



**UNIVERSIDAD AUTÓNOMA DEL
ESTADO DE HIDALGO**

INSTITUTO DE CIENCIAS BASICAS E INGENIERIA

**CENTRO DE INVESTIGACIÓN EN TECNOLOGÍAS DE
INFORMACIÓN Y SISTEMAS**

**“DESARROLLO DE UN SISTEMA DE INFORMACIÓN
ORIENTADO A LA WEB. CASO DE ESTUDIO ÁREA
ACADÉMICA DE CIENCIAS COMPUTACIONALES”**

T E S I S

**EN OPCIÓN AL GRADO DE LICENCIADO EN SISTEMAS
COMPUTACIONALES**

AUTOR: MAGÍN VILLEGAS LARA

DIRECTOR DE TESIS: DR. OMAR LÓPEZ ORTEGA

PACHUCA DE SOTO, HIDALGO., MARZO DE 2006

ÍNDICE

INTRODUCCIÓN	1
<i>Planteamiento del problema</i>	1
<i>Justificación</i>	1
<i>Objetivo General</i>	3
<i>Objetivos Específicos</i>	3
<i>Descripción del Método</i>	3
<i>Alcances y Limitaciones del Proyecto</i>	4
<i>Organización de la Tesis</i>	4
CAPÍTULO I. Marco Teórico	
1.1 Impacto de las tecnologías de información (TI) en las organizaciones.	5
1.2 <i>La Internet, los procesos internos y las TI's.</i>	7
1.3 <i>Modelos de información mediante el enfoque IDEF</i>	
1.3.1 <i>IDEF0</i>	8
1.3.2 <i>IDEF1x.</i>	8
1.3.2.1 <i>Entidades</i>	9
1.3.2.2 <i>Atributos.</i>	9
1.3.2.3 <i>Claves primarias y alternas</i>	10
1.3.2.4 <i>Claves foráneas (Claves externas).</i>	12
1.3.2.5 <i>Relaciones de conexión.</i>	14
1.3.2.6 <i>Relaciones de categorización.</i>	16
1.3.2.7 <i>Relaciones no específicas.</i>	18
1.4 <i>Guía de las cinco formas normales de las bases de datos relacionales</i>	20
1.4.1 <i>Primera Forma Normal</i>	20
1.4.2 <i>Segunda Forma Normal</i>	20
1.4.3 <i>Tercera Forma Normal</i>	20
1.4.4 <i>Dependencias Funcionales</i>	20
1.4.5 <i>Cuarta y Quinta Formas Normales</i>	21
1.4.6 <i>Cuarta Forma Normal</i>	21
1.4.7 <i>Quinta Forma Normal</i>	22
1.4.8 <i>Redundancias Inevitables</i>	23

1.4.9 Redundancias entre Registros	23
------------------------------------	----

CAPÍTULO II. EL MODELO RELACIONAL DE SISTEMAS DE INFORMACIÓN

2.1 Tecnología de implantación	24
2.1.1 Modelo relacional	26
2.1.1.1 Restricciones de dominio.	26
2.1.1.2 Restricciones de clave.	26
2.1.1.3 Integridad de entidades, integridad referencial y claves externas.	27
2.1.2 Operaciones de actualización con relaciones	27
2.1.2.1 La operación insertar.	28
2.1.2.2 La operación eliminar.	28
2.1.2.3 La operación modificar.	29
2.2 Lenguaje SQL	29
2.2.1 Tipos de campos.	30
2.2.2 Operaciones más comunes en SQL.	30
2.2.2.1 Crear nueva tabla.	31
2.2.2.2 Añadir un nuevo registro	31
2.2.2.3 Borrar un registro.	32
2.2.2.4 Actualizar un registro.	32
2.2.2.5 Consultas SQL	33
2.3 El lenguaje SQL en el SGBD MySQL	36
2.3.1 Conexión a una Base se Datos	36
2.4 MAGUMA STUDIO	39
2.4.1 Características de Maguma	39

CAPÍTULO III. DESCRIPCIÓN DEL SISTEMA DE INFORMACIÓN

3.1 Construcción del modelo IDEF0	41
3.1.1 Diagramas y Componentes	41
3.1.2 Reglas de Sintaxis de los Diagramas	42
3.1.3 Operaciones en Cadena	43
3.2 Construcción del modelo IDEF1x.	46

3.2.1	<i>Etapa cero.</i>	46
3.2.2	<i>Etapa 1.</i>	48
3.2.3	<i>Etapa 2.</i>	50
3.2.3.1	<i>Identificación de relaciones.</i>	52
3.2.3.2	<i>Definición de relaciones.</i>	52
3.2.3.3	<i>Construcción del diagrama inicial.</i>	55
3.2.4	<i>Etapa 3</i>	57
3.2.4.1	<i>Resolución de relaciones no específicas.</i>	57
3.2.4.2	<i>Vistas de función</i>	62
3.2.4.3	<i>Identificación del atributo clave.</i>	63
3.2.4.4	<i>Migración de claves.</i>	63
3.2.4.5	<i>Validación de claves y relaciones.</i>	64
3.2.4.6	<i>Definición del atributo clave</i>	70
3.2.4.7	<i>Resultados de la etapa 3.</i>	70
3.2.5	<i>Etapa 4.</i>	76
3.2.5.1	<i>Identificación de atributos no clave.</i>	76
3.2.5.2	<i>Dueños de los atributos.</i>	76
3.2.5.3	<i>Refinación del modelo.</i>	77
3.2.5.4	<i>Resultados de la etapa 4</i>	77

CAPÍTULO IV. DESARROLLO DEL SISTEMA

4.1	<i>Requerimientos de los Usuarios</i>	82
4.2	<i>Cómo funciona el Sistema</i>	82
4.3	<i>Modelado del Funcionamiento del Sistema</i>	83
4.4	<i>Modelos Navegacionales del Sistema</i>	87
4.5	<i>Implantación del Sistema.</i>	91

Conclusiones Generales	108
Glosario de términos	110
Apéndice	112
Referencias	113

Lista de Figuras

Capítulo I	Página
1.1 Sintaxis de entidad	8
1.2 Atributos y sintaxis de claves primarias	11
1.3 Atributos y sintaxis de las claves	11
1.4 Sintaxis de claves foráneas	13
1.5 Sintaxis del nombre del rol	14
1.6 Sintaxis de relación de cardinalidad	14
1.7 Sintaxis de identificación de relación	15
1.8 Complemento (a), (b) contra incompleto (b) categorización	17
1.9 Sintaxis de relación no específica	19
1.10 Rompimiento de relaciones no específicas	19
Capítulo III	Página
3.1 Elementos básicos del ICOM (A-n)	42
3.2 Operación de cadena	44
3.3 Diagrama de Procesos de alto nivel	45
3.4 Diagrama de Procesos detallados	45
3.5 Construcción del diagrama inicial IDEF1x	56
3.6a Resolución de relaciones no específicas	58
3.6b Resolución de relaciones no específicas	58
3.6c Resolución de relaciones no específicas	58
3.6d Resolución de relaciones no específicas	58
3.7 Atributos clave	63
3.8 Migración de identificadores dependientes e identificadores independientes de entidades.	64
3.5a No-repetición de claves	65
3.9b No-repetición de claves	66

Capítulo III	Página
3.9c No-repetición de claves	66
3.9d No-repetición de claves	67
3.10 Modelo IDEF1x al final de la etapa 3	75
3.11 Modelo IDEF1x al final de la etapa 4.	80
3.12 Modelo IDEF1x al final de la etapa 4.	81

Capítulo IV	Página
4.1 Diagrama de Flujo-Agregar	85
4.2 Diagrama de Flujo-Consultar	86
4.3 Diagrama de Flujo-Eliminar	87
4.4 Vista General del Modelo Navegacional	88
4.5 Modelo Navegacional de Usuario no Registrado	89
4.6 Modelo Navegacional de Usuario Registrado	90
4.7 Página Principal	91
4.8 Selección de Ingreso	92
4.9 Ingreso de Administrador	93
4.10 Password Incorrecto	94
4.11 Permiso Administradores	95
4.12 Opciones del Administrador	96
4.13 Operaciones para Agregar	97
4.14 Datos a Agregar	98
4.15 Agregar LGAC	99
4.16 Registro de LGAC	100
4.17 Datos Existentes	101
4.18 LGAC Ingresada	102
4.19 LGAC Agregada	103
4.20 Datos a Consultar	104
4.21 Datos de Consulta	105
4.22 Datos a Eliminar	106
4.23 Datos Eliminados	107

