

INDICE DE CONTENIDOS

CAPITULO 1	4
Introducción	4
1.1. RESUMEN	6
1.2. ALCANCE DEL PROYECTO	7
1.3. RESULTADOS Y BENEFICIOS ESPERADOS	9
CAPITULO 2	11
Gestión de la Investigación	11
2.1. LA INNOVACIÓN EN LOS PROCESOS DE GESTIÓN DE INVESTIGACIÓN.	12
2.2. CONOCIMIENTO DE LOS PROCESOS ACTUALES.	12
2.2.1. PLAN ESTRATEGICO DEL CUICYT	13
2.2.2. PLAN INFORMATICO CUICYT	20
PROCESOS SOPORTADOS	37
2.3. ELABORACION DE LA PROPUESTA	39
2.3.1. LOS SISTEMAS DE INFORMACIÓN GERENCIAL.....	41
CAPITULO 3	45
La Plataforma Cliente/Servidor	45
3.1. INTRODUCCION	46
3.2. AMBITO DE LAS ARQUITECTURAS CLIENTE/SERVIDOR	47
3.3. LOS SISTEMAS MANEJADORES DE BASES DE DATOS - DBMS	51
EL HARDWARE	52
3.4. EL MIDDLEWARE.	59
CAPITULO 4	65
Modelamiento Orientado a Objetos	65
4.1 INTRODUCCION	66
4.2. TENDENCIA DE LAS BASES DE DATOS ORIENTADO A OBJETOS	66
4.2.1. REQUISITOS DE LAS NUEVAS BASES DE DATOS ORIENTADAS A OBJETOS.....	67
4.2.2. ESTRUCTURA DE OBJETOS.....	68
I) VARIABLES	68
4.2.3. JERARQUIA DE CLASES.....	71
4.2.4 HERENCIA MULTIPLE	72
4.2.5. CONTENIDO DE OBJETOS	73
4.2.6. ORGANIZACIÓN FISICA.....	74
4.2.7. CONSULTAS ORIENTADAS A OBJETOS.....	74
4.2.8. MODIFICACION DE ESQUEMAS	77
4.3. METRICAS ORIENTADAS A OBJETOS PARA MEDIR SOFTWARE	78
4.3.1. DESARROLLO	79

4.3.1.1. LÍNEAS PARA LAS MÉTRICAS	81
4.3.1.2. REPRESENTACIONES DE LA TERMINOLOGÍA OO	82
4.3.1.3. OBJETO	82
4.3.1.4. OPERACIONES DEFINIDAS	86
4.3.1.5. UNIÓN DE JERARQUÍAS	86
4.3.1.6. PROPIEDADES DE LAS MÉTRICAS A NIVEL DE CLASES	86
4.3.1.7. EL CONJUNTO DE MÉTRICAS CK	88
4.3.2. CONCLUSIONES	90
4.3.3. BENEFICIOS ESPERADOS	91
4.4. MODELAMIENTO UML.....	92
4.4.1. MODELOS PARA CONSTRUCCIÓN DE SOFTWARE ORIENTADO A OBJETOS	95
DESCRIPCIÓN DE UN ESCENARIO	108
CAPITULO 5.....	122
Sistemas Colaborativos.....	122
5.1. INTRODUCCION	123
5.2. DEFINICION	123
5.3. ELEMENTOS Y AREAS DE TRABAJO	123
5.4. GROUPWARE	124
5.4.1. QUE ES EL GROUPWARE?	125
5.4.2 GROUPWARE Y LA REINGENIERÍA	126
5.4.3. LOS COMPONENTES DEL GROUPWARE	127
5.5. EL MODELO ACTION WORKFLOW	127
5.5.1. LA WORKFLOW COALITION	128
5.6. MODELOS CONVERSACIONALES	129
5.6.1 MODELO DE ARGUMENTACIÓN DE IBIS	129
5.6.2 MODELO SISCO	131
5.7. DOMINO Y LOTUS NOTES	135
5.7.1 LOTUS NOTES	135
5.7.2. GROUPWARE (DOMINO).....	135
5.7.3. RESULTADOS ESPERADOS DE LOTUS NOTES Y DOMINO EN LOS QUE SE BASA EL PROYECTO	136
5.7.4. SERVIDORES DOMINO Y CLIENTES LOTUS NOTES.....	136
CAPITULO 6.....	145
Aplicación.....	145
6.1. GESTION FINANCIERA	146
6.1.1. DETERMINACION DEL TIPO DE INFORMACION	146
6.1.2. ANALISIS Y DISEÑO DEL MODELO FINANCIERO	147
6.1.3. MODELO RELACIONAL	147
6.1.4. LA BASE DE DATOS RELACIONAL	149
6.1.5. CODIFICACION CLIENTES ACCESS.....	149
6.2. GESTION ACADEMICA	155
6.2.1. DETERMINACION DEL TIPO DE INFORMACION	155
6.2.2. INSTALACION DEL LOTUS NOTES, DOMINO Y CONFIGURACION	155
6.2.3. CODIFICACION LOTUS NOTES VIA LOTUS-SCRIPT	162
6.3. LISTA DE CONTROL DE ACCESO (LAC)	165

6.3.1. DEFINICION DE USUARIOS Y PERFILES DE USUARIO	165
6.3.2. CREACION DE ROLES	167
6.4. ESQUEMAS LENGUAJE DE MODELAMIENTO UNIFICADO	173
CAPITULO 7	177
Conclusiones, Recomendaciones y Verificación de la Hipótesis	177
7.1. CONCLUSIONES	178
7.2. RECOMENDACIONES	180
7.3. VERIFICACION DE LA HIPOTESIS	181
GLOSARIO DE TERMINOS	182
BIBLIOGRAFIA	184
ANEXOS	186

CAPITULO 1

Introducción



La Constitución de la República, las leyes de la Educación de Universidades y Escuelas Politécnicas, prevén que la **Universidad Ecuatoriana** sea considerada como eje, alrededor del cual se gesta el desarrollo del país, puesto que la educación es la base fundamental para propiciar el desarrollo social, cultural, económico y moral de nuestro país.

Dentro de la Universidad también se establecen interrelaciones entre sus sectores que propician su desarrollo. La Docencia, Investigación y Extensión Universitaria son los tres ejes que mediante programas de integración y estrategias para el desempeño, orientan su futuro. Dependiendo de la eficacia en el manejo de estos tres ejes, la universidad **obtendrá éxito en su gestión** o fracaso.

La Universidad Técnica del Norte buscando mejorar la eficiencia operacional de sus dependencias, ha dado impulso a proyectos que buscan apoyar la gestión universitaria para una mejor toma de decisiones. Entre las principales actividades que contempla la gestión de la investigación podemos señalar las siguientes:

- Promoción, organización, seguimiento e impulso a las actividades científicas en cada una de las Dependencias Administrativas y Facultades.
- Promover, Coordinar y Evaluar los Proyectos de investigación y supervisar los Convenios Interinstitucionales.
- Determinar las normativas organizativas (administrativas y operativas), en donde se especifican las actividades y responsabilidades del personal que intervenga en los proyectos de investigación.
- Buscar convenios con otras instituciones externas para cofinanciamiento de proyectos de interés mutuo y lograr la vinculación con sectores productivos.
- Diseño y búsqueda de instrumentos de evaluación para proyectos.
- Capacitación de grupos multidisciplinarios en Investigación.

La eficiencia y el éxito dependen de la capacidad de responder con rapidez y precisión a los requerimientos de información con una comunicación más directa, metas de la vida económica de las decisiones gerenciales, aplicables a las del Centro Universitario de Investigación Científica y Tecnológica, y que serán apoyadas por el Sistema Gerencial Informático para la **Gestión y Desarrollo** de la Investigación "**GEPRI-UTN**". aumentando la productividad con apoyo y soporte a:

- Participación y trabajo en grupo.
- Control, seguimiento y evaluación.
- Visión e Integración.

1.1. Resumen.



Figura 1.1. Universidad Técnica del Norte – Ciudadela Universitaria.

La Investigación como soporte al desarrollo de la humanidad a través del tiempo, ha generado la tecnología que actualmente conocemos. En el campo de la informática se ha experimentado la evolución de las computadoras y su arquitectura. Inició desde el uso de ábaco, las válvulas al vacío, luego se desarrollaron los transistores y en la actualidad se dispone de los circuitos integrados (CHIPS) a gran escala que optimizan el rendimiento en el ámbito computacional. Todo esto se ha logrado gracias a la investigación.

Uno de los principales objetivos de la Universidad es realizar investigaciones de esta manera contribuye al desarrollo de la ciencia y transferencia de la tecnología. Para que tenga eficacia esta actividad debe realizarse en concordancia con las líneas prioritarias de investigación que definan las principales áreas con problemas de mayor impacto en la sociedad. El tratamiento adecuado de estos **problemas** permitirá obtener resultados orientados a desarrollar el entorno social y productivo.

Para realizar investigación con niveles competitivos, la universidad busca utilizar recursos humanos con alto nivel de preparación académica e investigativa, invierte recursos económicos para equipamiento de laboratorios, base fundamental para realizar esta actividad. El manejo de métodos y recursos orientados a perfeccionar la tarea investigativa necesita de una adecuada

administración, permitiendo hacer más eficiente la gestión para el desarrollo de la Investigación. Esto permitirá a la Universidad obtener un nivel de mayor competitividad.

1.2. Alcance del Proyecto

El CUICYT, órgano gestor de la Investigación Científica y Transferencia de Tecnología conjuntamente con el Departamento de Informática de la Universidad están trabajando en un proyecto de aplicación del software informático a la gestión de ciencia y tecnología.

En este software se prevé manejar, entre otros los siguientes aspectos:

- Sistema de elaboración, presentación y aprobación de proyectos de investigación.
- Disponer en línea las diferentes fases de ejecución de proyectos.
- Sistema de reportes de avance de proyectos.
- Adquisiciones de equipos de laboratorio insumos y materiales.
- Control de gasto del fondo rotativo.
- Inventarios.
- Contrataciones y
- Cumplimiento de objetivos.

En estos aspectos se considera que el control académico y financiero son importantes. Es decir, se está construyendo una herramienta eficaz para el control cualitativo y cuantitativo de los proyectos de investigación.

Para el establecimiento del sistema computacional, se ha realizado el análisis orgánico funcional del CUICYT con su interacción en la UTN, habiendo investigado sobre las funciones que se realizan en centros de investigación de países desarrollados. Se comparó con la visión de futuro del CUICYT y la aplicación real actualizada de las funciones. Con esto se determinaron los puntos prioritarios del Software que deben desarrollarse, comprenden dos grandes fases: Gestión Administrativa y Financiera para la Investigación, y Transferencia de Tecnología, estos se detallan en el gráfico adjunto.

El flujo de información en las diferentes etapas y áreas involucradas en el cumplimiento de las funciones de investigación se ilustran en el gráfico de Procesos en la Gestión de Proyectos de Investigación, que se presenta en este documento en la figura 2.4.

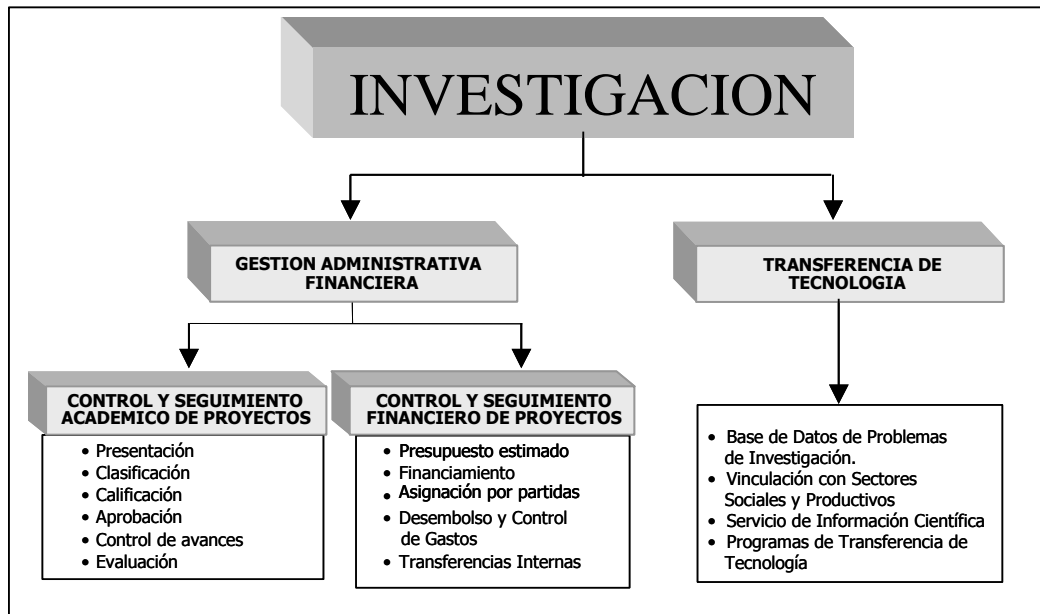


Figura 1.2. Puntos prioritarios del software.

En resumen los puntos prioritarios a investigar en la elaboración del Sistema Informático Gerencial para la Gestión y Desarrollo de la Investigación son:

1) **GESTIÓN ADMINISTRATIVA FINANCIERA**

- a) Control y Seguimiento Académico de Proyectos de Investigación e Inversión en las diferentes líneas prioritarias de Investigación en sus etapas:
 - Presentación
 - Clasificación
 - Calificación
 - Aprobación
 - Control de Avances
 - Evaluación

- b) Control y Seguimiento Financiero de proyectos.
 - Presupuesto estimado
 - Financiamiento
 - Asignación de partidas
 - Desembolso y control de gastos
 - Transferencias internas

2) TRANSFERENCIA DE TECNOLOGÍA

- a) Base de Datos de los problemas de investigación, resultado de los diagnósticos aplicados al entorno.
- b) Vinculación con sectores sociales y productivos
- c) Servicio de información científica
- d) Programas de transferencia de tecnología
- e) Banco de datos con información de Investigadores por áreas de especialización.
- f) Pagina Web CUICYT.
 - Organización
 - Legislación
 - Líneas prioritarias de Investigación
 - Proyectos Académico - Productivos
 - Publicaciones, logros alcanzados (Revistas: Universidad, El Investigador)
 - Eventos realizados y por realizarse
 - Convenios
 - Proyectos que buscan financiamiento
 - Investigadores

A más de fortalecer al Centro de Investigación, con el software aplicado al desarrollo del sistema se busca integrar todas las áreas, departamentos y facultades de la UTN. Se pretende dar apoyo a la participación de todos los **actores** que intervienen en la investigación, se privilegiará el trabajo en grupo, control de las actividades, visión clara de quienes gestionan la Investigación; constituyéndose en una herramienta útil con bases sustentadas en la información confiable y oportuna (información en línea) para una adecuada y oportuna toma de decisiones.

1.3. Resultados y Beneficios Esperados

- * Mayor productividad y eficiencia
- * Reducción de costos en papelería
- * Rastreo de información de proyectos paralizados o en demora.
- * Brindar servicio on-line y direccionar automáticamente los problemas (operativos - administrativos) en la gestión de la investigación.
- * Búsqueda eficiente de información y localización de documentos.
- * Cercanía del usuario a las aplicaciones y datos.
- * Disponibilidad y
- * Potencia de cálculo.

De esta manera se podrá ofrecer una estrategia eficaz y conveniente para encontrar, organizar y compartir información que se necesita para realizar el trabajo diario en la Gestión de la Investigación, teniendo la oportunidad de poder compartir información con el mundo exterior a través de Internet.

CAPITULO 2

Gestión de la Investigación



2.1. La Innovación en los Procesos de Gestión de Investigación.

Sinónimo de Modernización, transformación o reestructuración.

"Persigue aumentar la capacidad de competir en el mercado mediante la reducción de costos "

Se aplica a toda empresa, y en todos sus procesos.

Esta reestructuración está prevista en el Plan Estratégico del CUICYT, en el que se han podido determinar los proyectos y subproyectos que se deberán ejecutar.

METAS QUE PERSIGUE

1. Cambio Radical
2. Mejoramiento de procesos
3. Beneficios significativos, operacionales y administrativos.

PASOS PARA APLICAR LA INNOVACION/REINGENIERIA DE PROCESOS

Para hacer efectiva la innovación, se requiere de un trabajo de análisis orgánico-funcional que empieza desde las bases del departamento, realizar el reconocimiento de funciones y procesos actuales para continuar, un análisis de lo existente, búsqueda de información respecto a técnicas que se utilizan en organizaciones dedicadas a la investigación y similares para empezar por imitar, debiendo en lo posible luego de la innovación, superar las técnicas aplicadas, corrigiendo y optimizando tiempo y recursos, adecuados al medio en el que nos desenvolvemos.

Los pasos que se deben seguir para cumplir con tales objetivos se apegan a los de la reingeniería de procesos, por una parte, a un Plan Informático y por otra, al Plan Estratégico del CUICYT que se viene trabajando paralelamente con el PLANED de la UTN.

Un resumen de lo planificado en la etapa de búsqueda y análisis de información, se presenta a continuación conjuntamente con los documentos con los que se partirá para el análisis, tanto de formatos, como de procedimientos.

2.2. CONOCIMIENTO DE LOS PROCESOS ACTUALES.

En donde se debe involucrar a los dueños del proceso, buscando la participación para llegar a un consenso.

Este estudio se lo realiza haciendo un análisis orgánico funcional, por medio de técnicas como: el análisis FODA (Fortalezas, Oportunidades, Debilidades, Amenazas), PESI (Plan Estratégico de Sistemas de Información).

2.2.1. PLAN ESTRATEGICO DEL CUICYT

CONSIDERACIONES GENERALES

La enseñanza y la investigación comprometidos con los intereses nacionales deben ser la responsabilidad de las universidades del país ya que éstas reflejan las condiciones del desarrollo de la sociedad en la que se encuentran inmersas.

Conocemos que a la Universidad le compete la responsabilidad de realizar investigaciones que constituyan un verdadero aporte al desarrollo de la ciencia y la técnica, lo que contribuirá a la solución de los problemas y necesidades del país. La investigación constituye la actividad central que debe identificar a la Universidad. En efecto en todos los estatutos universitarios del mundo, la investigación es una de las funciones esenciales de los centros de educación superior, junto a la docencia y a la extensión. En el caso de nuestra Universidad, el Artículo primero del Estatuto establece como propósitos fundamentales las tres funciones antes mencionadas.

La razón de este interés radica en que las sociedades subdesarrolladas depositan en manos de la universidad sus mejores esperanzas para salir del atraso y la dependencia científico-tecnológica en que nos abatimos.

La Universidad Técnica del Norte siempre ha reconocido prioritario el desarrollo de la investigación. Desde el inicio mismo de su vida autónoma creó el Centro Universitario de Investigación Científica y Tecnológica, CUICYT, para generar y orientar la investigación en la Institución. Junto a este alto organismo se crean las Coordinaciones de Investigación en cada Facultad, de acuerdo al Estatuto Orgánico y al Reglamento General de la UTN.

PRESENTACION

Para la Universidad Técnica del Norte, la Investigación Científica y Tecnológica constituye, uno de los pilares fundamentales dentro del proceso de formación de profesionales. También es un instrumento muy útil en la búsqueda de alternativas de solución a los problemas prioritarios que aquejan a la sociedad y la región norte del país.

El CUICYT, con su equipo multidisciplinario de profesionales, además de los docentes investigadores, promueve e impulsa el desarrollo de la investigación universitaria en las diferentes áreas del conocimiento humano.

Un reto importante constituye los programas de vinculación entre la UTN y los sectores productivos. La inserción de innovadores procesos tecnológicos, ha permitido a la empresa (privada) abaratar costos de producción y maximizar su rendimiento.

OBJETIVOS GENERALES

- Impulsar el desarrollo de la investigación científica y tecnológica en la UTN para contribuir al desarrollo regional y nacional.
- Promover el conocimiento científico de la realidad regional y nacional, para identificar los problemas prioritarios de los sectores sociales y en especial de los productivos.
- Proponer líneas de investigación dirigidas al tratamiento de los problemas prioritarios que afectan a la vida nacional, de manera que su comprensión y solución converjan a la investigación multidisciplinaria y a la colaboración interinstitucional.
- Promover la más alta formación del recurso humano que desarrolla la investigación, con la finalidad de que se inserte en la problemática de los sectores sociales y productivos y mantenga competitividad en el desarrollo de la investigación y transferencia de tecnología.
- Impulsar la difusión de la información científica y tecnológica hacia los sectores que realizan la investigación en la UTN y hacia los sectores externos que utilizarían los beneficios de la información.
- Fomentar la cooperación de instituciones dedicadas a la investigación científica y transferencia de tecnología, a nivel nacional e internacional para lograr el mejoramiento en los resultados de las investigaciones.

OBJETIVOS ESPECÍFICOS

- Realizar estudios de diagnósticos básicos y aplicados, en los sectores de influencia de la UTN para contribuir a la solución de sus problemas prioritarios.
- Programar y ejecutar investigaciones científicas y tecnológicas para solucionar los problemas de los diferentes sectores sociales y productivos de acuerdo a las potencialidades de la UTN.
- Establecer un programa de formación de investigadores mediante cursos de postgrado a nivel nacional e internacional, en las diferentes especialidades relacionadas con los problemas prioritarios; y capacitación mediante cursos, seminarios, pasantías, y otros.

- Establecer un programa maestro de equipamiento de laboratorios en la UTN, para facilitar la realización de investigaciones y transferencia de tecnología.
- **Actualizar la Normatividad** para el desarrollo de la investigación en la UTN acorde a lineamientos modernos y de proyecciones futuras de la actividad.
- **Establecer un sistema de información científica y tecnológica** que sirva para publicar y difundir los resultados obtenidos de las diferentes investigaciones, los informes sobre la ejecución de los proyectos y toda información científica y tecnológica que requieran los usuarios al interior y exterior de la UTN.
- Establecer un programa de descentralización administrativa, económica y financiera, inicialmente para la transferencia de tecnología y posteriormente para el desarrollo de la investigación en la modalidad autogestionaria, acorde con las nuevas leyes del Estado y de la Universidad ecuatoriana.
- En coordinación con la Facultades de la UTN, integrar a los estudiantes al proceso investigativo mediante la generación de proyectos de acuerdo al pénsum de estudios de cada facultad.
- Promover las relaciones interinstitucionales afines a la investigación, a nivel nacional e internacional, para lograr la transferencia y actualización de conocimientos y métodos de gestión para el desarrollo de la investigación.
- Integrar a docentes, estudiantes y empleados en el proceso investigativo.

MISIÓN CUICYT

Impulsar el desarrollo de la Investigación Científica y Tecnológica como eje articulador entre la docencia, extensión y gestión universitaria; propiciar la adaptación, creación y transferencia de tecnología para contribuir a la solución de los problemas prioritarios de los sectores sociales y productivos y contribuir al desarrollo de la región y del país.

VISION CUICYT

1. El CUICYT pretende constituirse en la vanguardia de la Investigación Científica y Tecnológica; a nivel local, regional y nacional, eje articulador entre la docencia, extensión y gestión universitaria; descentralizado y autogestionario que mediante la adaptación, creación y transferencia de tecnología contribuya a la solución de los problemas prioritarios de los sectores productivos, de la sociedad y el medio ambiente.

2. El CUICYT se propone liderar el desarrollo de la Investigación Científica y Tecnológica en los ámbitos local, regional y nacional como base y núcleo de la transformación de la sociedad, investigación que será proceso de transformación de la sociedad.

LOGROS MÁS IMPORTANTES

Se ha logrado integrar un grupo importante de docentes de las diferentes Facultades a la realización de proyectos de investigación. Pese a las limitaciones económicas, en cuanto a asignaciones presupuestarias, en el CUICYT se están realizando varios proyectos de investigación.

A la presente fecha, el desarrollo de la investigación ha tomado un repunte muy importante, ya que en total se han propuesto 61 proyectos de investigación, de los cuales han culminado 22 proyectos. Se encuentran en ejecución 18 proyectos y cuentan con el financiamiento de la UTN. En espera para la aprobación del Consejo Directivo del CUICYT están 5 proyectos, estos se ejecutarían durante el año 2002.

En forma permanente, se realiza el seguimiento, control, evaluación técnica y administrativa de todos los proyectos que se proponen y ejecutan. Los docentes de las diferentes Facultades de nuestra universidad, han prestado todo el apoyo para la realización de los proyectos de investigación.

En cuanto a CAPACITACION del personal docente y del CUICYT, se han realizado varios cursos, seminarios y talleres en el campo de la investigación, los mismos que han sido impartidos por expertos nacionales y extranjeros, cursos que de alguna manera han hecho que la gente se interese, sobre la difícil tarea de la investigación.

Se ha logrado mantener la publicación del informativo "El Investigador", revista que publica los resultados de la investigaciones y noticias sobre el tema. Existen algunos libros que están siendo editados y se publicarán muy pronto, estos permitirán una mayor difusión de los resultados y servirán para como herramienta para aplicar las soluciones propuestas.

ASPIRACIONES A FUTURO

De acuerdo con las últimas propuestas emanadas del Congreso Nacional, que plantea al Ejecutivo la aprobación de la Ley para la creación de los Centros de Transferencia de Tecnología (CTT) en las universidades ecuatorianas, que coincide plenamente con la necesidad de modernizar la función

investigación orientada a resolver los problemas del sector social, especialmente productivo, se plantean las siguientes aspiraciones:

- Descentralizar administrativa, económica y financieramente a los Centros de Investigación, para dotarles de una autonomía controlada por la Universidad. Este proceso debería ejecutarse por etapas con el objeto de lograr resultados positivos, profesionalizando su función.
- Obtener un banco de datos sobre los problemas más importantes del sector social, y especialmente productivo, para luego de una selección conjunta con los beneficiarios, promover la ejecución de proyectos de investigación y transferencia de tecnología, mediante la formación de equipos multidisciplinarios de investigadores, provenientes de nuestra universidad e instituciones que pudieran colaborar.
- Formar a los investigadores universitarios, propiciando cursos de postgrado en las especialidades de mayor necesidad del entorno inmediato a la Universidad. Capacitar a los investigadores en las técnicas de investigación y temas de especialidad relacionados con las investigaciones, mediante cursos, seminarios, pasantías y otros.
- Lograr el equipamiento de laboratorios del Centro de Investigación y de las Facultades, con el objeto de apoyar la realización de proyectos.
- Actualizar la Normativa Legal, para dotar de procesos ágiles y orientados a la obtención de resultados de alta aplicabilidad y sostenibilidad.

UNIDADES PROPUESTAS

Se ha planteado la creación de unidades dedicadas a una función específica, funciones que han sido resultado de un análisis FODA, estas unidades son:

➤ **INFORMACIÓN CIENTÍFICA Y TECNOLÓGICA**

Para realizar los proyectos de Investigación Científica y Transferencia de Tecnología, es necesario disponer y manejar la información especializada y oportuna. Para ello la Universidad y sus Departamentos y centros como el CUICYT, disponen del servicio de Internet, con lo que se buscará información catalogada por temáticas, bibliografía, hemeroteca y toda la información que usted necesite.

➤ **DESARROLLO DE LA CIENCIA Y TECNOLOGÍA**

Actualmente el CUICYT, dispone de experiencia para el manejo de más de 50 proyectos de investigación. Estos se basan en la identificación de problemas del sector, realización de la propuesta, aprobación, conformación del equipo humano, búsqueda de financiamiento, ejecución y aplicación de las soluciones. Para ello se buscan proyectos con alta pertinencia, aplicabilidad y sostenibilidad.

➤ **FORMACIÓN Y CAPACITACIÓN DE RECURSOS HUMANOS**

EL CUICYT coordina con el Instituto de Postgrado para formar investigadores con estudios de cuarto nivel, realizar seminarios y cursos para perfeccionar la metodología en la investigación.

➤ **VINCULACIÓN UNIVERSIDAD - SECTORES SOCIALES Y PRODUCTIVOS**

Se podrán realizar todos los proyectos de investigación posibles, pero, no servirán para mucho si estos no tienen altos niveles de pertinencia y aplicabilidad, y esto se logrará mediante la vinculación entre la Universidad y los sectores Productivos.

ENTORNOS

1. GESTION DE CIENCIA Y TECNOLOGIA

2. GESTION ADMINISTRATIVA Y FINANCIERA

3. INFORMACION CIENTIFICA (Procesamiento electrónico de datos y Servicios de información científica)

4. LEGISLACION Y NORMAS JURIDICAS

5. CAPACITACION INTERNA Y EXTERNA PARA FORMACION DE INVESTIGADORES.

FORTALEZAS CUICYT

ENTORNOS	FORTALEZAS	PRIORIDADES
1	• Ejecución de Proyectos/historia investigativa.	2
2	• Seguimiento administrativo de proyectos.	2
3	• Disponibilidad de información en línea INTERNET.	1
1	• Alguna generación de conocimiento y de tecnología en la UTN.	2
3	• Alguna divulgación de la Revista El Investigador.	2
4	• Existencia de Políticas para investigación.	1
2	• Cumplimiento del Plan Integral de Desarrollo.	1
2	• Equipo multidisciplinario administrativo CUICYT/generación proyectos.	2

DEBILIDADES CUICYT

ENTORNOS	DEBILIDADES	PRIORIDADES
2	• Limitación del presupuesto planificado de la UTN.	1
2	• Proyectos demorados en ejecución – continuidad.	1
3	• Falta divulgación resultados.	2
2-4	• Falta de incentivos para investigación (Docentes –	2

	Empleados).	
4	• Insuficiente base legal de investigación.	1
5	• Falta capacitación en Investigación -Msc y PHD,	1
2	• Falta búsqueda de financiamiento externo.	1
2	• Falta de cumplimiento y seguimiento de políticas para Investigación.	1
4-5-2	• Ausencia de evaluación técnica de proyectos, para la evaluación /seguimiento.	1
3	• Limitado conocimiento de problemas de Sector Social y Sector Productivo.	1
3	• Falta de información Científica Documentada.	1

AMENAZAS CUICYT

ENTORNOS	AMENAZAS	PRIORIDADES
2	• Falta fluidez en gestión financiera	2
2	• Reducción del presupuesto a la UTN	2
1-2	• Insuficientes laboratorios y regulación de funcionamiento	1
5	• Carencia de investigadores con perfil y compromiso	1
2-4	• Falta de Comisión Técnica para evaluación de proyectos	1
2-3-1	• Falta de confianza y apoyo del sector productivo para la UTN	1
2	• Falta de apoyo institucional para ejecutar proyectos del CUICYT	1

OPORTUNIDADES CUICYT

ENTORNOS	OPORTUNIDADES	PRIORIDADES
1-2	• Necesidad de asesoramiento en el Sector Social - Sector Productivo.	1
1-2	• Oferta de recursos para investigación - Recursos Humanos.	1
1	• Aplicación de resultados de proyectos al Sector Productivo.	2
2-4	• Firma de convenios Internacionales, Nacionales y Locales.	1
2-4	• Ley de Fomento Industrial y Agraria de Imbabura.	2
2	• Equipamiento de algunos laboratorios.	2

2.2.2. PLAN INFORMATICO CUICYT

Un adecuado análisis de los procesos y de la información llevan a entender de mejor manera la realidad actual, para saber hacia donde nos podemos proyectar, para esto es una buena técnica el PLAN ESTRATEGICO DE SISTEMAS DE INFORMACION (PESI).

PERFIL DE LA EMPRESA				
• IDENTIFICACIÓN				
NOMBRE: “UNIVERSIDAD TECNICA DEL NORTE”				
CENTRO DE EDUCACION SUPERIOR.				
OBJETIVOS Y ESTRATEGIAS EMPRESARIALES:				
<p>DESCRIPCION</p> <p>El objetivo es la formación de profesionales competentes, críticos y creativos de alta calidad, acorde con las exigencias, utilizando tecnología y métodos innovadores de enseñanza/aprendizaje, para entregar a la sociedad recurso humano eficaz y competitivo que permita orientar el desarrollo social y productivo del país.</p>				
<p>• LA MISIÓN</p> <p>Contribuir al desarrollo integral de la región, mediante la formación de profesionales técnicos de calidad en los niveles de pre y postgrado, críticos, creativos y humanistas, con valores éticos y cívicos; la generación y adaptación del conocimiento científico y tecnológico; la difusión de valores de nuestra identidad cultural y la interacción con los sectores sociales y comunitarios, a través de la oferta de servicios especializados y de asistencia técnica.</p> <p>“Ciencia y Técnica al Servicio del Pueblo”</p>				
• LOCALIDADES DEL NEGOCIO (Actualizado al 3 de Junio del 2002.)				
SEGMENTO	TIPO DE LOCALIDAD	DESCRIPCION	UBICACIÓN GEOGRAFICA	Nro. EMPLEADOS
RECTORADO	DEPENDENCIA ADMINISTRATIVA	Rectorado	Ibarra – Ciudadela Universitaria “El Olivo”, Edificio Central, Primer Piso	6
VICERRECTORADO ADMINISTRATIVO	DEPENDENCIA ADMINISTRATIVA	Vicerrectorado	Ibarra – Ciudadela Universitaria “El Olivo”, Edificio Central, Primer Piso	11
VICERRECTORADO ACADEMICO	DEPENDENCIA ADMINISTRATIVA	Vicerrectorado	Ibarra – Ciudadela Universitaria “El Olivo”,	2

			Edificio Central, Primer Piso	
PROCURADURÍA GENERAL	DEPENDENCIA ADMINISTRATIVA	Procuraduría General	Ibarra – Ciudadela Universitaria “El Olivo”, Edificio Central, Primer piso	3
AUDITORIA INTERNA	DEPENDENCIA ADMINISTRATIVA	Auditoria Interna	Ibarra – Ciudadela Universitaria “El Olivo”, Edificio Central, Tercer Piso	2
PLANEAMIENTO UNIVERSITARIO	DEPARTAMENTO	Planeamiento Universitario	Ibarra – Ciudadela Universitaria “El Olivo”, Edificio Central, Segundo Piso	4
SECRETARIA GENERAL	DEPARTAMENTO	Secretaria General	Ibarra – Ciudadela Universitaria “El Olivo”, Edificio Central, Primer Piso	2
RELACIONES INTERSTITUCIONALES	DEPARTAMENTO	Direc.Relac.Inter nac.Interinst	Ibarra – Ciudadela Universitaria “El Olivo”, Edificio Central, Primer Piso	1
FINANCIERA	DIRECCION	Departamento Financiero	Ibarra – Ciudadela Universitaria “El Olivo”, Edificio Central, Planta Baja.	4
MANTENIMIENTO	DEPARTAMENTO	Unidad de Mantenimiento	Ibarra – Ciudadela Universitaria “El Olivo”, Edificio Central, Segundo Piso	9
Centro de Computo	DEPARTAMENTO	Centro de Computo UTN	Ibarra – Ciudadela Universitaria “El Olivo”, Edificio Central, Planta Baja.	6
DBU	DEPARTAMENTO	Departamento de Bienestar Universitario	Ibarra – Ciudadela Universitaria “El Olivo”, Edificio Central, Planta Baja.	12
CUICYT	DEPARTAMENTO	Centro Universitario de Investigación Científica y Tecnológica	Ibarra – Ciudadela Universitaria “El Olivo”, Edificio Central, Segundo Piso	6
CUDIC	DEPARTAMENTO	Centro Universitario de Difusión Cultural	Ibarra – Ciudadela Universitaria “El Olivo”, Edificio Central, Tercer Piso.	27
UCPE		Unidad de Capacitación Popular y Extensión	Ibarra – Ciudadela Universitaria “El Olivo”, Edificio Central, Tercer Piso.	7
PERSONAL	JEFATURA	Jefatura de Personal	Ibarra – Ciudadela Universitaria “El Olivo”, Edificio Central, Planta Baja	3
POSGRADO	INSTITUTO	Posgrado	Ibarra – Ciudadela Universitaria “El Olivo”, Edificio Central, Primer Piso.	2
CAI	CENTRO	Centro Académico de Idiomas	Ibarra – Ciudadela Universitaria “El Olivo”,	3

			Edificio Central, Tercer Piso.	
LABORATORIOS		Física, Química, Biología	Ibarra – Sector del Camal,	3
FCCAA	FACULTAD	Facultad de Ciencias Administrativas.	Ibarra – Ciudadela Universitaria “El Olivo”, Edificio Nro.	9
FCCEE	FACULTAD	Facultad de Ciencias de la Educación	Ibarra – Ciudadela Universitaria “El Olivo”, Edificio Nro.	27
FICA	FACULTAD	Facultad de Ingeniería en Ciencias Aplicadas	Ibarra – Ciudadela Universitaria “El Olivo”, Edificio Nro.	15
FICAYA	FACULTAD	Facultad de Ingeniería en Ciencias Agropecuarias y Ambientales.	Ibarra – Ciudadela Universitaria “El Olivo”, Edificio Nro.	19
FCCSS	FACULTAD	Facultad de Ciencias de la Salud	Ibarra – Antiguo Hospital San Vicente de Paúl.	18

Organigrama Estructural de la Universidad Técnica del Norte, actualizado al 3 de Junio del 2002.

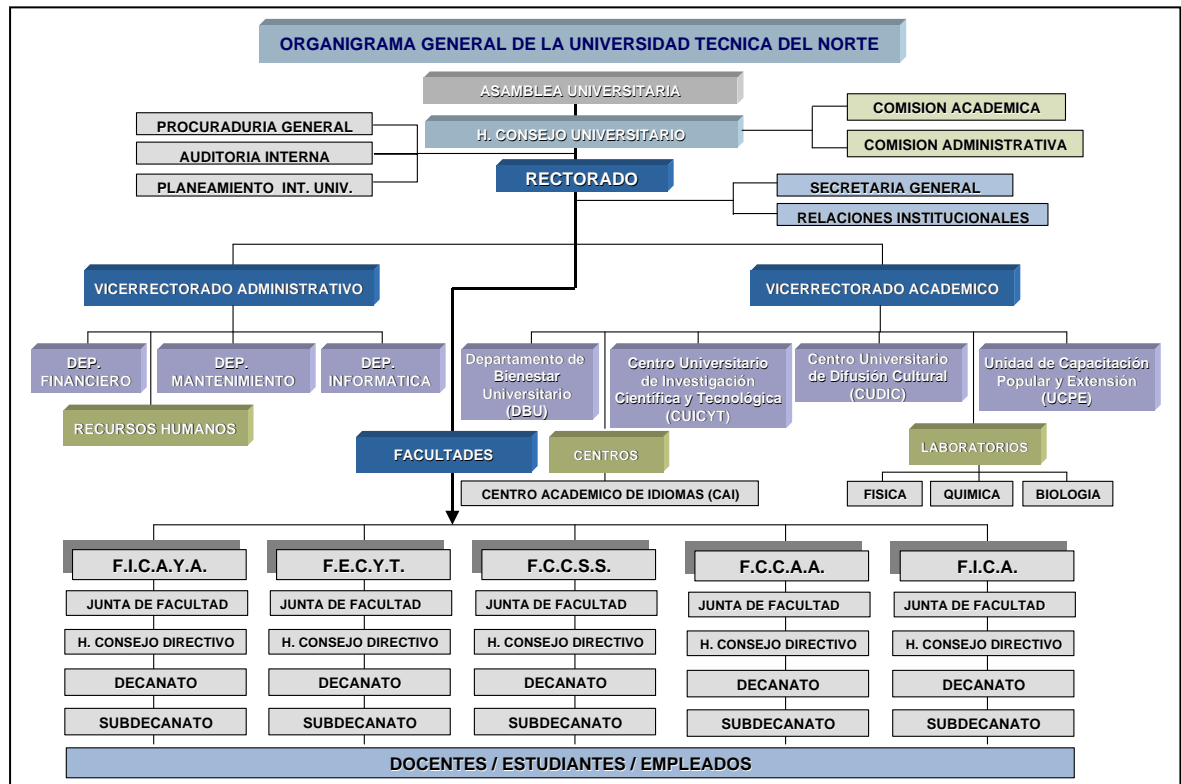


Figura 2.1. ORGANIGRAMA - UTN

PERFIL DEL SEGMENTO	PESI - 02
IDENTIFICACION	
NOMBRE: “ CUICYT ”	
DESCRIPCION: Centro Universitario de Investigación Científica y Tecnológica, organismo encargado de generar y orientar la Investigación.	
OBJETIVOS Y ESTRATEGIAS DEL SEGMENTO	
DESCRIPCION: <ul style="list-style-type: none">➤ Impulsar el desarrollo de la investigación científica y tecnológica en la UTN para contribuir al crecimiento regional y nacional.➤ Promover el conocimiento científico de la realidad regional y nacional, para identificar los problemas prioritarios de los sectores sociales y en especial de los productivos.➤ Proponer líneas de investigación dirigidas al tratamiento de los problemas prioritarios que afectan a la vida nacional, de manera que su comprensión y solución converjan a la investigación multidisciplinaria y a la colaboración interinstitucional.➤ Promover la más alta formación del recurso humano que desarrolla la investigación, con la finalidad de que se inserte en la problemática de los sectores sociales y productivos y mantenga competitividad en el desarrollo de la investigación y transferencia de tecnología.➤ Impulsar la difusión de la información científica y tecnológica hacia los sectores que realizan la investigación en la UTN y hacia los sectores externos que utilizarían los beneficios de la información.➤ Fomentar la cooperación de instituciones dedicadas a la investigación científica y transferencia de tecnología, a nivel nacional e internacional para lograr el mejoramiento en los resultados de las investigaciones.	
APOYO DE LA INFORMATICA	

COMENTARIOS, LIMITACIONES, SUPOSICIONES:

- Actualmente está culminado en más del 95% el proyecto de la red integral de la UTN.
- Se dispone de equipos interconectados físicamente a la red de datos de la UTN.
- Se dispone ya de un acceso directo a las coordinaciones de Investigación de todas las Facultades con la implementación de la red.
- Existe la infraestructura informática, pero no existen las aplicaciones informáticas que permitan integrar a las coordinaciones de investigación.
- No se dispone de un sistema informático para automatizar y optimizar las operaciones técnico administrativas del Centro de Investigación.

IMPORTANCIA ESTRATEGICA DEL SEGMENTO

DESCRIPCION:

Promueve el desarrollo de los sectores sociales y productivos a través de la generación de ciencia y tecnología aplicada a procesos de transferencia tecnológica.

Da impulso a la formación profesional de Investigadores para el desarrollo de Ciencia y Tecnología e Investigación Científica.

Coordinación de cursos de capacitación relacionados a proyectos de Investigación dirigida a Investigadores, Docentes, Estudiantes y Empleados inmersos en la Investigación.

Promueve el equipamiento de laboratorios para desarrollar proyectos, utilizar, adaptar, generar y transferir ciencia y tecnología.

ORGANIGRAMA ORGANICO – FUNCIONAL CUICYT Y ESPACIO FISICO (Actualizado al 3 de Junio del 2002).

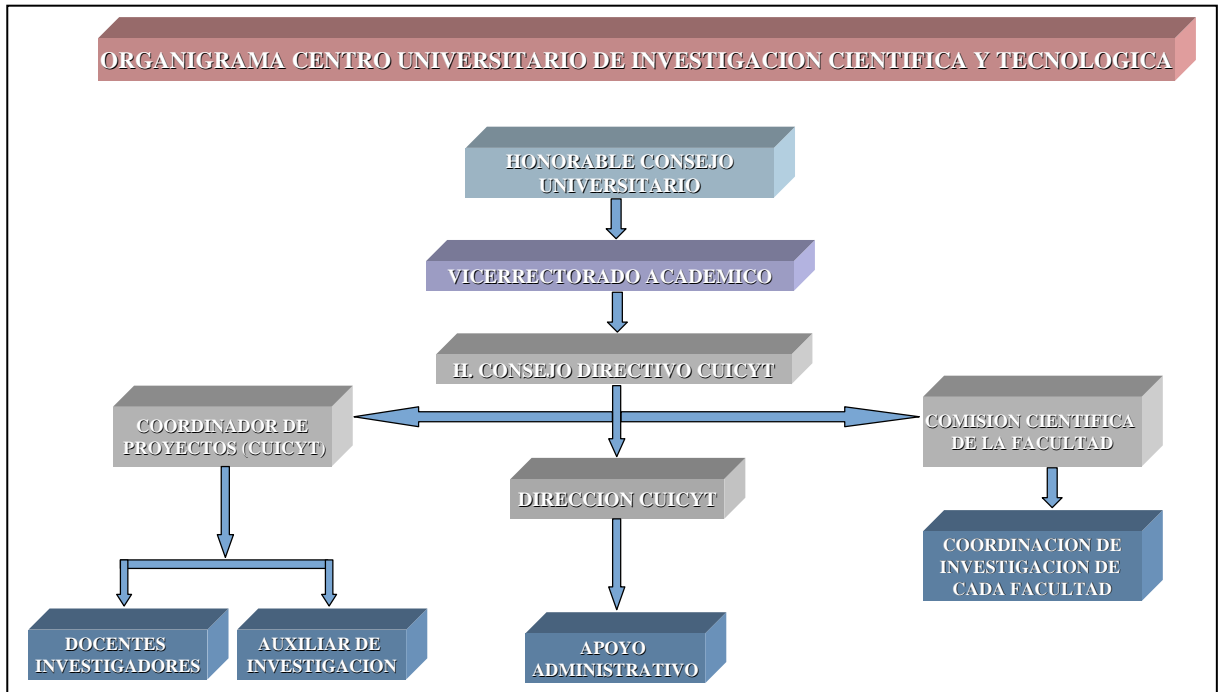


Figura 2.2. ORGANIGRAMA FUNCIONAL CUICYT

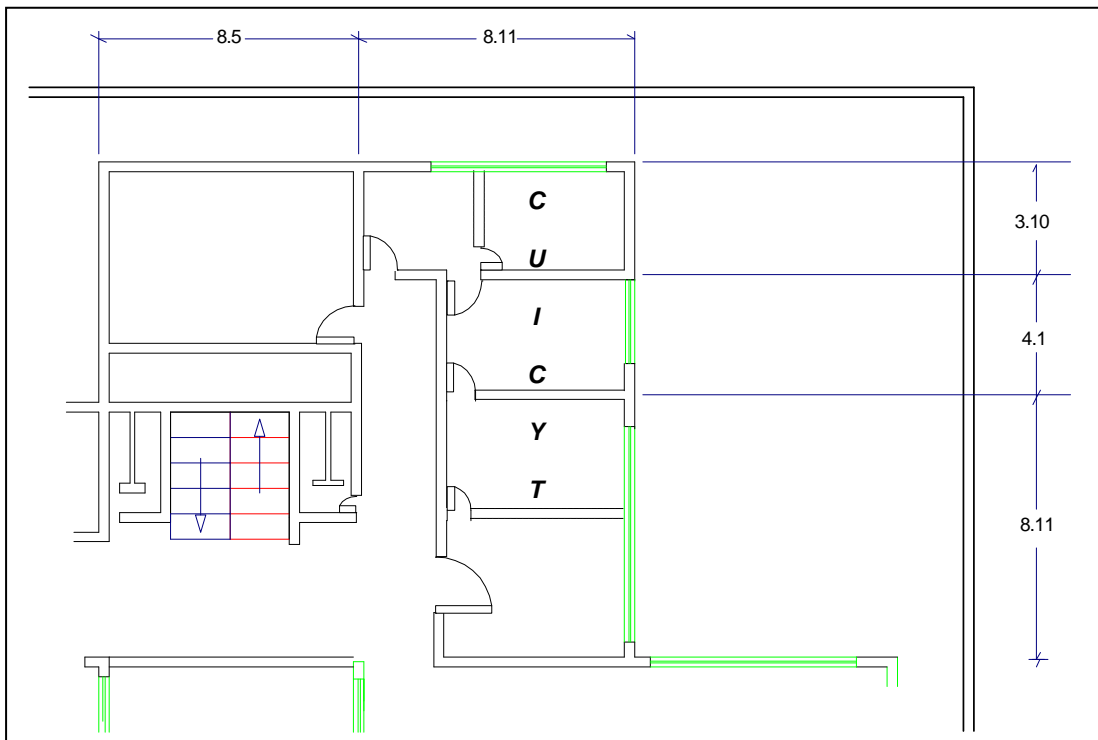


Figura 2.3. Espacio físico del CUICYT en la tercera planta del edificio central de la UTN.

SERVICIOS:		% PRESUP.	% MERCADO LOCAL	POSICION	CRECIMIE NTO
1. Promover la ejecución de proyectos.		40	15	2	Segundo
2. Control administrativo/financiero de proyectos.		30	35	1	Primero
3. Capacitación		10	15	4	Cuarto
4. Servicio de información Técnica-Científica		10	20	3	Tercero
5. Estudios técnico científicos		5	2	6	Sexto
6. Transferencia de Tecnología		5	13	5	Quinto

PERFIL DEL SEGMENTO

PESI – 02a

NOMBRE: “ CUICYT ”

FACTORES AMBIENTALES

COMPETIDORES:

Universidad Católica del Ecuador – Sede IBARRA

DESTINATARIOS:

- La comunidad universitaria.
- Sector social del área de influencia.
- Sector productivo.

FACTORES ESTRATÉGICOS

FORTALEZAS:

- Ejecución de Proyectos/historia investigativa.
- Seguimiento administrativo de proyectos.
- Disponibilidad de información en línea INTERNET.
- Alguna generación de conocimiento y de tecnología en la UTN.
- Divulgación de investigaciones a través de la revista “El Investigador”.
- Existencia de Políticas para el desarrollo de la Investigación.
- Cumplimiento del Plan Integral de Desarrollo de la UTN.
- Se encuentra conformado por un equipo multidisciplinario de investigadores, personal administrativo que ejecuta proyectos.
- Visión para diferentes áreas.
- Experiencia y conocimientos en el diseño de proyectos.

DEBILIDADES:

- Limitación del presupuesto destinado a la investigación en la UTN.
- Proyectos demorados en ejecución – limitación en la continuidad.
- Limitada divulgación resultados de investigación.
- Falta de incentivos para investigación (Docentes).
- Falta actualización de base legal para investigación.
- Ausencia de capacitación a Investigadores – MSc y PhD.
- Falta búsqueda de financiamiento externo.
- Falta de cumplimiento y seguimiento de políticas para Investigación.
- Ausencia de evaluación técnica de proyectos, para la evaluación y seguimiento.
- Limitado conocimiento de problemas del Sector Social y Sector Productivo.
- Falta de información Científica Documentada.
- Falta de equipamiento de laboratorios para investigación.

OPORTUNIDADES:

- Necesidad de asesoramiento en el Sector Social y Sector Productivo.
- Oferta de recursos para investigación - Recursos Humanos.
- Aplicación de resultados de proyectos al Sector Productivo.
- Firma de convenios Internacionales, Nacionales y Locales.
- Ley de Fomento Industrial y Agraria de Imbabura.
- Equipamiento de algunos laboratorios.
- Convenios o financiamiento de FUNDACYT y CONESUP.
- Existe bastante apertura por parte de las autoridades para la realización de proyectos.
- Impulsar o proponer incursiones en proyectos de inversión y productivos.

AMENAZAS:

- La reducción del presupuesto general para todas las Universidades y Escuelas Politécnicas del País.
- El trámite burocrático en el Departamento Financiero.
- Falta fluidez en gestión financiera.
- Reestructuración del presupuesto de la UTN, disminución drástica del presupuesto real para investigación.
- Insuficientes laboratorios y regulación de funcionamiento.
- Carencia de investigadores con perfil y compromiso.
- Falta de Comisión Técnica para evaluación de proyectos.
- Relativo nivel de confianza y apoyo del sector productivo para la UTN.
- Limitado apoyo institucional para ejecutar proyectos del CUICYT.

PERFIL DEL SEGMENTO		PESI – 02b
FACTORES CRITICOS DE ÉXITO		
DESCRIPCION:	CRITERIOS DE CUMPLIMIENTO:	
<ul style="list-style-type: none"> • Realizar investigaciones actuales y propuestas. • Normatividad completa para desarrollo y administración de proyectos. • Investigadores con alto perfil académico y experiencia. • Recursos Económicos. • Laboratorios. • Diagnostico de problemas para investigación y vinculación con Sectores Sociales y Productivos. • Transferencia de Tecnología. • Proyectos Académico – Productivos. 	<ul style="list-style-type: none"> • Número de investigaciones culminadas. • Normatividad aceptable. • Formación del 40% de investigadores con Maestrías y PhD. • Financiar los proyectos de investigación en ejecución, por ejecutar y nuevas propuestas. • Ejecutar un programa de equipamiento de laboratorios para investigación. • La ejecución de proyectos de diagnostico en áreas prioritarias. • Número de proyectos cumplidos. • Instalación y puesta en marcha de proyectos. 	
AREAS DE VENTAJA ESTRATEGICA POTENCIAL:		
AREA:	VENTAJA ESTRATEGICA:	
<ul style="list-style-type: none"> • Investigación • Apertura a Proyectos de Inversión y Producción. 	Flexibilidad y apertura al desarrollo.	
DESCOMPOSICION FUNCIONAL		
FUNCIONES DEL NEGOCIO:	CLASES DE DATOS:	
<ul style="list-style-type: none"> • Consejo Directivo. • Director. • Coordinador de proyectos. • Investigadores. • Apoyo Administrativo. • Coordinador de Investigación. 	Informes financieros - y de gestión de investigación. Realizar a pedidos del Director reuniones periódicas de trabajo, en donde se analiza el avance de proyectos, desarrollo, problemas de proyectos en ejecución. Análisis del departamento y actividades a futuro. Abalzar, aprobación de proyectos.	

