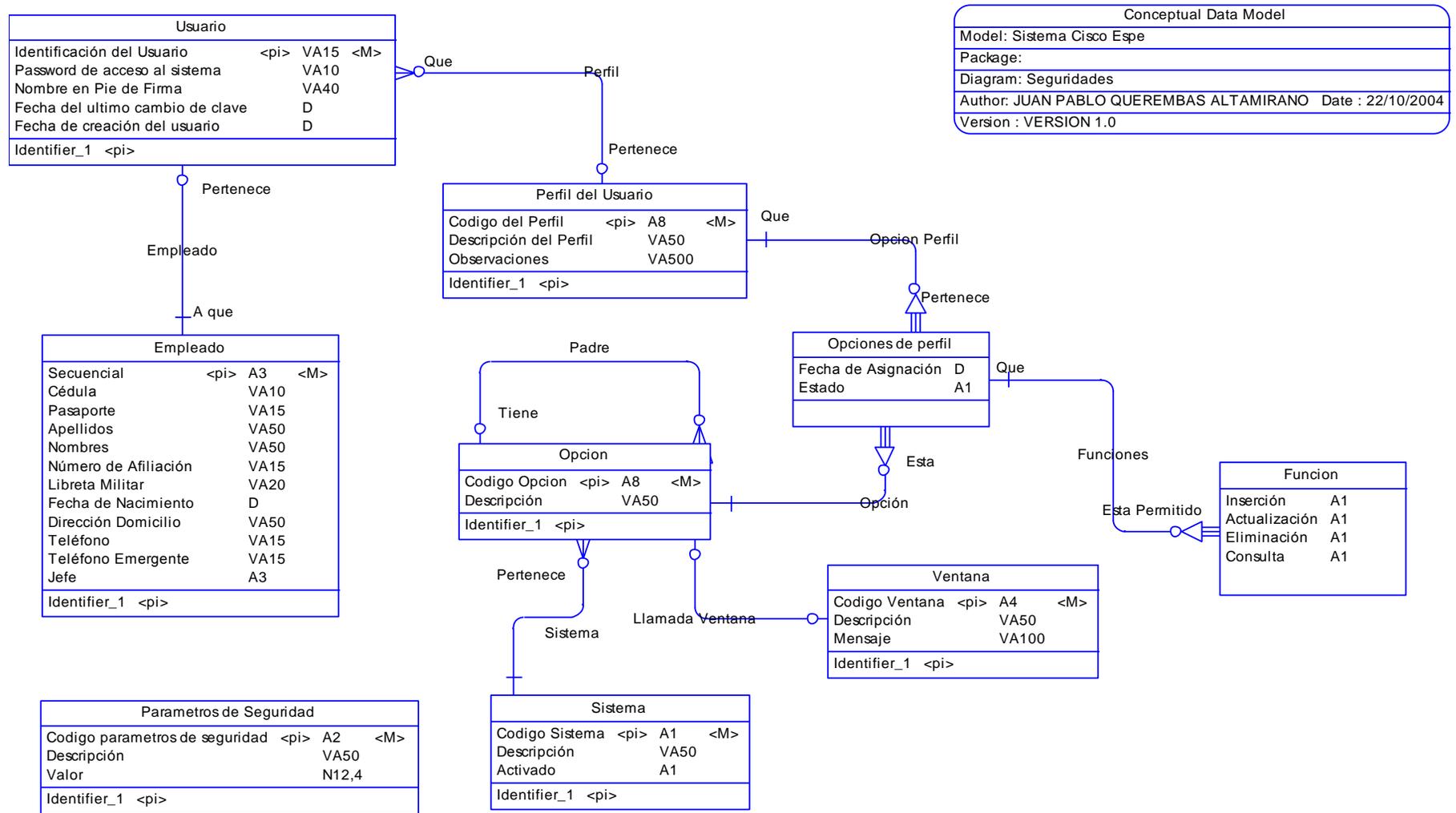


Figura 3.6. Diagrama Conceptual Seguridades

3.5.5. Casos de Uso (Anexo E)

3.5.5.1. Especificación de Actores

ACTOR	CASO DE USO
USUARIO	Ingreso de Academias Regionales Ingreso de Academias Locales Ingresos de Proyectos Ingreso de Subproyectos Ingreso de Ciudadanos Ingreso de Docentes Ingreso de Estudiantes Ingreso de Proveedores Ingreso de Tipo de Cartera Ingreso de Tipos de Comprobantes Ingreso de Estados de Pago Ingresos de Tipo de Pago Registro de Cursos Asignación de Estudiantes a un Curso Seguimiento de Cartera Consultas y Reportes
SUPERVISOR	Modificación de Academias Regionales Modificación de Academias Locales Modificación de Proyectos

	<p>Modificación de Subproyectos</p> <p>Ingreso de Aportes</p> <p>Modificación de Aportes</p> <p>Ingreso del Período Académico</p> <p>Modificación del Período Académico</p> <p>Ingreso del Estado del Plan</p> <p>Modificación del Estado del Plan</p> <p>Modificación de Ciudadanos</p> <p>Modificación de Docentes</p> <p>Modificación de Estudiantes</p> <p>Modificación de Proveedores</p> <p>Modificación de Tipo de Cartera</p> <p>Modificación de Tipos de Comprobantes</p> <p>Modificación de Estados de Pago</p> <p>Modificación de Tipos de Pago</p> <p>Ingreso de Usuarios</p> <p>Consultar Usuarios por Perfil</p> <p>Asignación de Perfil de Usuario</p>
<p>ADMINISTRADOR</p>	<p>Ingreso de Academias Regionales</p> <p>Ingreso de Academias Locales</p> <p>Ingresos de Proyectos</p> <p>Ingreso de Subproyectos</p> <p>Ingreso de Ciudadanos</p>

	Ingreso de Docentes
	Ingreso de Estudiantes
	Ingreso de Proveedores
	Ingreso de Tipo de Cartera
	Ingreso de Tipos de Comprobantes
	Ingreso de Estados de Pago
	Ingresos de Tipo de Pago
	Registro de Cursos
	Asignación de Estudiantes a un Curso
	Seguimiento de Cartera
	Consultas y Reportes
	Modificación de Academias Regionales
	Modificación de Academias Locales
	Modificación de Proyectos
	Modificación de Subproyectos
	Ingreso de Aportes
	Modificación de Aportes
	Ingreso del Período Académico
	Modificación del Período Académico
	Ingreso del Estado del Plan
	Modificación del Estado del Plan
	Modificación de Ciudadanos
	Modificación de Docentes

	<p>Modificación de Estudiantes</p> <p>Modificación de Proveedores</p> <p>Modificación de Tipo de Cartera</p> <p>Modificación de Tipos de Comprobantes</p> <p>Modificación de Estados de Pago</p> <p>Modificación de Tipos de Pago</p> <p>Ingreso de Usuarios</p> <p>Consultar Usuarios por Perfil</p> <p>Asignación de Perfil de Usuario</p> <p>Creación de Nuevos Perfiles</p>
--	---

3.5.5.2. Estructura de Casos de Uso

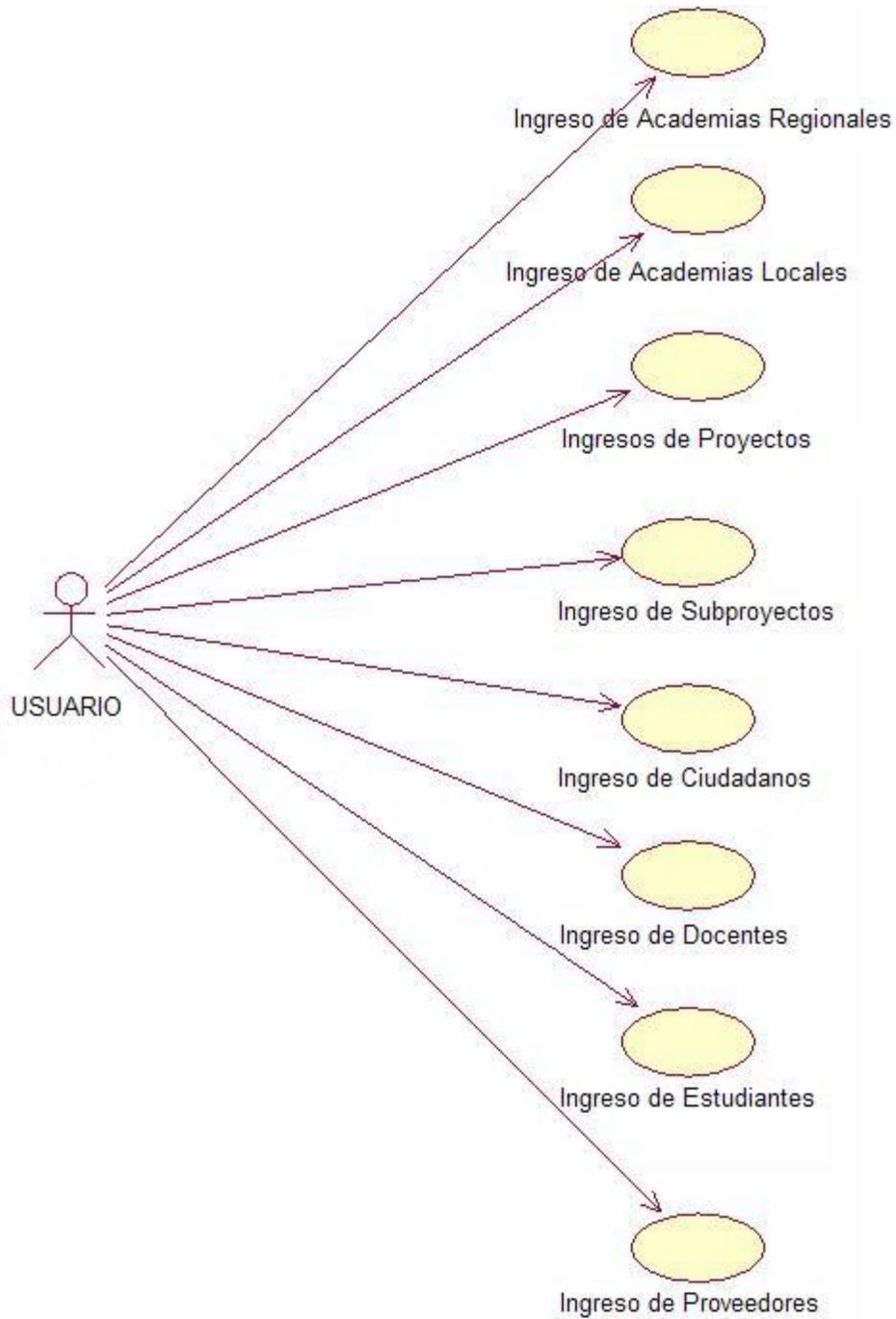


Figura 3.7. Diagrama de Caso de Uso Usuario

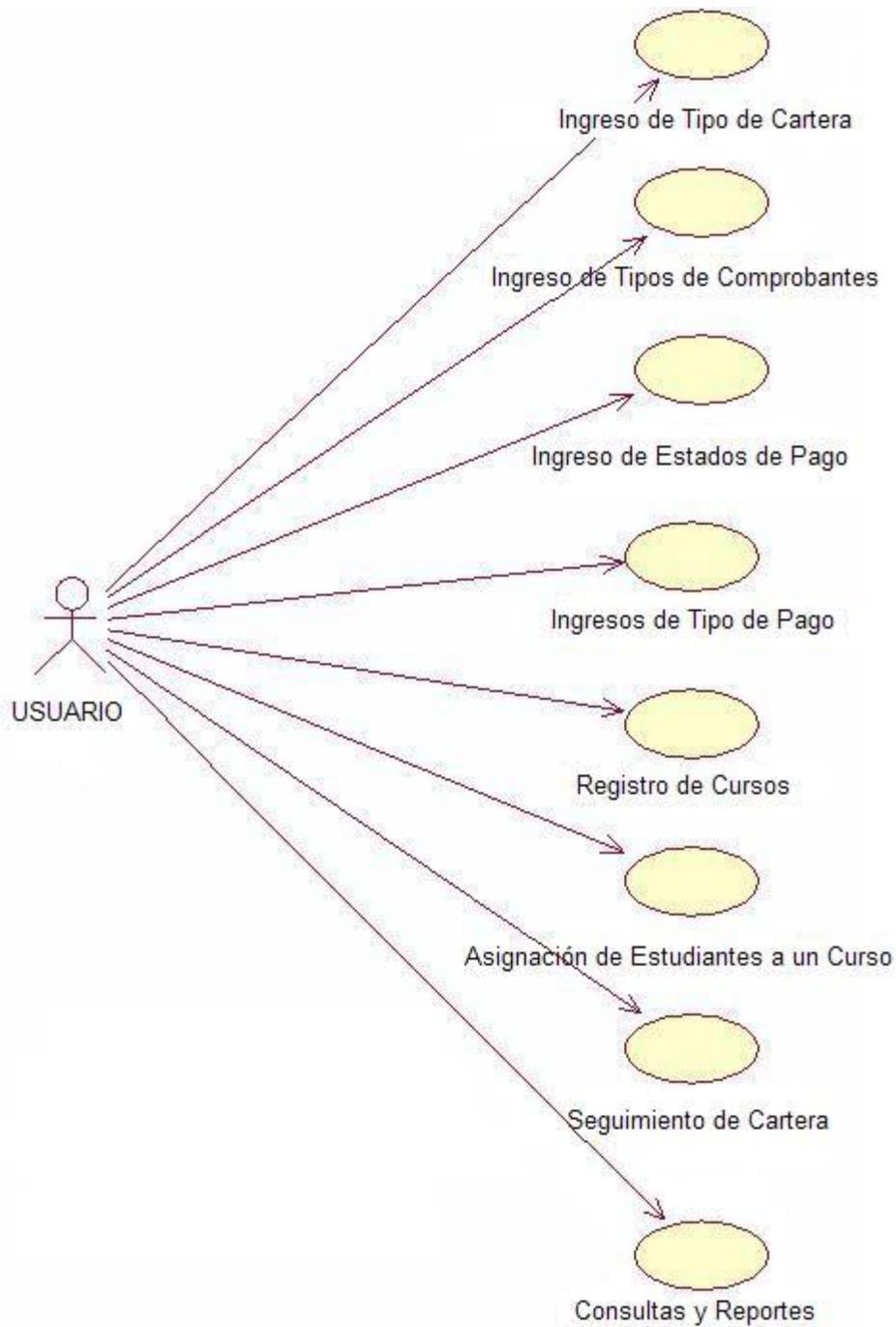


Figura 3.8. Diagrama de Caso de Uso Usuario

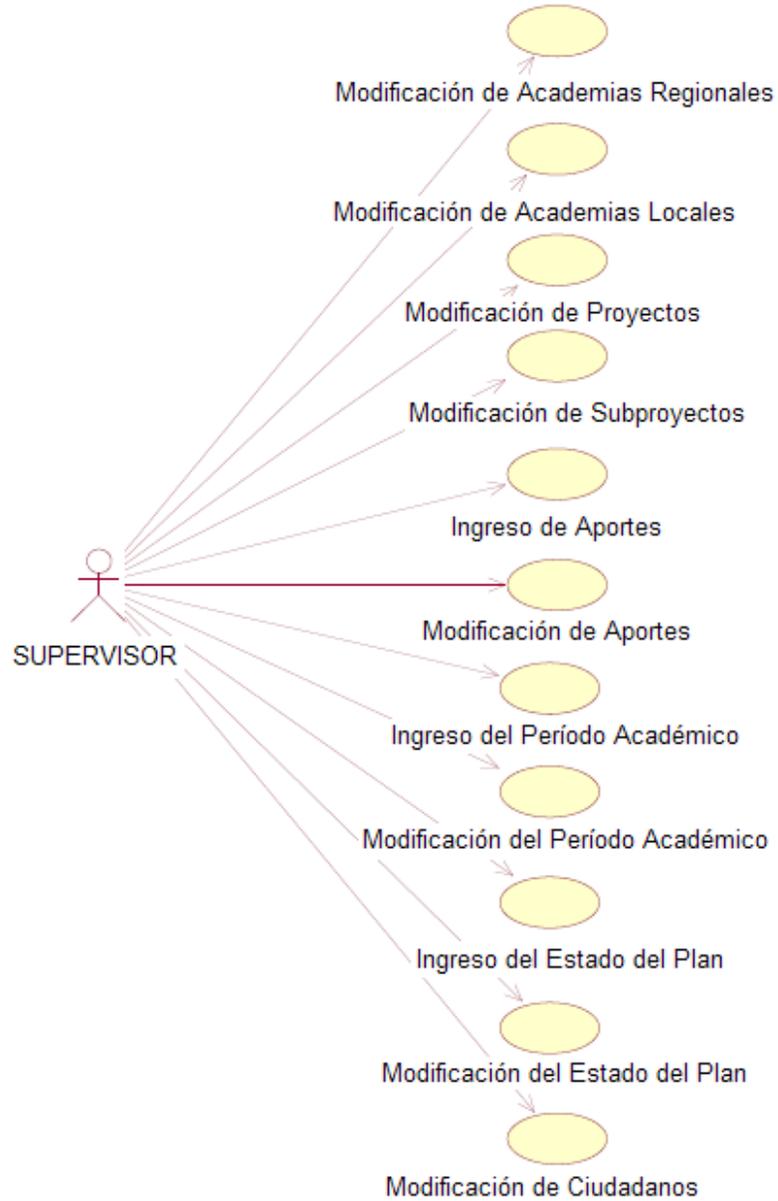


Figura 3.9. Diagrama de Caso de Uso Supervisor

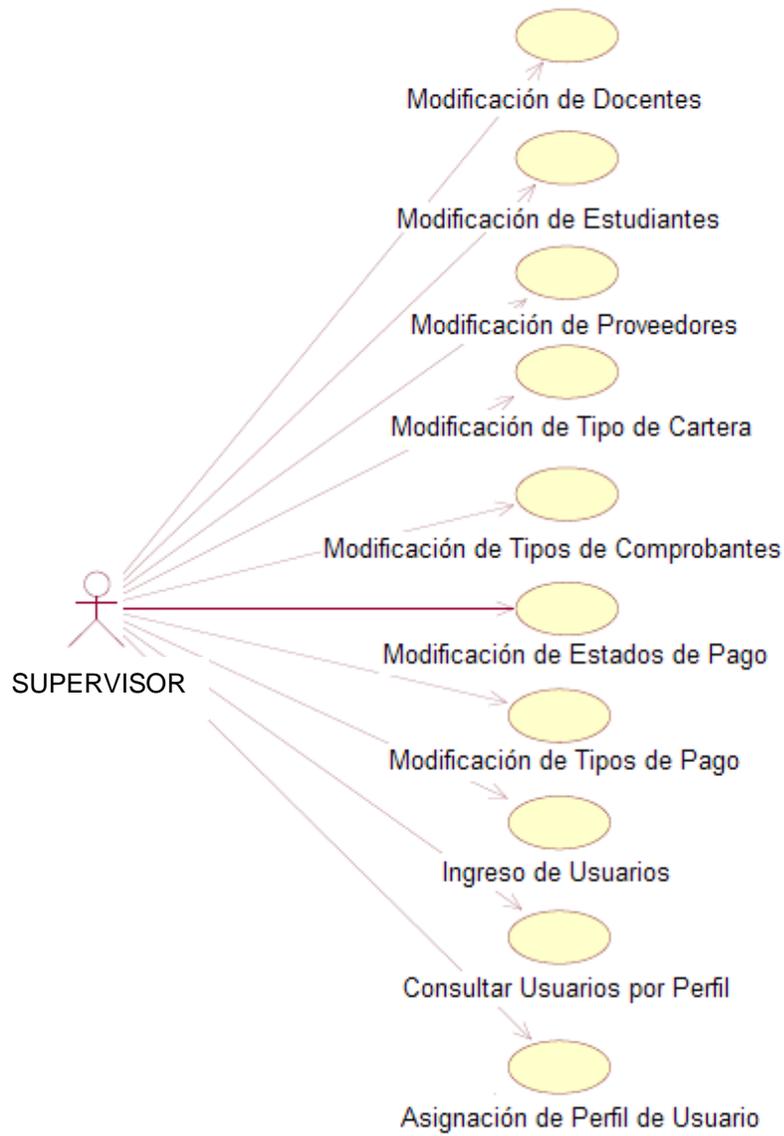


Figura 3.10. Diagrama de Caso de Uso Supervisor

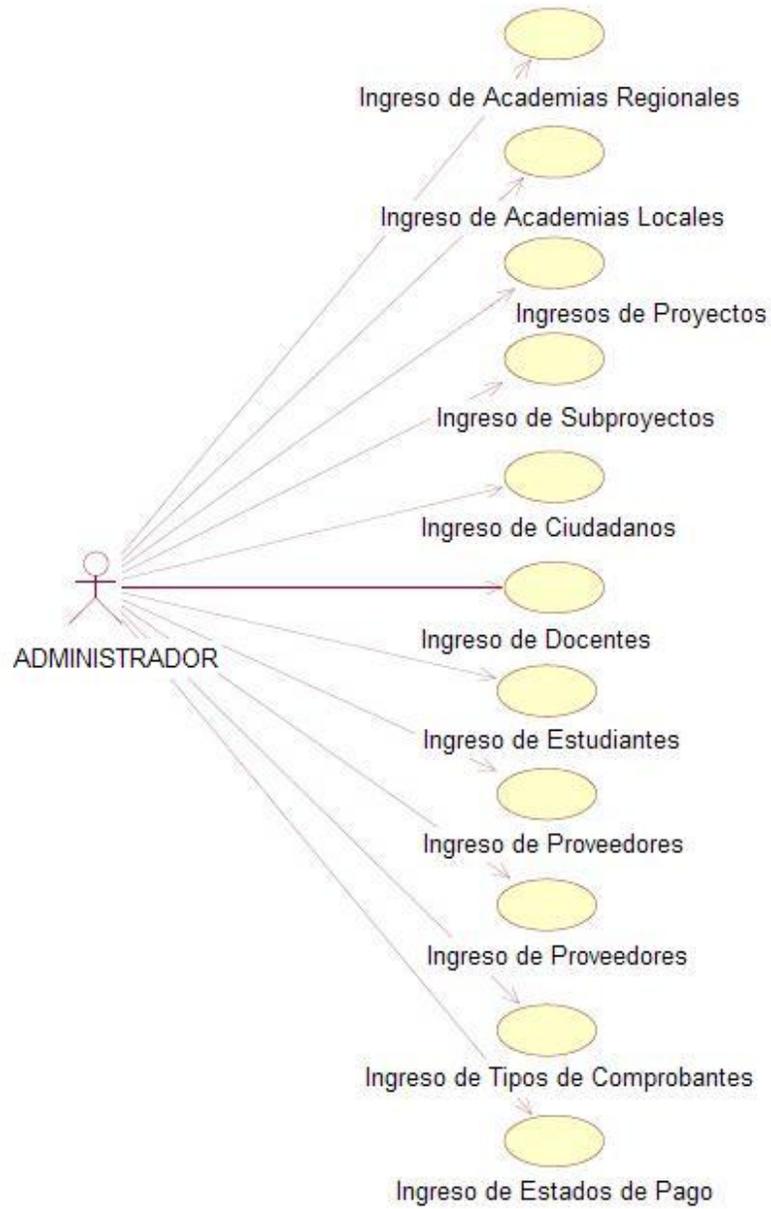


Figura 3.11. Diagrama de Caso de Uso Administrador

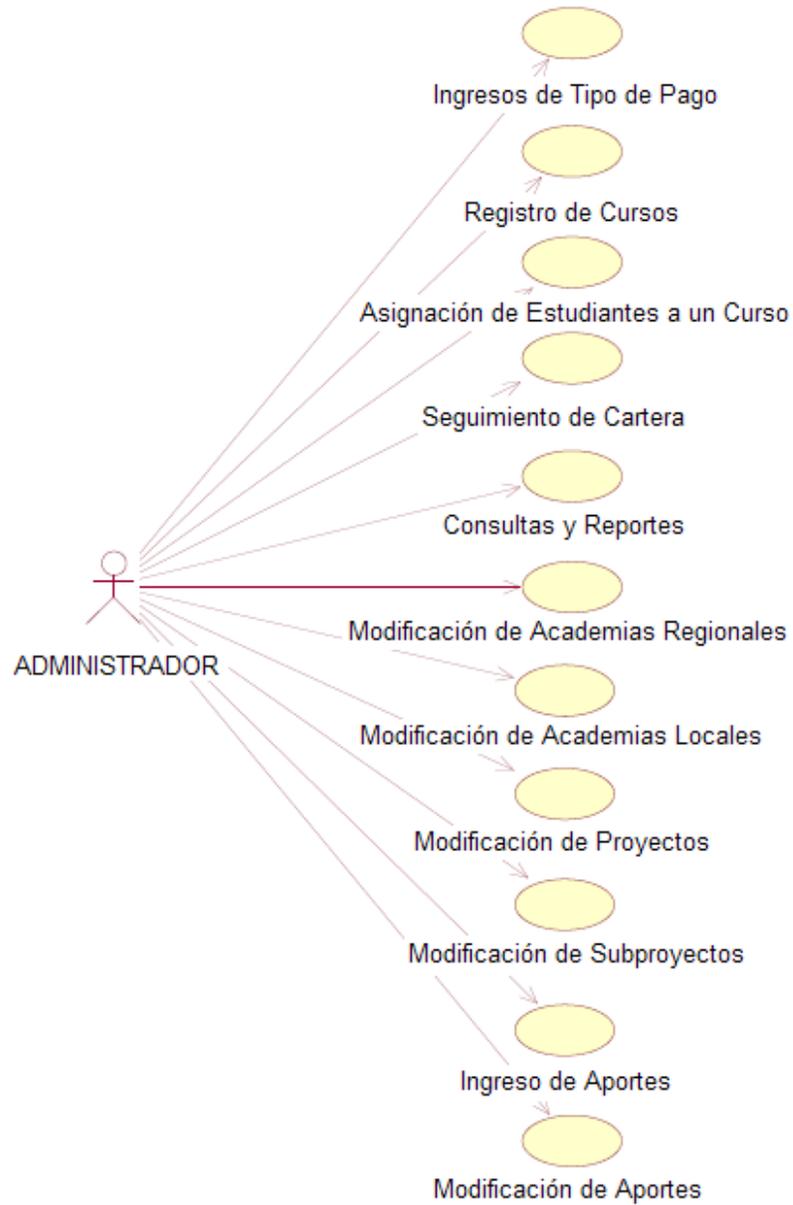


Figura 3.12. Diagrama de Caso de Uso Administrador

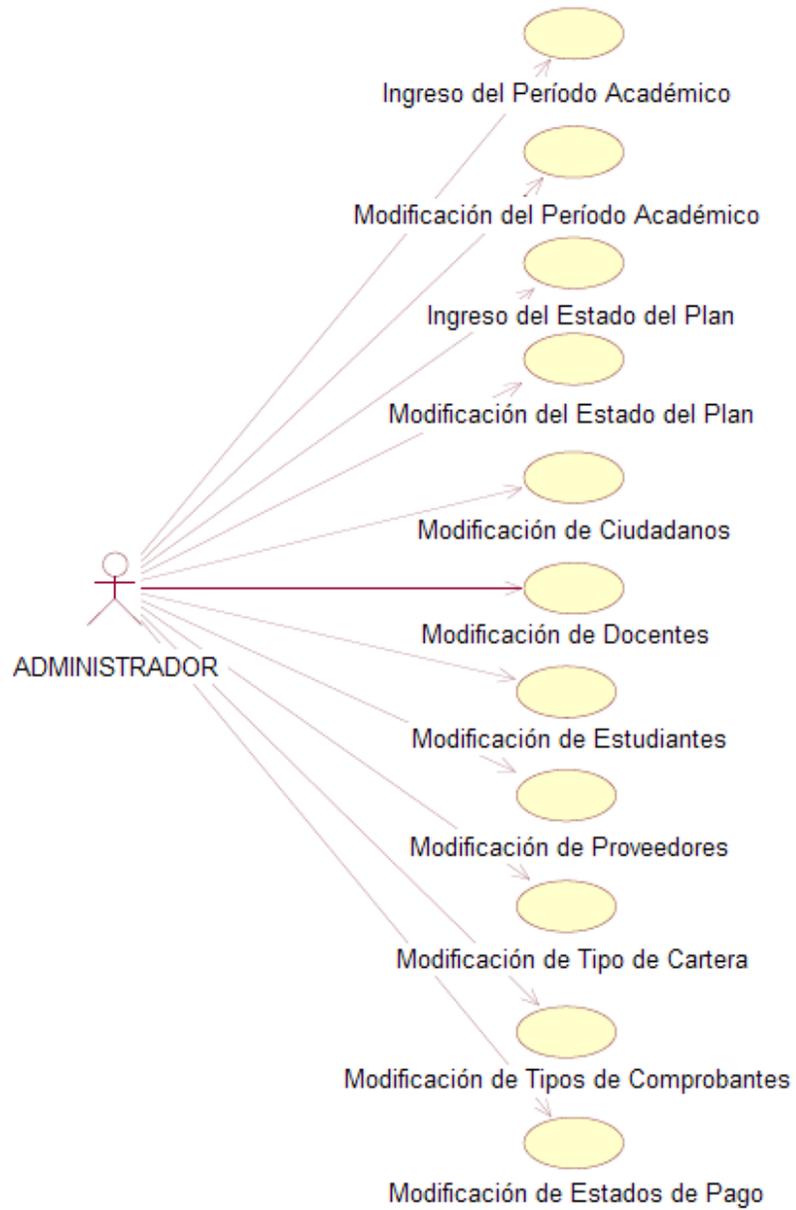


Figura 3.13. Diagrama de Caso de Uso Administrador

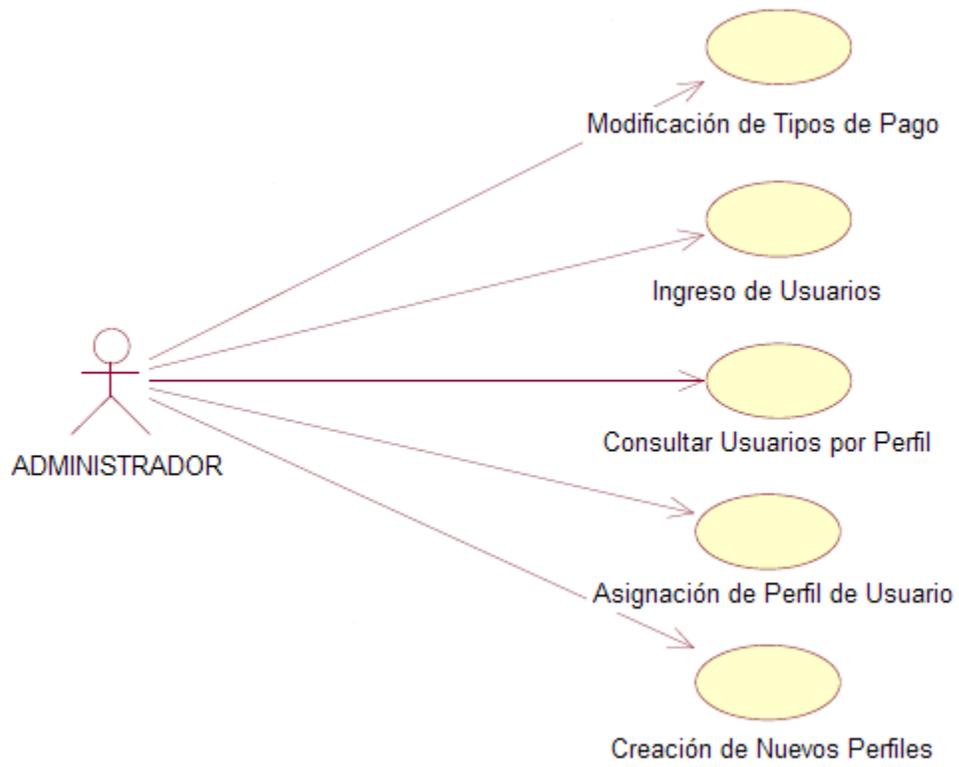


Figura 3.14. Diagrama de Caso de Uso Administrador

3.5.5.3. Detalles de Caso de Uso

Ingreso y seguimiento de la Información para el proceso de Matriculación de los Alumnos al curso de CISCO en la Escuela Politécnica del Ejercito.

Caso de Uso 01: Administración de Usuarios

Descripción: Se asigna el perfil de Administrado a una persona calificada para que tenga acceso total al sistema y pueda crear y asignar perfiles de usuarios dentro del mismo.

Precondición: El administrador del sistema es la persona que tiene acceso total al mismo, por ende debe estar ingresado y autenticado en el mismo previamente.

Flujo Básico:

- ✓ El administrador es la persona autorizada a ingresar nuevos usuarios al sistema, de igual forma es la persona encargada de crear los niveles de acceso de cada uno de ellos.
- ✓ El administrador a través del sistema registra los nuevos usuarios y sus niveles de acceso.
- ✓ El administrador modifica, elimina, y cambia las funciones del usuario en el sistema.
- ✓ El administrador guarda los cambios realizados en el sistema.

- ✓ El administrador termina la cesión del sistema.

Flujo Alternativo:

- ✓ El administrador puede cancelar la operación.
- ✓ El sistema no Guarda los cambios realizados.

Caso de uso 02: Validación de Usuarios

Descripción: El sistema debe de estar en la capacidad de validar nuevos usuarios.

Precondición: El usuario y sus funciones deben ser registrados en el sistema previo su ingreso.

Flujo Básico:

- ✓ El usuario ingresa al sistema.
- ✓ El sistema solicita autenticación al usuario (Usuario y Clave de Ingreso)
- ✓ El usuario ingresa su usuario y clave de acceso.
- ✓ El sistema autentifica al usuario y permite su ingreso al sistema de acuerdo a su nivel de acceso.
- ✓ El usuario realiza las funciones asignadas dentro del sistema.
- ✓ El usuario cierra la sesión.