Lista de Tablas

Capítulo II	Página
2.1 Ejemplo de la tabla.	26
2.2 Tipos de datos de los SGBD	30
2.3 Operadores matemáticos	35
2.4 Operadores lógicos	36
2.5 Comodines	36
Capítulo III	
3.1 Determinación de las entidades iniciales a ser modeladas.	48
3.2 Glosario de entidades	49
3.3 Matriz de identificación de relaciones	51
3.4 Definición de relaciones entre entidades	53
3.1.1 Actualización de la tabla 3.1	60
3.2.1 Actualización de la tabla 3.2	60
3.3.1 Actualización de la tabla 3.3	61
3.4.1 Actualización de la tabla 3.4	62
3.5 Matriz Entidad/Atributo	68
3.6 Documentación correspondiente a la etapa 3	72
3.7 Lista de relaciones de identificación para las cuales la entidad actúa como padre	73
3.8 Lista de entidades de identificación para las cuales le entidad actúa como hijo	74
3.9 Lista de entidades de no-identificación para las cuales la entidad actúa como padre	74
3.10 Lista de entidades de no-identificación para las cuales la entidad actúa como padre	74
3.11 Documentación correspondiente a la etapa 3	78

INTRODUCCIÓN

A continuación se presenta un panorama general, mediante la descripción del planteamiento del problema, del método empleado para la solución, así como una descripción de las tecnologías empleadas para implantar el sistema. Además, se menciona cómo está organizado el presente documento de tesis.

PLANTEAMIENTO DEL PROBLEMA

El problema fundamental a resolver con el desarrollo de este trabajo se refiere al manejo inadecuado de la información generada por los Cuerpos Académicos, relacionada con las actividades académicas que llevan a cabo. La carencia de un sistema de información provoca problemas en la obtención y consulta de la misma, tanto para los Cuerpos Académicos, como para el Área Académica correspondientes. La tesis se enfoca a manejar información de los Cuerpos Académicos inmersos en el Área Académica de Sistemas Computacionales, del Instituto de Ciencias Básicas e Ingeniería, mediante un sistema orientado a la Web, desarrollado de acuerdo con modelos de procesos, de datos y navegacionales.

JUSTIFICACIÓN

PROMEP es el Programa de Mejoramiento del Profesorado. Tiene como objetivo mejorar la formación y el desempeño de los Cuerpos Académicos de las Instituciones de Educación Superior, como un medio para elevar la calidad de la educación en este nivel. Además, pretende que los profesores de tiempo completo sean profesionales en la enseñanza, de modo que las instituciones alcancen niveles competitivos en los ámbitos nacional e internacional.

Los Cuerpos Académicos (CA) son un grupo de profesores de tiempo completo que comparten objetivos académicos y una o varias Líneas de Generación o

Aplicación del Conocimiento (LGAC) afines. Los CA son la fuerza motriz del desarrollo institucional que garantizan el cumplimiento de los objetivos institucionales tanto educativos como de generación y aplicación del conocimiento.

Existen tres grados de consolidación de los cuerpos académicos que son:

- *Cuerpos académicos consolidados*: todos tienen la máxima habilitación académica que los capacita para generar el conocimiento y aplicarlo, participan conjuntamente en líneas de generación del conocimiento bien definidas.
- *Cuerpos académicos en consolidación*: la mitad tiene el grado preferente y cuentan con productos sólidos de generación del conocimiento.
- *Cuerpos académicos en formación*: son aquellos que, aunque no alcanzan los parámetros para ser considerados como en etapa de consolidación tienen bien identificados sus actuales integrantes, al igual tienen bien definidas las líneas de generación de aplicación del conocimiento que desean atender.

El modelo educativo nacional se basa en la conformación de áreas académicas y de cuerpos académicos. La idea fundamental radica en que los profesores realicen actividades académicas, como investigación, docencia, difusión y vinculación. Estas actividades generan información que debe ser almacenada y clasificada adecuadamente de modo que se pueda actualizar conforme se generan nuevos productos. Es deseable para toda Institución de Educación Superior (IES) contar con un Sistema de Información que permita el uso eficiente de los datos referentes a los CA, y que cuente con funciones que faciliten el manejo de los datos. Por ejemplo que los Profesores Investigadores sean los que actualicen la información que ellos mismos generan, mediante la conexión desde su lugar de trabajo al Sistema de Información. Un segundo nivel de funcionalidad es requerido por la Dirección, tanto del Área Académica como de la Dependencia de Educación Superior, y se refiere a la obtención de información ya clasificada en diversos rubros. Dicha funcionalidad permite tomar decisiones y vigilar el desempeño

individual tanto de los Profesores – Investigadores como del los Cuerpos Académicos.

La presente tesis propone un sistema de información orientado a la Web que cumpla con las necesidades de gestión de la información, tomando como caso de estudio los Cuerpos Académicos del Área Académica de Sistemas Computacionales.

OBJETIVO GENERAL

Implantar un Sistema de Información orientado a Internet, para la adquisición, actualización y modificación de datos referentes a los Cuerpos Académicos, tomando como caso de estudio el Área Académica de Sistemas Computacionales del Instituto de Ciencias Básicas e Ingeniería.

OBJETIVOS ESPECÍFICOS

- Diseñar la base de datos según la metodología IDEF0 e IDEF1x.
- Realizar modelo para la implantación del sistema.
- Desarrollar la base de datos en MySQL.
- Desarrollar el sistema orientado a la Web, utilizando PHP.
- Validar el sistema con datos del Área Académica de Sistemas Computacionales.

DESCRIPCIÓN GENERAL DEL MÉTODO

Este trabajo propone la creación de un Sistema de Información bajo un enfoque integrador. Se plantea la integración de modelos IDEF0, correspondientes al modelo de procesos; el modelo de datos surge del modelo IDEF0 y se emplea IDEF1x. Finalmente, los modelos navegacionales basados en casos de uso, permiten el diseño final del sistema orientado a la Web. Los detalles de la creación de dichos modelos, así como su explicación, se muestran en el capítulo III.

Una vez desarrollados los modelos, el sistema se implantó con el motor de base de datos conocido como MySQL. Del mismo modo, la dinámica fue lograda mediante el lenguaje de programación conocido como PHP. En el Capítulo IV se ilustra el funcionamiento del sistema, con datos reales (restringidos por razones de confidencialidad) tomados del Área Académica de Sistemas Computacionales.

La combinación de estas metodologías se resume en las siguientes etapas:

- Etapa de análisis. Documentos PROMEP, entrevistas con PTC y directivos.
- Etapa de modelado de datos. IDEF1x
- Etapa de modelado navegacional. Diagramas de Flujo.
- Etapa de implantación. Código SQL.
- Etapa de dinámica del sistema. Código PHP – SQL – HTML.
- Etapa de pruebas y validación. Datos relacionados con el Área Académica de Sistemas Computacionales.

ALCANCES Y LIMITACIONES DEL PROYECTO

El resultado principal obtenido tras la realización de este trabajo de tesis, se refiere a un sistema de información orientado a la Web. Dicho sistema permite la interacción con los usuarios en diferentes niveles:

1. El PTC puede actualizar su información a través del sistema
2. El coordinador de los programas académicos accede y valida la información que vacían los PTC
3. La coordinación de investigación está habilitada para dar seguimiento a los proyectos de investigación llevados a cabo dentro del CITIS.

El sistema mostrado es un prototipo, implantado para validar los modelos desarrollados. Es posible adicionar funcionalidades, de acuerdo a las necesidades de información que planteen los usuarios del mismo. Sin embargo, el método de

integración planteado, así como los modelos desarrollados, constituyen un avance significativo en la construcción continúa del sistema, pues se contempla el tipo de información que se genera y consulta, a lo largo de la dinámica real de los Cuerpos Académicos.