PERFIL DE LA FUNCION	PESI – 03
IDENTIFICACION	
NOMBRE: “ Consejo Directivo CUICYT ”	
TIPO:	PRINCIPAL: (x) SOPORTE: ()
DESCRIPCION: Encargado de orientar y regular los proyectos de investigación	
APOYO DE LA INFORMATICA	
COMENTARIOS, LIMITACIONES, SUPOSICIONES: Deberían estar bien asesorados y conocer la tecnología informática, la cual les puede orientar en cuanto a la toma de decisiones.	
PROCESOS DEL NEGOCIO:	
NOMBRE: <ul style="list-style-type: none">• Delinear políticas de investigación científica y tecnológica de la UTN y planificar su desarrollo.• Asesorar al H. Consejo Universitario y autoridades.• Aprobar y evaluar los proyectos.• Calificar, supervisar y evaluar los proyectos.• Promover la formación de proyectos de investigación.• Procurar el logro de fuentes de financiamiento.• Suspende los proyectos por problemas insalvables.	

FUENTE: Estatuto Orgánico de la Universidad Técnica del Norte. 1990. – pag. 25.

PERFIL DE LA FUNCION	PESI – 03
IDENTIFICACION	
NOMBRE: “ La Dirección ”	
TIPO:	PRINCIPAL: (x) SOPORTE: ()
DESCRIPCION: Coordina con el equipo de investigación sobre los proyectos en proceso.	
APOYO DE LA INFORMATICA	
COMENTARIOS, LIMITACIONES, SUPOSICIONES: Limitaciones en cuanto a equipamiento.	
PROCESOS DEL NEGOCIO:	
NOMBRE: <ul style="list-style-type: none">• Cumplir y hacer cumplir las disposiciones legales y reglamentarias.• Presentar el plan anual de trabajo al Consejo Directivo.• Elaborar el proyecto de presupuesto.• Asesorar el trabajo de las Coordinaciones de Investigación de las Facultades.• Buscar financiamiento para los proyectos de investigación.• Mantener relaciones con entidades afines tanto nacionales como extranjeras, promoviendo la firma de contratos y convenios para la ejecución de proyectos de investigación.	

FUENTE: Estatuto Orgánico de la Universidad Técnica del Norte. 1990. – pag. 25.

PERFIL DE LA FUNCION	PESI – 03
IDENTIFICACION	
NOMBRE: “ El Coordinador de Proyectos ”	
TIPO:	PRINCIPAL: (x) SOPORTE: ()
DESCRIPCION: Coordinar las actividades con los Directores de proyectos administrados por el CUICYT.	
APOYO DE LA INFORMATICA	
COMENTARIOS, LIMITACIONES, SUPOSICIONES: Limitaciones en cuanto a equipamiento.	
PROCESOS DEL NEGOCIO:	
NOMBRE: <ul style="list-style-type: none">• Actualizar permanentemente la presentación y formulación de proyectos.• Tramitar a través del director los asuntos administrativos, desembolsos y requerimientos de los proyectos.• Velar por la continuidad y secuencia de las investigaciones.• Colaborar con la elaboración del plan anual y presupuesto del CUICYT.• Buscar vínculos idóneos de investigación.• Llevar un listado de profesores a dedicación exclusiva, tiempo completo y medio tiempo, controlar los plazos establecidos en el Reglamento General, comunicando novedades al director del CIUCYT.	

PERFIL DE LA FUNCION	PESI – 03
IDENTIFICACION	
NOMBRE: “ El Coordinador de Investigación ”	
TIPO:	PRINCIPAL: (x) SOPORTE: ()
DESCRIPCION: Docente encargado de coordinar acciones de gestión de investigación en cada Facultad, con el propósito de integrar a la investigación a Docentes Investigadores y estudiantes.	
APOYO DE LA INFORMATICA	
COMENTARIOS, LIMITACIONES, SUPOSICIONES: Equipamiento adecuado.	
PROCESOS DEL NEGOCIO:	
NOMBRE: <ul style="list-style-type: none">• Contribuir a la elaboración y revisión periódica del plan de investigaciones de la facultad.• Desarrollar el plan de investigaciones, partiendo de la elaboración y ejecución de proyectos prioritarios.• Contribuir al perfeccionamiento y actualización de profesores y estudiantes en el conocimiento de su ámbito y especialmente en las técnicas de investigación, por todos los medios a su alcance.• Apoyar a los programas de extensión universitaria, aportando con los resultados obtenidos en la investigación.• Buscar el asesoramiento de organismos nacionales e internacionales vinculados con el campo de su competencia y mantener las relaciones de interés mutuo y de colaboración conveniente.	

FUENTE: Reglamento General de la Universidad Técnica del Norte. 1991. Pag. 22 – Art. 140.

EL ANÁLISIS DE PROCESOS.

PERFIL DEL PROCESO	PESI – 04
IDENTIFICACION	
NOMBRE: “ Investigación ”	
DESCRIPCION: Razón de ser de la Universidad, en donde se conjugan recursos: humano, científico y tecnológico con un objetivo común, el beneficio de la colectividad.	
PROBLEMAS MÁS IMPORTANTES:	
DESCRIPCION: La crisis por la que atraviesan las Universidades y Escuelas Politécnicas del país en cuanto a la reducción del presupuesto del Estado para la educación superior. La falta de interés y participación directa de los docentes, quienes son los indicados en realizar, incentivar y ejecutar proyectos de investigación.	
APOYO DE LA INFORMATICA	
COMENTARIOS, LIMITACIONES Y SUPOSICIONES: Al momento del análisis situacional cuenta con programas que se utilizan para seguimiento y control de proyectos como: SECOPRE. Sistema para seguimiento presupuestario de proyectos. SACAD. Seguimiento Académico. PRESENTA. Formulación de Proyectos. MANAGER. Manejador de proyectos. Muchos de estos paquetes han caducado, por lo que se ha tenido que efectuar dichos controles, en hoja electrónica y paquetes de utilería.	

CRITERIOS DE EVALUACION:

DESCRIPCION:

Son netamente técnicos, se evalúa en base a resultados, estas evaluaciones se las realiza con personal capacitado en las áreas de desarrollo de dichos proyectos, en caso de no tenerlos en el propio personal son asesorados y evaluados por técnicos especializados conocedores del área.

SUGERENCIAS PARA MEJORAR EL DESEMPEÑO:

DESCRIPCION:

- Aprobar las propuestas de reglamentos y normas para la realización de proyectos de inversión y producción.
- Aumentar las categorías de los docentes que realicen proyectos.
- Dar un reconocimiento al docente realizador del proyecto, con un porcentaje de utilidades del mismo.

CLASES DE DATOS DEL SEGMENTO						PESI – 05
CLAVE: C = CREAR; M = MODIFICAR; U = UTILIZAR						
CLASES DE DATOS PROCESOS	Recopilación de Datos	Clasificación y Selección de Información	Análisis y diseño del Proyecto	Desarrollo de Proyecto	Documentación y pruebas del Sistema	Implantación, Capacitación del personal y entrega
Formulación	CU					
Presentación	U	CU				
Financiación			C	U	U	U
Aprobación		CU	U			
Ejecución				CU	U	U
Seguimiento				CU	U	U
Evaluación			CU	U	U	U
Difusión de resultados					CU	U
Aplicabilidad de resultados						U
ORGANIZACION: UNIVERSIDAD TECNICA DEL NORTE SEGMENTO: CUICYT				FECHA: Febrero del 2001	PAGINA: 1 DE: 1	

APLICACIONES ACTUALES				PESI – 06				
NOMBRE	IMPORTAN-CIA ESTRATE-GICA	OPORTU- NIDAD DE MEJORA	FACILIDAD DE MEJORA	PROCESOS SOPORTADOS				
	1-5	1-5	1-5	1	2	3	4	5
SECOPRE	5	5	1		X			
SACAD	5	5	1			X		
PRESENTA	5	5	1	X				
MANAGER	5	5	1	X		X		
PROCESOS SOPORTADOS								
DESCRIPCION:								
<ol style="list-style-type: none"> 1. FORMULACIÓN y PRESENTACION 2. FINANCIACION, APROBACION 3. SEGUIMIENTO, EVALUACION 								

TECNOLOGIA ACTUAL		PESI – 07					
LOCALIDAD	DESCRIPCION	APLICACIONES SOPORTADAS					
		1	2	3	4	5	6
CIUCYT	Equipos de computación. 1 Equipo en red, Pentium de 32 RAM. Procesador de 244Mhz 1 Equipo en red, Compaq de 128 RAM y Procesador de 800Mhz. 2 Equipos aislados 486 de 20 RAM y 286 de 2 RAM	x	x	x	x		
APLICACIONES ACTUALES SOPORTADAS							
DESCRIPCION:							
<ol style="list-style-type: none"> 1. SECOPRE 2. SACAD 3. PRESENTA 4. MANAGER 							

NOTA:

ALGUNOS DE LOS FORMATOS PARA LA GESTION DE PROYECTOS DE INVESTIGACION RECOLECTADOS SE ENCUENTRAN EN LOS ANEXOS, ADEMAS DE OTROS DOCUMENTOS, COMO EL INSTRUCTIVO PARA EL MANEJO DE LA CUANTIA DE PROYECTOS.

FORMAN PARTE DE LAS GUIAS QUE FUERON UTILIZADAS PARA, LA ADECUADA IMPLEMENTACION DEL APLICATIVO.

2.3. ELABORACION DE LA PROPUESTA

DIAGRAMACION DE PROCESOS.

Esquematisando el flujo de información y el flujo de trabajo, en donde se tomará en consideración: Tiempos de los procesos, tiempos de operación, tiempos de espera por otros procesos (**TEOP**), tiempos muertos (**TM**).

Nro	Proceso	Actor del proceso	Tiempo actual	Tiempo propuesto	TEOP, Tiempo del proceso	TM, Tiempo del proceso	Tipo proceso actual	Tipo proceso propuesto
1	Formulación de Proyecto	Investigador	Definido por el Investigador	No determinado			Manual	Manual/Automatizado
2	Revisión Proyecto	Director - CUICYT	Definido por el Director	7 días			Manual	Manual/Automatizado
3	Aprobación Proyecto	Consejo Directivo CUICYT	Definido por el Vicerrector	7 días	1,2		Manual	Manual/Automatizado
4	Ejecución Proyecto	Investigador	Definido por Investigador Cronograma	Definido por Investigador Cronograma	3		No definido	No definido
5	Elaboración Presupuesto	Director CUICYT, Supervisor Financiero	2-5 días	8 horas	3	4	Manual	Manual/Automatizado
6	Generar Autorización Gasto	Coordinador proyecto	1 día	2 horas	3,5	4	Manual	Manual/Automatizado
7	Generar Pago Autorización de Gasto	Tesorero	1 día	5 minutos	6		Manual	Manual/Automatizado
8	Uso Autorización de Gasto	Investigador	1 mes	30 días			Manual	Manual/Automatizado
9	Registro de Bienes	Coordinador Proyecto	2 semanas	2 horas	7,8		Manual	Manual/Automatizado
10	Cierres Período para Proyecto	Supervisor Financiero	1 semana	3 horas	4,5,8		Manual	Manual/Automatizado
11	Transferencia Interna	Coordinador Proyecto	1 día	10 minutos	4,5		Manual	Manual/Automatizado
12	Registro informe trimestral	Coordinador Proyecto	2 días	30 minutos			Manual	Manual/Automatizado
13	Registro informe final	Coordinador Proyecto	5 días	1 hora	12		Manual	Manual/Automatizado
14	Registrar Proyecto Finalizado	Coordinador Proyecto	5 días	2 horas	4,12,13		Manual	Manual/Automatizado

2.3.1. LOS SISTEMAS DE INFORMACIÓN GERENCIAL

DEFINICIÓN

“ Es un sistema integrado usuario-máquina para proveer información que apoye las operaciones, la administración y las funciones de toma de decisiones en una empresa” (SISTEMAS DE INFORMACIÓN GERENCIAL, Gordon Davis - Margarethe H. Olson).

Los elementos que utiliza son:

- Hardware y Software.
- Procedimientos manuales.
- Modelos para el análisis, planeación, control y toma de decisiones y una base de datos.

Un sistema de información gerencial podría describirse como una estructura piramidal Figura 2.5, cuyos niveles se describen, y se hace referencia a los subsistemas de la aplicación informática de este proyecto.

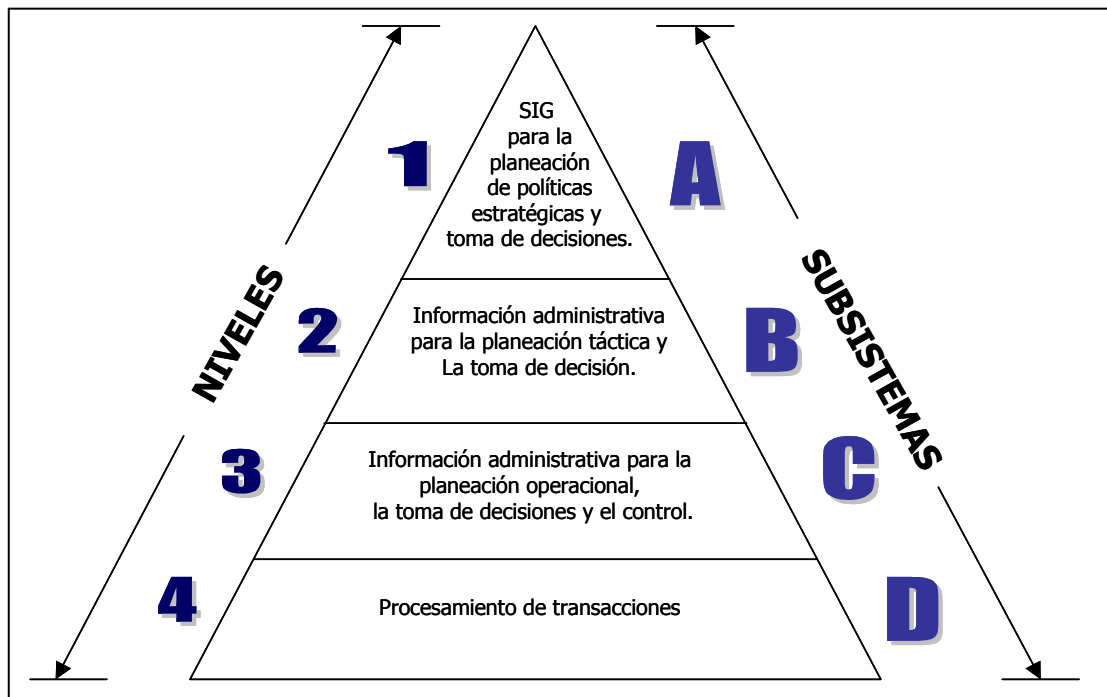


Figura 2.5. Niveles en un sistema de información gerencial.

DESCRIPCIÓN DE NIVELES

Nivel 4._ Comprende información relacionada con el procesamiento de las transacciones, preguntas sobre su estado, etc.;

- Nivel 3._** Comprende los recursos de información para apoyar las operaciones diarias de control.
- Nivel 2._** Agrupa los recursos del sistema de información para ayudar a la planeación táctica y la toma de decisiones relacionadas con el control administrativo.
- Nivel 1._** Comprende los recursos de información necesarios para apoyar la planeación estratégica y la definición de políticas en los niveles más altos de la administración.

Cada nivel del procesamiento de información puede utilizar datos suministrados por los niveles más bajos, es decir son alimentados de información, por lo que la base de la generación de información está en los niveles inferiores operacionales y transaccionales.

LOS SUBSISTEMAS DE UN SISTEMA DE INFORMACIÓN GERENCIAL

El enfoque puede orientarse de dos maneras: de acuerdo con las funciones de la organización que soportan y de acuerdo con las actividades gerenciales.

Este enfoque resulta el más adecuado para el análisis de los subsistemas de "SIGI", con lo que se determinará las áreas involucradas, y actividades que desempeña el personal del CUICYT-UTN.

Los subsistemas de la función organizacional

MACROFUNCIONES DE LA UTN

1. DOCENCIA
2. INVESTIGACION
3. EXTENSION UNIVERSITARIA
4. BIENESTAR UNIVERSITARIO
5. GESTION

En el análisis funcional presentado en los PESI, y resultados de una análisis FODA, se determinaron las siguientes funciones del CUICYT:

FUNCIONES DEL CUICYT

1. GESTION Y DESARROLLO DE CIENCIA Y TECNOLOGIA
2. GESTION ADMINISTRATIVA Y FINANCIERA
3. INFORMACION CIENTIFICA Y TECNOLOGICA (Procesamiento electrónico de datos y Servicios de información científica)

4. CAPACITACION INTERNA Y EXTERNA DE RECURSOS HUMANOS PARA FORMACION DE INVESTIGADORES.
5. VINCULACIÓN UNIVERSIDAD - SECTORES SOCIALES Y PRODUCTIVOS

Los subsistemas de actividades

Las actividades se muestran como estándares para todo tipo de empresa y son las siguientes:

- Procesamiento de transacciones
- Control Operacional
- Control Administrativo
- Planeación estratégica

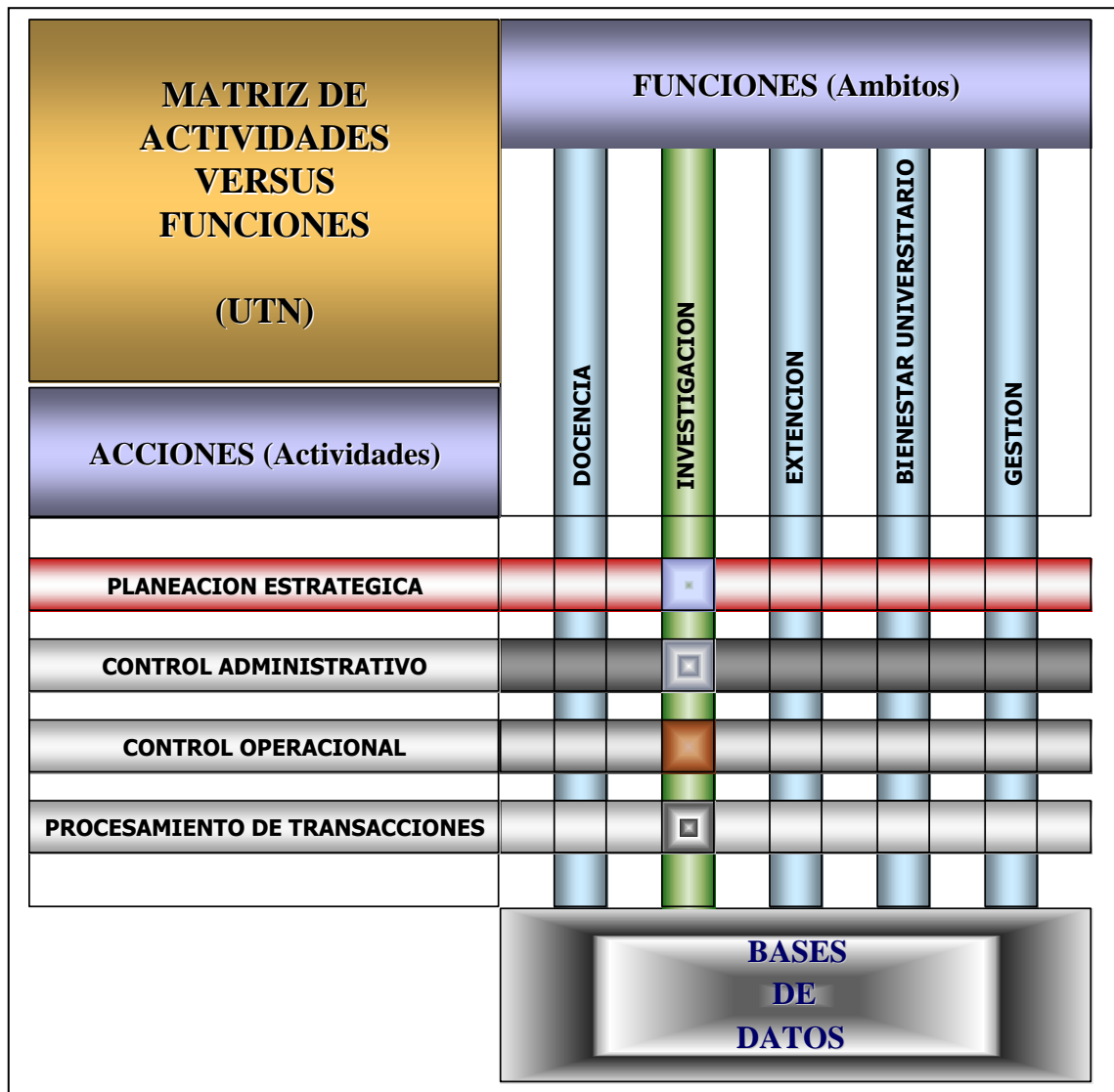


Figura 2.6. Matriz de actividades versus funciones UTN.

PROCESOS CUICYT

1. Promoción y desarrollo de Investigaciones
2. Transferencia de tecnología (Ejecución de proyectos de Transferencia de tecnología)
3. Seguimiento administrativo (Control Económico Financiero y Recurso Humano)
4. Difusión de Información Científica Tecnológica, (Programa radial, Internet, Bases de datos, etc.)
5. Control y seguimiento de Patentes y Propiedad Intelectual.
6. Control Académico de Proyectos.
 - Asesorar al H. Consejo Universitario y autoridades.
 - Aprobar y evaluar los proyectos.
 - Calificar, supervisar y evaluar los proyectos.
 - Promover la formación de investigación.
 - Procurar el logro de fuentes de financiamiento.
 - Suspender los proyectos por problemas insalvables.

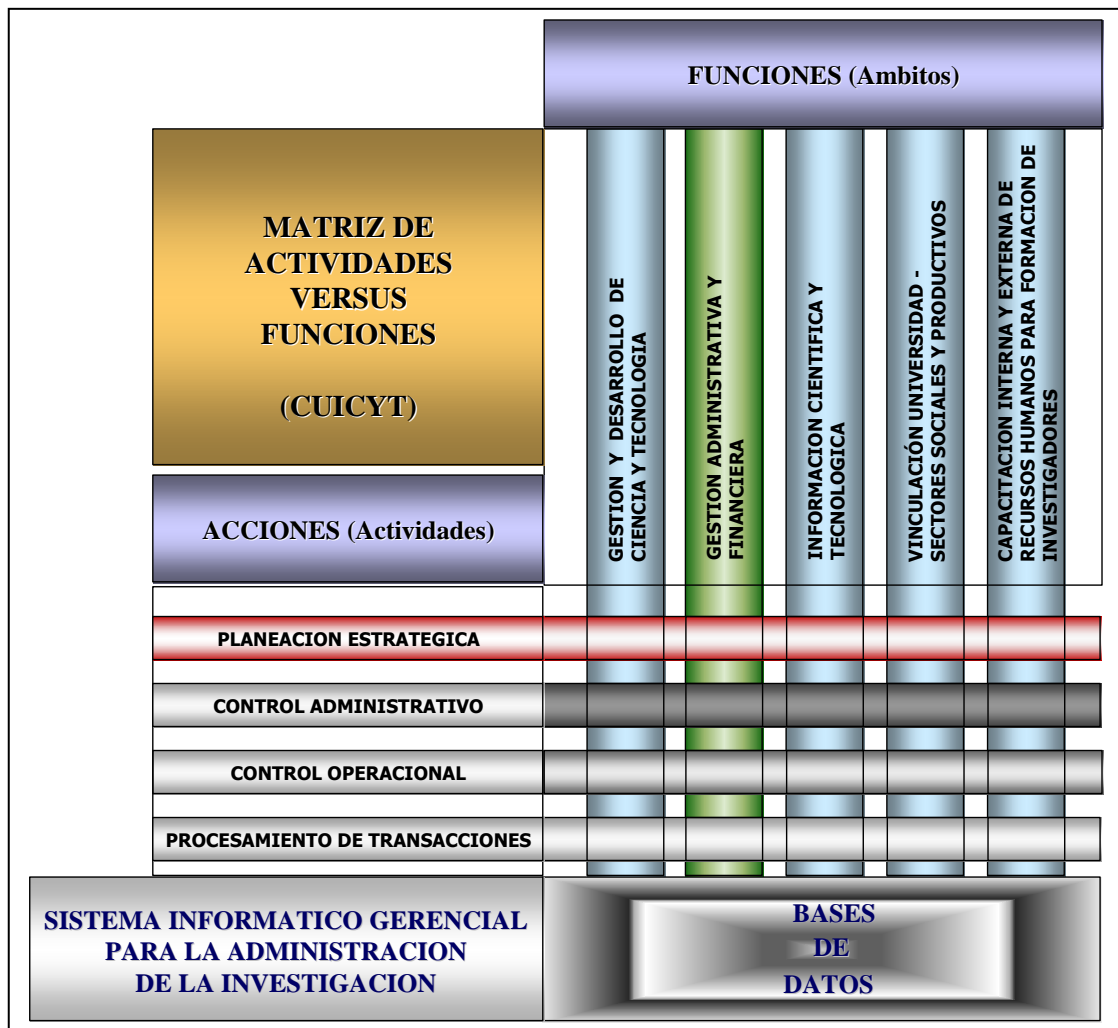
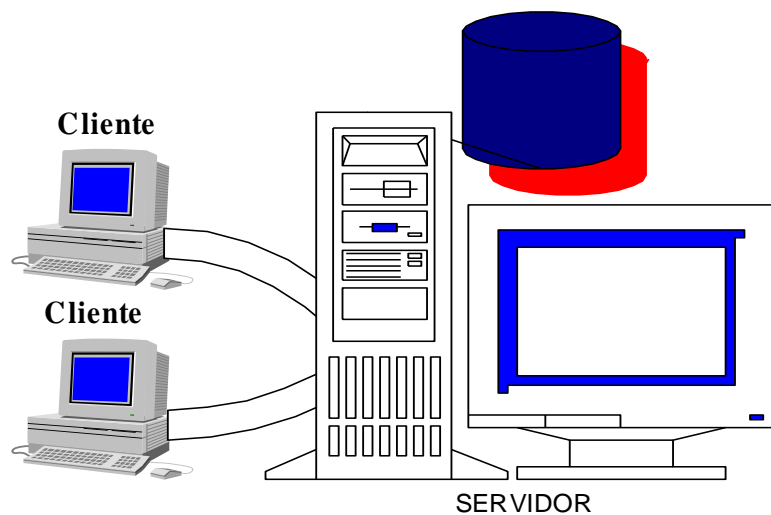


Figura 2.7. Matriz de actividades versus funciones CUICYT.

CAPITULO 3

La Plataforma Cliente/Servidor



3.1 INTRODUCCION

Departamento de Informática, han decidido desarrollar el Sistema Gerencial Informático para la **Gestión y Desarrollo de la Investigación "GEDI-UTN"** con la finalidad de mejorar y estandarizar los procesos operativos de control.

El almacenamiento, manipulación y recuperación de información en forma eficiente, es vital y estratégico para la **Universidad Técnica del Norte (UTN)**. Las Bases de Datos (**BDD**) juegan un rol crítico en las áreas académicas- administrativas - financieras, en donde los datos son derivados de acuerdo a algún grado de interacción con eventos dentro de su campo de acción y con usuarios que están activamente interesados en el contenido de la base de datos.

La importancia de almacenar, manipular y recuperar la información en forma eficiente ha llevado a desarrollar Sistemas de Información (**SI**) en la UTN, capaces de realizar procesamiento sobre un conjunto de datos, en que cada sistema maneje sus propios datos y archivos, en donde los datos se almacenen en forma integrada y se encuentren bajo un control distribuido o centralizado.

Como una Base de Datos es una "colección de datos relacionados" que representa un cierto modelo o abstracción del mundo real (mini-mundo), los cambios en este mini-mundo son reflejados en la BDD.

De acuerdo a este nuevo enfoque de las Bases de Datos de estar orientados hacia los datos y no hacia los programas, ha generado un nuevo paradigma de desarrollo de aplicativos, capaces de mantener un único almacén de datos que se defina de una sola vez y al cual tengan acceso varios usuarios. La información de la BDD debe estar integrada, en donde la integración consista en unificar datos y eliminar la redundancia evitando caer en las siguientes circunstancias:

- La redundancia (datos repetidos).
- La inconsistencia (varias copias de una dato con valores distintos).
- La inseguridad (usuarios no autorizados accedan a los datos).
- La integridad (proteger los datos contra valores no permitidos).
- La concurrencia (uno o más usuarios puedan acceder simultáneamente los datos).

Una característica fundamental de las BDD es que proporcionan cierto nivel de abstracción de los datos, al ocultar detalles de almacenamiento que la mayoría de los usuarios no necesitan conocer. Por tanto los “**modelos de datos**” son el principal instrumento para ofrecer dicha abstracción.

Un modelo de datos es un conjunto de conceptos que pueden ser usados para describir la estructura de una BDD. Con el concepto de estructura de una BDD nos referimos a los tipos de datos, las relaciones y las restricciones que deben cumplirse para esos datos. Por lo general, los modelos de datos contienen además un conjunto de operaciones básicas para especificar lecturas y actualizaciones de la base de datos.

En consecuencia este trabajo permitirá tener una base de entendimiento sólido de los datos involucrados dentro del proceso operativo de gestión de la investigación, sus potencialidades y su impacto dentro de la formulación, análisis, diseño e implementación del Sistema de Gestión de Proyectos de Investigación.

3.2. AMBITO DE LAS ARQUITECTURAS CLIENTE/SERVIDOR

Una plataforma cliente/servidor normalmente es una red de área local (LAN) a la que se enlaza un conjunto de computadores personales y que contiene un computador especial que llamamos un servidor. Los computadores personales son los clientes, que solicitan al servidor diferentes servicios. La característica de cliente/servidor permiten la fácil distribución de inteligencia a lo largo de una red.

La plataforma cliente/servidor no se limita solamente a los proveedores de servicios de bases de datos, sino que se extiende a objetos distribuidos, monitores de **TP** (Procesamiento de Transacciones), groupware e Internet.

Un sistema cliente/servidor posee las siguientes características:

- **Servicio** donde las funciones se distribuyen en equipos distintos.
- **Recursos compartidos** para que un servidor atienda a muchos clientes al mismo tiempo.
- **Protocolos asimétricos** entre clientes y servidores donde los clientes inician la petición de servicio y el servidor espera las solicitudes del cliente.
- **Transparencia de ubicación**, donde los procesos del cliente y del servidor pueden residir en el mismo equipo.
- **Mezcla e igualdad**, donde el reto es tener un software independiente del hardware y la plataforma del sistema operativo.

- **Intercambios basados en mensajes**, donde el mensaje se convierte en el medio para el cliente entregar solicitudes y el servidor devolver respuesta de un servicio.
- **Encapsulamiento de servicios**, donde un servidor está hecho para cumplir tareas específicas, es decir es un especialista.
- **Facilidad de escalabilidad**, donde puede ser escalable horizontalmente aumentando o disminuyendo el número de clientes, o verticalmente con mejora física de equipos.
- **Integridad de los datos**, dado que un servidor tiene programas de protección, recuperación, y seguridades robustas, y permite que el cliente sea independiente.

Brevemente se analizará los diferentes esquemas de servidores.

Los Servidores de Archivos, con esta idea nacieron las bases de datos, donde existe una modalidad primitiva de servicio. Aquí se requieren numerosos intercambios de mensajes a través de la red para obtener los datos solicitados. Estos servidores son muy útiles para compartir archivos y se los utiliza como depósitos de archivos.

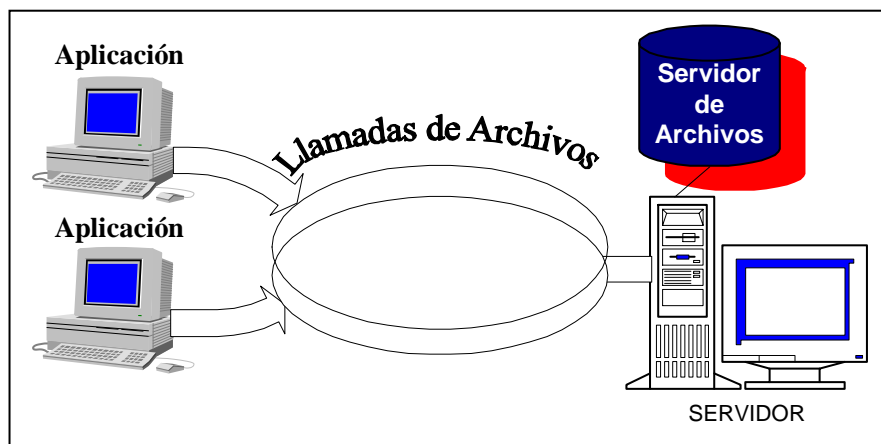


Figura 3.1 CLIENTE/SERVIDOR con servidores de archivos.

Los Servidores de Bases de Datos, el cliente envía solicitudes de SQL en calidad de mensajes al servidor de bases de datos, los resultados de cada SQL son devueltos por medio de la red.

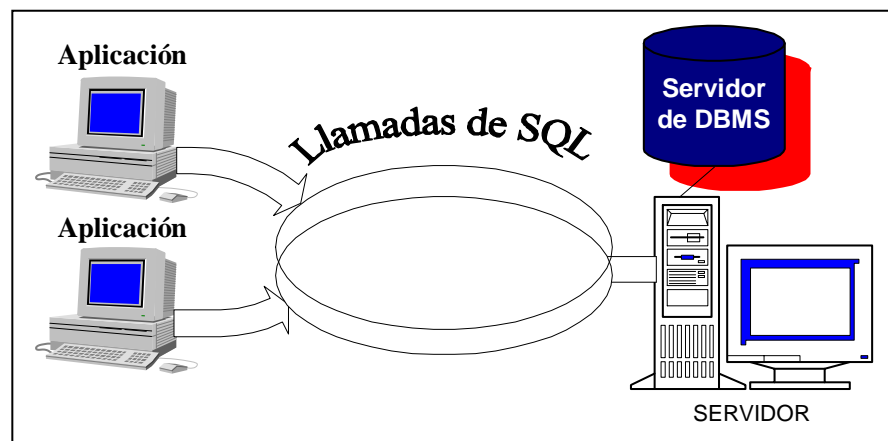


Figura 3.2 CLIENTE/SERVIDOR con servidores de bases de datos.

Los servidores de transacciones, el cliente invoca procedimientos remotos, que residen en el servidor con un mecanismo de bases de datos de SQL, la ventaja es que puede ser invocado a la vez un buen grupo de instrucciones SQL en una sola petición de servicio, en lugar de varias peticiones SQL, como ocurre con un servidor de bases de datos. Estas instrucciones SQL agrupadas se les conoce como transacciones.

Un ejemplo de estos servidores transaccionales son las aplicaciones que soportan los procedimientos almacenados, o los DBMS que soportan el modo transaccional.

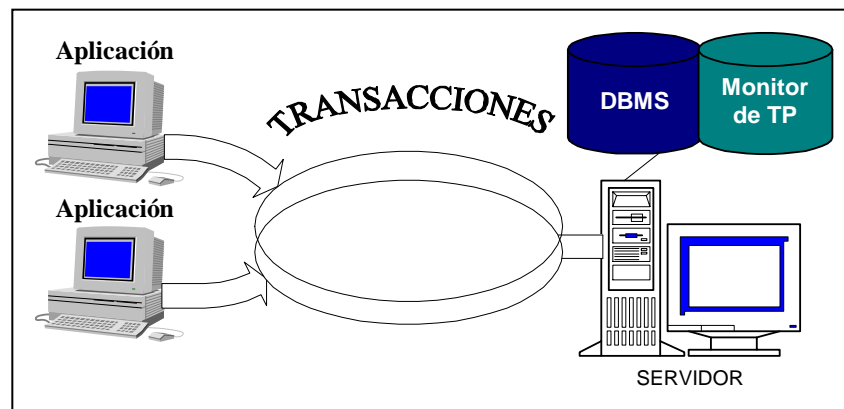
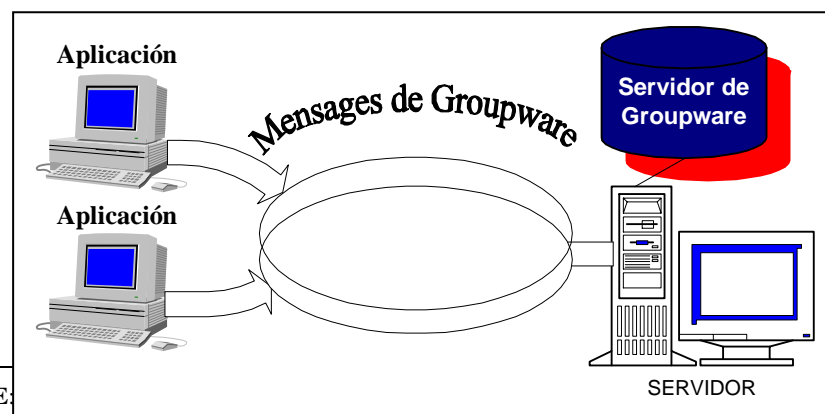


Figura 3.3. CLIENTE/SERVIDOR con servidores de transacciones.

Servidores de Groupware, que administran información semiestructurada como texto, imagen, correo, tableros de aviso y flujos de trabajo. **Lotus Notes**, es un ejemplo de esta arquitectura. Lotus Notes es un ejemplo de software especializado que puede integrarse a API de cliente/servidor de un proveedor. Lotus Notes utiliza un lenguaje de creación de scripts, y puede enlazarse a bases de datos relacionales por medio de **middleware**¹ (batido de tecnologías de software) de comunicación.



¹ MIDDLEWARE:

2da Edición. Mc.Graw - Hill)

de Supervivencia:

Figura 3.4. CLIENTE/SERVIDOR con servidores de groupware.

Servidor de Objetos, con un servidor de objetos, la aplicación cliente/servidor se genera como un conjunto de objetos de comunicación. Los objetos del cliente se comunican con los objetos del servidor mediante un **corredor de solicitudes de objetos (ORB: object request broker)**. El cliente invoca un método del objeto remoto, esto quiere decir que se comunican por medio de los mensajes. El ORB localiza la instancia de esa clase de servidor de objetos, invoca el método solicitado y envía los resultados al objeto del cliente. Los objetos del servidor deben ofrecer soporte de concurrencia y participación. EL ORB, se encarga de juntar todos estos elementos. Entre los servidores de componentes más comunes y comerciales se encuentra a la cabeza CORBA y DCOM de Microsoft.

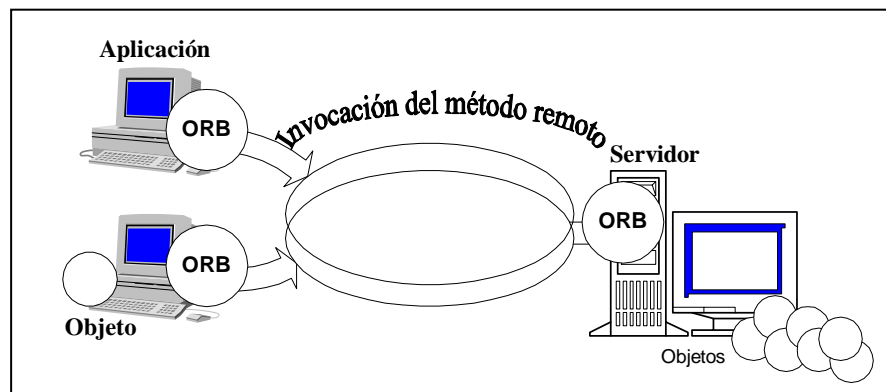
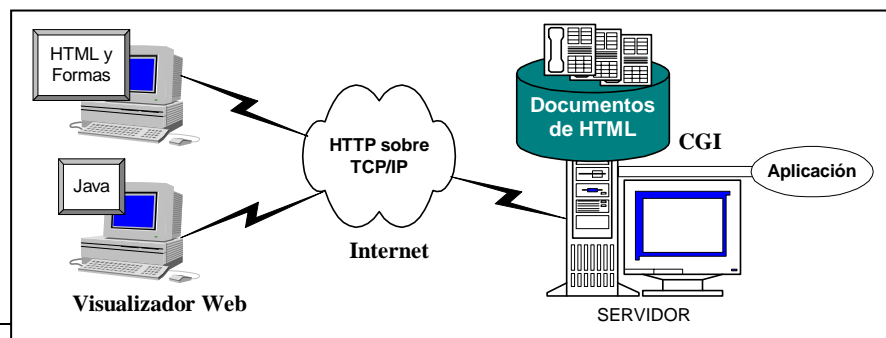


Figura 3.5. CLIENTE/SERVIDOR con servidores de objetos distribuidos.

Servidores Web, es un conjunto de aplicaciones cliente/servidor intergaláctica, sus clientes son universales, portátiles en comunicación con servidores amplios.

Un servidor Web envía documentos cuando los clientes lo piden por su nombre. Los clientes y servidores se comunican mediante un protocolo semejante a **RPC²** denominado **HTTP³**. En la actualidad Java es considerado como el nuevo Object Web.



² **RPC** : Remote Procedure Call (Llamada a Procedimiento Remoto) - ISO/IEC JTC1, [SC 33](#), 1996.

³ **HTTP**: HyperText Transfer Protocol, Protocolo de Transferencia de Hipertexto;
www.linti.unlp.edu.ar/trabajos/tesisDeGrado/tutorial/protocolos/http.htm

Figura 3.6. CLIENTE/SERVIDOR con servidores web.

3.3. Los Sistemas Manejadores de Bases de Datos - DBMS

Inicialmente los servidores se instalaron para controlar la impresión y el acceso a los archivos. Por ejemplo, el servidor de archivos al responder a la solicitud de un cliente sobre un archivo específico, enviará el archivo completo a través de la red al computador cliente. Hoy la mayoría de los servidores son **servidores de bases de datos** - (programas que se ejecutan en el hardware servidor y proporcionan los servicios de las bases de datos a los computadores clientes).

Un cliente que está ejecutando un proceso de una aplicación y necesita una interrogación (consulta) a una base de datos, envía la petición al servidor de la base de datos y éste le devolverá los datos solicitados. El programa de aplicación puede enviar datos al servidor con la solicitud de actualizar la base de datos. El servidor efectuará esta actualización.

La potencia de la plataforma cliente/servidor descansa en el concepto de división de funciones.

FUNCIONES DEL CLIENTE

- El cliente es el computador frontal que tiene interfaz directamente con el usuario.
- Manipula la Interfaz Gráfica de Usuario (**GUI**)⁴ y
- Realiza los cálculos y otros programas de interés para el usuario final.

FUNCIONES DEL SERVIDOR

- El servidor es el que gestiona fundamentalmente el acceso a los datos atendiendo las diferentes peticiones de los clientes.
- Realiza funciones de control y seguridad de acceso a los datos.

La plataforma cliente/servidor son parte del concepto de sistemas abiertos, en el cual todo tipo de computadores, sistemas operativos, protocolos de redes y otros software y hardware pueden interconectarse y trabajar coordinadamente para lograr los objetivos del usuario.

⁴ **GUI.** Interfaz Gráfica de Usuario. Pantallas y funciones que proporcionan al usuario final un medio gráfico para tener acceso al sistema de computación. (DISEÑO Y ADMINISTRACION DE BASE DE DATOS ; HANSEN)

El objetivo de los sistemas abiertos consiste en lograr la interoperabilidad, que es el estado de dos o más sistemas heterogéneos comunicándose y contribuyendo cada uno a alguna parte del trabajo que corresponde a una tarea común.

Un DBMS es una colección de numerosas rutinas de software interrelacionadas, cada una de las cuales es responsable de una tarea específica.

El objetivo primordial de un sistema manejador de base de datos es proporcionar un contorno que sea a la vez conveniente y eficiente para ser utilizado al extraer, almacenar y manipular información de la base de datos. Todas las peticiones de acceso a la base, se manejan centralizadamente por medio del DBMS, por lo que este paquete funciona como interfaz entre los usuarios y la base de datos.

Un sistema de base de datos es algo más que simples datos o que los datos en combinación con un software de gestión de base de datos. En cualquier organización, un sistema de base de datos completo esta compuesto por cuatro componentes: el hardware, el software, los datos y las personas.

El Hardware

Es todo un conjunto de dispositivos físicos y periféricos sobre los que reside una base de datos. Consiste en una o más computadoras, cables de conexión, equipos auxiliares y de conexión del equipamiento. Las computadoras utilizadas para procesar los datos de las bases de datos, pueden ser un mainframe, minicomputadoras o computadoras personales. El servidor en sí mismo puede ser una computadora más potente con multiprocesador.

Las unidades de disco constituyen el mecanismo de almacenamiento principal para las bases de datos y son esenciales, ellas permiten el acceso directo, sin el cual el procesamiento de las bases de datos sería imposible.

El éxito de los sistemas de base de datos ha dependido fuertemente de los adelantos en la tecnología del hardware. Para mantener y controlar la cantidad enorme de datos almacenados en una base de datos se requiere una memoria principal y un espacio de almacenamiento en disco muy grandes. Adicionalmente, se necesitan computadoras rápidas, redes y periféricos para ejecutar el alto número de accesos requeridos para recuperar la información en un tiempo aceptable en un ambiente que tenga una cantidad grande de usuarios.

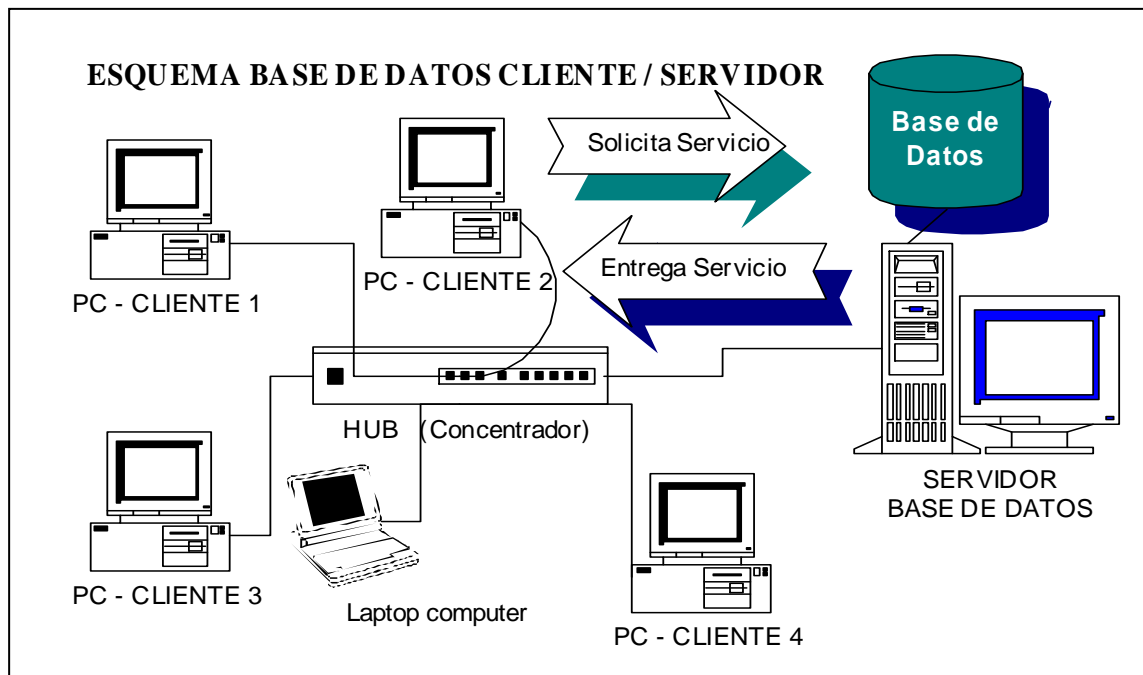


Figura 3.7. CONFIGURACION DE BASE DE DATOS CLIENTE / SERVIDOR

El Software

Un sistema de base de datos incluye dos tipo de software:

- El software de propósito general para la gestión de base de datos, comúnmente llamado Sistema de Gestión de Base de Datos (SGBD).
- El software de aplicación, que usa las facilidades del SGBD para manipular la base de datos con el fin de llevar a cabo una función específica.

El software de aplicación

Generalmente se desarrolla para resolver un problema específico. Puede estar escrito en un lenguaje de programación (usualmente llamado lenguaje de cuarta generación) suministrado con el sistema de gestión de base de datos. El software de aplicación usa las facilidades del SGBD para el acceso y la manipulación de los datos en la base de datos, proporcionando los informes y los documentos necesarios para los requisitos de información de una institución.

El sistema de gestión de bases de datos

El sistema de gestión de bases de datos (SGBD) es un software, parecido a un sistema operativo o a un compilador, que brinda un conjunto de servicios a los usuarios finales, los programadores y otros. Como su nombre indica, el SGBD existe para facilitar la gestión de una base de datos. Con este fin, un SGBD típicamente brinda la mayoría de los servicios siguientes:

- Herramienta para la definición y control centralizados de datos (LDD – Lenguaje de definición de datos), conocida como diccionario de datos/directorio (DD/D) o catálogo.
- Mecanismos de seguridad e integridad de los datos (LCD – Lenguaje de control de datos).
- Acceso concurrente a los datos para varios usuarios.
- Utilidades para la consulta , la manipulación y la elaboración de informes orientados al usuario (LMD – Lenguaje de manipulación de datos).
- Utilidades para el desarrollo de sistemas de aplicación orientados al programador.

Diccionario de Datos/Directorio (DD/D). El subsistema del diccionario/directorio de datos almacena las definiciones de todos los elementos de todos los datos en la base de datos. Esto incluye el nivel primario de los elementos de los datos (campos), las estructuras de los datos a nivel de grupo y a nivel de registro y los archivos o tablas relacionales. El DD/D no sólo mantiene esta información, sino que, además almacena las interrelaciones que existen entre las diversas estructuras de datos. Adicionalmente, mantiene los índices que se utilizan para garantizar el acceso a los datos rápidamente. Asimismo, almacena las definiciones de los formatos de las pantallas y de los informes, que pueden utilizarse por varios programas de aplicación.

El diccionario de los datos puede verse como una parte de datos en sí misma. De esta manera, la base de datos es autodescriptiva, pues contiene información que describe su propia estructura. La información en el diccionario de datos se llama **metadatos**⁵ o "datos de los datos". Los metadatos están disponibles para la interrogación y la manipulación, como lo están los otros datos en la base de datos.

El LDD tiene como principales funciones :

- Crear, suprimir o modificar la definición de una tabla.
- Definir y suprimir una vista de datos

⁵ **Metadatos.** Datos en el diccionario de datos que describen a la base de datos. Microsoft Access, Sanchez Navarro.

Estas funciones permiten definir y modificar el esquema de una base de datos y por lo tanto provoca una modificación del catálogo del sistema, que contiene el esquema de la base de datos.