Flujo Alternativo:

- ✓ El usuario o contraseña son mal ingresados, el sistema no permite el acceso al mismo.
- ✓ Después de tres intentos el sistema se bloquea
- ✓ Se puede cancelar la operación realizada.
- ✓ El sistema no guarda la información.

Caso de uso 03: Ingreso de Academias Regionales

Descripción: El sistema debe de estar en la capacidad de permitir nuevos registros de Academias Regionales.

Precondición: El usuario debe estar registrado en el sistema, el nombre de la Academia Regional debe ser especificado previamente.

Flujo Básico:

- ✓ El usuario ingresa al sistema
- ✓ El usuario registra la nueva Academia Regional
- ✓ El usuario Guarda la información.

Flujo Alternativo:

- ✓ El usuario puede cancelar el ingreso.

- ✓ El sistema no registra la nueva Academia Regional.

Caso de uso 04: Ingreso de Academias Locales

Descripción: El sistema debe de estar en la capacidad de ingresar nuevas Academias Locales.

Precondición: El usuario debe estar registrado en el sistema, el nombre de la Academia Local debe ser especificado previamente.

Flujo Básico:

- ✓ El usuario ingresa al sistema
- ✓ El usuario registra la nueva Academia Local
- ✓ El usuario Guarda la información.

Flujo Alternativo:

- ✓ El usuario puede cancelar el ingreso.
- ✓ El sistema no registra la nueva Academia Local.

Caso de uso 05: Ingreso de Proyectos

Descripción: El sistema debe estar en la capacidad de ingresar nuevos proyectos y registrarlos en el mismo.

Precondición: El usuario debe estar registrado en el sistema, el nombre del Proyecto debe ser especificado previamente.

Flujo Básico:

- ✓ El usuario ingresa al sistema
- ✓ El usuario registra el nuevo Proyecto
- ✓ El usuario Guarda la información.

Flujo Alternativo:

- ✓ El usuario puede cancelar el ingreso.
- ✓ El sistema no registra el Nuevo Proyecto.

Caso de uso 06: Ingreso de Ciudadanos

Descripción: El sistema debe estar en la capacidad de ingresar nuevos ciudadanos y registrarlos en el mismo, además para este caso debe permitir la migración de bases de datos alternativas de sistemas preestablecidos en la ESPE.

Precondición: El usuario debe estar registrado en el sistema, y conocer los datos de los ciudadanos a ser ingresados en el sistema

Flujo Básico:

- ✓ El usuario ingresa al sistema
- ✓ El usuario registra el nuevo Ciudadano
- ✓ El usuario Guarda la información.
- ✓ Cierra la sesión del sistema

Flujo Alternativo:

- ✓ El usuario puede cancelar el ingreso.
- ✓ El sistema no registra el Nuevo Ciudadano.

Caso de uso 07: Ingreso de Docentes

Descripción: El sistema debe estar en la capacidad de ingresar nuevos Docentes tomando de la base de datos de los ciudadanos y registrarlos en el mismo.

Precondición: El usuario debe estar registrado en el sistema, y conocer los datos del Nuevo Docente a ser ingresado.

Flujo Básico:

- ✓ El usuario ingresa al sistema
- ✓ El usuario registra el nuevo Docente.

- ✓ El usuario Guarda la información.

Flujo Alternativo:

- ✓ El usuario puede cancelar el ingreso.
- ✓ El sistema no registra el Nuevo Docente.

Caso de uso 08: Ingreso de Estudiantes

Descripción: El sistema debe estar en la capacidad de ingresar nuevos Estudiantes tomando de la base de datos de Ciudadanos y registrarlos en el mismo.

Precondición: El usuario debe estar registrado en el sistema, y conocer los datos del Estudiante a ser ingresados.

Flujo Básico:

- ✓ El usuario ingresa al sistema
- ✓ El usuario registra el nuevo Estudiante
- ✓ El usuario Guarda la información.

Flujo Alternativo:

- ✓ El usuario puede cancelar el ingreso.
- ✓ El sistema no registra el Nuevo Estudiante.

Caso de uso 09: Ingreso de Proveedores

Descripción: El sistema debe estar en la capacidad de ingresar nuevos Proveedores y registrarlos en el mismo.

Precondición: El usuario debe estar registrado en el sistema, y conocer los datos del Proveedor a ser ingresados.

Flujo Básico:

- ✓ El usuario ingresa al sistema
- ✓ El usuario registra el nuevo Proveedor
- ✓ El usuario Guarda la información.

Flujo Alternativo:

- ✓ El usuario puede cancelar el ingreso.
- ✓ El sistema no registra el Nuevo Proveedor.

Caso de uso 10: Registro de Cursos

Descripción: El usuario y el sistema debe estar en la capacidad de registrar nuevos cursos a dictarse en el periodo indicado.

Precondición: Los datos de las Academias Regionales, locales, proyectos, subproyectos, alumnos, profesores, y cursos deben ser previamente ingresados.

Flujo Básico:

- ✓ El usuario ingresa al sistema
- ✓ El usuario Verifica que la información requerida haya sido ingresada con anterioridad.
- ✓ El usuario registra el nuevo Curso
- ✓ El usuario Guarda la información.

Flujo Alternativo:

- ✓ El usuario puede cancelar el ingreso.
- ✓ El sistema no registra el Nuevo Curso.

Caso de uso 11: Asignación de Estudiantes al Curso

Descripción: El sistema debe de estar en la capacidad de asignar los Estudiantes a un curso creado y registrarlos en el mismo.

Precondición: El usuario debe estar registrado en el sistema, y haber creado el curso para asignar los estudiantes al mismo.

Flujo Básico:

- ✓ El usuario ingresa al sistema
- ✓ El usuario registra los nuevos Estudiantes en el curso creado.
- ✓ El usuario Guarda la información.

Flujo Alternativo:

- ✓ El usuario puede cancelar el ingreso.
- ✓ El sistema no registra los Nuevos Estudiantes al curso.

Caso de uso 12: Consultas y Reportes

Descripción: El sistema debe de estar en la capacidad de generar reportes al usuario.

Precondición: El usuario debe estar registrado en el sistema, y estar conciente de ingreso previo de la información al sistema para generar los reportes necesario o requeridos.

Flujo Básico:

- ✓ El usuario ingresa al sistema
- ✓ El usuario genera el reporte requerido
- ✓ El usuario imprime en pantalla o papel el informa requerido.

Flujo Alternativo:

- ✓ El usuario puede cancelar el informe.

3.5.5.4. Diagramas de Caso de Uso

3.5.5.4.1. Administración de Usuarios

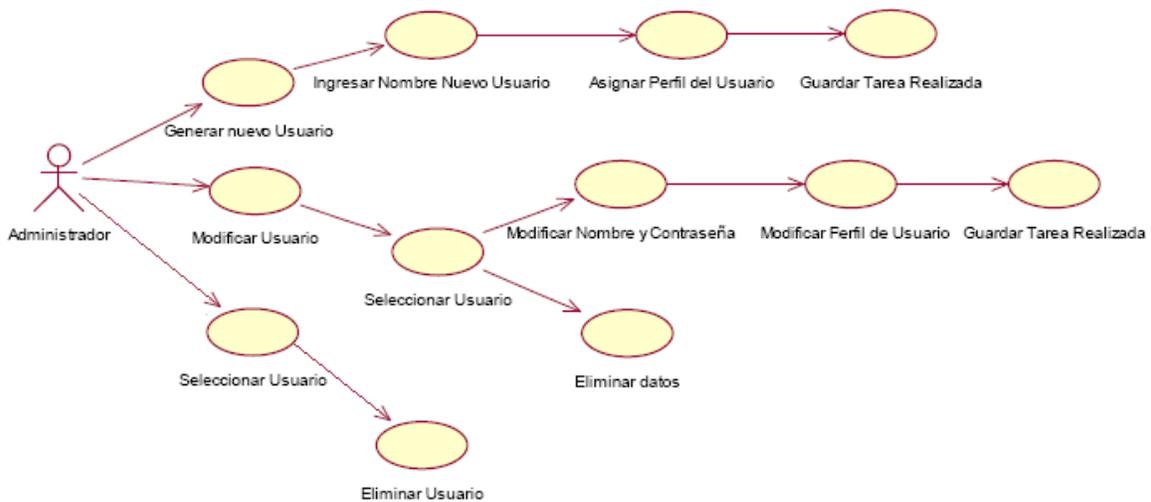


Figura 3.15. Diagrama de Administración de Usuarios

3.5.5.4.2. Validación de Usuarios

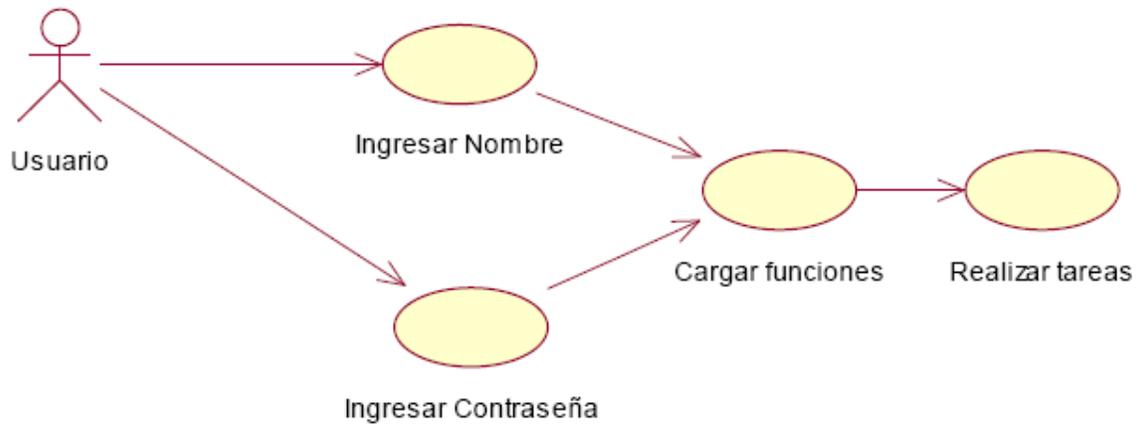


Figura 3.16. Diagrama de Validación de Usuarios

3.5.5.4.3. Ingreso Academia Regional

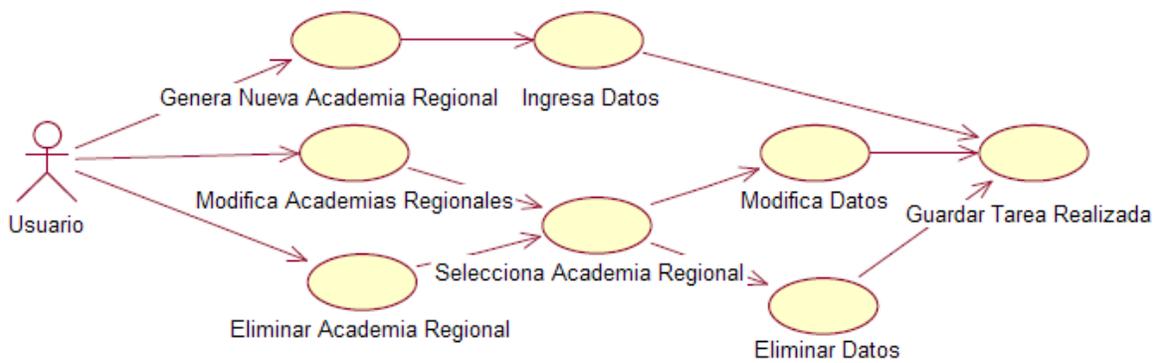


Figura 3.17. Diagrama de Ingreso de Academia Regional

3.5.5.4.4. Ingreso Academia Local

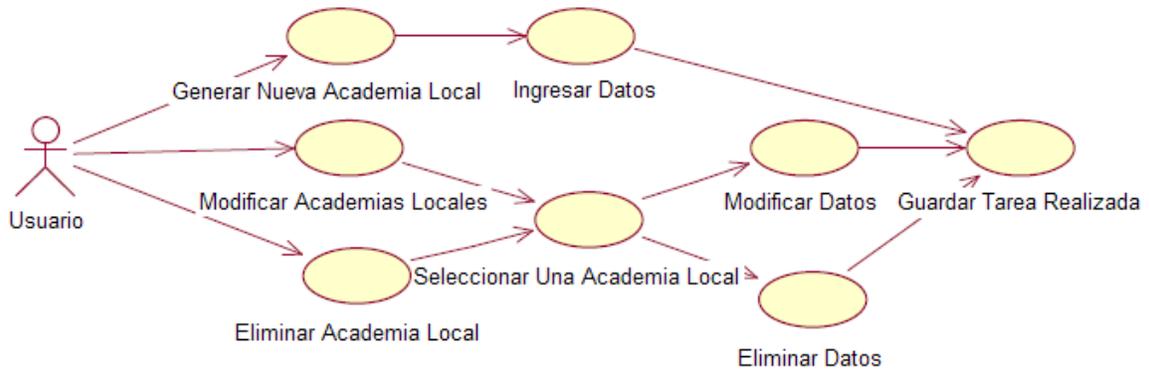


Figura 3.18. Diagrama de Ingreso de Academia Local

3.5.5.4.5. Ingreso de Proyectos

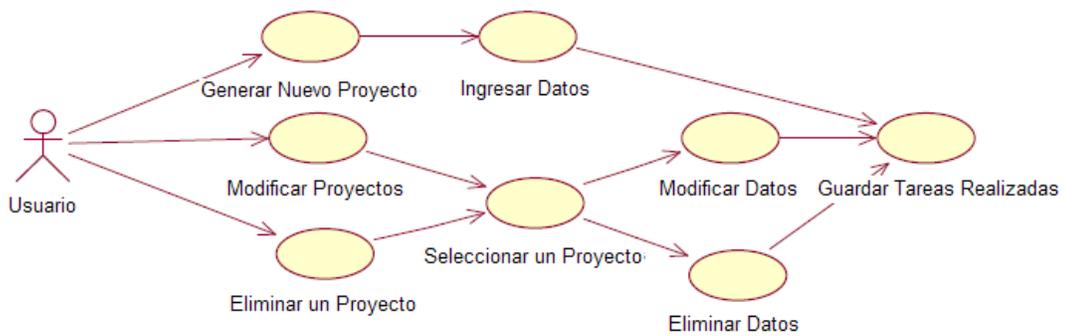


Figura 3.19. Diagrama de Ingreso de Proyectos

3.5.5.4.6. Ingreso de Ciudadanos

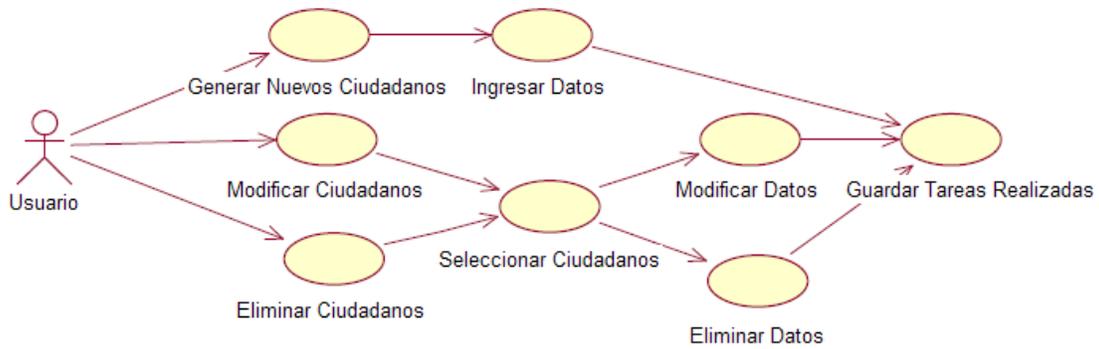


Figura 3.20. Diagrama de Ingreso de Ciudadanos

3.5.5.4.7. Ingreso de Docentes

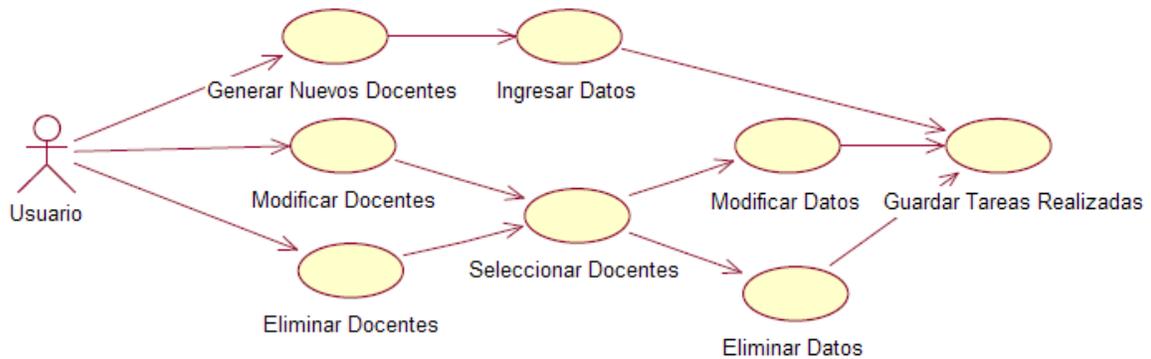


Figura 3.21. Diagrama de Ingreso de Docentes

3.5.5.4.8. Ingreso de Estudiantes

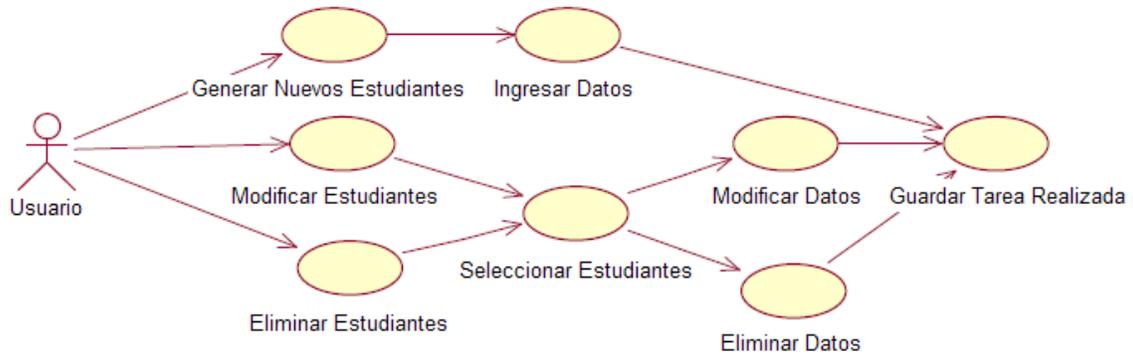


Figura 3.22. Diagrama de Ingreso de Estudiantes

3.5.5.4.9. Ingreso de Proveedores

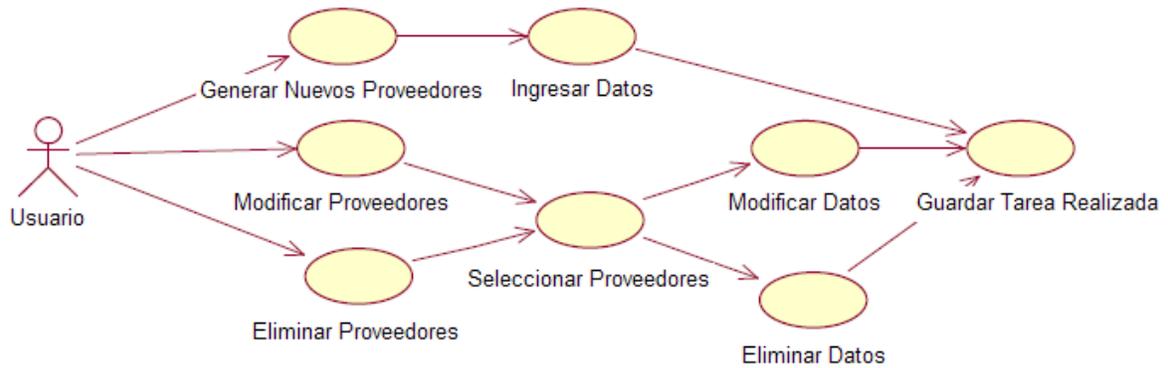


Figura 3.23. Diagrama de Ingreso de Estudiantes

3.5.5.4.10. Registro de Cursos

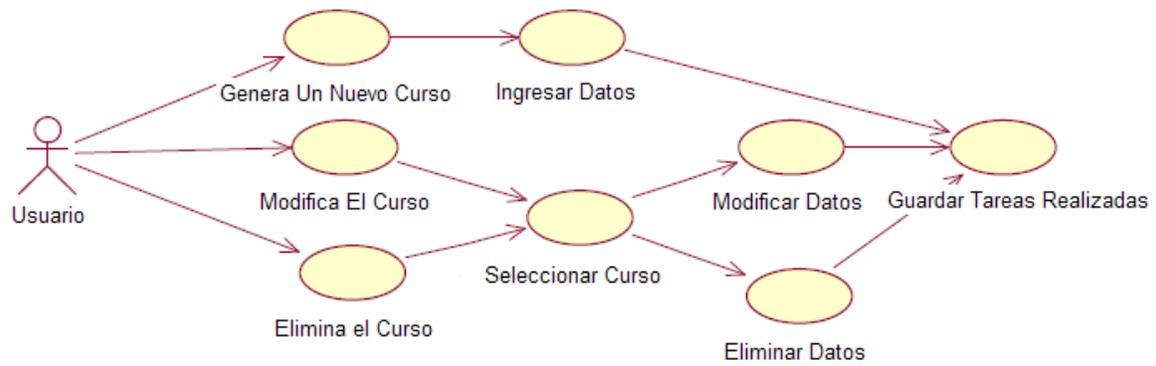


Figura 3.24. Diagrama de Ingreso de Estudiantes

3.5.6 Diccionario Datos Procesos (Anexo F)

3.5.7 Modelo de datos del nivel Lógico (Anexo G)

3.5.8 Diccionario Datos Modelo Lógico (Anexo H)

3.5.9 Modelo de datos del nivel Físico (Anexo I)

3.5.10 Diccionario de Datos del Modelo Físico (Anexo J)

3.5.11 Script de creación de la base de datos (Anexo K)

3.5.12 Base de Datos (Anexo L)

3.5.13 Manual Técnico (Anexo M)

3.6. IMPLEMENTACIÓN

CREANDO UNA APLICACION EN 3 CAPAS

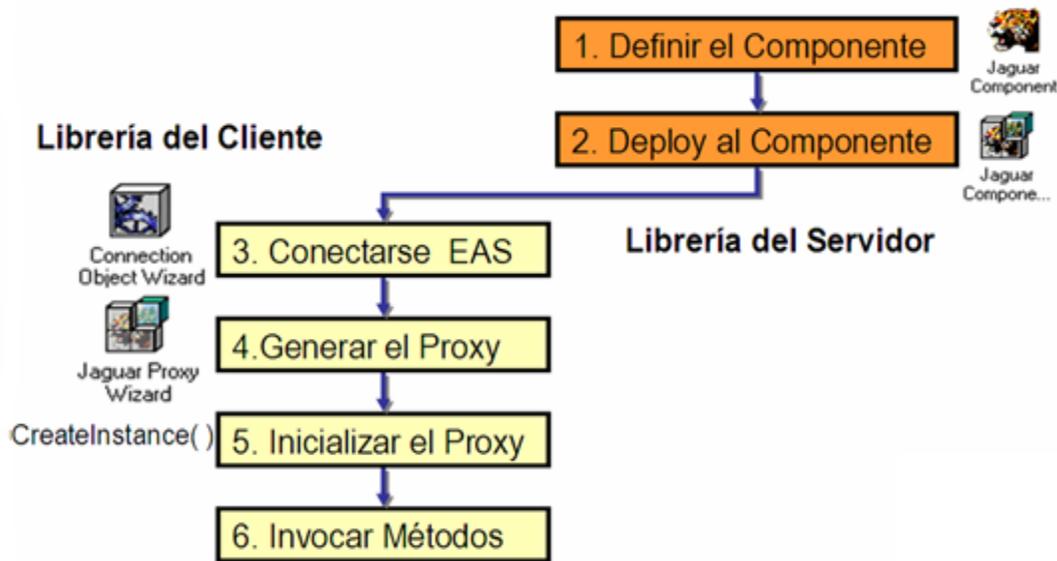


Figura 3.25. Creando una aplicación en 3 capas

PARA CREAR UN COMPONENTE

Abrir Inicio / Programas / Sybase / EAServer / Jaguar Manager.

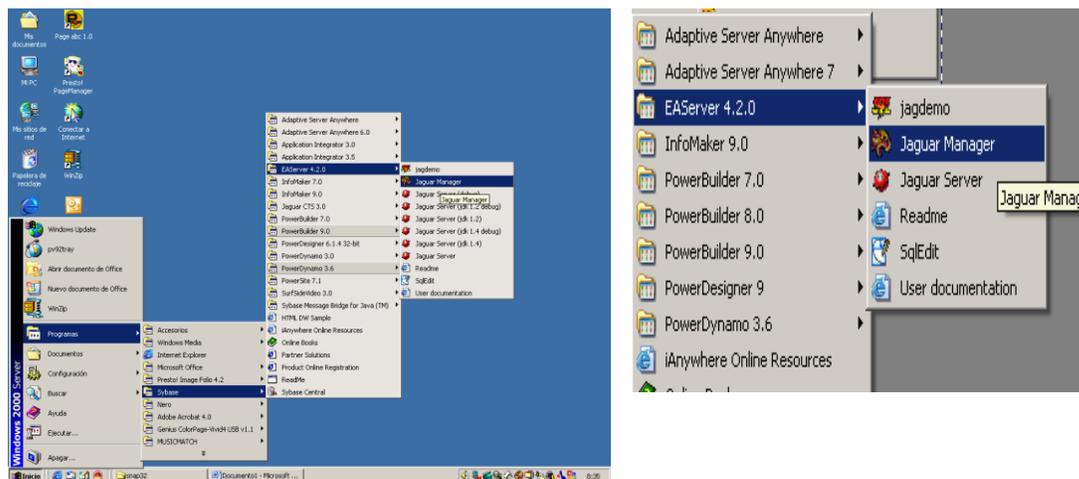


Figura 3.26. Ingresando a Jaguar Manager

El siguiente gráfico muestra la pantalla principal del Jaguar Manager:

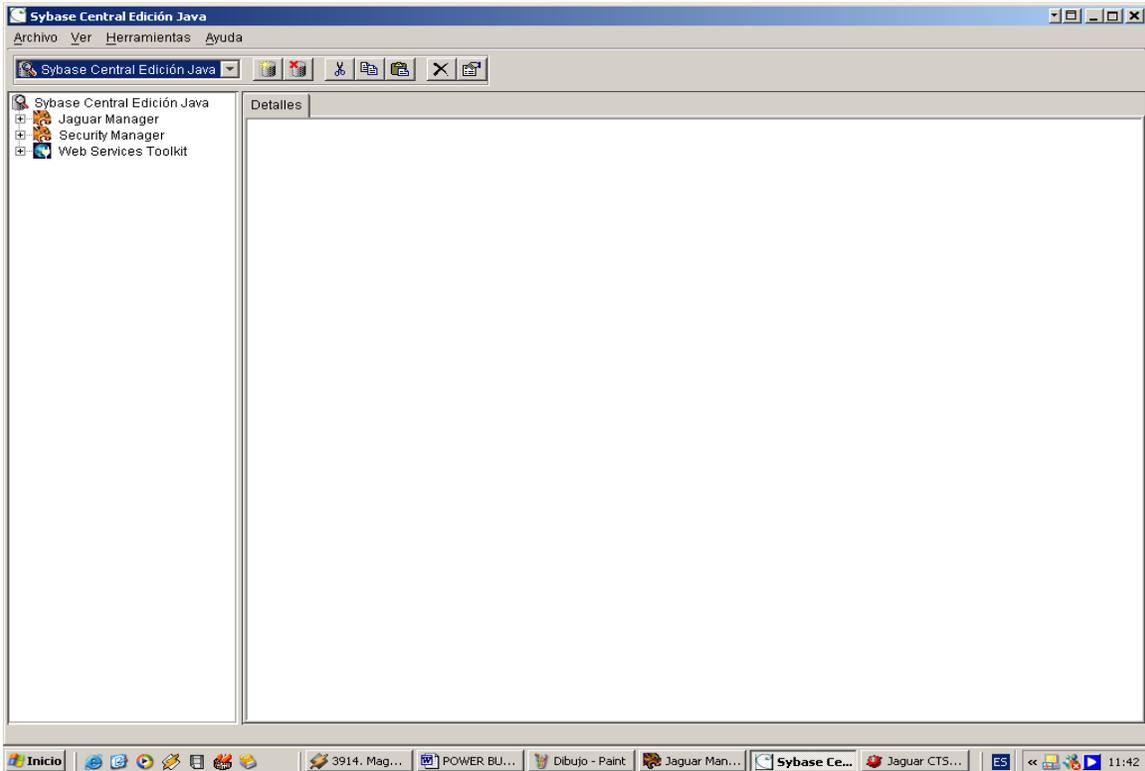


Figura 3.27. Pantalla Principal Jaguar Manager

Se procede a dar un click derecho sobre Jaguar Manager y elegir la opción de Conectar o si no la otra forma de conectarse es dando un click en el botón que se asimila a un tambor como se muestra en el gráfico 3.28.

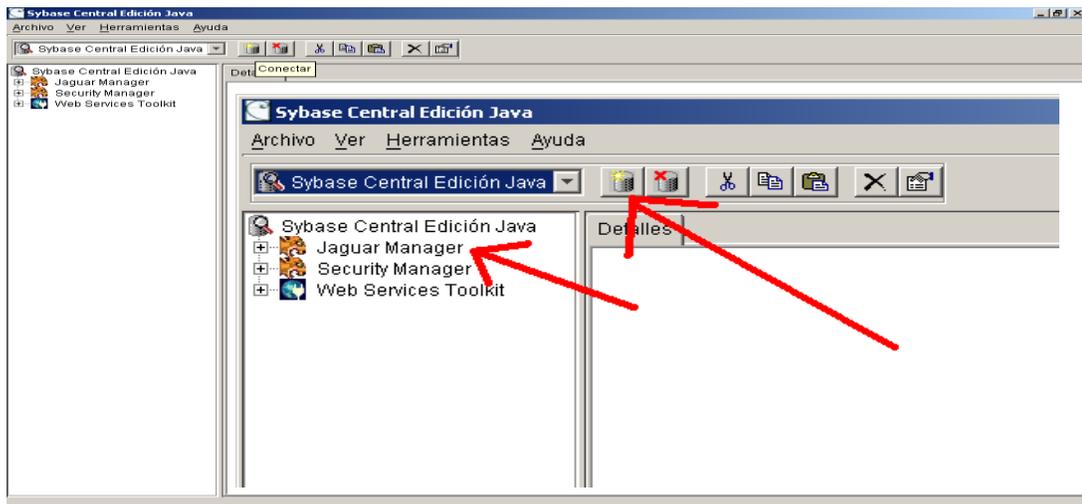


Figura 3.28. Conectándose al Jaguar Manager

Automáticamente aparecerá un listbox o dropdown (lista desplegable) de la que se procede a escoger la opción Jaguar Manager como se muestra en el gráfico 3.29.

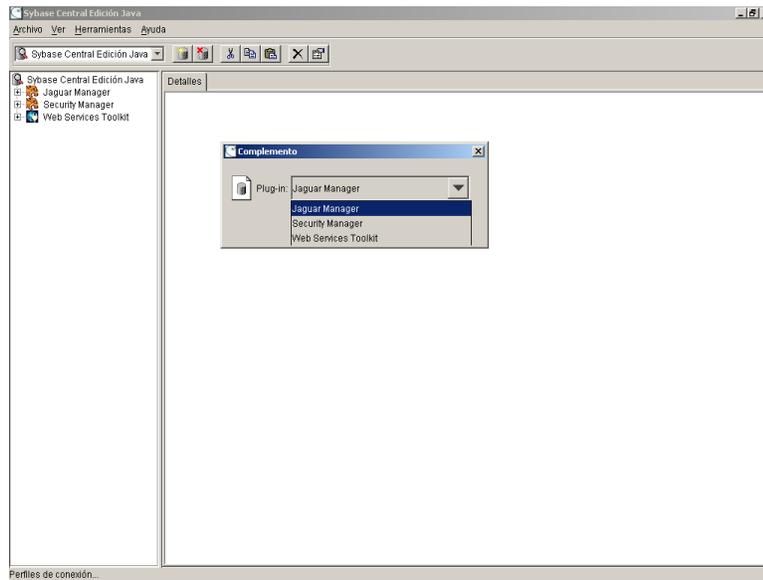


Figura 3.29. Lista Desplegable Jaguar Manager

Se procede a dar un click sobre el botón OK.

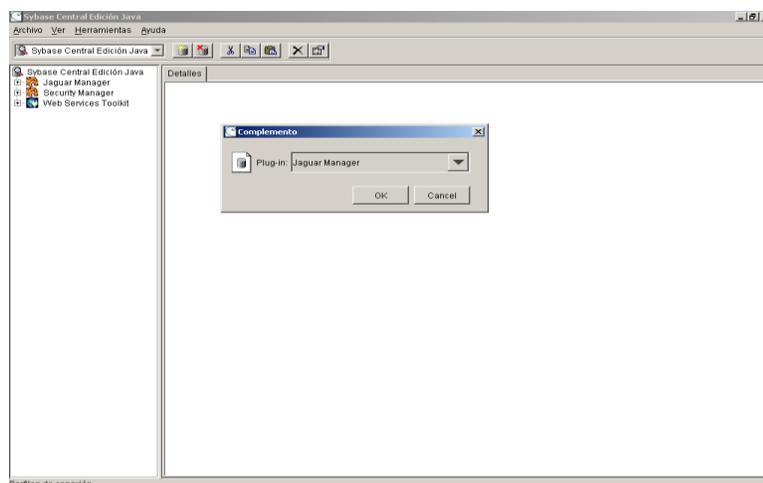


Figura 3.30. Conectándose al Jaguar Manager

Acto seguido se debe llenar los valores que se necesitan como se detalla a continuación para poder establecer la conexión con el Jaguar Manager

- ✓ **Profile :** **NONE**
- ✓ **User Name:** **jagadmin**
- ✓ **Password:** dejar en blanco.
- ✓ **HostName:** Nombre del equipo que servirá como servidor de componentes o la dirección IP del mismo o la palabra localhost cuando el servidor de componentes estará en este equipo.