ORGANIZACIÓN DE LA TESIS

En el Capítulo I se presenta un resumen del impacto de las tecnologías de información en las organizaciones, y un panorama de los modelos de información empleados para el desarrollo del sistema. En el Capítulo II se describe el modelo relacional de bases de datos, pues es el que se emplea en la implantación del sistema de información. En el Capítulo III describen a detalle los modelos desarrollados, y en Capítulo IV la implantación del sistema. El documento de tesis termina con las conclusiones generales.

CAPÍTULO

1

**TECNOLOGÍAS DE INFORMACIÓN,
MODELOS DE DATOS Y
ORGANIZACIONES**

CAPÍTULO I.

TECNOLOGÍAS DE INFORMACIÓN, MODELOS DE DATOS Y ORGANIZACIONES

Se presenta un resumen del impacto de las tecnologías de información en las organizaciones, así como un panorama de los modelos de información empleados para el desarrollo de sistemas. La integración de procesos organizacionales y modelos de información permiten desarrollar sistemas de información que respondan adecuadamente a la dinámica del sistema real modelado. El caso de estudio fue desarrollado siguiendo los conceptos teóricos descritos en este capítulo.

1.1 Impacto de las Tecnologías de Información (TI) en las Organizaciones.

Conceptualmente, una organización debe ser capaz de rediseñar los procesos internos sin ayuda de las TI's (Tecnologías de Información). Sin embargo, el rediseño de procesos es muy difícil de consumir sin las TI's habilitantes. Al mismo tiempo, una aplicación adecuada de estas tecnologías debe surgir a partir del entendimiento sobre la naturaleza de los procesos que se presupone debería apoyar. La introducción de TI's se debe reflejar, por ejemplo, en la reducción de acoplamiento físico entre los departamentos que llevan a cabo uno o más procesos [4].

La reducción del acoplamiento físico puede ser llevada a cabo a través de la compartición de recursos computacionales como las bases de datos. A través del

acceso directo a una base de datos compartida se pueden llevar a cabo varias funciones de modo paralelo.

Mientras la compartición de los Sistemas de Información (SI) permiten la reducción de acoplamiento físico, mejora al mismo tiempo el intercambio colaborativo de información. Éste se lleva a cabo a través de medios físicos de comunicación (como una red de computadoras), junto con la serie de programas de computadora que acceden a los SI según el proceso vaya necesitando datos. La conjunción de esta tecnología tiene como finalidad mejorar la comunicación y la colaboración entre las funciones que participan en un proceso interno.

Las TI's juegan un rol importante en la transformación de las tareas, tradicionalmente lentas y secuenciales, en trabajos paralelos, por medio de los cuales se incrementa la comunicación y la productividad. Por tal motivo es necesario diseñar procesos para que se integren sistemas de información adecuados, del mismo modo, es importante diseñar SI (Sistemas de Información) de acuerdo a los procesos para que se integren sistemas de las organizaciones.

Cabe mencionar que la automatización de lo procesos por sí misma no aporta soluciones sostenibles. Para facilitar la cooperación multi-funcional, se deben modificar las estructuras organizacionales tradicionales a través de equipos multidisciplinarios gestionados por generalistas de procesos. En la mayoría de los casos este tipo de habilitadores estructurales son un medio poderoso para complementar a las TI's [12].

Aunque los procesos de negocios y las TI's son componentes naturales, su relación nunca ha sido explotada totalmente. Por un lado se espera que la forma particular de conducir un negocio o una organización necesariamente influirá en el diseño y estructura de los sistemas de información que sirven de soporte a los procesos internos; por otro lado los avances en tecnología de información pueden generar nuevas oportunidades para las organizaciones, lo que a su vez influirá en el rediseño de los procesos de negocios.

Un proceso de negocios es un conjunto de actividades que reciben uno o varios insumos para crear un producto de valor para el cliente, así el término de Diseño de Procesos de Negocios se deriva de la práctica del desarrollo de Sistemas de Información. No obstante, la aplicación del Diseño de Proceso de Negocios no se basa necesariamente en la implementación de un nuevo sistema de tecnología de información. El diseño se basa principalmente en utilizar dichos sistemas para lograr que las organizaciones se organicen en torno a sus procesos; es decir, que el Diseño de Procesos de Negocios es la revisión fundamental y el diseño de procesos [5].

1.2 Internet, procesos internos y las TI's.

Debido a que todos los procesos tienen un cliente que puede ser interno o externo a la organización, una organización puede reducir confusiones y optimizar funciones si se obtiene un cumplimiento del proceso global. Podemos visualizar aplicaciones innovadoras de las TI's.

En la actualidad tanto las Intranet como Internet son un factor clave para lograr el intercambio colaborativo de información sin necesidad de contar con interacción física entre las funciones involucradas en la realización de un proceso. Por lo tanto, los procesos de la organización deben reflejarse en sistemas orientados a Internet, en los cuales se incrementa la eficiencia en el flujo de información mediante bases de datos accesibles desde Internet para soportar los procesos. El flujo y la secuencia de dichos procesos también deben ser implantados con aplicaciones sobre Internet.

1.3 Modelos de información mediante el enfoque IDEF

1.3.1 IDEF0

IDEF0 es un método diseñado para el modelado de decisiones, acciones y actividades de una organización o sistema. IDEF0 se derivó de un lenguaje gráfico estructurado llamado Técnicas de Diseño para Análisis Estructurado (SADT) [12]. IDEF0 realiza el dominio experto envolvente y los consensos de toma de decisiones a través de recursos gráficos simplificados. Como una herramienta de comunicación asiste al modelador para identificar qué funciones está ejecutando y qué se necesita ejecutar de estas funciones, qué es lo que está correcto del sistema actual y qué es lo que está mal.

1.3.2 IDEF1x.

La técnica IDEF1x se diseñó para modelar y analizar estructuras de datos que sirven de soporte al establecimiento de sistemas de información. IDEF1x emplea un conjunto de notaciones gráficas para expresar el conjunto de reglas y relaciones entre las entidades que componen un sistema, basado tanto en un enfoque de relación-entidad, como en la teoría relacional de base de datos [12]. La metodología IDEF1x comparte muchas de las construcciones encontradas en modelos de relación-entidad: entidades, atributos y relaciones entre estas entidades. Las entidades son objetos acerca de los datos que pueden ser almacenados.

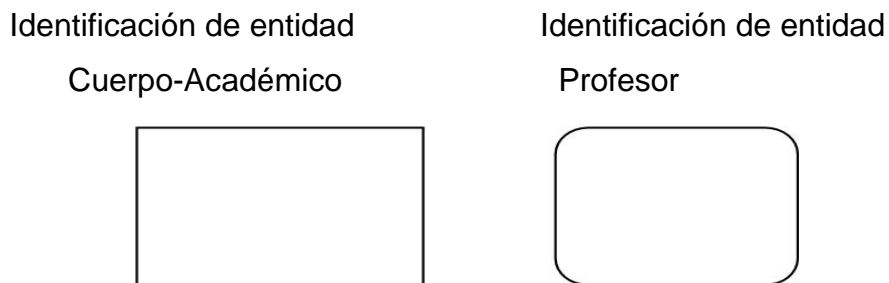


Figura 1.1. Sintaxis de entidad.

Las características de las entidades que pueden ser cuantificables, (e.g., nombre de la publicación, motivo de la publicación, nombre del proyecto de investigación, etc.), se les conoce como atributos. También existen relaciones entre las entidades. Por ejemplo, entre profesor y cuerpo académico la relación podría ser “Los Cuerpos Académicos se componen de profesores”.

1.3.2.1 Entidades

Una entidad es una clase de objetos que comparten características comunes. Un miembro individual de esta clase es llamado entidad caso (muestra). Si la entidad clase fue “Cuerpo Académico”, una entidad caso debe ser el Cuerpo Académico “**Computación Inteligente**”. Una entidad caso debe de corresponder a más de una entidad clase. Por ejemplo, si otra clase fuera “LGAC”, Agentes Inteligentes es una Línea de Investigación, en este caso podría ser miembro de la clase LGAC.