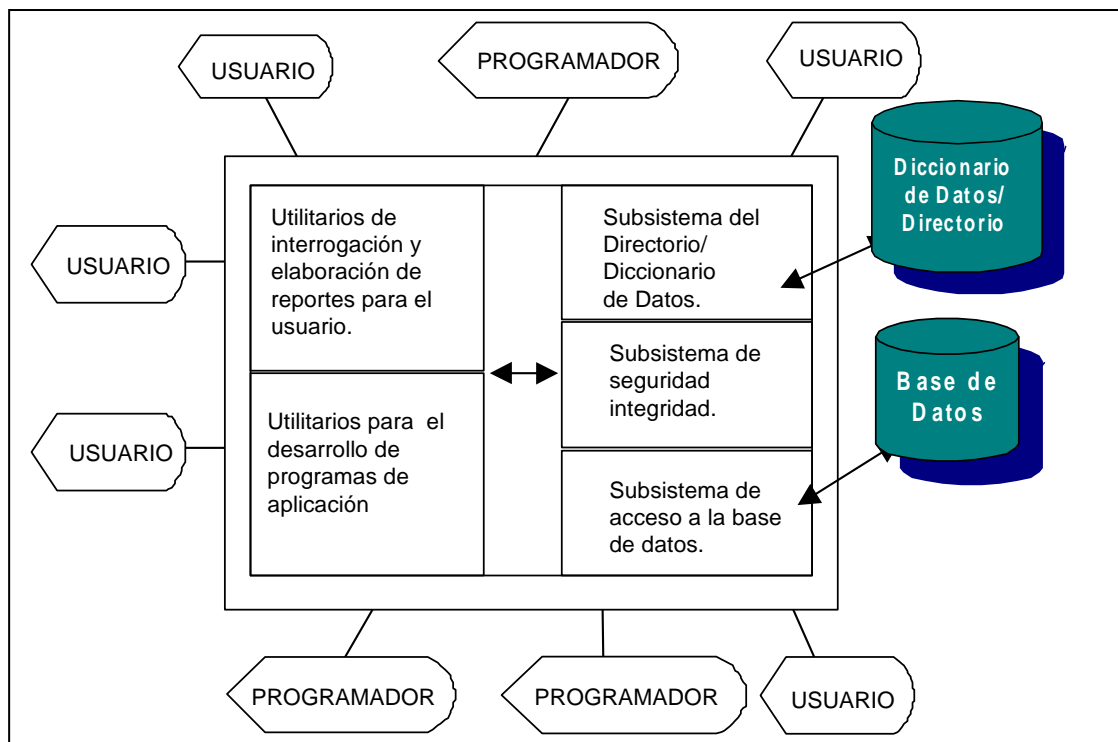


Figura 3.8. COMPONENTES DE UN SISTEMA DE GESTION DE BASE DE DATOS

Mecanismos de seguridad e integridad de los datos. La base de datos es un recurso valioso que necesita protección. El SGBD garantiza la seguridad de la base de datos limitando el acceso a la misma al personal autorizado. Para ello utiliza la concesión de privilegios para permitir a los usuarios el acceso completo o restringido a las tablas de la base de datos.

Los usuarios autorizados, generalmente, estarán restringidos en cuanto al acceso a ciertos datos en particular y a aquellos que puedan actualizar. Tal acceso se controla frecuentemente mediante contraseñas y mediante las **vistas de los datos**, que constituyen definiciones de porciones restringidas de la base de datos, como se ilustra en la Figura 3.9; la integridad y la consistencia de la base de datos se protegen por medio de restricciones sobre los valores que pueden tomar los elementos de los datos y por las capacidades de recuperación y de respaldos suministradas por el SGBD. Las definiciones de las restricciones de los datos se almacenan en el diccionario de datos. La recuperación y el respaldo se soportan por el software, el que automáticamente almacena los cambios de la base de datos en el catálogo y brinda los medios para restablecer el estado actual de la base de datos en el caso de un fallo del sistema.

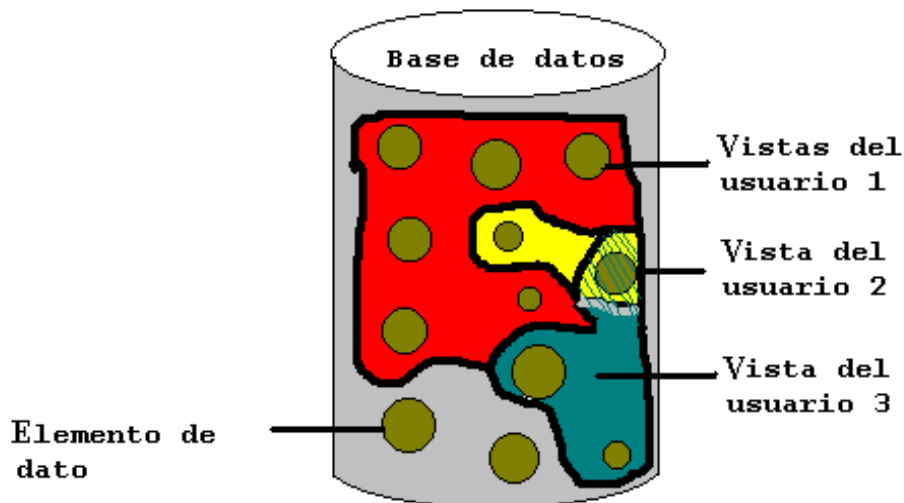


Figura 3.9. VISTAS DE USUARIO DE UNA BASE DE DATOS.

Acceso concurrente a los datos por varios usuarios. Una de las funciones principales del SGBD es garantizar el acceso, la recuperación y actualización de los datos en la base de datos. El SGBD proporciona los mecanismos físicos que permiten a varios usuarios tener acceso de forma rápida y eficiente a diferentes datos relacionados. Esta posibilidad se extiende a usuarios localizados remotamente, quienes tendrán acceso a la base de datos mediante un sistema de telecomunicaciones, de modo que las solicitudes de datos y las respuestas resultantes se enrutan adecuadamente.

La centralización de los datos en una base de datos aumenta la probabilidad de que dos o más usuarios quieran tener acceso a los mismos datos concurrentemente. Si el SGBD permite esto, los dos usuarios seguramente afectarían el trabajo del otro y podrían dañarlo. Así, es importante que el SGBD proteja los datos, a los que tiene acceso un usuario, de la actualización simultánea por otro usuario. Para hacer esto, el SGBD utiliza mecanismos sofisticados de bloqueo para proteger los datos que se están actualizando por un usuario, mientras al mismo tiempo garantiza el acceso concurrente a la base de datos y un tiempo de respuesta del sistema a otros usuarios aceptable.

Consultas e informes dirigidos al usuario. Uno de los aspectos más valiosos de un SGBD es que brinda herramientas de manipulación de los datos dirigidas al usuario. Estos lenguajes de interrogación fáciles de usar permiten a los usuarios formular sus consultas y pedir informes únicos directamente de la base de datos. Esto releva al personal de programación de la compañía de la carga formular estas consultas o de escribir paquetes de programas de aplicación de propósito especial.

Los generadores de informes están asociados a los lenguajes de consulta.

Frecuentemente el lenguaje de consulta contendrá facilidades para estructurar los resultados de las consultas como informes. La consulta formulada puede almacenarse para su uso ulterior y sus resultados pueden producirse en forma de un informe ordinario. Cuando este es el caso, el lenguaje de consulta puede considerarse como un generador de informes. Además los generadores de informes pueden también brindar facilidades mucho más poderosas que las disponibles en el lenguaje de consulta.

Facilidades para el desarrollo de aplicaciones. Aparte de facilitar al usuario el acceso a la base de datos para obtener información, el SGBD también garantiza una asistencia importante al programador de aplicaciones. Tales aplicaciones como los generadores de pantallas , de menús y de reportes; los generadores de aplicaciones, los compiladores, y las facilidades para la definición de los datos y las vistas de los datos son estándares. Más importantes aún, los sistemas de bases de datos modernos brindan componentes del lenguaje mucho más poderosos que los de los lenguajes tradicionales, lo que contribuye a que el proceso de programación sea apreciablemente más eficiente.

Los datos.

Obviamente, ningún sistema de base de datos puede existir sin los datos, los hechos básicos sobre los que se fundamentan las necesidades de información y de procesamiento de una compañía o institución. Sin embargo, el factor esencial a considerar es que los datos que conforman una base de datos tienen que ser cuidadosa y lógicamente estructurados. Las funciones del negocio debe analizarse, los elementos de los datos y las interrelaciones deben identificarse y definirse cabalmente; estas definiciones deben almacenarse de manera precisa en el diccionario de datos. Entonces, los datos pueden obtenerse e introducirse en la base de datos según la estructura definida. Una base de datos construida en armonía con estos procedimientos puede ser un recurso poderoso para garantizar información oportuna a una organización.

Las personas.

La información que se disponga la utiliza y maneja de acuerdo a las funciones y requerimientos de información del usuario. Cada usuario tiene un privilegio o categoría de usuario, ejemplos de personas en cada una de las categorías pueden ser los siguientes:

- **Usuarios:** Los ejecutivos, los gerentes, los administradores, el personal de oficina. Personas que necesitan información de la base de datos para desarrollar su responsabilidad primaria en el negocio.
- **Profesionales de la computación:** Los administradores de la base de datos, los analistas, los programadores, los diseñadores del sistema y de la base de datos, los administradores de los sistemas de información. Personas responsables del sistema de la base de datos y del paquete de programas de aplicación asociado.

Procedimiento. Instrucciones escritas que describen los pasos necesarios para realizar una tarea determinada en un sistema.

Las personas encargadas de los procedimientos usados para lograr las metas del sistema constituyen una parte importante de este componente. Virtualmente, ningún sistema automatiza por completo una tarea del usuario. Es necesario desarrollar procedimientos manuales para brindar una interfaz suave entre los usuarios del sistema y el sistema en sí mismo.

Un ejemplo de procedimiento sería el control de auditoría por parte de los usuarios con respecto a que cantidad total de dinero depositada en el banco correspondiente a un día determinado concuerde con la cantidad total de efectivo recibido mostrada por el sistema ese mismo día. Este es el caso de los procesos de cierres diarios del Sistema de Recaudación, y los informes mensuales, normalmente existen muchos procedimientos en un sistema y el éxito del sistema, con frecuencia, depende de la habilidad con que tales procedimientos se desarrollen para articular con las funciones del sistema, así como de la estructura del sistema en sí mismo.

Interrelación (relación) entre los cuatro componentes del sistema.

Los profesionales de la computación (los diseñadores de base de datos y analistas) en consulta con los usuarios, identifican las necesidades de datos y diseñan las estructuras de las bases de datos para responder a estas necesidades. Entonces, las estructuras de la base de datos se especifican al SGBD mediante el diccionario de datos. Los usuarios introducen los datos en el sistema siguiendo procedimientos específicos. Los datos introducidos se almacenan en los medios de hardware, tales como discos y cintas.

Los programas de aplicación que tienen acceso a la base de datos se escriben por los profesionales de la computación y por los usuarios para ejecutarlos sobre las computadoras. Estos programas utilizan el lenguaje de órdenes del SGBD y hacen uso de la información contenida en el diccionario

de datos. Estos programas generan información, que puede utilizarse por los ejecutivos y los gerentes para tomar decisiones en el negocio. Los programas de aplicación también pueden generar las notificaciones de documentos utilizados por los clientes del negocio. De este modo, puede verse que en un sistema adecuadamente diseñado y en funcionamiento, sus cuatro componentes – el hardware, el software, los datos y las personas - conforman un sistema único con el fin de alcanzar las metas de la organización.

3.4. EL MIDDLEWARE.

Generalmente los DBMS tienen sus clientes específicos como frontales, pero en la mayoría de los casos un frontal es independiente del back-end, por lo que se necesita algún modo de enlazarse a las Bases de datos, cómo se lo hace, con Middleware.

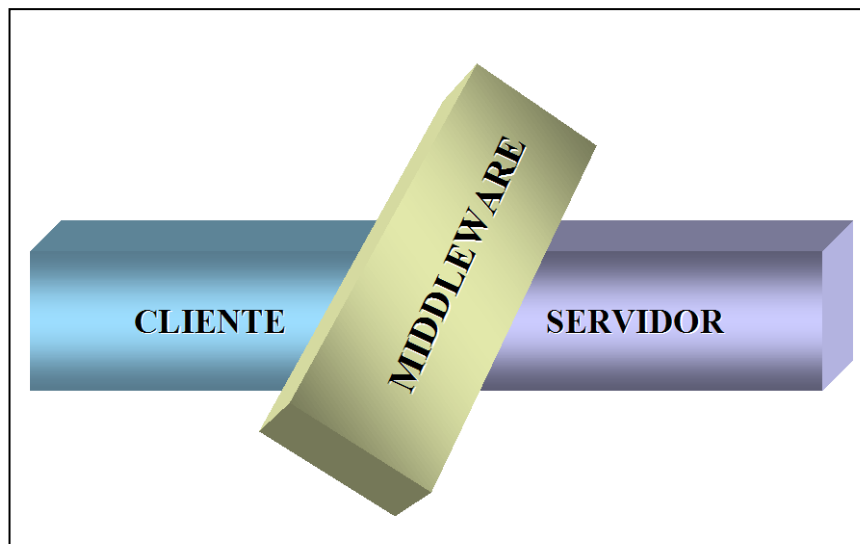


Figura 3.10. Funcionalidad del MIDDLEWARE.

Middleware es un término que abarca al software distribuido necesario para la interacción entre clientes y servidores. Es el enlace que permite que un cliente obtenga un servicio de un servidor. Empieza en el módulo de API de la parte del cliente que se emplea para solicitar un servicio y comprende la transmisión de los datos por la red. El Middleware se divide en dos clases:

El Middleware general.- incluye las pilas de comunicaciones, directorios distribuidos, servicios de autenticación, llamadas a procedimientos remotos y servicios en cola. También se incluyen las extensiones del sistema operativo de redes como los servicios de archivos e impresión, ejemplos de Middleware son:

Netware, Named Pipes, LAN Server, LAN Manager, TCP/IP, y NetBIOS.

El Middleware de servicios específicos.- Necesario para cumplir un tipo particular de servicio de cliente servidor. Pertenecen a esta categoría:

- El Middleware para Bases de Datos, como ODBC, DRDA⁶, EDA/SQL, SAG/CLI.
- Para OLTP, como ATMI, Transactional RPC, de Encina y TxRPC, XATMI de X/Open.
- Para GroupWare como MAPI, VIM, VIC, SMTP y las llamadas de Lotus Notes.
- Para Objetos como CORBA de OMG, y DCOM de Microsoft.
- Para Internet como HTTP, HTTPS y SSL.
- El Middleware para administración de Sistemas, como SNMP, CMIP y ORB.

EL MIDDLEWARE DE SQL

El Middleware específico para bases de datos es necesario para igualar los diferentes dialectos y extensiones de SQL, protocolos de manejo de mensajes en Red y API nativas del proveedor. Este Middleware es necesario para lograr que clientes y servidores de SQL trabajen en redes de Base de Datos heterogéneas de proveedores múltiples.

CLI⁷ DE MICROSOFT

El estándar de API de Windows ODBC de Microsoft para SQL es una versión ampliada de la CLI de SAG.ODBC2.0 define alrededor de 61 llamadas de API, correspondiendo a los siguientes niveles:

- El núcleo, que ofrece 23 llamadas base que permiten conexión con una base de datos, ejecución de instrucciones SQL, búsqueda de resultados, grabación y retroceso de transacciones, manejo de excepciones y conclusión de la conexión.
- El nivel 1, ofrece 19 llamadas adicionales que permiten recuperar información de un catálogo de base de datos, buscar objetos grandes (BLOB) y asumir funciones específicas del controlador.
- El Nivel 2, ofrece otras llamadas de API.

CARACTERISTICAS DE DRDA.

El DRDA, se encarga de varios asuntos de Red y la exportabilidad del código, tales como:

⁶ **DRDA.** Distributed Relational DataBase Architecture o Arquitectura distribuida de bases de datos de IBM. IBM desarrolló esta arquitectura para hacer posibles las operaciones entre bases de datos distribuidas.

⁷ **CLI:** call-level interface; Interfaz del nivel de llamada. (CLIENTE SERVIDOR - Guía de Supervivencia: 2da Edición. Mc.Graw - Hill). pag 183.

- Contenido de los mensajes y protocolo de intercambio de SQL: DRDA maneja las negociaciones entre clientes y servidores para atributos del servidor soportados. Efectúa traducciones de mensajes solo en caso necesario.
- Dependencia de la Pila de Transporte: DRDA soporta la interfaz de MPTN⁸, lo que significa que puede correr sobre APPC/APPN o TCP/IP para comunicaciones de cliente/servidor.
- Preparación de programas de plataformas múltiples: un programa se crea localmente; su producción puede distribuirse a servidores múltiples por medio de una utilería de BIND remota.
- Soporte de SQL estático o dinámico: Un cliente de DRDA puede invocar una por una las instrucciones SQL en el servidor mediante la identificación de un paquete y de la instrucción al interior de esta. Los paquetes también permiten ejecutar instrucciones de SQL estático precompilado.
- Diagnostico común: DRDA emite información de estado sobre la terminación de cada comando de SQL. Ofrece asimismo un conjunto estándar de códigos de resultado final en el campo SQLSTATE.
- Sintaxis de SQL común: DRDA recomienda el uso de un subconjunto de SQL92 para la aplicación de exportabilidad entre plataformas. Soporta también comandos de SQL específicos del destino para situaciones en las que es más eficiente utilizar extensiones de SQL.

UTILIZACION DE CLI - MICROSOFT

Un ejemplo es la conexión que se puede lograr utilizando Access versión 97, 2000, que tiene la capacidad de conectarse con otras bases de datos por medio de ODBC.

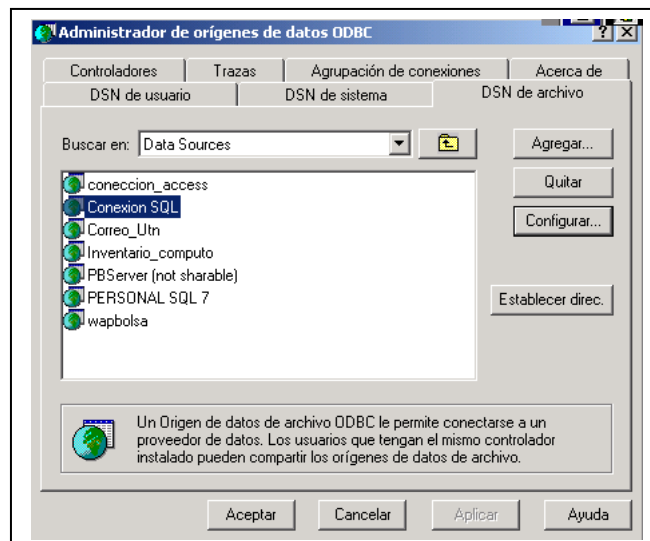


Figura 3.11. Ejemplo: El Administrador de ODBC.

⁸ MPTN: MultiProtocol Transport Network; encontrado en: www.galego21.org/nos/anxo_forxan/siglas/siglas.html.

A continuación se especifican los pasos que se deben seguir para una conexión mediante ODBC, el mismo que puede ser provisto por el DBMS al cual nos vamos a enlazar.

7. Levantar la Base de Datos y los servicios de SQL-SERVER, previamente instalada.

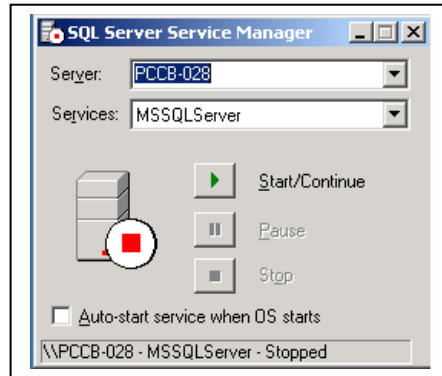


Figura 3.12. Inicio de servicios de SQL-SERVER

8. Crear el ODBC, especificando el nombre del ODBC, una breve descripción, el nombre del motor Servidor de la Base de datos, y la base de datos a donde apuntará el ODBC, en nuestro caso Investigación.

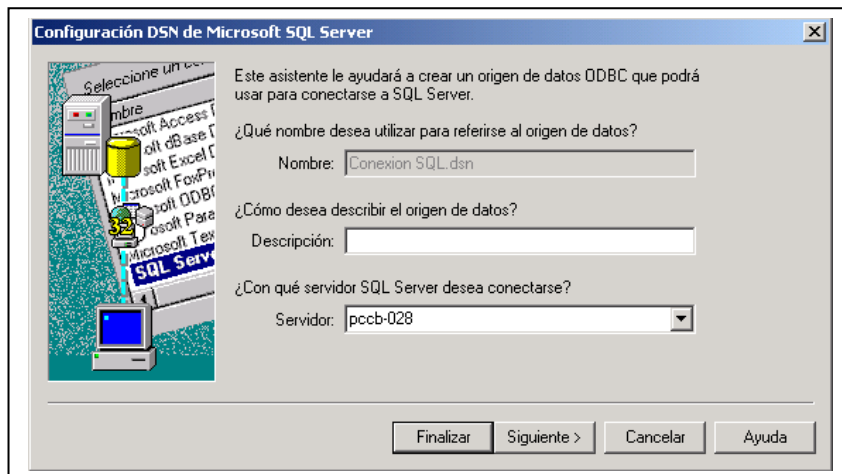


Figura 3.13. Ejemplo: El Administrador de ODBC.

9. Diseñar los atributos de las Tablas

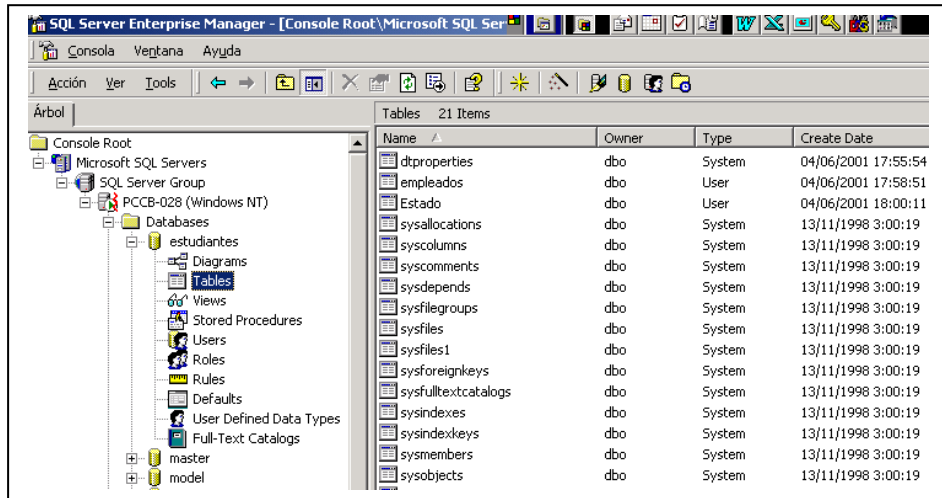


Figura 3.14. Entorno de trabajo de SQL-SERVER 7.0, vista de objetos.

10. Vincular desde Access las tablas de la base de datos hecha en SQL-SERVER.

a) Conexión utilizando el ODBC, desde Access.

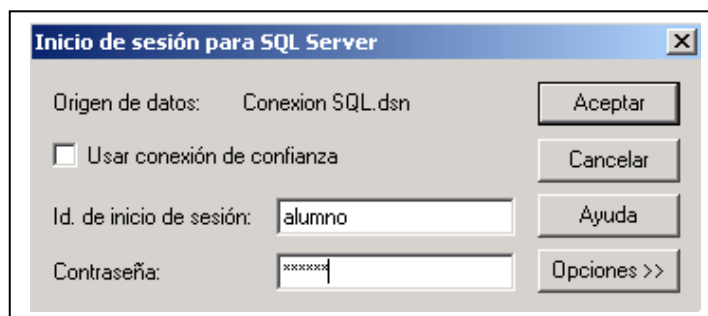


Figura 3.15. Conexión ODBC, DBMS.

b) Seleccionar las tablas necesitadas.

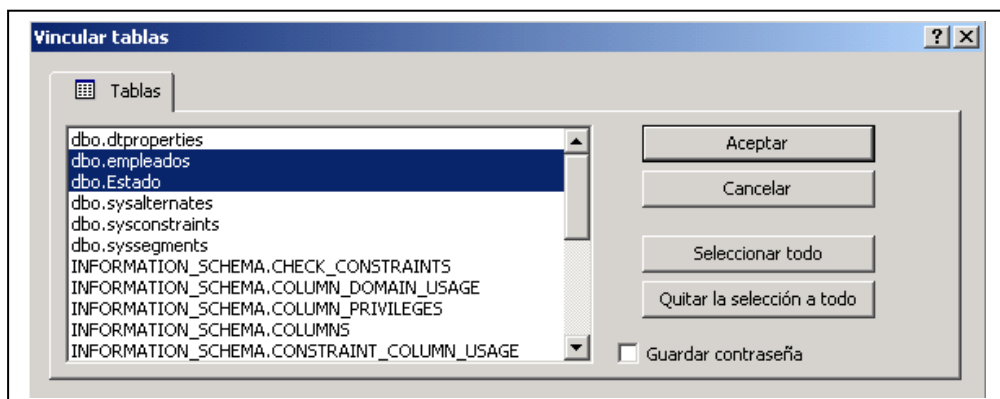


Figura 3.16. Seleccionando las tablas a vincular

c) Utilizar las tablas vinculadas como propias de Access, para trabajar en los formularios.

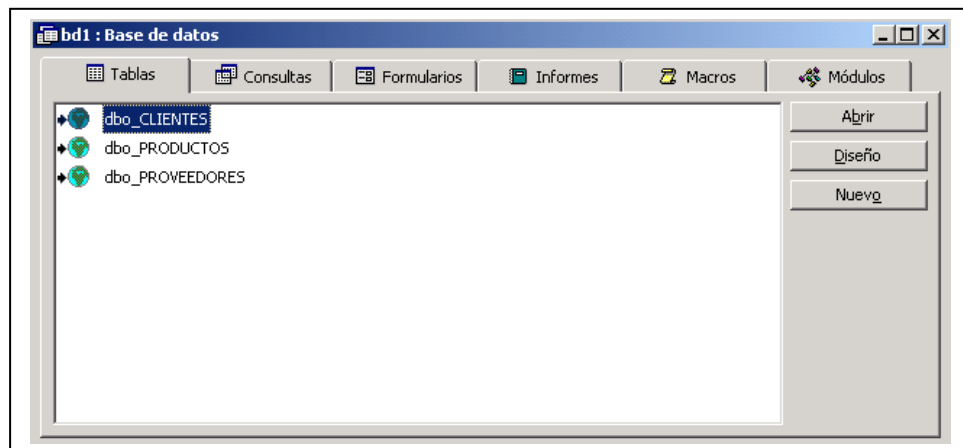


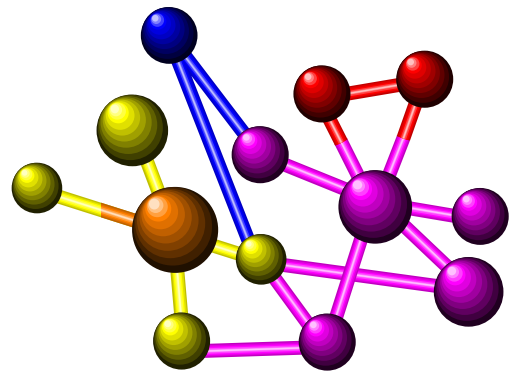
Figura 3.17. Resultado de vincular

Con las tablas vinculadas Access trabajará con ellas como si fueran propias de la base, pero en realidad son tablas que pertenecen a un motor de base de datos, en este caso SQL-Server 2000.

En conclusión, mediante el uso de ODBC, se puede lograr no únicamente trabajar para el desarrollo con bases de datos en frontales clientes de terceros, sino también para el modelamiento de bases de datos, como son ERWIN 4.0, Power Designer versiones 7, 8, 9, en este último orientado a objetos, al igual que Rational Rose para modelamiento UML, herramientas que facilitan y optimizan el trabajo del diseñador de aplicaciones orientadas a datos y objetos, es decir el **Middleware** sirve como canal de comunicación entre un DBMS y herramientas de desarrollo de terceros.

CAPITULO 4

Modelamiento Orientado a Objetos



4.1 INTRODUCCION

La cuarta era del software de computadora está empezando ahora. Las tecnologías orientadas a los objetos están desplazando rápidamente a enfoques de desarrollo de software más convencionales en muchas áreas de aplicación. Las técnicas de cuarta generación para el desarrollo de software están cambiando la forma en que algunos segmentos de la comunidad informática construyen los programas para computadora. Por fin, estas técnicas proporcionan una serie de beneficios significativos en la calidad del software como: portabilidad, reusabilidad, mantenibilidad, etcétera. Obviamente estos factores de calidad del software no se logran automáticamente, aún cuando se use la orientación a objetos, se debe realizar cambios en todo el proceso de desarrollo (planificación y evaluación de recursos, análisis de requerimientos, aseguramiento, documentación, diseño, codificación y pruebas).

4.2. TENDENCIA DE LAS BASES DE DATOS ORIENTADO A OBJETOS

Deben haber DBMS que almacén objetos como tales, es decir datos y métodos para ser consideradas como tales, lo que en la mayoría de las bases de datos es disponer de procedimientos almacenados que equivalen a los métodos.

Las primeras bases de datos surgieron del desarrollo de los sistemas de gestión de archivos. Estos sistemas primero evolucionaron en bases de datos en red o en bases de datos jerárquicas y más tarde, en bases de datos relacionales.

En años recientes la tecnología de bases de datos se ha adaptado a aplicaciones fuera de ámbito del procesamiento, estas nuevas aplicaciones incluyen:

- DISEÑO ASISTIDO POR COMPUTADOR (Computer - aided design (CAD)): Una base de datos CAD almacena datos pertenecientes a un diseño de ingeniería, incluyendo las componentes del dato que se está diseñando, la relación de las componentes y las versiones antiguas de los diseños.
- INGENIERIA DE SOFTWARE ASISTIDA POR COMPUTADOR (Computer – aided software engineering (CASE)): Una base de datos CASE almacena los datos requeridos para asistir a los que desarrollan el software. Estos datos incluyen código fuente, dependencias entre

módulos software, definiciones y usos de variables y la historia del desarrollo del sistema software.

- **BASES DE DATOS MULTIMEDIOS:** Una base de datos multimedia contiene datos de audio, datos de video, y similares. Las bases de datos de este tipo derivan de datos geofísicos, sistemas de correo por voz y aplicaciones gráficas.
- **SISTEMAS DE INFORMACION DE OFICINA (Office information systems (OIS)):** La ofimática incluye herramientas basadas en estaciones de trabajo para creación y recuperación de documentos, herramientas para mantener calendarios con citas, etc. Una base de datos OIS debe permitir consultas que pertenezcan a planificaciones, documentos y contenidos de documentos.
- **SISTEMAS EXPERTOS DE BASES DE DATOS:** Un sistema experto de base de datos, además de datos, incluye reglas explícitas que representan las restricciones de integridad, disparadores y otros conocimientos acerca de la empresa que está modelando la base de datos.

Estas nuevas aplicaciones requieren nuevos modelos de datos, nuevos lenguajes de consulta y nuevos modelos de transacciones. Entre los requisitos de estas nuevas aplicaciones están:

4.2.1. REQUISITOS DE LAS NUEVAS BASES DE DATOS ORIENTADAS A OBJETOS

- **OBJETOS COMPLEJOS:** Un objeto complejo es un dato que es visto como un simple objeto en el mundo real, pero que contiene otros objetos. Estos objetos pueden tener una estructura interna compleja arbitraria. A menudo los objetos están estructurados jerárquicamente, representando la relación entre ellos. El modelo de objetos complejos ha llevado al desarrollo de las ***bases de datos orientadas a objetos***, las cuales están basadas en los lenguajes de programación orientada a objetos y a las bases de datos relacionales anidadas, en las que las relaciones pueden almacenarse dentro de otras relaciones.
- **DATOS DE COMPORTAMIENTO:** Puede que distintos objetos necesiten responder de diferentes formas a la misma orden. Por ejemplo la eliminación de ciertas tuplas puede requerir la eliminación de otras tuplas, como en el caso de las entidades débiles, (Eliminación o actualización de desencadenantes en Visual Fox Pro, o Actualización y eliminación en cascada en otras bases de datos como Access). En las aplicaciones CAD y CASE el comportamiento de distintos objetos en respuesta a una orden dada puede ser muy diferente. La información

sobre el comportamiento puede capturarse almacenando código ejecutable con objetos en la base de datos.

Los métodos de los sistemas de bases de datos orientados a objetos y la regla base de los sistemas basados en comportamientos proporcionan esta capacidad.

- **META CONOCIMIENTO:** Se ilustra con un ejemplo de un banco en el que: << Todas las cuentas de cheques pagan el 5% de interés si el saldo es mayor a 100 dólares, en caso contrario no paga interés>>. Este tipo de restricciones no pueden representarse fácilmente en sistemas de bases de datos tradicionales. Sin embargo puede representarse fácilmente mediante reglas en base de datos lógicas. Estas reglas forman una parte importante de los sistemas expertos de bases de datos.

4.2.2. ESTRUCTURA DE OBJETOS

El modelo orientado a objetos se basa en encapsular código y datos en una única unidad llamada **objeto**. La interfaz entre un objeto y el resto del sistema se define mediante un conjunto de mensajes.

Un objeto tiene asociado:

- Un conjunto de **variables** que contienen los datos del objeto. El valor de cada variable es un objeto.
- Un conjunto de **mensajes** a los que el objeto responde.
- Un **método**, que es el código para implementar cada mensaje. Un método devuelve un valor como **respuesta** al mensaje.

I) VARIABLES

Ilustrando la estructura de los objetos podemos señalar que las **variables** corresponden a las propiedades de un objeto, se toma un cuadro combinado para la explicación en donde sus propiedades son las variables.



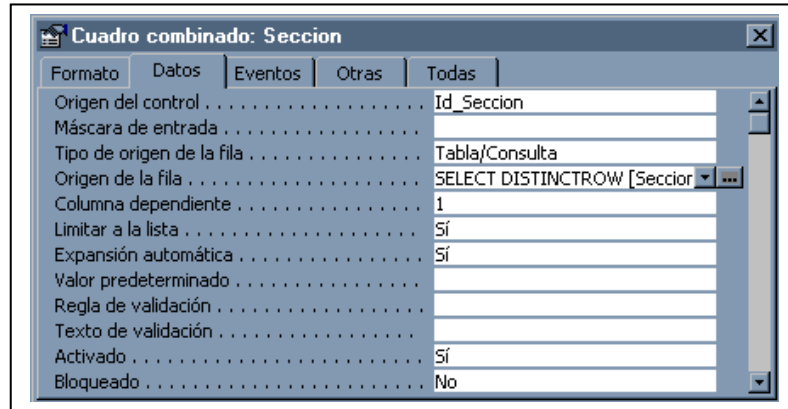


Figura 4.1. Un Combo box, objeto para el análisis.

En las propiedades de este objeto se puede manipular variables como:

- a) **Origen de Control.**_ Indica en donde se almacena la información.
- b) **Tipo de origen de la fila.**_ Por defecto aparecerá, **Tabla/Consulta**; las otras opciones son Lista de valores y Lista de campos, que tienen por objetivo desplegar una lista fija de valores y una lista de campos respectivamente dependiendo del **Origen de la fila**.
- c) **Origen de la fila.**_ Es en donde se determina que tablas y que campos intervendrán en la selección de los datos de origen.

La selección de campos que se va a mostrar es:

Campo	Id_Seccion	Seccion
Tabla	Seccion	Seccion
Orden		Ascendente
Criterios		
O		

- d) **Columna dependiente.**_ Es el número de la columna con la que guarda relación, es decir, el campo de la tabla origen.

Es por esto que se tiene como **columna dependiente: 1**, que corresponde a la columna **Id_Seccion** de la tabla **Seccion**.

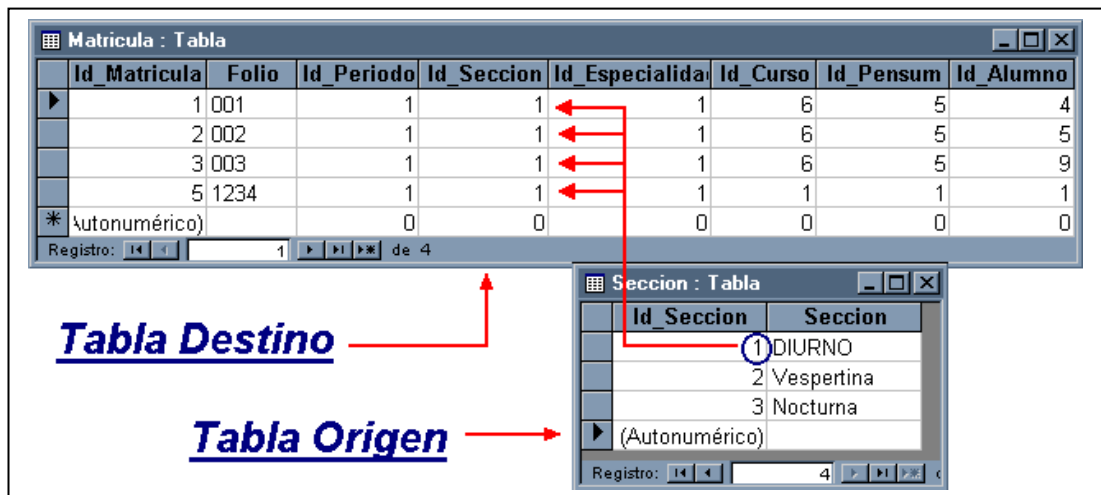


Figura 4.2. Un Combo box, objeto para el análisis.

- e) **Número de columnas.**_ Indica el número de columnas a desplegar en este caso: **2**.
- f) **Encabezado de columna.**_ Se puede seleccionar que se muestren o no los encabezados de las columnas.
- g) **Ancho de columnas.**_ Seleccione desde las propiedades del cuadro combinado la carpeta correspondiente a **Formato**, proceda a trabajar con el ancho de las columnas.

Ancho	de	0cm; 2.54cm
columnas:		

Indica el ancho de 0cm para la primera columna y de 2.54cm para la segunda columna, es decir para Id_Seccion y Seccion respectivamente de la tabla **SECCION**.

II) MENSAJES

Haciendo referencia al mismo objeto, los mensajes pueden ser los eventos que se quiera manipular, como por ejemplo:

- Después de actualizar.
- Al hacer doble click.
- Al no estar en lista, etc.

III) METODO

El método sería el código que se debe implementar para por ejemplo: Al hacer doble click, abrir un formulario en donde se ingresen los datos que no existen en el cuadro combinado:

```
Private Sub ID_PERIODO_DblClick(Cancel As Integer)
    DoCmd.OpenForm "PERIODO", , , , , acDialog, "GotoNew"
    Forms![ASIGNACION DE PARTIDAS A]![PARTIDA_AÑO].Form![ID_PERIODO].Requery
    Me.Refresh
End Sub
```

Figura 4.3. Un Combo box, objeto para el análisis.

Código correspondiente a ejecutar la apertura del formulario " Periodos" cuando se requiere adicionar un nuevo período en el Control Financiero de Proyectos, al momento de hacer la asignación de partidas presupuestarias.

La capacidad de modificar la definición de un objeto sin afectar al resto del sistema está considerada como una de las mayores ventajas del modelo de programación orientada a objetos, por ejemplo se pueden modificar las propiedades de un cuadro de texto desde las propiedades mismas del objeto o alterando estas con código Visual Basic en un evento o módulo.

4.2.3. JERARQUIA DE CLASES

En una base de datos existen muchos objetos similares, es decir tienen variables del mismo nombre y tipo. Por tanto se agrupan los objetos similares para que formen una clase. A cada uno de estos objetos se le llama instancia de su clase.

Para el ejemplo del Sistema de Gestión de Investigación, supongamos que se tienen datos de los Empleados, datos de los Docentes y datos de los Estudiantes, sería innecesario duplicar la información si un empleado, fuera docente, y cualquiera de estos sea Investigador. Tanto docentes, estudiantes y empleados pueden representarse mediante una clase especializada llamada **Persona**. Las variables y métodos se aplican a estas tres subclases.

Las variables asociadas a las clases del ejemplo serían:

Persona : Cédula de identidad, Apellidos, Nombres, Fecha nacimiento, País, Provincia/ Estado, Teléfono.

Empleado : Fecha contrato, salario, numero de dependientes

Docente : Titulo, Carga horaria,

Alumno : Matricula, Facultad, Escuela, Especialidad, Curso/Nivel

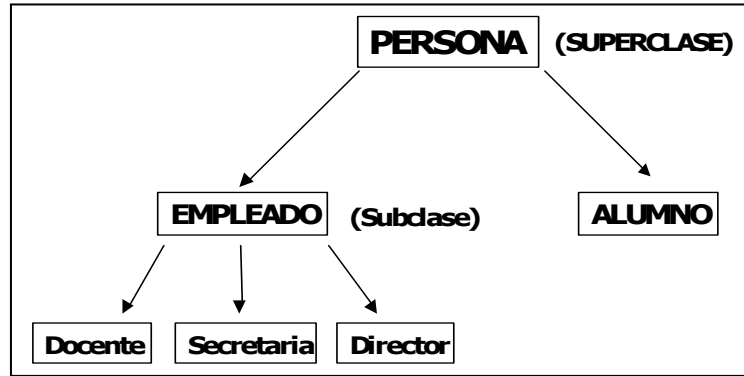


Figura 4.4. Jerarquía de Clases para ejemplo de personal en Investigación.

Un objeto que representa a un director contiene todas las variables de la clase Director, Empleado y Persona. A esto se le conoce como HERENCIA, en donde Director hereda las propiedades de una clase más general.

4.2.4 HERENCIA MULTIPLE

Existen situaciones que no pueden representarse bien en una jerarquía de clases. Si suponemos que queremos distinguir entre docente y secretaria a tiempo completo y tiempo parcial. O queremos diferenciar quienes son directores de proyectos e investigadores. Se podían crear clases para << Directores de proyectos >> y otra para << Investigadores >>. Pero se complicaría aún más si se tratara de clasificar por facultades y por escuelas. En este modelo jerárquico de clases, las clasificaciones que pudieran resultar son una limitación en la estructuración de clases.

El concepto de **herencia múltiple** trata estos problemas en el modelo orientado a objetos, este concepto se refiere a la capacidad de las clases para heredar variables y métodos de múltiples superclases. La relación clase/superclase se representa por un grafo con raíz acíclica dirigido (**DAG**¹) en el que una clase puede tener más de una superclase.

Utilizando un DAG podemos definir propiedades de Docente a tiempo completo o a tiempo parcial en un sitio, como también podríamos definir las propiedades de Docente de la FICA, Docente de la FICAYA, etc. En los datos personales de un docente o Investigador su área de especialización, etc. Así por ejemplo: Docente a tiempo completo de la FICA, con la especialidad de Ing. en Sistemas.

En este modelo de bases de datos orientado a objetos, las superclases se convierten en entidades fuertes, que no tiene dependencia de otras, y las subclases se convierten en entidades débiles que si dependen de otras.

¹ **DAG**: Grafo con raíz acíclica dirigido.

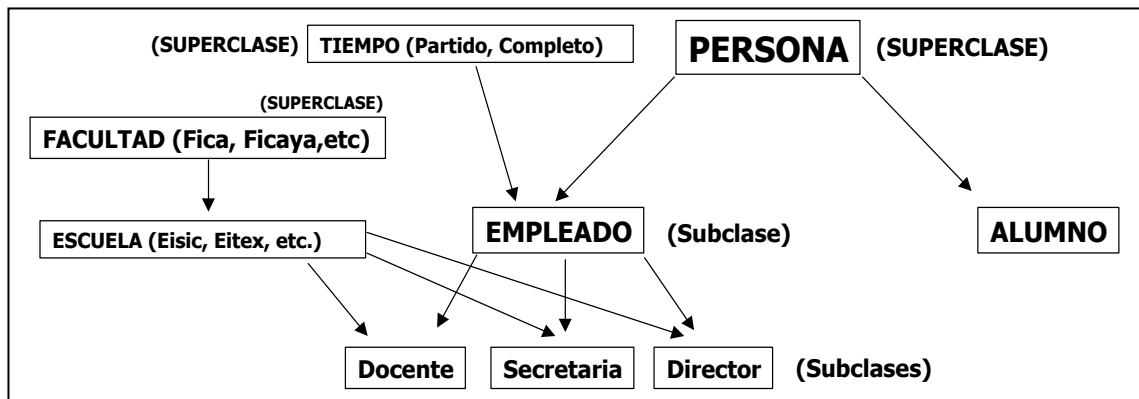


Figura 4.5. CLASE DAG, para el ejemplo del Personal de investigación de la Universidad.

En las bases de datos orientadas a objetos la herencia múltiple, se la interpretaría como una característica de una entidad para lograr una adecuada organización, por ejemplo, las subclases Docente, Secretaria y Director, lograrían una adecuada organización con un campo en donde se especifique la escuela a la que pertenecen. Similar organización se trabajaría para la Subclase Alumno.

4.2.5. CONTENIDO DE OBJETOS

Los objetos que contiene otros objetos se denominan objetos complejos o compuestos. Son muchos los casos en los que se puede ilustrar este caso, se mencionan dos:

El primero, el de una base de datos con el diseño de un sistema de computadores. Cada sistema contiene un conjunto de tarjetas, un conjunto de buses, un conjunto de dispositivos y un conjunto de instrucciones a nivel de máquina. La tarjeta contiene un conjunto de chips y los buses proporcionan un conjunto de interfaces.

El segundo, el de una organización de estudios superior, en donde La institución, Universidad para el ejemplo, tiene varias Facultades, las Facultades, a su vez tiene varias Escuelas o Especialidades y cada Especialidad tiene sus Niveles, sin mencionar los paralelos.

En muchas aplicaciones, un objeto puede estar contenido en varios objetos. En tales casos, la relación de contenido se representa por un DAG en vez de mediante una jerarquía.

4.2.6. ORGANIZACIÓN FISICA

La estructura de las bases de datos orientadas a objetos no presenta la uniformidad de las bases de datos relacionales. Una estructura física sencilla y fácil de manipular parte de una clase básica. Las clases básicas corresponden a los tipos de datos de los lenguajes de programación estándar: entero, flotante, carácter, cadena, etc.

Las instancias de clases que no son básicas pueden representarse mediante campos que contienen el valor del objeto, esto se interpreta como los métodos que permiten acceder o manipular a las variables de la clase.

La estructura física hace que sea posible utilizar registros de longitud fija para implementar una base de datos orientada a objetos. Las aplicaciones pueden requerir o contener tipos de datos que son grandes físicamente y que, por razones prácticas, normalmente se manipulan mediante programas de aplicación especializados.

Estos campos largos utilizan métodos de acceso Verificación de datos de entrada (checkin)/ Verificación de resultados de salida(checkout).

Se puede ilustrar de mejor manera con el siguiente ejemplo: al diseñar un formulario que contenga imágenes incrustadas, estas imágenes siendo una imagen de mapa de bits, la misma que se puede modificar haciendo doble click sobre ella; pasa ahora a ser controlada por un programa que manipula la imagen, puede ser Paint Brush, se procede a realizar cambios en la imagen y al momento de cerrarse Paint se obtiene una nueva versión del objeto o campo largo. Con esta deducción se puede demostrar que Access en sus versiones 97/2000, es una base de datos que maneja a objetos.

4.2.7. CONSULTAS ORIENTADAS A OBJETOS

Los lenguajes de programación orientados a objetos requieren que toda la interacción con objetos sea mediante envío de mensajes. Años atrás representaba serias limitaciones en las aplicaciones de bases de datos.

Actualmente existen bases de datos orientadas a objetos, que ya se soluciona este inconveniente de las consultas que pasan como mensajes parámetros dinámicos de consultas, y permiten crear

consultas temporales, y estas a su vez se consulten para realizar algún tipo de operaciones. A esto puede llamarse cursores en unos casos, Recordset o QueryDef en el aplicativo financiero.

En el Sistema de Gestión de Investigación (Control Financiero) se utilizan en un 90% este tipo de consultas temporales, como en la Asignación presupuestaria a Proyectos y el Control de Gastos.

```
Public Function TOTALIZAR_PARTIDAS_ANUALES(PROYECTO1 As Integer)
Dim BASEDATOS As Database, DBS As Database, qdf As QueryDef
Dim MyTbOrigen As Recordset
Dim select1 As String, from1 As String, group1 As String, having1 As String
Dim strsql As String, UPDATE1 As String, WHERE1 As String
Dim qdf1 As QueryDef
Dim TOTAL1 As Long, ID_PAÑO1 As Integer

Set BASEDATOS = CurrentDb
Set DBS = CurrentDb

select1 = " SELECT PARTIDA_AÑO.ID_PROYECTO, PARTIDA_AÑO.ID_PAÑO, Sum([PARTIDA
PRESUPUESTADA].[MONTO PARTIDA]) AS TOTAL_AÑO"
from1 = " FROM PARTIDA_AÑO INNER JOIN [PARTIDA PRESUPUESTADA] ON PARTIDA_AÑO.ID_PAÑO =
[PARTIDA PRESUPUESTADA].ID_PAÑO"
group1 = " GROUP BY PARTIDA_AÑO.ID_PROYECTO, PARTIDA_AÑO.ID_PAÑO"
having1 = " HAVING PARTIDA_AÑO.ID_PROYECTO=" & PROYECTO1
strsql = select1 + from1 + group1 + having1

' Crea un nuevo objeto QueryDef. Consulta temporal
' Consulta Temporal para sumar los totales de las partidas de un periodo anual
Set qdf = DBS.CreateQueryDef("TEMPORAL", strsql)
' Abre la consulta temporal para encontrar los totales
Set MyTbOrigen = BASEDATOS.OpenRecordset("TEMPORAL", dbOpenSnapshot)
' Se elimina la consulta << TEMPORAL >>
DBS.QueryDefs.Delete qdf.Name

MyTbOrigen.MoveFirst
While (Not MyTbOrigen.EOF)
ID_PAÑO1 = MyTbOrigen![ID_PAÑO]
TOTAL1 = MyTbOrigen![TOTAL_AÑO]
UPDATE1 = "UPDATE [PARTIDA_AÑO] SET [PARTIDA_AÑO].[MONTO PERIODO] =" & TOTAL1
WHERE1 = " WHERE [PARTIDA_AÑO].ID_PAÑO=" & ID_PAÑO1
strsql = UPDATE1 + WHERE1
' Nuevamente se crea una consulta temporal para actualizar
' el total de partidas en el periodo indicado como parámetro
Set qdf1 = DBS.CreateQueryDef("TEMPORAL1", strsql)
qdf1.Execute
Debug.Print qdf1.RecordsAffected
' Se elimina la consulta << TEMPORAL1 >>
DBS.QueryDefs.Delete qdf1.Name
MyTbOrigen.MoveNext
Wend

' Se destruyen los objetos temporales, donde DBS y BASEDATOS en una Base
de datos temporal
Set DBS = Nothing
Set BASEDATOS = Nothing
End Function
```

Figura 4.6. Ejemplo de una Consulta orientada a objetos.

El párrafo anterior es uno de los 3 procedimientos de Cierre de Asignación Presupuestaria que TOTALIZA el monto de un Período Presupuestario de un Proyecto. En esta función se crean variables u objetos, en donde se ejecutan instrucciones SQL, para actualizar datos, y luego

desaparecen las consultas porque no hace falta tener almacenadas consultas con parámetros dinámicos, (PROYECTO1, ID_PANO1), que para otro período anual o proyecto va a cambiar.

Estas consultas temporales ocupan espacio en la base de datos, inclusive es posible visualizarlas en Access 97/2000, si en la ejecución se establece un punto de ruptura que detiene la ejecución del programa en una línea determinada, recomendable antes de la eliminación de la Consulta, y se hace un seguimiento paso a paso.

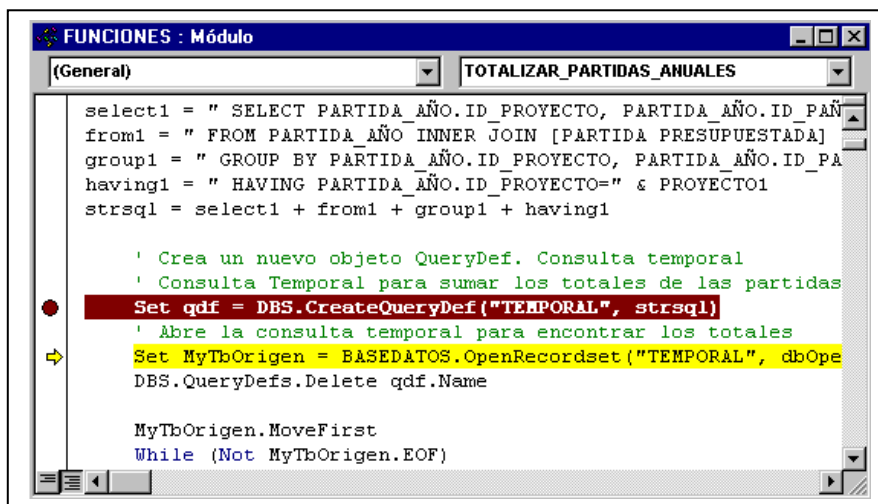


Figura 4.7. Uso del Depurador Visual Basic de Access.

En ese instante se puede ver en el área de consultas y verificar la existencia de la consulta temporal <<TEMPORAL>>, y ver los campos de las tablas y parámetros que contiene la consulta temporal, o la vista de código SQL.