- ✓ **Port Number:** **9000**

Una vez ingresados dichos valores se procederá a dar click sobre el botón Connect, cabe recalcar que debe estar el servidor levantado, es decir que el jaguar Server debe estar corriendo, como se ve en el gráfico 3.31:

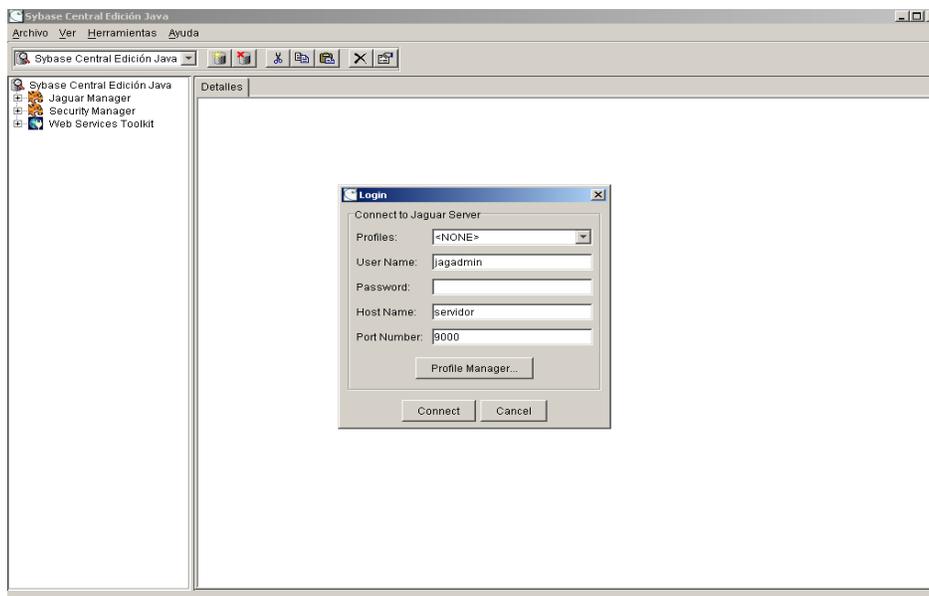


Figura 3.31. Conectándose a la Base de Datos

Una vez conectado a la base de datos se ingresa al área administrativa del Jaguar Manager, como se ilustra en el Gráfico 3.32:

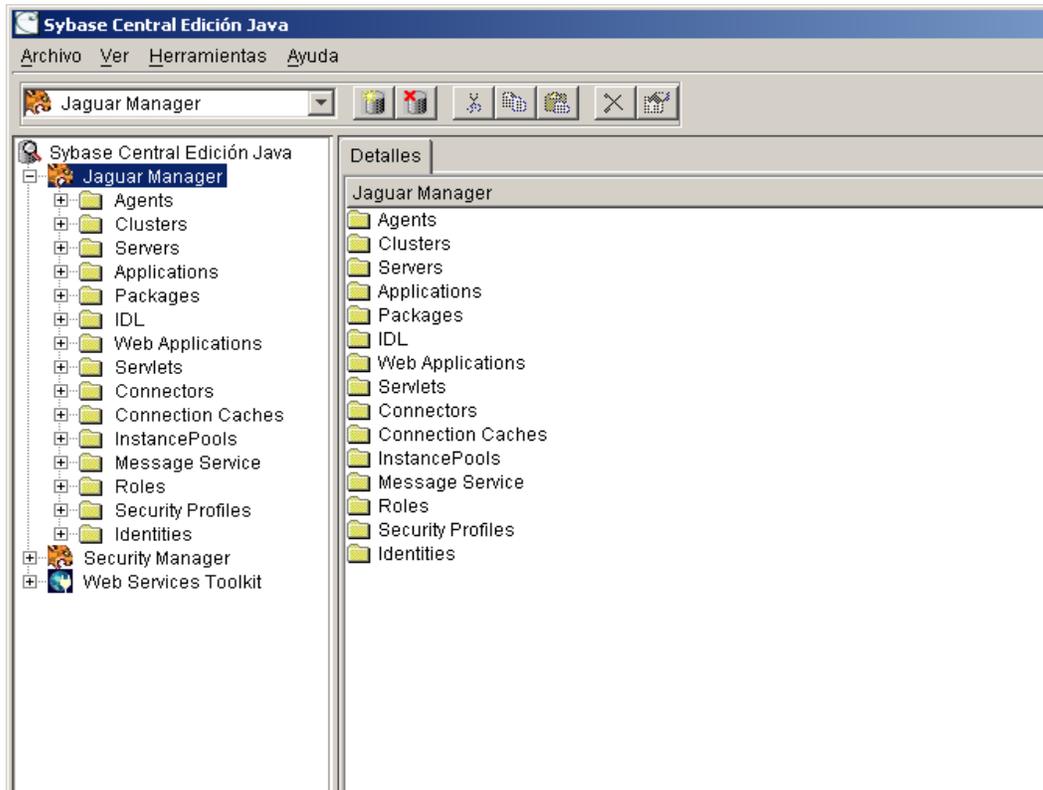


Figura 3.32. Área Administrativa Jaguar Manager

Para crear el modulo del componente se debe dar clic derecho en IDL, y se escoge New IDL Module, como se ve en el gráfico 3.33:

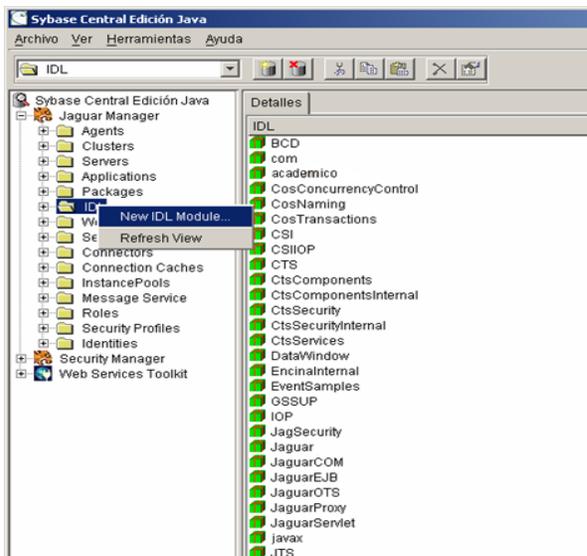


Figura 3.33. Creando un nuevo IDL

Luego de dar un clic en New IDL Module se presenta la siguiente ventana donde se debe ingresar el nombre del módulo, en este caso se llamará “académico”, como se ilustra en el gráfico 3.34:

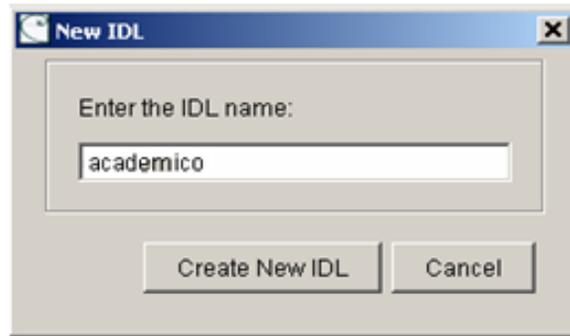


Figura 3.34. Ingresando el nombre al IDL

De esta forma ya se cuenta con el módulo que se ingreso anteriormente, y se encuentra en el área administrativa del Jaguar Manager, como se ilustra el gráfico 3.35:

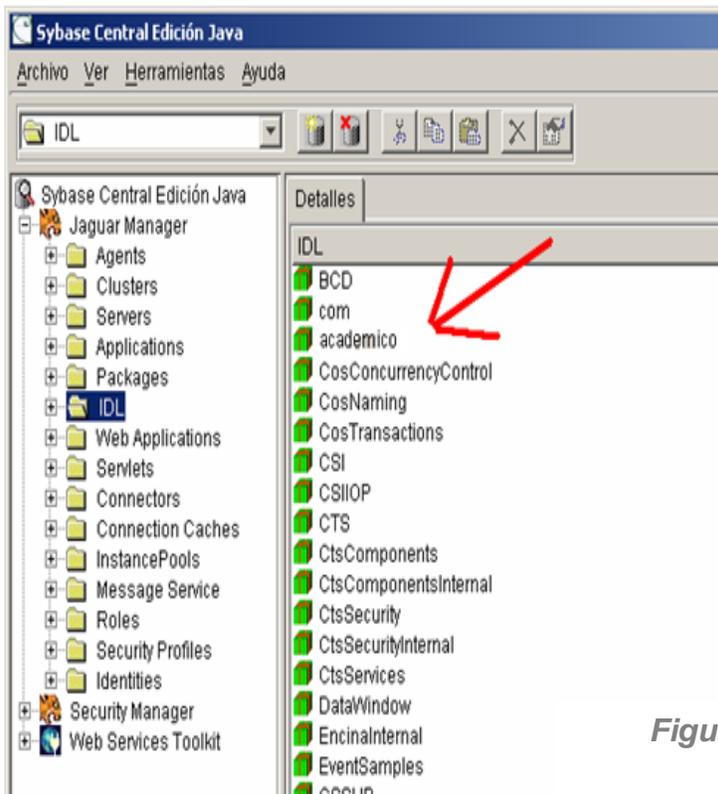


Figura 3.35. Nuevo IDL Creado

ASIGNACION DE LA CLASE AL MODULO CREADO EN POWER BUILDER

En este caso se tiene previamente creada la clase que se va a manejar todas las funciones del proyecto, se llama “n_cst_academico”, así:

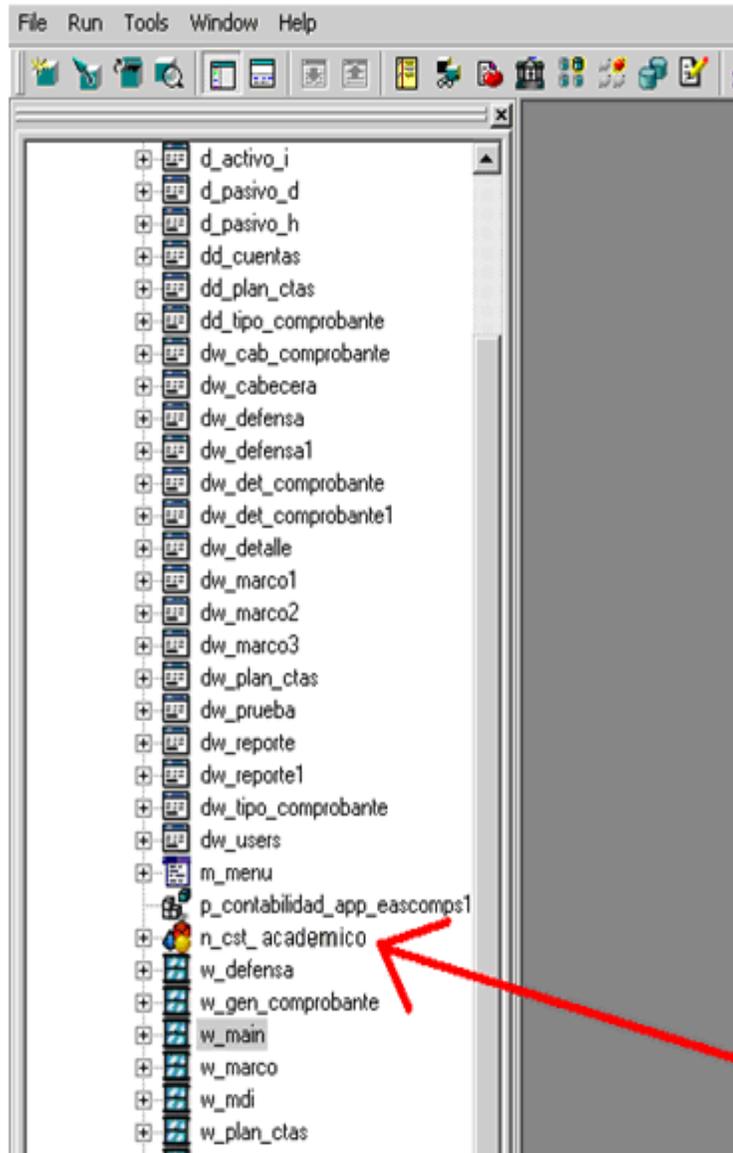


Figura 3.36. Asignando una Clase

CREACION DEL EASERVER PROFILE DE JAGUAR

Para la creación de un componente es necesario contar con un EA Server Profile para lo cual desde la aplicación realizada en Power Builder se dará un click en el icono del EA Server Profile como se muestra en la figura.

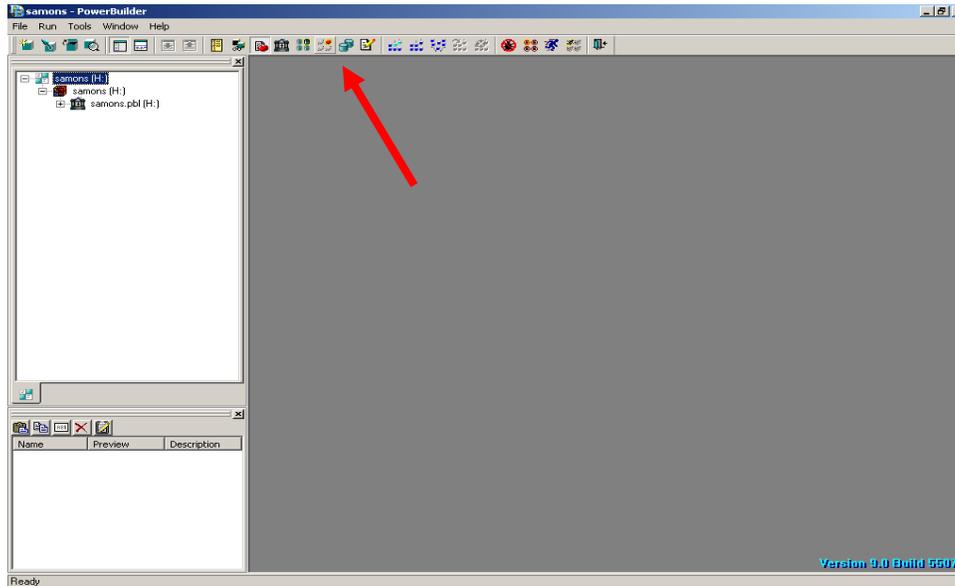
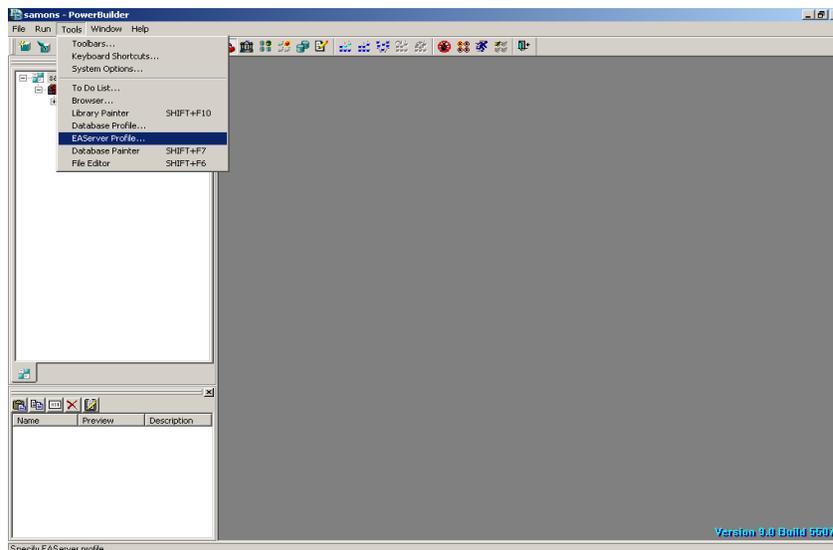


Figura 3.37 Creación de un Componente

Otra manera de hacer lo mismo es elegir EA Server Profile del menú Tools como se muestra en la siguiente figura.



*Figura 3.38.
Creación de un Componente*

Creado el componente se obtiene la administración de los EaServers Profiles, como se ilustra en el gráfico 3.39, tome en consideración que el profile de jaguar permite establecer las características al momento de conectarse con el servidor de componentes.

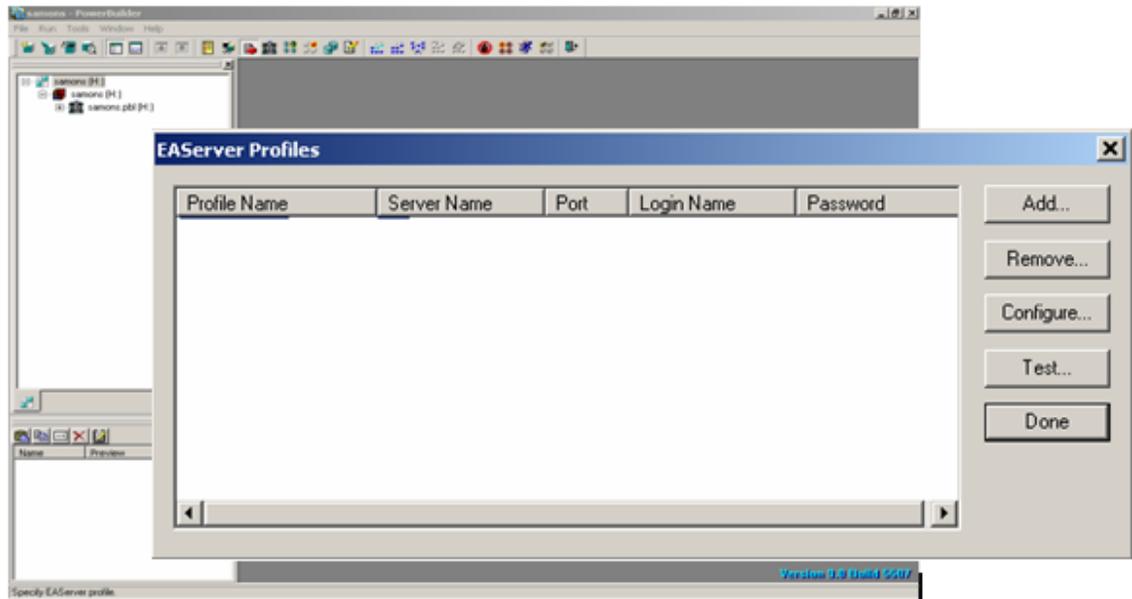
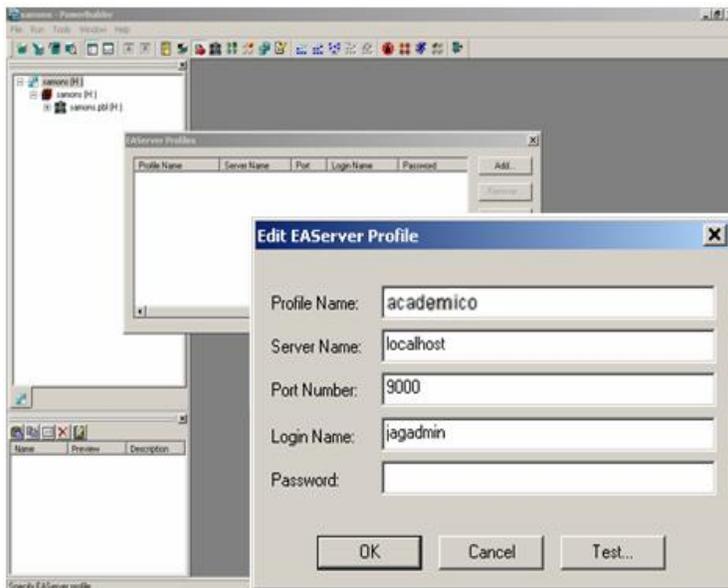


Figura 3.39 Easerver Profile

A continuación se deberá dar un click en el botón Add, como se ilustra en el gráfico 3.40.



*Figura 3.40.
Añadiendo un Nuevo Profile*

Se proceden a llenar los valores indicados en el gráfico 3.40 como se detalla a continuación para poder establecer la creación del EaServer Profile:

- ✓ **Profile :** El nombre que usted quiera darle al nuevo profile
“académico”
- ✓ **User Name:** jagadmin
- ✓ **Password:** dejar en blanco
- ✓ **HostName:** Nombre del equipo que servirá como servidor de componentes o la dirección IP del mismo o la palabra localhost cuando el servidor de componentes estará en este equipo.
- ✓ **Port Number:** 9000

Se deberá dar un click en el botón Test para asegurarse de que el EaServer Profile ha sido creado de manera correcta verificando que aparezca el siguiente mensaje.

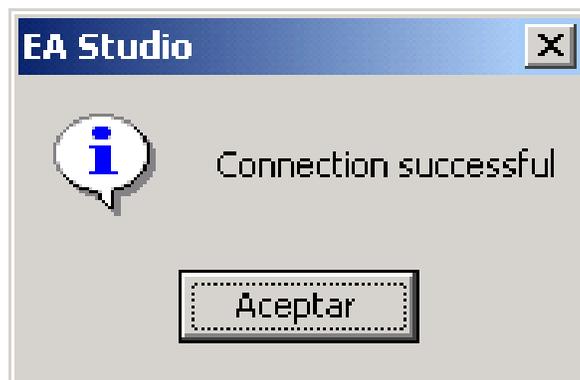


Figura 3.41. Conexión Satisfactoria

Se da un click en Aceptar de la gráfica 3.41 y se tiene los perfiles actuales, y los botones Add, Remove, Configure, Test y Done para salir, como se ilustra en la gráfica 3.42.

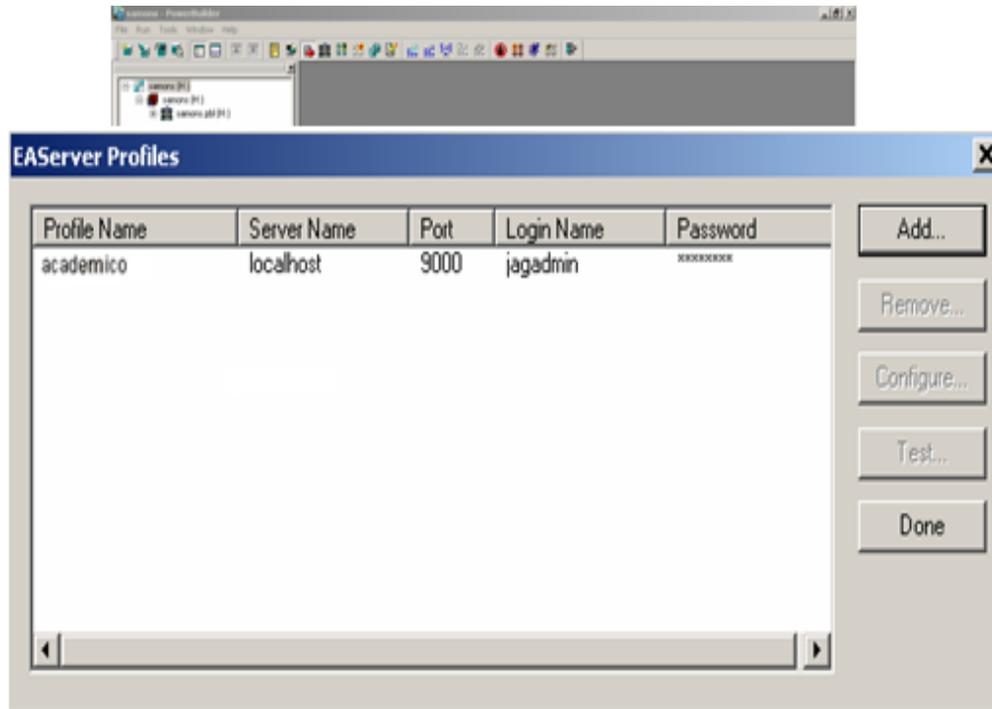


Figura 3.42. EaServer Profile

Una vez creado el profile necesario para el desarrollo de la aplicación, automáticamente en la pantalla de administración del Jaguar Manager aparece una carpeta con el nombre de servidor en donde se encuentra el servidor Jaguar y el componente al cual esta direccionado, como se ilustra en la gráfica 3.43

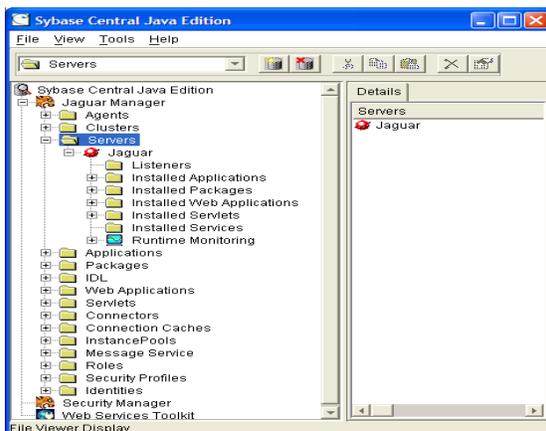


Figura 3.43. Servidor Jaguar

El siguiente paso es crear un paquete, se hace click derecho sobre la carpeta Packages, y se escoge la opción New Packages, con el objetivo de ponerle un nombre. (No es case – sensitive). Como se ilustra en la Gráfica 3.44.

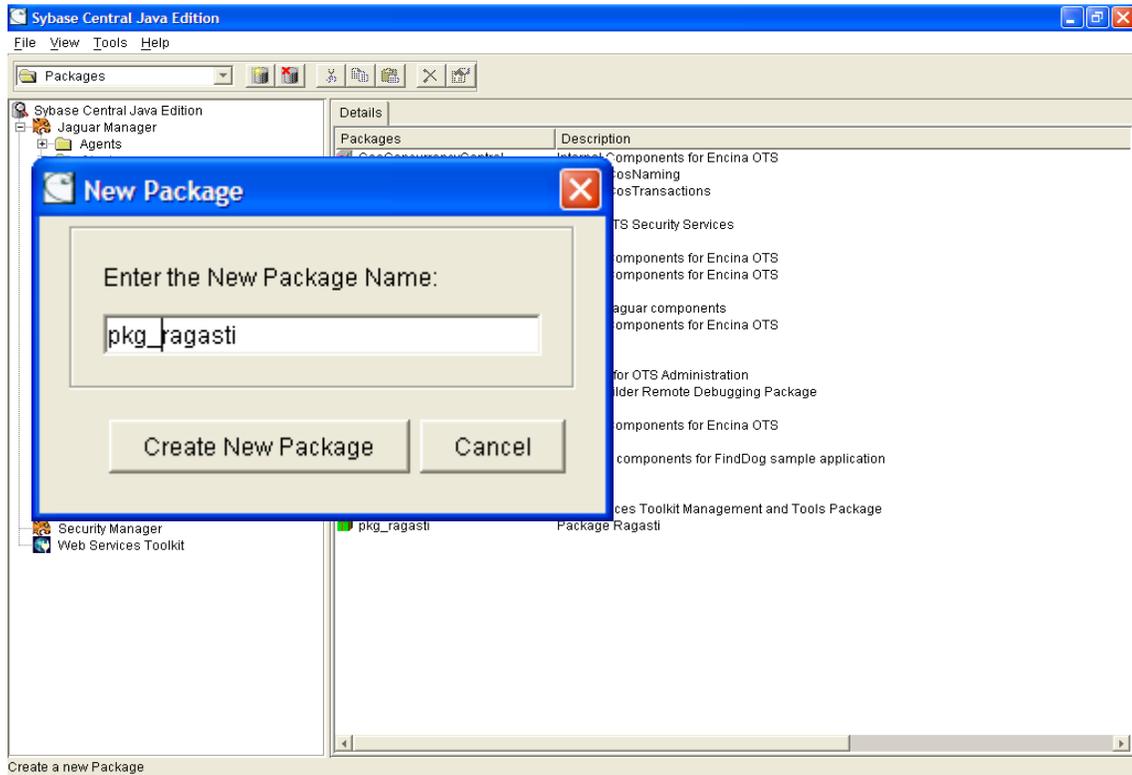


Figura 3.44. New Package

PARA INSTALAR UN PAQUETE EN EL SERVIDOR

En la pantalla de administración del Jaguar Manager dentro de la carpeta Servers se encuentra la carpeta Installed Packages, se da un click derecho sobre esta carpeta y se escoge la opción install package, como se ilustra en la gráfica 3.45.

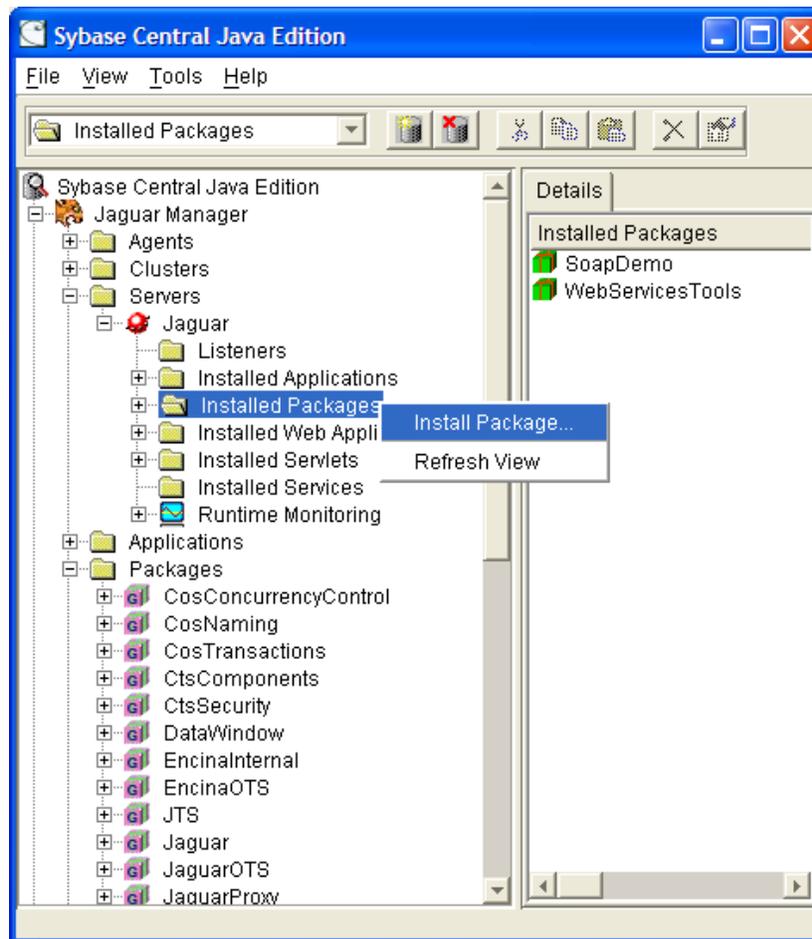


Figura 3.45. Instalando un Nuevo Paquete

Realizado el paso anterior automáticamente aparece una pantalla en donde se encuentran dos opciones, “Install an Existing Package”, que me permite instalar un paquete creado con anterioridad, o “Create and install New Package”, que es otra forma de crear e instalar evitándose un paso a la hora de crear un paquete, esto se lo puede ver en el gráfico 3.46.

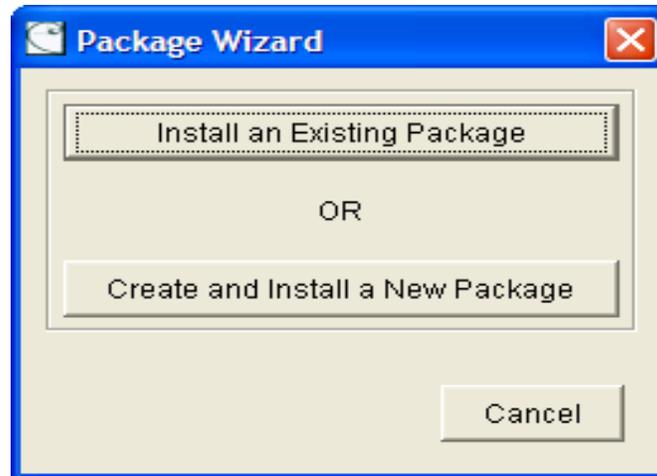
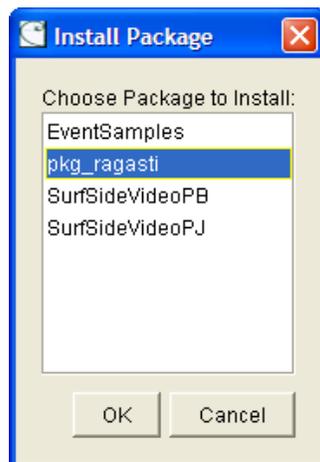


Figura 3.46. Instalando un Paquete

Para este caso se selecciona la opción 1 ya que se creó el paquete con anterioridad, automáticamente aparece una pantalla con todos los paquetes creados, se escoge el creado para nuestra aplicación en este caso **pkg_ragasti**, como se ilustra en la gráfica 3.47



*Figura 3.47.
Escogiendo el Paquete creado*

CREAR COMPONENTES EN EL PAQUETE INSTALADO EN EL SERVIDOR

Una vez instalado el paquete, hacemos click derecho sobre el mismo y escogemos la opción New Component, como se ilustra en la Gráfica 3.48

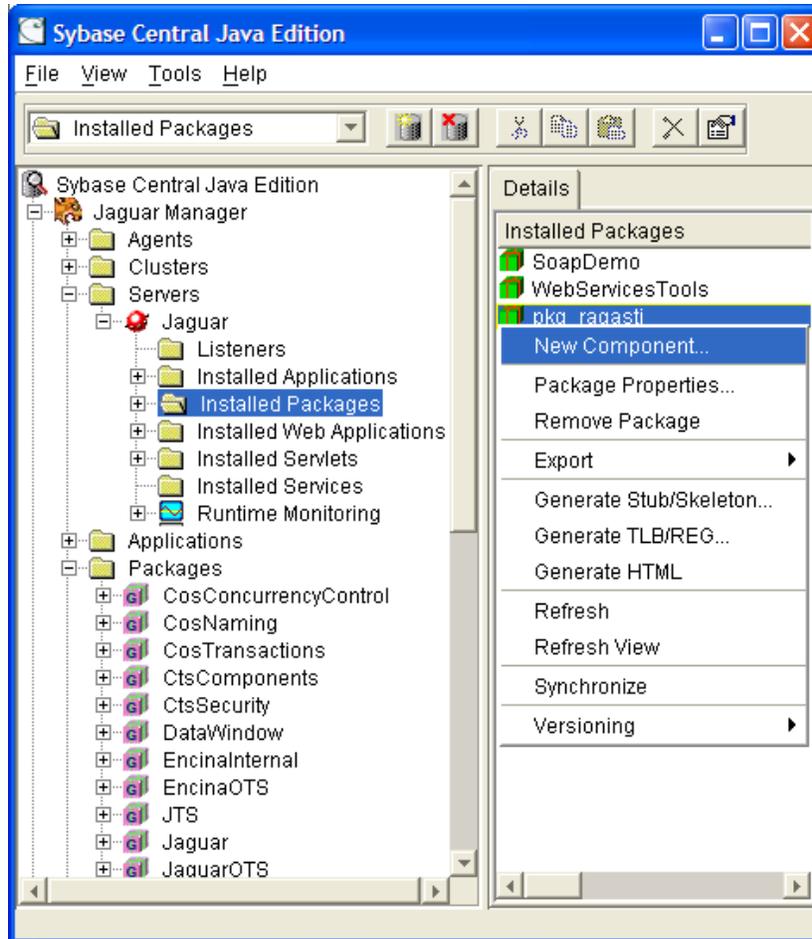
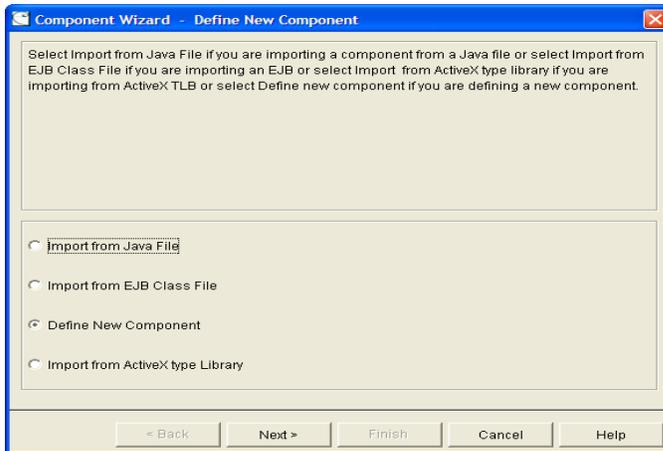


Figura 3.48. Nuevo Componente

A continuación se elige la opción **Define New Component** y se presiona **Next**, como se ilustra en la gráfica 3.49



*Figura 3.49.
Define New Component*

Después de escoger la opción anterior aparece una nueva ventana con la lista de clases, en este caso no se hace ninguna modificación y da click en el botón next, como se ilustra en la gráfica 3.50.