Una entidad está representada como una casilla como se muestra en la *Figura 1.1*. Cabe señalar que existen dos tipos de entidades: dependientes e independientes. Una entidad es llamada para ser identificada independientemente si cada caso de la entidad puede ser únicamente identificada sin determinar sus relaciones a otra entidad. Si este no es el caso, la entidad es dependiente de identificador.

1.3.2.2 Atributos.

Una entidad se caracteriza por un conjunto de atributos. “LGAC”, un atributo podría ser “código de la LGAC”. Un atributo puede ser también un conjunto de valores característicos. El valor de un atributo es el contenido particular para el atributo de la entidad. En este ejemplo el valor de un atributo podría ser “11234” como el valor del atributo “código de la LGAC”. Una entidad debe tener un atributo o un conjunto de atributos cuyos valores específicos identifiquen a una instancia única de la entidad. Este atributo o grupo de atributos se conocen como la **clave primaria** de entidad. Si la entidad “LGAC” tiene un conjunto de atributos como “código de la LGAC”, “nombre de la LGAC”, “descripción de la LGAC”, “fecha de inicio de la

LGAC”; el atributo “código de la LGAC” podría ser la clave primaria, debido a que es muy probable que más de una LGAC tenga el mismo nombre o fecha de inicio.

El modelo IDEF1x nos dice que cada atributo es usado únicamente por una entidad, lo que se conoce como la regla del propietario único, y cada instancia de la entidad debe tener un valor para cada atributo de la entidad, es decir, el atributo debe ser aplicable a cada instancia asociada con la entidad, lo que se conoce como la regla que prohíbe el conjunto vacío. Del mismo modo ninguna instancia de la entidad puede tener más de un valor para cualquier atributo de la entidad, esa regla se conoce como la regla de no-duplicación. Los atributos se identifican con nombre único e intuitivo, los atributos se muestran al colocarlos dentro de la gráfica que identifica a la entidad.

Los atributos de clave primaria se colocan en la parte superior de la lista y se separan de los demás atributos mediante una línea como se muestra en la *Figura 1.2*.

1.3.2.3 Claves primarias y alternas

La clave de una entidad es el más pequeño conjunto de atributos que identifican a una instancia particular de todas las demás instancias de la entidad, esto se conoce como la regla de la clave más pequeña. Normalmente la clave es un atributo único.

Sin embargo, en ocasiones un solo atributo no provee la suficiente información para ser considerado como una clave. Cada entidad debe tener al menos una clave y, en algunos casos, puede existir más de una clave candidata. En el ejemplo de la *Figura 1.2* el atributo “**Id_profesor**” es la clave primaria. Cuando esto ocurre, uno de los dos atributos debe ser designado como la clave primaria mientras que las claves candidatas se conocen formalmente como claves alternas. Mientras que las claves primarias se identifican como lo muestra la *Figura 1.3*, las claves alternas se identifican colocando “(AKn)” enseguida del nombre del atributo. *n* es un entero único. Si la clave primaria se compone de más de un atributo, el valor de cada atributo, que no es una clave, debe ser funcionalmente dependiente de la clave primaria completa, esto se conoce como la regla de la dependencia funcional completa. Del mismo modo cada atributo no clave debe ser funcionalmente dependiente únicamente de la clave primaria.

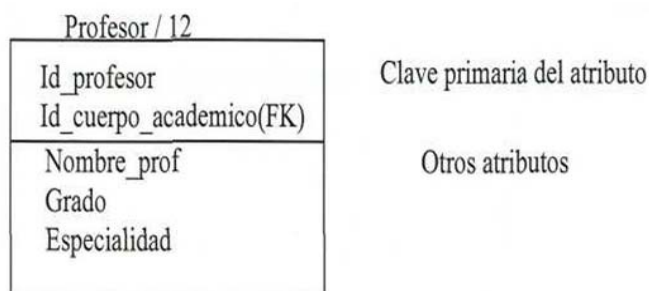


Figura 1.2. Atributos y sintaxis de claves primarias.

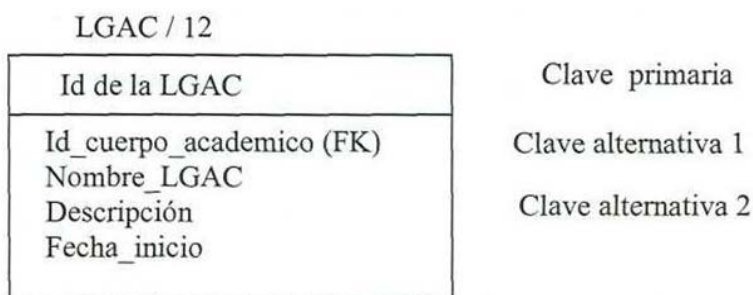


Figura 1.3. Atributos y sintaxis de las claves

1.3.2.4 Claves foráneas (Claves externas).

Si existe una relación específica entre dos entidades, o si existe una relación de jerarquía (categorización), entonces los atributos que constituyen la clave primaria de la entidad padre o genérica respectivamente, son heredados como atributos de la entidad hija o heredada respectivamente. Estos atributos heredados se conocen como claves externas ó foráneas. Un atributo heredado puede ser una clave primaria, una parte de la clave primaria, una clave alterna o cualquier otro atributo. Si todos los atributos que constituyen la clave primaria de una entidad padre se heredan como parte de la clave primaria de la entidad hijo, entonces la relación entre el padre y el hijo es una relación de identificación. Si cualquiera de los atributos heredados no forman parte de la clave primaria entonces se tiene una relación de no-identificación. En una relación de categorización tanto la entidad genérica como las entidades heredadas representan el mismo concepto del mundo real.

La clave primaria para todas las entidades heredadas se hereda a través de la relación de categorización y pertenece generalmente a la entidad genérica. Una clave externa o foránea se muestra al colocar el texto "(FK)" enseguida del nombre del atributo de la entidad hijo. Si este atributo pertenece a la clave primaria de la entidad hijo se coloca encima de la línea horizontal y la caja de la entidad tendrá entonces esquinas redondeadas. Si la clave externa no pertenece a la clave primaria de la entidad hijo se coloca debajo de la línea horizontal y la entidad se representa por un cuadrado. Una entidad debe tener cuando menos una clave

externa por cada entidad de la cual dependa (ya sea por medio de una relación o por medio de una categorización), como se muestra en la *figura 1.4*.

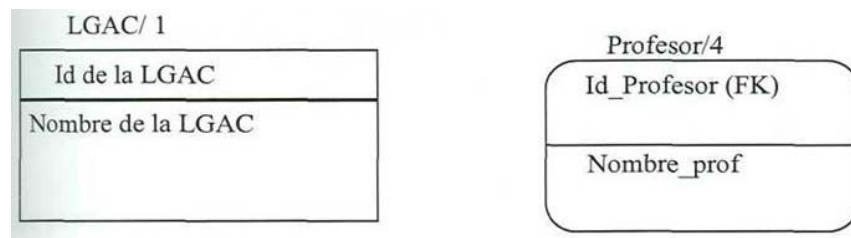


Figura 1.4. Sintaxis de claves foráneas

En algunas ocasiones una entidad hijo tendrá más de una relación con la misma entidad padre, en este caso el hijo heredará la clave primaria del padre más de una ocasión, y cada instancia de atributo heredado puede tener diferentes valores. La lista de materiales de la *figura 1.5* se representa mediante dos entidades, “Producto de investigación” y “Tesis”. En esta relación una pieza dada puede ser un componente de una estructura y también puede ser parte de una estructura inferior. Si la clave primaria de “Producto de investigación” es “Id_proyecto_investigación” entonces el atributo “Id_producto_investigación” puede ser heredado dos veces por la entidad “Tesis”. Cuando esto ocurre se le debe asignar un nombre de rol para cada ocurrencia (para cada valor del Id del proyecto de investigación) y así distinguirlos. Los nombres de rol se separan de los nombres de los atributos mediante un punto. Para entender la sintaxis utilizada en la *figura 1.5* analice las relaciones que se presentan en la *figura 1.7*.