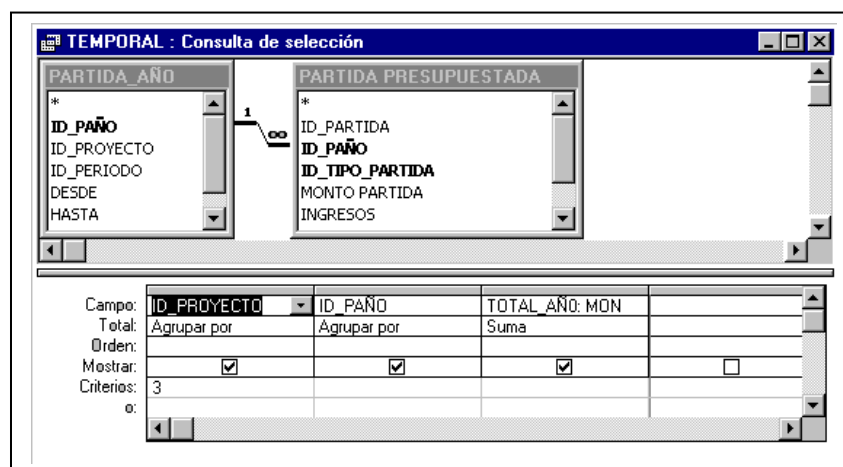


Figura 4.8. Consulta creada con parámetros.

4.2.8. MODIFICACION DE ESQUEMAS

Los cambios en la estructura de un objeto en una base de datos orientada a objetos es demasiado compleja, citemos algunos ejemplos de los inconvenientes:

- **Adición de una nueva clase:** La adición de una nueva clase en una base de datos orientada a objetos, implica que debe tener una jerarquía entre las clases, subclases o DAG, y debe resolver los problemas de herencia, es decir debe tener un espacio, reconocimiento y funcionalidad. Se debe tomar en cuenta que puede llegar a ser una SUPERCLASE de otras, que se creen o relacionen.
- **Eliminación de una clase:** La eliminación de una clase en una base de datos orientada a objetos se complica porque puede ser una clase padre de otras subclases, y los cambios que se generen para esta repercutirían en las subclases dependientes o hijas.
- **Modificación de una definición de clase:** Una clase podría variar en sus variables o atributos, esta nueva definición afecta directamente a las subclases, en donde se hace referencia a las variables anteriores y caería en la inconsistencia. Se debe señalar que estos cambios deben ser manejados cuidadosamente, y verificar cada una de las clases afectadas.

Todos estos ejemplos de modificación de esquemas, o reestructuración de clases en una base de datos orientada a objetos, requieren de un minucioso seguimiento, por lo que más adelante se hace un estudio de la complejidad de las clases.

Mientras tanto para el ejemplo, supóngase que se tiene un campo "TOTAL_MES" en una consulta <<TOTALES>>, esta consulta se utiliza para calcular los totales mensuales de facturas, el solo cambio de la forma de calcular ese total afectaría al gran total que se desea obtener. Imagínese que se tiene dos formularios que utilizan la misma consulta, y se requiere que en el primer formulario calcule el total sin IVA, y el segundo formulario calcule el mismo total pero con IVA. Los cambios que se hagan a la consulta afectarían a ambos formularios, por ser dependientes de la misma consulta.

4.3. METRICAS ORIENTADAS A OBJETOS PARA MEDIR SOFTWARE.

El analizar las métricas que existen para la ISO, permitirá a los ingenieros de software modelar sus sistemas planificando y siguiendo las actividades requeridas dentro del modelado de objetos, fundamentados en bases matemáticas, que permitan evaluar la calidad del diseño y análisis a nivel de clases, en la descomposición de las clases, en la organización de clases que proporcionan información acerca del nivel de esfuerzo requerido para implantar un sistema.

Las métricas son un elemento clave para valorar la calidad del software en base a un conjunto de reglas bien definidas. El Diseño Orientado a Objetos (DOO) a diferencia del diseño estructurado no cuenta con un conjunto de métricas desarrolladas en base a fundamentos matemáticos (Fetcke 1995). Un conjunto de métricas orientada a objetos que ha tenido mayor aceptación (aunque criticadas por su falta de rigurosidad matemática) es la planteada por Chidamber y Kemerer (1991) quienes utilizan el método de Booch (1996). Este conjunto de métricas esta basada en clases. Considerando que una clase es la unidad fundamental para el diseño. Por consiguiente las métricas a nivel de clases resultan sumamente valiosas para estimar la calidad de un diseño (Pressman 1998).

Se han propuesto una amplia gama de métricas para el diseño orientada a objetos, pero muchas de ellas se desvían del verdadero significado de orientado a objetos, caso de algunos programadores en lenguaje orientado a objetos no utilizan la herencia, esto no se puede obviar porque no se cuenta hasta la actualidad con formas de evaluar los productos desarrollados.

Conforme nos movemos en la cuarta era, continúan intensificándose los problemas asociados con el software de computadoras:

- La sofisticación del hardware ha dejado desfasada nuestra capacidad de construir software que pueda explotar el potencial del hardware.
- Nuestra capacidad de construir nuevos programas no puede dar abasto a la demanda de nuevos programas.
- Nuestra capacidad de mantener los programas existentes está amenazada por el mal diseño y el uso de recursos inadecuados.

Como respuesta a la crisis del software, muchas industrias están adoptando prácticas de ingeniería del software según las técnicas de métricas de diseño orientado a objetos.

4.3.1. DESARROLLO

Es difícil establecer un método dentro de la ingeniería de software (**IS**) que permita definir y establecer un conjunto de métricas para el diseño orientado a objetos. El objetivo es dar a conocer algunas investigaciones en este campo que ayuden a plantear nuevos desarrollos y que el paradigma vaya a cubrir y proporcionar una serie de beneficios significativos en la calidad del software, abarcando desde implantar líneas de construcción de las métricas hasta establecer fundamentos matemáticos para la construcción de las métricas para el diseño orientado a objetos (**DOO**).

Fundamentos

Una **métrica** es una medida esencial para comprender que afecta a la calidad, oportunidad, utilidad y funcionalidad del software y en la mejora de los procesos y productos del software.

Es decir, son evaluaciones objetivas de los sistemas en forma de mediciones cuantificadas, permitiendo a los usuarios y a los desarrolladores de Sistemas de Información medir el desempeño del sistema e identificar problemas en forma oportuna.

Los objetivos de las métricas vienen de los objetivos del negocio. Con las métricas se pretende:

1. Comprender como nos permiten caracterizar, evaluar, predecir y mejorar los procesos y productos software.
2. Aprender como aplicar las métricas del software.
3. Conocer y valorar los efectos de las métricas en:
 - a) Mejorar el proceso de desarrollo.
 - b) Mejorar la estimación del software.
 - c) Mejorar el seguimiento del proyecto.
 - d) Minimizar el tiempo de desarrollo.
 - e) Minimizar el coste de desarrollo.
 - f) Mejorar la calidad del software.
 - g) Mejorar la productividad.
4. Aprender a realizar un plan de métricas que comprenda:
 - a) ¿Porqué se necesita la medición?: ¿cuáles son los productos, procesos, recursos y objetivos del proyecto que deben ser considerados por las métricas?

- b) ¿Qué va a ser medido: ¿qué atributos de proyectos, procesos, productos van a ser caracterizados cuantitativamente?
- c) ¿Quién estará implicado: ¿quién definirá las métricas, recogerá los datos, analizará los datos y presentará los resultados y quién necesita ver los resultados?
- d) ¿Dónde se necesita la medición?: ¿qué procesos y subprocesos (tales como aseguramiento de la calidad o gestión de configuración) van a ser instrumentados con métricas?
- e) ¿Cuándo serán tomadas las medidas?: ¿cuándo durante el desarrollo o mantenimiento del proceso serán recogidas las métricas y con qué frecuencia?
- f) ¿Cómo serán recogidas y analizadas las métricas?: ¿qué técnicas manuales o automatizadas serán utilizadas para capturar, almacenar y analizar las métricas?

Las respuestas a estas cuestiones son fundamentales para el éxito de la medición. Conocer por qué estamos midiendo nos da el objetivo y razón para capturar y evaluar los datos. Conocer qué, dónde y cuándo medir nos dice como la medición estará mezclada con las actividades cotidianas de desarrollo. Y conociendo como medir y quién estará implicado, explica quién es responsable de la recogida, almacenamiento y presentación de los datos.

Las Métricas de Ingeniería de Software Orientado a Objetos (ISOO) son unidades de medida que se usan para caracterizar:

- **ISOO** de diseño de productos, p.e. planes, código fuente, y casos de prueba,
- **ISOO** de diseño de procesos, p.e. las actividades de análisis, diseño, y codificación, y
- **ISOO** organizacional de las personas, p.e. la eficacia de un probador individual, o la productividad de un diseñador individual.

Las métricas de ISOO se diferencian debido a: localización, encapsulamiento, ocultamiento de información, herencia, y técnicas de abstracción de objetos.

La **localización** es el proceso de descomposición funcional alrededor de la información de los procesos, la información de los datos y de los objetos. Esta basado en la funcionalidad de los componentes y de las interrelaciones, reconociendo a un objeto como la unidad básica del software. La localización entre la funcionalidad y un objeto no es una relación uno-a-uno, ya que una función puede involucrar algunos objetos, y un objeto puede proveer muchas funciones.

El **encapsulamiento** es el empaquetado de una colección de ítems, incluyendo datos y funciones que se encuentran en un objeto, como protección para aislar los detalles funcionales del objeto. El

encapsulamiento tiene dos impactos sobre las métricas: la unidad básica no es un subprograma más bien es un objeto, y que puede ser reutilizado de acuerdo a la caracterización del sistema.

El **ocultamiento de información** es la supresión de los detalles operativos de un componente de un programa. La idea general es mostrar solamente la información necesaria para acoplar a las metas inmediatas. El encapsulamiento y ocultamiento de la información no son lo mismo, ya que un ítem puede ser encapsulado pero puede ser visible. Aquellas métricas que proporcionen una indicación del grado en que se ha logrado el ocultamiento proporcionarán una indicación de calidad del diseño OO.

La **herencia** es un mecanismo de que un objeto pueda adquirir características de uno o más objetos. Muchas métricas de ISOO se basan en la herencia, ya que pueden adquirir especificaciones de la jerarquía de herencia establecida, la capacidad de derivar un objeto de un superobjeto existente.

Las **técnicas de abstracción** es un mecanismo que enfoca lo esencial de un concepto o ítem, ignorando los detalles innecesarios. Se puede tener la abstracción funcional, de datos, procesos y objetos, considerando objetos como entidades de nivel superior que proveen más específicas vistas de un concepto o ítem. Así son tres las comúnmente vistas usadas en la definición de clases.

Una **clase** es la implantación de un tipo de objeto. Especifica una estructura de datos y los métodos operativos permisibles que se aplican a cada uno de sus objetos.

Mientras una **metaclass** es una superclase que instancia a estas clases, suministrada algunos parámetros que son usados para crear una clase. Una metaclass, entonces es una abstracción de éstas instancias.

Las métricas de ISOO están relacionadas al número de instancias por clase y por aplicación, al número de clases parametrizadas por aplicación, y a la cobertura de clases parametrizadas a clases no parametrizadas (**polimorfismo**).

4.3.1.1. Líneas para las métricas

Primeramente, se escribe el nombre de la métrica seguido de su correspondiente abreviación. Luego, se detalla la siguiente información:

Apreciación General._ proporciona una visión general de su significado y utilidad.

Base Teórica._ anexa la base teórica en la que se fundamenta la métrica para su validez.

Cómputo._ describe cómo calcular la métrica.

Rango Válido._ Refiere el rango de valores válidos que son dados como resultado del cálculo. Un resultado que cae fuera del rango, indica que la métrica no ha sido correctamente calculado.

Rango Nominal._ si la métrica falla dentro del rango nominal, entonces la métrica ha logrado su objetivo de calidad.

Análisis._ incluye información útil para interpretar el significado de los valores nominales.

4.3.1.2. Representaciones de la terminología OO

La definición formal de los principales términos utilizados en el ISOO, considerando en definiciones de conjuntos y utilizando la simbología propuesta por Chidamber & Kemerer (1991) se detalla a continuación.

4.3.1.3. Objeto

Un objeto se puede representar como:

Notación:

$$X = \langle x, p(x) \rangle \quad p(x) = \{Mx\} \cup \{Ax\} \cup \{Ix\}$$

donde:

x: es la esencia individual

p(x): colección finita de sus propiedades

{Mi}: es el conjunto de métodos

{Ai}: es el conjunto de atributos, y

{Ii}: es el conjunto de variables de instancia del objeto i.

La **complejidad de un objeto** puede ser definida como la cardinalidad de sus propiedades. Luego la complejidad se representa por:

$$X = \langle x, p(x) \rangle = |p(x)|$$

Donde $|p(x)|$ es la cardinalidad de p(x). X posee un estado, comportamiento e identidad mediante el cual se representa en el sistema.

En la terminología OO las variables de instancia, los atributos y los métodos constituyen las propiedades. Cabe notar, que las variables de instancia tienen un sentido intuitivo siendo que los métodos que operan sobre las variables de instancia poseen un grado de similaridad.

Un conjunto de objetos que comparten una estructura y comportamiento común constituyen una clase. Para denotar una clase se utiliza letras mayúsculas. Cuando se requiera analizar las propiedades específicas de un objeto se hace referencia a la notación definida.

Herencia

Se dice que Y hereda de X si las propiedades pueden ser definidas como:

Sea $X = \langle x, p(x) \rangle$ & $Y = \langle y, p(y) \rangle$ dos objetos.

$$p(x) = \{Mx\} \cup \{Ax\} \cup \{Ix\}$$

$$p(y) = \{Mx\} \cup \{My\} \cup \{Ax\} \cup \{Ay\} \cup \{Iy\}$$

Donde $\{Mi\}$ es el conjunto de métodos, $\{Ai\}$ es el conjunto de atributos, y $\{Ii\}$ es el conjunto de variables de instancias del objeto i.

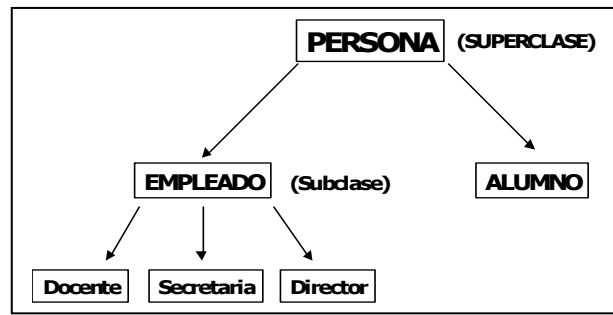


Figura 4.9. Herencia.

Ejemplo: Un objeto que representa a un director contiene todas las variables de la clase Director, Empleado y Persona. A esto se le conoce como **HERENCIA**, en donde director hereda las propiedades de una clase más general.

Polimorfismo

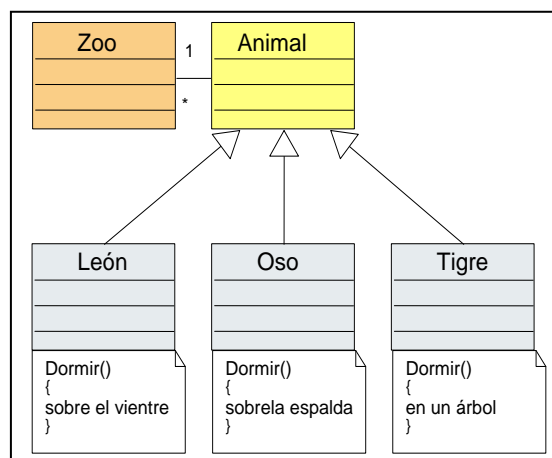


Figura 4.10. Polimorfismo o Comportamiento.

Ejemplo: Todo animal duerme, pero cada clase lo hace de forma distinta, sobre el vientre, la espalda, en un árbol.

Se dice que X y Y tienen polimorfismo si los objetos pueden ser definidos como:

Sea $X = \langle x, p(x) \rangle$ & $Y = \langle y, p(y) \rangle$ dos objetos.

$$p(x) = \{M_x\} \cup \{A_x\} \cup \{I_x\}, \quad 1 \text{ si } i=j$$

$$p(y) = \{M_x\} \cup \{M_y\} \cup \{A_x\} \cup \{A_y\} \cup \{I_y\} \text{ donde } (M_i, M_j) = 0 \text{ si } i < j$$

Si algún método i del conjunto de métodos de $\{M_x\}$ tiene el mismo nombre que algún método j del conjunto de métodos de $\{M_y\}$ existe polimorfismo, en caso contrario, no se da el polimorfismo. Esto también es aplicable a los atributos. Donde $\{M_i\}$ es el conjunto de métodos, $\{A_i\}$ es el conjunto de atributos y $\{I_i\}$ es el conjunto de variables de instancia del objeto i . Lo anterior es una definición de polimorfismo a través de la herencia.

Encapsulamiento

Este concepto en la fase del diseño se refiere más al ocultamiento de la información, el cual se define como:

Sea $X = \langle x, p(x) \rangle$;

$$p(x) = \{M_x\} \cup \{A_x\} \cup \{I_x\},$$

$$1 \text{ si } M_i \text{ no es visible a } C_j$$

donde: $(M_i, C_j) = 0$ de otra forma.

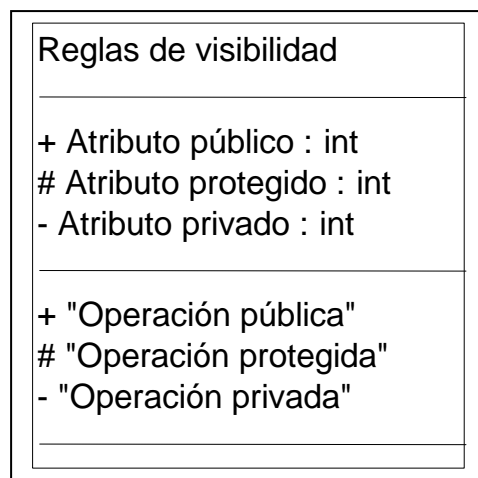


Figura 4.11. Encapsulamiento.

Encapsulación

Para toda clase C_j un método se cuenta como cero si este puede ser usado por otras clases y 1 en caso contrario. Donde $\{M_i\}$ es el conjunto de métodos $\{A_i\}$ es el conjunto de atributos y $\{I_i\}$ es el conjunto de variables de instancia del objeto i , y C_j es el objeto j del conjunto de objetos C (otra clase).

Acoplamiento

Se dice que X actúa sobre Y si la ocurrencia de Y es afectada por X , donde la ocurrencia es definida como los estados ordenados cronológicamente a través del tiempo.

Sea $X = \langle x, p(x) \rangle$ y $Y = \langle y, p(y) \rangle$ dos objetos

$$p(x) = \{M_x\} \cup \{I_x\};$$

$$p(y) = \{M_y\} \cup \{I_y\}$$

Donde $\{M_i\}$ el conjunto de métodos y $\{I_i\}$ es el conjunto de variables de instancia del objeto i .

De acuerdo a esta definición cualquier acción de $\{M_x\}$ en $\{M_y\}$ o en $\{I_y\}$ constituye un acoplamiento.

Cohesión

Se define como la intersección de las propiedades de dos objetos X y Y .
Sea $X = \langle x, p(x) \rangle$ y $Y = \langle y, p(y) \rangle$ dos objetos.

$$s(X, Y) = p(x) \cap p(y)$$

Siguiendo el principio general de definición en términos de conjuntos, el grado de similaridad de los métodos de los objetos puede ser definido como la intersección de los conjuntos de variables de instancia que son utilizados por los métodos.

$$s(M_1, M_2, \dots, M_n) = \{M_1\} \cap \{M_2\} \cap \dots \cap \{M_n\}$$

Donde $s()$ = grado de similaridad y $\{M_i\}$ es el conjunto de variables de instancia usados por el método M_i .

El alto grado de similaridad de métodos, hace más cohesivo los métodos y mayor grado de encapsulación de los objetos.

4.3.1.4. Operaciones Definidas

Se considera una clase como un conjunto sobre la cual se pueden realizar las siguientes operaciones:

Unificación de Clases (UC)

La unión de dos clases no es una expresión del modelo OO. El resultado de la unión de dos clases es una nueva clase que combina todas las propiedades de las clases unidas. Por ejemplo, todos los atributos y métodos, las propiedades que son comunes sólo aparecen en la nueva clase una vez. Esta propiedad es muy similar a la unión de conjuntos.

Intersección de Clases (IC)

El resultado de la intersección de dos clases es una nueva clase que consta de sólo las propiedades comunes a ambas clases.

4.3.1.5. Unión de Jerarquías

Dos jerarquías independientes son unidas a través de una nueva clase (la cual puede tener sus propias propiedades) formando una nueva jerarquía, las jerarquías independientes conservan sus propiedades sin ninguna alteración.

Unión de Jerarquía de Herencia

Dos jerarquías de herencia independientes son unidas para formar una nueva jerarquía de herencia definiendo esa jerarquía con una sola clase raíz, la cual consta de todas propiedades comunes de las clases unidas a través de la herencia y las subclasses constan de las propiedades que no se repiten.

4.3.1.6. Propiedades de las métricas a nivel de clases

Uno de los principales problemas de las métricas existentes es la falta de propiedades teóricas, que permitan calcular o estimar la complejidad del modelado del DOO. La escala de razón tradicional no es aplicable a las métricas de complejidad para el DOO, ya que no se cumple el axioma de Archimedian. Por lo tanto, para contar con una base formal de construcción de las métricas se

acepta como pautas para establecer una escala de medición los axiomas planteados por Fetcke (1995).

Formalmente se considera las siguientes operaciones:

- La unificación (U) representa la unificación de objetos de las clases A y B. A través de esta operación se obtiene una nueva clase $A \cup B$, la cual consta de un conjunto de propiedades sin repetición. La clase unificada $A \cup B$ hereda todas las propiedades de las clases originales. También se utiliza esta notación para representar la unión de jerarquías y de la unión de jerarquías de herencia.
- Otra operación es la intersección de clases (\cap) que origina una nueva clase constituido de todas las propiedades comunes de las clases de que está hereda.
- Se define una clase vacía \emptyset como una clase que no tiene propiedades, esto implica que de esta clase no se puede heredar atributos ni métodos.
- Se dice que una clase A es subconjunto (c) de B, si B tiene a todas las propiedades de A, esto se da mediante la relación de herencia donde B es la superclase y A subclase.
- $B - A$ comprende a todas las propiedades de B que no están en A, este caso se da en la relación de herencia, donde A es la superclase y B subclase. La clase B tiene atributos propios no heredados.

Dadas las clases A, B y C se definen las siguientes propiedades:

1. $A \cup A = A$ (idempotencia)
2. $A \cap A = \emptyset$ (disyunción de la intersección)
3. $A \cup B = B \cup A$ (conmutatividad de la unificación)
4. $A \cap B = B \cap A$ (conmutatividad de la intersección)
5. $(A \cap B) \cap C = A \cap (B \cap C)$ (asociatividad con respecto a la intersección)
6. $(A \cup B) \cup C = A \cup (B \cup C)$ (asociatividad con respecto a la unificación)
7. $(A \cap B) \cup C = (A \cup C) \cap (B \cup C)$ (distributividad con respecto a la unificación)
8. $(A \cup B) \cap C = (A \cap C) \cup (B \cap C)$ (distributividad con respecto a la intersección)
9. $A \subset B$ y $B \subset C$ entonces $A \subset C$ (Transitividad).

Al aplicar la métrica μ según la definición de escala un sistema de relaciones empíricas (punto de vista) al ser transformado a un sistema de relaciones formales debe preservar todas las relaciones y operaciones.

Se definen las siguientes propiedades al aplicar la métrica:

Sean X, Y, Z objetos, μ métrica.

P1. Si \emptyset es una clase vacía entonces $\mu(\emptyset)=0$

P2. $\mu(X) = 0$

P3. $|\mu(X)| = c$; existe un número entero finito para la métrica (cardinalidad), c pertenece R

P4. Si $X \subset Y$ entonces $\mu(X) \leq \mu(Y)$

P5. $\mu(X) \neq \mu(Y)$; cada objeto no puede tener el mismo valor para una métrica, dada su posición.

P6. $\mu(X) = \mu(Y)$; dos objetos pueden tener el mismo valor de métrica.

P7. $\mu(Y) = \mu(X \cup Y)$; la unificación de dos objetos aumenta la complejidad.

P8. $\mu(X) \cup \mu(Y) = \mu(X \cup Y)$; la unión disminuye la complejidad.

P9. $\mu(X) \cap \mu(Y) = \mu(X \cap Y)$; la intersección disminuye la complejidad.

Las anteriores operaciones y propiedades satisfacen la escala modificada de Fetcke (1995).

4.3.1.7. El conjunto de métricas CK

A continuación se define el conjunto de métricas del diseño orientado a clases propuestas por Chidamber y Kemener, las mismas que han llegado a tener mayor aceptación, a las cuales suele llamarse con el nombre de *conjunto de métricas CK* para sistemas OO.

Métodos ponderados por clase (MPC)

Supóngase que se define n métodos de complejidad c_1, c_2, \dots, c_n para una clase C . La métrica de complejidad específica que se selecciona debe de normalizarse de tal modo que la complejidad nominal para un método tome el valor de 1,0.

$$\mathbf{MPC} = \sum C_i, \text{ para } i = 1 \text{ hasta } n.$$

El número de métodos y su complejidad es un indicador razonable de la cantidad de esfuerzo necesaria para implementar y comprobar una clase.

Arbol de profundidad de herencia (APH)

Esta métrica se define como "la longitud máxima desde el nodo hasta la raíz del árbol" (CHI94). En la figura siguiente el valor de APH es 4. Los valores grandes de APH implican que se pueden reutilizar muchos métodos, pero lleva a una mayor complejidad de diseño.

Número de descendientes (NDD)

La jerarquía de subclases subordinadas a una clase se denominan sus descendientes. Al incrementarse los descendientes, se incrementa la reutilización, pero la abstracción representada por la clase predecesora puede verse diluida, ya que algunos descendientes no son miembros propios de la clase predecesora, y mientras va creciendo la profundidad de descendientes, se incrementan sustancialmente las pruebas.

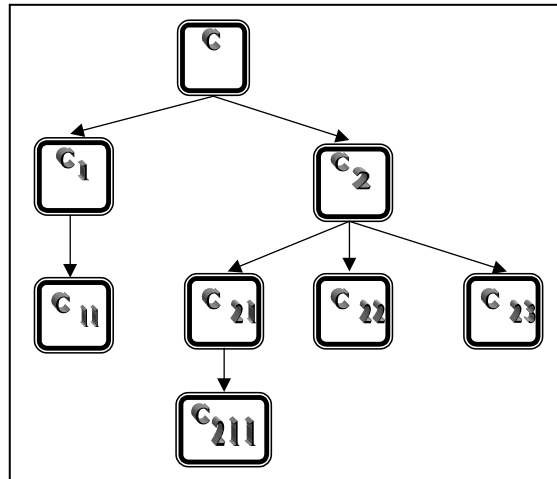


Figura 4.12. Una Jerarquía de Clases

Acoplamiento ente clases objeto (ACO)

ACO es "el número de colaboraciones enumeradas para una clase en su tarjeta índice del modelo de Clases Responsabilidades Colaboraciones CRC" (Press98, Pág. 443).

La clase padre con sus responsabilidades y colaboraciones, transformado a objetos potenciales, proveerá la información necesaria para completar sus responsabilidades usando las jerarquías que la reutilización de la clase que vaya descendiendo de acuerdo al nivel de acoplamiento.

Respuestas para una clase (RPC)

RPC se define como el número de métodos existentes en el conjunto de respuestas "es un conjunto de métodos que pueden ser ejecutados potencialmente en respuesta a un mensaje recibido por un objeto de esa clase" (CHI94); a medida que crece RPC de acuerdo a la profundidad del árbol, el esfuerzo necesario para comprobación crece también, porque la sucesión de comprobación va creciendo, y la complejidad global de diseño de la clase crece también.

La siguiente función responde de diferentes maneras al momento de ser utilizada al momento de actualizar el Stock del Presupuesto, el parámetro OPERAR de tipo String, puede ser "+" para eliminar un registro del Presupuesto e incrementar al Stock la cantidad eliminada, y "-" para actualizar el Stock luego de un gasto del presupuesto o disminución.

```
Function Actualizar_Stock_Presupuesto_Pedido(ITEM_PRESUPUESTO As Integer, CANTIDAD As Integer, OPERAR As String)
Dim DBS As Database
Dim STRSQL As String
Set DBS = CurrentDb
STRSQL = "UPDATE Presupuesto_Mes_Partida set Presupuesto_Mes_Partida.STOCK=Presupuesto_Mes_Partida.STOCK" &
OPERAR & " " & CANTIDAD & " ,
Presupuesto_Mes_Partida.PREAUTORIZADO=Presupuesto_Mes_Partida.PREAUTORIZADO" & OPERAR & " " &
CANTIDAD & " WHERE Presupuesto_Mes_Partida.ID_ITEM_PRESUPUESTO=" & ITEM_PRESUPUESTO
DBS.Execute STRSQL
DBS.Close
End Function
```

Figura 4.13. Ejemplo de una Consulta parametrizada.

Carencia de cohesión en los métodos (CCM)

Existe CCM si no existen variables de instancia (atributos) donde ningún método accede a los mismos atributos. Por ejemplo, si una clase tiene 6 métodos, donde cuatro de los métodos tienen en común uno o más atributos, por consiguiente CCM es igual a 4, en caso contrario si ninguno de los métodos accede a los mismos atributos CCM es igual a 0.

Es recomendable que CCM tienda a cero, ya que disminuye la complejidad del diseño de clases.

Esto tiene que ver mucho con polimorfismo, ya que se pueden utilizar los mismos métodos con diferente número de atributos, e incluso tipos de datos. Por ejemplo, si creo una clase debería tenerse un menor número de atributos fijos, pero cualquier cantidad de atributos opcionales, para mantener un buen CCM.

4.3.2. Conclusiones

Luego de algunas revisiones del análisis del Diseño de Software Orientado a Objetos y las Métricas de Software Orientado a Objetos se ha determinado que se focalizan al estudio de las características de los objetos y sus primitivas que son las clases; estas métricas hacen énfasis en analizar la complejidad de las clases, sus elementos (atributos y métodos).

Con este enfoque de desarrollo de software lo que se pretende es medir el nivel de esfuerzo requerido para implementar un sistema, utilizando a los objetos como referencia para su definición,

programación, documentación y reutilización en aplicaciones futuras, por las características apropiadas que presenta.

4.3.3. Beneficios Esperados

Debido a reutilizabilidad de los objetos, el desarrollo de software orientado a objetos permite que se logre una reducción en tiempo y costo de escribir software.

Estos métodos aumentaran el involucramiento de los usuarios, ya que los objetos son más fáciles de entender y son más naturales, hasta cierta parte no nos preocuparemos de cómo hacerlos sino de cómo utilizarlos.

4.4. MODELAMIENTO UML.

La notación UML (no hay que confundir con las metodologías que utilizan dicha notación), se ha convertido desde finales de los 90 en un estándar para modelar con tecnología orientada a objetos todos aquellos elementos que configuran la arquitectura de un sistema de información y, por extensión, de los procesos de negocio de una organización. De la misma manera que los planos de un arquitecto disponen el esquema director a partir del cual levantamos un edificio, los diagramas UML suministran un modelo de referencia para formalizar los procesos, reglas de negocio, objetos y componentes de una organización. La interacción de todos estos elementos es una representación de nuestra realidad.

Nuestros límites para entender esta realidad están en nuestro lenguaje. El mundo es la suma total de lo que podemos definir, organizar y visualizar. Cabe preguntarse ¿de qué manera un modelo en UML representa nuestra experiencia?. Enseñar a utilizar un lenguaje formal siempre es problemático. Es necesario mostrar como este lenguaje puede ser aplicado a la realidad tal como la conocemos y tal como la compartimos con los demás.

No es mediante el descubrimiento de nuevas objetos y de sus múltiples conexiones que avanzamos en las respuestas a nuestros interrogantes cuando modelamos un dominio, sino mediante la disolución de las contradicciones que existen entre los términos (conceptos) ya conocidos y, en último caso, mediante la reducción de su número despejando aquellos conceptos que no añaden valor a la comprensión de dicho dominio.

Reconsiderar lo obvio, desenmascarar presunciones, desambiguar conceptos conocidos, todo en busca de la simplicidad y la usabilidad.

La tecnología orientada a objetos persigue el antiguo principio del divide y vencerás. Su objetivo es descomponer la complejidad en partes más manejables y comprensibles. No parece que esto sea algo novedoso con respecto a la tradicional descomposición funcional de los métodos estructurados. Sin embargo, la gran diferencia reside en aplicar la dualidad *estructura-función* en pequeñas unidades capaces de comunicarse y reaccionar en base a la aparición de una serie de eventos. El esquema dominante de la separación de estructuras de datos y funciones (bases de datos y programas) está amenazado pero se resiste a desaparecer.

Mucha gente cree que la principal utilidad de la orientación a objetos es la reutilización del código para conseguir un desarrollo más rápido de las aplicaciones (Rapid Application Development). Si hay algo que caracteriza un entorno de desarrollo actual es la constante del *cambio*. Todo proyecto

que sobrepase los tres meses está amenazado por la aparición de nuevas plataformas más exigentes, la extinción de herramientas sin previo aviso y, de manera sistemática, por la rotación del personal crítico encargado del proyecto. También está sometido, como no, a los cambios de requerimientos del cliente que a su vez están plenamente justificados por la readaptación de sus procesos de negocio a un mercado inestable.

Ante esta incertidumbre, el mayor desafío de una metodología de desarrollo es su adaptación para el cambio. Esto significa crear modelos que faciliten la comunicación entre todos los agentes involucrados en el sistema en construcción; que hagan visible la trazabilidad entre la concepción de los componentes, su especificación, implementación e instalación; significa el diseño de arquitecturas que faciliten la gestión de las dependencias entre estos componentes, que sean en fin, fácilmente reemplazables por otros más optimizados o bien por componentes que aporten una mayor funcionalidad y/o facilidad de uso.

La dinámica de cambio no se desarrolla en la ingeniería del software con la misma velocidad vertiginosa con que nos tiene acostumbrados la tecnología del hardware. La clave reside en que a diferencia de la electrónica, en los dominios del desarrollo de software no existe un vocabulario unificado. Desde la fase de concepción de un sistema a la instalación de sus componentes hay que mapear entre sí una gran diversidad de lenguajes orientados al análisis, diseño, código ejecutable, esquemas de bases de datos, componentes de páginas web, entre otros. Esta distancia entre la concepción y la usabilidad de un producto o de un proceso de negocio, exige cada vez más la capacidad de cooperación y comunicación de un equipo interdisciplinar muy especializado.

Existen muchas fuentes de información que ha sido fuente constante de información en cuanto al Modelado Orientado a Objetos.

Desde principios de los 90, los artículos publicados en el **Journal of Object Oriented Programming** (JOOP) por **James Odell, James Rumbaugh, Grady Booch, Desmond d'Souza, Bertrand Meyer, Steve Cook, John Daniels, Sally Shlaer** y **Stephen J. Mellor** entre otros, han sido una constante fuente de conocimiento.

Publicaciones pioneras como el *Object Oriented Technology, A Manager's Guide* de **David A. Taylor**, en su primera edición de 1990 y en la segunda ampliada de 1998, han tenido una gran influencia en como abordar la presentación didáctica. También los libros de **Peter Coad** et al, *Object Oriented Analysis, Design and Programming*, *Object Models* y *Java Modeling Color with UML*, han sido de una ayuda extraordinaria. La obra enciclopédica *The Unified Modeling Language: Reference Manual* de **Rumbaugh & Jacobson & Booch**, es un punto de referencia constante.

Sin duda, uno de los autores más influyentes ha sido **Martin Fowler**. Su primer libro Analysis Patterns continua siendo una referencia clave. Posteriormente, la primera edición de UML Distilled en 1997 y su última edición ampliada en 2000, se ha convertido en el libro de cabecera de UML. Otro clásico por la excelencia de su trabajo es Applying UML and Patterns de **Craig Larman** que en su segunda edición aparecida en verano de 2001 se ha superado a sí mismo. También recientes y con muy buen material que ha sido incorporado a la guía, tenemos los libros de **Wendy & Michael Boggs**, Mastering UML with Rational Rose, de **Alistair Cockburn**, Writing Effective Use Cases; de **Scott W. Ambler**, The Object Primer segunda edición; y de **John Chessman & John Daniels**, UML Components, una de las novedades más interesantes de 2001.

4.4.1. Modelos para construcción de Software Orientado a Objetos

Las partes esenciales para el desarrollo de Sistemas de Información son: los Procesos, las Herramientas pero sobre todo para entender los que se va a hacer se necesita de una **NOTACION**, que ayude al desarrollo de Software con Calidad.

Para manejar la complejidad de debe hacer una abstracción del problema enfocando los procesos principales del Negocio, este modelo captura las partes esenciales del sistema para trasladarlo al Sistema computacional.

Los pasos a seguir se determinan con el siguiente esquema de trabajo:

A) PLAN DIRECTOR

Debe existir un plan general director, que permita cuantificar los esfuerzos en cada etapa de desarrollo de un Sistemas de Información, en donde se correlacionen las acciones en función de los objetivos del desarrollo de una aplicación, es equivalente a un cronograma de trabajo.

Esfuerzo en el desarrollo

Se miden por un lado las acciones a ejecutar o etapas: Concepción, Formalización, Construcción, Transición, en función de lo que se quiere finalizar: Misión, Modelo, Prototipo, Componentes y Certificación.

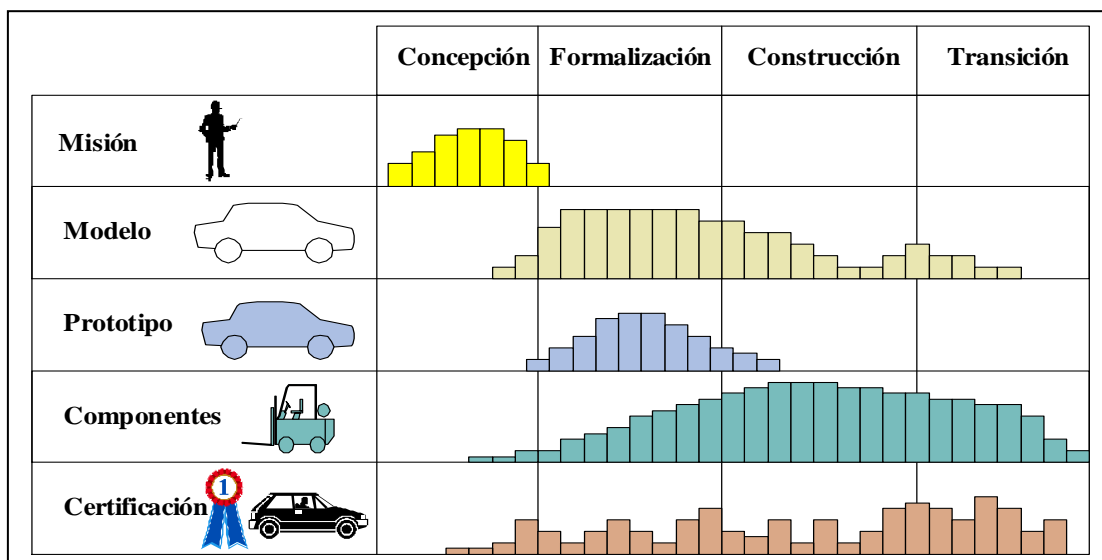


Figura 4.14. Parámetros que se consideran en el Esfuerzo de desarrollo.

Concepción y Formalización

Permiten plasmar una realidad, en función de un esquema de trabajo a ejecutar.

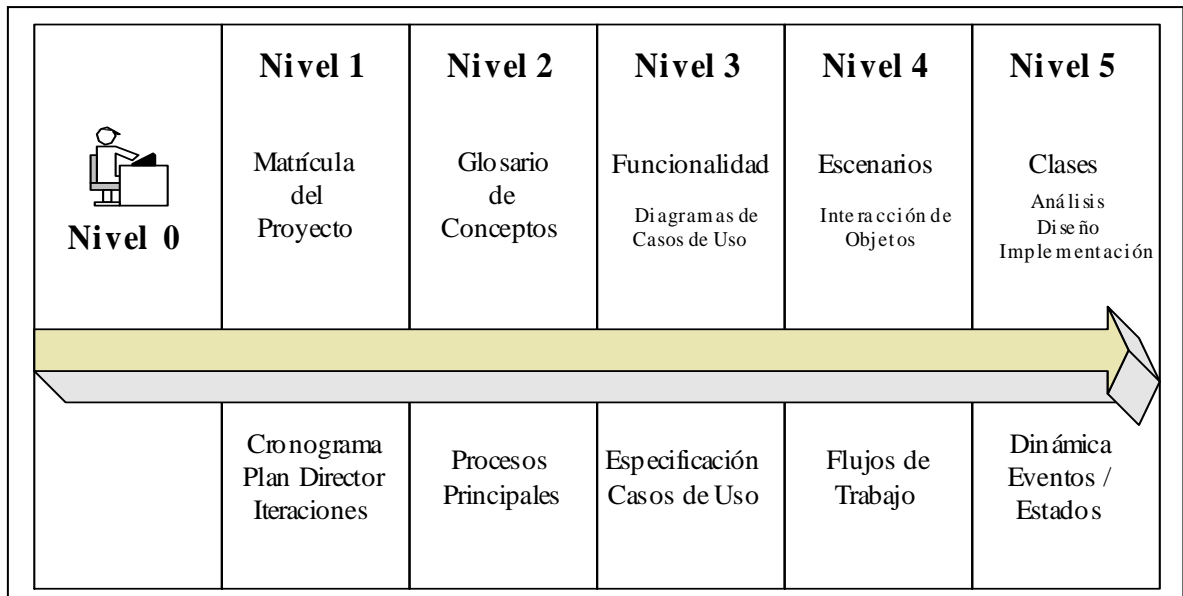


Figura 4.15. Panorama de Concepción y Formalización.

Misión

La Concepción y Formulación utilizando Patrones de Análisis y Funcionales se unen para generar un Anteproyecto, que plantea una Misión a cumplir. El esfuerzo es mayor en la primera etapa, la de Concepción, como muestra la figura 4.16.

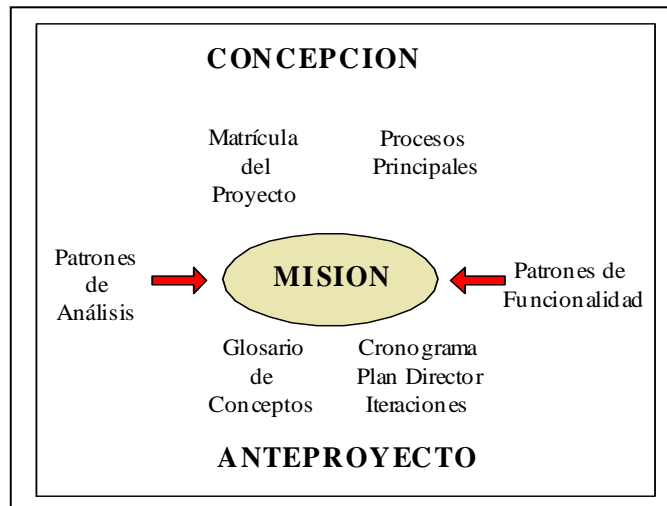


Figura 4.16. Entorno de la Misión.

Modelo

El modelamiento requiere de trabajar más detenidamente en la Formalización: Funcionalidad, Escenarios y Clases, y con detalle la especificación de Casos de Uso, Flujos de trabajo y la Dinámica de Eventos y Estados. El esfuerzo es mayor en la etapa de la Formalización.

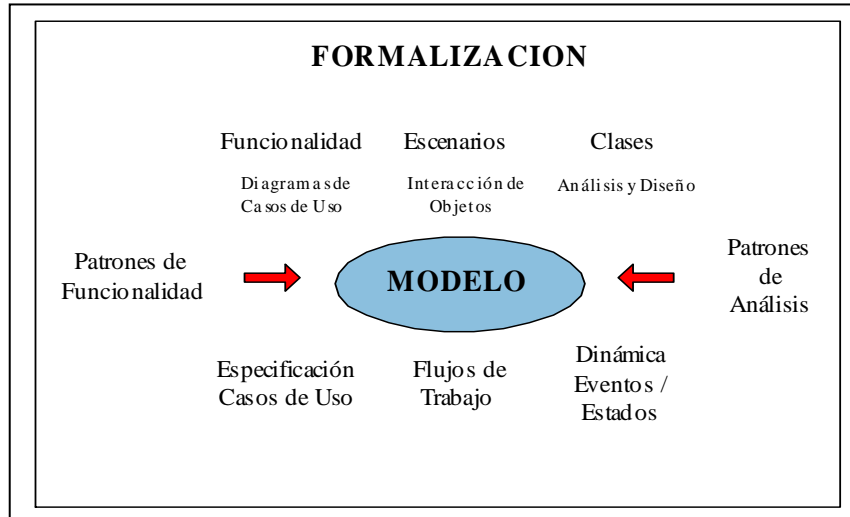


Figura 4.17. Entorno del Modelo.

Componente

Los componentes requieren de las especificaciones de diseño dadas en el modelo, fundamental para proceder a la implementación de las clases y su comportamiento. Esta es la etapa de Construcción o desarrollo por lo que requiere de un minucioso trabajo y mayor esfuerzo, ya que de aquí saldrán las especificaciones del comportamiento de los objetos, para la elaboración de los componentes también se requiere del esfuerzo en la etapa de Transición.

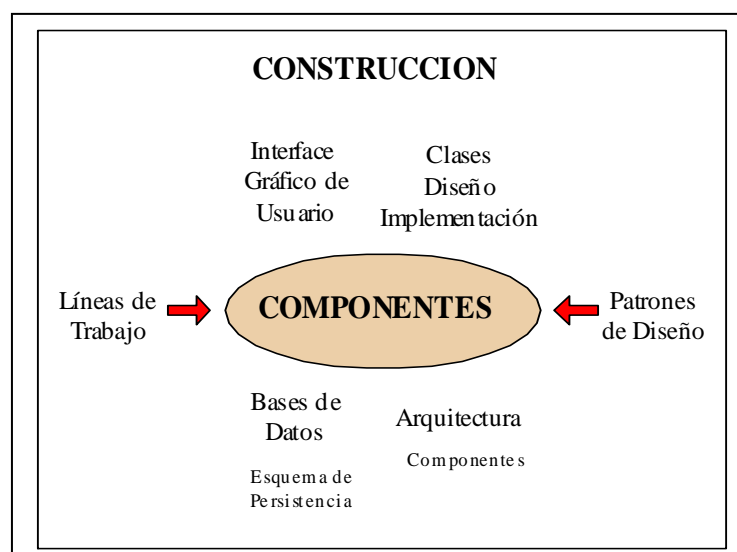


Figura 4.18. Entorno de los Componentes.

Visión Global

Todo el trabajo del análisis orientado a objetos se puede resumir en el gráfico 4.14 donde las acciones a ejecutar o etapas son: Concepción, Formalización, Construcción, Transición, y los resultados que se buscan son: Misión, Modelo, Componentes y Certificación.

El esfuerzo para concluir dicho análisis se muestra en el gráfico 4.19.

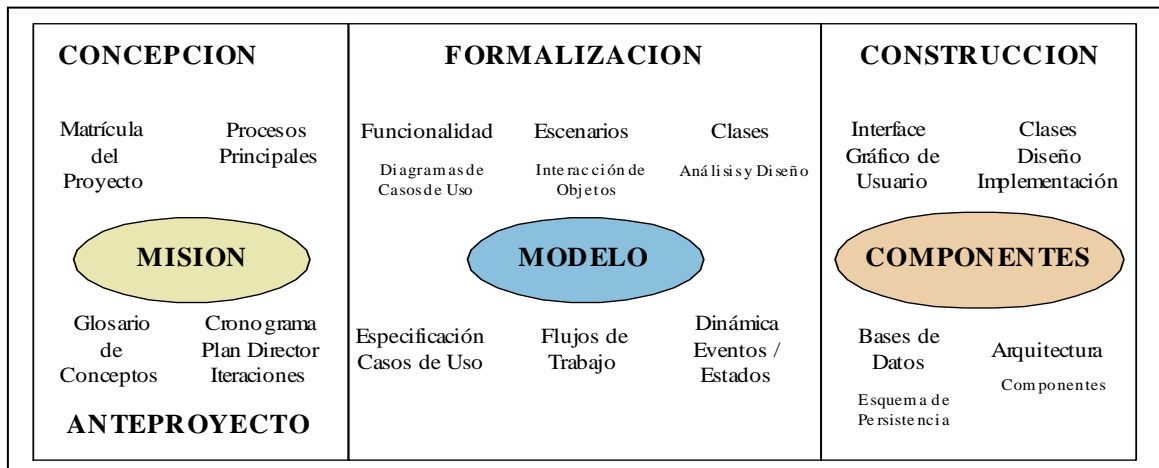


Figura 4.19. Visión Global del Modelamiento.

Lo que ahora debemos es aclarar ciertos aspectos que deben ir en el modelamiento, y los lineamientos a seguir, así tenemos:

B) LOS CASOS DE USO

1. Identificar los Actores.

Qué son los Actores y que hacen:

- Representan a un agente que interactúa con el sistema.
- No son parte del sistema que se desarrolla.
- Proveen de información al sistema.
- Reciben información del sistema.
- Ingresan y reciben información.

Cómo buscar Actores?

1. Quién está interesado en un requerimiento concreto?
2. En qué dominios de la organización se usará el sistema?
3. Quién será beneficiario de la nueva funcionalidad?
4. Quién proveerá, usará y/o retirará, información?
5. Quién dará soporte y administrará el sistema?

6. Usará el sistema un recurso externo?
7. Un usuario actuará con diferentes roles?
8. Diferentes usuarios actuarán con un mismo rol?
9. Interaccionará el nuevo sistema con un sistema antiguo?

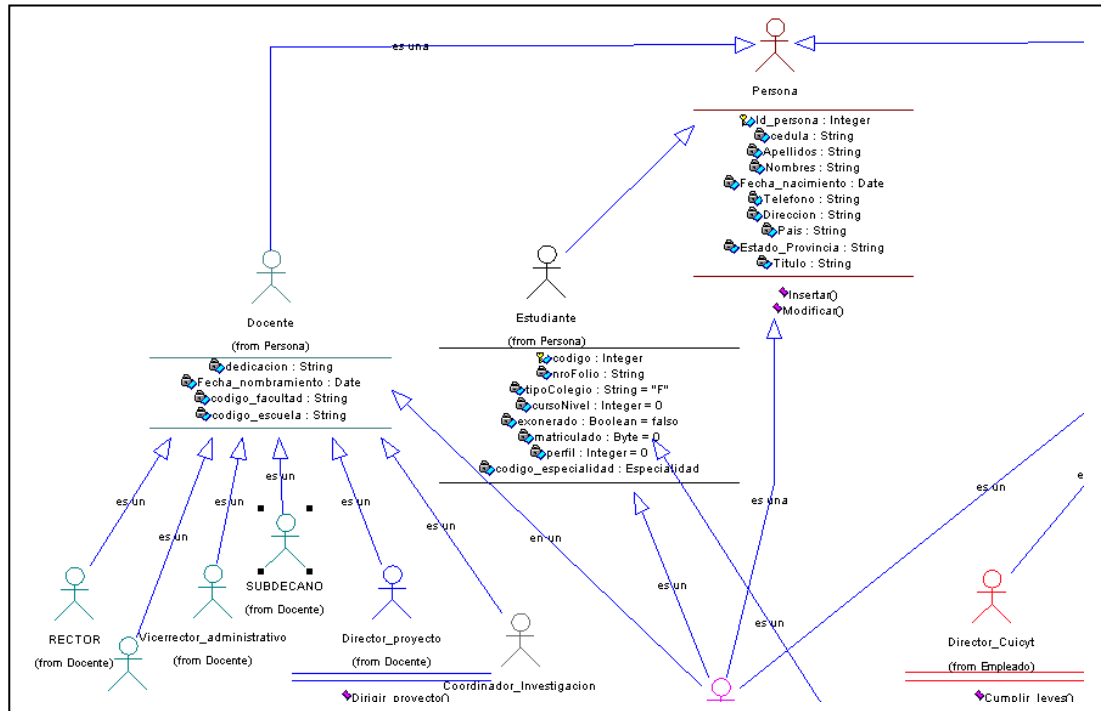


Figura 4.20. Identificación de Actores.

Para el caso del Sistema de Gestión de Proyectos de Investigación, se tiene una clase principal generalizada llamada **PERSONA**, y las clases especializadas **DOCENTE**, **ESTUDIANTE**, **EMPLEADO_ADMINISTRATIVO**, como actores del ámbito Universitario.