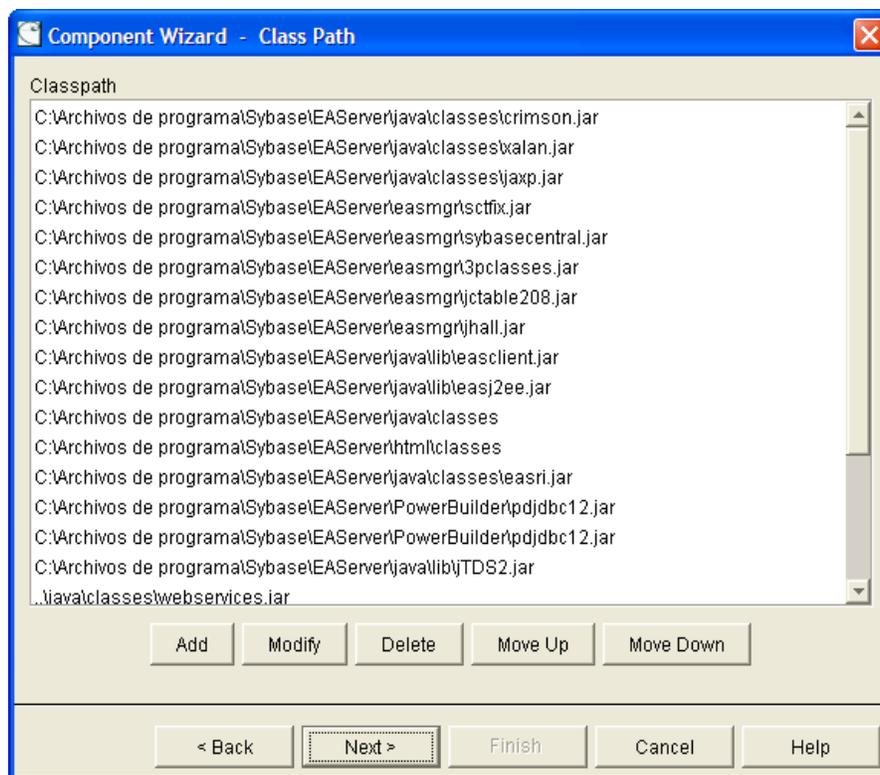
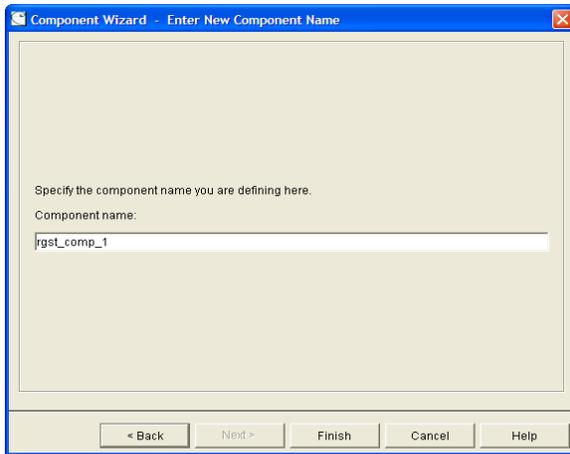


Figura 3.50. Lista de Clases

Como paso final de la configuración del componente es asignarle un nombre, y se da click en el botón finish, como se ilustra en la gráfica 3.51



*Figura 3.51.
Ingresando un nombre al Componente*

Se puede especificar las propiedades del Componente, y al terminar presionar **OK**, como se ilustra en la gráfica 3.52

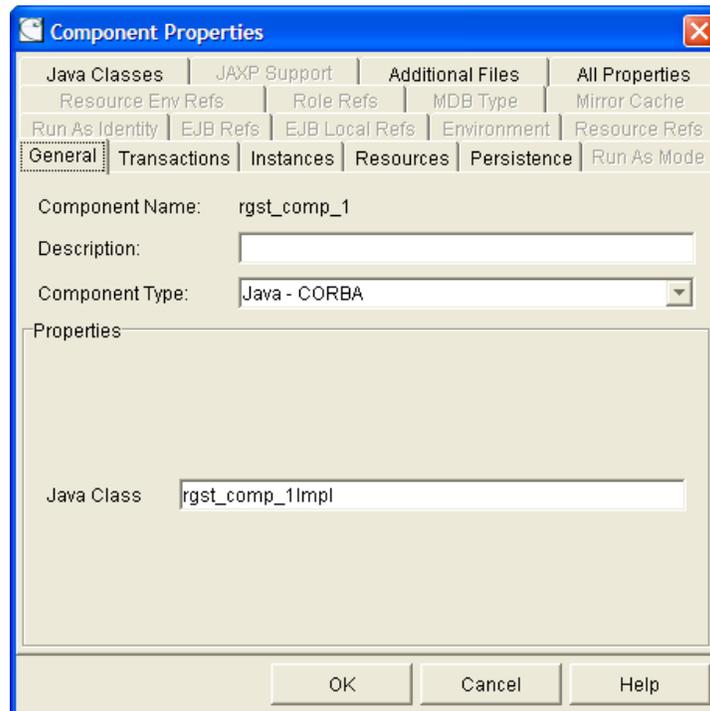


Figura 3.52. Propiedades del Componente

CREANDO UN COMPONENTE EASERVER EN POWER BUILDER

De haberse creado con éxito los profile necesarios según la aplicación con la base de datos, el paso siguiente es crear los componentes que permitirán la transmisión de los datos entre back end y en front end, dicho de otra manera entre la base de datos y la interfase de usuario. Para la creación del componente nos ubicamos en el work space y damos click sobre el icono New, luego en Target y se elige la opción Easerver Component y se presiona Ok, como se ilustra en el gráfico 3.53

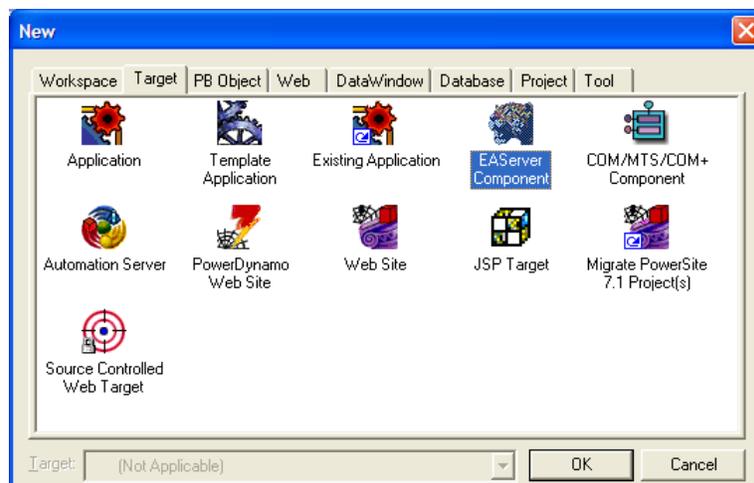


Figura 3.53. Creación de un componente en P. W.

En esta opción se inicia un wizard donde se presenta pantallas que indican que va a crear el wizard y como conectarlas, a las que hay que presionar seguir las instrucciones para la conexión con la base de datos, como se ilustra en el gráfico 3.54 y 3.55.



Figura 3.54.
Creando un componente en P. W.

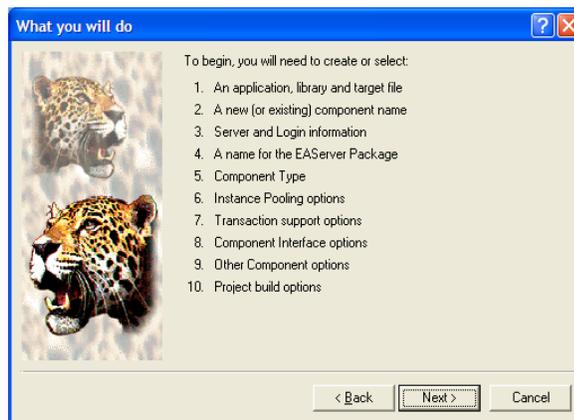


Figura 3.55.
Creando un componente en P. W.

Es necesario asignar un nombre para la aplicación a crear y luego se presionar en **NEXT**, como se ilustra en la gráfica 3.56.

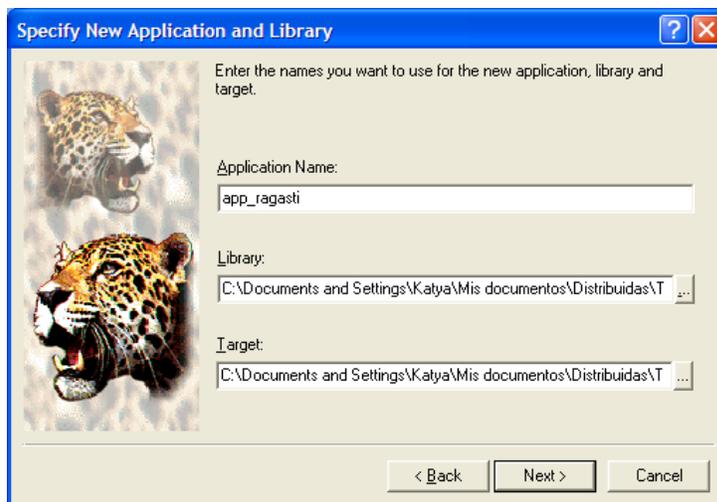


Figura 3.56.
Creando un componente en P. W.

Se puede o no ingresar otras librerías, después presionar NEXT, como se puede observar en el Gráfico 3.57

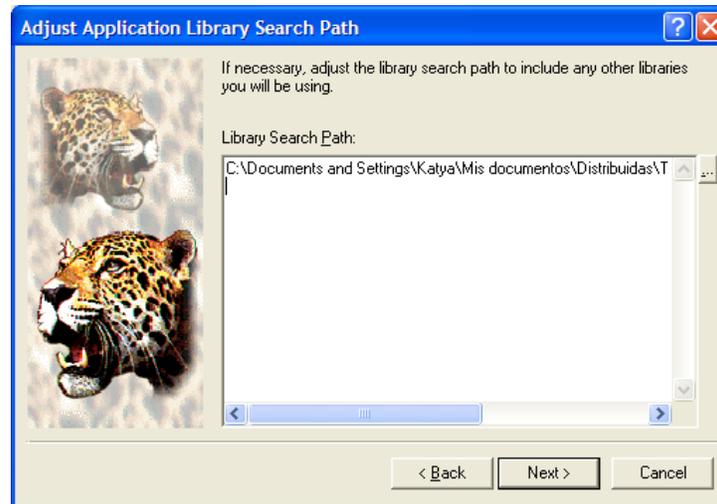


Figura 3.57. Agregando Librerías de P. W.

Se debe crear una nueva interfaz si no se tiene una ya creada y presionar **NEXT**, como se ilustra en el Gráfico 3.58

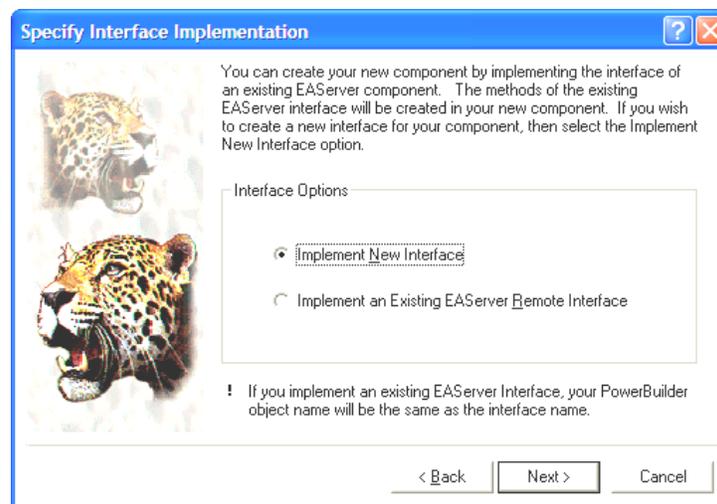


Figura 3.58. Implementando una nueva interfase de P. W.

Se debe especificar el nombre del objeto no visual y dar una descripción del mismo después presionar **NEXT**, como se ilustra en el gráfico 3.59

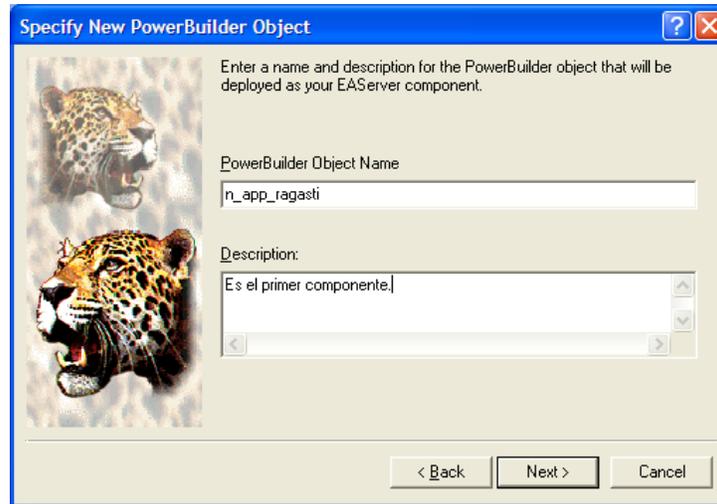


Figura 3.59. Especificando un nombre al objeto de P. W.

A continuación se debe escribir nuevamente el nombre del componente y presionar **NEXT**, como se ve en el Gráfico 3.60.



Figura 3.60. Ingresando el Nombre del Objeto de P. W.

Acto seguido se debe elegir el profile creado previamente y que contiene el Servidor Jaguar y presionar **NEXT**, como se ilustra en el gráfico 3.61

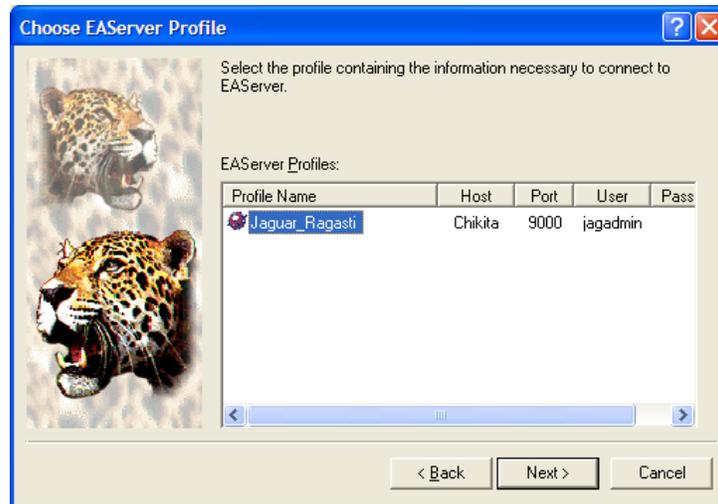


Figura 3.61. Eligiendo el Profile de P. W.

Luego se debe elegir el paquete en el Jaguar Server en el que se va a crear este componente, como se puede observar en el Gráfico 3.62

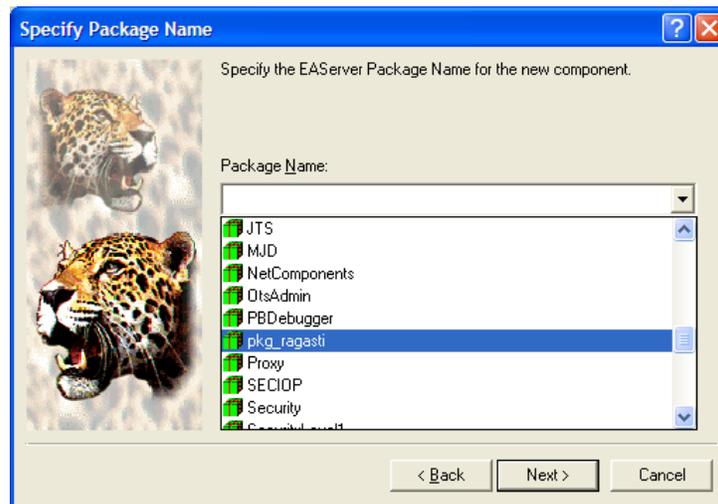


Figura 3.62. Eligiendo el Paquete de P. W.

A continuación se debe elegir el tipo de componente y presionar **NEXT**, estos tipos de componentes depende de en que se los va a usar, como se puede observar en el Gráfico 3.63.



Figura 3.63. Eligiendo el Tipo de Componente

Se elige la opción Supported ya que esta significa que si existe muchas personas ejecutando el componente se va a crear nuevas instancias y al cerrar se cierra solamente la instancia abierta. Una vez seleccionado presionar el botón **NEXT**, como se observa en el Gráfico 3.64

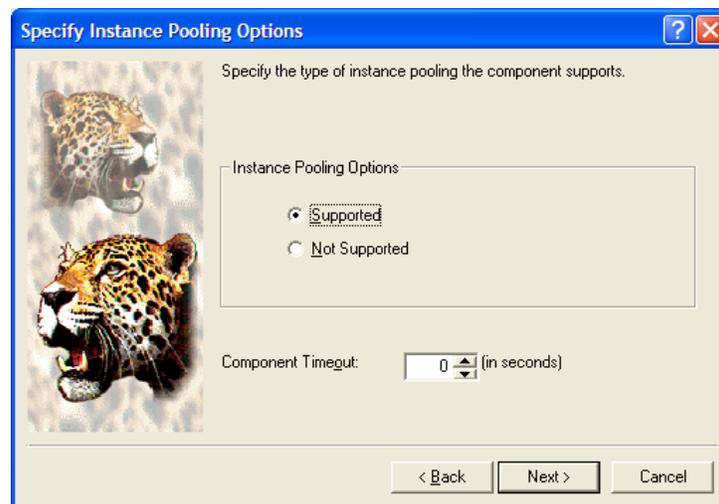


Figura 3.64. Eligiendo el Tipo de Interfase

A continuación se debe elegir la opción de transacciones dependiendo si el componente va a soportar o no, presionar **NEXT** al finalizar, como se ilustra en el gráfico 3.65.



Figura 3.65. Eligiendo el Tipo de Transacción del Componente

De igual manera se debe elegir si se desea exponer información de la interfaz entre otras, se recomienda no elegir estas opciones, y presionar **NEXT**, como se ilustra en el gráfico 3.66

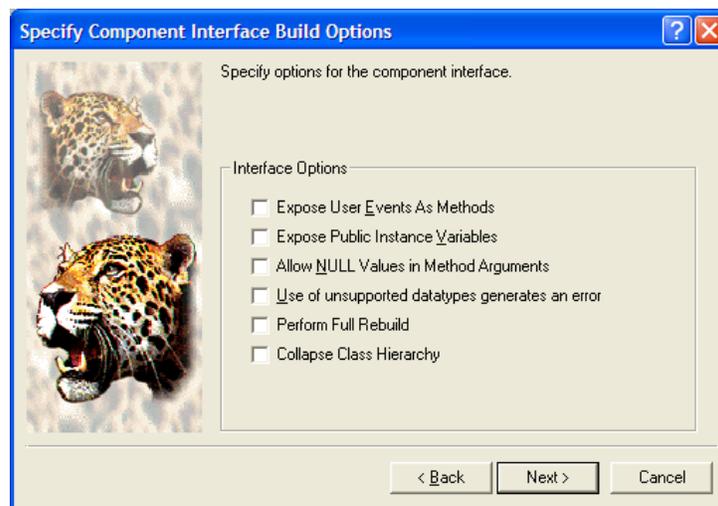


Figura 3.66. Compartiendo Información entre Componentes

Acto seguido, se debe elegir la opción sobre actividades remotas en los componente, se recomienda elegir Support Remote Debbing y presionar **NEXT**, como se ilustra en la gráfica 3.67

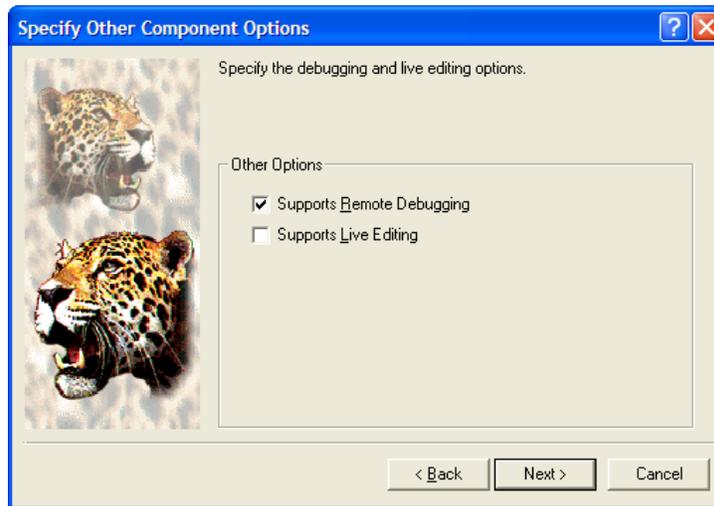


Figura 3.67. Escogiendo Opciones del Componente

Como uno de los últimos de los pasos, se debe escribir el nombre del proyecto y realizar una descripción del mismo, luego presionar **NEXT**, como se puede observar en el gráfico 3.68.

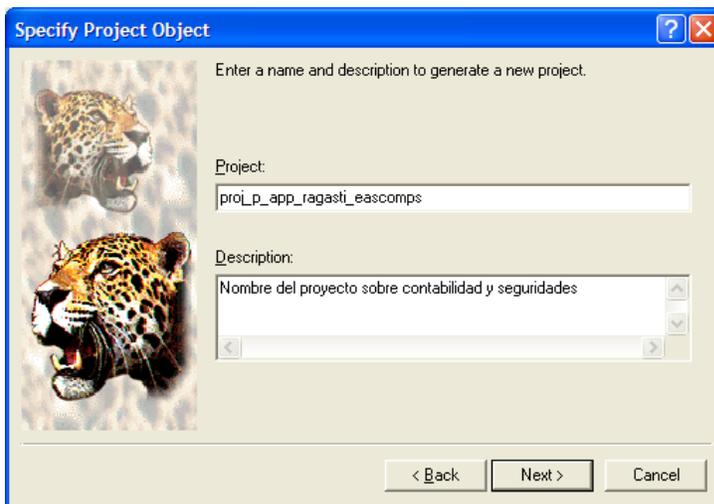


Figura 3.68. Asignando un Nombre al Proyecto

Se recomienda dejar las mismas opciones de organización y dar clic sobre **NEXT**, como se observa en el Gráfico 3.69.

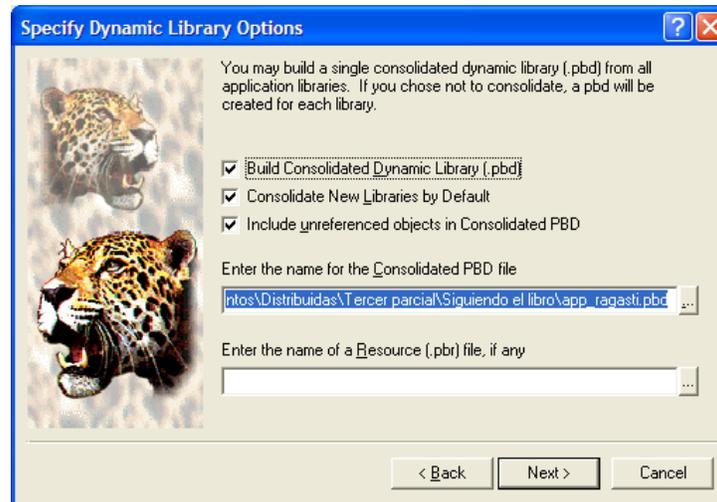


Figura 3.69. Opciones de Organización del Proyecto

Para terminar se comprueba los datos ingresados y se presiona el botón **Finish** para terminar el Wizard, como se ilustra en el Gráfico 3.70.

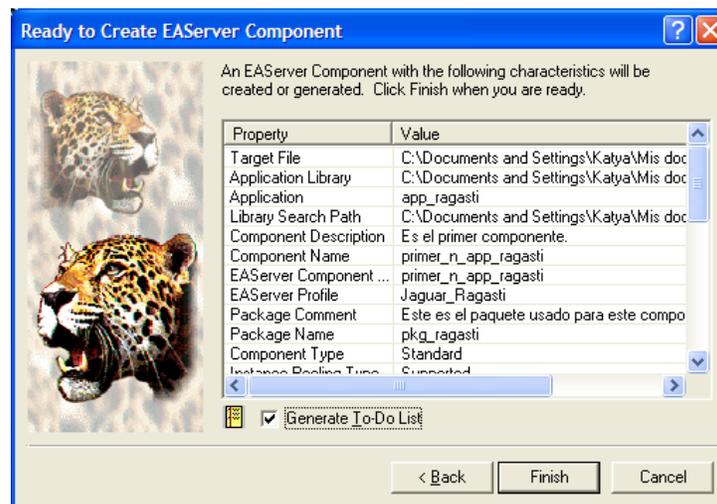


Figura 3.70. Configuración del Componente

La forma de comprobar que el wizard debe haber creado todos los elementos es dirigiéndose al workspace de Power Builder como se ve en el Gráfico 3.71.

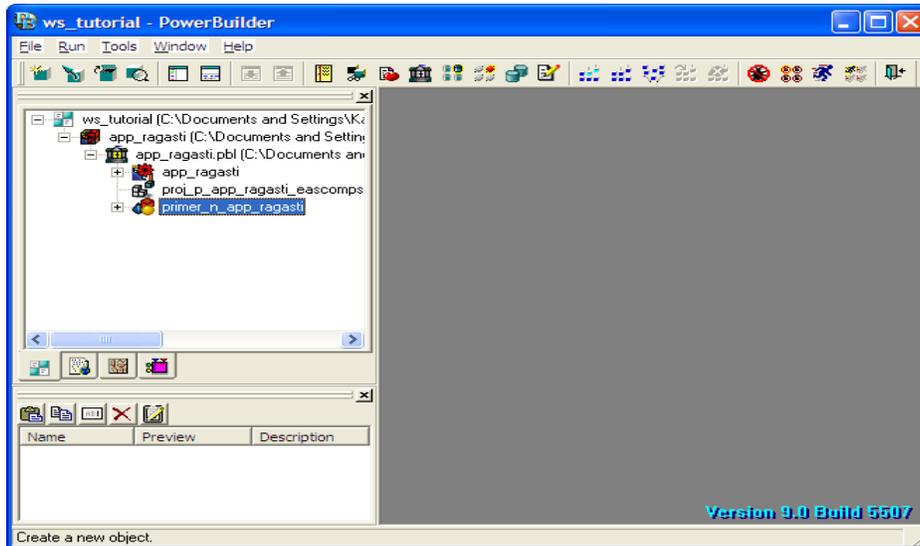


Figura 3.71. Comprobación del Componente Creado

Es necesario abrir el User Object y elegir la opción de nueva función, como se ilustra en el Gráfico 3.72.

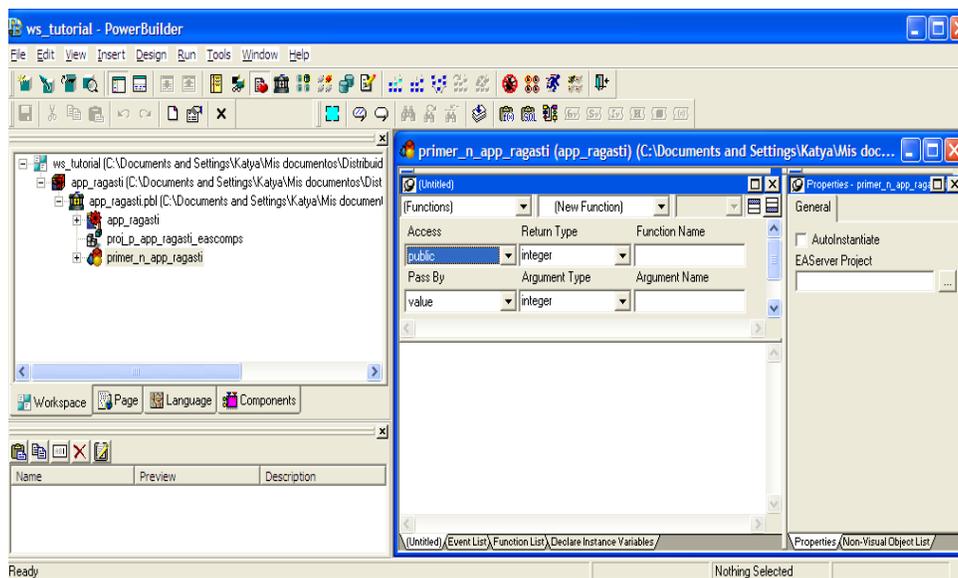


Figura 3.72. User Object

Luego se debe elegir los parámetros de la función y escribir la función en el espacio en blanco. Para aumentar valores recibidos por la función dar un tab en **Argument Name**. Si con el Wizard no se habilitó la opción de Live-Editing, de debe hacer el deploy en el proyecto y Build, como se ve en el Gráfico 3.73.

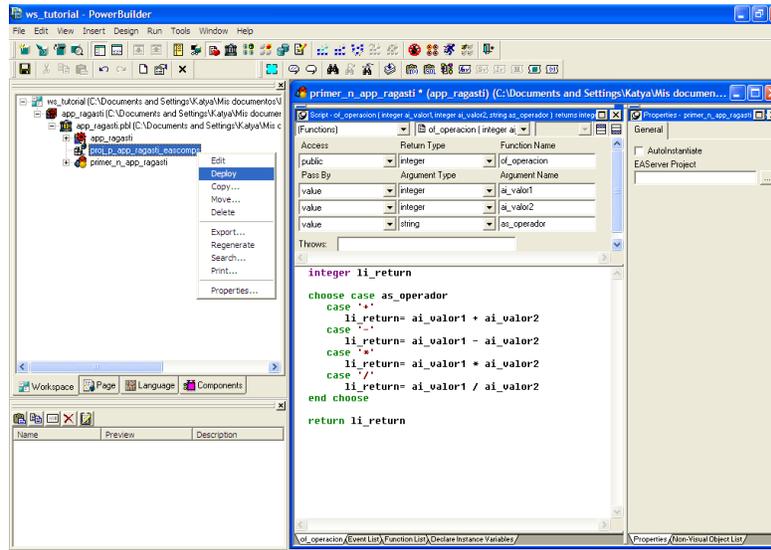


Figura 3.73. Parámetros de la Función

A continuación se debe abrir el proyecto donde se encuentra la información que se genero con el Wizard anteriormente, como se puede observar en el Gráfico 3.74

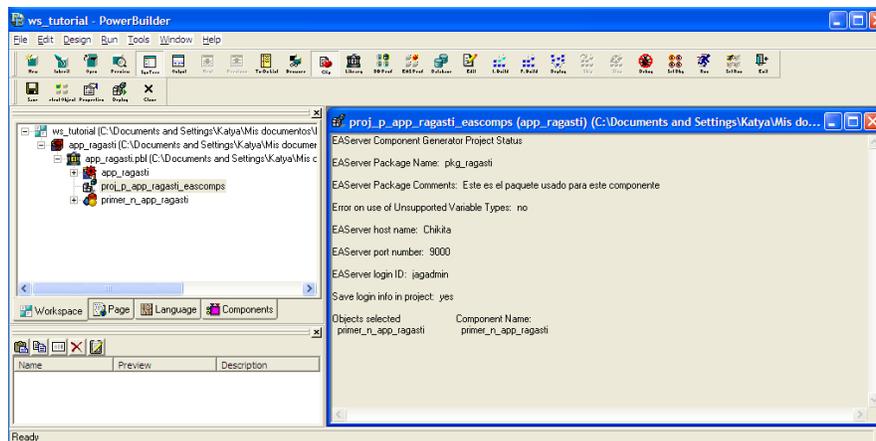


Figura 3.74. Información Generada del Proyecto

La información generada del proyecto se despliega en pantalla, para actualizar dicha información, se debe ir a la pestaña de componentes y dar clic derecho sobre el servidor EAServer y elegir la opción **Refresh**, como se ilustra en el grafico 3.75

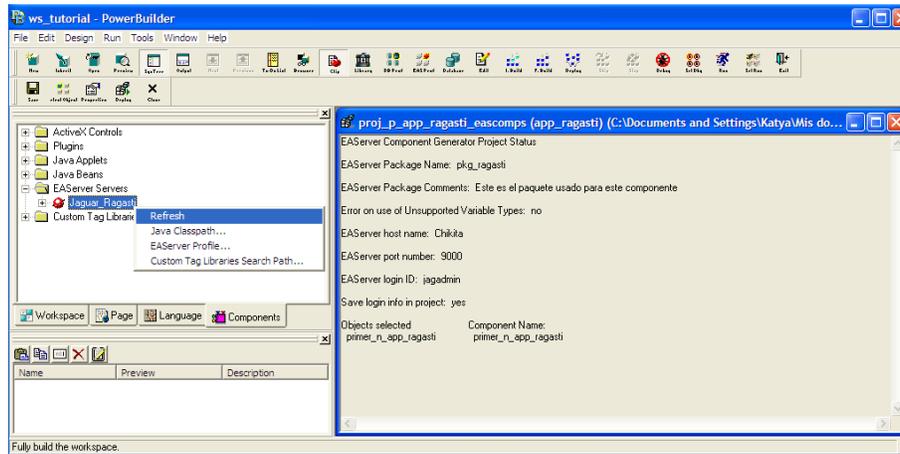


Figura 3.75. Información Actualizada del Proyecto

De esta manera, al actualizar la información, se despliega el componente creado ya en el paquete del EAServer, como se ilustra en el gráfico 3.76.