Figura 1.5. Sintaxis del nombre del rol

1.3.2.5 Relaciones de conexión.

Una relación de conexión es una asociación entre dos entidades. Cada instancia de la primera entidad (la entidad padre) de algún modo está relacionada con 0, 1, ó más de una instancia de la segunda entidad (la entidad hijo). Cada instancia de la entidad hijo está asociada exactamente con una instancia de la entidad padre. En otras palabras, una entidad hijo existe sólomente si su correspondiente entidad padre existe. La relación de conexión puede ser definida con más detalle por medio de la cardinalidad de la relación, es decir, el número de entidades hijo para cada entidad padre. En un modelo IDEF1x.

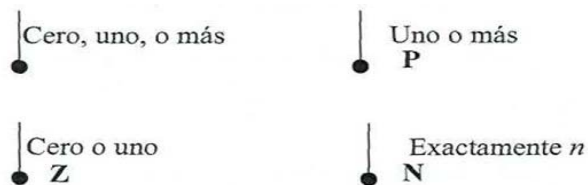


Figura 1.6. Sintaxis de relación de cardinalidad

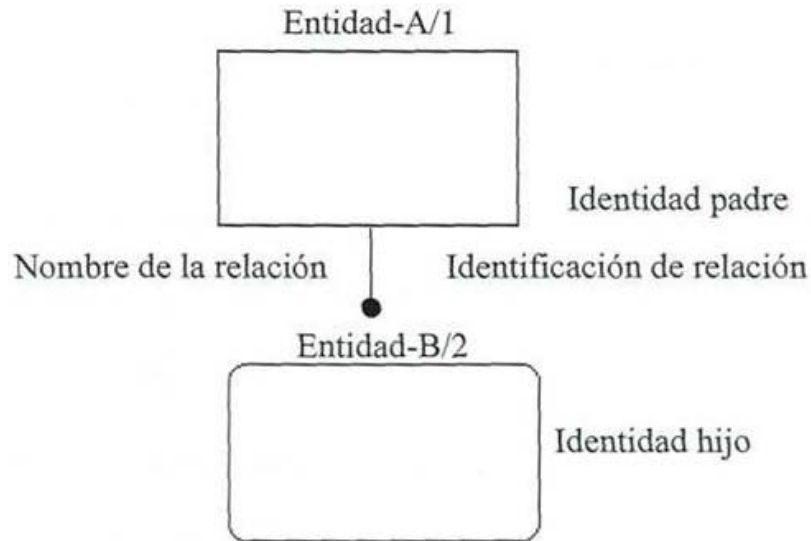


Figura 1.7. Sintaxis de identificación de relación

Se pueden expresar las siguientes relaciones:

1. Cada instancia de la entidad padre puede tener 0, 1, o más instancias de la entidad hijo.
2. Cada instancia de la entidad padre puede tener exactamente 0, ó exactamente 1 instancia de la entidad hijo.
3. Cada instancia de la entidad padre puede tener 1 o más instancias de la entidad hijo.
4. Cada instancia de la entidad padre tiene un número preciso de instancias de la entidad hijo.

Si la instancia de alguna entidad hijo se identifica por medio de su asociación con la entidad padre se conoce como una relación de no-identificación. También se puede definir reglas acerca de las relaciones múltiples. Un ejemplo de esto sería el operador lógico o exclusivo. Esta regla nos dice que para una entidad padre dada,

si existe una instancia de la entidad hijo, cualquier otra no debe existir. Esto es una relación de categorización descrita más adelante.

Una relación de conexión específica se describe mediante una línea que va de la entidad padre a la entidad hijo, con un círculo al final de la línea la cardinalidad por defecto es 0, 1, ó más. Se coloca una P, una Z ó una n para el tipo de cardinalidades específicas, tal como se ve en la *figura 1.6*. Una línea sólida describe una relación de identificación entre las entidades padre e hijo. Una línea punteada describe una relación de no-identificación. Es decir las entidades padre e hijo pueden ser identificadas independientemente una de otra.

Es necesario darle un nombre a la relación. Los nombres de las relaciones entre entidades deben ser únicos para cada par. Las relaciones siempre se expresan en la relación padre e hijo, formando una oración que se lee en el sentido entidad padre, nombre de la relación, cardinalidad de la relación y entidad hijo. Las relaciones deben ser válidas si se siguen en la relación opuesta.

1.3.2.6 Relaciones de categorización.

Típicamente algunas entidades pueden ser categorizadas a partir de otras, por ejemplo, la entidad obrero y la entidad gerente son entidades específicas de la entidad empleado. Las relaciones de categorización pueden ser completas o incompletas.

Una categorización completa es una relación en la cual cada instancia de la entidad genérica está asociada exactamente con una instancia de las entidades de categorización. Ambas comparten el mismo identificador.

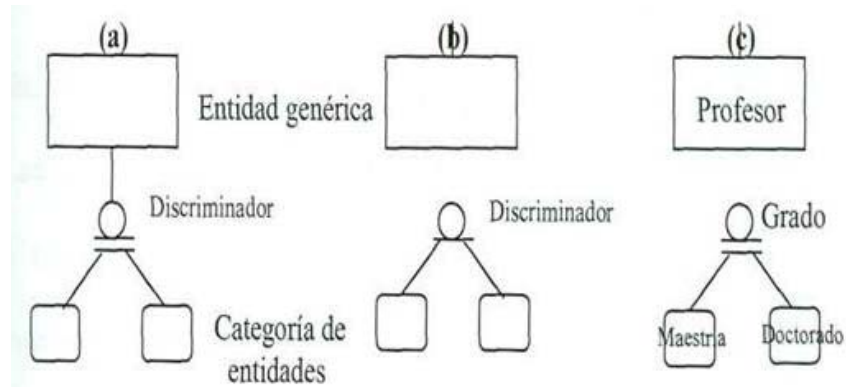


Figura 1.8. Completo (a), (c) contra incompleto (b) categorización

Las entidades de categorización deben ser mutuamente excluyentes. Esto significa que cualquier instancia de la entidad genérica exactamente en una entidad de categorización. También es posible que una instancia de la entidad genérica no esté representada en ninguna de las entidades de categorización, lo que da lugar a una categorización incompleta.

Cada instancia de la entidad genérica debe tener un atributo llamado discriminador que identifique a cuál entidad de categorización pertenece. Una relación de categorización se muestra en la figura 1.8. Como se ve en la figura, si la categorización es completa el círculo estará asociado con 2 líneas horizontales.

Las relaciones de categorización se leen de la siguiente forma: "Un Profesor puede tener una Maestría o un Doctorado" si la categorización es incompleta. Si la

categorización es completa se lee “Un Profesor puede tener el grado de Maestría o de Doctorado”. La relación se lee “es un / una” si se lee en sentido opuesto.

Es importante hacer notar que se pueden tener categorizaciones infinitas, es decir, una entidad de categorización puede también ser una entidad genérica en otra relación de categorización.

1.3.2.7 Relaciones no específicas.

Las relaciones de categorización y padre-hijo son consideradas relaciones específicas porque muestran con precisión cómo una instancia de entidad se relaciona con otra. En un modelo IDEF1x determinado, todas las relaciones deben ser específicas. Sin embargo, durante el desarrollo del modelo puede resultar más fácil definir una relación no específica y entonces refinarla en una relación específica.

Las relaciones no específicas también se conocen como relaciones muchos a muchos. En una relación como ésta cada instancia de entidad en la primera relación se puede asociar con 0, 1 ó más instancias de la entidad en la segunda relación, mientras que una instancia de entidad en la segunda relación se puede asociar con 0, 1 ó más instancias de entidad de la primera relación.

Las relaciones no específicas se pueden definir al expresar la cardinalidad a cada lado de la relación. Si existe una cardinalidad de exactamente 1 en cada extremo de la relación, entonces la relación es específica.

Por ejemplo, cuando una relación de este tipo se lee en cada dirección, una forma la frase “Las investigaciones tienen como un producto la tesis” y la otra se lee “La tesis es un producto de investigación” (Ver figura 1.9).