Realizando un análisis de Actores que forman parte del entorno de Gestión de Investigación dentro de la Universidad, podemos desglosar algunas Jerarquías de Clases de Actores como las siguientes:

NIVEL 0	NIVEL 1	NIVEL 2
PERSONA	DOCENTE	Rector
		Vicerrector
		Decano
		Subdecano
		Director_Proyecto
		Coordinador_Investigacion
		Investigador

	ESTUDIANTE	Representante_Estudiantil
		Investigador
	EMPLEADO_ADMINISTRATIVO	Director_CUICYT
		Coordinador_Proyecto
		Director_Financiero
		Pagador_Financiero (Tesorero)
		Investigador
		Secretaria

Una vista en árbol de la jerarquía de clases se muestra en la figura 4.21, donde se puede apreciar la Generalización y Especialización de Actores. En este caso el Arbol de Profundidad de Herencia APH = 3.

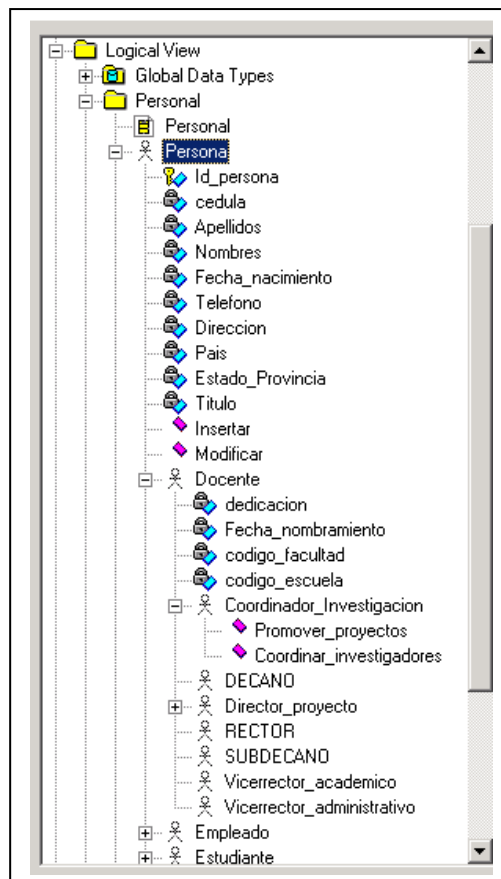


Figura 4.21. Arbol de Jerarquía de Actores.

2. Identificar Casos de Uso

Los Casos de Uso, se deben buscar en función de las siguientes preguntas:

1. Cuales son las tareas y responsabilidades de cada actor?

2. Algún actor creará, almacenará, cambiará, borrará o leerá información del sistema?
3. Qué Casos de Uso crearán, almacenarán, cambiarán, borrarán o leerán esta información?
4. Es necesario que un Actor informe al sistema sobre cambios externos?
5. Es necesario que un Actor sea informado sobre ciertas incidencias del sistema?
6. Qué Casos de Uso darán soporte y mantendrán el sistema?
7. Pueden ser realizados por los Casos de Uso todos los requerimientos funcionales documentados?

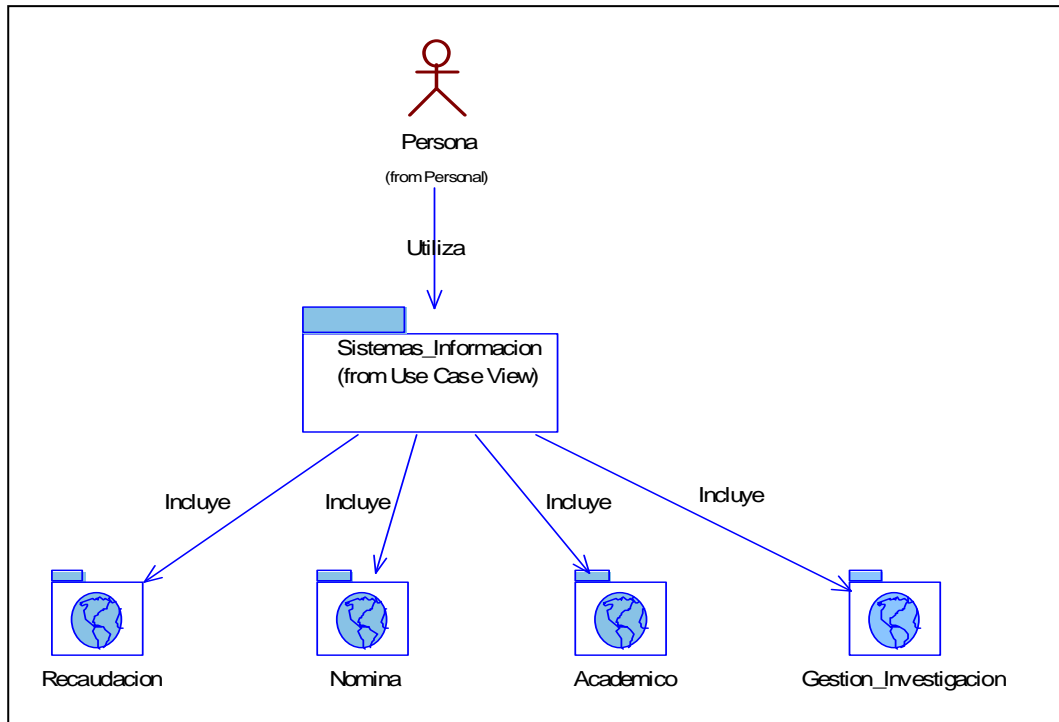


Figura 4.22. Caso de Uso: Sistemas de Información - UTN.

En este caso se plantea en forma macro las principales gestiones en la Universidad Técnica del Norte, que pueden ser representadas y agrupadas en paquetes macro, estos son: Gestión Financiera (Recaudación y Nómina) y Gestión Académica (Académico Estudiantil y Gestión de Investigación).

En un Caso de Uso se debe conocer:

- Límites: Cuando empieza y cómo termina el Caso de Uso.
- Interacciones: Comportamiento de Actores y Sistema. Acción-Reacción dentro del Caso de Uso.
- Masa: Conjunto de Objetos e Interfaces que requiere el Caso de Uso.

- Índice de escenarios: Flujo principal de eventos y secuencia de variaciones posibles dentro de un Caso de Uso.
- Tribulaciones: Contingencias probables que pueden afectar al flujo de los eventos y son excepciones del Caso de Uso.

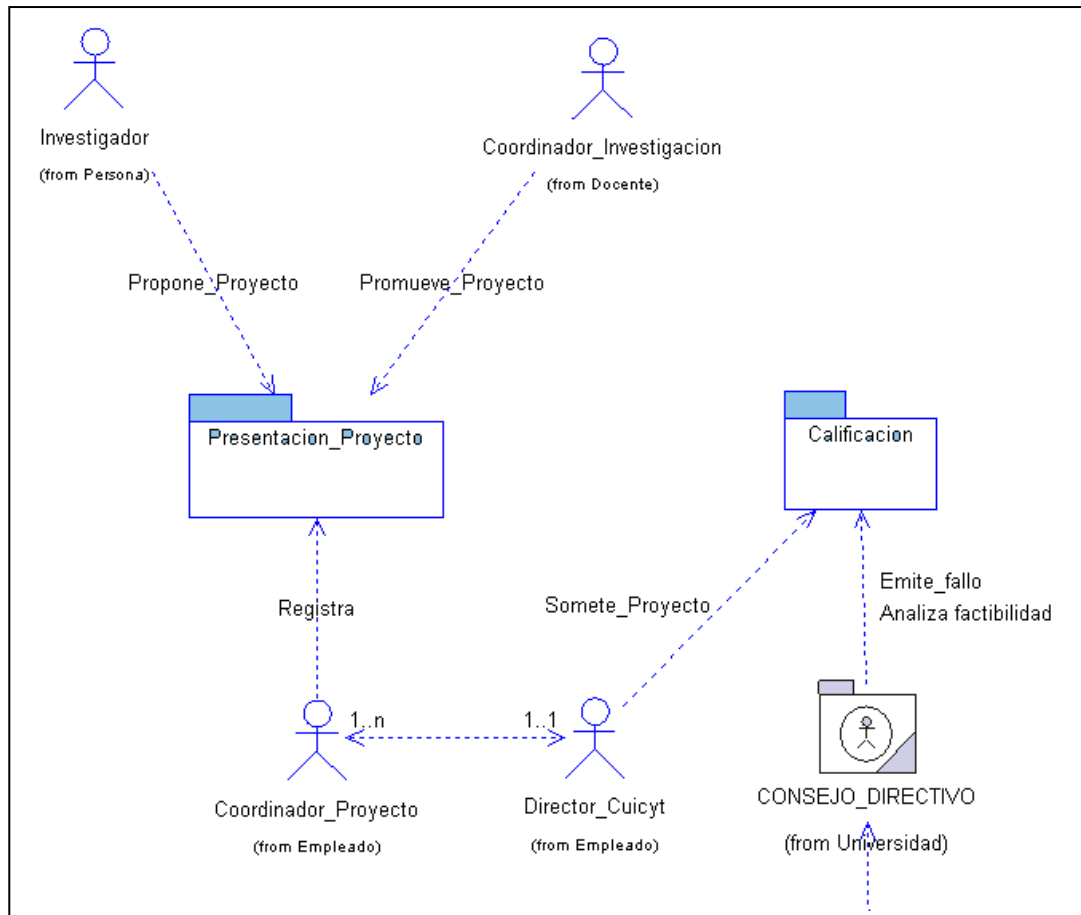


Figura 4.23. Caso de Uso: Gestión Administrativa y Financiera de Proyectos.

En un Caso de Uso se debe especificar:

- Propósito o Regla de Negocio
- Precondiciones
- Activación
- Flujo Principal
- Variaciones
- Excepciones

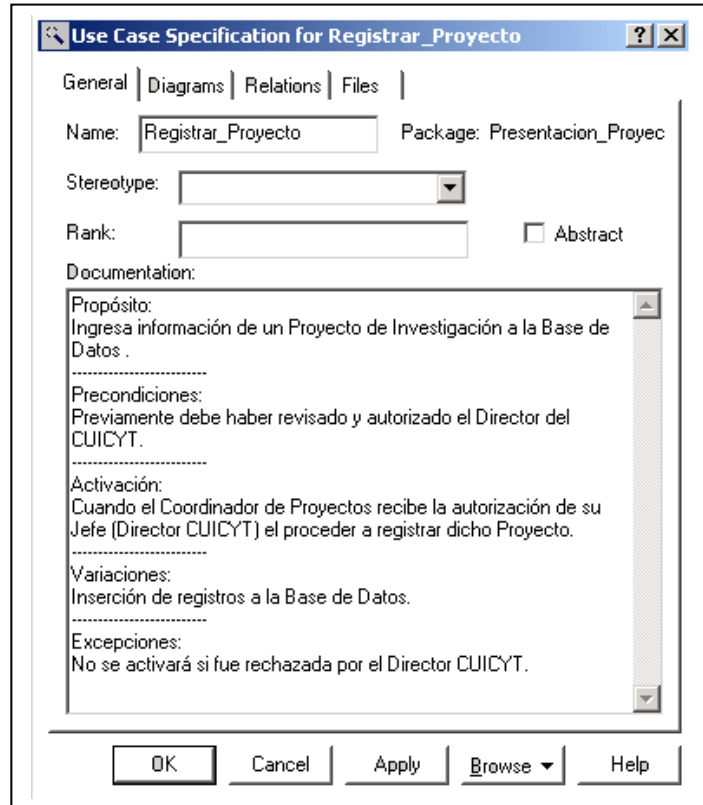


Figura 4.24. Especificación de un Caso de Uso: Registrar Proyecto.

Ventajas del modelo Use Case.

1. Lenguaje de comunicación entre usuarios y desarrolladores.
2. Comprensión detallada de la funcionalidad del sistema.
3. Acotación precisa de las habilitaciones de los usuarios.
4. Gestión de riesgo más eficiente para gobernar la complejidad.
5. Estimación más exacta para determinar tiempo, recursos y prioridades en la dosificación de esfuerzo de desarrollo.
6. Fiel trazabilidad para verificar la traducción de requerimientos en código ejecutable.
7. Mayor control para mantener las sucesivas revisiones de los programas.
8. Certificación contractual Cliente-Desarrollador.
9. Documentación orientada al usuario: Helps - Manual de Procedimientos - Reglas de Negocio.
10. Documentación orientada al administrador del sistema: Soporte de Mantenimiento.

Requerimientos y Casos de Uso

Los Casos de Uso son requerimientos funcionales que describen de una manera detallada el comportamiento del sistema con los distintos Actores que interactúan con él.

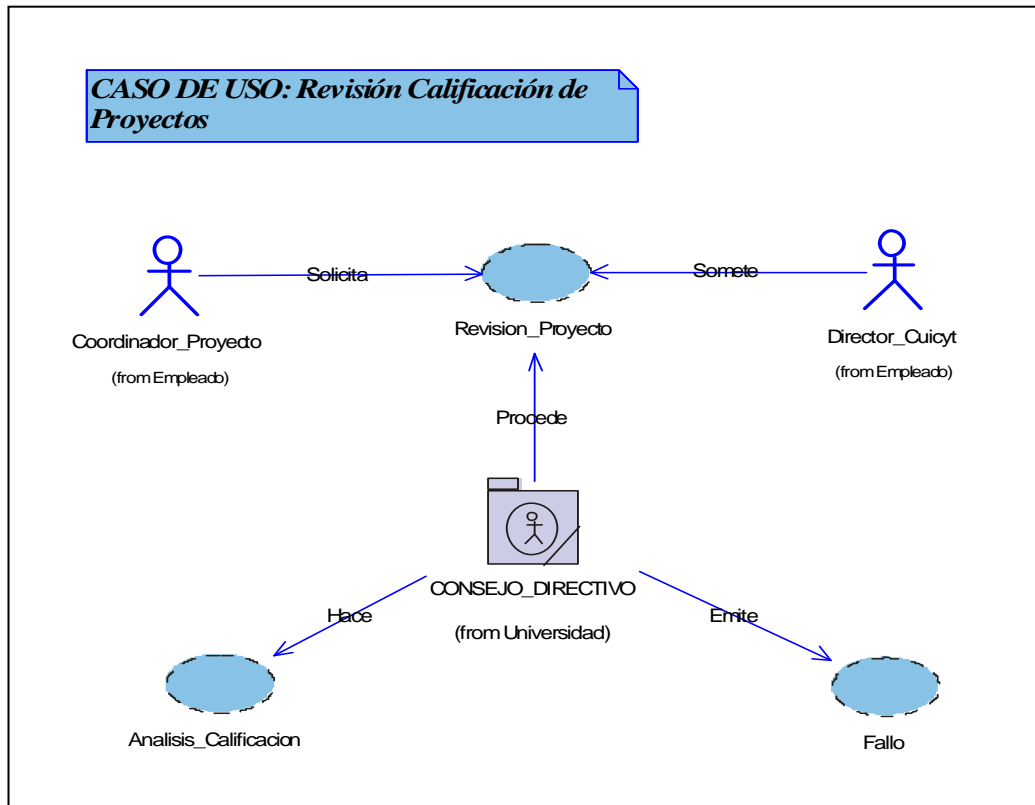


Figura 4.25. Caso de Uso: Revisión de Proyecto.

No definen todos los requerimientos (por ej. los tipos de datos, interfaces externas, niveles de rendimiento esperado, etc.), pero representan el hilo conductor que vincula a todos los requerimientos posibles (actuales y futuros) de una aplicación.

C) DIAGRAMAS DE ACTIVIDAD

1. Descripción de un Flujo de Trabajo

Un flujo de trabajo muestra la secuencia de actividades que se desarrollan dentro de uno o varios Casos de Uso, como una pieza de funcionalidad concreta que satisface los requerimientos de un Actor.

Se deben especificar el flujo principal y las variaciones en paralelo.

En el cuadro siguiente se describe el Flujo de Trabajo para diferentes Actores, con el FLUJO PRINCIPAL y VARIACIONES, en el proceso del Ciclo de Vida de una Autorización de Gastos.

FLUJO PRINCIPAL	VARIACIONES
1. El Coordinador de Proyectos identifica proyecto, y procede a Crear Autorización de Gasto , para lo cual llama al CU Crear_autorizacion .	
2. El Supervisor Financiero procede a Revisar Presupuesto con el uso del sistema, para no generar una autorización desfinanciada, o fuera de tiempo.	a. Si existe presupuesto procede como Favorable , esto quiere decir Aprobado por el CUICYT, Autorizacion.estado=6. b. Si no existe presupuesto, o existen items fuera de tiempo se comprueba como NO Favorable , Autorizacion.estado=5, no hay cambio hasta nueva revisión.
3. El Tesorero identifica las Autorizaciones de Gasto como: Autorizadas por el CUICYT, y procede a Comprobar Presupuesto asignado para Investigación.	a. Si existe Presupuesto para Investigación, Generará Orden de Pago, para que autorice el Director Financiero, con lo cual pasa la Autorización a AUTORIZADO – FINANCIERO, Autorizacion.estado=7 b. Si NO existe Presupuesto , o existe otra novedad, la autorización es RECHAZADA FINANCIERO, Autorizacion.estado=9.
4. El Pagador o Tesorero procede a Generar Pago , para dar a paso a efectivizar la Autorización.	
5. El Pagador o Tesorero procede a Imprimir Cheque , para hacer efectiva la Autorización, con lo cual también imprime la Autorización efectivizada, cheque para el Director del Proyecto.	
6. El Director del Proyecto procede a Utilizar Dinero , en los bienes o servicios que contempla su Proyecto.	
7. El Director del Proyecto procede a Justificar Gastos del Proyecto, para lo cual entrega las facturas y documentos que certifiquen la utilización del dinero de la Autorización de Gasto, esta documentación es entregada al	

Coordinador del Proyecto.	
8. El Coordinador del Proyecto procede a Registrar Justificación de gastos del Proyecto.	

El flujo de trabajo antes descrito se documenta en el diagrama de actividad de la figura 4.26, el Flujo Principal representa la secuencia o etapas por las que deberá pasar una Autorización de Gasto sin ningún contratiempo, es decir con éxito, y las variaciones representan casos especiales o alternativos en los que podría caer un proceso.

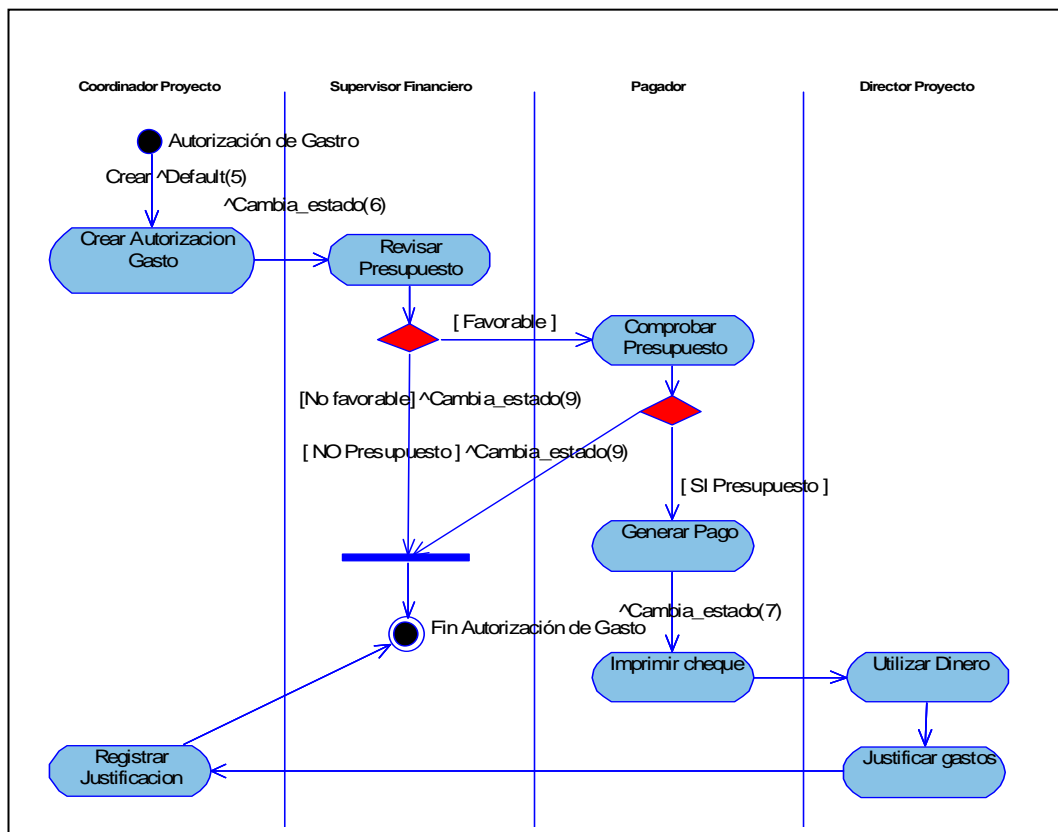


Figura 4.26. Flujo de Trabajo: Ciclo de vida de una Autorización de Gasto.

Cada Item del Flujo Principal equivale a un diagrama de Actividad por Actor, este es un Diagrama de Flujo global, en donde se esquematiza todos los procesos y todos los actores que intervienen en el Ciclo de vida de una Autorización de Gastos.

2. Diagrama de Actividad

Un diagrama de actividad describe una secuencia de actividades que pueden exhibir un comportamiento en paralelo o estar sujetas a condiciones lógicas.

Los procesos de negocio no muestran siempre una secuencia fija en su desarrollo, es una ventaja así poder modelar bloques de actividades sin restricciones de concurrencia. Un diagrama de actividad puede mostrar la secuencia de actividades que se desarrolla en un paquete de Casos de Uso que define un proceso de negocio y sus áreas de responsabilidad.

Su objetivo no es relacionar actividad con objetos, sólo comprender qué actividades son necesarias y cuales son sus relaciones de dependencia.

El diagrama de actividad nos permite definir en qué orden se van a realizar distintas tareas. Los diagramas de flujo (flowchart) sólo nos permiten modelar procesos secuenciales, mientras que los diagramas de actividad nos permiten establecer procesos en paralelo.

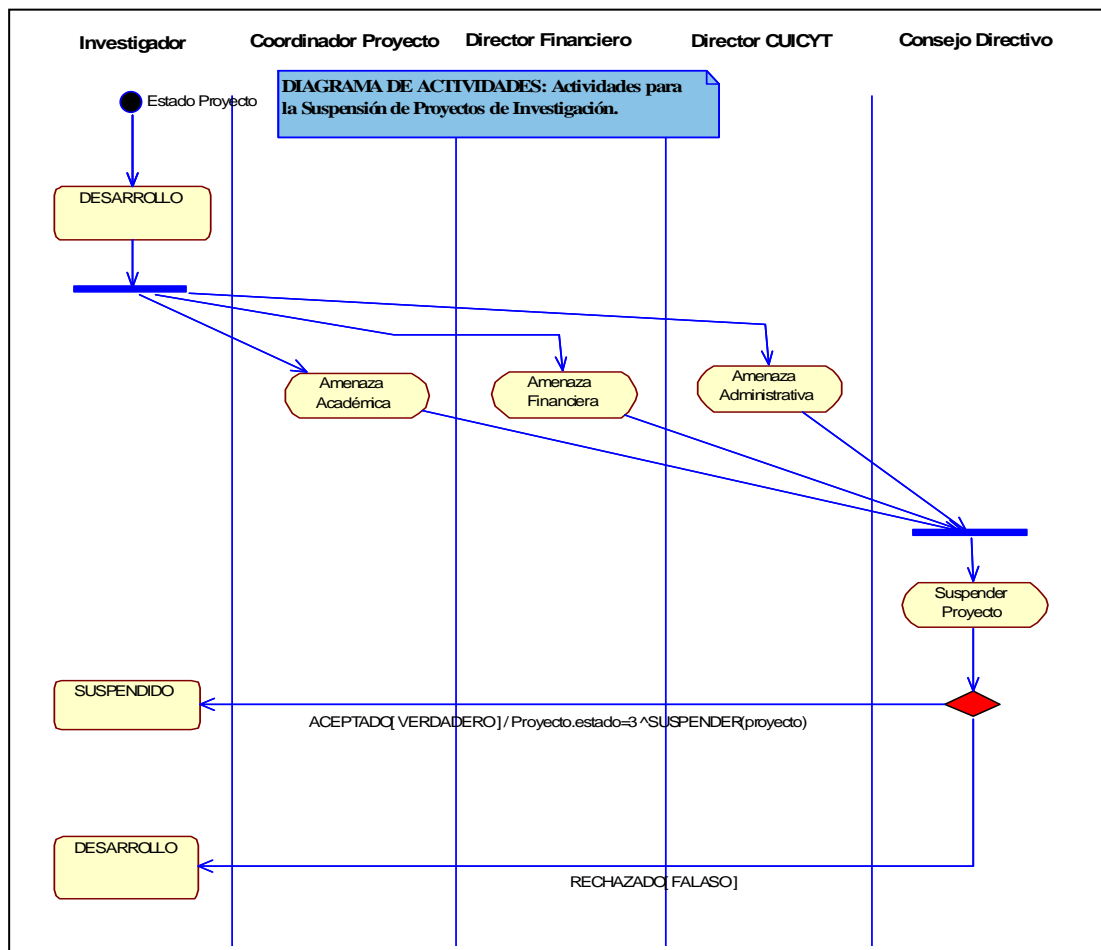


Figura 4.27. Diagrama de Actividad: Suspensión de un Proyecto de Investigación.

Para el caso de crear una Autorización de Gasto, se tiene un conjunto de actividades como las que se muestran en la figura 4.28, donde se especifica más detalladamente las actividades, con su Flujo Principal y Variaciones posibles.

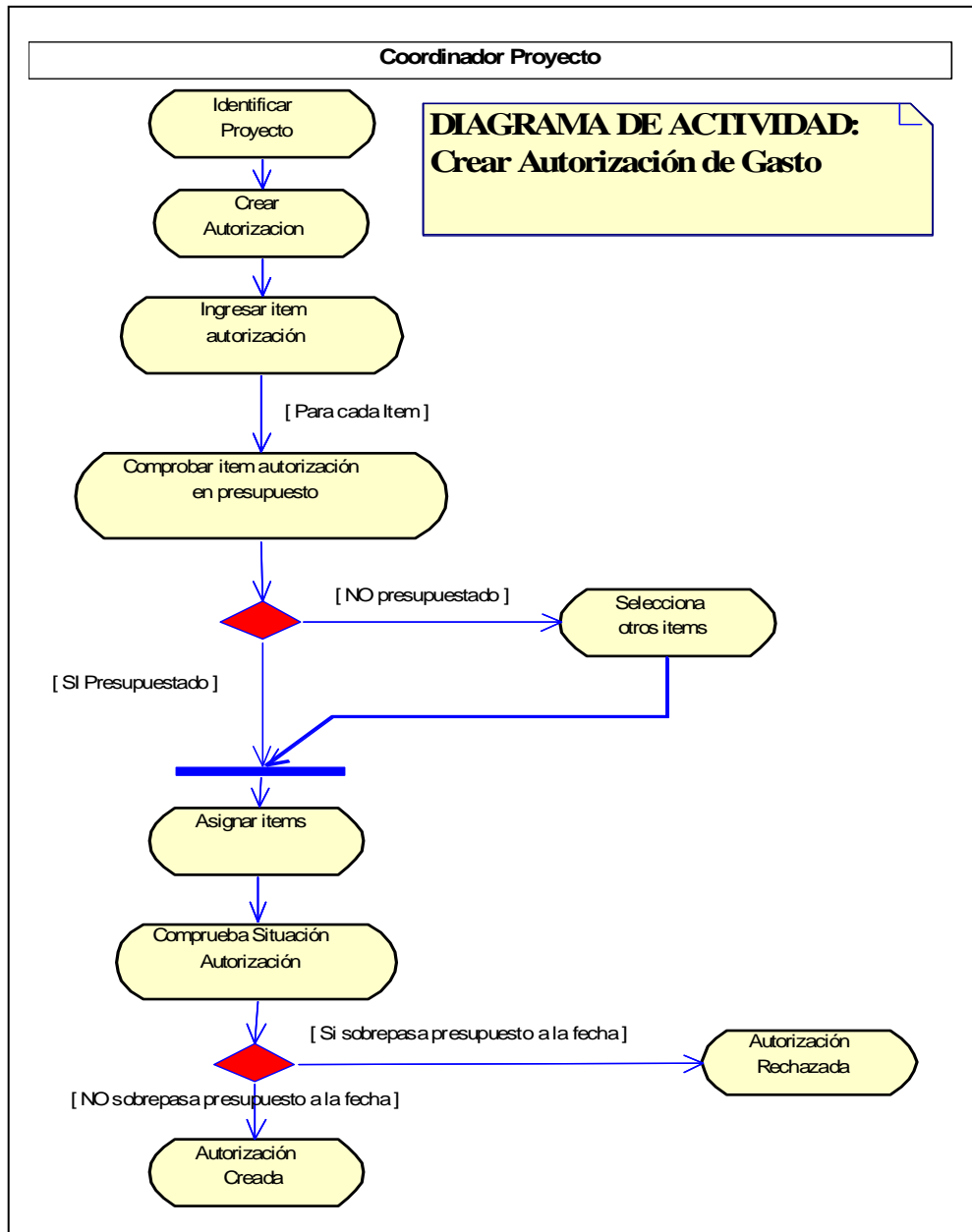


Figura 4.28. Diagrama de Actividad: Crear Autorización de Gasto.

D) DIAGRAMAS DE INTERACCION DE OBJETOS

Descripción de un Escenario

Un escenario muestra de que manera interactúan los distintos objetos dentro del flujo principal de eventos de un Caso de Uso con alguna variación o extensión concreta del mismo.

Utilizamos dos diagramas de interacción:

- a) Colaboración
- b) Secuencia

Su finalidad es describir los mensajes que intercambian los distintos objetos para cumplir con las responsabilidades definidas en un escenario concreto de un Caso de Uso.

1. Diagrama de Colaboración.

Representa una posible interacción de los objetos ordenados a partir de la topología que muestra el envío de sus mensajes.

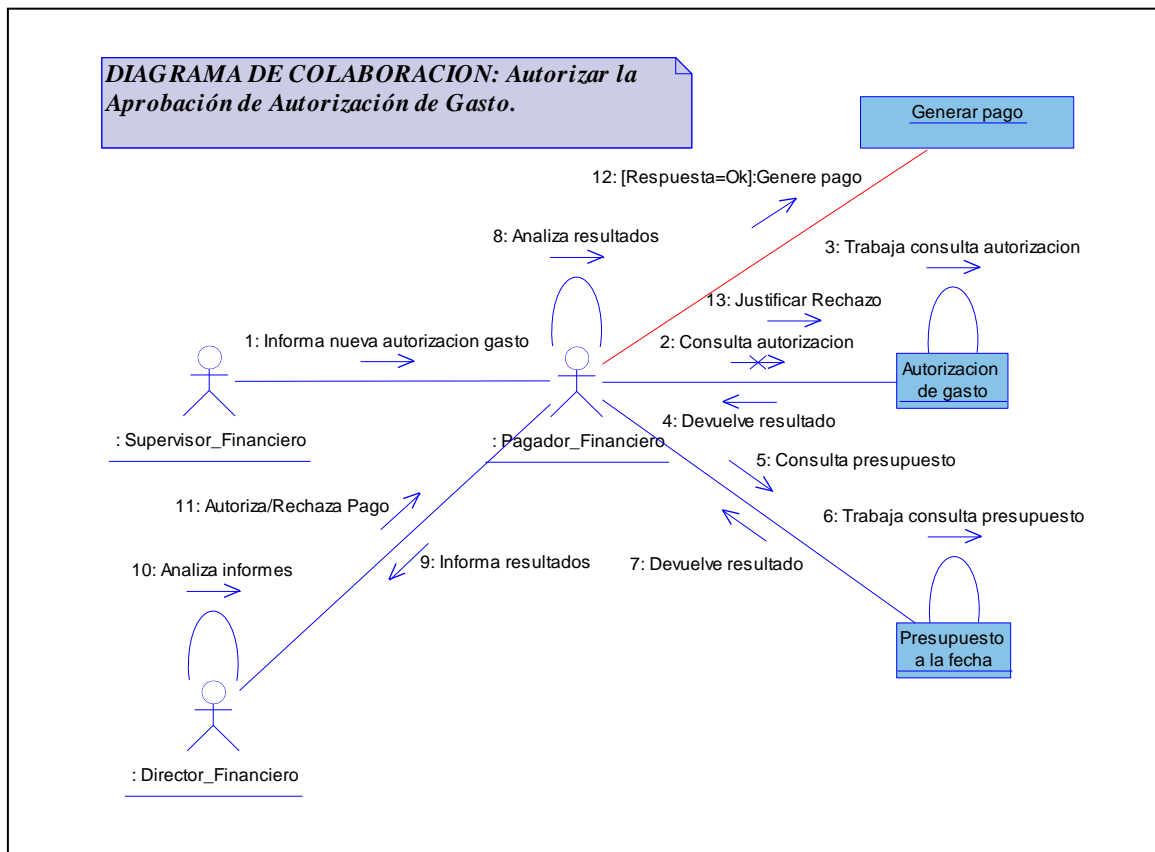


Figura 4.29. Diagrama de Colaboración: Autorizar Aprobación de Autorización de Gasto.

A partir del Diagrama de Colaboración se puede generar el Diagrama de Secuencia, sin mayor trabajo, abierto el Diagrama de Colaboración seleccionar en el menú principal la opción BROWSE, y luego seleccione la última opción Go To Sequence Diagram o presione la tecla F5, con lo queda creado el Diagrama de Secuencia.

2. Diagrama de Secuencia

Representa las interacciones de objetos ordenadas en una serie temporal que muestra su ciclo de vida. Mensajes enviados entre los objetos descritos en el flujo de eventos de un Caso de Uso. Estos mensajes muestran el nivel de colaboración entre los distintos objetos existentes e indican cuando se requiere generar nuevos objetos para cumplir con las responsabilidades asignadas.

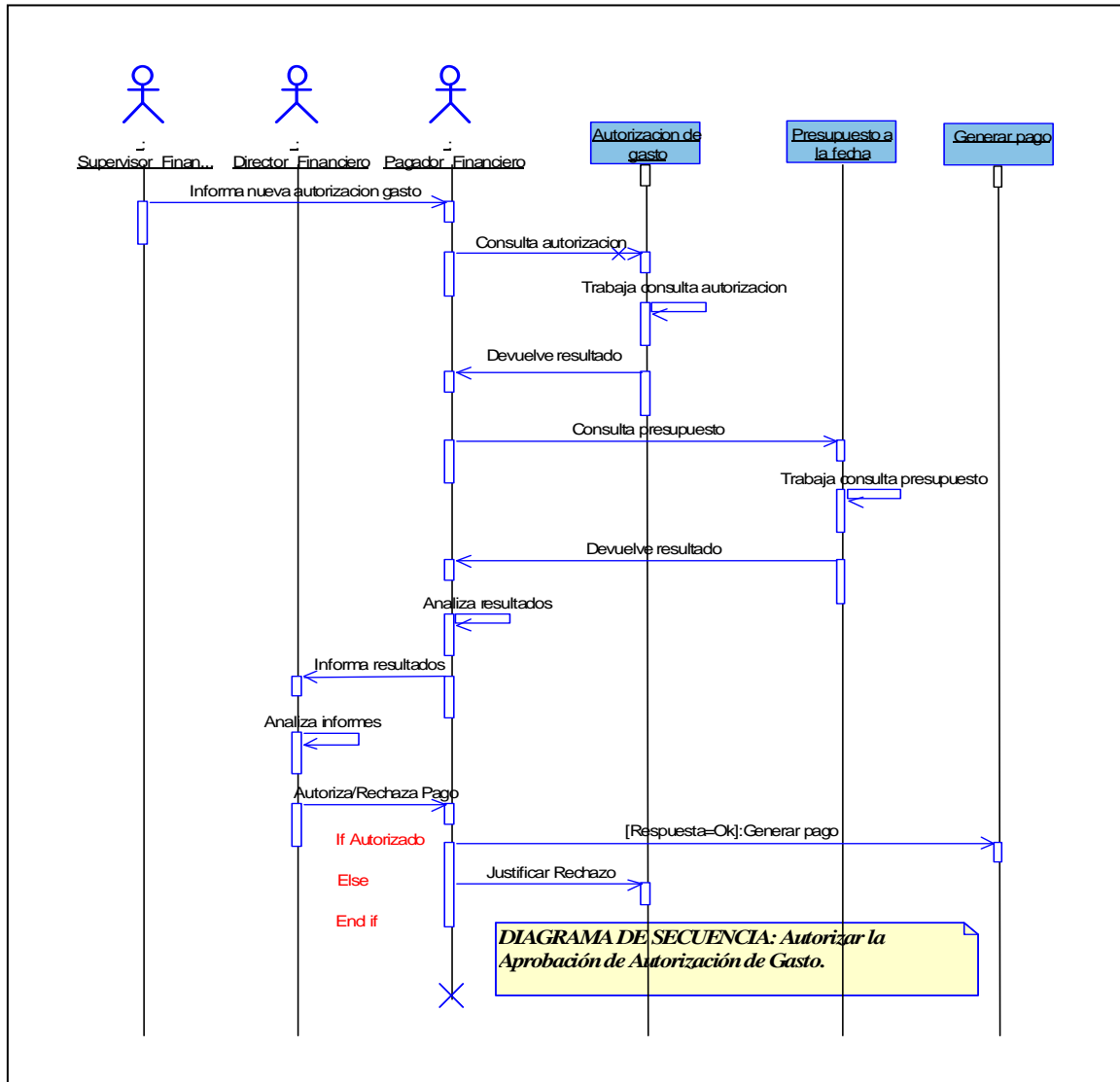


Figura 4.30. Diagrama de Secuencia: Autorizar Aprobación de Autorización de Gasto.

En un diagrama de secuencia se puede especificar algunas instrucciones de condición, como en el caso de la figura 4.30, con el uso del If-Else, como opciones que parten luego de emitir un criterio el Actor Director Financiero al Tesorero o Pagador, quien tiene dos opciones: Generar Pago o Justificar rechazo, con lo que este diagrama de secuencia finaliza.

Se puede apreciar un control centralizado por parte del Actor **Pagador Financiero**, con las acciones **Justificar Rechazo** y **Generar Pago**.

E) CLASE Y OBJETOS

Definiciones de:

Clases

Desde una perspectiva conceptual, una Clase representa un conjunto de Objetos que comparten:

- Las mismas propiedades (Atributos)
- El mismo comportamiento (Métodos)
- Las mismas relaciones con otros Objetos (Mensajes)
- La misma semántica dentro del sistema

Un Objeto representa una entidad del mundo real o inventado.

Es un concepto, una abstracción o algo que dispone de unos límites bien definidos y tiene una significación para el sistema que se pretende modelar.

Estructura y función:

- Identidad ¿Quién soy? = Atributos
- Propósito ¿Cual es mi misión? = Justificación
- Responsabilidades ¿Qué debo hacer? = Métodos
- Procedencia ¿De qué Clase provengo? = Origen
- Relaciones ¿Qué mensajes entiendo? = Comportamiento

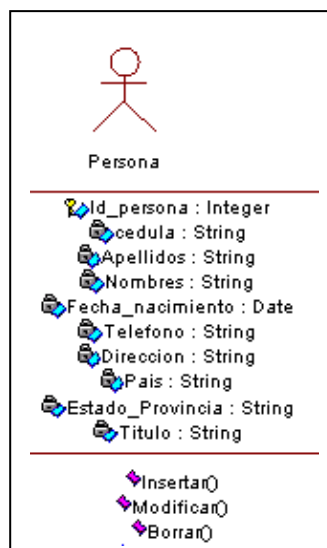


Figura 4.31. Clase Persona.

La clase PERSONA, representa una realidad definida a personas que se encuentran en el entorno Universitario o tengan alguna relación con la Universidad, esta es una Superclase a partir de la cual se generan Actores con funciones, propósitos y responsabilidades específicas.

Objetos

Desde una perspectiva física, una Clase es una pieza de software que actúa como un molde para fabricar tipos particulares de objetos que disponen de los mismos atributos y métodos.

Estos elementos que configuran cada tipo de objeto sólo se definen una vez, cuando especificamos la estructura de la Clase a la que pertenecen.

Los objetos que se han creado a partir de una Clase concreta, se llaman instancias de esta Clase y se diferencian entre ellos únicamente por los valores de sus atributos (variables).

Abstracción

A partir de una abstracción del mundo real, definimos objetos que representan micromódulos de software ideales.

Desde su creación, se mantienen de manera independiente unos de otros, sólo interactúan con otros objetos a través de sus mensajes. Cada objeto configura un universo ordenado y autosuficiente.

Mensaje

Una aplicación orientada a objetos consiste en un número determinado de objetos que interactúan entre sí enviándose mensajes unos a otros para invocar sus métodos. Este intercambio de mensajes facilita su comportamiento, cambios de estado, destrucción o almacenamiento.

Ya que todo lo que un objeto puede realizar está expresado en sus métodos, este simple mecanismo de mensajes soporta todas las posibles interacciones entre ellos.

Encapsulación

El empaquetado de métodos y atributos dentro de un objeto mediante una interfaz de mensajes, es lo que denominamos encapsulación. La clave está precisamente en este envoltorio del objeto. La interfaz que rodea por completo al objeto actúa como punto de contacto para todos los mensajes que llegan desde cualquier objeto emisor.

La interfaz de mensajes tiene la misma función que la membrana de una célula, disponer de una barrera esencial entre la estructura interna de un objeto y el exterior.

Su propósito es garantizar que todas las interacciones del objeto tengan lugar a través de un sistema de mensajería predefinido con el que es capaz de entenderse y reaccionar adecuadamente. No existe ninguna otra manera con la que un objeto externo pueda acceder a los atributos y métodos escondidos dentro de la interfaz de mensajes de otro objeto.

Herencia

Es el mecanismo por el cual una clase de objetos puede ser definida como un caso especial de otra clase más genérica. Los casos especiales de una clase se denominan Subclases.

La clase más genérica, se conoce como la Superclase de todos sus casos especiales. Además de los métodos y atributos que heredan, las Subclases definen sus propios métodos y atributos. También, pueden redefinir algunos de los heredados (overriding - sobremontando).

La interfaz de mensajes definida para una Superclase también es heredada por las subclases. Esto implica que todas las Subclases de una Superclase están preparadas para responder correctamente a los mismos mensajes que la Clase padre. Esta propiedad es extremadamente útil porque nos permite tratar de la misma manera a todas las especializaciones de una Clase.

Composición

Los objetos que contienen a otros objetos se denominan Objetos Compuestos. Los atributos de un objeto pueden utilizarse de dos maneras. Podemos usarlos para almacenar valores como el número 15, o bien, el texto "Autorizado - CUICYT".

También pueden contener referencias de otros objetos. Las referencias almacenadas en los atributos de un objeto compuesto, permiten a este objeto enviar los mensajes apropiados a todos los objetos contenidos.

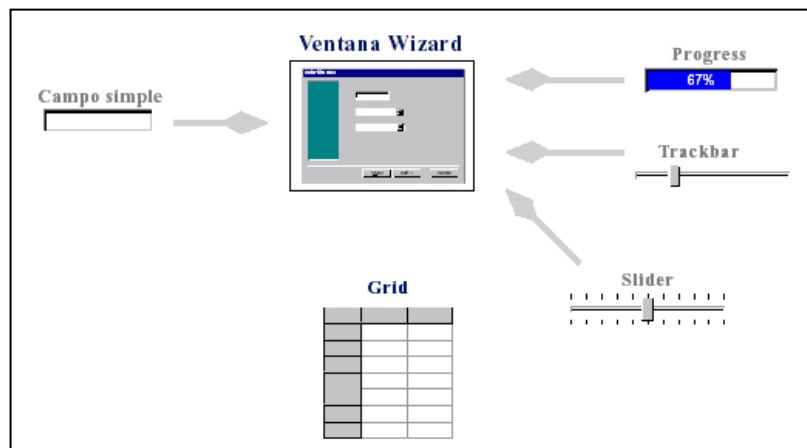


Figura 4.32. Esfuerzo Global del Modelamiento.

Polimorfismo

Diferentes objetos pueden responder a un mismo mensaje de diferentes maneras. El polimorfismo permite a los objetos interactuar entre ellos sin necesidad de conocer previamente a que tipo pertenece.

Un objeto puede tener diferentes comportamientos.

El llamado a un método: **objeto.listar_datos_empleado(pos)**, con parámetro, puede especificar que se visualice al empleado con identificación =10, y.

El llamado a un método: **objeto.listar_datos_empleado()**, sin parámetro, especificará que se liste a todos los empleados.

No hay necesidad así de picar un código específico para un reporte, con lo cual, nos ahorramos prolijas sentencias IF o CASE que son complejas de mantener y de actualizar cuando se incorporan nuevos tipos de objetos.

Visibilidad

Toda Clase encapsula unos elementos (atributos y operaciones) que disponen de ciertos criterios de visibilidad y manipulación para otras Clases.

Los elementos públicos pueden ser usados por cualquier otra Clase. Los elementos privados pueden ser usados sólo por la Clase propietaria. Cada plataforma de desarrollo (C++, Smalltalk, Java) desarrolla sus propias reglas con respecto a la visibilidad y manipulación de atributos y operaciones.

La notación UML especifica que todo atributo y operación de una Clase ha de disponer de un indicador de visibilidad.

F) DIAGRAMA DE TRANSICION DE ESTADOS

Descripción de la Dinámica del Sistema. La dinámica de un sistema está determinada por:

- Todos los posibles estados que sus objetos pueden tener.
- Todas las posibles secuencias de eventos que pueden ocurrir.
- Todas las transiciones posibles, de un estado a otro, como consecuencia de los eventos que afectan a los objetos.

Utilizamos el **diagrama de estados** para describir el comportamiento de una Clase dentro de una serie temporal. Construimos el diagrama de estados a partir de una clase concreta para mostrar el comportamiento de un objeto durante su ciclo de vida.

Un Evento no es un objeto. Un Evento es la "causa" que justifica la existencia de una información. Sólo podemos conocer que un Evento ha ocurrido, detectando sus "efectos" en nuestro modelo. Sólo nos **interesan** aquellos **Eventos que provocan un cambio de estado en los objetos de nuestro modelo**. Hay que distinguir un Evento como tal, del objeto que representa el registro de los efectos de dicho Evento.

Los diagramas de estados son muy útiles para describir el comportamiento de un objeto a través de múltiples Casos de Uso.

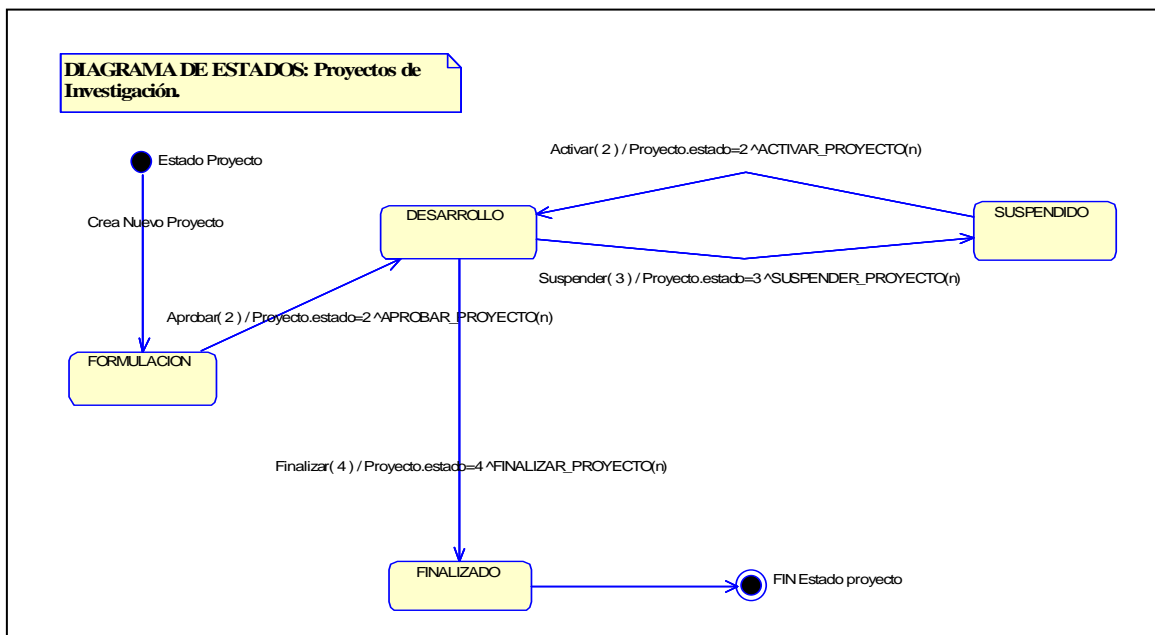


Figura 4.33. Diagrama de Estados: Estados posibles de un Proyecto de Investigación.

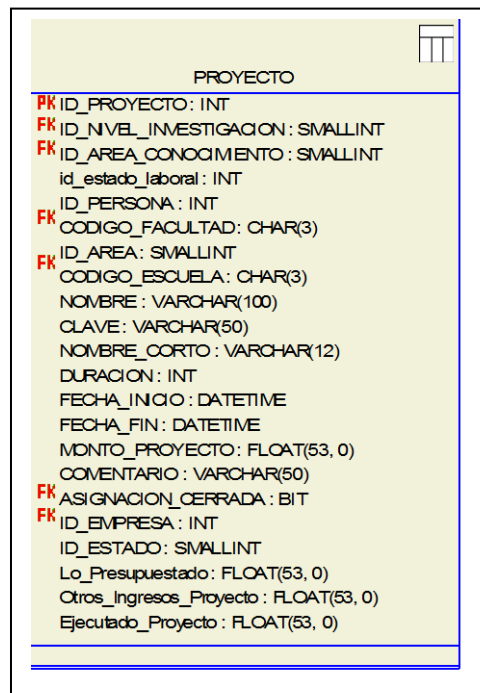


Figura 4.34. Clase PROYECTO, definida para registrar Proyectos de Investigación.

G) ARQUITECTURA DEL SISTEMA

Una Arquitectura es un conjunto organizado de elementos. Se utiliza para especificar las decisiones estratégicas acerca de la estructura y funcionalidad del sistema, las colaboraciones entre sus distintos elementos y su despliegue físico para cumplir unas responsabilidades bien definidas.

1. Modelo de Referencia

La vista del Modelo de Referencia (Reference Information Model), está determinada por la arquitectura lógica. La arquitectura lógica es capturada por los diagramas de Clases que contienen las Clases y relaciones que representan las abstracciones esenciales del sistema a desarrollar.

- Clases
- Asociaciones
- Agregaciones
- Generalización
- Packages

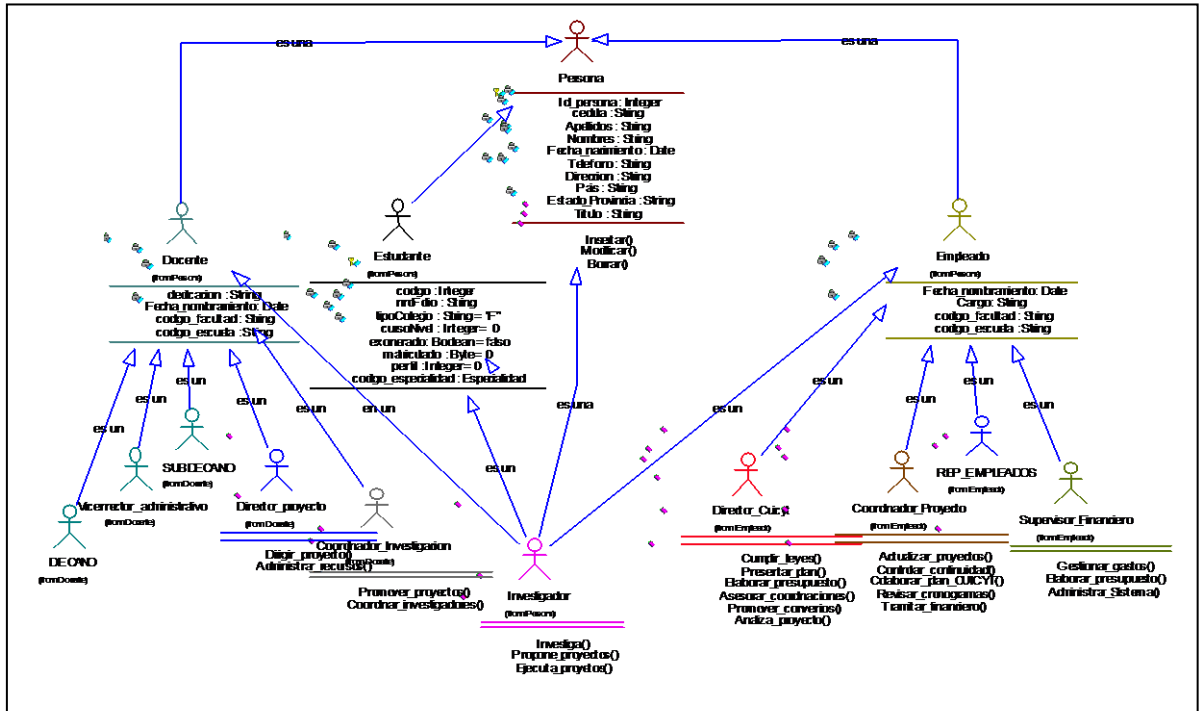


Figura 4.35. Clases y Generalizaciones del paquete PERSONAL.