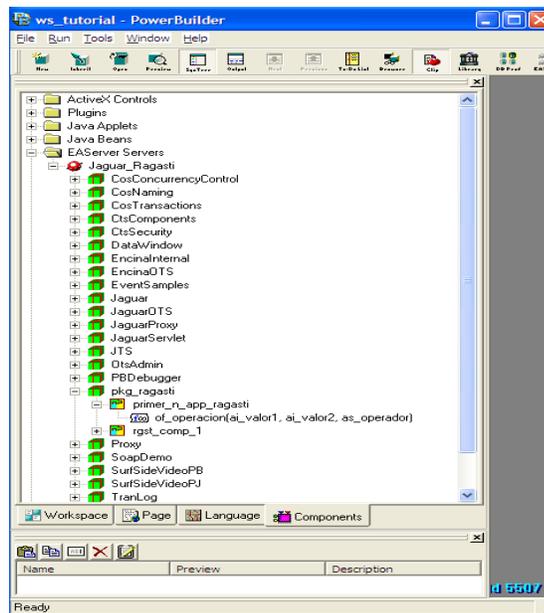


Figura 3.76. Información del Componente Creado

Las áreas de trabajo definidas por la herramienta Power Builder y utilizadas en el Sistema ESPE – CISCO son las siguientes:

Area		Descripción
	Application	Permite definir las características comunes que tendrá la aplicación, como son fuente de letra para las etiquetas, controles, etc. También permite definir las Librerías que conforman la aplicación, y las acciones que se deben realizar al iniciar/terminar la aplicación, como son la ventana de inicio, etc. La aplicación se torna en la cabeza de la aplicación.
	Project	Permite crear un archivo ejecutable, y especificar los componentes que irán con esta aplicación.
	Window	Permite trabajar con las ventanas que conformaran la aplicación. Las ventanas se convierten en contenedores de los objetos visuales, es decir, están puestos encima de esta; para que el usuario pueda manipularlos.
	User Object	Permite definir nuevos tipos de objetos; a partir de los ya existentes o bien creándolos con una programación C++. El primer tipo es el más utilizado, ya que se puede formular código estándar, y de esa manera disminuir el número de líneas de código. Estos objetos aquí definidos pueden ser utilizados en cualquier ventana de la aplicación.
	Menu	Permite declarar los mouse. Esto significa que pueden existir diversos listados de mouse que sean utilizados por una aplicación. Junto con esto es posible definir las acciones a realizar cuando se selecciona cualquier ítem del menú hecho.
	Structure	Permite definir las estructuras de datos que serán utilizadas por la aplicación. Se entiende por estructura de datos, un elemento que contiene subelementos en el, y que son dependientes de este. Se puede reflejar mucho mejor esto en la idea de 'registro'.
	Function	Permite definir las funciones que serán utilizadas por la aplicación, y podrán ser accedidas desde cualquier parte de la aplicación. Estas funciones podrán o no retornar un valor, y pueden o no tener parámetros.
	Datawindow	Permite definir los datawindows, estas son consultas que están unidas a una presentación definida por uno. Donde se puede definir reglas de validación para cada campo, y estilos de presentación, Ud. puede definir argumentos, uniones, ordenamiento criterios de selección y agrupamiento.
	Quero	Es utilizado para definir y guardar una consulta Select en forma gráfica, esto permite que pueda ser utilizada por objetos datawindows.
	Pipeline	Un objeto 'Tubería' es usada para administras el traspaso de datos durante la ejecución de una aplicación. Normalmente se hace esto para hacer traspasos masivos de datos desde una Base de Datos a otra.
	Configure ODBC	Permite definir un profile (definición) de conexión, el que utilizara el controlador seleccionado; el que deberá estar presente en la máquina.
	DB Profile	Permite seleccionar un profile, previamente definido, con el que se conectara a la Base de Datos correspondiente. Para cada tipo de Base de Datos los parámetros con los que se sirve para conectarse podrían variar, infórmese.
	Database	Permite trabajar con la Base de Datos conectada actualmente al seleccionar un profile del área DB Profile. Aquí es posible

		Crear/Modificar Tablas, Claves Primarias/Foráneas, Manipular datos.
	Library	Use está área de trabajo para crear y mantener las librerías, y los objetos que contienen estas. Estos objetos pueden ser la Aplicación, Ventanas, ouse, etc. Use el botón ‘Properties’  , o bien la misma opción del menú al presionar el botón derecho del ouse; para modificar la descripción dada a cada objeto. También es posible copiar, mover objetos de una librería a otra, o bien eliminarlos expresamente.

Tabla 3.3 Tabla Definición de Herramientas P.W.

CREANDO UNA APLICACIÓN CLIENTE EN POWER BUILDER QUE USE EL COMPONENTE

En necesario crear un nuevo target de aplicación para crear los objetos visuales, es decir la interfaz, luego se presionar **NEXT**.

OBJETO CONNECTION: El cliente se va a comunicar con el servidor mediante el objeto conexión, en este se define la información para que se conecte con el servidor Jaguar como driver (protocolos de red), location (la ubicación del servidor y el puerto que ocupa), el user id, password y la cadena de conexión (para establecer la conexión). Este objeto tiene 3 funciones el constructor, destructor y error.

Para crear un objeto conexión mediante el wizard, se debe Ubicar en la aplicación cliente, entrar al menú **Connection Object Wizard** y presionar **OK**, como se ilustra en el gráfico 3.77.

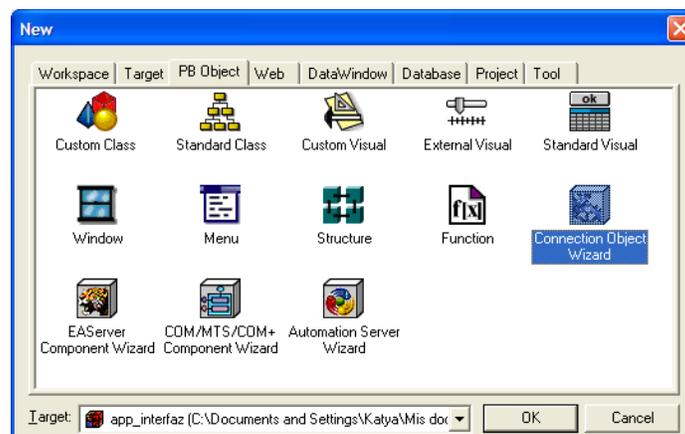


Figura 3.77. Wizard del Objeto Conexión

A Continuación se obtiene una pantalla con la explicación del wizard, se debe dar un clic en el botón **NEXT**, como se ilustra en el Gráfico 3.78



Figura 3.78. Explicación del Wizard

Después de haber dado un click sobre el botón Next se debe determinar la ubicación de los elementos con el fin de mantener una buena organización, a continuación se da un clic en **NEXT**, como se muestra en el Gráfico 3.79.

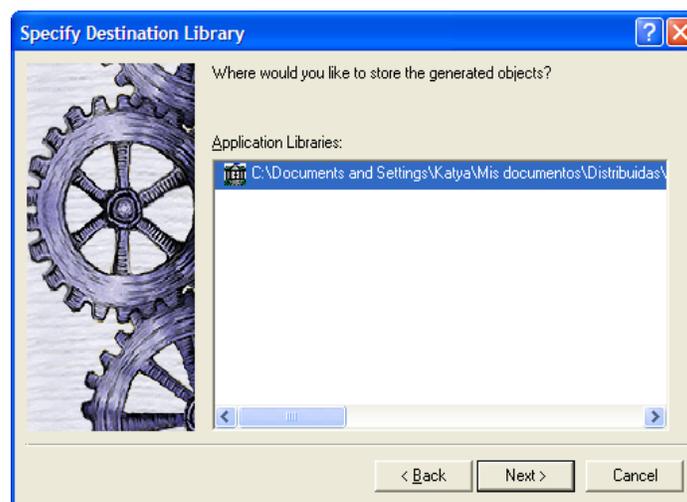


Figura 3.79. Ubicación de los Elementos

Luego de haber elegido la ubicación en donde reposaran los elementos para facilidad de administración, se requiere una conexión al Servidor EASERVER por ello se debe elegir la opción y presionar **NEXT**, como se ilustra en el gráfico 3.80.

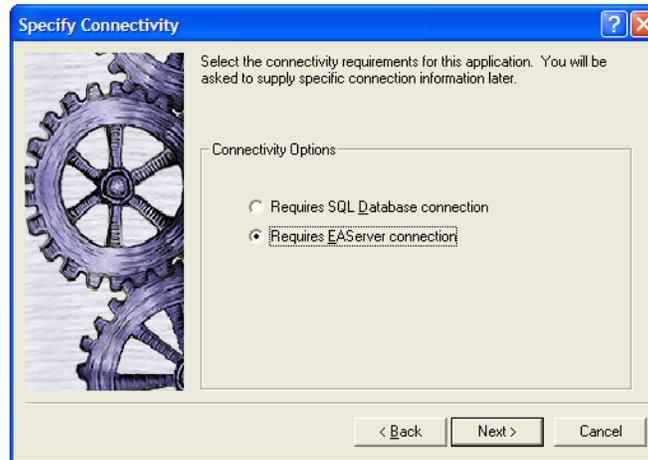


Figura 3.80. Conexión al Servidor EASERVER

Se debe elegir el direccionamiento anteriormente creado en el profile y dar clic sobre **NEXT**, como se ilustra en el Gráfico 3.81.

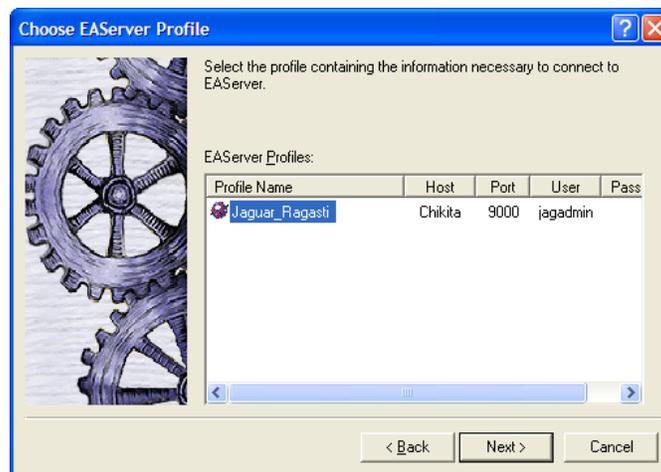


Figura 3.81. Direccionamiento del Profile

A continuación se debe escoger en el paquete donde se encuentra el componente a ejecutar en el servidor y presionar **NEXT**, como se ilustra en el Gráfico 3.82.

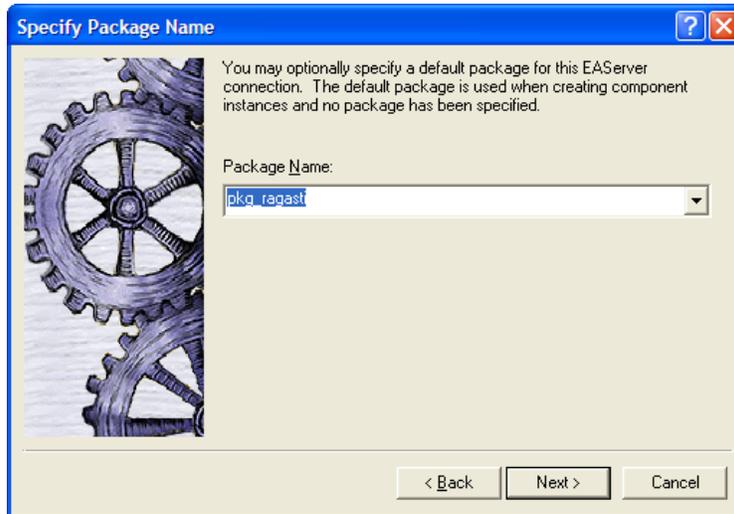


Figura 3.82. Escogiendo el Paquete

Es necesario dar un nombre a la conexión y a continuación se da un clic en **NEXT**, como se ilustra en el Gráfico 3.83.



Figura 3.83. Asignando un Nombre a la Conexión

Para realizar una conexión se lo hará por medio de un archivo .ini, se elige la opción y se presiona **NEXT**, como se ilustra en el Gráfico 3.84.

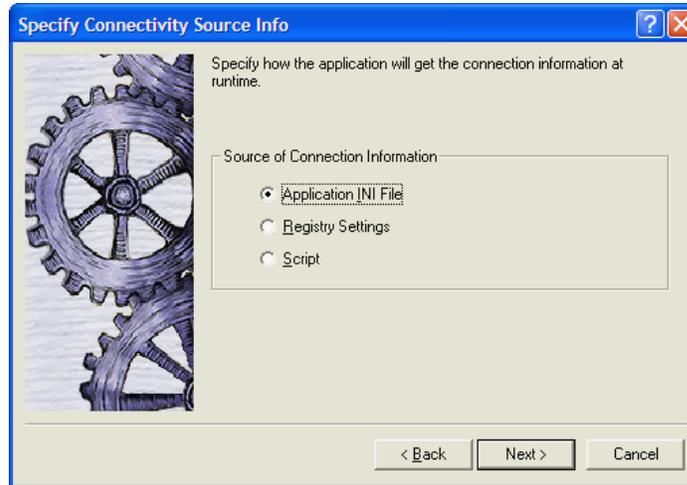


Figura 3.84. Eligiendo la opción de archivo .ini

A continuación se determina la ubicación del archivo .ini y luego presionar **NEXT**, como se ilustra en el Gráfico 3.85.

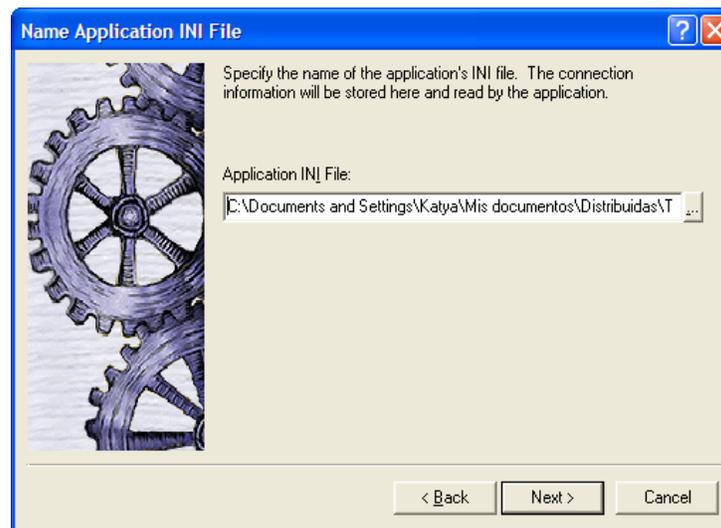


Figura 3.85. Eligiendo la ubicación del archivo .ini

Para terminar con el Wizard, se muestra un reporte de las actividades realizadas, es necesario elegir la opción To-do list y dar clic en **Finís**, como se ilustra en el Gráfico 3.86.

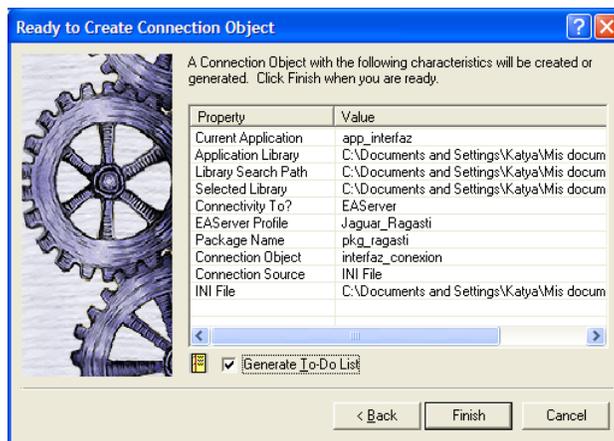


Figura 3.86. Reporte de Actividades

CREAR UN OBJETO EASERVER PROXY MEDIANTE WIZARD

Al igual que el objeto conexión, se debe ubicar en la opción NEW, en la pestaña de Project elegir la opción de EAServer Proxy para crear con wizard, dar clic en **OK**, como se ilustra en el gráfico 3.87

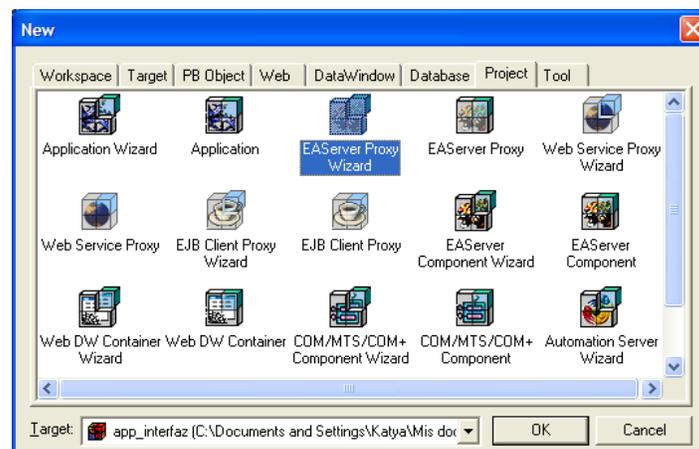


Figura 3.87. Creación del EASERVER PROXY

A continuación se obtiene al igual una pantalla de información, se debe dar clic en **NEXT**, como se ilustra en el gráfico 3.88.

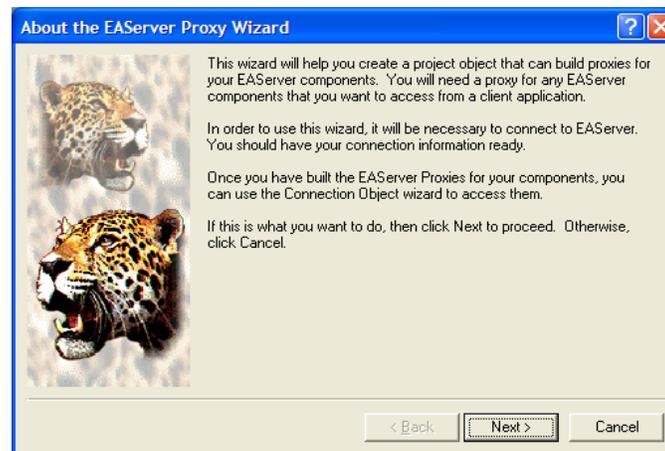


Figura 3.88. Pantalla de información del EASERVER PROXY

Es necesario especificar el lugar donde se desea guardar el EAServer Proxy creado, como se ilustra en el gráfico 3.89.

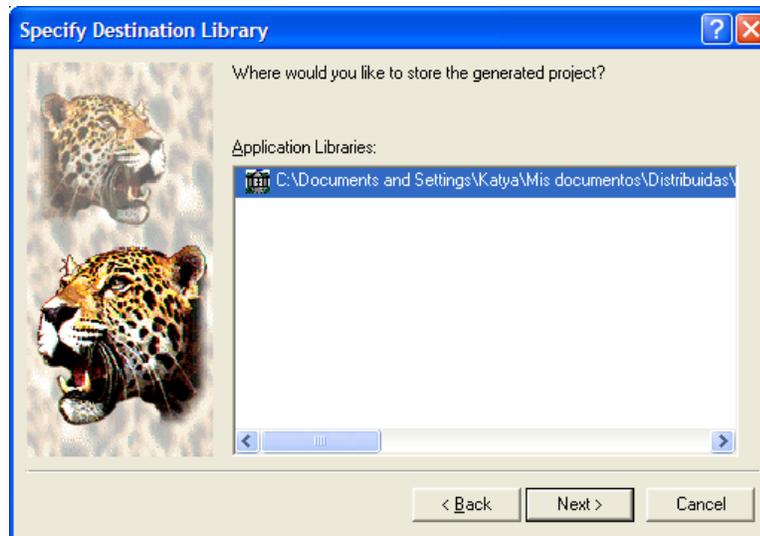


Figura 3.89. Ubicación del EASERVER PROXY

Se debe asignar un nombre al proyecto a crearse, una descripción del mismo y dar clic en **NEXT**, como se ilustra en el Gráfico 3.90.

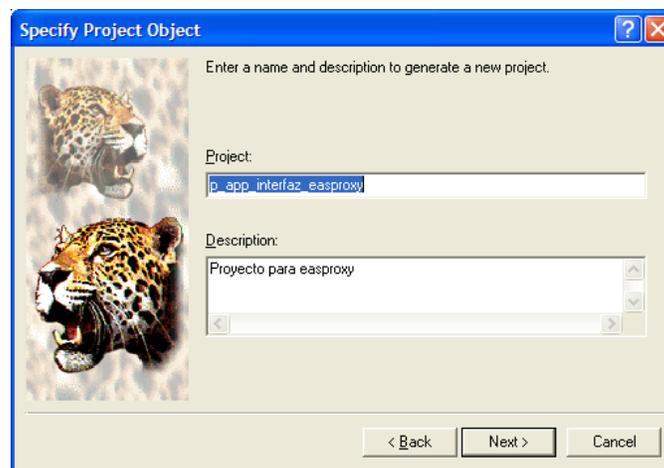


Figura 3.90. Asignando un nombre al Proyecto del EASERVER PROXY

A continuación es necesario elegir el profile de EAServer necesario para obtener la información para la conexión y presionar NEXT, como se ilustra en el Gráfico 3.91.

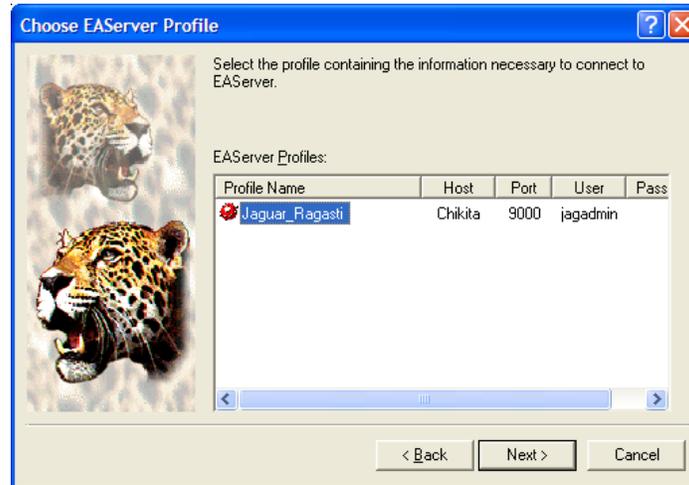


Figura 3.91. Escogiendo el Profile del EASERVER PROXY

Es necesario también elegir el paquete donde se encuentran los componentes a utilizar en la aplicación, y luego se presiona en NEXT, como se ilustra en el Gráfico 3.92.



Figura 3.92. Escogiendo los Paquetes de los Componentes del EASERVER PROXY

De igual manera se debe especificar en donde se va a almacenar este Proxy, con el mismo objetivo de tener organizada la información, luego se da un clic en **NEXT**, como se ilustra en el gráfico 3.93.

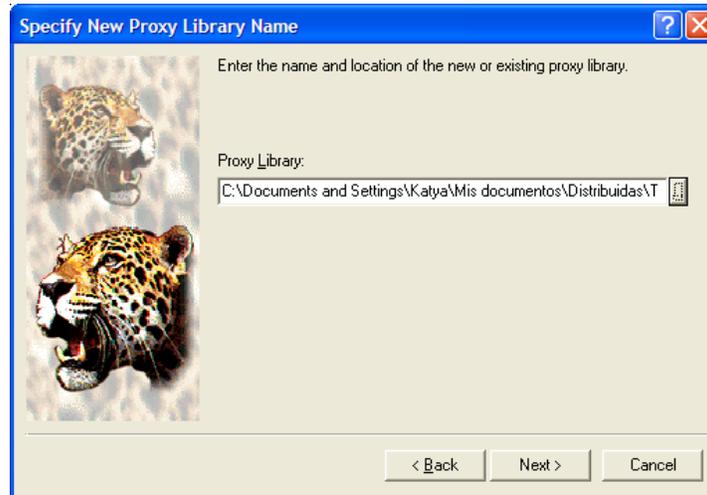


Figura 3.93. Eligiendo la ubicación del EASERVER PROXY

Automáticamente se tiene una nueva pantalla en donde se escoge la opción de construcción, se debe proseguir con **NEXT** sin escoger ninguna opción, como se muestra en El gráfico 3.94.

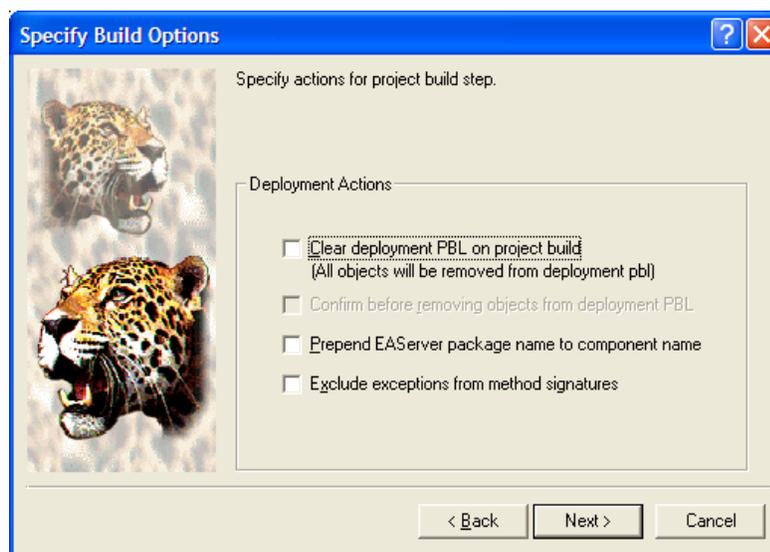


Figura 3.94. Opciones de Construcción del EASERVER PROXY

Al final de la creación y la configuración del EAServer Proxy, se presenta el reporte sobre el wizard, para finalizar presionar **Finís**, como se ilustra en el Gráfico 3.95.

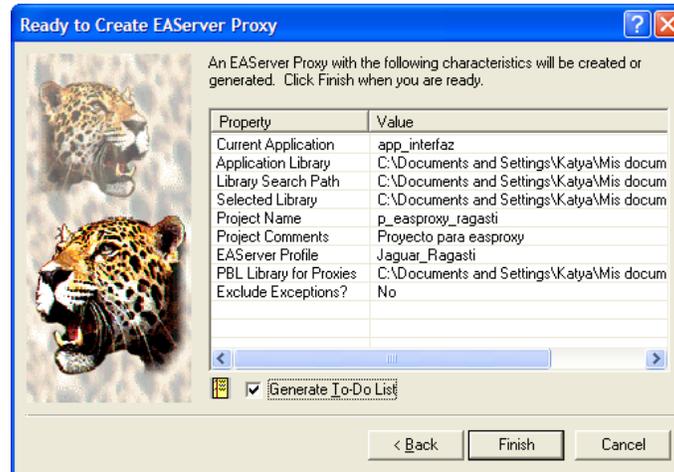


Figura 3.95. Reporte del EASERVER PROXY

CREAR LA INTERFAZ DE LA APLICACIÓN CLIENTE

Mediante el menú **NEW** en la pestaña **Target**, se elige la opción **Application Template**, y para iniciar el wizard, se debe de hacer click sobre el botón **OK**, como se ilustra en el Gráfico 3.96.

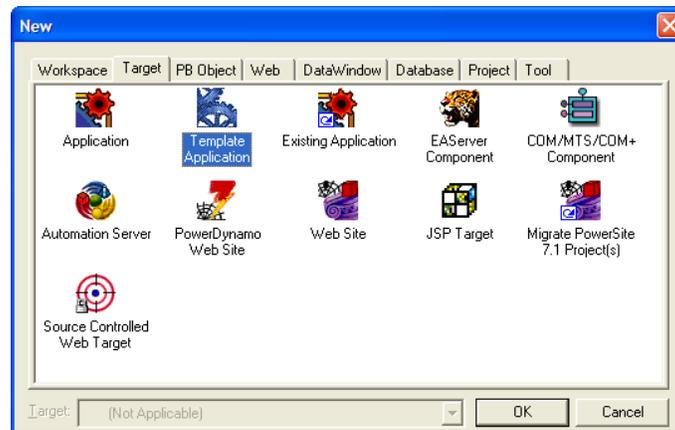


Figura 3.96. Creación de la Interfase del Cliente

Se obtiene nuevamente una pantalla informativa, para continuar con la siguiente pantalla presionar **NEXT**, como se ilustra en el Gráfico 3.97



Figura 3.97. Pantalla Informativa de la Interfase del Cliente

Automáticamente se despliega una pantalla en donde se obtiene la lista de actividades a realizar el Wizard, continuar presionando en **NEXT**, como se ilustra en el gráfico 3.98.

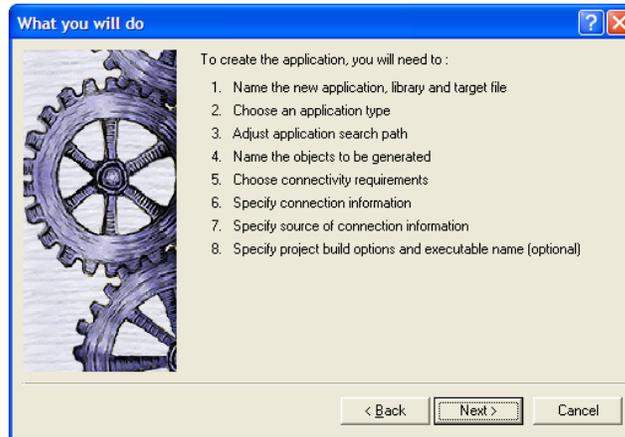


Figura 3.98. Lista de Actividades de la Interfase del Cliente

Es necesario determinar el nombre de la aplicación a crear, determinar las librerías entre otros datos y presionar **NEXT**, como se ilustra en el gráfico 3.99.

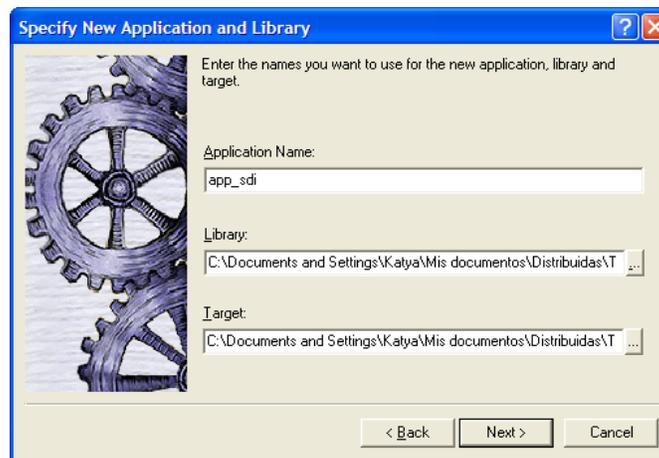


Figura 3.99. Determinación de Datos Necesarios de la Interfase del Cliente

Se debe elegir la opción de una aplicación SDI y se da clic en **NEXT**, como se ilustra en el gráfico 3.100

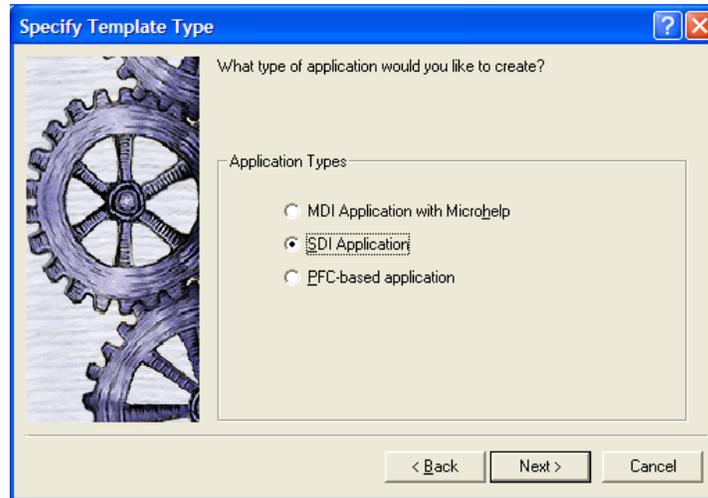


Figura 3.100. Opción SDI de la Interfase del Cliente

De igual manera se elige donde se va a ubicar la aplicación, a continuación se presiona en **NEXT**, como se ilustra en el Gráfico 3.101.