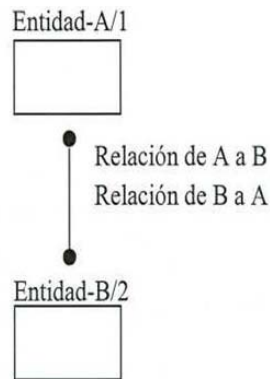


Figura 1.9. Sintaxis de relación no específica

Como se muestra en la *Figura 1.10*, las relaciones no específicas deben ser reemplazadas por 1 o más relaciones específicas.

Es importante señalar que IDEF1x permite modelar y analizar estructuras de datos que sirven de soporte al establecimiento de sistemas de información, las reglas y las relaciones entre las entidades que componen al sistema. El modelo IDEF1x se describe en el Capítulo III.



Figura 1.10 Rompimiento de relaciones no específicas

1.4 Guía de las cinco formas normales de las bases de datos relacionales

1.4.1 Primera forma normal

La primera forma normal trata con la forma del registro. Bajo la primera forma normal, todas las ocurrencias de tipo registro deben contener el mismo número de campos. La primera forma normal excluye campos repetidos y grupos.

La segunda y tercera formas normales tienen que ver con las relaciones entre campos clave y campos no clave. Bajo la segunda y tercera forma normal en campo no clave debe proveer un hecho (valor) referente al campo clave. Además, el registro debe satisfacer la primera forma normal.

1.4.2 Segunda forma normal

La segunda forma normal es violada cuando un campo que no es clave tiene un valor de un subconjunto del campo de la clave. Es sólo relevante cuando el campo clave es compuesto por varios campos.

Cuando el diseño de datos es cambiado por reemplazar registros no normalizados por registros normalizados, a este proceso se le llama normalización. El diseño normalizado refuerza la integridad de los datos minimizando la inconsistencia y redundancia de datos.

1.4.3 Tercera forma normal

La tercera forma normal es violada cuando un campo no clave contiene un valor de otro campo no clave.

1.4.4 Dependencias funcionales

En la teoría de las bases de datos relacionales, la segunda y la tercera forma normal son definidas en términos de dependencias funcionales. Un campo Y es

funcionalmente de un campo (o campos) X si es inválido tener dos registros con el mismo valor de X pero diferentes valores en Y. Cuando X es un campo clave, entonces todos los campos son por definición funcionalmente dependientes en X, siempre y cuando no existan dos registros que tengan el mismo valor en X.

Las dependencias funcionales sólo existen cuando las cosas involucradas tienen identificadores únicos y singulares. Es importante puntualizar que las dependencias funcionales y las varias formas normales son realmente sólo definidas por situaciones en donde existen identificadores únicos y singulares.

1.4.5 Cuarta y Quinta formas normales

La cuarta y quinta formas normales tratan con campos que pueden tener diferentes valores. Un campo que puede tener diferentes valores debe corresponder a una relación muchos a muchos o muchos a uno. En ese sentido, la cuarta y quinta forma normal son también llaves compuestas. Estas llaves normales intentan minimizar el número de campos involucrados en la llave compuesta.

1.4.6 Cuarta forma normal

Bajo la cuarta forma normal, un registro no debe contener campos que acepten diferentes valores independientes en una entidad. Además el registro debe satisfacer la tercera forma normal.

El problema principal al violar la cuarta forma normal es que lleva hacia incertidumbres en las políticas de mantenimiento. Varias políticas son posibles para el mantenimiento de dos campos que aceptan diferentes valores independientes en un registro.

Otros problemas por violar la cuarta forma normal son similares a esos mencionados antes por violar la segunda y tercera forma normal. Esto toma diferentes variaciones dependiendo de las políticas de mantenimiento seleccionadas.

Si hay repeticiones, entonces la actualización debe hacerse en múltiples registros, y los registros pueden volverse inconsistentes.

Las dependencias multivaluadas son definidas esencialmente como una relación que acepta la política del mantenimiento cruzado. La dependencia multivaluada y la cuarta forma normal también aplica a las restricciones que involucran a más de dos campos.

1.4.7 Quinta forma normal

La quinta forma normal trata con casos donde la información puede ser reconstruida por piezas pequeñas de información que pueden ser mantenidas con menos redundancia. La segunda, tercera y cuarta formas normales también sirven a este propósito, pero la quinta forma normal generalizan los casos no cubiertos por las otras. En otras palabras podemos decir que un registro si cumple con la quinta forma normal cuando la información que contiene no puede ser reconstruida por varios registros pequeños (de registros que cada uno tiene menos campos que el registro original). El caso donde todos los registros pequeños tienen la misma llave es excluido. Si un registro puede ser sólo descompuesto en registros pequeños donde todos tienen la misma llave, entonces el registro es considerado dentro de la quinta forma normal sin descomposición. Un registro que cumple con la quinta forma normal cumple también con la cuarta, tercera, segunda y primera forma normal.

La quinta forma normal no difiere de la cuarta forma normal a menos que exista una limitación simétrica. Una ventaja de la quinta forma normal es que ciertas redundancias pueden ser eliminadas.

La cuarta y quinta formas normales tratan con combinación de campos que contienen valores diferentes. Si un registro viola la cuarta forma normal, el proceso de normalización asociado lo descompone en dos registros, cada uno contiene menos campos que el registro original. Cualquiera de estas violaciones a la cuarta forma normal es descompuesta en dos registros y así sucesivamente hasta que

todos los registros resultantes cumplan con la cuarta forma normal. En cada etapa el conjunto de registros después de la composición contiene exactamente la misma información que antes de la descomposición.

1.4.8 Redundancias inevitables

La normalización ciertamente no remueve todas las redundancias. Ciertas redundancias parecen ser inevitables, particularmente aquellas que contienen diferentes valores ya definidos son más dependientes que independientes.

1.4.9 Redundancia entre registros.

Las formas normales anteriormente descritas, tratan solo con redundancias ocurridas dentro de un registro. La quinta forma normal es considerada la última forma normal con respecto a tales redundancias. La redundancia entre registros ha sido reconocida durante algún tiempo y recientemente ha sido direccionada en términos de la forma normal y la normalización.

CAPÍTULO

2

**EL MODELO RELACIONAL
DE SISTEMAS DE INFORMACIÓN**

CAPÍTULO II. EL MODELO RELACIONAL DE SISTEMAS DE INFORMACIÓN

En este capítulo se describe el modelo relacional, así como el lenguaje SQL para la implantación del sistema de información. Con esto se pretende observar de modo integral los procesos internos y el sistema de información que lo soporta.

Del mismo modo, se presenta SQL en el SGBD MySQL como el lenguaje de implantación de los modelos, y la descripción de algunas técnicas de modelado que al igual algunas fueron descritas en el capítulo anterior. Estos modelos van de la mano con el modelo relacional para la completa realización de un sistema de información.

2.1 Tecnología de implantación

En esta sección se presenta uno de los varios modelos de bases de datos, como es el modelo relacional, ya que los modelos IDEF1x fueron diseñados originalmente de acuerdo a este tipo de bases de datos, además de que los SGBD's relacionales permiten construir aplicaciones informáticas utilizando un lenguaje anfitrión (host language) como puede ser C, C++, Java, o PHP, como en el caso del sistema MySQL, que fue el empleado en el presente trabajo de tesis. Esto quiere decir que mediante un lenguaje anfitrión es posible reproducir los procesos dinámicos llevados a cabo dentro de las organizaciones. Un sistema de información no sólo debe estar diseñado para realizar transacciones simples, sino

todas las transacciones que surgen a lo largo de las actividades generadas durante los procesos. Así se habilita el intercambio colaborativo de información entre los elementos que componen a la organización, se reduce la interacción física de los mismos y se permite el acoplamiento físico paralelo.