Un extracto del paquete de personal se muestra en el gráfico anterior, y su implementación de clases se la realiza en los esquemas de personal, mostrada en el archivo adjunto a esta tesis, en el anexo del modelo UML del aplicativo.

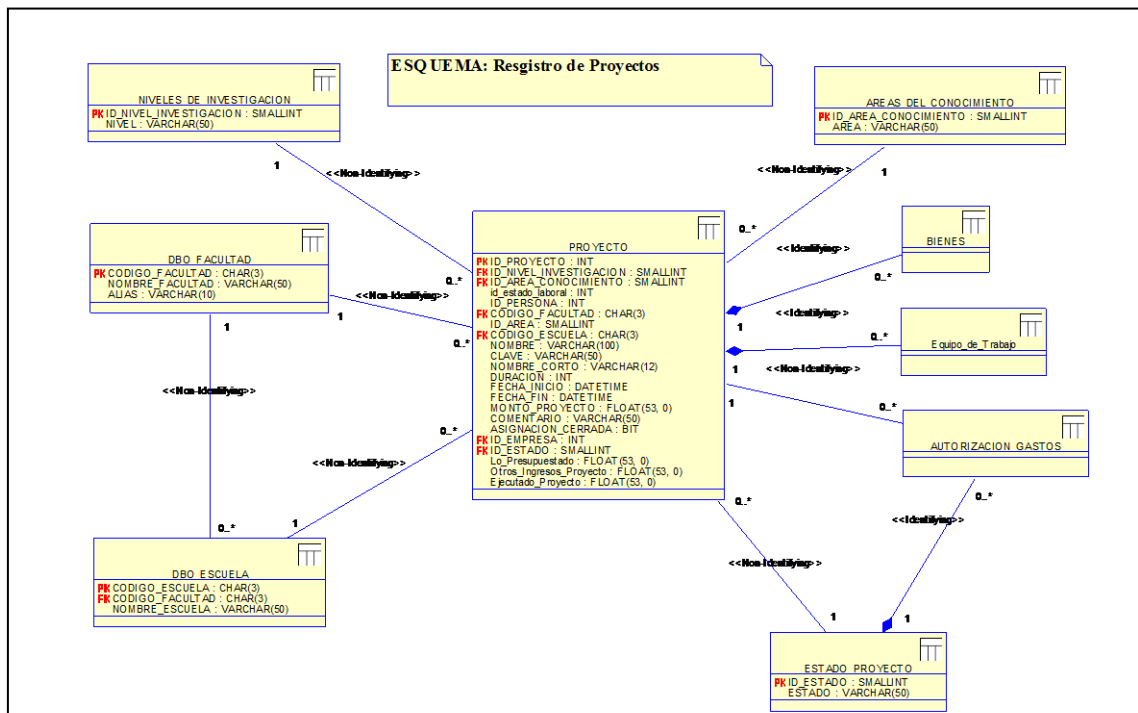


Figura 4.36. Diagrama de Clases: Esquema para el registro de Proyectos.

Un método alternativo con el que cuentan la gran mayoría de herramientas de Modelamiento de Software, es la Ingeniería Inversa que se puede aplicar a un modelo de Base de Datos Relacional existente, lo cual significa un ahorro de trabajo en esta etapa y dedicar más tiempo a la documentación en el modelamiento del software.

2. Vista de Componentes Modulares

La vista de Componentes Modulares refleja la organización de módulos de software dentro del entorno de desarrollo. Esta vista de arquitectura toma en consideración los requerimientos que facilitan la programación, los niveles de reutilización, y las limitaciones impuestas por el entorno de desarrollo.

Disponemos de dos elementos para modelar esta vista:

- Packages: Esta vista representa una **partición física del sistema**.
- Componentes: Esta vista representa la organización de **módulos de código fuente**.

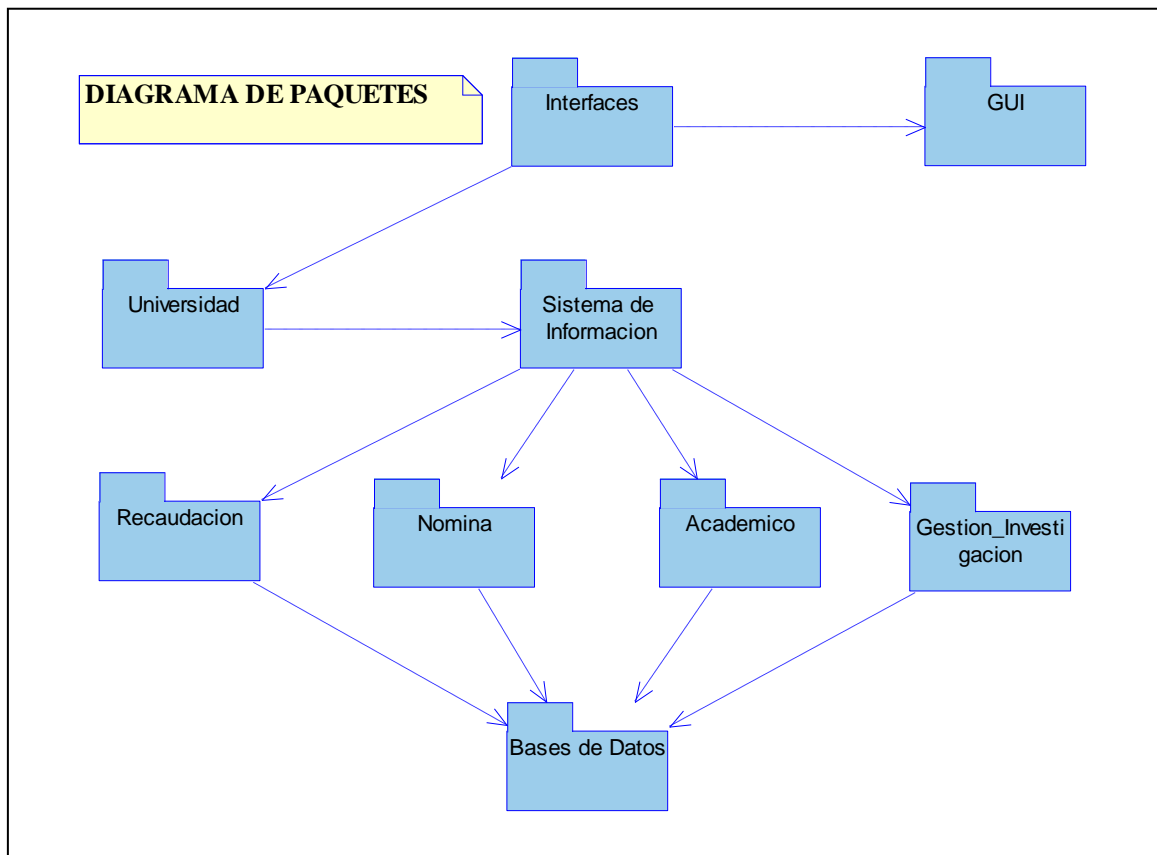


Figura 4.37. Ejemplo de Diagrama de Paquetes.

Este es un diagrama de paquetes macro a nivel de Universidad donde se especifican las aplicaciones macro y sus relaciones.

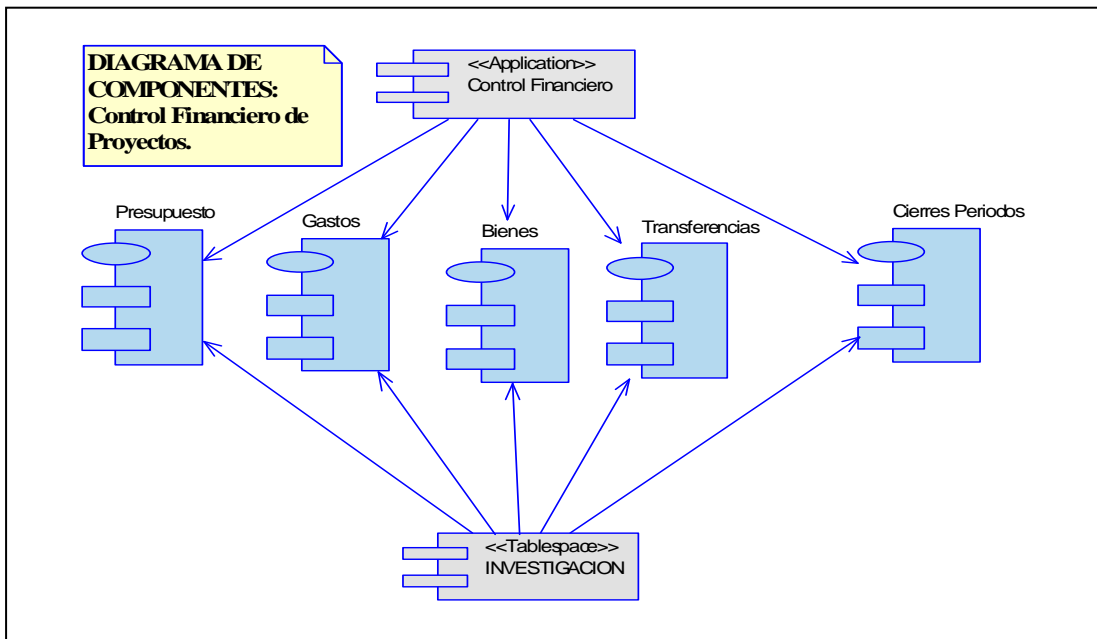


Figura 4.38. Ejemplo de Diagrama de Componentes: Control Financiero de Proyectos.

En este diagrama de componentes consta la Aplicación **Control Financiero** con Presupuesto, Control de Gastos, Registro de Bienes, Transferencias Internas y Cierres de Periodo como Componentes de Especificación y un Tablespace como un espacio en la gran Base de Datos – UTN.

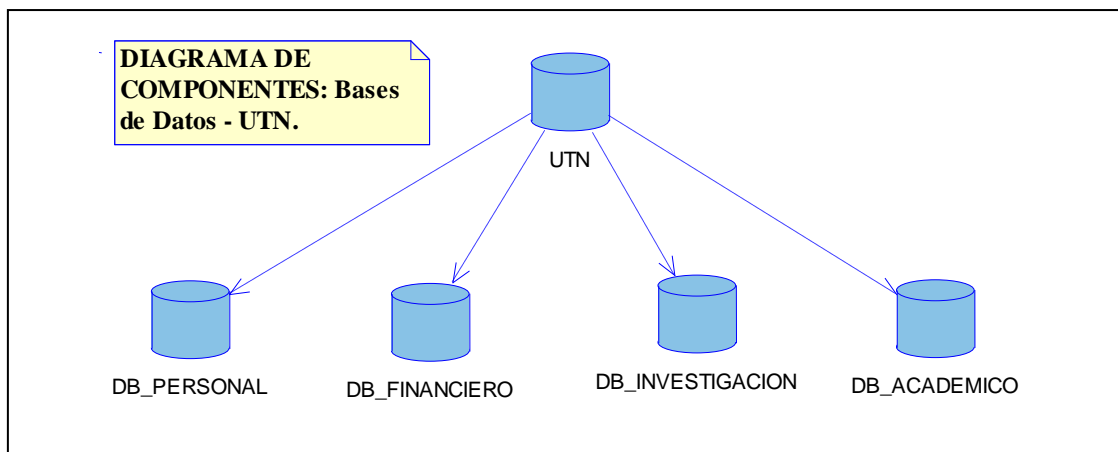


Figura 4.39. Ejemplo de Diagrama de Componentes: Bases de Datos - UTN.

Con otro diagrama de componentes se podría especificar los motores de Bases de Datos a utilizar, para las diferentes aplicaciones; la Universidad cuenta con un motor de Base de Datos al cual se

deben estandarizar las aplicaciones que es INFORMIX, actualmente tiene el inconveniente del soporte técnico en esta herramienta.

3. Arquitectura del Sistema o Vista de Distribución Física de Elementos

Esta vista presenta el mapping de componentes de software ejecutables con los nodos de procesamiento (hardware).

Esta arquitectura tiene en cuenta los siguientes requerimientos:

- Disponibilidad del sistema
- Rendimiento
- Escalabilidad

Los diagramas de distribución muestran el despliegue de nodos (locales y remotos), en la organización de la empresa.

Permite al equipo de desarrollo comprender mejor la topología de un sistema distribuido.

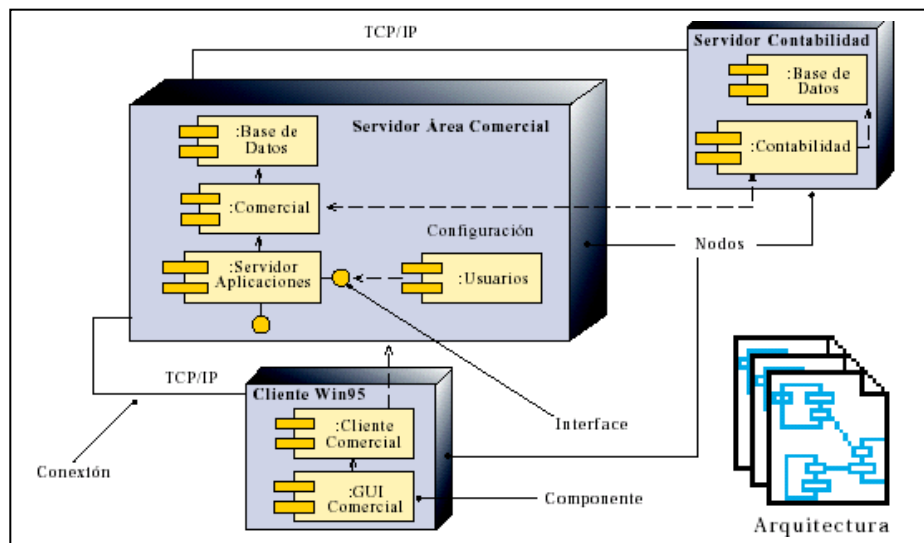


Figura 4.40. Arquitectura del Sistema.

Esta vista certifica la validez de:

- Modelo de Referencia
- Componentes modulares de software
- Ejecutables
- Distribución de recursos informáticos

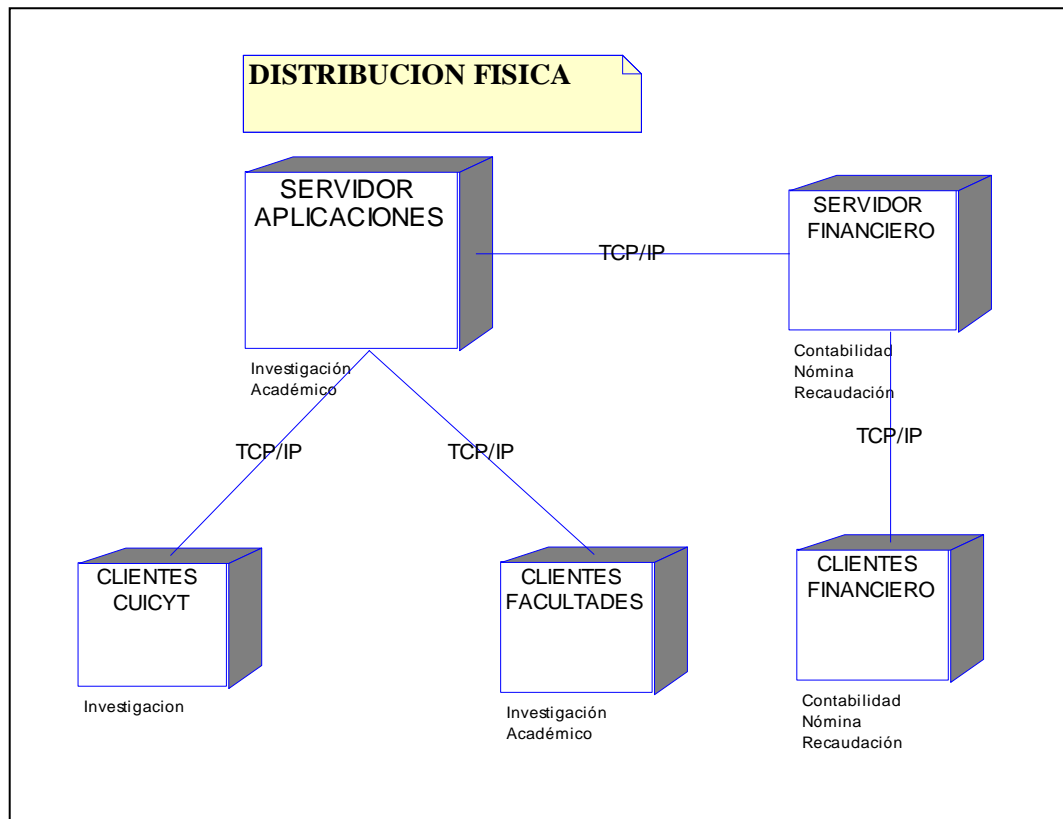


Figura 4.41. Arquitectura del Sistema: Distribución Física de Aplicaciones y Clientes.

RESUMEN - UML

Con la funcionalidad requerida del sistema. Utilizamos los siguientes elementos para describir la funcionalidad:

- Diagramas de Casos de Uso
- Especificación de Casos de Uso
- Diagramas de Interacción (Escenarios)
- Diagramas de Actividad (Flujos de Trabajo)
- Diagramas de Estados (Dinámica)

CAPITULO 5

Sistemas Colaborativos



5.1. INTRODUCCION

Los sistemas colaborativos se presentan como una herramienta esencial para comprometer el trabajo de las personas de una organización, la Universidad Técnica del Norte, ha venido dando pasos innovadores en la parte informática a través de aplicaciones que han ayudado a mejorar las actividades operativas especialmente financieras y académicas. Por lo que es importante innovar la metodología de trabajo de ciertas áreas prioritarias; la investigación y sus procesos de gestión son el pilar fundamental en el área académica por lo que se pretende hacer una demostración de lo que se puede hacer para mejorar los procesos administrativos y operativos de la gestión de la investigación, es aquí en la parte administrativa académica, en donde entran los sistemas colaborativos, para ayudar a un grupo de personas que realicen un trabajo común e integrar las actividades de tal manera que el trabajo se vuelva más eficiente y menos burocrático.

5.2. DEFINICION

Una aplicación colaborativa se puede definir como un sistema computacional que asiste a un grupo de personas dedicadas a una tarea común, cada una de ellas trabajando en su propio computador, pero compartiendo datos y programas a través de una interfaz multiusuario.

5.3. ELEMENTOS Y AREAS DE TRABAJO

La construcción de una aplicación colaborativa es una tarea compleja ya que involucra diversas áreas de trabajo tales como sistemas distribuidos, comunicaciones, interfaces humano-computador, inteligencia artificial, bases de datos y otras áreas como la sociología. La Sociología por ejemplo estudia diversos aspectos del comportamiento humano, principalmente en lo referido a interacciones grupales donde se debe tomar en cuenta distintas características de los integrantes del grupo como la cohesión, las formas de comunicación y retroalimentación, el grado de satisfacción con el trabajo realizado, formas de lograr acuerdos, y otros factores que influyen en la eficacia del trabajo en grupo.

El principal objetivo de las aplicaciones colaborativas consiste en proveer a los usuarios de herramientas que les permitan coordinar sus actividades de trabajo, para lo cual se requiere analizar los elementos que interviene en los sistemas colaborativos:

Comunicación, Colaboración y Coordinación en los Sistemas Colaborativos

1. La Comunicación.

Es el elemento fundamental para el trabajo en grupo, se podría decir que es como su **"sistema nervioso"**. Sin ella no existe el grupo, sino un conjunto de personas aisladas.

2. La Colaboración.

La colaboración funciona como el **"cerebro"** de la computación grupal ya que permite que el grupo utilice los recursos de manera inteligente.

3. La Coordinación.

"La efectividad de la comunicación y la colaboración puede ser incrementada si las actividades de grupo son coordinadas" [YankeeGroup95]. La coordinación se podría ver como **los músculos de las actividades grupales**, ya que sincroniza los esfuerzos y asegura que los procesos se desarrollen de la manera más eficiente posible. Sin coordinación muchas actividades pueden hacerse repetidas veces o peor aún pueden dejar de hacerse.

Así como hay diversidad de definiciones de los sistemas colaborativos, también existen muchos dominios de aplicaciones colaborativas. Por ejemplo hay aplicaciones que facilitan la interacción a distancia entre miembros de un equipo de trabajo. Otras aplicaciones apoyan la comunicación de personas reunidas físicamente. Otro tipo de aplicaciones ayudan a distintas personas en la confección de un mismo documento electrónico. A pesar de los distintos tipos de aplicaciones que hay, todas tienen un factor común que consiste en hacer más fácil el proceso de compartir información entre un grupo de personas que realizan una tarea común.

5.4. GROUPWARE

El groupware de cliente/servidor es un conjunto de tecnologías que nos permiten representar procesos complejos centrados en actividades humanas de colaboración. Se apoya en cinco tecnologías fundamentales: administración de documentos con multimedia, flujo de trabajo, correo electrónico, conferencias y planificación.

El concepto de "Trabajo Cooperativo Asistido por Computador" (en inglés, Compute Supported Cooperative Work), en adelante **CSCW**, pareciera tener su origen en el año 1984 Grudin94, año en el cual en respuesta a una iniciativa de la Corporación de Equipos Digitales (en inglés, Digital Equipment Corporation) y el Instituto Tecnológico de Massachusetts (en inglés, Massachusetts Institute of Technology, MIT). En ese entonces se reunió un grupo de desarrolladores e investigadores de distintas áreas para explorar el rol de la tecnología en el ambiente de trabajo y acuñaron el término "Trabajo Cooperativo Asistido por Computador", para describir dicho rol.

5.4.1. Que es el Groupware?

Groupware es el software que soporta la creación, flujo y seguimiento de información no estructurada en apoyo directo a actividades grupales en colaboración. El groupware interviene en la administración tanto de información como de actividades.



Figura 5.1. Tareas de un grupo de trabajo.

El término Groupware es una contracción de las palabras Group y Software. Este término fue utilizado inicialmente para referirse a un sistema basado en computador más los procesos sociales de grupos - Ellis91, sin embargo, fue restringido luego a sistemas basados en computador.

Los sistemas considerados como groupware, se enmarcan dentro del área de investigación de CSCW. En función de lo anterior, se desea resaltar el cambio al que apuntan lo cooperativo apoyado por computador. CSCW puede ser vista como una disciplina que guía el correcto análisis, diseño y desarrollo de los sistemas colaborativos.

El propósito de groupware se puede resumir en asistir la colaboración, comunicación y coordinación de las actividades de las personas que trabajan en grupo. De aquí nace una de las definiciones más usadas de groupware, que lo define como: "Sistema basado en computador que apoya a un grupo de personas dedicadas a una tarea o meta común y que provee los servicios para apoyar la labor de los usuarios a través de una interfaz de un ambiente compartido" Ellis91.

Las nociones de tarea común y ambiente compartido son cruciales, ya que determinan la exclusión de sistemas que no tienen tareas ni objetivos comunes, por ejemplo excluye a los sistemas multiusuarios de tiempo compartido, Ellis91.

5.4.2 Groupware y la reingeniería

El groupware puede transformar a una compañía al modificar el estilo de comunicación entre las personas y, en consecuencia, el cambio en los procesos de negocios. El groupware puede automatizar el servicio a clientes y volver más sensible a una compañía, así como también el potencial para aplanar a las organizaciones y eliminar capas de burocracia.

La burocracia en la mayoría de las organizaciones es muy resistente; hará falta mucho más que una buena dosis de groupware para eliminarla. De hecho, el groupware puede ser mal utilizado y dar lugar a la automatización de la burocracia, con lo que su permanencia quedaría garantizada.

En el movimiento de la reingeniería se afirma que no sirve de nada lanzar tecnología a un proceso con desempeño deficiente. Estamos de acuerdo; nadie pretende oponerse a la evidencia de que no tiene sentido automatizar un proceso que, para comenzar, ni siquiera debería existir. Sin embargo, el groupware puede automatizar todo tipo de procesos, incluyendo los malos. Los procesos ineficientes pueden volverse eficientemente ineficientes.

LA DIFERENCIA

Las bases de datos relacionales fueron dedicadas a datos sumamente estructurados a los que se accede por medio de SQL. Son excelentes para la administración de aplicaciones que requieren controles de alta concurrencia, como candados y características de aislamiento, necesarios para actualizaciones inmediatas. El groupware ofrece las herramientas para prender estos datos en el punto de entrada y organizarlos en algo llamado documento. Un documento puede concebirse como un recipiente de diversos tipos de información. El documento es a un grupo de trabajo lo que una tabla a una base de datos de SQL: la unidad básica de manejo. El groupware permite que los usuarios finales creen bases de datos de documentos. Puede mover esos documentos a través del correo electrónico y réplicas de base de datos. Y proporciona todo lo necesario para consultar, administrar y navegar por bases de datos de documentos. Los documentos son la moneda corriente del groupware.

Los BLOB

SQL no especifica un estándar de autodescripción de datos para BLOB. La determinación de qué se encuentra en el BLOB tiene que ser reinventada por cada aplicación. Las herramientas de GUI son mudas en cuanto al contenido de cada BLOB y la forma en que debería procesarseles. En lo que se

refiere a BLOB y multimedia, una base de datos de SQL es solo un glorificado y costoso servidor de archivos sin valor agregado. En contraste, los servidores de documentos de groupware han hecho grandes avances hacia la provisión de cierto tipo de solución de cliente/servidor con multimedia. Es de señalar que algunos proveedores de bases de datos de SQL ofrecen actualmente extensiones de objetos propietarias que permiten la manipulación de tipos de datos semejantes a BLOB, tales como imágenes, impresiones digitales, archivos y mensajes de correo electrónico. A medida que las bases de datos de SQL se orienten más a los objetos, brindarán bases de datos de documentos que valgan lo que cuestan.

5.4.3. Los componentes del groupware

Como ya se mencionó, el groupware se apoya en cinco tecnologías básicas: administración de documentos con multimedia, flujo de trabajo, correo electrónico, conferencias y planificación.

Modelos de flujo de trabajo

Las tres R del flujo de trabajo son: rutas, reglas y roles.

- **Las rutas** definen las vías de desplazamiento de los objetos. Incluyen también definiciones de los objetos - documentos, formatos, eventos, contenedores y partes electrónicos, mensajes, etc. por enrutar.
- **Las reglas** definen qué información se enruta y a quién. Definen tanto las condiciones que debe cumplir el flujo de trabajo para avanzar al siguiente paso como la clase de manejo de las excepciones: "Si la carga es mayor de \$10,000 dólares, enviarla al supervisor o a alguien que la envíe al siguiente salto."
- **Los roles** definen funciones de trabajo independientemente de las personas que las llevan a cabo. Por ejemplo, el rol del "supervisor-financiero" puede ser ejercido por los usuarios "Arturo" y "Lenin". Cualquiera de estas personas puede hacer el trabajo; basta con colocarlo en la siguiente cola disponible del supervisor.

5.5. El modelo Action Workflow

Action Workflow de Action Technologies, innovador producto de flujo de trabajo ad hoc, se basa en las investigaciones realizadas por Terry Winograd y Fernando Flores sobre el modo de comunicación entre personas para realizar una acción.

De acuerdo con la metodología de Action Technologies, cada acción en un flujo de trabajo se compone de cuatro fases, en las que clientes y ejecutantes se coordinan entre sí:

1. La **preparación** ocurre cuando el cliente se prepara a solicitar algo; a menudo, por ejemplo, mediante el llenado de una forma.
2. La **negociación** tiene lugar cuando el cliente y el ejecutante se ponen de acuerdo sobre el trabajo por realizar y sobre las condiciones de satisfacción; es decir, determinan exactamente qué se debe hacer para concluir el trabajo en cuestión a satisfacción del cliente.
3. La **ejecución** es la fase de realización efectiva del trabajo. Una vez concluida, el ejecutante le informa al cliente que la misión ha sido cumplida.
4. La **aceptación** de un trabajo o tarea no se considera consumada hasta que el cliente la otorga y expresa su satisfacción al ejecutante.

El método de Action Technologies va más allá de la sola coordinación de las tareas: permite especificar el contrato de cliente/servidor y lo hace todo muy recursivamente. La aplicación en sí es una tarea de cliente/servidor, que se divide recursivamente en subtareas según lo definido por el flujo de trabajo. Se trata de una serie de pasos del tipo "quién se encarga de lo que se hace para quién" y de condiciones de satisfacción.

5.5.1. La Workflow Coalition

La *Workflow Management Coalition (WfMC)* fue fundada en agosto de 1993 en calidad de organismo internacional no lucrativo para el desarrollo y promoción de estándares de flujo de trabajo. Su membresía está abierta a todas las partes interesadas o implicadas en la creación, análisis o desarrollo de sistemas de administración de flujo de trabajo. En 1996, WfMC tenía más de 150 miembros, entre ellos la totalidad de los principales proveedores de flujo de trabajo y los mayores integradores de sistemas.

El objetivo a largo plazo de WfMC es definir un conjunto de API que pueda ser usado por aplicaciones y herramientas del cliente para invocar y definir funciones de flujo de trabajo y controlar un mecanismo de flujo de trabajo. WfMC trabaja también en un protocolo de interoperabilidad que permite la comunicación entre sí de mecanismo de flujo de trabajo de proveedores múltiples. En diciembre de 1995, WfMC dio a conocer su primer conjunto de API, llamado *API del cliente de flujo de trabajo (Workflow Client API)*. Esta API es utilizada por aplicaciones cliente de flujo de trabajo para invocar los servicios de mecanismos de flujo de trabajo de proveedores múltiples.

La API del cliente de flujo de trabajo se compone de 56 llamadas de funciones divididas en cuatro grandes categorías:

- **Las funciones de conexión** comprenden 21 llamadas de API que le permiten conectarse con un mecanismo de flujo de trabajo y desconectarse luego de realizado el trabajo.
- **El control y estado de procesos** consta de 23 llamadas de API que permiten consultar la definición de un proceso, leer y actualizar su estado corriente, crear una instancia de proceso, arrancar el proceso y obtener el estado del proceso.
- **El control y estado de actividades** se compone de 13 llamadas de API que permiten consultar una actividad, leer y actualizar sus atributos y estado, y obtener su estado corriente.
- **El control de administración y lista de trabajo** consiste en 18 llamadas de API que permiten consultar una lista de trabajo, manipular las unidades de trabajo que contiene y administrar la ejecución de procesos y actividades.

5.6. MODELOS CONVERSACIONALES

5.6.1 Modelo de Argumentación de IBIS

Una discusión constructiva consiste en la presentación de ideas en una forma organizada con algún argumento de soporte que pueda ser entendido y revisado por otros miembros. En una reunión de personas cara a cara los miembros usan el lenguaje natural para expresar sus posiciones. Aunque rico por naturaleza un texto o un discurso es ambiguo cuando no se analiza apropiadamente. La organización del discurso o texto dependerá mucho de la capacidad de los individuos para poner sus ideas en forma simple, concisa y objetiva. Aun cuando esta meta es alcanzada, debido a muchas posibilidades de organización de los discursos, podría ser difícil para otros individuos separar los muchos elementos del discurso con el fin de entenderlo mejor. Si el discurso es significativo para ser almacenado por un sistema, la clasificación o interpretación automática es aun más difícil.

En 1970 Kunz y Rittel, proponen un modelo llamado IBIS (Issue Based Information System), **Sistema de Información Basado en el Problema**, traducido al español, compuesto de 3 elementos y 9 relaciones. El modelo va dirigido a representar los elementos cruciales de una discusión permitiendo a las personas entender y facilitar contribuciones adicionales. El esquema conceptual de IBIS es mostrado en la Figura 5.2.

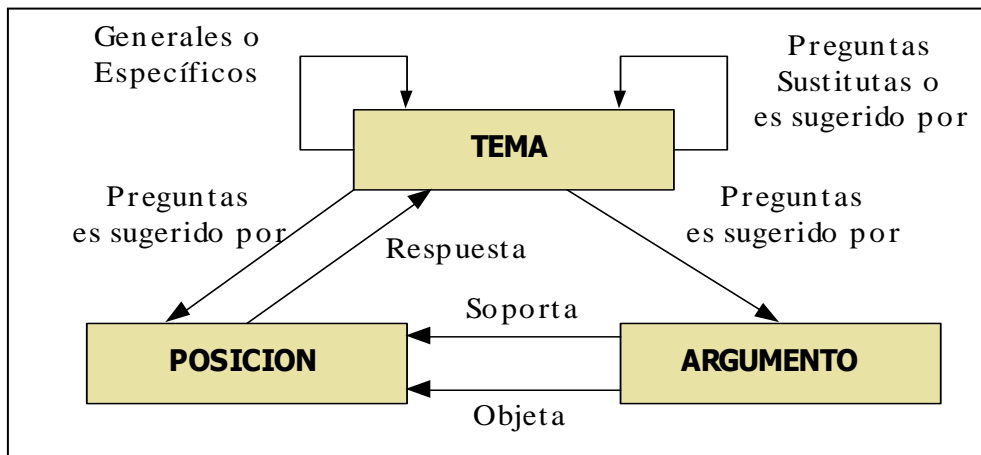


Figura 5.2. El modelo IBIS (Issue Based Information System) Kunz, W. y Rittel, H.

La estructura de IBIS consiste de: Tema (problema o pregunta), posición (posible solución o respuesta a un tema) y argumento (situación a favor o en contra relacionado a una posición). Un ejemplo para instanciar las entidades y las relaciones de dicho modelo, el cual se visualiza a continuación.

EJEMPLO:

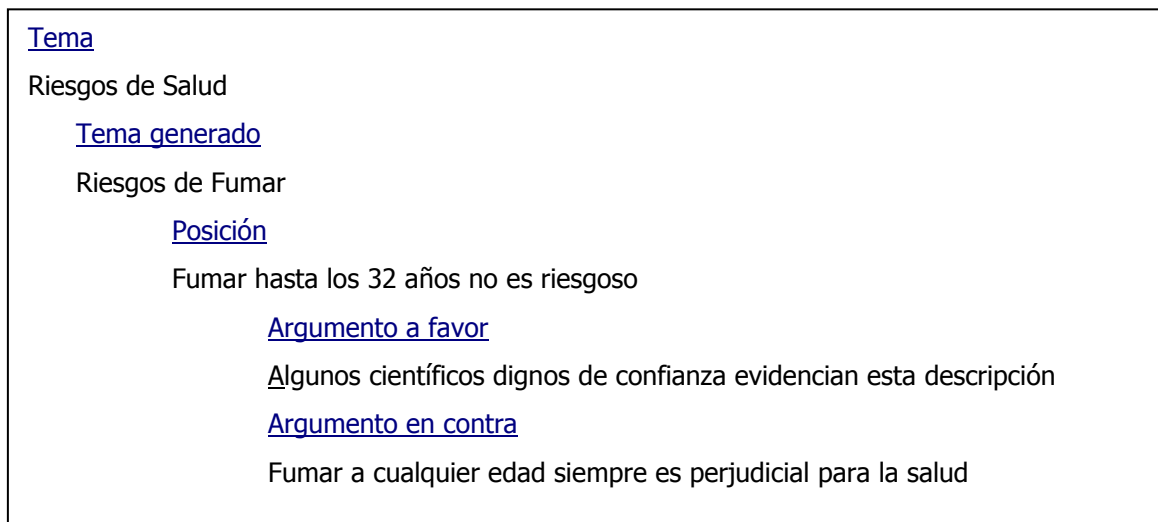


Figura 5.3. Ejemplo de una conversación utilizando el Modelo IBIS

El modelo IBIS es una plantilla de puntos de vista a considerar cuando se va a realizar una discusión sobre algún tema. La principal ventaja del modelo es su implícita intuitividad.

5.6.2 Modelo SISCO

Una extensión del modelo de argumentación IBIS es el modelo SISCO, en el que se acomodan nuevos tipos de declaraciones. Cada una de estas declaraciones es expresada más adelante. Por otro lado simplifica las relaciones de IBIS para remover algunas de ellas.

En SISCO se provee un ambiente de discusión donde no se toma ninguna decisión.

La diferencia entre IBIS y SISCO radica en que como ya se mencionó, el último es una extensión del primero. Mientras que en IBIS los objetivos son solamente representaciones implícitas, en SISCO estas relaciones están explícitamente enlazadas mediante los ítems. En general la estructura de SISCO es más motivante y expresiva que la de IBIS.

El propósito de SISCO es extender el modelo conversacional de IBIS. Los elementos básicos son: tema, posición y argumento, e incluye comentario y tarea. El modelo de datos (Figura 4-3) también incluye información acerca de los participantes, señala al coordinador de la reunión, los ítems de la agenda con cada objetivo, algunas restricciones predefinidas y se asume que se puede consultar información almacenada en un repositorio externo.

Se plantean algunos supuestos necesarios básicos para una mejor definición del modelo de datos:

- Se define una reunión asincrónica distribuida, como aquella reunión destinada a soportar los conceptos de apoyo a la reunión síncrona cara a cara.
- En las reuniones asincrónicas no se toman decisiones respecto a los ítems de la agenda de una reunión síncrona;
- A toda reunión le corresponde un coordinador, éste especifica la lista de participantes, las pre-decisiones, restricciones, ítems y objetivos;
- Los participantes interactúan en forma asincrónica, aportando temas, posiciones, argumentos y comentarios;
- Algunos temas son clasificados como propuestas, éstos en ocasiones sugieren una tarea a realizar;
- Los participantes son candidatos a llevar a cabo una tarea. El coordinador asigna a un participante la responsabilidad por una tarea;
- Las fechas tope de realización de una tarea sólo pueden ser cambiadas por el coordinador.

El enfoque de SISCO reconoce el papel de las reuniones cara a cara, especialmente para la toma de decisiones. Se enfatiza la preparación de la reunión con herramientas tecnológicas de información como una forma para hacerla efectiva y eficiente.

5.6.2.1. Elementos del modelo SISCO

Los elementos que aporta SISCO al modelo IBIS son:

- **Participante.** Se define el concepto de un participante de la discusión con el fin de identificar el origen de las declaraciones, sin embargo también se provee anonimato ya que la relación entre los participantes y las declaraciones son opcionales. También se define la especialización de un participante relacionadas al papel de coordinador.
- **Pre-decisión.** Aunque el modelo presentado no pretende tomar ninguna decisión durante la pre-reunión, se asume que sería importante almacenar decisiones de reuniones previas o afirmaciones a una discusión en la forma de pre-decisiones. Las pre-decisiones pueden en algunos casos tomar la forma de restricciones.
- **Propuesta.** Es una clase especial de resultados que sugiere una acción tal como la ejecución de una tarea o una re-definición de algún resultado.
- **Tareas.** Como un resultado de una discusión se necesita algunas veces recoger información adicional para apoyar o ayudar a analizar un tema.
- **Comentario.** En una discusión hay una necesidad de hacer declaraciones que no pueden ser clasificadas por algún elemento del modelo IBIS. Por lo tanto con el fin de evitar distorsión de los elementos básicos es creado el elemento comentario que puede agregarse como alguna declaración que no se ajusta a la clasificación del modelo IBIS original.

En la próxima sección se presenta el modelo de datos orientado a objetos que soporta estos objetos.

5.6.2.2. El Modelo de Datos Orientado al Objeto de SISCO

El enfoque central del problema es manejar la discusión de los participantes en la preparación de una reunión cara a cara, utilizando como medio una pre-reunión. Para asegurar que todo participante pueda, en cualquier momento estar informado de los elementos de la discusión presentados. El modelo SISCO provee una memoria común, la cual es el banco de datos de la discusión.

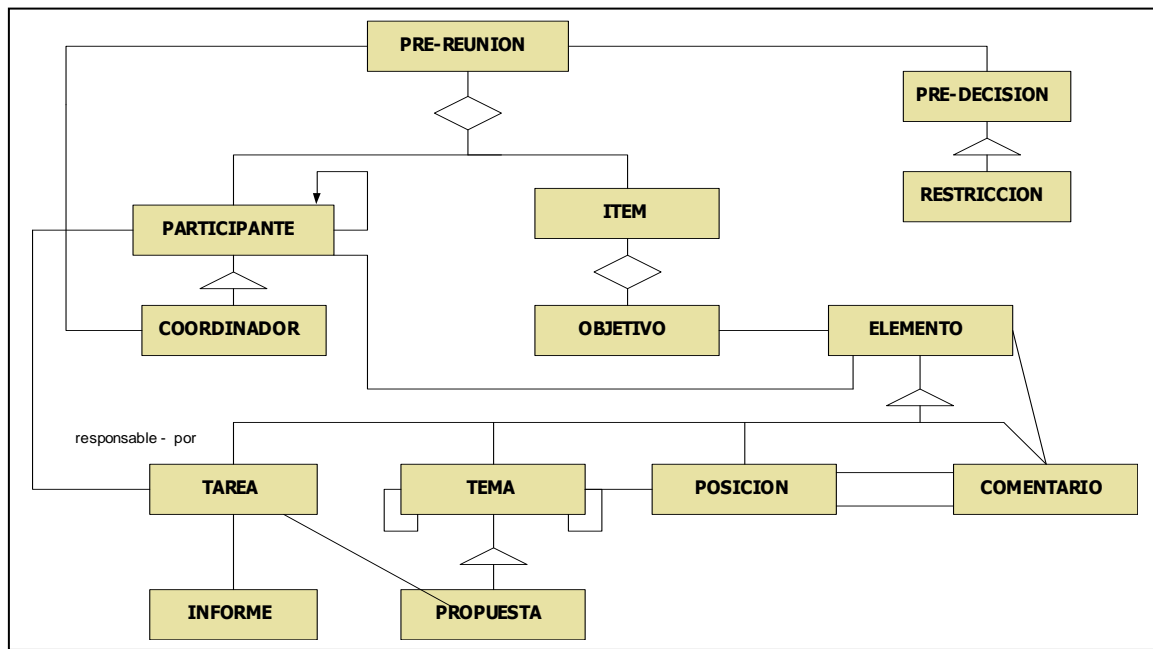


Figura 5.4. El Modelo Orientado al Objeto de SISCO (Sistema Colaborativo) Bellasai, G., Borges, M., Fuller, D., Pino, J.A., Salgado, A.C.

5.6.2.3 Ciclo de Reuniones que propone SISCO

Tener reuniones previas a las cara a cara determina muy interesantes posibilidades para que se tomen decisiones serias en los aspectos a tratar en la reunión cara a cara. Para ello el Modelo SISCO propone un ciclo de reuniones que se inicia con una reunión asincrónica distribuida en la que se discuten los aspectos a ser tratados en la reunión cara a cara sincrónica, la cual en primer instancia podría asumirse como una pre-reunión. En esta reunión asincrónica se trata de completar el encuentro posterior cara a cara, desde las oficinas de cada uno de los miembros del grupo, para así sacarle mayor provecho al encuentro. En la reunión cara a cara los miembros del grupo toman decisiones respecto a los aspectos ya discutidos en la reunión asincrónica, esto genera una nueva entrada a una reunión asincrónica, que en algunos aspectos podría considerarse como una post-reunión, respecto a la reunión sincrónica precedente, pero bajo el contexto de una nueva reunión sincrónica pasaría a ser considerada como una pre-reunión.

Las discusiones generadas en las reuniones asincrónicas, tanto como las decisiones tomadas en las sincrónicas pasarían a constituir la Memoria Organizacional del Ciclo de Reuniones SISCO.

La Memoria Organizacional es el registro de una organización que está contenido en documentos y artefactos, note que el concepto de Memoria Organizacional excluye el concepto de Memoria Grupal.

La Memoria Organizacional (MO) extiende y amplifica el conocimiento a través de la captura, organización y re-utilización del conocimiento creado por los empleados de una organización, de aquí que la MO no sólo permite acumular y preservar el conocimiento, sino que también permite compartirlo implicando con esto una mayor comunicación y aprendizaje en la organización. El ciclo de Reuniones que SISCO propone se aprecia en la Figura 5.5.

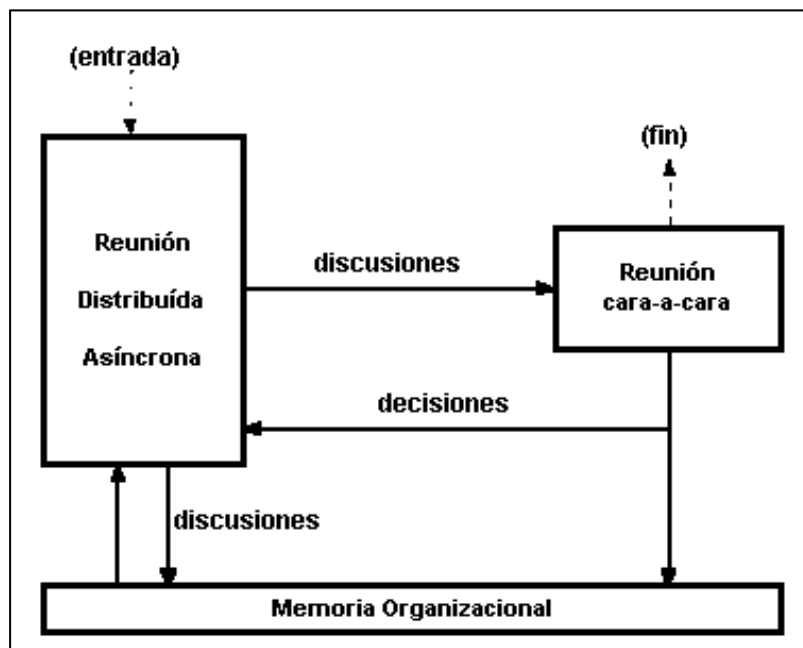


Figura 5.5. Ciclo de Reuniones SISCO

En la siguiente sección se analiza una herramienta que utiliza todos los beneficios del groupware, workflow, modelos conversacionales y porqué es posible su aplicabilidad en instituciones como la nuestra.

5.7. DOMINO Y LOTUS NOTES

5.7.1 LOTUS NOTES

Es un sistema de aplicaciones de bases de datos distribuido y basado en documentos. Es un producto de comunicaciones y colaboración en un grupo que permite a sus usuarios acceder, seguir, compartir y organizar información de forma innovadora.

La implementación de Lotus Notes en la aplicación dedicada al CUICT y su entorno:

- Permitirá a los grupos comunicarse entre sí, de forma que los usuarios puedan crear y acceder a información orientada a documentos en la Red de Gestión de Datos, canales de Internet y líneas de acceso telefónico.
- Permitirá a los usuarios seguir, compartir y organizar su información.

VENTAJAS LOTUS NOTES

- Permitirá planificar y acelerar las tareas diarias, maximizando la dedicación y atención a las tareas administrativas y operativas de proyectos de investigación.
- La integración con herramientas productivas tales como Word, PowerPoint y Excel.
- Asegurará la actualización en tiempo real del calendario y el planificador de citas y reuniones.
- Soportará una gran variedad de plataformas de hardware y de Sistemas Operativos.
- Permite que los usuarios trabajen con herramientas conocidas, y permitiendo a su vez la integración con browsers.
- Una administración eficiente de los contactos.

5.7.2. GROUPWARE (DOMINO)

Es una Base de Datos Documental que proporciona los servicios para construir aplicaciones seguras e interactivas para Internet, intranets y extranets.

VENTAJAS DOMINO

- Controla el acceso a la información permitiéndole extender su Intranet con absoluta seguridad y confiabilidad.
- Optimiza la coordinación de las actividades, en este caso el de gestión de investigación.
- Soporta un amplio número de servidores y usuarios.

- Ofrece opciones de conectividad para la integración con sistemas RDBMS y transaccionales.
- Los documentos en una base de datos Domino, pueden contener innumerables objetos y tipos de datos, incluyendo texto, texto enriquecido, texto numérico, datos estructurados, imágenes, gráficos, sonido, vídeo.

5.7.3. RESULTADOS ESPERADOS DE LOTUS NOTES Y DOMINO EN LOS QUE SE BASA EL PROYECTO

- Mayor productividad del departamento.
- Reducción de costos, papelería.
- Mejoras en la eficiencia del trabajo diario del personal CUICYT.
- Rastreo de información referente a proyectos en demora o paralizados.
- Brindar servicio on-line (en línea), y direccionar automáticamente los problemas operativos o administrativos en la gestión de investigación.
- Integración a sistemas existentes, extendiendo sus datos y resultados de proyectos a Internet.
- Permitirá la sincronización de la información y las aplicaciones a través de sitios geográficamente dispersos, como la Facultad de Ciencias de la Salud, para que los usuarios puedan acceder desde cualquier lugar y en cualquier momento.
- La búsqueda de información se incrementará en forma eficiente ya que posee un motor de búsqueda de texto que facilita la localización de documentos.

5.7.4. SERVIDORES DOMINO Y CLIENTES LOTUS NOTES

El groupware está listo para misiones críticas. Lotus Notes es el primer producto en proveer una infraestructura de groupware verdaderamente robusta, ampliable e independiente de OS. Ahora incluye importantes características de red principal, como enrutamiento dinámico, duplicación, servidores con aberturas, enrutamiento de segundo plano basado en hilos, seguridad y sistema de administración global. Ha impuesto así una nueva marca para las robustas características que serían de esperar de un servidor de groupware.

El groupware se extiende a nuevas áreas. Si tienen éxito en su inclusión en Internet, los proveedores de groupware lograrán penetrar en otras modalidades de cliente/servidor, como bases de datos y monitores de TP. Asimismo, el groupware incursiona ya en nuevas áreas, como la telefonía. Finalmente, el flujo de trabajo y los sistemas de correo basados en agentes de groupware son fuertes candidatos a administradores de procesos de negocios al interior de una empresa.

El groupware se reconcilia con las herramientas. Al igual que los monitores de TP y las bases de datos de SQL, la industria del groupware se dio cuenta de que para ganarse a los desarrolladores de software cliente/servidor necesitaba buenas herramientas. Ahora es posible crear aplicaciones de groupware en las que se incorporen datos procedentes tanto de almacenes de documentos como de bases de datos de SQL usando conocidas herramientas de cliente/servidor, como *Delphi*, *Visual Basic*, *PowerBuilder*, *SQL Windows* y *VisualAge*. Sin embargo, no sabemos de ninguna herramienta de groupware para cliente/servidor en 3 planos; por ejemplo, una herramienta que permita crear flujos de trabajo en el plano intermedio.

El groupware ha andado ya un largo camino. No obstante, es una tecnología que sigue padeciendo la falta de soporte de transacciones ACID. Resiente asimismo su condición de tecnología monolítica, poco apta para el trato de componentes distribuidos. En otras palabras, el groupware carece del fundamento de middleware para componentes de un bus de objetos distribuidos como CORBA. A pesar de ello, la tecnología del groupware está ahí, y puede servir para crear aplicaciones cliente/servidor muy interesantes, especialmente en Internet e intranets.

LOTUS NOTES 5.0

En la cuarta versión, Lotus Notes es el producto de groupware cliente/servidor por antonomasia en la industria. Algunos lo llaman el "tablero fino", otros "correo electrónico glorificado" y algunos puristas de SQL o etiquetan de "base de datos no estructurada".

En realidad, Notes es un producto multifacético de groupware cliente/servidor. El secreto de un buen paquete de groupware es que sea capaz de crear un todo que es *mucho* más que la suma de sus partes.

5.7.4.1. ¿Qué es Lotus Notes?

La versión 1 de Notes fue de conferencias, la versión 2 de información compartida, la versión 3 de flujo de trabajo y la versión 4 de aplicaciones interempresariales.

Lotus Notes permite que grupos de usuarios interactúen y compartan información que bien puede ser de naturaleza altamente desestructurada. Su desarrollo de aplicaciones cliente/servidor y entorno en tiempo de ejecución ofrecen las siguientes funciones:

- Un servidor de base de datos de documentos almacena y administra el acceso de clientes multiusuarios a datos semiestructurados, incluidos texto, imágenes, audio y vídeo. La versión

4 soporta hasta 1.000 clientes de Notes concurrentes sobre plataformas de servidor de SMP de 32 bits (orden de magnitud mayor a la versión 3).