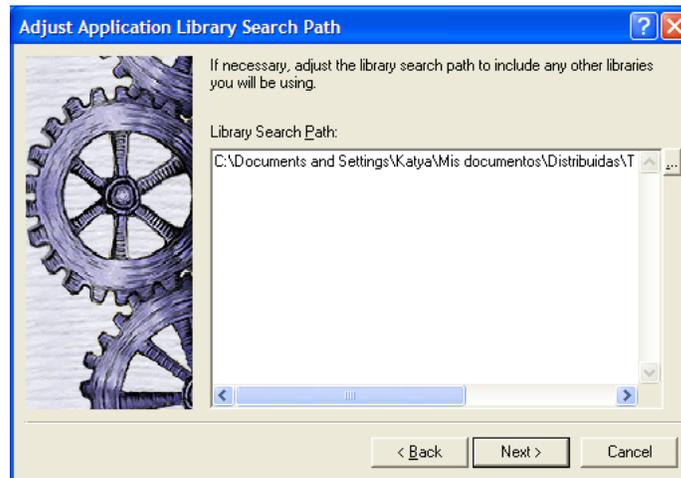


Figura 3.101. Ubicación de la Interfase del Cliente

Es necesario especifica el nombre de la pantalla principal y del menú, después se presiona en el botón **NEXT**, como se ilustra en la pantalla 3.102.

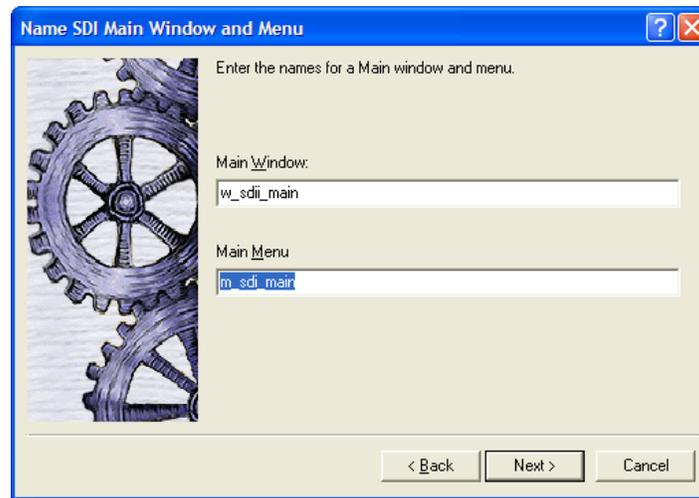


Figura 3.102. Especificación del Nombre y Menú de la Interfase del Cliente

Se debe especificar también el nombre de la ventana de información y presionar **NEXT**, como se ilustra en el gráfico 3.103.

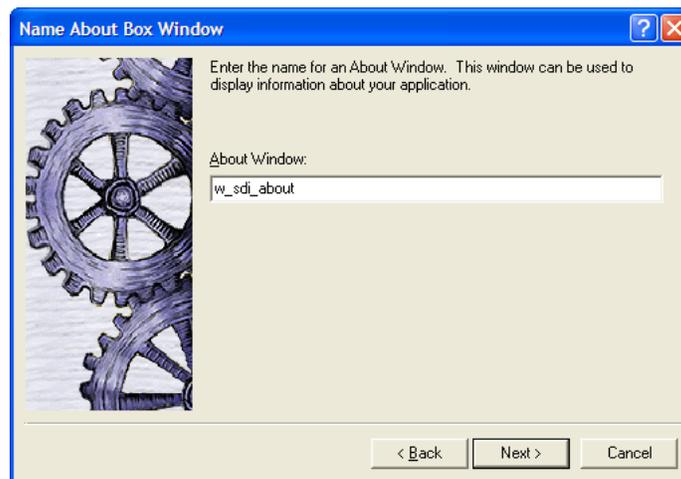


Figura 3.103. Especificación del Nombre de la Ventana de la Interfase del Cliente

Luego es necesario especificar la conexión con el EAServer y presionar **NEXT**, como se ilustra en el gráfico 3.104

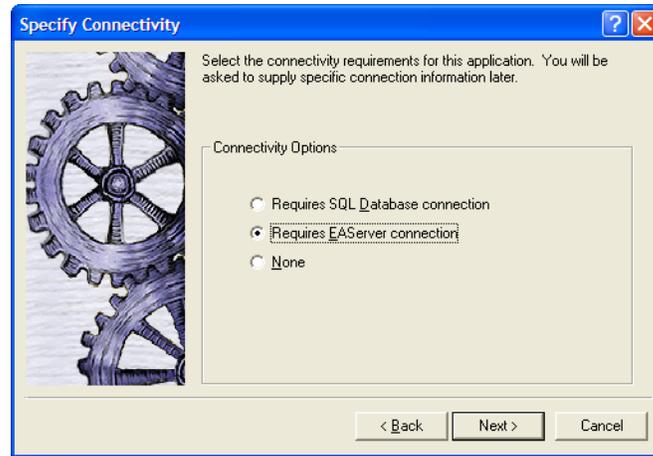


Figura 3.104. Conexión del EAServer con la Interfase del Cliente

Automáticamente se selecciona el profile de EAServer y el servidor después presionar **NEXT**, como se ilustra en el Gráfico 3.105.

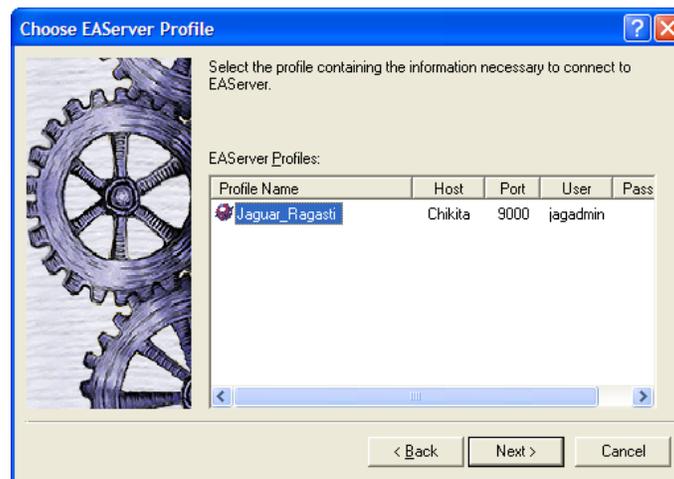


Figura 3.105. Conexión del Profile con la Interfase del Cliente

Es necesario elegir el paquete donde se encuentran los componentes y presionar **NEXT**, como se ilustra en el gráfico 3.106

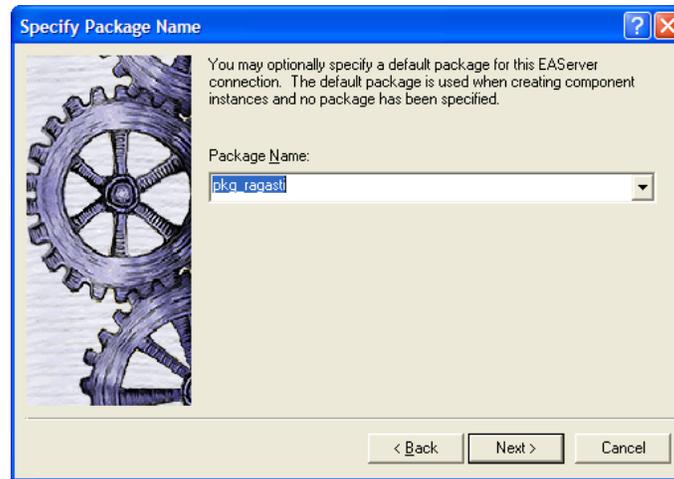


Figura 3.106. Elección de los paquetes con los Componentes

CREACIÓN DEL MENU PRINCIPAL DEL SISTEMA

El menú principal del sistema fue creado en la librería **suca.pbl** el cual contiene un listado desplegable de opciones a ser utilizadas por los usuarios del sistema.

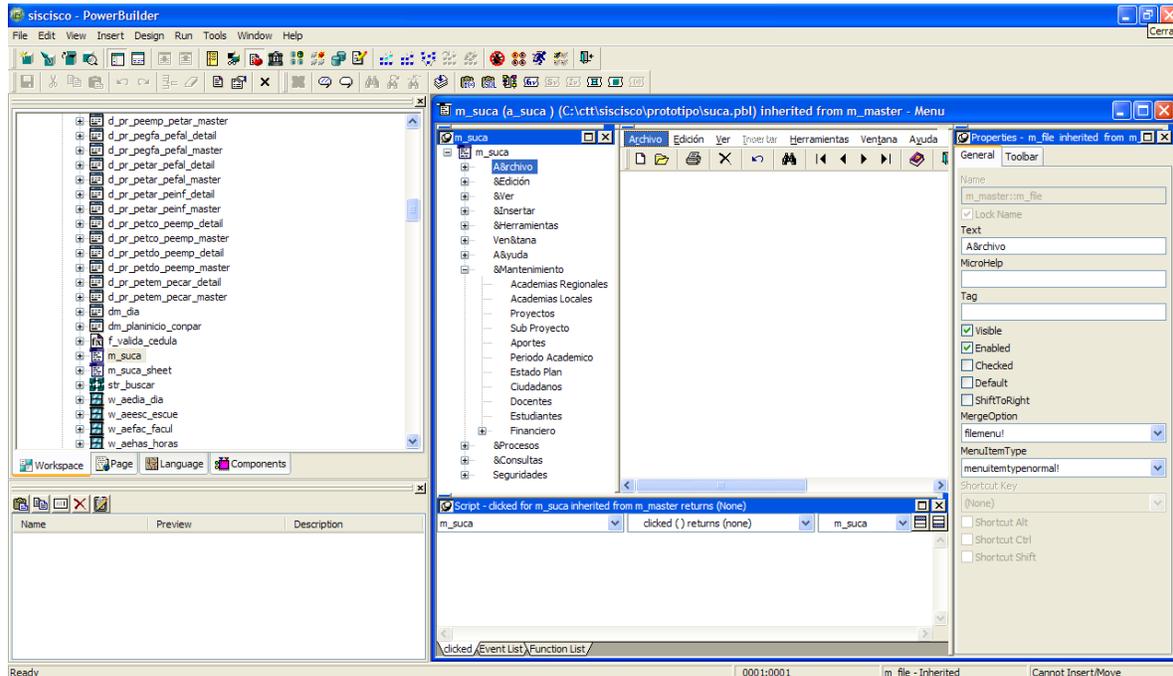


Figura 3.107. Creación del Menú Principal

CREACION DE EVENTOS

Los eventos son acciones que se pueden realizar sobre determinados objetos, siendo estas independientes unas de otras.

A continuación se procedió a crear los eventos que son utilizados en los diferentes formularios del proyecto, los mismos que tienen la siguiente presentación:

Botón de Evento	DESCRIPCIÓN DEL EVENTO
	Permite Añadir un nuevo registro a la tabla.
	Permite Abrir un Registro.
	Permite Guardar un Registro.
	Permite Imprimir el registro actual en la tabla activa
	Permite Eliminar el registro actual en la tabla activa
	Permite Insertar una nueva fila para el ingreso de nuevos registros
	Permite Borrar una fila del registro en la tabla activa
	Permite Cortar los valores escogidos de un registro en la tabla activa
	Permite Copiar los valores escogidos de un registro en la tabla activa.
	Permite Pegar los valores Copiados o Cortados de un registro
	Permite deshacer una acción mal ingresada.
	Permite Buscar un registro en la tabla activa
	Permite ubicarse sobre el primer registro de la tabla abierta.
	Permite retroceder al registro anterior en la tabla actual
	Permite avanzar al registro siguiente en la tabla actual
	Permite ubicarse sobre el último registro de la tabla abierta.
	Despliega la ayuda del sistema.
	Permite salir del sistema.

Tabla 3.4 Tabla Descripción de Eventos.

CREACIÓN DE VENTANAS (WINDOWS)

Todas las librerías del Sistema están basadas en ventanas las mismas que se constituyen en los contenedores de los objetos o controles.

Para la definición de las ventanas se consideraron las propiedades que estas tienen y que están organizados en forma de tópicos. Las propiedades de las ventanas se definen en el siguiente cuadro:

Tópico	Descripción
General	<p>Casilla 'Title', indica el texto de título de la ventana. La casilla 'Menu Name', indica el nombre del objeto Menu para la ventana. 'Window Type', indica el tipo de ventana:</p> <ul style="list-style-type: none"> • Child, ventana que permanece inserta dentro de una ventana MDI. • Main, ventana normal, con posibilidad de menú. • MDI Frame, ventana que posee área interna para contener otras ventanas. • MDI Frame with ..., igual que la anterior, pero con barra de estado inferior. • Response, necesita cerrar esta ventana para continuar ejecución.
Position	Permite indicar la posición de la ventana dentro de la pantalla, como será presentada. Utilice el esquema inferior para mover la ventana.
Icon	Identifica al Icono que aparecerá en la parte izquierda de la cabecera de la ventana.
Scroll	Permite definir si la ventana tendrá barras de desplazamiento, y la unidad a utilizarse por ellas.
ToolBar	Permite definir si se utilizara la caja de herramienta definida para el menú, su presentación y posición.

Tabla 3.5 Tabla Definición de Ventanas.

FUNCIONES PARA EL MANEJO DE LAS VENTANAS(WINDOWS)

Entre las principales funciones utilizadas para el manejo de las diferentes ventanas utilizadas en el sistema son las siguientes:

Añadir/Insertar

Permite añadir o insertar un nuevo registro para lo cual se utiliza la siguiente sintáxis:

<nom Dw>.InsertRow(<pos>)

Donde:

- *<nom Dw>* es el nombre del Datawindow Control.
- *<pos>* Es la posición, dentro del número de registros mostrados en el Datawindow Control, donde se insertara el nuevo registro. Si *<pos>= 0*, añade un registro al final de los ya existentes. El añadir un registro, para ser llenado los datos de los campos por el usuario, no implica que el foco se mueva a este nuevo registro.

Eliminar

Para eliminar un registro debe conocer cual es la posición, dentro del número de registros mostrados en el Datawindow Control, del registro a eliminar.

Esto se hace, con la siguiente función:

<nom Dw>.DeletedRow(*<pos>*)

Donde:

- *<nom Dw>* es el nombre del Datawindow Control.
- *<pos>* es la posición del registro a eliminar. Si desea eliminar el registro donde esta el foco (cursor actualmente), utilice la siguiente función: *<nom Dw>*.GetRow()

Buscar

Permite buscar un registro, dentro de los listados en el Datawindow Control, que coincida con la expresión definida. Para esto utilice la siguiente función:

<nom Dw>.Find(*<expre cond>*)

Donde:

- *<nom Dw>* es el nombre del Datawindow Control.
- *< expre cond >* es una expresión que se escribe igual que una expresión Where. Esta función devuelve la posición del registro que cumpla con la condición, o un cero si no encuentra ninguno.

CREACIÓN DE LAS VENTANAS DE DATOS (DATAWINDOWS)

Las ventanas de datos o datawindows utilizadas en el Sistema están ligadas a representaciones gráficas y contienen campos de las diferentes entidades definidas.

La definición de las datawindows indica dos elementos unidos, aunque distintos entre si mismos:

a. **Origen de los Datos:**

Corresponde a la sentencia SQL que tiene características de Agrupamiento, Ordenamiento, Condicionales (extraer un conjunto definido de registros) y en fin todas las pertinentes a una sentencia SQL pura.

b. **Presentación:**

Corresponde a la distribución gráfica de los elementos de la consulta y otros añadidos, dentro de un área definida para ello. La que podrá ser utilizada para presentar datos, ingresar datos, listarlos o imprimirlos; tal y como aparecen en esta hoja.

SELECCIÓN TIPO Y PRESENTACIÓN

Para trabajar con las Datawindows se presiona el botón , luego de lo cual aparecerá la ventana de selección; se puede seleccionar una ya existente, para modificarla, o presionar el botón 'New' para crear una nueva Datawindows del sistema.

Si escogemos el botón 'New' se debe seleccionar el modo de trabajo para el 'Origen de datos' y también el tipo de 'Presentación'.

MODOS DE TRABAJO

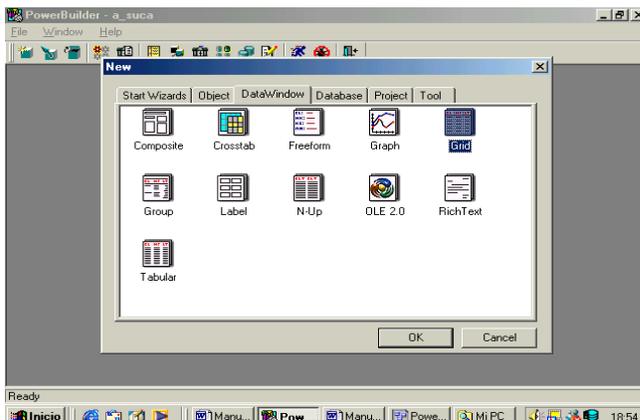
El modo de trabajo define como se trabajara con el Origen de Datos. Existen 5 tipos de Modos de trabajo.

Modo	Descripción
Quick Select	No presenta las tablas en forma gráfica, y solo permite las sentencias ordinarias de SQL (Sort y Criteria) La primera para poder ordenar por el campo específico y la segunda para indicar registros a mostrar dependiendo de un valor condicional (Ej. <39)
Sql Select	Presenta las tablas en forma gráfica, y permite trabajar con todas las características de Sql estándar (Sort, Group, Having, Compute y Where, entre otros)
Quero	El origen de Datos es una Consulta definida previamente en el Area de Trabajo Query; esta consulta queda almacenada como un objeto de Powerbuilder; Se debe seleccionar la consulta que desee utilizar como Origen de Datos de este Datawindows.
External	El origen de datos no esta relacionado con ninguna Tabla. Ud. debe definir los campos que interactúan (nombre, tipo y largo) Utilice los botones 'Add' para añadir un nuevo campo al final de los ya existentes, el botón 'Insert' le permitirá insertar un campo entre otros y el botón 'Delete' le permitirá eliminar el campo actual.
Stored Procedure	El origen de datos es un procedimiento que ha sido almacenado en la Base de Datos. Seleccione uno si lo ha creado, o créelo previamente.

Tabla 3.6. Tabla de Modos de Trabajo.

TIPOS DE PRESENTACIÓN

El Tipo de presentación define como se presentaran los datos en el DataWindows, existen 11 tipos de presentación las que son:



*Figura 3.108
Tipos de Presentación*

Presentación	Descripción
Composite	Utiliza como presentación un reporte previamente creado.
Freeform	La presentación es libre, donde cada campo esta en una línea distinta. Muy utilizado para el ingreso de datos.
Graph	Diseña un gráfico dependiendo de una consulta, esta debe contener datos relacionados.
Gris	Diseña una malla de datos en forma columnada, y cada columna con una justificación completa.
Group	Diseña listado por agrupación de campos. Muy utilizado para reportes de resumen de información.
Label	El diseño es de etiquetas, donde Ud. podrá elegir el tipo de etiqueta de las prediseñadas, en papel continuo o separado, y otras opciones.
Rich Text	Permite definir una carta con combinación de campos del Origen de datos fijado. Se pueden utilizar todas las características del formato de texto RTF.
Tabular	Presenta el origen de datos en forma columnada pero sin ajuste total. Permite una mayor libertad en el diseño.

Tabla 3.7. Tabla de Tipos de Presentación

FUNCIONES DE LAS DATAWINDOWS

Las funciones de las datawindows del sistema son utilizadas en el código de programación y entre las más importantes se detallan a continuación:

AcceptText	Provoca la validación del campo actual si el usuario no pulsó tab
DBCcancel	Provoca la cancelación de un retrieve() en curso
DeletedCount	Devuelve el número de filas que el usuario ha borrado
DeleteRow	Borra una fila
Describe	Devuelve información muy detallada sobre la estructura de la datawindow
Filter	Ejecuta el filtro actual (vease setFilter)

GetChild	Devuelve un apuntador a una datawindow contenida dentro de otra.
GetColumn	Devuelve el número de columna actual
GetColumnName	Devuelve el nombre de la columna actual
GetItemxxx	xxxx representa un tipo de dato. Devuelve el valor contenido en una columna
GetItemStatus	Devuelve el status de una fila o columna
GetRow	Devuelve el número de fila actual
GetSQLSelect	Devuelve la select de la datawindow
GetText	Devuelve un string que contiene el valor de una columna que esta siendo editada antes de que el usuario la confirme cambiando de campo
GetValue	Devuelve un string conteniendo el valor de una columna
InsertRow	Inserta una fila en blanco
Modif.	Modifica la estructura interna de una datawindow. Ni se te ocurra pensar que puedes realizar programas brutales sin conocer a fondo esta función.
Print	Imprime una datawindow
Retrieve	Ejecuta una select contra la base de datos y muestra los datos en la datawindow
RowCount	Cuenta las filas en una datawindow
SaveAs	Salva el contenido de una dw en un fichero. Soporta diversos formatos incluyendo excel, texto, rtf y html
ScrollToRow	Scrollea la datawindow hasta una fila y la hace visible (no confundir con setRow())
SetColumn	Cambia el foco a una columna en concreto
SetFilter	Modifica las condiciones de filtro de una DW
SetFocus	Coloca el foco en una datawindow
SetItem	Modifica el valor de una columna
SetItemStatus	Modifica el status de una fila o columna
SetRow	Hace que la fila especificada sea la actual
SetSort	Modifica las condiciones de ordenador

SetTransObject	Enlaza una datawindow con el objeto trasaccion. Para que una datawindow funcione es imprescindible ejecutar la siguiente instrucción: NombreDw.SetTransObject(SQLCA)
ShareData	Hace que una datawindow comparta datos con otra (Se puede conseguir un efecto de espejo)
Sort	Ordena una Dw con las condiciones especificadas con SetSort()
Update	Graba las modificaciones en la base de datos.

Tabla 3.8. Tabla Funciones de Data Windows.

WEB DW CONTAINER WIZARD

Es necesario crear un Target JSP para crear la página Web, señalando el Target creado, dar clic derecho para elegir la opción nuevo y en la pestaña de Project, escoger la opción de Web DW Container Wizard. JSP datawindow page, para iniciar el wizard al presionar OK.

PARA CREAR UNA PAGINA WEB CON COMPONENTES

Señalando el Target creado, dar clic derecho para elegir la opción nuevo y en la pestaña de Web, escoger la opción de Web JSP datawindow page, para iniciar el wizard al presionar OK, como se ilustra en el Gráfico 3.109

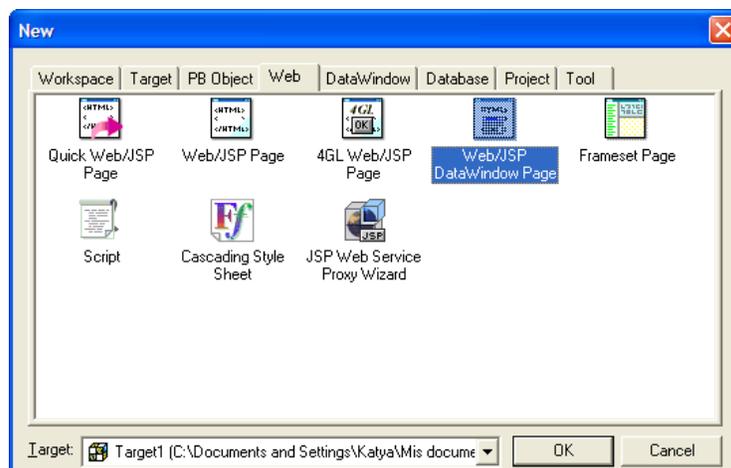


Figura 3.109. Creación del Web/JSP DataWindowsPager

automáticamente un informativo sobre lo que va a realizar el wizard, se debe dar clic sobre NEXT, como se ilustra en el gráfico 3.110.

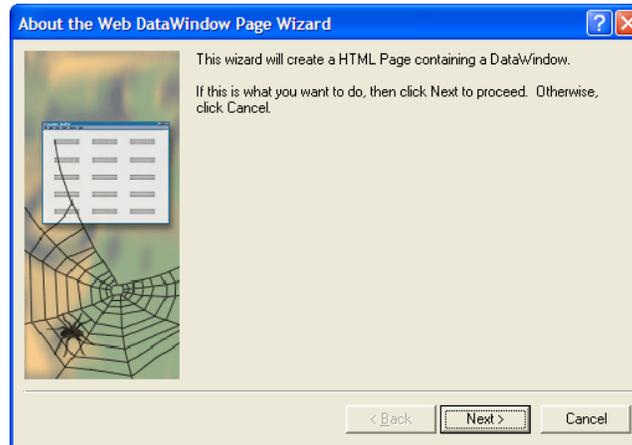


Figura 3.110. Informativo del Web/JSP DataWindowsPager

De igual manera es necesario especificar el nombre de la página Web, como se ilustra en el Gráfico 3.111.

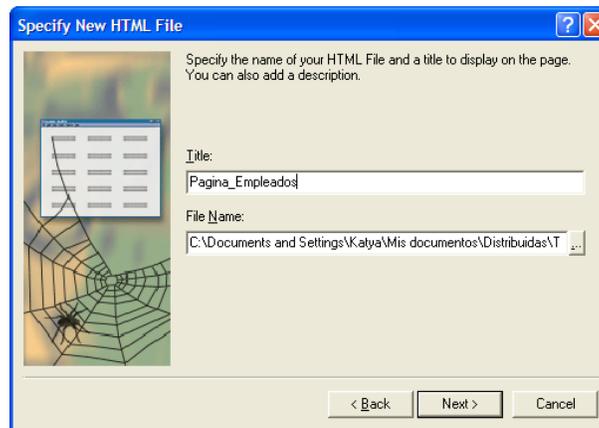


Figura 3.111. Especificación del Nombre del Web/JSP DataWindowsPager

Se debe elegir páginas con diseño establecido o crear un nuevo diseño, como se ilustra el gráfico 3.112



Figura 3.112
Páginas con Diseños Web/JSP
DataWindowsPager

A continuación se elige el Profile del EAServer que tiene la información necesaria para la utilización de los componentes, como se ilustra en el Gráfico 3.113

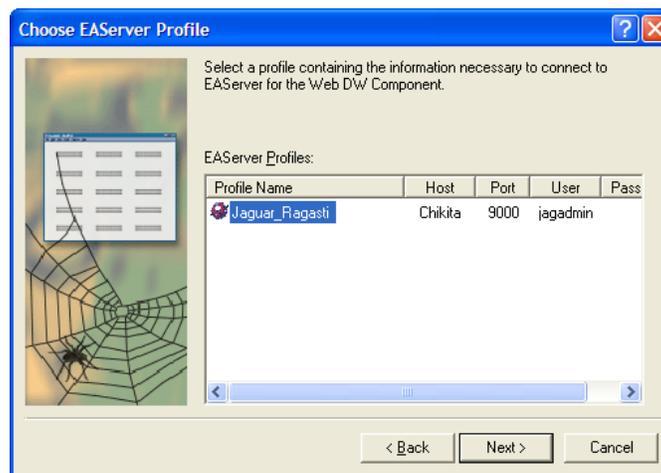


Figura 3.113. Elección del Profile para el Web/JSP
DataWindowsPager

Es necesario especificar la propiedad del datawindow como página Web o como se requiera que este funcione, se ilustra en el gráfico 3.114.

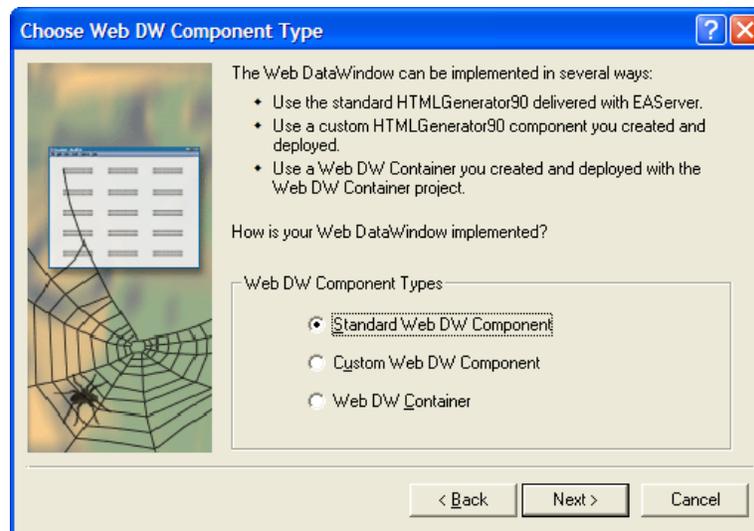


Figura 3.114. Propiedad para el Web/JSP DataWindowsPager

A Continuación se escoge una librería o pbl anteriormente creado, como se ilustra en el Gráfico 115

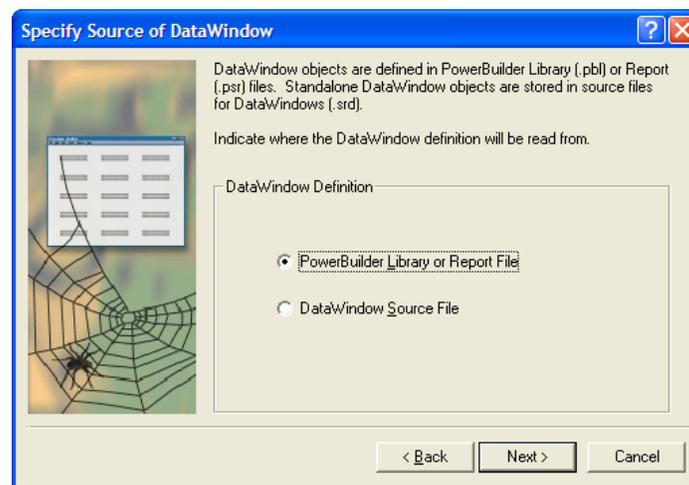


Figura 3.115. Elección de la Librería para el Web/JSP DataWindowsPager

Acto seguido se debe especificar donde se encuentra el PBL escogido en la pantalla anterior, como se ilustra en el Gráfico 3.116.

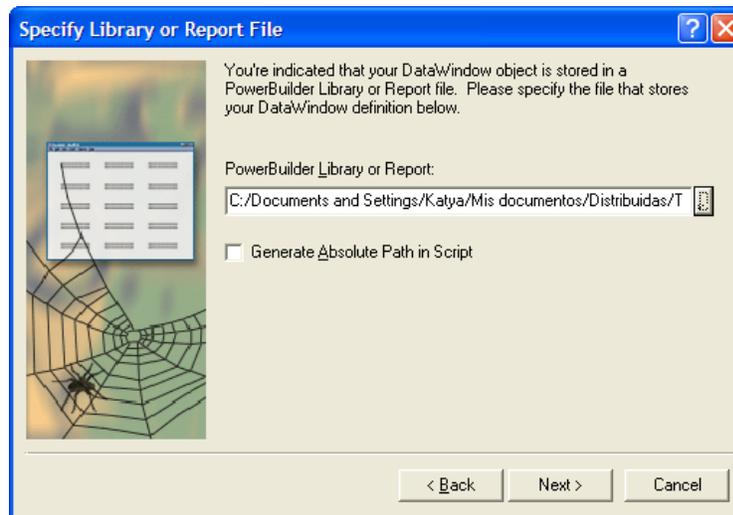


Figura 3.116. Especificación de la Librería para el Web/JSP DataWindowsPager

Para finalizar se debe seleccionar el datawindows, como se ilustra en el Gráfico 3.117.



Figura 3.117. Selección del Data Windows para el Web/JSP DataWindowsPager

Después de seleccionar el Data Windows se despliega una pantalla de información que contiene toda la configuración del DataWindowsPager, se debe hacer click en el Botón Next.

3.7. MANUAL DE USUARIO

3.7.1. Introducción

Las herramientas escogidas para el desarrollo del sistema CISCO – ESPE permitieron aplicar los conceptos y metodologías anteriormente mencionadas, es por esta razón que se aplicó de herramientas para cada fase del ciclo de vida del software, en tal virtud se tiene que:

Para el Análisis y Diseño de Software se escogió Power Designer, herramienta de desarrollo que permite modelar los datos aplicando metodologías Orientadas a Objetos como son OMT y UML.

Para la etapa de codificación, pruebas y mantenimiento del sistema se divide en dos partes: para el desarrollo de la base de datos o Back End, se escoge la herramienta de desarrollo Oracle 9i, que permite tener una base de datos consistente y adecuada, y para el desarrollo de la interfaz del usuario se escoge Power Builder, herramienta de desarrollo que tiene gran compatibilidad con la base de datos descrita anteriormente. La razón para incluir herramientas que también cuenten con la fase de diseño, orientado a objetos, es debido a que el proyecto se propone como la primera parte de un ambiente integrado.

3.7.2. REQUERIMIENTOS TÉCNICOS PARA EL USO DEL SISTEMA

Con la finalidad de obtener un buen funcionamiento y rendimiento del sistema, se deberá cumplir como mínimo con los requerimientos que se detalla a continuación:

3.7.3. Interfaces de usuario

El usuario utilizó como ambiente gráfico el Sistema Operativo Windows XP Profesional y la interfaz gráfica que nos provee la herramienta Power Builder. Todas las aplicaciones funcionan en las estaciones cliente bajo Windows XP Profesional, los cuales posibilitan que el usuario se desenvuelva en un ambiente visual, para el desarrollo se utilizó Power Builder 9.0i el que soporta ambientes gráficos y ofrece una tecnología moderna para poder crear interfaces (GUI) amigables.

3.7.4. Interfaces Hardware

Se recomienda que para el mejor desempeño deberá ser instalado en máquinas Pentium IV, de 128 MB de RAM y por lo menos 20 GB en disco duro, con tarjeta de red.