Desde el punto de vista de un modelo relacional, una base de datos es una colección de relaciones. Si se visualiza una relación como una tabla de valores, cada fila de la tabla representa una colección de datos relacionados entre sí. Dichos valores se pueden interpretar con hechos que describen una entidad o un vínculo entre entidades del mundo real. El nombre de la tabla y los nombres de las columnas ayudan a interpretar el significado de los valores que están en cada fila de la tabla. Por ejemplo, la tabla se llama Profesor por que cada fila representa hechos acerca de esta entidad en particular. Los nombres de las columnas – Nombre del profesor, id, grado, especialidad- especifican cómo interpretar los valores de datos de cada fila como base en la columna en que se encuentra cada valor. Todos los datos de una misma columna tienen el mismo tipo de datos. En términos de un modelo relacional una fila se denomina Tupla, una cabecera de columnas es un atributo y una tabla es una relación. El tipo de datos que pueden aparecer en cada columna se llama dominio. A continuación se definen estos términos con mayor precisión.

PROFESOR	Nombre	Apellidos	Id	Grado	Especialidad
	Omar	López	olopez	Doctorado	Informática Industrial
	Roberto	Suárez	jsuarez	Doctorado	

Tabla 2.1. Ejemplo de la Tabla

2.1.1 Modelo relacional

En este apartado se describen a groso modo las restricciones que se pueden especificar en un esquema de base de datos relacional. Entre las cuales tenemos a las restricciones de dominio, de clave, de integridad de entidades y de integridad referencial.

2.1.1.1 Restricciones de dominio.

Especifican que el valor de cada atributo A un valor atómico del dominio $\text{dom}(A)$. Para este atributo, los tipos de datos asociados a los dominios por lo regular incluyen los tipos de datos numéricos estándar de los números enteros (como entero-corto, entero-largo) y reales (flotantes y flotantes de doble precisión). También se dispone de caracteres, cadenas de longitud fija y cadenas de longitud variable, así como tipos de datos, fecha, hora, de marca de tiempo y dinero.

2.1.1.2 Restricciones de clave.

Una relación se define como un conjunto de tuplas y por definición, todos los elementos de un conjunto son distintos; por tanto, todas las tuplas de una relación deben ser distintas. Esto significa que no puede haber dos tuplas que tengan la misma combinación de valores para todos sus atributos. Por lo regular existen subconjuntos de atributos de un esquema de relación R con la propiedad de que no deben existir dos tuplas que tengan la misma combinación de valores para otros atributos.

Por ejemplo, consideremos la *Tabla 2.1*. El conjunto de atributos Id del Profesor es una clave del profesor, por que no puede haber dos tuplas PROFESOR que tengan el mismo valor de Id del Profesor.

2.1.1.3 Integridad de entidades, integridad referencial y claves externas.

La restricción de integridad de entidades establece que ningún valor de la clave primaria puede ser nulo. Esto es por que el valor de la clave primaria sirve para identificar las tuplas individuales en una relación. El que la clave primaria tenga valores nulos impide que podamos identificar algunas tuplas.

Las restricciones de clave y de integridad de entidades especifican relaciones individuales. La restricción de integridad referencial específica entre dos relaciones y sirve para mantener la consistencia entre tuplas de las dos relaciones. En términos informales, la restricción de integridad referencial establece que una tupla en una relación que haga referencia a otra relación deberá referirse a una tupla existente en esa relación. Cabe hacer mención que estas reglas son tomadas en cuenta al momento de crear los modelos IDEF1x como se mencionó con anterioridad. Por lo tanto al seguir las reglas IDEF1x se cumplen las restricciones de integridad de claves, integridad referencial y claves externas.

2.1.2 Operaciones de actualización con relaciones

Es posible clasificar las operaciones del modelo relacional en obtenciones y actualizaciones, de aquí que sean tres las operaciones básicas de actualización

que se efectúan con relaciones: Insertar, Eliminar y Modificar. Insertar sirve para insertar una o mas tuplas nuevas en una relación; Eliminar sirve para eliminar tuplas y Modificar sirve para modificar los valores de algunos atributos. Cabe señalar que siempre que se apliquen actividades de actualización, se debe cuidar de no violar las restricciones de integridad especificadas en el esquema de bases de datos relacional.

2.1.2.1 La operación insertar.

Esta operación proporciona una lista de valores de atributos para una nueva tupla t que se ha de insertar en una relación R . La inserción puede violar cualquiera de los cuatro tipos de restricciones anteriormente mencionadas. Las restricciones de dominio pueden violarse si se proporciona un valor de atributo que no aparezca en el dominio correspondiente. Las restricciones de clave pueden violarse si un valor de la nueva tupla t ya existe en otra tupla de la relación $r(R)$. La integridad de entidades puede violarse si la clave primaria de la nueva tupla t es nula. Y la integridad referencial puede violarse si el valor de cualquier clave externa de t hace referencia en una tupla que no existen las relaciones referidas.

Ejemplo:

```
INSERTAR <López, olopez, doctorado>
```

2.1.2.2 La operación eliminar.

Esta operación sólo puede violar la integridad referencial, si las claves externas de otras tuplas hacen referencia a la tupla que se ha de eliminar. Para especificar la

eliminación, una condición expresada en términos de los atributos de la relación selecciona la tupla o las tuplas por eliminar.

Ejemplo:

```
Elimina la tupla Profesor con el ID del Profesor = olopez.
```

2.1.2.3 La operación modificar.

Esta sirve para cambiar los valores de uno o más atributos de una tupla (o tuplas) de una relación R. Es necesario especificar una condición para los atributos de R a fin de seleccionar la tupla o tuplas que se modificarán.

Ejemplo:

```
Modificar el grado de la tupla PROFESOR con ID de  
Profesor olopez cambiándolo a omarl.
```

Es importante hacer mención que las operaciones descritas en este trabajo, se programan actualmente con un lenguaje de alto nivel como SQL (Structured Query Language).

2.2 Lenguaje SQL

El Structured Query Language (SQL) es más que un lenguaje de programación de base de datos. Hablamos por tanto de un lenguaje normalizado que nos permite trabajar con cualquier tipo de lenguaje (PHP, HTML) en combinación con cualquier tipo de base de datos (como MySQL). El hecho que sea estándar no quiere decir que sea idéntico para cada base de datos. En efecto, determinadas bases de

datos implantan funciones específicas que no tienen necesariamente que funcionar en otras.

La etapa de implantación necesariamente requiere de herramientas computacionales, hardware y software, para llevarse a cabo. SQL sin duda representa una herramienta muy eficaz en la etapa de implantación del sistema de información que soportará, dará servicio, apoyo, etc. a la ejecución eficiente de los procesos internos de las organizaciones. En el caso de este trabajo de tesis el modelo IDEF1x tiene su correspondiente implantación con código SQL-HTML como se describe en el capítulo III.

2.2.1 Tipos de campos.

Una base de datos está compuesta de tablas donde almacenamos registros catalogados en función de distintos campos (atributos). Un aspecto previo a considerar es la naturaleza de los valores que introducimos en esos campos. Dado que una base de datos trabaja con todo tipo de información, es importante especificarle qué tipo de valor le estamos introduciendo. Estos tipos comunes se muestran en la *Tabla 2.2*.

Tabla 2.2. Tipos de datos de los SGBD

Alfanuméricos	Contienen cifras y letras. Presentan una longitud limitada (255 caracteres).
Numéricos	Existen de varios tipos, principalmente, enteros (sin decimales) y reales (con decimales).
Boléanos	Poseen dos formas: Verdadero y falso (Si o No).
Fechas	Almacenan fechas solicitando posteriormente su explotación. Almacenar fechas de esta forma posibilita ordenar los registros por fechas o calcular los días entre una fecha y otra.
Memos	Son campos alfanuméricos de longitud limitada. Presentan el inconveniente de no poder ser indexados.
Autoincrementables	Son campos numéricos que incrementan en una unidad su valor para cada registro incorporado. Su utilidad resulta más que evidente: Servir de identificador ya que resultan exclusivos de un registro.

2.2.2 Operaciones más comunes en SQL.

Es importante mencionar que las operaciones más frecuentes usadas en SGBD's, independientemente de los campos en los cuales se almacenan los datos son:

2.2.2.1 Crear nueva tabla.

La orden *Create table* sirve para especificar una relación dándole un nombre y especificando sus atributos y restricciones.