- Un servidor de correo electrónico administra acceso de clientes multiusuarios a correo. Notes incluye una infraestructura de red principal de correo; se dispone de X.400 como componente opcional. La versión 4 integra plenamente opciones de red de X.400 y SMTP/MIME de Internet.
- Una infraestructura de red principal de servidor/servidor soporta tanto enrutamiento de correo como duplicación de base de datos. El mecanismo de duplicación sincroniza copias de la misma base de datos, la cual puede residir en máquinas servidoras múltiples (o clientes). La versión 4 soporta ya una nueva característica de paso de servidor, que permite conectarse con un servidor de Notes y tener acceso desde ahí a cualquier otro servidor para el cual se tenga autorización. Esta característica también permite tener acceso a múltiples bases de datos en servidores múltiples al mismo tiempo.
- Un entorno cliente de GUI presenta vistas de las bases de datos de documentos y ofrece un primer plano para correo electrónico. Los usuarios pueden navegar por las bases de datos y el contenido de sus documentos. Las vistas son consultas almacenadas que actúan como filtros de información en las bases de datos. El primer plano de correo electrónico es sencillamente una vista especializada de una base de datos de correo. Notes puede anexar formas de GUI (privadas o públicas) a las diversas bases de datos usadas para entrada de datos. La versión 4 introdujo el nuevo cliente Notes Mail; éste soporta una flexible interfaz de usuario trilateral que integra el cc:Mail, la visualización del Web y al cliente tradicional de Notes. Ofrece así mismo un buzón de entrada (*in-box*) universal capaz de aceptar todo tipo de correo, faxes y formatos.
- Los *servicios distribuidos* incluyen firmas electrónicas, seguridad y listas de control de acceso, servicios de administración de base de datos, administración de sistemas y un espacio global para nombres basado en X.500.
- Las *herramientas de desarrollo de aplicaciones* incluyen: 1) un generador de formas de GUI; 2) herramientas y plantillas para la creación de bases de datos; 3) un lenguaje primitivo de creación de scripts consistente en *fórmulas*, y 4) un conjunto de API abierta, que contiene la API de Notes, VIM, MAPI y ODBC. La versión 4 soporta *agentes inteligentes*. Éstos son aplicaciones de Notes con scripts útiles para la automatización de tareas repetitivas, como duplicación de bases de datos, manipulación de datos, las acciones de buzón de entrada y los

servicios de manejo de mensajes. La versión 4 introduce igualmente *LotusScript* 3.0, lenguaje de programación orientado a objetos semejante a BASIC y capaz de ser aplicado entre plataformas. Además, usted puede generar aplicaciones de Notes mediante el uso de herramientas muy conocidas de cliente/servidor de terceros, como *Delphi*, *SQLWindows*, *PowerBuilder*, *New Era*, *VisualAge*, *Notes ViP* y *Visual Basic* (vía los VBX *HiTest* proporcionados por Lotus).

Todas las comunicaciones de Notes cliente/servidor y servidor/servidor se realizan con una RPC propietaria. Notes soporta controladores de cliente/servidor para pilas NetBEUI, TCP/IP, IPX/SPX y AppleTalk. Para las comunicaciones servidor a servidor se dispone de controladores de APPC y X.25 opcionales.

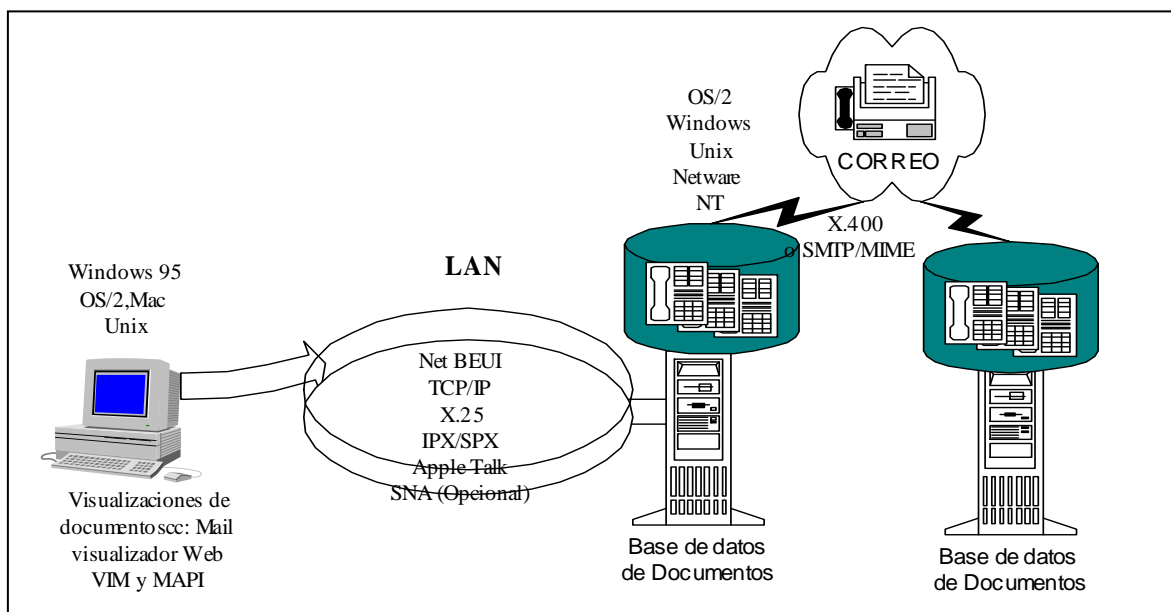


Figura 5.6. Componentes de Lotus Notes.

5.7.4.2. La base de datos de documentos con multimedia

Ray Ozzie, fundador de Iris Associates, compañía que desarrolló el Notes original bajo contrato con Lotus describe los cimientos de la arquitectura de Notes como "un mecanismo de base de datos para información semiestructurada y no estructurada", Este modelo se asemeja más a las conferencias en computadora que al procesamiento de transacciones en línea (OLTP: *online transaction processing*). La base de datos de Notes fue diseñada como vehículo para la reunión y disseminación de todo tipo de información; no se pensó como "base de datos de registros" para que refleje el estado en tiempo real de la empresa. A este respecto, Notes tiende a ser una bodega de datos, salvo que los datos suelen ser muy eclécticos. Para decirlo de otra manera, a

Ray Ozzie le interesó más conjuntar y capturar información de tiempo real que ofrecer acceso sincronizado a datos compartidos para actualizaciones.

La unidad primaria de un sistema Notes es un *documento* semiestructurado con multimedia que puede contener una amplia variedad de tipos de datos, como voz, BLOB, vídeo y texto de fuentes múltiples.

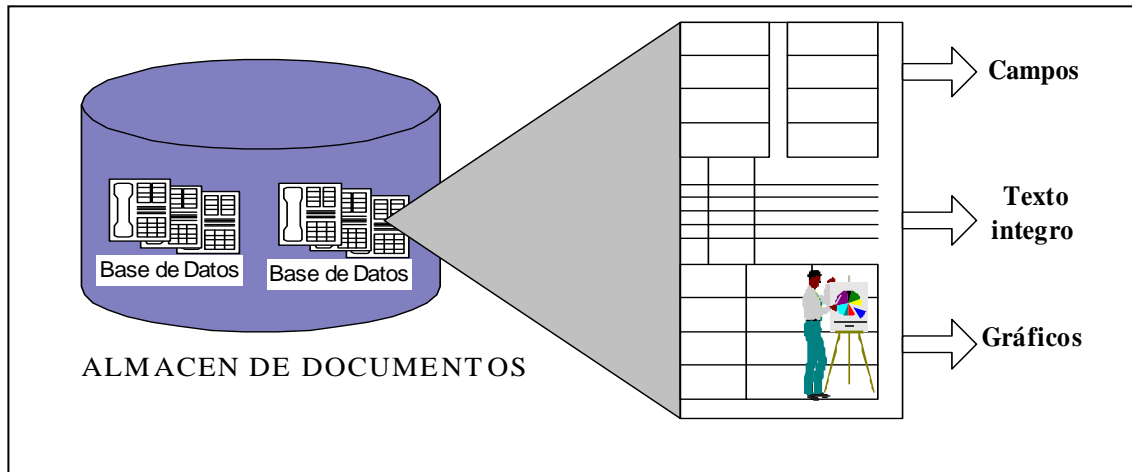


Figura 5.7. Jerarquía de almacén de documentos Notes.

5.7.4.3. Potencialidades de Notes

Un sistema Notes organiza, almacena, duplica y ofrece acceso compartido a documentos. Conjuntos afines de documentos de Notes se almacenan en una *base de datos*, que usted puede indexar y acceder por medio de cualquiera de las propiedades de los documentos, o del contenido real de éstos. Notes soporta indexación y búsqueda de texto íntegro. Un *documento* de Notes se compone de un conjunto de campos, también conocidos como *propiedades*, cada uno de los cuales posee un nombre, tipo y valor. Por ejemplo, usted puede etiquetar un documento de Notes con propiedades como *cliente*, *región* y *tema*. Las regiones pueden contener todo número de *anexos* (o archivos incorporados) semejantes a BLOB. Los archivos incorporados son administrados y organizados como parte de un documento de Notes.

Notes permite crear bases de datos verdaderamente gigantescas. Aquí algunas de las nuevas capacidades:

- **Los almacenes de objetos pueden ser de tamaño ilimitado.** Un administrador asigna tamaños máximos de almacén de objetos de porciones hasta de 4 GBytes, llamadas

segmentos. Notes tiende automáticamente datos entre segmentos múltiples y fronteras de almacenamiento físico.

- **Las bases de datos pueden contener un número ilimitado de documentos.** Notes almacena una base de datos completa en un solo archivo como objeto. Una aplicación Notes se compone habitualmente de muchas bases de datos organizadas por temas.

5.7.4.4. Cómo se crea una aplicación Notes

Por lo general se crea una nueva *base de datos* de Notes con una de las plantillas provistas por Lotus, la cual es sometida a personalización. Una base de datos es simplemente un nuevo archivo; puede asignársele un icono identificador, título, paneles de ayuda y un *documento de políticas* en el que se explique de qué trata el asunto. Se *usan formatos* para introducir o visualizar información en una base de datos. Para crear un formato, se puede partir de uno de los formatos preexistentes para modificarlo con el editor de GUI. Los formatos ofrecen campos de entrada de datos, campos de texto y áreas gráficas para la inserción (o incorporación) de imágenes u otras fuentes de datos multimedia. Lotus proporciona un lenguaje macro similar a las fórmulas de hoja de cálculo de 1-2-3, para asociar comandos con eventos y acciones específicos. Además, se puede usar *LotusScript* para generar manipuladores de eventos.

En el futuro usted podrá generar sus manipuladores de eventos de Notes con el lenguaje Java. Los formatos que crea se asocian con la base de datos. Puede asignarlos como públicos, lo que significa que está a disposición de todas las aplicaciones cliente con acceso a la base de datos, o como *privados*, lo que significa que sólo su creador podrá usarlos.

Las *vistas* son consultas almacenadas que muestran el contenido de una base de datos o de un documento en particular. Sirven para la navegación y filtración de información; como ejemplo puede citarse el caso de muestra de documentos de hasta un mes "por región" o "por vendedor".

La vista exhibirá la lista de documentos en formato tabular o resaltado. Todas las bases de datos cuentan con una o más vistas creadas por el diseñador para facilitar el acceso a la información. Los usuarios también pueden crear vistas *privadas* para disponer de listados o criterios de acceso no previstos por el diseñador de la base de datos.

5.7.4.5. La API de Lotus Notes

Notes brinda también una API de programación en C++, que los desarrolladores pueden usar para almacenar y acceder a documentos de Notes. Asimismo, da amplio acceso a muchas de las características de la interfaz del usuario de Notes. La API le permite:

- Crear o borrar bases de datos.
- Leer, escribir y modificar cualquier documento y cualquier campo del documento.
- Crear y usar vistas de base de datos.
- Controlar el acceso a las bases de datos con listas de control de acceso.
- Reunir y reportar estadísticas de desempeño del servidor y registrar nuevas estaciones de trabajo y servidores.
- Generar tareas personalizadas que puede añadir al software servidor de Notes y especificar el itinerario de ejecución de la tarea personalizada.
- Crear, leer y correr macros de Notes con el uso de la API.
- Realizar búsquedas de texto íntegro con el nuevo mecanismo de búsqueda.
- Emitir llamadas para restringir los documentos por intercambiar durante la duplicación.
- Obtener la lista de nombres y libretas de direcciones en uso local o en un servidor
- Emitir llamadas de gateway por correo.
- Además, Notes soporta las API del cliente de correo electrónico tanto de VIM como de MAPI, lo mismo que ODBC para acceso a bases de datos de SQL.

Notes soporta las API del cliente de correo electrónico tanto de VIM como de MAPI, lo mismo que ODBC para acceso a bases de datos SQL.

5.7.4.6. Notes y las bases de datos de SOL

Una API para base de datos externa basada en controladores de DataLens puede ser adquirida sin costo. Esta permite a los usuarios de Notes tener acceso a datos almacenados en bases de datos externas mediante claves y capacidades de búsqueda de Notes. Los datos externos pueden mostrarse en campos de Notes. Además, Lotus proporciona una interfaz de base de datos relacional que soporta la CLI de ODBC. El manejador de ODBC permite a los programadores generar consultas de SQL susceptibles de ser traducidas a consultas de Notes. Esto ofrece la posibilidad de insertar datos de Notes en una base de datos relacional. En el siguientes gráfico se muestra la capacidad bidireccional Notes/SQL.

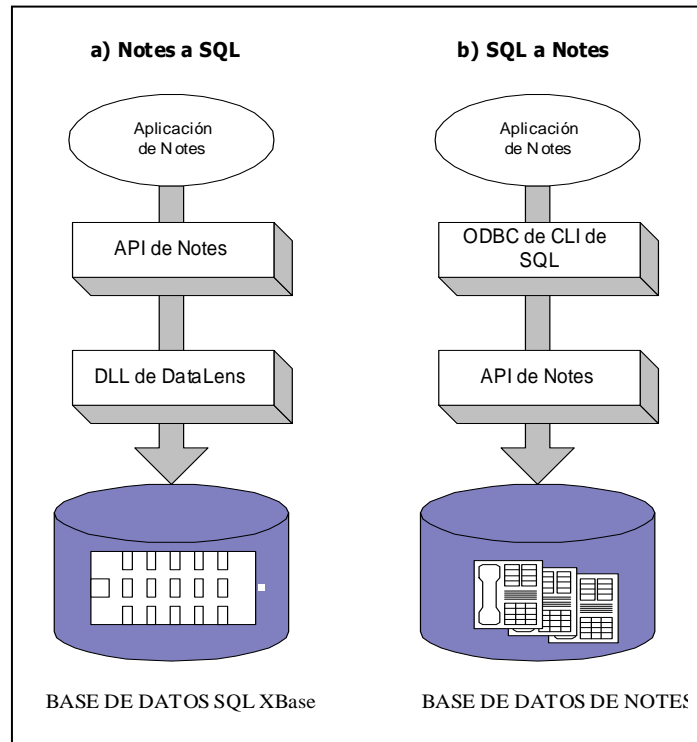


Figura 5.8. Intercambios bidireccionales de datos entre bases de datos de Notes y SQL.

5.7.4.7. Correo electrónico de Notes

Desde la perspectiva de la estación de trabajo cliente, el correo electrónico es simplemente otra base de datos de Notes, la cual contiene un grupo de documentos de correo. Los procedimientos para la lectura del correo recibido, su clasificación o la creación de un documento de correo son los mismos que se emplean para crear y leer documentos de cualquier otra base de datos de Notes. Sencillamente se usan formatos y vistas a la medida de los documentos de correo propios. Por supuesto que una de las diferencias es que los documentos de correo que usted crea serán enviados al buzón de alguien. Notes ofrece indicadores visuales para hacerle saber que ha recibido correo. Se abre la base de datos y la lee. El diseño de Ray Ozzie parece ser muy consistente.

Las funciones de red principal de correo, presentan las siguientes características:

- **Optimización de enrutamiento.** Las técnicas incluyen prioridad en mensajes no acoplados y selección de ruta de adaptación dinámica basados en costos de vinculación.
- **Hilos enrutadores independientes.** Todas las comunicaciones de servidor a servidor son manejadas por hilos de transferencia independientes. Los hilos permiten la ocurrencia de múltiples transferencias simultáneas en diferentes rutas de la red. Además, impiden que mensajes grandes de correo retrasen otras tareas del servidor.

- **Notificación de falla de entrega.** Los remitentes pueden ser notificados cuando no es posible realizar una entrega (y los motivos de ello).
- **Soporte de espacio para nombres de X.500.** Notes soporta en su totalidad la asignación de nombres jerárquicos compatibles con X.500 como medio propio de identificación de *usuarios* dentro del sistema. Esto facilita a los directorios de Notes interoperar en el nivel de asignación de nombres con otros sistemas compatibles con X.500. Evita dolores de cabeza por conflictos de asignación de nombres.
- **Gateways de correo y servicios de directorio.** Notes proporciona gateways de correo electrónico a las redes de correo electrónico más populares, como X.400, SMTP, cc:Mail, MHS, PROFS, Exchange, VinesMail, Fax, VAXmail y SoftSwitch.

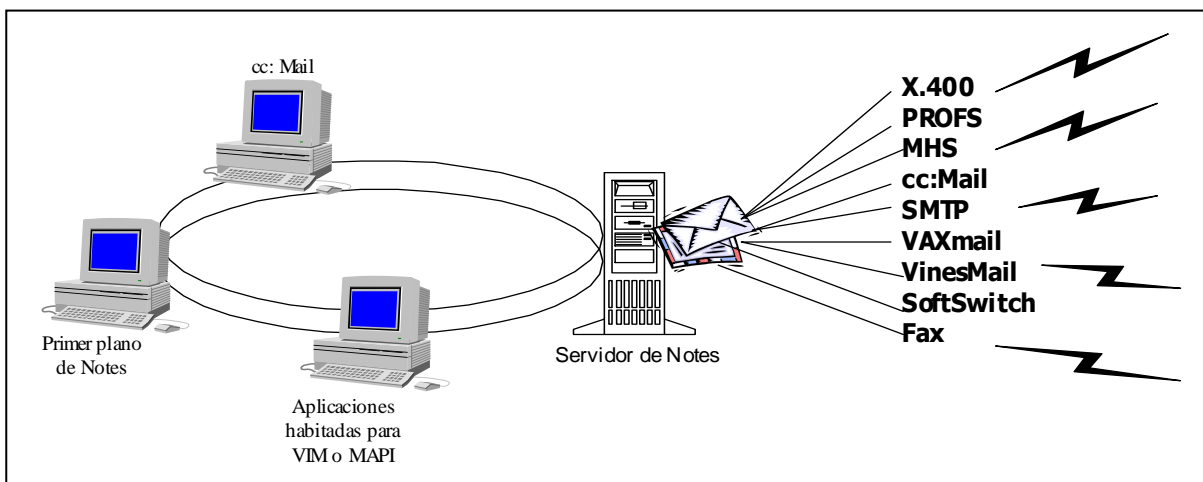


Figura 5.9. Alcance del correo electrónico de Notes a través de gateways.

- **Firmas electrónicas.** Notes usa la criptografía de llave pública de RSA para todo lo relacionado con la seguridad de Notes, incluida la codificación. RSA es apta para firmas electrónicas, pero muy lenta en lo que se refiere a codificación general. Si usted firma un mensaje, a Notes le lleva unos segundos más enviarlo, porque debe generar una firma electrónica de RSA. Si la sola codificación de un nombre tarda un par de segundos, ¿cuánto tardará la codificación de un documento de tamaño mediano?

El mecanismo de correo de Notes 4.0 sirve de primer plano lo mismo de Notes que de cc:Mail. Notes brinda una auténtica arquitectura de correo electrónico cliente/servidor: Clientes y servidores se comunican a través de RPC. VIM y MAPI ofrecen la interfaz del nivel de llamada en la parte del cliente.

CAPITULO 6

Aplicación



El aplicativo consiste en dos aplicaciones que comparten información en común, pero las herramientas que se utilizan como bases de datos y frontales son muy distintos mismos que tienen funcionalidades acordes a la información que requieren manejar, el esquema de las dos aplicaciones se muestra en el siguiente gráfico:

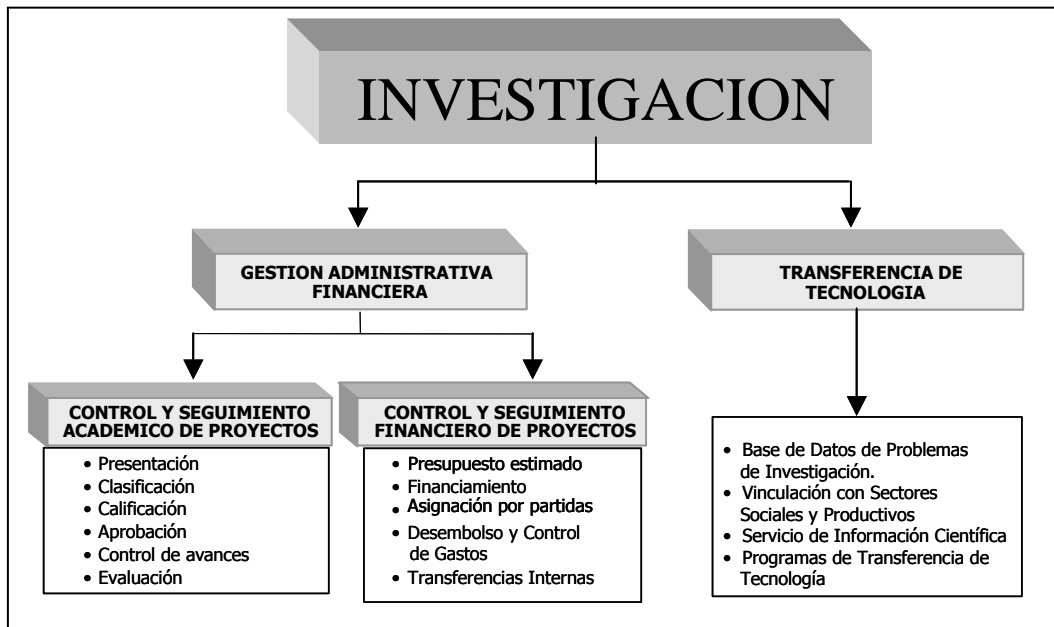


Figura 6.1. Visión global del proyecto GEDI-UTN.

En este capítulo se muestra un extracto de las principales funciones del aplicativo, y su enfoque en la segunda etapa en el modo de enlace con bases de datos documentales, junto a su instalación.

6.1. GESTION FINANCIERA

Los procesos financiero juegan un rol protagónico en el manejo de proyectos, por lo que se hace necesario un análisis profundo de estos procesos operativos, y determinar cuál sería una herramienta adecuada y óptima para el tratamiento de este tipo de información.

6.1.1. DETERMINACION DEL TIPO DE INFORMACION

La mayor parte de la información son de tipo numérico (presupuestos, control de gastos, costos de bienes, transferencias internas de dinero entre otra) y texto como información del personal que maneja proyectos, organización institucional (facultades, escuelas, algunas definiciones de datos) entre otros, estos son justificativos que nos llevan a determinar que lo ideal es utilizar como repositorio a las Bases de datos Relacionales.

El manejo de este tipo de información y la frecuencia con la que se procesan, llevan a determinar que lo más adecuado es manejar esta información en Bases de Datos Relacionales.

6.1.2. ANALISIS Y DISEÑO DEL MODELO FINANCIERO

Luego de sesiones de trabajo con personal del CUICYT, su director y personal operativo, se ha hecho el análisis del flujo de trabajo, flujo de procesos y de información que se debe manejar en el control Financiero, se ha procedido con los análisis de: el flujo de trabajo, flujo de Procesos y su información, identificando los principales actores de los procesos.

El Modelos Entidad – Relacional, que se propone es el mostrado en la figura 6.2, cuyo modelo ha estado sometidos a cambios frecuentes hasta obtener un modelo adecuado y flexible. Eso se ha logrado haciendo constantes pruebas en prototipos iniciales hasta tener un modelo aceptable, eso quiere decir que el usuario final operativo será el que determine si los requerimientos han sido satisfechos en su totalidad, o parcialmente satisfechos en donde se debió reacondicionar el modelo con nuevos cambios.

El éxito de un sistema estará dado, en cuanto menores sean los cambios a la programación interna, sino más bien estos cambios se los efectúe a nivel de datos, es decir parámetros manejables del sistema informático financiero.

6.1.3. MODELO RELACIONAL

El modelo relacional se genera posterior al modelo entidad-relación, este modelo se presenta en la siguiente página, y fue diseñado en un modelador para bases de datos, Erwin 4.0, aplicando Ingeniería Inversa, por haberse estado desarrollando como prototipo en Access 97, para posterior en la etapa de producción generarla para cualquier motor de base de datos.

Figura 6.2. Modelo Relacional, GEDI-UTN.

6.1.4. LA BASE DE DATOS RELACIONAL

Por la baja cantidad de datos procesada y frecuencia de transacciones que se deben procesar se podría compara a los procesos de un sistema de rol de pagos de baja escala, pero la dificultad de los procesos se comparan a los procesos de una sistema contable y de presupuesto; de todo esto podemos concluir que podríamos necesitar de una Base de Datos de mediana escala: SQL-Server 7.0 o 2002, o el mismo Access, que brinda las facilidades de conexión entre clientes Access.

Actualmente como prototipo de desarrollo y pruebas se tiene la siguiente estructuración de la Base y frontal:

BASE DE DATOS: SQL – SERVER 2002 (Prueba), Access 97 (Producción)
CLIENTES, FRONTALES: Access 97

Se podría utilizar como base de datos en producción SQL-SEVER 7.0 o 2000, Informix 9.20 NT, una equivalente, o una de menor proporción.

El inconveniente podría ser las licencias que dispone la Universidad, por lo que una base de datos Access es una opción económicamente accesible.

6.1.5. CODIFICACION CLIENTES ACCESS

Los clientes Access pueden ser configurados para trabajar como frontal de cualquier base de datos mediante ODBC, y tomar a las tablas de otro motor de BDD como suyas.

La programación no se pierde, será la misma lo que ha cambiado es la ruta lógica de acceso, que en lugar de apuntar a una base de datos local, apuntará a una base de datos que puede estar en cualquier parte de la red LAN.

A continuación se muestran algunos de los procesos principales del control financiero.

El acceso: está dado por un acceso a usuarios del sistema, donde se pide Usuario y Clave de Acceso.



Figura 6.3. Acceso al Sistema Financiero GEDI-UTN.

Dependiendo del tipo de usuario se le presentará la barra de trabajo o menú correspondiente al tipo de usuario.

Existe un administrador de proyectos que permitirá las acciones de:

- Crear proyectos nuevos, o formular.
- Asignar Presupuestos.
- Aprobar Proyectos.
- Controlar Gastos.
- Autorización de Gastos.

Estos procesos se activan conforme al estado de los proyectos.

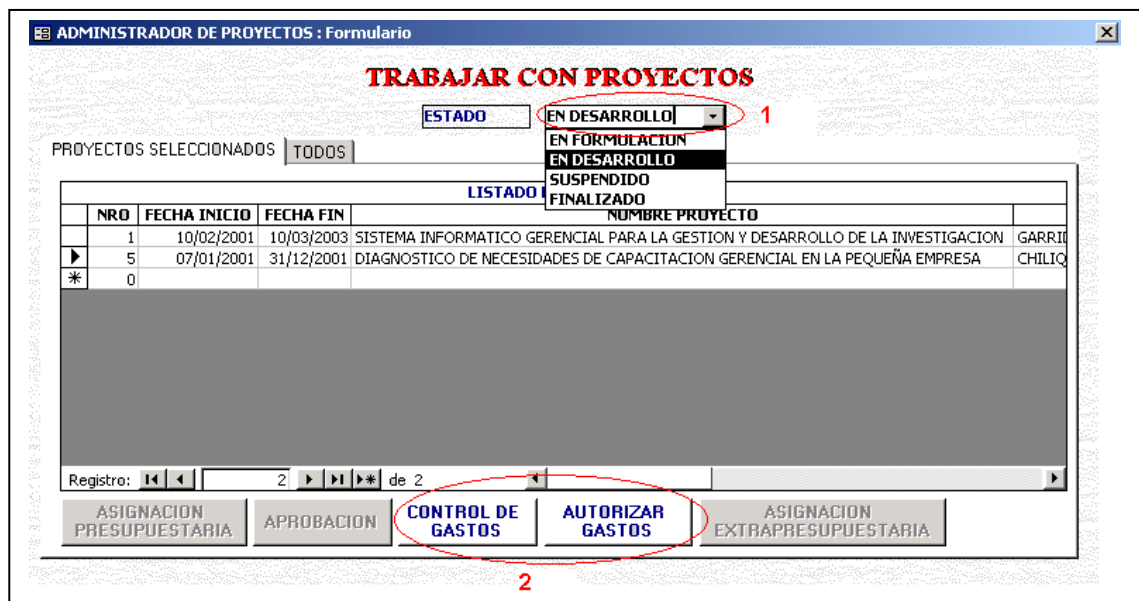


Figura 6.4. Administrador de Proyectos.

La Asignación Presupuestaria.

Consiste en asignar el presupuesto previsto para un proyecto donde se deben especificar o justificar los gastos por cada partida creada, en cada mes de los períodos previstos para dicho proyecto.

Los primero será crear las partidas del proyecto, para luego continuar con la justificación de gastos por mes, los cuadro combinado o Combobox, tiene opciones abiertas para que el administrador de proyectos pueda crear alguna categoría de Rubro, y los ítem que lo componen:

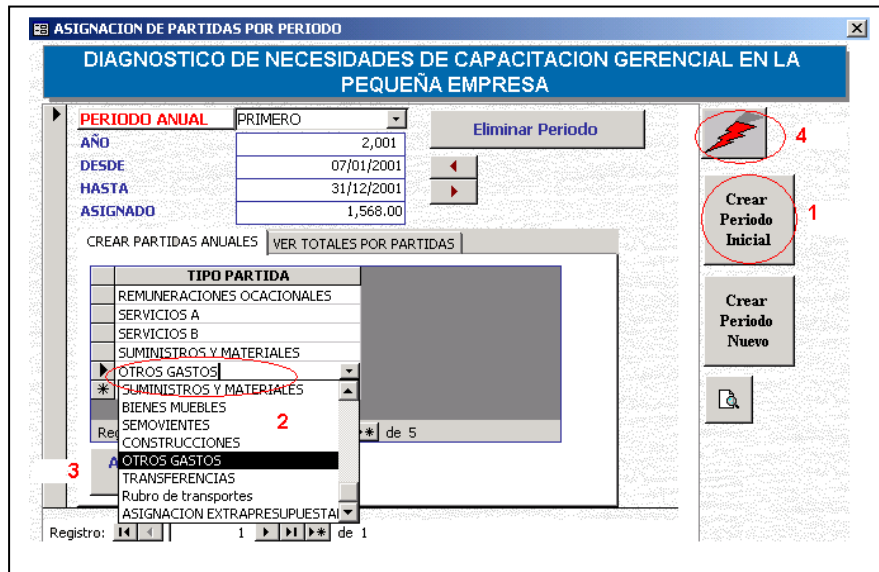


Figura 6.5. Asignación de Partidas a Proyectos.

En caso de no existir alguna partida se procederá haciendo doble click sobre el cuadro combinado TIPO PARTIDA.

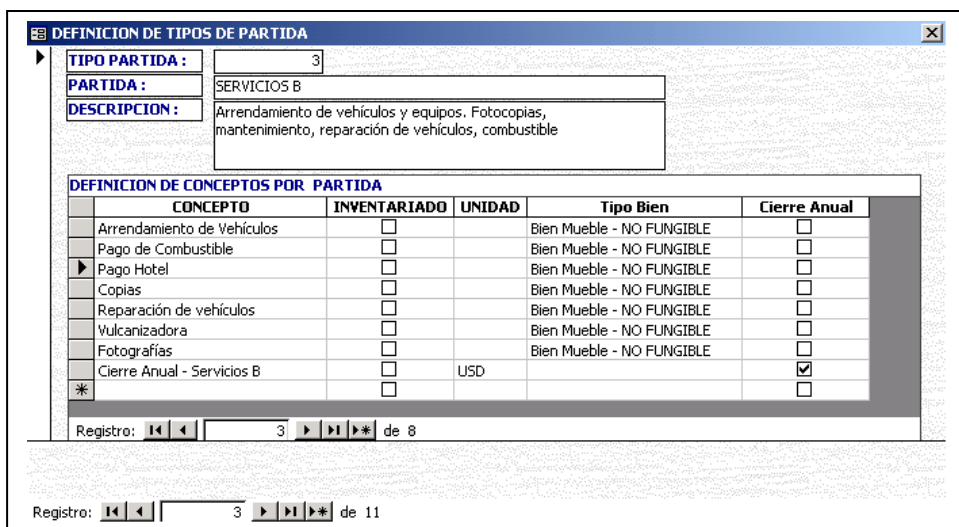


Figura 6.6. Definición de Tipos de Partida para proyectos.

Posteriormente se deberá proceder a seleccionar el mes que corresponda para llenar los justificativos o presupuesto del proyecto, como en el siguiente gráfico:

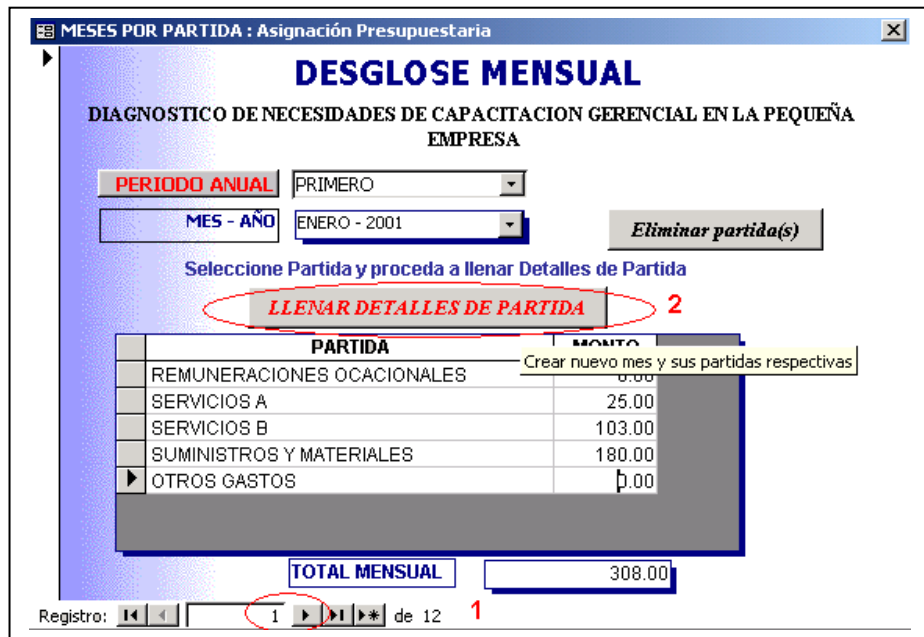


Figura 6.7. Selección del mes a justificar el presupuesto.

Cuando se ha seleccionado el mes se procede a justificar el presupuesto para el mes y rubro especificado, como sigue:

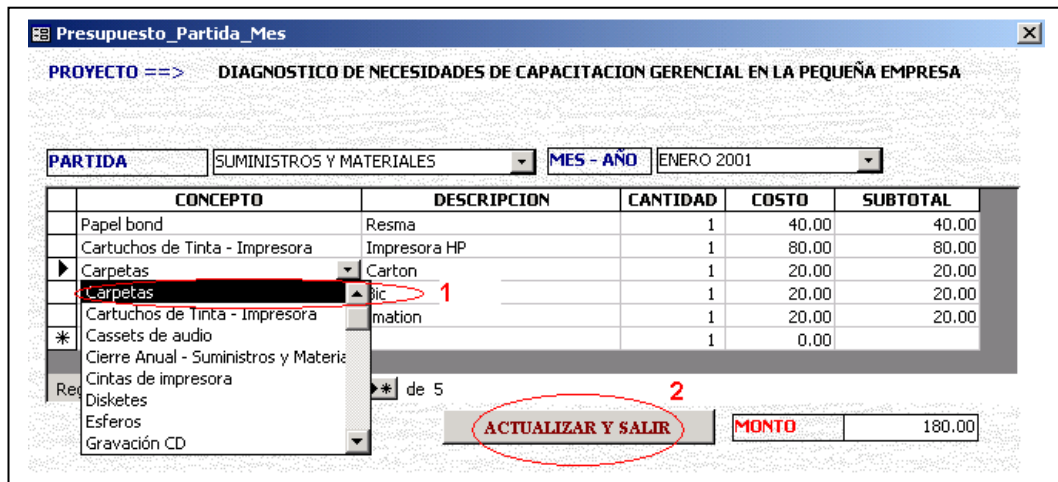


Figura 6.8. Justificación del presupuesto, para la partida Suministros y Materiales en Enero 2001.

Una vez llenado el presupuesto justificado para cada mes que dure el proyecto y por partida, se procede a cerrar el presupuesto como en la figura 6.5, botón 4.

Aprobar Proyectos.

La aprobación se la efectúa una vez que se ha llenado el presupuesto, caso contrario no será considerado como aprobado.

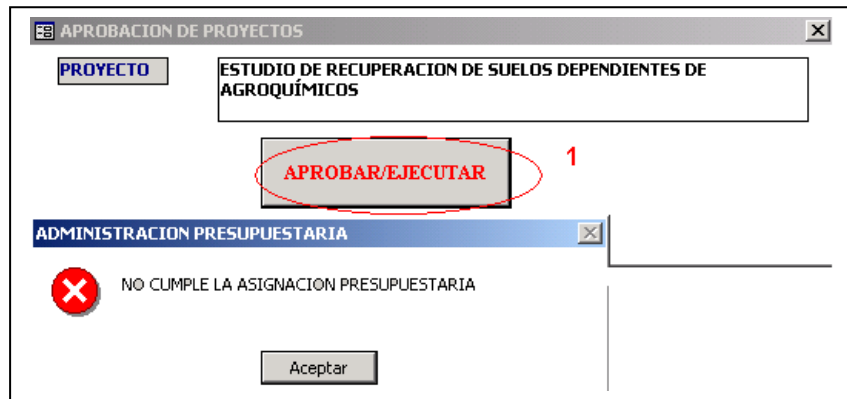


Figura 6.9. Formulario para procesar aprobación de proyectos.

Autorización y Control de Gastos

La autorización de gastos se la hace luego de haber sido aprobado dicho proyecto por medio del formulario de la figura 6.10.

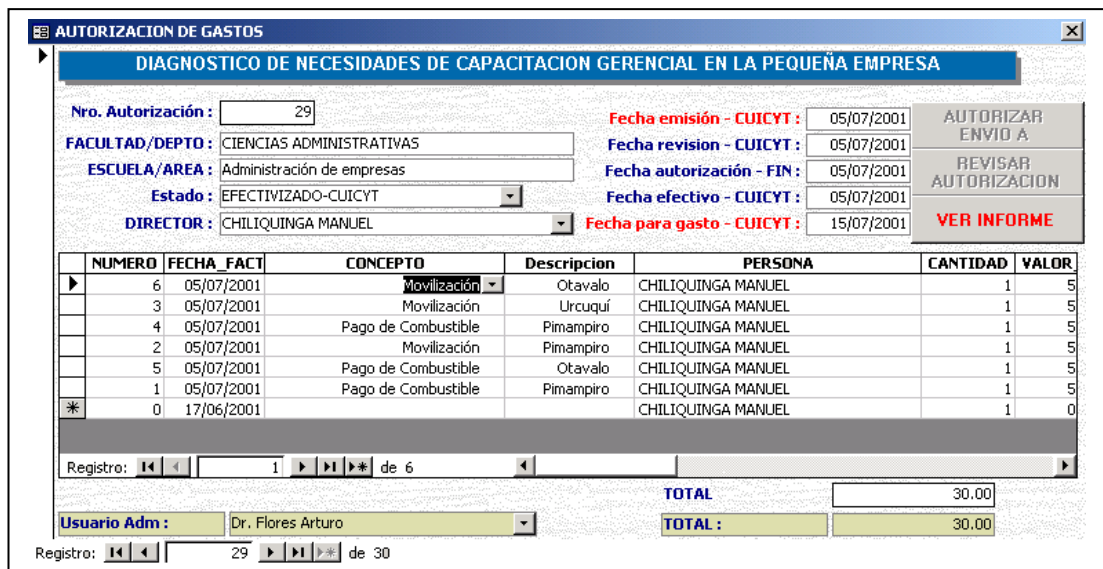


Figura 6.10. Autorización de Gastos para un Proyecto.

Luego existen procesos de autorización Financiera de la UTN, otro departamento que interviene, en estos procesos.

Entre otros procesos se encuentra, Curriculum de Personal de Investigación, Registro de Bienes, y un conjunto de Reportes.

Registro de Bienes

El registro de bienes se activa cuando en la Autorización de Gastos consta un bien clasificado como inventariado.

The screenshot shows a web application window titled 'PROYECTO: BIENES'. The main section is 'INFORMACION GENERAL DE PROYECTOS DE INVESTIGACION'. It contains several input fields: 'BUSCAR PROYECTO' (empty), 'ID_PROYECTO' (1), 'ESTADO PROYECTO' (EN DESARROLLO), 'NOMBRE PROYECTO' (SISTEMA INFORMATICO GERENCIAL PARA LA GESTION Y DESARROLLO DE LA INVESTIGACION), and 'DIRECTOR' (GARRIDO SANCHEZ FERNANDO JOSE). Below this is a table titled 'Bienes detalle' with columns: Id_bien, BIEN, Caracteristica, Fecha_adquisicion, and Proveedor. The table lists 7 items, including computers and printers. At the bottom, there are two buttons: 'INFORME BIENES DEL PROYECTO' and 'DETALLE PARTIDA GASTO'. Navigation controls show 'Registro: 5 de 12' and 'Registro: 1 de 5'.

Id_bien	BIEN	Caracteristica	Fecha_adquisicion	Proveedor
1	Computador		16/02/2002 18:21:37	FERROINDUSTRIAL
2	Computador		16/02/2002 18:21:37	FERROINDUSTRIAL
3	Impresora		16/02/2002 18:21:37	FERROINDUSTRIAL
4	Anaqueles	UNO	18/02/2002 17:59:01	LA MACARENA
5	Impresora	EPSON LX-300	18/02/2002 17:59:01	WORLD COMPUTERS
6	Computador		15/03/2002 19:06:24	WORLD COMPUTERS
7	Impresora		15/03/2002 19:06:24	WORLD COMPUTERS

Figura 6.11. Registro de Bienes.

Curriculum Personal de Investigación

The screenshot shows a web application window titled 'DATOS DE INVESTIGADORES'. The main section is 'CURRICULUM VITAE INVESTIGADORES'. It contains several input fields: 'ID_PERSONA' (empty), 'CI' (100218058-4), 'APELLIDOS' (PITA SEVILLA), 'NOMBRES' (ELCIRA), 'FACULTAD / DEPARTAMENTO' (CIENCIAS ADMINISTRATIVAS), 'ESCUELA' (Contabilidad Superior y Auditoria), 'TITULO' (INGENIERO), and 'AREA ESPECIALIZACION' (COMERCIAL). Below this are tabs for '1. FORMACION', '2. CURSOS Y POSTGRADOS', '3. EXPERIENCIA', '4. POR REALIZAR', '5. HORARIO', and '6. TRABAJO'. The '1. FORMACION' tab is active, showing a table with columns: TITULO OBTENIDO, INSTITUCION DONDE SE GRA, LUGAR, and DURACION. The table lists two entries: 'LICENCIADA CONTABILIDAD Y AUDITORIA' and 'INGENIERIA COMERCIAL'. At the bottom, there are navigation controls showing 'Registro: 1 de 2' and 'Registro: 1 de 1 (Filtrado)'.

TITULO OBTENIDO	INSTITUCION DONDE SE GRA	LUGAR	DURACION
LICENCIADA CONTABILIDAD Y AUDITORIA	PUCE-I	IBARRA	6 AÑOS
INGENIERIA COMERCIAL	PUCE-I	IBARRA	1 AÑO

Figura 6.12. Formulario de Datos Personales y Curriculum.

Existen un conjunto de procesos para el Administrador - CUICYT, el Cliente común y personal del departamento Financiero.

6.2. GESTION ACADEMICA

Los procesos académicos son de cierta manera, procesos formales de los proyectos de investigación, no requieren transaccionalidad constante, pero si requieren de organización y control.

6.2.1. DETERMINACION DEL TIPO DE INFORMACION

Esta información requiere ser organizada y documentada, la mayor parte de información no son datos numéricos, sino más bien datos no cuantificables, como documentos de aprobación de proyectos, resoluciones tomadas, y por sobre todo lo que se necesita es coordinar con las facultades a través de los coordinadores de investigación, sobre los justificativos de avances de proyectos, a continuación se listan algunos de los procesos de gestión académica que se requiere mejoren en su tratamiento para beneficio de una mejor organización y coordinación de trabajo:

- Presentación de perfiles de proyectos.
- Clasificación
- Calificación
- Aprobación
- Control de avances y
- Evaluación

6.2.2. INSTALACION DEL LOTUS NOTES, DOMINO y CONFIGURACION

Para la Instalación del Servidor Domino elegimos Install.



Figura 6.13. Instalación de Herramientas Lotus Notes.

A continuación se muestra una pantalla que indica el tipo de instalación que se desea hacer en el equipo para clientes, servidores u otro, si la implementación de la aplicación requiere de la base de datos domino se elige Servers, la cual muestra 3 opciones:

- Servidor Mail
- Servidor de Aplicaciones
- Servidor Empresarial de Domino

Por motivos de errores en la instalación aplicando la tercera opción es preferible instalar una a una las dos primeras: el Servidor Mail y el Servidor de Aplicaciones.



Figura 6.14. Instalando Clientes y Servidores.

Finaliza el proceso de copia de los componentes de la aplicación.



Figura 6.15. Finalizando la Instalación.

Una vez que se haya hecho la instalación del programa se despliega una pantalla que indica su finalización.

Lotus Notes R5 es un entorno que mejora la eficacia del trabajo en equipo.

Independientemente del tipo de plataforma, los usuarios pueden trabajar juntos, lo que agiliza el trabajo en grupo. Además no depende del punto geográfico donde no encontremos.

La tecnología de Groupware (trabajo en grupo) hace que Lotus Notes esté integrado totalmente a la misma. Esta tecnología tiene tres características principales:

1. La Comunicación: Lotus Notes tiene como unas de las características principales tener un gestor de correo, lo que hace de él que nos podamos comunicar de forma fácil y rápida con otras personas.
2. La Colaboración. La existencia de las Bases de Datos, genera que sea sencillo compartir Datos desde cualquier punto del planeta.
3. La Coordinación. Podemos construir flujos de trabajo completamente controlados.

Como podemos observar Lotus Notes no es simplemente un programa de correo, el potencial y funcionalidad que podemos aportar a una organización hace de él que sea un producto a tomar en cuenta en cualquier empresa

Ya ha pasado la era de enviar información por papel. Todo se envía por el correo electrónico, uniendo más rápidamente a personas y organizaciones, además de ser un soporte de almacenamiento y medio de transporte de la información.

Configuración

Una vez instalado todos los servicios, se debe levantar el configurar el servidor DOMINO, y levantar el servidor Domino, para que ejecute todas las tareas, como se muestra a continuación:

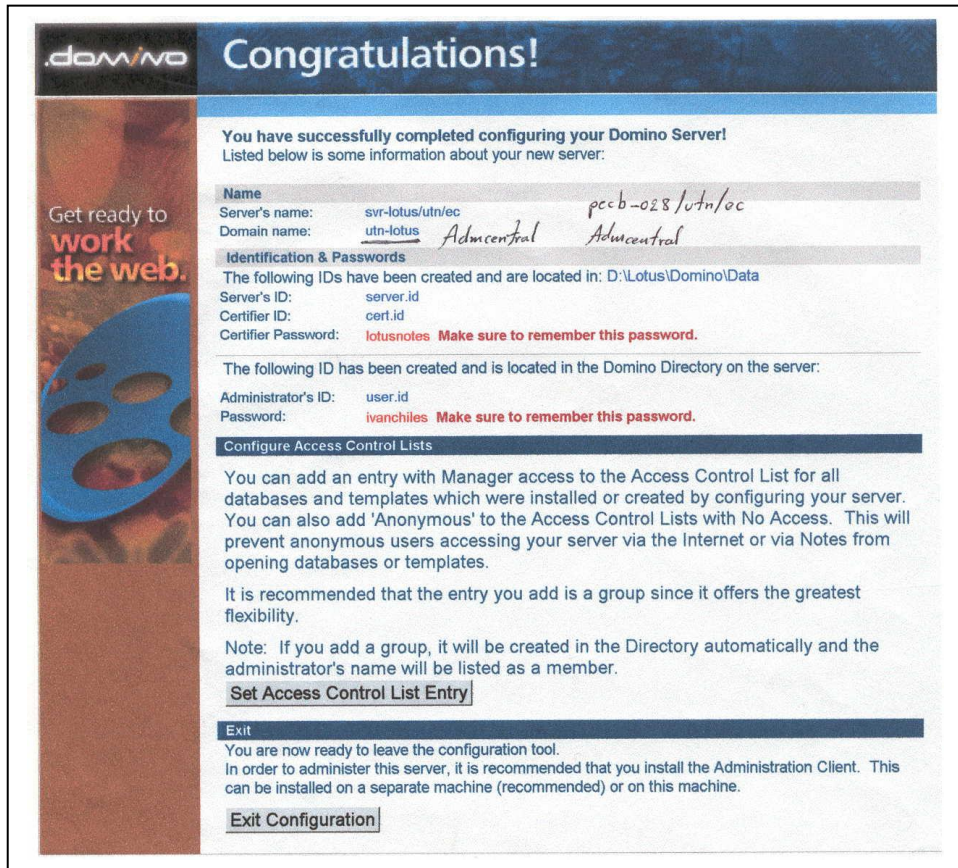


Figura 6.16. Finalizando la Instalación.

Levantar el Servidor Domino, desde **c:\Lotus\Domino\server.exe**, con lo que queda listo para recibir las peticiones de los clientes.

Lotus Notes

Es la ventana que aparece al arrancar Lotus Notes y tiene un aspecto similar al siguiente cuadro. Nos muestra las opciones para acceder rápidamente a nuestro correo, agenda, libreta de direcciones, tareas y algunas opciones más.

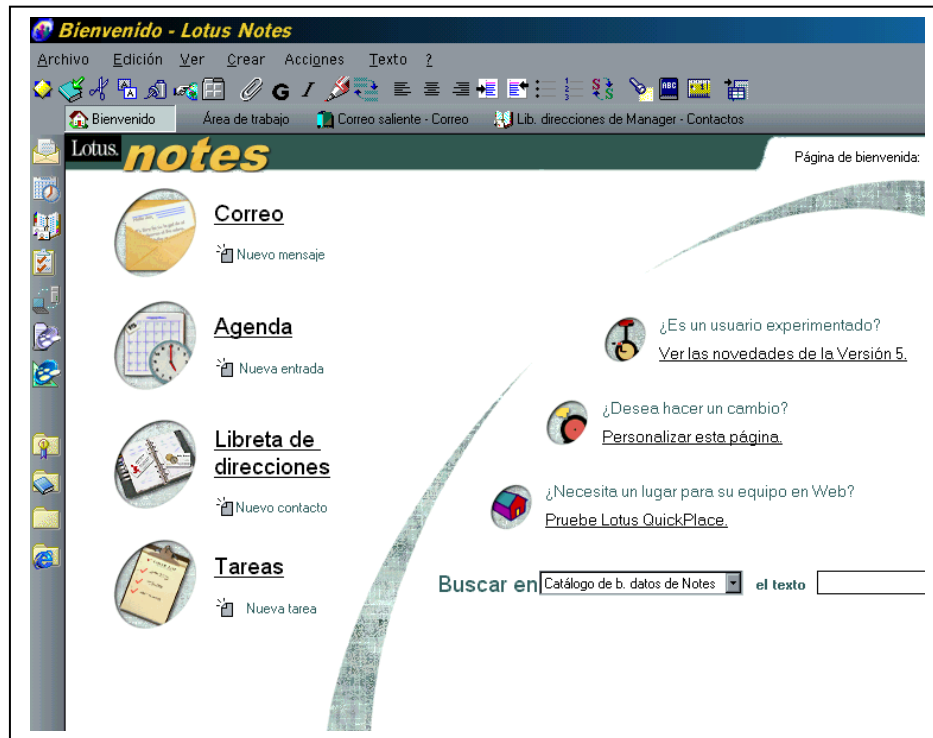


Figura 6.17. Pantalla Lotus Notes.

Nueva barra de Smarticons

Existe una nueva barra en el lado izquierdo de Lotus Notes.



Figura 6.18. Barra Smarticons.