El Motor de Base de datos debe estar instalado en un servidor de mejores características a las estaciones de trabajo.

3.7.5. Interfaces de software.

El sistema se desarrolla utilizando las siguientes herramientas:

- Diseño en Power Designer 6.1 de 32 Bits
- Desarrollo en Power Builder 9.0
- Windows XP Professional (Servidor)
- Windows Windows XP Professional (Estaciones)
- Oracle 9.i (Servidor de Base de Datos)

- Personal Oracle 9.i (Clientes de Base de Datos)

3.7.6. Interfaces de comunicaciones

Para un mejor rendimiento del sistema, deberá ser instalado mediante una red de datos, la misma que deberá tener como mínimo un Switch, cableado estructurado, conectores RJ45, entre otros.

Los protocolos que deberán emplearse serán TCP/IP o NetBios, además de esto debe indicarse que la conexión hacia la Base de Datos será mediante el protocolo ODBC.

Para la implementación del sistema de Administración Académica y Financiera del Proyecto “La ESPE como Academia Regional Cisco, se requiere de las siguientes especificaciones de Hardware y Software:

3.7.7. AMBIENTE CLIENTE – SERVIDOR

- | | |
|-------------------------------------|---------------------------|
| ✓ Sistema Operativo: | Windows XP Professional |
| ✓ Comunicaciones: | TCP/IP Protocolos |
| ✓ Base de Datos: | ORACLE |
| ✓ Case de modelamiento de Procesos: | Interactúe con Oracle 9.i |
| ✓ Case de modelamiento de Datos: | Interactúe con Oracle 9.i |

- ✓ Software de Desarrollo: Interactúe con Oracle 9.i y (Power Builder)

Tabla 3.9 Ambiente Cliente Servidor

3.7.8. PASOS PARA EL INGRESO DEL SISTEMA

Cabe destacar que el desarrollo del sistema se lo hizo pensando en usuarios con un nivel básico de conocimientos informáticos, de tal forma el sistema esta constituido por interfases gráficas amigables y menús entendibles que despliegan la información a ser ingresada por el usuario.

En tal virtud para empezar con la utilización del sistema, el usuario deberá encender el computador, una vez cargado el sistema operativo Windows XP se procede a dar clic en el botón INICIO ubicado en la parte inferior de la pantalla, se escoge la opción PROGRAMAS en el cual se desplegará un menú de opciones en donde se deberá escoger el nombre del programa Cisco-Espe (SISCISCO), como se puede ver en el Grafico 3.118.

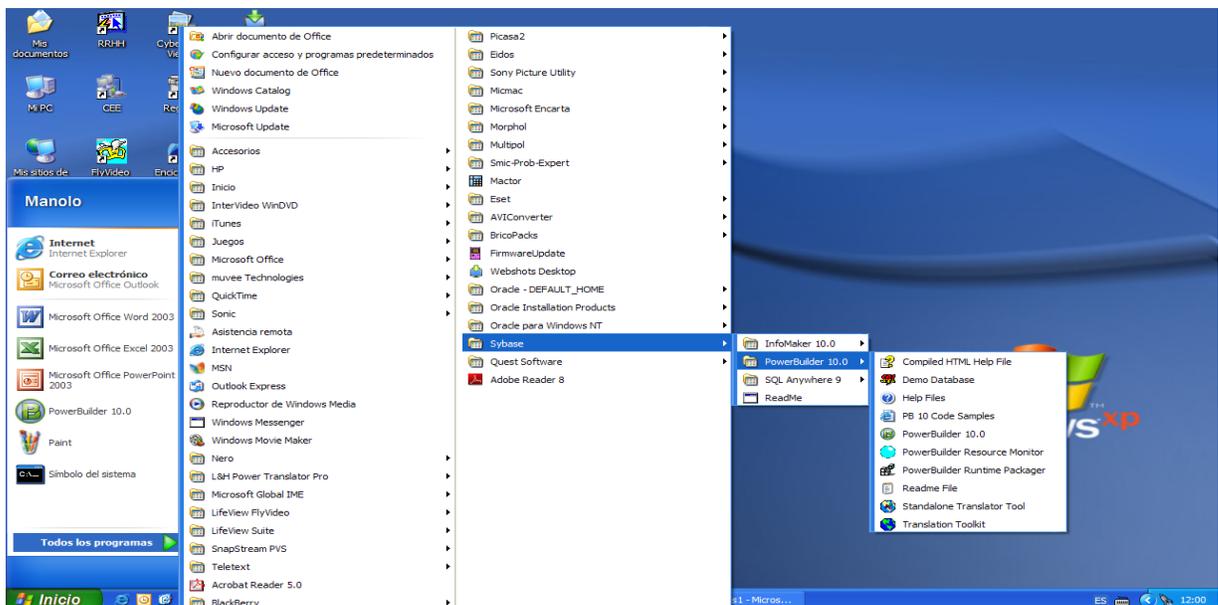


Figura 3.118. Ingreso al Sistema

Al correr el sistema y de acuerdo a las características de procesamiento de la máquina se demora en cargar unos 5 segundos teniendo la siguiente pantalla como se ilustra en el gráfico 3.119.

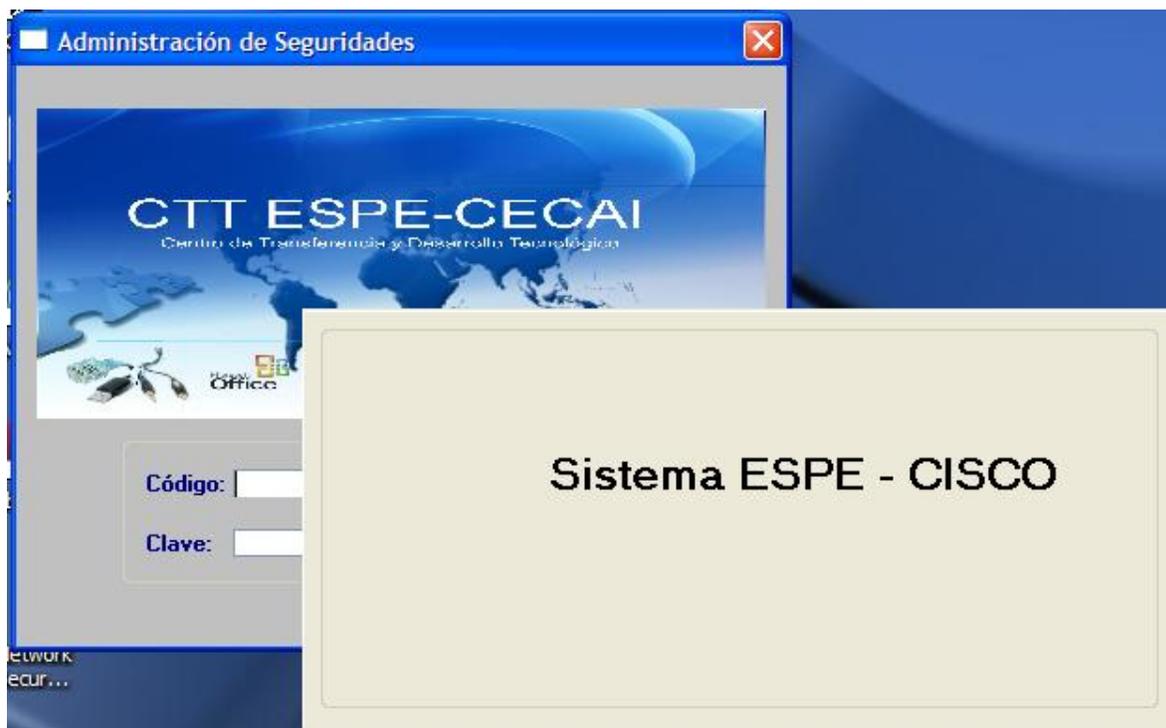


Figura 3.119. Carga del Sistema

Una vez cargado el sistema se encuentra la primera pantalla que corresponde a la seguridad del mismo, esta pantalla consta de código y clave de acceso del usuario, procedemos a ingresar la clave en este caso del Administrador que tiene acceso total del Sistema que es:

Código: Jquerembas

Clave: 10021979

Así tenemos la siguiente pantalla:



Figura 3.120.
Seguridad del Sistema

En este punto es importante mencionar que el Sistema tiene varios Niveles de Acceso que son configurados por el Administrador del sistema que tiene acceso total, de acuerdo a investigaciones con respecto a otros sistemas los niveles de Acceso mas comunes son: Administrador, Supervisor, Usuario o Digitador, estos accesos pueden ser configurados por el administrador como a bien necesiten los usuarios. La clave esta compuesta por un máximo de 8 caracteres alfanuméricos y no debe poseer caracteres especiales, así como tampoco puede tener una combinación de los dos juntos.

Unas ves ingresadas el usuario y la clave de acceso, podemos ver la primera pantalla del sistema, como se puede verificar en el siguiente grafico:

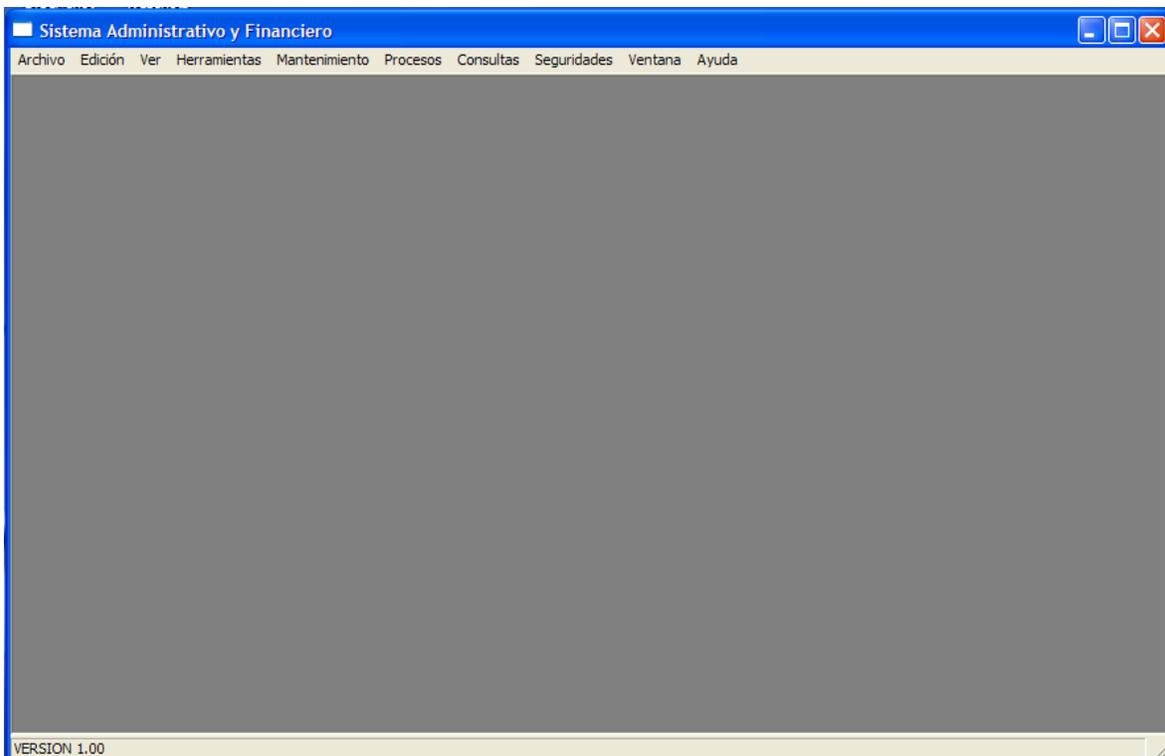


Figura 3.121. Sistema CISCO - ESPE

Como se puede observar en la pantalla inicial del sistema podemos ver el menú principal que consta de:

- Archivo
- Edición
- Ver
- Herramientas
- Mantenimiento
- Procesos
- Consultas
- Seguridades
- Ventana

- Ayuda

Para poder acceder a cada una de estas opciones del menú principal, el usuario con la ayuda del Mouse deberá apuntar a la opción requerida y dar un clic.

En el caso de las 4 primeras opciones del menú principal que son Archivo, Edición, Ver, y Herramientas se activan cuando se usan las operaciones restantes, en tal virtud procedemos a explicar a partir del menú Mantenimiento.

MENU MANTENIMIENTO

Dentro del menú de mantenimiento encontramos las Opciones pertinentes a la administración del sistema como podemos ver en el siguiente gráfico:

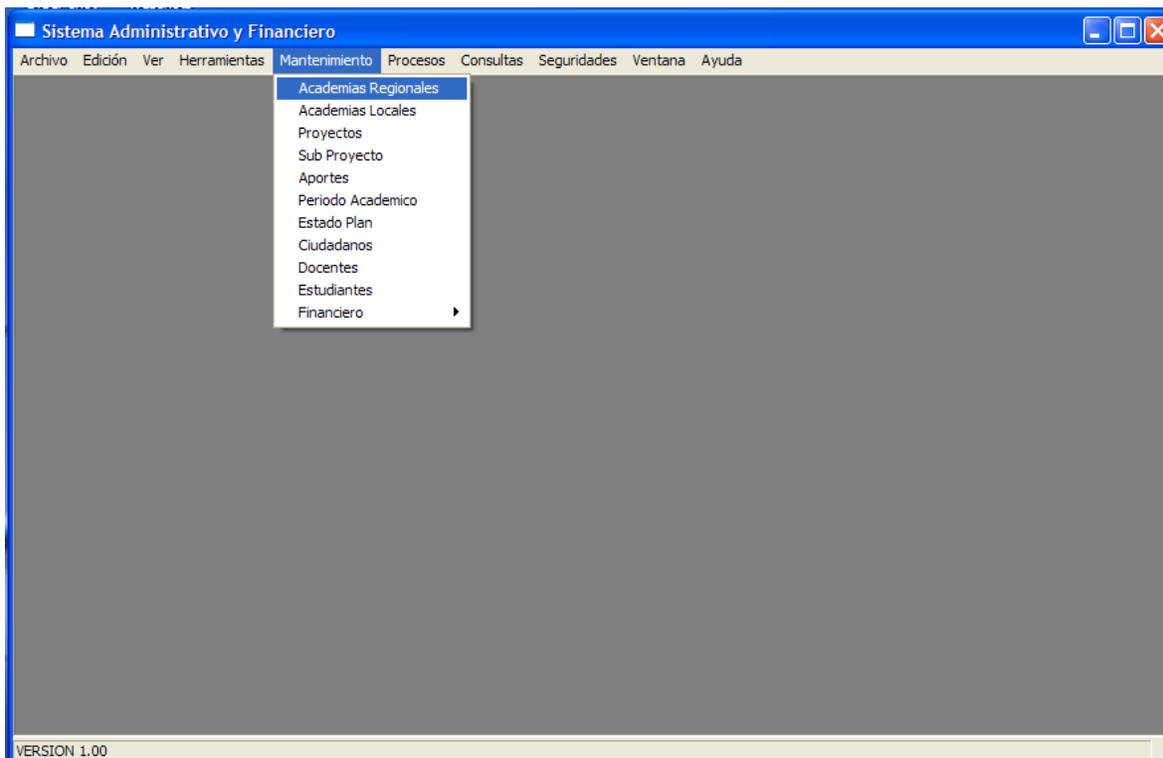
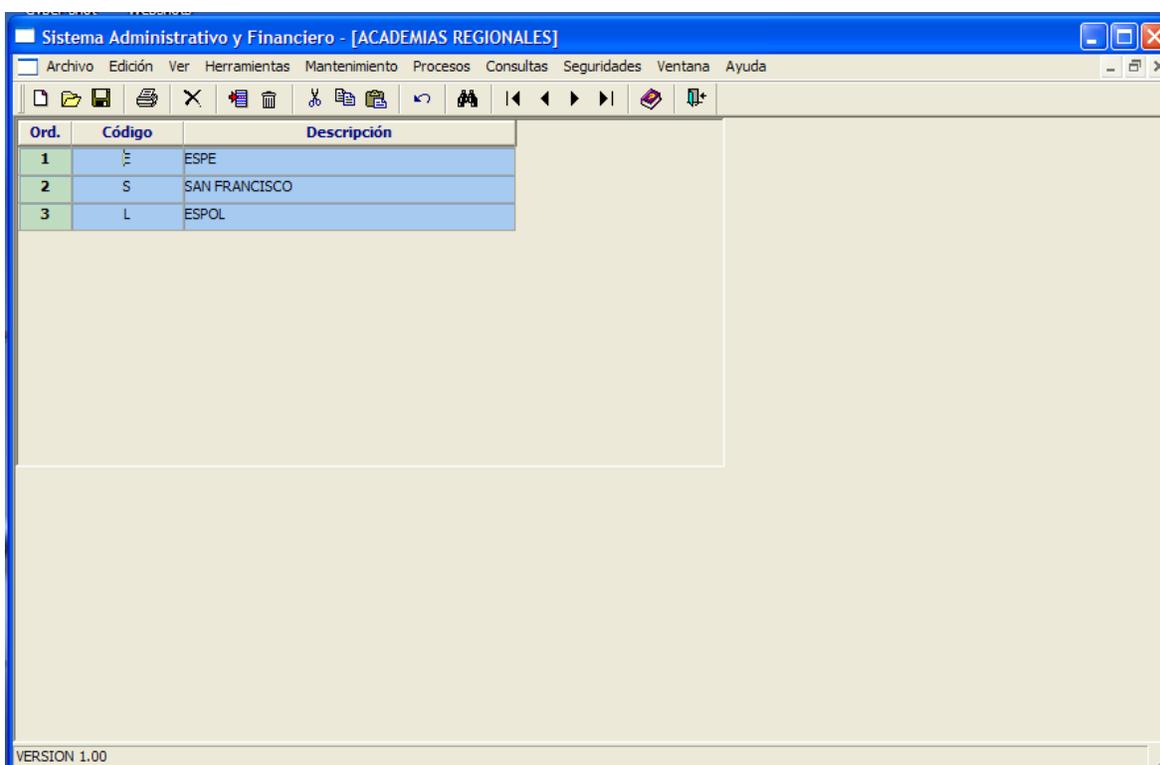


Figura 3.122. Menú de Mantenimiento

Dentro de este menú tenemos la primera opción que es Academias Regionales, esta opción nos permite hacer el ingreso de las diferentes Academias Regionales, cabe destacar que en el proceso de análisis y diseño del sistema se decidió que el mismo sea válido para varias instituciones educativas o bien llamadas Academias Regionales y no sea cerrado solo para la Escuela Politécnica del Ejercito, así podemos ver en el siguiente gráfico para efectos del ejemplo las academias regionales creadas:



Ord.	Código	Descripción
1	ESPE	ESPE
2	S	SAN FRANCISCO
3	L	ESPOL

Figura 3.123. Ingreso de Academias Regionales

Como podemos ver en la pantalla anterior a parte del menú principal se activa automáticamente las opciones de los botones que se describen a continuación, cabe indicar que en las pantallas que se activan trabajan con los registro de dicha pantalla activa, así tenemos:

Botón de Evento	DESCRIPCIÓN DEL EVENTO
	Permite Añadir un nuevo registro a la tabla.
	Permite Abrir un Registro.
	Permite Guardar un Registro.
	Permite Imprimir el registro actual en la tabla activa
	Permite Eliminar el registro actual en la tabla activa
	Permite Insertar una nueva fila para el ingreso de nuevos registros
	Permite Borrar una fila del registro en la tabla activa
	Permite Cortar los valores escogidos de un registro en la tabla activa
	Permite Copiar los valores escogidos de un registro en la tabla activa.
	Permite Pegar los valores Copiados o Cortados de un registro
	Permite deshacer una acción mal ingresada.
	Permite Buscar un registro en la tabla activa
	Permite ubicarse sobre el primer registro de la tabla abierta.
	Permite retroceder al registro anterior en la tabla actual
	Permite avanzar al registro siguiente en la tabla actual
	Permite ubicarse sobre el último registro de la tabla abierta.
	Despliega la ayuda del sistema.
	Permite salir del sistema.

Tabla 3.10. Tabla Descripción de Eventos

Como podemos ver en la descripción anterior cada uno de estos botones tiene su función respectiva pero este depende de lo que en realidad el usuario desea hacer, por ejemplo: se puede crear una Academia Regional de Cisco pero

demostramos ser conscientes que esta creación conlleva a la creación de subprocesos que se establecieron en el análisis y diseño del sistema, dicho de otras palabras los procesos hijos dependen de los procesos padres por ende los procesos padres no se puede borrar del sistema si existen procesos hijos dependientes, por este motivo si el usuario desea borrar una Academia Regional debería empezar borrando todos los procesos hijos que esta tuviere.

De todos modos el sistema se encuentra validado para poder desplegar una alerta al usuario indicando que no se pueden borrar ciertos procesos ya ingresados, así podemos ver en la siguiente pantalla:

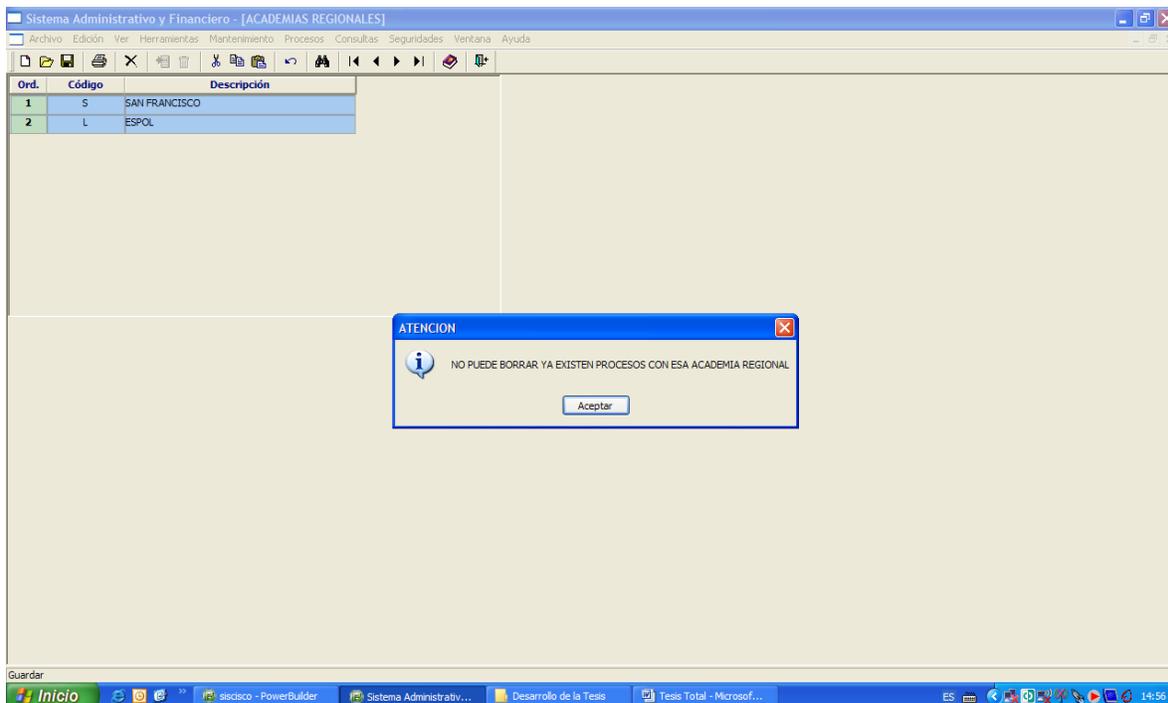


Figura 3.124. Alerta del Sistema

Como podemos observar el sistema nos ha presentado una alerta en donde se indica que no se pueden borrar porque ya existe procesos con esta academia, pero a parte de esta alerta el sistema nos indica que no se ha grabado

la acción de borrado en la base de datos, esto lo podemos ver en la siguiente pantalla.

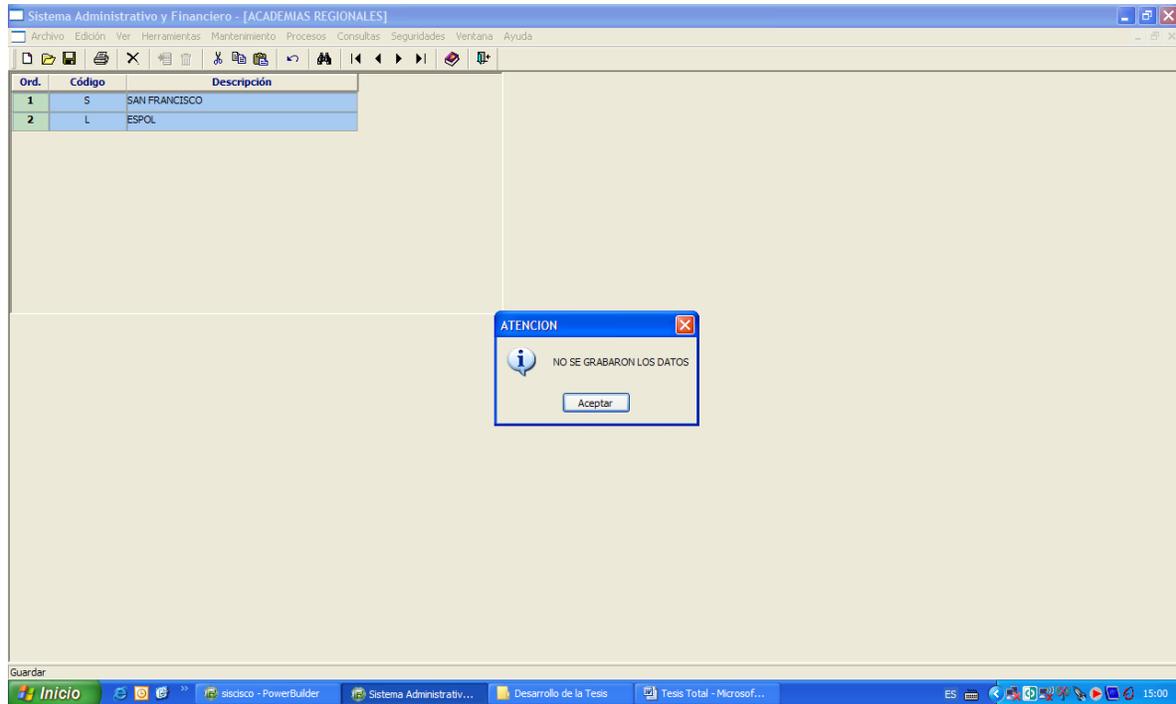


Figura 3.125. Alerta del Sistema

Después que se presentan las dos alertas del sistema el proceso que no se pudo borrar en este caso ESPE vuelve a su estado original, ahora bien en el caso de que se creó una Academia Regional y no se crearon procesos dependientes, esta Academia de ser el caso si se podría borrar como podemos ver en el siguiente ejemplo con la ESPOL, se procederá a borrar y la grabación debería ser exitosa, así tenemos en la siguiente pantalla:

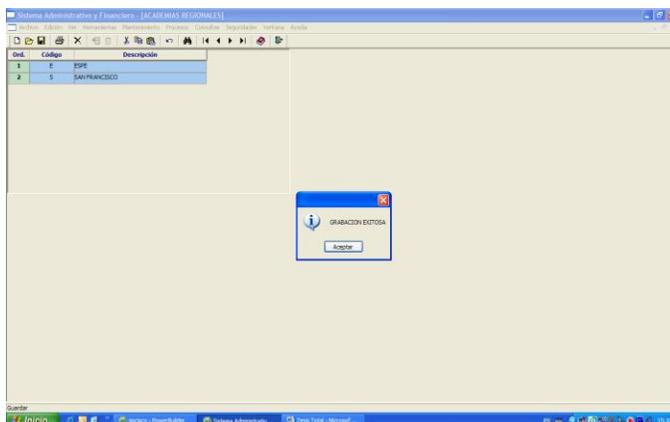


Figura 3.126.
Borrando una Academia Regional

Para la creación de una nueva Academia Regional debemos hacer clic en el botón que nos permite insertar una fila, como vemos en la siguiente pantalla:

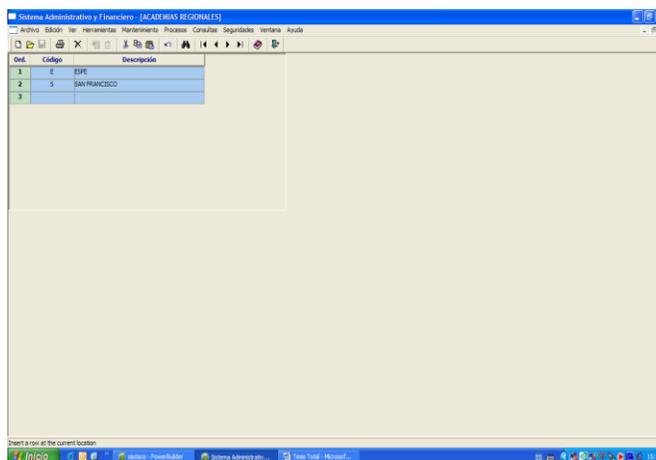


Figura 3.127.
Creación de una Academia Regional

Insertada dicha fila procedemos a ingresar los datos del nuevo registro para efectos de este ejemplo hemos ingresado la Academia Regional ESPOL, así tenemos:

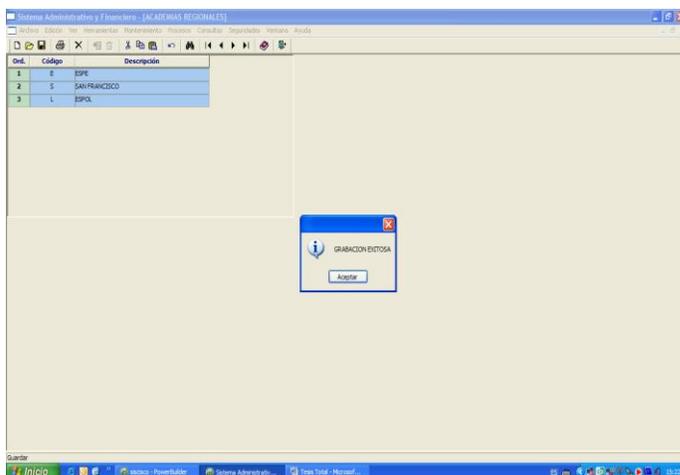


Figura 3.128
Ingreso de la Academia Regional

Realizado el primer paso de ingreso de las Academias Regionales, el paso a seguir es el ingreso de las Academias Locales, para esto damos clic en la opción de mantenimiento del menú principal y escogemos la segunda opción Academias Locales.

Al escoger esta opción se nos despliega la siguiente pantalla en donde antes de hacer el ingreso de la Academia Local nos pide a través de un menú de búsqueda que se escoja la Academia Regional a la cual pertenecen las Academias Locales. Este menú de búsqueda tiene dos opciones a través del código o la descripción de la Academia Regional, así tenemos:

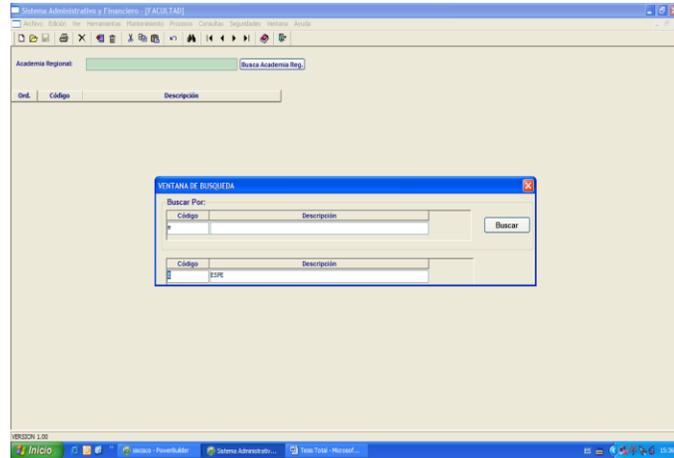


Figura 3.129
Ingreso de Academia Local

En este caso vemos que se ingreso el código de la Academia Regional ESPE, al hacer clic en el botón e búsqueda automáticamente el sistema despliega los datos de la Academia Regional, hacemos clic sobre la Academia Regional y se despliega la siguiente pantalla en donde nos permiten ingresar las Academias Locales, así tenemos:

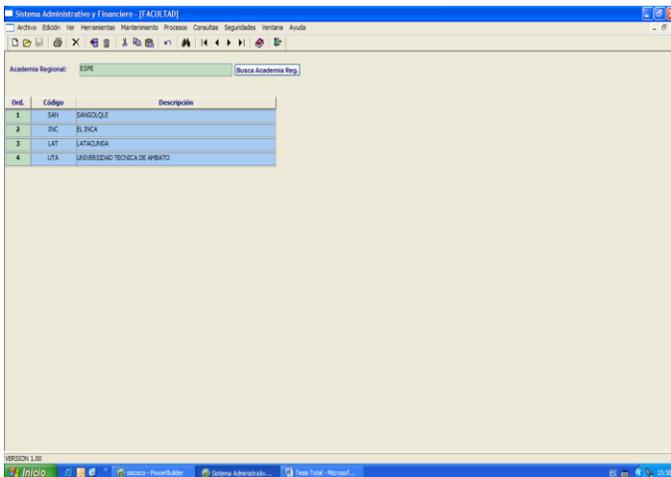


Figura 3.130
Ingreso de Academias Locales

Como se puede observar en la pantalla anterior se ha escogido la Academia Regional ESPE y se ha realizado el ingreso de las Academias Locales en este caso Sangolquí, El Inca, Latacunga, Universidad Técnica de Ambato.

Para el ingreso de una nueva Academia Local realizamos el mismo proceso de la Academia Regional, como podemos observar el sistema tiene interfases amigables y de fácil utilización porque los procesos de ingreso de registros, borrado de registros son los mismos para todas las pantallas activas.

En el caso que el usuario se olvido del nombre y el código de la Academia Regional, en la pantalla de búsqueda sin ingresar el código o el nombre hacemos clic en buscar y nos despliega todas las Academias Regionales ingresadas en el paso anterior, así tenemos:

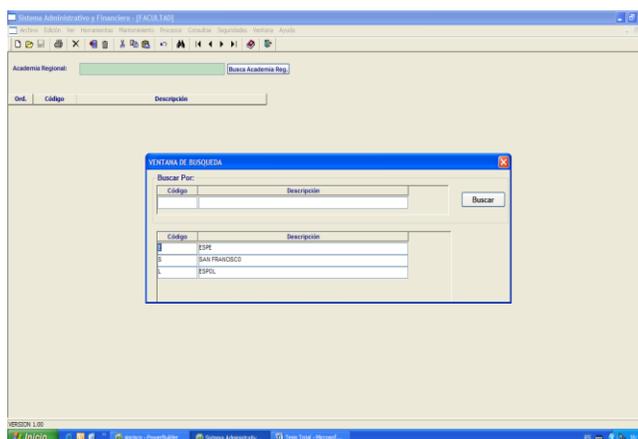


Figura 3.131.
Modo de búsqueda

Hasta este momento hemos ingresado ya las Academias Regionales y sus respectivas Academias Locales, el paso a seguir es el ingreso del proyecto o proyectos de dichas Academias Locales, de igual forma durante el análisis y diseño del sistema se decidió que el sistema funcione para varios proyectos y no

solo para el proyecto CISCO, este es otro factor de valor agregado que se desarrollo en el sistema CISCO – ESPE.

En tal virtud para hacer el ingreso de los Proyectos hacemos clic en la opción de mantenimiento del menú principal y escogemos la opción de Proyectos, así tenemos:

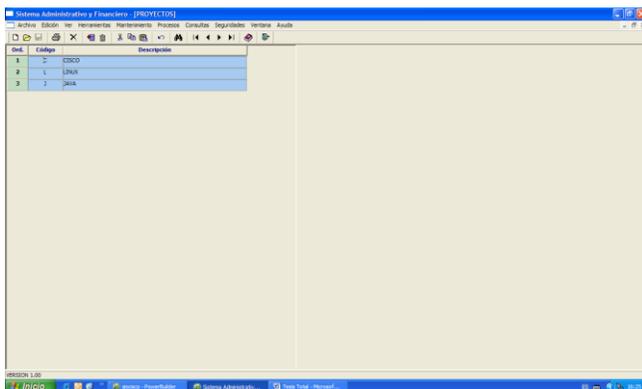


Figura 3.132.
Ingreso de Proyectos

Como se puede observar en la pantalla anterior el ingreso de los Proyectos es igual al ingreso de las Academias Regionales en relación a la funcionalidad de la pantalla, para casos de este ejemplo se han ingresado los proyectos de Cisco, Linux y Java, tomar en cuenta que el menú de funciones actúa de igual forma que las anteriores.

Una vez ingresado los Proyectos debemos ingresar los subproyectos pertenecientes a cada Proyecto, esto depende de las políticas que implante la academia local con respecto a los Proyectos y sub proyectos a impartirse en algún periodo académico.

De esta forma se procede al ingreso de cada sub proyecto, al igual que el ingreso de las Academias Locales en esta pantalla encontramos un botón de búsqueda, este funciona igual que el anterior, así tenemos:

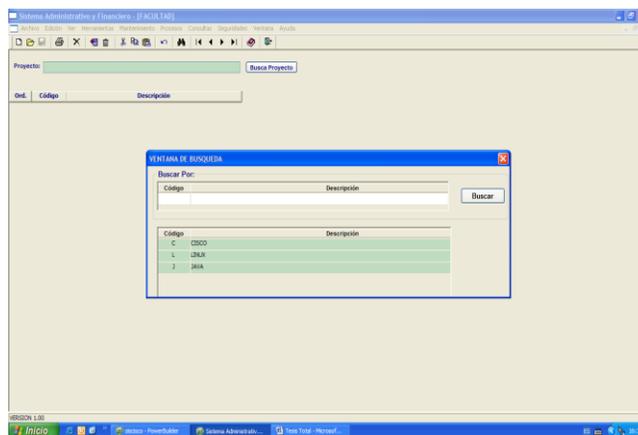


Figura 3.133.
Ingreso de Subproyectos

Para caso de este ejemplo se escogió el proyecto CISCO y se procede con el ingreso de los subproyectos, así tenemos:

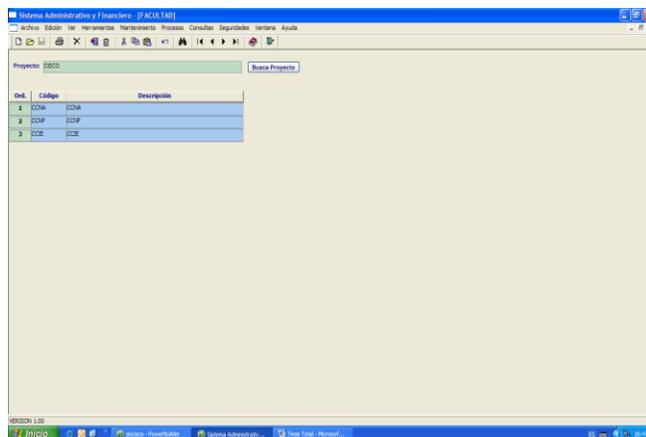


Figura 3.134.
Ingreso de Subproyectos

Una vez ingresado los datos de las Academias Regionales con sus respectivas Academias Locales, Proyectos y Subproyectos, se procede a realizar los ingresos de los aportes académicos, este ingreso dependerá mucho de las políticas implantadas por la Escuela Regional CISCO.

Para el ingreso de los aportes se hace clic en la opción de mantenimiento del menú principal y escogemos la opción aportes, como vemos en la pantalla siguiente:

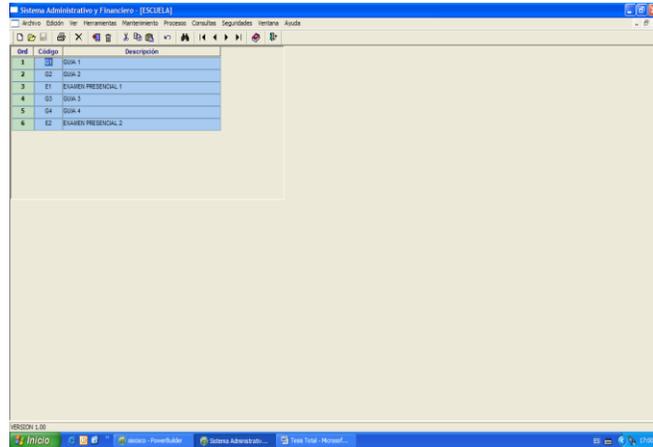


Figura 3.135.
Ingreso de Aportes

Como se puede observar en la pantalla anterior se han realizado el ingreso de varios aportes para efectos del ejemplo, para el ingreso de los aportes se debe tomar el mismo criterio ya adquirido de las pantallas anteriores.

El paso a seguir es ingresar el periodo Académico del Proyecto para poder publicarlo y que el mismo tenga acogida por los diferentes estudiantes, de esta forma tenemos:

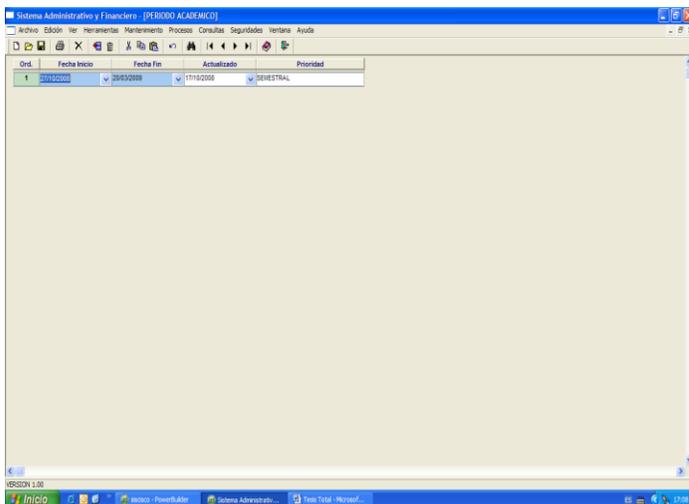


Figura 3.136.
Ingreso Periodo Académico

En base al análisis de los datos para el desarrollo del sistema, el periodo académico debe de estar activo o cancelado, son los dos únicos estados en donde un periodo académico puede estar, para el ingreso de dichos estados lo hacemos en la siguiente pantalla, hacemos clic en la opción de mantenimiento del menú principal y escogemos la opción estado plan, así tenemos la siguiente pantalla:

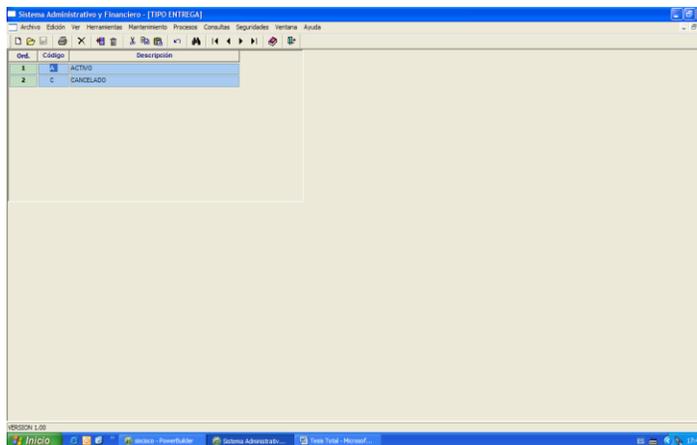


Figura 3.137.
Estado del Plan

Uno de los aspectos importantes dentro del análisis y diseño del sistema fue el incorporar una base de datos facilitada por la Facultad de Sistemas que permitió contar con los alumnos y profesores posibles candidatos a recibir e impartir clases respectivamente en los distintos proyectos, de todos modos esta base de datos dentro del sistema no es una base de datos cerrada, dicho en otras palabras nos permite el ingreso de nuevos alumnos o profesores, tomando en cuenta que los proyectos que se abren en diferentes Academias Locales están orientadas para todo el público y no solo para los alumnos o profesores pertenecientes a dicha Academia.

La base de datos de los alumnos y profesores se la ha llamado Ciudadanos, para el ingreso de un nuevo ciudadano se debe hacer clic en la

opción de Mantenimiento del menú principal, y luego hacer clic en la opción Ciudadanos, obteniendo la siguiente pantalla:

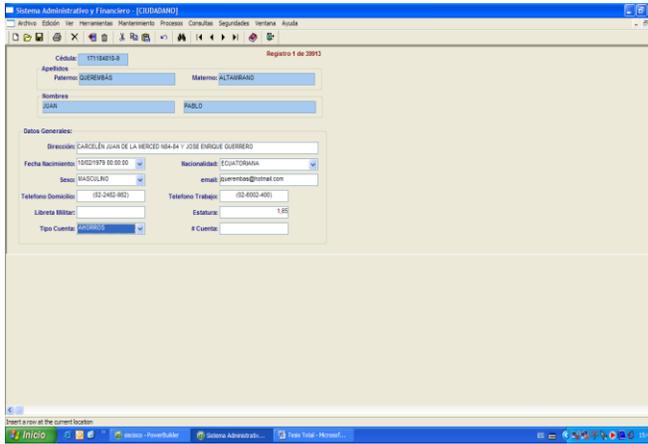


Figura 3.138. Ingreso de Ciudadanos

Al momento se cuenta con un total de registros de 39990 ciudadanos ingresados en esta base de datos, es importante aclarar que dicha base de datos debe estar bien ingresada para poder hacer uso de los datos en los procesos que incorporan alumnos y profesores.

Para el caso del ingreso y el registro de los docentes para un periodo académico, se lo hace de la siguiente forma, hacemos clic en la opción de Mantenimiento del menú principal y escogemos la opción Docentes como se puede ver en la siguiente pantalla:

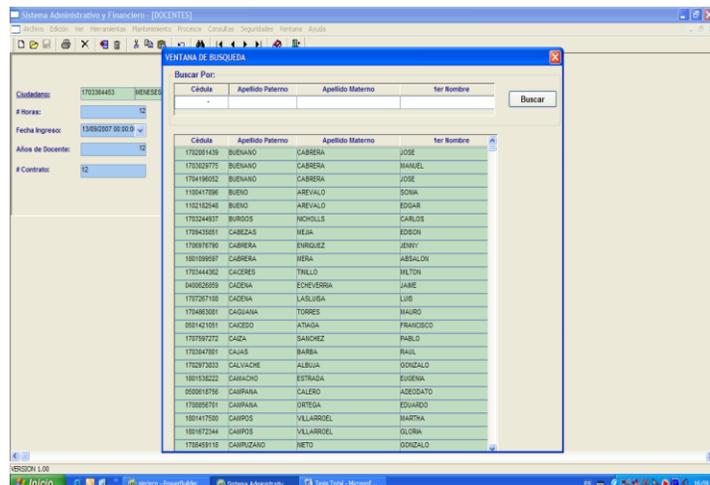


Figura 3.139. Ingreso de Docentes

Como se puede observar en la pantalla anterior el ingreso de los docentes, la palabra Docentes esta vinculado a un proceso de búsqueda, dicho proceso despliega una pantalla con los ciudadanos previamente ingresados, de la misma forma que en ocasiones anteriores esta búsqueda se la puede realizar por: Cédula, Apellido Paterno, Apellido Materno, o Primer Nombre, de no recordar ninguno de estos parámetros se da un clic en el botón Buscar y se despliega toda la pantalla de Ciudadanos.

Una vez escogido el Ciudadano Docente, se procede a ingresar el resto de datos necesarios para el periodo académico siguiente, así tenemos:

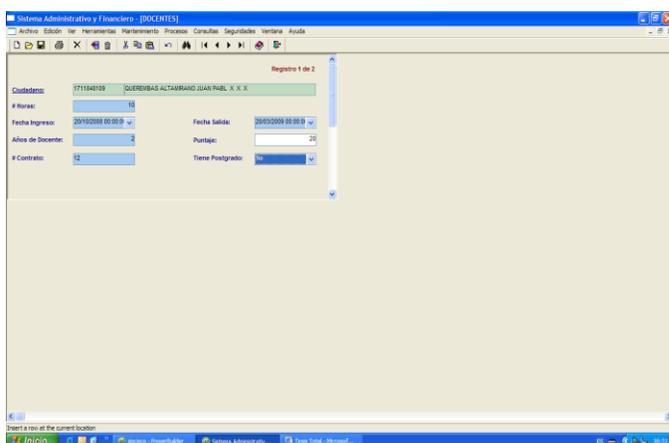


Figura 3.140.
Ingreso de Docentes

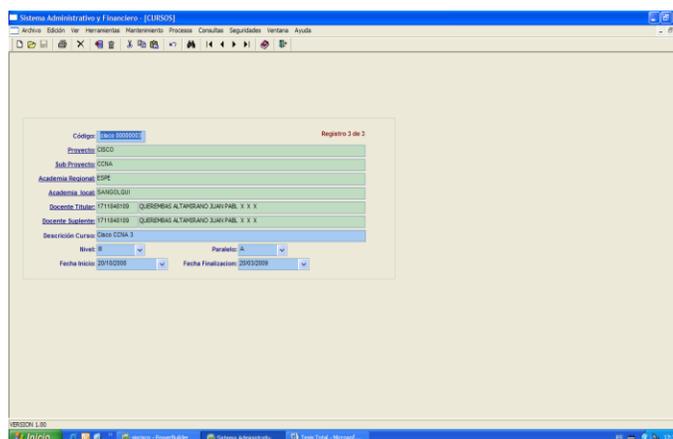
Se procede de igual forma para el ingreso de los Estudiantes.

Dentro de la opción Mantenimiento del Menú Principal, la siguiente opción es Proveedores, al hacer clic en esta opción se nos despliega un menú flotante con las siguientes opciones: Proveedores, Tipo de Cartera, Tipo de Comprobante, Estado de Pago, y Tipo de Pago, es necesario indicar que estas opciones serán

parametrizadas solo una vez a excepción de la opción de proveedores que no es más que una pantalla en donde se hace el ingreso de los proveedores al sistema.

Dentro de la opción de Mantenimiento del menú principal, la opción de proveedores es la última, de tal forma que seguimos con la siguiente opción que es Procesos. La opción de procesos es la que nos va a permitir procesar los datos ingresados en la opción de Mantenimiento, en tal virtud la primera opción de Procesos es Cursos.

Esta opción nos permite ingresar un Curso a ser tomado dentro de un periodo académico, el manejo de esta pantalla es mucho mas sencillo porque se hace a través de procesos de búsqueda, como podemos ver la mayoría de las opciones tienen vínculos que nos llevan a buscar la información requerida por el usuario, de esta forma podemos ver la siguiente pantalla:



Código:	0000000000	Registro 3 de 3
Proceso:	CISCO	
Sub-Proceso:	CCNA	
Academia:	ESPE	
Docente Titular:	1711546159 QUISHUAGUATAMBIANO JUNIPAB, X X X	
Docente Asistente:	1711546159 QUISHUAGUATAMBIANO JUNIPAB, X X X	
Descripción Curso:	Cisco CCNA 3	
Nivel:	3	Paralelo: A
Fecha Inicio:	2019/02/08	Fecha Finalizacion: 2019/02/09

Figura 3.141.
Ingreso de un Curso

Para la asignación de estudiantes a un curso ya creado se debe proceder de la siguiente forma, damos un clic en la opción de Procesos del menú Principal, luego escogemos la opción de Asignación de Estudiantes a Curso, al hacer clic en esta opción tenemos la siguiente pantalla:

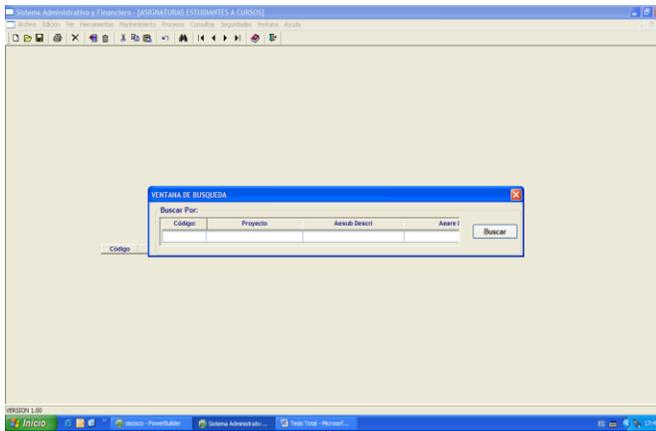


Figura 3.142. Asignación de Alumnos a Curso

Como podemos ver en la pantalla anterior automáticamente nos despliega un proceso de búsqueda, si recordamos los datos del curso lo ingresamos, caso contrario hacemos clic en el botón de búsqueda y se nos despliega los cursos ingresados, a continuación hacemos clic sobre el curso requerido y tenemos la siguiente pantalla:

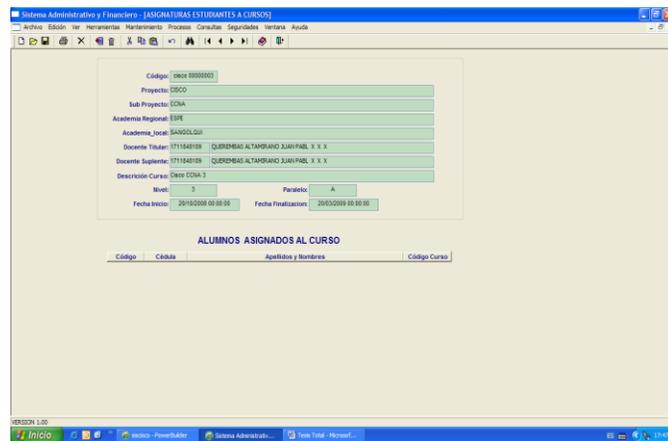


Figura 3.143. Asignación de Alumnos a Curso

Como se puede observar en la pantalla anterior se nos despliega todos los datos del curso que fueron ingresados previamente. A continuación damos clic en el botón que nos permite insertar una fila, acto seguido damos clic derecho con el mouse sobre la fila insertada para que aparezca un menú flotante y escogemos la opción Añadir, como podemos observar en la siguiente pantalla:

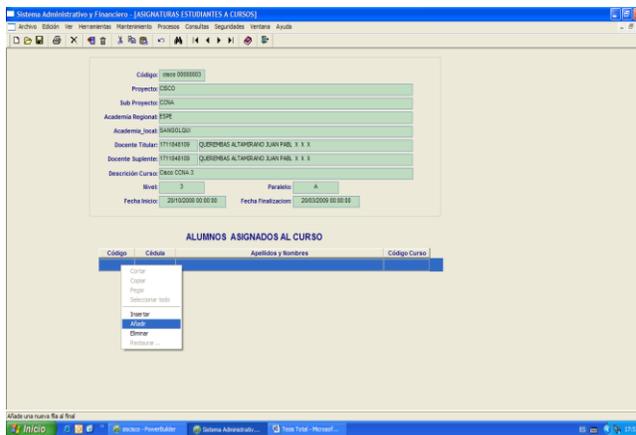


Figura 3.144.
Añadiendo un Nuevo Alumno

Al hacer clic en la opción de Añadir, automáticamente el sistema nos despliega una nuevo menú de búsqueda que al hacer clic en buscar se nos despliega todos los estudiantes previamente ingresados, hacemos clic sobre el estudiante y este automáticamente se añade al curso creado, como podemos ver en la siguiente pantalla:

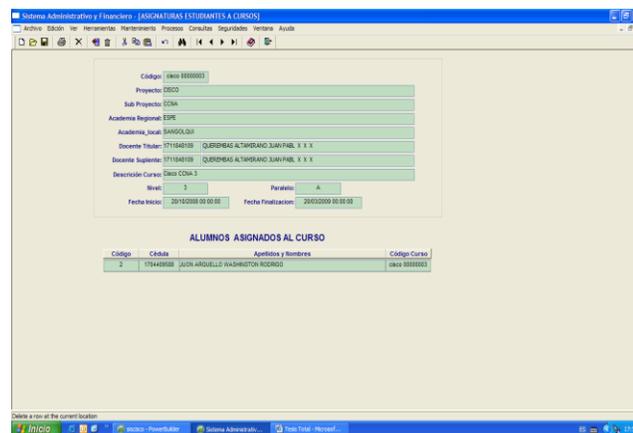


Figura 3.145.
Añadiendo un Estudiante

Por último procedemos a grabar y el estudiante ya pertenece al curso creado dentro del periodo académico indicado.

Dentro del proceso financiero tenemos la opción de cartera, el sistema puede manejar dos tipos de cartera que son las cuentas por pagar (Proveedores)

y las cuentas por cobrar (Estudiantes o Cliente), en tal razón tenemos el registro de estos dos casos en el sistema, así tenemos:

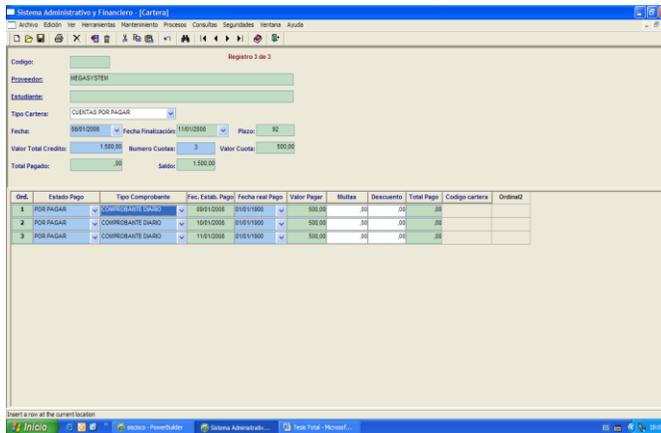


Figura 3.146.
Pago Proveedores

CAPITULO IV

PRUEBAS Y EVALUACION DE RESULTADOS

CAPITULO IV

PRUEBAS Y EVALUACION DE RESULTADOS

Este capítulo viene a constituirse en un corolario de todo lo anteriormente analizado y expuesto, tiene como objetivo el realizar un análisis de resultado tanto en el campo técnico – operativo como en el económico – financiero y de esta forma poder determinar si se ha cumplido con los objetivos planteados al inicio de este proyecto.

4.1 PRUEBAS DE FUNCIONAMIENTO

Debido a que el diseño del presente proyecto se ha basado en gran parte en los resultados de pruebas experimentales se hizo indispensable realizar ensayos de funcionamiento basados en los resultados que el sistema ha arrojado al ingresar datos reales. El procedimiento para la elaboración de dichas pruebas experimentales consiste en el análisis de los datos resultantes de la matriculación del alumno, instructores de la academia, tipos de alumnos, cargas horarias, horarios de matriculación, y su respectiva facturación.

4.2 EVALUACION TECNICA

Durante la realización de las pruebas de funcionamiento se observó que el desempeño del sistema satisface los requerimientos para los que fue diseñado, el comportamiento que mantuvieron los datos al ser procesados durante el periodo de pruebas se ajustan a las especificaciones expuestas al inicio del proyecto en el capítulo 1.

La capacidad mínima de procesamiento de datos del sistema sobrepasa ampliamente las necesidades del CTT (Demanda por obtener un sistema que automatice el proceso de matriculación del proyecto CISCO – ESPE). El rango de operación del diseño para el servidor utiliza como ambiente gráfico el Sistema Operativo Windows 2000 o XP, y la interfaz gráfica que nos provee las herramientas en las cuales el sistema fue desarrollado, como son: Power Builder 9.0 como Front End y como Back End Oracle 9.0i. Todas las aplicaciones funcionan en estaciones clientes bajo Windows 2000 o Windows XP, lo que posibilita que el usuario se desenvuelva bajo un ambiente visual e interfaces amigables.

El sistema ha sido desarrollado con la flexibilidad de adaptarse a los requerimientos mínimos de hardware, es recomendable para el mejor desempeño del mismo ser instalados en máquinas Pentium IV de 128 MB de Ram y por lo menos 20Gb en disco duro con una tarjeta de red 10/100. El motor de la base de datos se encuentra instalado en un servidor de mejores características.

Para un mejor rendimiento del sistema, este fue instalado mediante una red la misma que consta de un Switch, cableado estructurado, conectores RJ 45, protocolos de comunicación como TCP / IP, NetBios, y su respectiva conexión a la Base de Datos mediante el protocolo ODBC, que fue proveída por el CTT.

Finalmente el Sistema de Administración Académica y Financiera del Proyecto “La ESPE como Academia Regional CISCO” esta diseñado para garantizar la correcta asistencia de los alumnos a sus clases en los horarios establecidos, y ha contribuido notablemente en la presentación de información pertinente a las diferentes áreas interrelacionadas cuando sea necesarias, esto es un indicativo de la facilidad de operación del sistema.

4.3 EVALUACION ECONÓMICA

La evaluación económica del presente proyecto se la realiza en base a los costos de desarrollo, que engloban el costo de la mano de obra directa y cargos administrativos, así como otros gastos. Los datos utilizados para esta evaluación han sido proporcionados por el CTT de la Escuela Politécnica del Ejercito. Los costos de desarrollo se pueden dividir en:

- ✓ Costos Directos
- ✓ Costos Indirectos

A su vez los costos directos se divide en:

- ✓ Materiales Directos (Herramientas de Desarrollo)
- ✓ Mano de Obra Directa

4.3.1 Materiales Directos (Herramientas de Desarrollo)

Se entiende por materiales directos a las herramientas de desarrollo que forman parte del proyecto, y cuya cuantificación económica, no reviste mayor dificultad; es así que en concordancia con los diferentes sistemas que constituyen el proyecto se ha hecho una clasificación de las herramientas utilizadas para su desarrollo, las cuales se dividen en dos:

- ✓ Herramientas de Análisis y Diseño
- ✓ Herramientas de Desarrollo

4.3.1.1 Herramientas de Análisis y Diseño

Como se mencionó en el Capítulo I la metodología OMT es orientada a objetos por lo que se necesita herramientas que permitan realizar los diagramas requeridos para lo cual existe Power Designer, que es una herramienta muy

popular en ambientes de Diseño Orientado a Objetos por su soporte a UML (Unified Modeling Language), la cual se utilizó en las fases de análisis y diseño.

4.3.1.2 Herramientas de Desarrollo

Las Herramientas de Desarrollo que se utilizaron para la elaboración del proyecto fueron Oracle 9.0i y Power Builder, estas herramientas son de última generación y proporciona facilidad en el desarrollo de diferentes aplicaciones. Un resumen de estos costos se presentan en el Anexo M.

4.3.2 Mano de obra directa

Se entiende por mano de obra directa aquella que interviene en el desarrollo del Sistema CISCO - ESPE utilizando las diferentes herramientas CASE para convertirlos en subsistemas, y sistemas que una vez ensamblados en armonía, constituyen el proyecto propiamente dicho. Un resumen de estos costos se presentan en el Anexo M.

4.3.3 Costos Indirectos

El otro grupo de costos que forman parte del costo total del producto terminado es el correspondiente a los costos indirectos. Dentro de este grupo se incluyen los materiales indirectos, mano de obra indirecta, energía, servicios, algunas herramientas, costos de dirección, costo de levantamiento de textos, fotocopias. Un resumen de estos costos se presentan en el Anexo M.

4.4 Análisis de Resultados

La calidad de los datos procesados con el Sistema objeto del presente proyecto, compite satisfactoriamente con los sistemas implantados por la ESPE en sus diferentes Cedes, tanto en sus características como en su desempeño. A esto se suma la conveniencia económica que presenta la alternativa de desarrollo frente a la de comprarlos; contribuyendo en forma significativa a la independencia tecnológica que la ESPE a logrado.

CONCLUSIONES Y RECOMENDACIONES

- El desarrollo del Sistema de Administración Académica y Financiera del proyecto Cisco-Espe, constituye una alternativa eficiente tanto técnica como económica, para satisfacer las necesidades de la Escuela Politécnica del Ejército.
- El sistema es la base esencial para el desarrollo de nuevos subsistemas, que pueden integrarse fácilmente por la flexibilidad en su programación orientándolo a nuevos paradigmas de programación orientada a objetos. Esto constituye un tema interesante de investigación y diseño para una futura Tesis en la especialización de Sistemas.
- Es recomendable respaldar la información que el sistema genera en discos externos, para conservarlos listos como partes de reemplazo por alguna eventualidad no deseada.
- La capacidad de producción del sistema se incrementa notablemente cuando es operada por personal familiarizado con el mismo, por lo que se recomienda que sea operado por el mismo personal permanentemente.
- La investigación de sistemas similares implantados en la ESPE fue un factor determinante que contribuyó en el diseño definitivo del sistema; en base a esta experiencia, se recomienda que en proyectos similares (donde no se cuente con la adecuada sustentación teórica y bibliográfica), se considere la posibilidad de investigar experiencias de sistemas similares implantados y en funcionamiento.

- El procesamiento de los datos al que se deben someter, son de gran importancia para determinar las características finales de desempeño de los mismos, por esta razón se recomienda hacer un estudio mas profundo sobre este tema.
- En vista de que el sistema se alimenta de información en línea, se recomienda tener un formato estipulado para cada tipo de estudiante que se inscriba en el curso, este formato puede ser una pre-matricula.
- El punto más difícil en la ejecución de este proyecto, fue el de escoger las herramientas adecuadas de desarrollo y lograr un balance entre los requerimientos técnicos y económicos, ya que cualquier alteración en estos parámetros afecta la rentabilidad del proyecto.

CAPITULO V

BIOGRAFÍA

BIOGRAFÍA

5.1. LIBROS

- KENDALL, E.; KENDALL, J. Análisis y Diseño de Sistemas de Análisis Estructurado. 1.ed. México, Susana Fisher, 1988.
- YOURDON, B. Análisis y Diseño de Sistemas de Análisis Estructurado Moderno. 1.ed. México, MacGraw-Hill, 1990.
- SENN, P. Análisis Estructurado Moderno de Senn. 1.ed. México, MacGraw-Hill, 1989.
- PRESSMAN, R. **Ingeniería Del Software** un enfoque práctico, 3.º edición, Mc Graw Hill, Mexico, 1995

5.2. DOCUMENTOS

- SENDA, DINAIN; Metodología de desarrollo de Sistemas.
- Banco Central del Ecuador; Estándares del Power Builder.
- INFOPAWER; Manuales de Power Builder y Power Designer.
- INEN; Normas I.E.E.E. para especificación de requerimientos.
- INEN; Normas I.E.E.E. para verificación y validación de planes de software.
- INEN; Normas I.E.E.E. para revisiones de software.
- INEN; Normas ISO 9000.
- Oracle, Jdeveloper
- Soft Power S.A., Appeon TM 2.5 for Power Builder, for Windows
- ESCOBAR, L., Compendio De Seguridad Informática Escobar Luis, Sangolquí, 1998. 150p, Facultad de Sistemas, ESPE.

5.3. Enlaces

http://en.wikipedia.org/wiki/Unified_Modeling_Language

http://www.ingenieria.cl/escuelas/informatica/apuntes_curso_uml/ciclo%20de%20desarrollo%20usando%20UML.pdf

<http://login.osirislms.com/offline/uml/>

<http://www.clikear.com/manuales/uml/index.aspx>
<http://www.itox.mx/Comunidad/Librero/umlTotal.pdf>
<http://www.cs.ualberta.ca/~pfiguero/soo/metod/requerimientos.html>
<http://www.geocities.com/txmetsb/req-mgm-2.htm>
<http://www.willydev.net/descargas/articulos/general/cualxpfdgrup.PDF>
<http://www.rena.edu.ve/cuartaEtapa/Informatica/Tema11.html>
<http://www.scribd.com/doc/395783/RUP-etapa-diseno>
<http://www.agapea.com/SQL-Server-2000-n10210i.htm>
<http://www.adrformacion.com/cursos/puntonet/leccion1/tutorial2.html>
http://es.wikipedia.org/wiki/Visual_Basic#Compilador
<http://www.dcc.uchile.cl/~psalinas/uml/introduccion.html>
<http://www.ingenierosoftware.com/analisisydiseno/uml.php>

HOJA DE LEGALIZACIÓN DE FIRMAS

ELABORADO POR

Juan P. Querembás A.

COORDINADOR DE LA CARRERA

Ing. Mauricio Campaña

Sangolquí 10 de Noviembre del 2008