Pulsándolo accedemos a nuestro buzón de correo



Accedemos a nuestra agenda



Tenemos acceso al listin de personal



Vamos a tareas pendientes



Nos muestra la página del replicador



Abrimos Lotus Domino Administrator (Sólo si lo tenemos instalado)



Abrimos Lotus Designer (Si lo tenemos instalado)



Nos muestras las bases de datos que tenemos como Favoritas



Nos aparecen todas las bases de datos que tenemos en nuestro Area de Trabajo



Más Marcadores (Direcciones de Internet de utilidad)



Vínculos del Internet Explorer

La tarea de como crear nuevos usuarios:

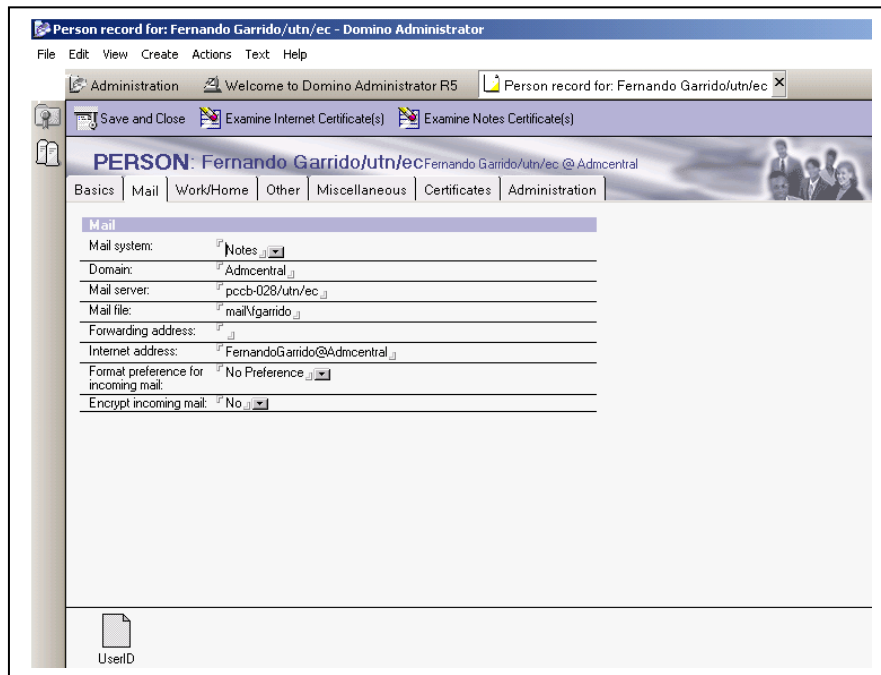


Figura 6.19. Creando Nuevo Usuario.

Migrando usuarios de NT.

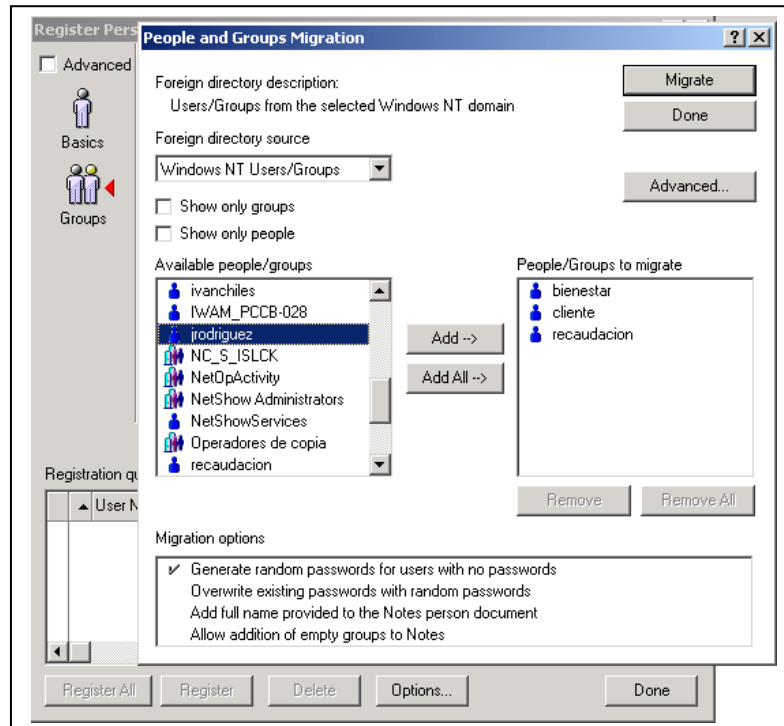


Figura 6.20. Migrando usuarios de NT.

6.2.3. CODIFICACION LOTUS NOTES VIA Lotus-Script

Podemos programar e interactuar con bases de datos relacionales mediante programación LotusScript, que es un lenguaje algo similar a Visual Basic, aquí un ejemplo de un formulario que trabaja en Lotus Notes y puede enlazarse a SQL- Server 7.0, mediante ODBC.

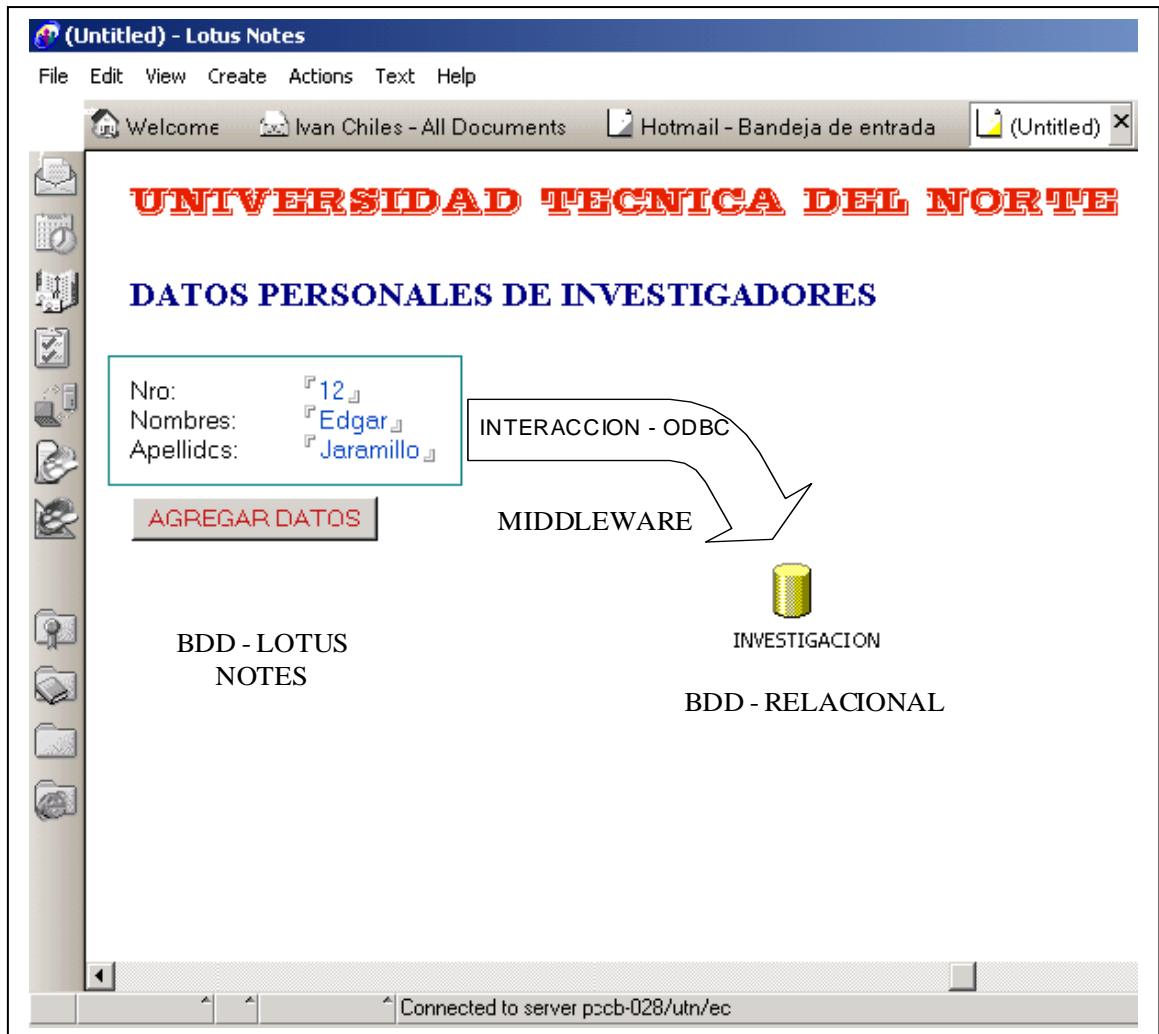


Figura 6.21. Interacción Lotus Notes – SQL-Server 7.0

Aquí un poco de código Lotus Script, para inserción de datos desde Lotus a SQL-Server vía ODBC.

```
*****  
'EN OPTIONS DEL FORMULARIO  
*****  
Usesx "*LSXODBC"  
*****
```

'AL INICIO DEL FORMULARIO

```
Sub Initialize
  Dim con As New ODBCConnection
  Dim dsn As String
  Dim msg As String
  Dim dataSource As String
  Dim userName As String
  Dim password As String

  userName="admin"
  password="cuicyt"
  dsn = Inputbox("Cuicyt - Financiero", "Ingrese nombre ODBC")
REM   If Not con.ConnectTo(dsn) Then
  con.ConnectTo(dsn)
  If Not con.IsConnected Then
    Messagebox "Could not connect to " & dsn,, "Error"
    Exit Sub
  End If
  tables = con.ListTables(dsn)
  msg = dsn & " contains the following tables:" & Chr(10)
  For n% = Lbound(tables) To Ubound(tables)
    msg = msg & Chr(10) & tables(n%)
  Next
  Messagebox msg,, "Tables for " & dsn
  con.Disconnect

End Sub
```

'EN FORMULARIO BASE CUICYT FUNCIONANDO CORRECTAMENTE

'Acceso a campos de Lotus Notes y manipulación

```
Sub Click(Source As Button)
  Dim firstName As String
  Dim lastName As String
  Dim fullName As String

  Dim workspace As New NotesUIWorkspace
  Dim uidoc As NotesUIDocument

  Set uidoc = workspace.CurrentDocument
  firstName = uidoc.FieldGetText("firstName")
  lastName = uidoc.FieldGetText("lastName")
  fullName = firstName & " " & lastName
  MsgBox fullName
  Call uidoc.FieldSetText("Nombres", fullName)

End Sub
```

End Sub

'Código de acceso e inserción de datos desde Lotus Notes hasta SQL Server

```
Sub Click(Source As Button)
  Dim con As New ODBCConnection
  Dim qry As New ODBCQuery
  Dim result As New ODBCResultSet
  Dim APELLIDO1 As String
```

```
Dim lastName As String
Dim msg As String
Dim NOMBRE1 As String
On Error Goto errorHandler

Set qry.Connection = con
Set result.Query = qry

NOMBRE1="Edgar"
APELLIDO1="Jaramillo"
con.ConnectTo "prueba sql","sa","sa"
qry.SQL = "INSERT INTO Alumno ( Idalumno, Apellidos, Nombres ) SELECT 12,'" &
APELLIDO1 & "' , '" & NOMBRE1 & "'"

result.Execute

result.Close(DB_CLOSE)
con.Disconnect
Exit Sub
errorHandler:
Messagebox result.GetExtendedErrorMessage,, result.GetErrorMessage
Exit Sub

End Sub
```

El entorno de desarrollo es el que se muestra en la figura siguiente:

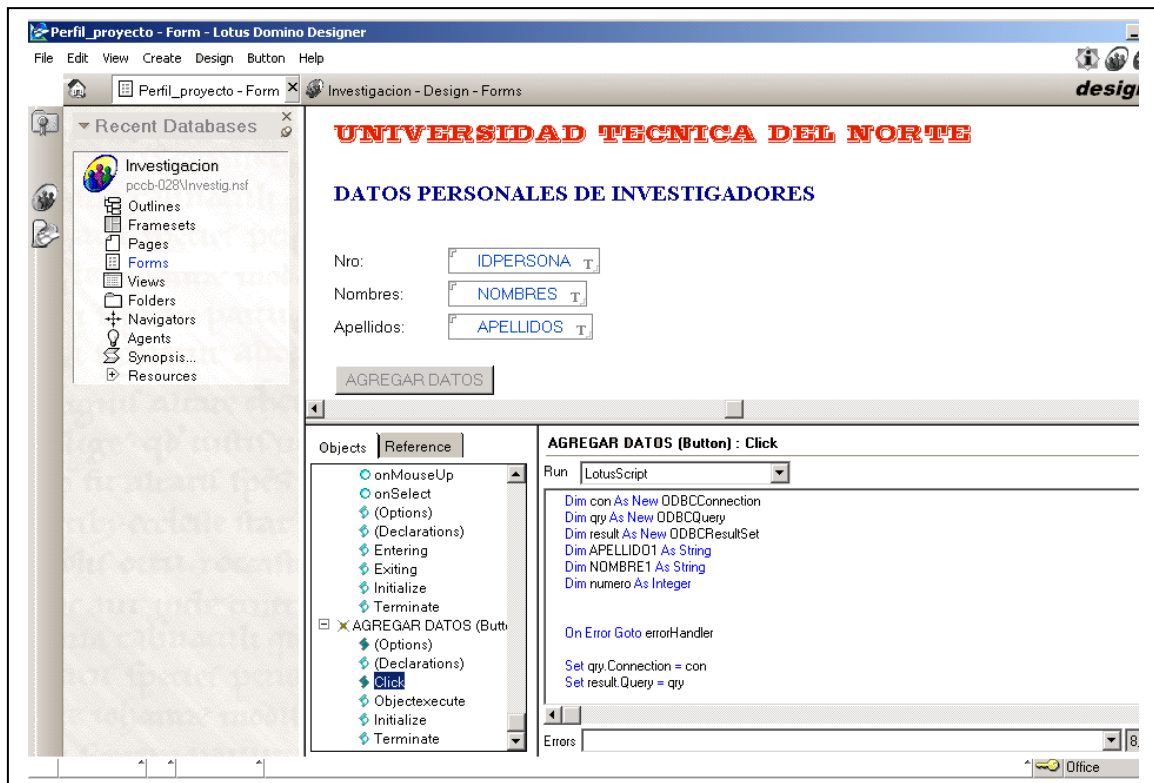


Figura 6.22. Entorno de Programación y Diseño Lotus Notes.

Se tiene un entorno de diseño donde muy claramente se puede distinguir el área de diseño del formulario los objetos que se manejan, y sus métodos o funciones.

Toda la gestión Académica es desarrollada bajo este esquema de formularios, y se provee de un entorno de reuniones, correo, citas y más funcionalidades propias de Lotus Notes para trabajo en grupo.

6.3. LISTA DE CONTROL DE ACCESO (LAC)

6.3.1. DEFINICION DE USUARIOS Y PERFILES DE USUARIO

Para definir los perfiles de usuarios, se ha utilizado la siguiente nomenclatura:

PERFIL: **C** = Coordinador Proyectos, **F** = Supervisor Financiero, **A**= Administrador, **I** = Coordinador de Investigación. **S** = Secretaria. **P** = Jefe de Personal. **U** = Autoridad

PERMISOS: **S**elect=Consulta, **U**ppdate=Modificación, **I**nsert= Inserción, **D**elete=Borrar

A continuación describimos la matriz de utilización de recursos, en grupos de tablas los permisos que los usuarios tienen para trabajar con los datos de ellas.

GRUPO 1:

dbo_escuela, dbo_facultad, dbp_dedicacion, dbp_designacion, dbp_estado_civil, dbp_estado_laboral, dbp_nivel_educacion, dbp_puesto, dbp_relacion_funcional, dbp_tipo_personal, dbp_tipo_relacion_laboral, empresa, evento, institucion, mes, niveles_de_investigacion, periodo, proveedores, tipo_bien, tipo_institucion, tipo_partida, tipo_us, titulo, unidad, usuarios.

USUARIO	PERFIL	S	U	I	D
Director CUICYT	U	X			
Director Financiero	F	X			
Jefe de Personal	P	X			
Vicerrector Administrativo	U	X			
Director de Planeamiento	U	X			
Supervisor 1	R	X			
Coordinador Proyecto 1	C	X			
Coordinador Investigacion1	I	X			
Secretaria 1	S	X			
Administrador GEDI-UTN	A	X	X	X	X

GRUPO 2:

autorizacion_gastos, autorizacion_gastos_detalle, cierre_periodo, cierre_periodo_partida, concepto_partida, dbf_pago, dbf_presupuesto, dbf_presupuesto_detalle, dbf_rubro, dbo_banco, dbo_cuenta_banco, items_gasto, meses_por_partida, partida_año, artida_mes, partida_presupuestada, presupuesto_general_por_partida, presupuesto_general_por_proyectos, presupuesto_mes_partida, transferencia_interna.

USUARIO	PERFIL	S	U	I	D
Director CUICYT	U	X			
Director Financiero	F	X	X	X	X
Jefe de Personal	P	X			
Vicerrector Administrativo	U	X			
Director de Planeamiento	U	X			
Supervisor 1	R	X			
Coordinador Proyecto 1	C	X			
Coordinador Investigacion1	I	X			
Secretaria 1	S	X			
Administrador GEDI-UTN	A	X			

GRUPO 3:

bienes, equipo_de_trabajo, informe_ejecucion, informe_final, proyecto,

USUARIO	PERFIL	S	U	I	D
Director CUICYT	U	X			
Director Financiero	F	X			
Jefe de Personal	P	X			
Vicerrector Administrativo	U	X			
Director de Planeamiento	U	X			
Supervisor 1	R	X			
Coordinador Proyecto 1	C	X	X	X	X
Coordinador Investigacion1	I	X			
Secretaria 1	S	X			
Administrador GEDI-UTN	A	X			

GRUPO 4:

act_por_realizar, cursos, estado_prov, experiencia, formacion, funcion, horario, persona , Trabajo.

USUARIO	PERFIL	S	U	I	D
Director CUICYT	U	X			
Director Financiero	F	X			
Jefe de Personal	P	X	X	X	X
Vicerrector Administrativo	U	X			
Director de Planeamiento	U	X			
Supervisor 1	R	X			
Coordinador Proyecto 1	C	X			

Coordinador Investigacion1	I	X			
Secretaria 1	S	X			
Administrador GEDI-UTN	A	X			

GRUPO 5:

areas_del_conocimiento, especializacion, estado_proyecto, marcas, pais

USUARIO	PERFIL	S	U	I	D
Director CUICYT	U	X			
Director Financiero	F	X			
Jefe de Personal	P	X			
Vicerrector Administrativo	U	X			
Director de Planeamiento	U	X			
Supervisor 1	R	X	X	X	X
Coordinador Proyecto 1	C	X	X	X	X
Coordinador Investigacion1	I	X	X	X	X
Secretaria 1	S	X	X	X	X
Administrador GEDI-UTN	A	X	X	X	X

6.3.2. CREACION DE ROLES

Luego de haber definido los perfiles de usuario, procedemos a crear las Listas de Control de Acceso **LACs**, siguiendo los siguientes pasos:

1. Crear Usuarios de la Base de Datos
2. Crear Roles
3. Asignar Roles por Usuario

```
REM *****
REM CREACION DE USUARIOS Y PRIVILEGIOS A LA BASE DE DATOS GEDI-UTN
REM AUTOR:
REM      Iván Chiles
REM DIRECTOR:
REM      Fernando Garrido
REM *****
```

```
REM Ingresar como superusuario
connect system;
show user;
```

REM CREACION DE USUARIOS

REM *****

REM Usuario: Iván Chiles

REM Perfil: Administrador-SARE

create user ivan identified by ivanc;

grant connect, resource to ivan;

REM Usuario: Arturo Flores

REM Perfil: Supervisor Financiero

create user arturof identified by arturof;

grant connect, resource to arturof;

REM Usuario: Lenin Escobar

REM Perfil: Coodinador Proyectos

create user lenine identified by lenine;

grant connect, resource to lenine;

REM Usuario: Carlos Maldonado

REM Perfil: Jefe de Personal

create user carlosm identified by carlosm;

grant connect, resource to carlosm;

REM Usuario: Edgar Jaramillo

REM Perfil: Coordinador Investigación

create user edgarj identified by edgarj;

grant connect, resource to edgarj;

REM Usuario: Bélgica Bermeo

REM Perfil: Director CUICYT

create user belgicab identified by belgicab;

grant connect, resource to belgicab;

REM Usuario: Mirian Viteri

REM Perfil: Secretaria

create user mirianv identified by mirianv;

grant connect, resource to mirianv;

REM PRIVILEGIOS DE SUPERSUSUARIO AL Administrador-GEDI

REM *****

grant all privileges to ivan;

REM Ingresar como Administrador-SARE

connect ivan/ivanc;

show user;

REM CREACION DE ROLES

REM *****

create role ADMINISTRADOR;

create role SUPERVISOR;

create role COORDINADORP;

create role COORDINADORI;

create role AUTORIDAD;

create role SECRETARIA;

create role JEFEPERSONAL;

REM CREACION DE PRIVILEGIOS PARA EL ROL ADMINISTRADOR

REM *****

REM GRUPO 1

grant select,insert,update on dbo_facultad to ADMINISTRADOR;

grant select,insert,update on dbo_escuela to ADMINISTRADOR;

grant select,insert,update on dbp_dedicacion to ADMINISTRADOR;

grant select,insert,update on dbp_designacion to ADMINISTRADOR;

grant select,insert,update on dbp_estado_civil to ADMINISTRADOR;

grant select,insert,update on dbp_estado_laboral to ADMINISTRADOR;

grant select,insert,update on dbp_nivel_educacion to ADMINISTRADOR;

grant select,insert,update on dbp_puesto to ADMINISTRADOR;

grant select,insert,update on dbp_relacion_funcional to ADMINISTRADOR;

grant select,insert,update on dbp_tipo_personal to ADMINISTRADOR;

grant select,insert,update on dbp_tipo_relacion_laboral to ADMINISTRADOR;

grant select,insert,update on empresa to ADMINISTRADOR;

grant select,insert,update on evento to ADMINISTRADOR;

grant select,insert,update on institucion to ADMINISTRADOR;

grant select,insert,update on mes to ADMINISTRADOR;

```
grant select,insert,update on niveles_de_investigacion to ADMINISTRADOR;  
grant select,insert,update on periodo to ADMINISTRADOR;  
grant select,insert,update on proveedores to ADMINISTRADOR;  
grant select,insert,update on tipo_bien to ADMINISTRADOR;  
grant select,insert,update on tipo_institucion to ADMINISTRADOR;  
grant select,insert,update on tipo_partida to ADMINISTRADOR;  
grant select,insert,update on tipo_us to ADMINISTRADOR;  
grant select,insert,update on titulo to ADMINISTRADOR;  
grant select,insert,update on unidad to ADMINISTRADOR;  
grant select,insert,update on estado_proyecto to ADMINISTRADOR;  
grant select,insert,update on usuarios to ADMINISTRADOR;
```

REM GRUPO 2

```
grant select on autorizacion_gastos to ADMINISTRADOR;  
grant select on autorizacion_gastos_detalle to ADMINISTRADOR;  
grant select on cierre_periodo to ADMINISTRADOR;  
grant select on cierre_periodo_partida to ADMINISTRADOR;  
grant select on concepto_partida to ADMINISTRADOR;  
grant select on dbf_pago to ADMINISTRADOR;  
grant select on dbf_presupuesto to ADMINISTRADOR;  
grant select on dbf_presupuesto_detalle to ADMINISTRADOR;  
grant select on dbf_rubro to ADMINISTRADOR;  
grant select on dbo_banco to ADMINISTRADOR;  
grant select on dbo_cuenta_banco to ADMINISTRADOR;  
grant select on items_gasto to ADMINISTRADOR;  
grant select on meses_por_partida to ADMINISTRADOR;  
grant select on partida_año to ADMINISTRADOR;  
grant select on artida_mes to ADMINISTRADOR;  
grant select on partida_presupuestada to ADMINISTRADOR;  
grant select on presupuesto_general_por_partida to ADMINISTRADOR;  
grant select on presupuesto_general_por_proyectos to ADMINISTRADOR;  
grant select on presupuesto_mes_partida to ADMINISTRADOR;  
grant select on transferencia_interna to ADMINISTRADOR;
```

REM GRUPO 3

```
grant select on bienes to ADMINISTRADOR;  
grant select on equipo_de_trabajo to ADMINISTRADOR;
```

```
grant select on informe_ejecucion to ADMINISTRADOR;  
grant select on informe_final to ADMINISTRADOR;  
grant select on proyecto to ADMINISTRADOR;
```

REM GRUPO 4

```
grant select on act_por_realizar to ADMINISTRADOR;  
grant select on cursos, estado_prov to ADMINISTRADOR;  
grant select on experiencia to ADMINISTRADOR;  
grant select on formacion to ADMINISTRADOR;  
grant select on funcion to ADMINISTRADOR;  
grant select on horario to ADMINISTRADOR;  
grant select on persona to ADMINISTRADOR;  
grant select on trabajo to ADMINISTRADOR;
```

REM GRUPO 5

```
grant all on areas_del_conocimiento to ADMINISTRADOR;  
grant all on especializacion estado_prov to ADMINISTRADOR;  
grant all on marcas to ADMINISTRADOR;  
grant all on pais to ADMINISTRADOR;
```

REM ASIGNAR ROLES A USUARIOS

```
REM *****
```

```
grant ADMINISTRADOR to ivan;  
grant SUPERVISOR to arturof;  
grant COORDINADORP to lenine;  
grant COORDINADORI to edgarj;  
grant AUTORIDAD to belgicab;  
grant SECRETARIA to mirianv;  
grant JEFEPERSONAL to carlosm;
```

Los roles son como una plantilla de usuario que se puede aplicar a grupos de usuarios, con la finalidad de no definir privilegios por cada usuario sino por ROL, y estos roles a su vez poderlos asignar a un nuevo usuario.

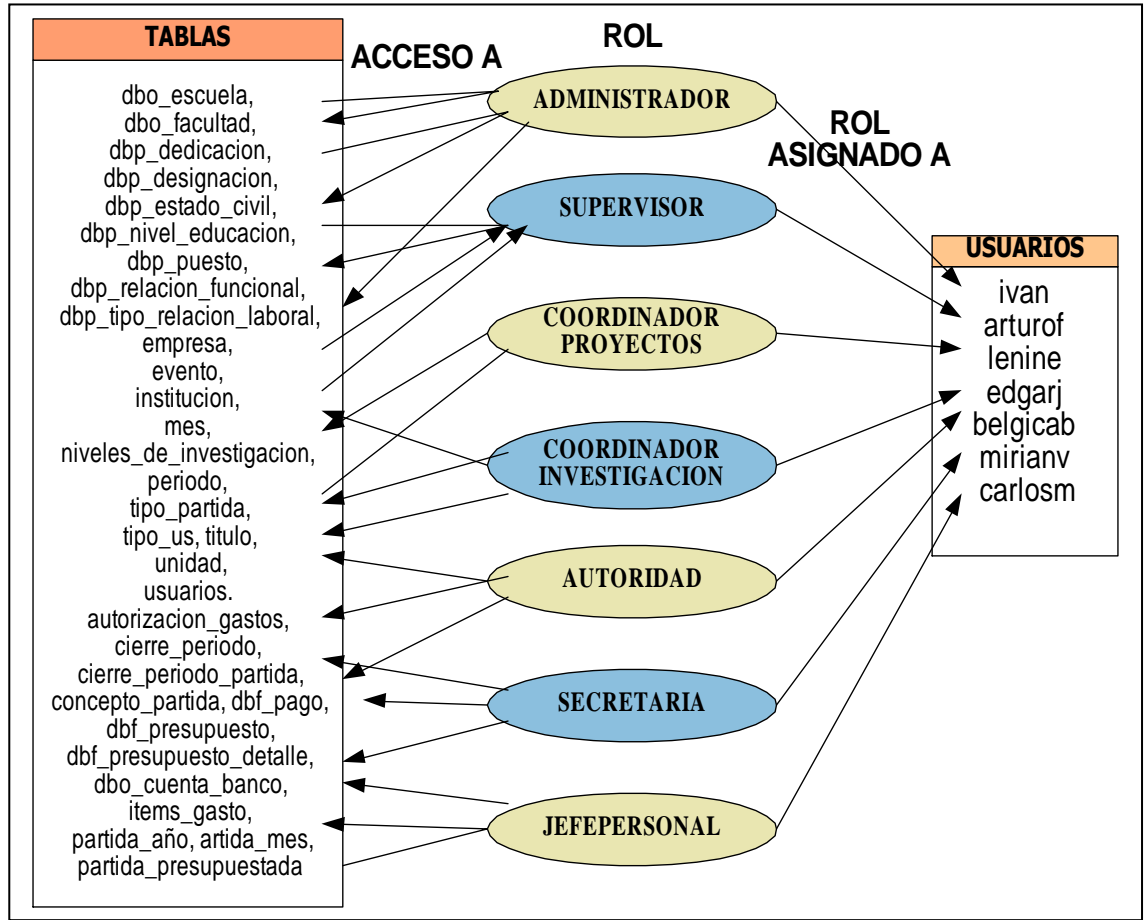


Figura 6.23. Control de Acceso y Asignación de Roles.

6.4. ESQUEMAS LENGUAJE DE MODELAMIENTO UNIFICADO

DEFINICION DE ACTORES GEDI – UTN

Figura 6.24. Actores y Personal UTN.

CASO DE USO GESTION INVESTIGACION

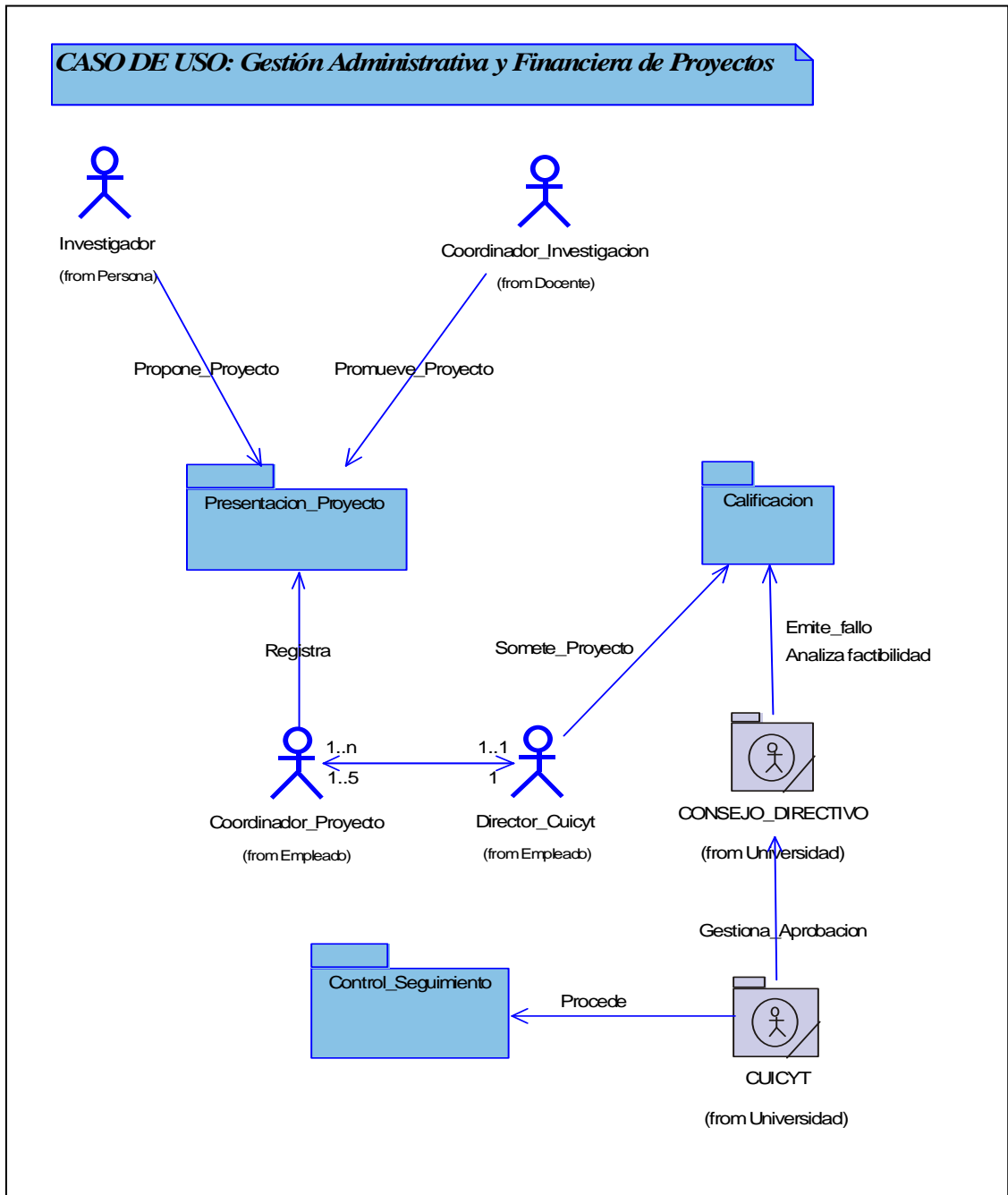


Figura 6.25. Caso de Uso Gestión Investigación.

ESQUEMA PERSONAL

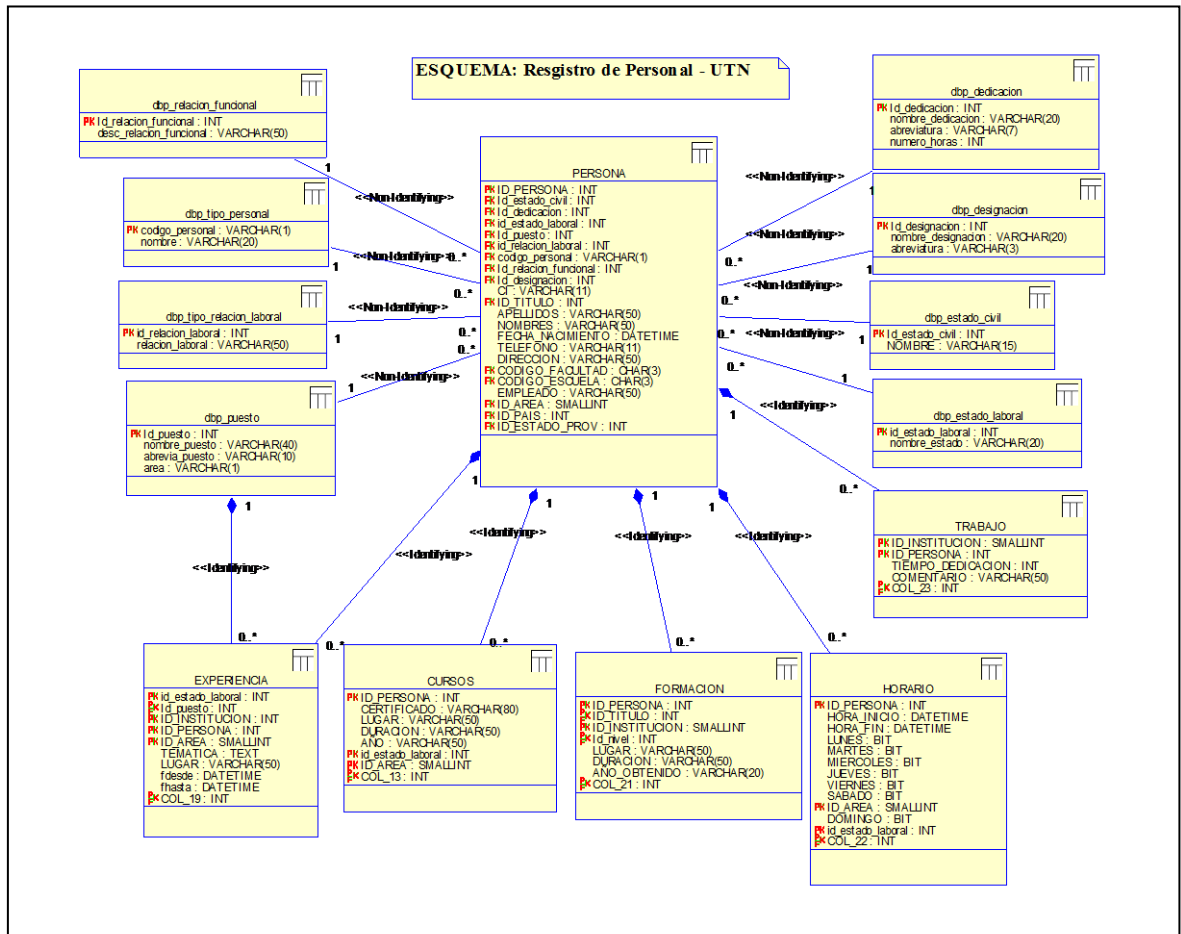


Figura 6.26. Esquema Personal.

ESQUEMA AUTORIZACION DE GASTO

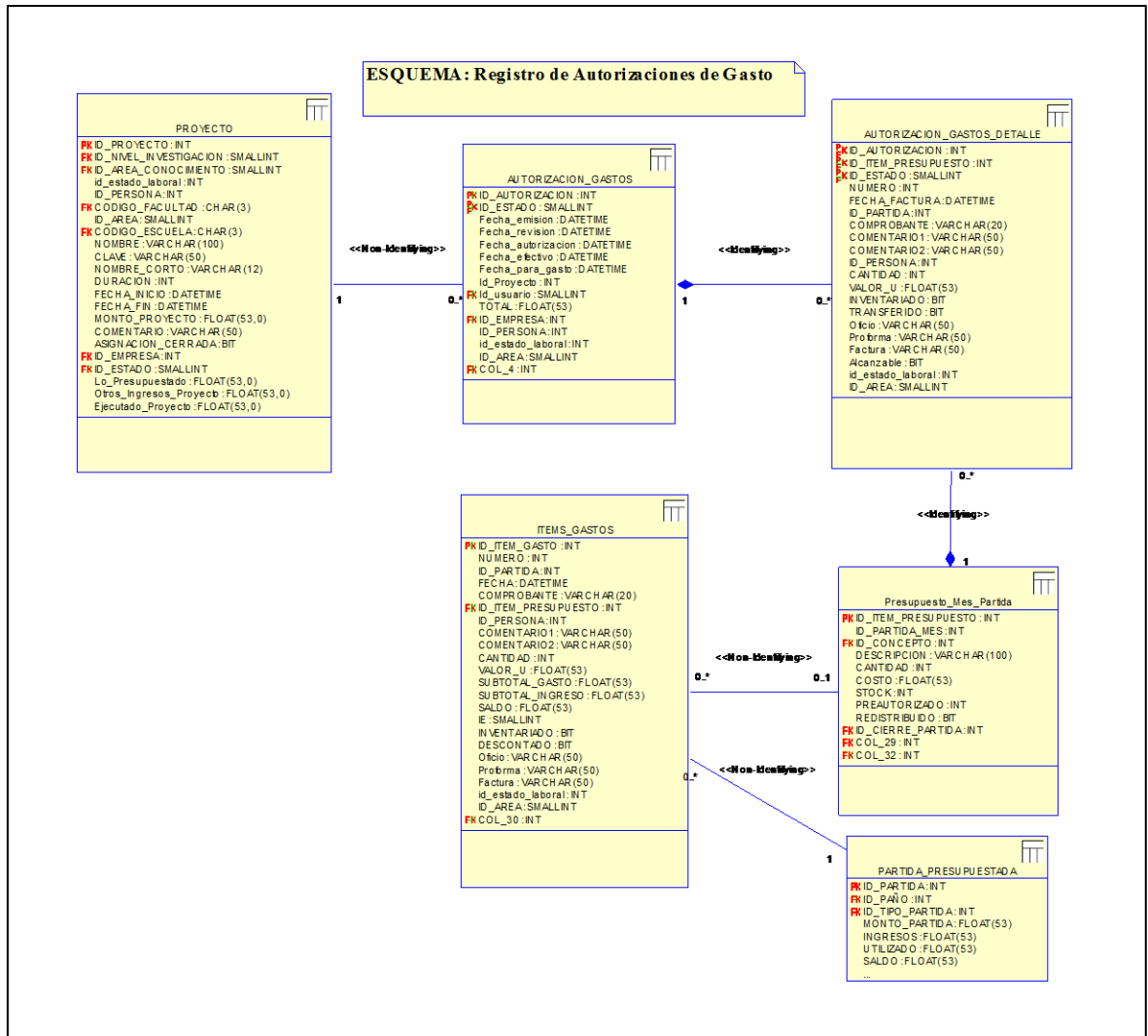


Figura 6.27. Esquema Autorización de Gasto.

En el CD, se pone a disposición el archivo de modelamiento UML para el aplicativo de gestión de Investigación llamado "modelo_cuicyt.mdl", el mismo que se debe abrir con el paquete de modelamiento Rational Rose, y contiene los diseños de:

- Diagramas de Casos de Uso
- Especificación de Casos de Uso
- Diagramas de Interacción (Escenarios)
- Diagramas de Actividad (Flujos de Trabajo)
- Diagramas de Estados (Dinámica)

CAPITULO 7

Conclusiones, Recomendaciones y Verificación de la Hipótesis

7.1. CONCLUSIONES

- Lotus Notes ha desarrollado y aprovechado las expectativas que desde los años 70 hasta el año de 1991 era una mitología, al intentar crear un servidor de grupos de trabajo en tiempo real – sincrónico y en tiempo asincrónico o tiempo compartido, lo que se vio en el capítulo V, con los modelos conversacionales, IBIS propuesto por Kunz y Rittel en 1970 y el modelo SISCO donde se hace referencia a la Memoria Organizacional (MO) que no sólo permite acumular y preservar el conocimiento, sino que también permite compartirlo implicando con esto una mayor comunicación y aprendizaje en la organización, además de poseer control sobre grupos de trabajo. Con esta metodología y herramientas de software que soportan a los modelos conversacionales junto con la Memoria Organizacional, se logra reducción de tiempo de comunicación, disposición en línea de acuerdos y memorias organizacionales anteriores, con el propósito de mejorar la gestión administrativa, por el trabajo grupal que se logra con los actores protagónicos en una organización.
- El gran cambio en el manejo de objetos, se ve repercutido en lo que se debe hacer para cumplir con las garantías de calidad del software y en sus especificaciones. En el capítulo IV, se hace referencia a las métricas de garantía de calidad de software, y se llegó a determinar que mientras mayor sea el árbol de herencia mayor será el grado de dificultad de mantener los métodos que son comunes a las subclases. Esto lleva a concluir que la construcción de software basado en objetos resulta mantenible y menos compleja, trabajando con métodos basados en programación corta (funcional y modular), siempre y cuando se cuente con la debida documentación.
- La Tecnología Informática, poco podría hacer frente a los procesos burocráticos de las instituciones, se necesita para mejorar compromiso y organización de procesos de sus actores mediante innovación y reingeniería de procesos para que la TI tenga un potencial efecto en la gestión de la áreas estratégicas.
- Son muchas las bondades de Lotus Notes, pero sobre todo el permitir el acceso a bases de datos relacionales por medio de Middleware (ODBC) y su programación LotusScript, la hace aún más funcional al momento de necesitar recuperar información almacenada en un DBMS, lo que evita el tener que reingresar información para alimentar nuevos procesos.
- El software orientado al flujo de trabajo, software para grupos, son características de las bases de datos documentales, existe software orientado al manejo de información estructurada cuyo procesamiento es transaccional con esto nos referimos a motores de bases de datos

relacionales, todos este software tiene su propósito específico; en cambio existe software subutilizado, económico y poco conocido como Visual Fox, Access, que puede ser utilizado y orientado al manejo de procesos transaccionales con lo que se convierten en **herramientas transparentes** al momento de interactuar como frontales con Motores de Bases de Datos de diferentes proveedores y las utilizan y trabajan a estas bases como suyas a través de ODBC.

- El modelamiento orientado a objetos y el modelo conceptual tienen una gran similitud en consideración de que se modela la funcionalidad y no los detalles, que en objetos son parte de la implementación.

7.2. RECOMENDACIONES

- Este proyecto pretende ser el inicio de lo que puede ser la Universidad, una Institución electrónicamente coordinada y comunicada, aprovechando las bondades de Lotus Notes para el trabajo en grupo.
- A todos aquellos estudiantes de Sistemas, que deseen incursionar en los campos del trabajo en grupo, exploten las potencialidades de Lotus Notes, ya que esta herramienta cuenta con un sinnúmero de potencialidades para cumplir grandes expectativas empresariales.
- Ha sido palpable para mí y para quienes hemos laborado en el Departamento de Informática de la Universidad Técnica del Norte, de los constantes cambios que se presentan en las fases de desarrollo de una aplicación informática, por esto es necesario que exista una base de parámetros de configuración en cada aplicativo informático que se desarrolle, como una solución a estos cambios. Por lo expuesto, los diseñadores de Modelos de Base de Datos incorporen esos cambios, con la finalidad de que no dependa en mayor grado de la programación, como sustenta la teoría de la Ingeniería de Software, pero sobre todo una estandarización de procesos debidamente legalizada y formalizada.
- Es imprescindible que en la Universidad Técnica del Norte existan políticas y planes de desarrollo informático y tecnológico a mediano y largo plazo; esto permitiría fortalecer la gestión en los procesos administrativos, financieros, académicos y de investigación todos integrados y a disposición de los actores del entorno universitario.
- Al Centro Universitario de Investigación Científica y Tecnológica - CUICYT, órgano beneficiado de este proyecto, apoye los proyectos de desarrollo reales como éste, tal como lo ha hecho el Departamento de Informática – UTN, en estos dos años, y que forma parte de un conjunto de aplicaciones que se integrará a otras que contemplan procesos académicos, administrativos y financieros.
- Al CUICYT la estandarización formal de formatos; la difusión del mecanismo de los procesos para la consecución, desarrollo y finalización de proyectos de investigación, con el propósito de eliminar o reducir los trámites burocráticos, que permitan al investigador preocuparse menos de los contratiempos por falta de recursos.

7.3. VERIFICACION DE LA HIPOTESIS

El haber seguido las fases de modelamiento y desarrollo Orientado a Objetos ha permitido culminar satisfactoriamente este proyecto informático, el cual viene a contribuir con el desarrollo tecnológico del Centro Universitario de Investigación Científica y Tecnológica CUICYT, facilitando estandarizar procesos financieros, con procesos de gestión investigativa, y sobre todo incorporar nuevas herramientas informáticas de gestión que permiten al usuario final una interfaz amigable de trabajo y que ayuda a los órganos administrativos a una adecuada toma de decisiones en base a información racionalmente procesada. Tecnológicamente se ha logrado demostrar la conexión que se puede lograr entre bases de datos relacionales y bases de datos documentales, siendo esta última una alternativa para el flujo de documentos entre las principales áreas administrativas de la Universidad Técnica del Norte.

GLOSARIO DE TERMINOS

TERMINO	DESCRIPCION	PAGINA
BLOB	Binary Large Object - Objeto Largo Binario.	62
CAD	Computer aided design - Diseño asistido por computador.	67
CASE	Computer – aided software engineering - Ingeniería De Software Asistida Por Computador.	67
CLI	Call-Level Interface - Interfaz del nivel de llamada.	62
CMIP	Common Management Information Protocol - Protocolo de Información de Dirección Común.	60
CSCW	Compute Supported Cooperative Work - Trabajo Cooperativo Asistido por Computador.	127, 128
DAG	Grafo con raíz acíclica dirigido.	72
DBMS	Database Management System – Sistema Manejador de Base de datos. Un DBMS es una colección de numerosas rutinas de software interrelacionadas, cada una de las cuales es responsable de una tarea específica.	51,53,54, 61,63,65
DRDA	Distributed Relational DataBase Architecture o Arquitectura distribuida de bases de datos de IBM.	61
GROUPWARE	El término Groupware es una contracción de las palabras Group y software.	49,51,61, 125,126
GUI	Interfaz Gráfica de Usuario. Pantallas y funciones que proporcionan al usuario final un medio gráfico para tener acceso al sistema de computación.	53
HTTP	HyperText Transfer Protocol, Protocolo de Transferencia de Hipertexto.	60
HTTPS	Hypertext Transfer Protocol Security - Protocolo de Transferencia de Hipertexto Seguro.	60

IBIS	Issue Based Information System - Sistema de Información Basado en el Problema.	129, 130
LAN	Local Area Network - Red de Area Local.	49, 61
MAPI	Messaging Application Program Interface – Interface Programa de Aplicación de mensajes.	144
MIDDLEWARE	Batido de tecnologías de software de comunicación.	61,62,139 , 169
MPTN	MultiProtocol Transport Network – Multiprotocolo de Transporte en la Red.	62
ODBC	Open Database Conectivity – Conexión abierta a bases de datos.	60,61,64, 138,142, 162
OIS	Office information systems - Sistemas de Información de Oficina	68
OLTP	Online transaction processing – Procesamiento de transacciones en línea.	60,139
ORB	Object Request Broquer - corredor de solicitudes de objetos.	52,61
RPC	Remote Procedure Call - Llamada a Procedimiento Remoto.	144
SGBD	Sistema de Gestión de Base de Datos	55,57,58, 59,60
SNMP	Simple Network Managenement Protocol, Protocolo Simple de Administración de Redes.	60
SQL	Structured Query Language – Lenguaje de Consulta Estructurado.	50,51
SSL	Secure Socket Layer – Capa de enlace segura	60
TP	Procesamiento de Transacciones	43,47
VIM	Protocolo correo independiente del proveedor	144
WfMC	La Workflow Management Coalition	128

BIBLIOGRAFIA

LIBROS

- Orfali - Harkey - Edwards; "CLIENTE SERVIDOR - Guía de Supervivencia"; 2da Edición. Mc.Graw – Hill, año 1999, México.
- GOMEZ Guillermo; "Planeación y Organización de Empresas"; Mc.Graw – Hill. 1994, México.
- SAPAG CHAIN Nassir, SAPAG CHAIN Reinaldo; "Preparación y Elaboración de Proyectos"; Mc.Graw Hill. 1989, Colombia.
- DRA. BAENA PAZ Guillermina; "Instrumentos de Investigación" Editores Mexicanos Unidos. 1995, México.
- JAMES MARTIN, JAMES, ODELL; "Análisis y Diseño Orientado a Objetos"; Prentice-Hall, 1992, México
- GRADY BOOCH; "Análisis y Diseño Orientado a Objetos con Aplicaciones"; 2da Edición; Prentice-Hall, 1992, México.
- Gordon B. Davis, Margarethe H. Olson; "SISTEMAS DE INFORMACIÓN GERENCIAL"; Mc.Graw Hill. 1991, México.
- (LOR94) Lorenz, Mark y Jeff Kidd, "Object-Oriented Software Metrics", Prentice-Hall, 1994.
- (PRE98) Pressman, Roger, "Ingeniería de Software, Un enfoque Práctico", cuarta edición, McGraw-Hill, 1998.
- [Ellis91] Ellis, C.A., Gibbs, S J., Rein, G.L.: "Groupware some issues y experiences", Comm. of the ACM, Vol. 34 No. 1,1991.

REVISTAS Y FOLLETOS

- Microsoft Visual Interdev (Web solution KIT); CD de referencia.
- Desarrollo de Aplicaciones 1; Folleto Lotus Domino 4.5 y Notes 4.5
- Lotus Development Corporation; Folleto IBM del Ecuador.

DIRECCIONES INTERNET

- LOTUS NOTES <http://Web6.notes.net>
- WORKFLOW <http://www.workflowsoftware.com>
- POWERFLOW <http://www.workflow.co.nz/applications>
- LOTUS <http://www.lotus.com/bpg>

ARTICULOS

- (BER95) Berard, Edward V., "Metrics for Object-Oriented Software Engineering", encontrado en www.cetus-links.org/oo-metrics.html, y www.toa.com/pub/moose.htm 30 de octubre del 2001.
- (FER95) Fernández, Luis, "Calidad de Software", Madrid 1995, encontrado en <http://www.ati.es/gt/calidad-software/presentacion.htm>, 9 de noviembre del 2001.
- (ZEN01) Zenteno, Julia, "Métricas para el Diseño Orientado a Objetos", UMSA – La Paz – Bolivia, encontrado en www.jupiter.umsanet.edu.bo/postgrado/informatica/post/pjz21.htm, 17 de octubre del 2001.
- (UML01) Departamento Sistemas Informáticos y Computación, Universidad Politécnica de Valencia (España), "Notación UML", encontrado en www.dsic.upv.es/~uml 18 octubre del 2001.
- INFORMATICA DISTRIBUIDA; encontrado en:
<http://www.sapmania.com/comunicaciones/com.htm>,
<http://www.map.es/csi/caibi/ibst/estandar/8/ibdcom.htm>,
http://www.windowstimag.com/atrasados/2000/41_abr00/articulos/afondo2.htm,
15 de Julio 2002.

ANEXOS