La sintaxis utilizada es la siguiente:

```
CREATE TABLE PROFESOR
(NOMBRE VARCHAR (15) NOT NULL,
APELLIDO VARCHAR (15) NOT NULL,
CODIGO DEL PROFESOR CHAR (9) NOT NULL,
GRADO INT,
ESPECIALIDAD CHAR,
PRIMARY KEY (CÓDIGO DEL PROFESOR));
```

2.2.2.2 Añadir un nuevo registro

Los registros pueden ser introducidos a partir de sentencias que emplean la instrucción *Insert*. La sintaxis utilizada es la siguiente:

```
Insert Into nombre_tabla (nombre_campo1, nombre_campo2,...)
values (valor_campo1,valor_campo2,...)
```

Un ejemplo sencillo a partir de nuestra tabla modelo es la introducción de un nuevo profesor lo cual se haría con una instrucción de este tipo:

```
Insert Into PROFESOR (nombre, apellido, grado, especialidad)
Values ('Omar', 'López', 'Doctorado', 'Informática industrial')
```

Por supuesto no es imprescindible rellenar todos los campos del registro. Esto sí, puede ser que determinados campos sean definidos cuando construimos nuestra tabla mediante la base de datos.

2.2.2.3 Borrar un registro.

Para borrar un registro nos servimos de la instrucción *Delete*. En este caso debemos especificar cuál o cuáles son los registros que queremos borrar. Es por ello necesario establecer una sección que se llevará a cabo mediante la cláusula *Where*.

Ahora se muestra cuál es el tipo de sintaxis utilizado para efectuar estas supresiones:

```
Delete From nombre_tabla Where condiciones_de_selección
```

Si queremos por ejemplo borrar todos los registros de los profesores que se llamen Omar lo haríamos del siguiente modo:

```
Delete From PROFESOR Where nombre='Omar'
```

Hay que tener cuidado con esta instrucción, ya que si no especificamos una condición con *Where*, lo que estamos haciendo es borrar toda la tabla.

2.2.2.4 Actualizar un registro.

Update es la instrucción que nos sirve para modificar nuestros registros. Como para el caso de *Delete*, necesitamos especificar por medio de *Where* cuáles son

los registros en los que queremos hacer efectivas nuestras modificaciones. Además, obviamente, tendremos que especificar cuáles son los nuevos valores de los campos que deseamos actualizar. La sintaxis es de este tipo:

```
Update nombre_tabla set nombre_campo1 =
valor_campo1, nombre_campo2 = valor_campo2,...
valor_campo_n condiciones_de_selección
```

Un ejemplo aplicado:

```
Update PROFESOR Set nombre='Roberto' Where nombre='Joel'
```

Mediante esta sentencia cambiamos el nombre de Roberto por el de Joel en todos los registros cuyo nombre sea Roberto. Aquí también hay que ser cuidadoso de no olvidarse de usar Where, de lo contrario, modificaríamos todos los registros de nuestra tabla.

2.2.2.5 Consultas SQL

SQL cuenta con una instrucción básica para obtener información de una base de datos: La instrucción Select, la cual no tiene que ver con la operación seleccionar del álgebra relacional. La forma básica de la instrucción Select consta de tres cláusulas, Select, from y where. Esta instrucción nos permite la selección total o parcial de una tabla.

En dicha selección hay que especificar:

- Los campos que queremos seleccionar
- La tabla en la que hacemos la selección.

En nuestra tabla modelo de profesores podríamos hacer, por ejemplo, una selección del nombre y grado de los profesores con una ilustración de este tipo:

```
Select nombre, grado From PROFESOR
```

Si quisiésemos seleccionar todos los campos, es decir, toda la tabla, podríamos utilizar el comodín * del siguiente modo:

```
Select * From PROFESOR
```

Resulta también muy útil filtrar los registros mediante condiciones que vienen expresadas después de la cláusula *Where*. Si quisiésemos mostrar los profesores de un determinado grado usaríamos una expresión como esta:

```
Select * From PROFESOR Where grado Like 'doctorado'
```

Además podríamos ordenar los resultados en función de uno o varios de sus campos. Para último ejemplo los podríamos ordenar por nombres así:

```
Select * From PROFESOR Where grado Like 'doctorado' OrderBy nombre
```

Teniendo en cuenta que puede haber más de un Profesor con el mismo nombre, podríamos dar un segundo criterio que podría ser la especialidad:

```
Select * From PROFESOR Where grado Like 'doctorado' Order By  
nombre, especialidad
```

Si invirtiésemos el orden << nombre, especialidad>> por <<especialidad, nombre>>, el resultado sería distinto. Tendríamos los profesores ordenados por

especialidad y aquellos que tuviesen especialidad idéntica se subclasificarían por el nombre.

Es posible también clasificar por orden inverso. Si por ejemplo quisiésemos ver nuestros profesores por orden de grados, teniendo a los doctores en primer lugar escribiríamos así:

```
Select * From PROFESOR Order By grado
```

Una opción interesante es la de efectuar selecciones sin coincidencia. Si por ejemplo buscásemos el saber de qué grado de estudios se encuentran nuestros profesores sin necesidad de que para ello aparezca varias veces el mismo grado usaríamos una sentencia de esta clase:

```
Select Distinct grado From PROFESOR Order By Grado
```

Así evitaríamos ver repetido el grado de Maestrías por ejemplo, tantas veces como profesores tengamos en ese grado.

En las siguientes tablas se muestran ciertos operadores que pueden resultar útiles en determinados casos. Estos operadores serán utilizados después de la cláusula Where y pueden ser combinados hábilmente mediante paréntesis para optimizar nuestra selección a muy altos niveles. En las tablas número 2.3 y 2.4 muestran algunos operadores empleados para optimizar selecciones, al igual que algunos comodines que facilitan las acciones (*Tabla 2.5*)

Tabla 2.3. Operadores matemáticos

Operadores matemáticos:	
>	Mayor que
<	Menor que
>=	Mayor o igual que
<=	Menor o igual que
<>	Distinto
=	Igual

Tabla 2.4. Operadores Lógicos

Operadores lógicos
AND
OR
NOT

Tabla 2.5. Comodines

Comodines
* Sustituye a todos los campos
% Sustituye a cualquier cosa o nada dentro de una cadena
- Sustituye un solo carácter dentro de una cadena

Gracias a estos operadores se pueden establecer patrones de búsqueda que permiten a la vez facilitar las búsquedas de información ya que se establecen condiciones para lo que estamos buscando.

2.3 El lenguaje SQL en el SGBD MySQL

MySQL es un sistema de administración de bases de datos relacional (RDBMS)[9]. Se trata de un programa capaz de almacenar una enorme cantidad de datos de gran variedad y de distribuirlos para cubrir las necesidades de cualquier tipo de organización, desde pequeños establecimientos comerciales a grandes empresas y organismos administrativos. MySQL compite con sistemas RDBMS propietarios conocidos, como Oracle, SQL Server y DB2. Es un servidor de bases de datos

relacionales muy rápido, multiusuario y multihilo, usado sobre todo en Internet en conjunción con PHP.

MySQL incluye todos los elementos necesarios para instalar el programa, preparar diferentes niveles de acceso de usuarios, administrar el sistema y proteger y hacer volcados de datos. Puede desarrollar sus propias aplicaciones de base de datos en la mayor parte de los lenguajes de programación utilizados en la actualidad y ejecutarlos en casi todos los sistemas operativos.

MySQL utiliza el lenguaje de consulta estructurado (SQL). Se trata del lenguaje utilizado por todas las bases de datos relacionales. Este lenguaje permite crear bases de datos, así como agregar, manipular y recuperar datos en función de criterios específicos [9].

2.3.1 Conexión a una base se datos

El equipo en el que se ejecuta MySQL y que almacena los datos se denomina *servidor MySQL*. Para establecer una conexión a este servidor, dispone de varias opciones de instalación.

1. En primer lugar se puede instalar el cliente y el servidor MySQL en su equipo de escritorio.
2. En segundo lugar, puede instalar el cliente MySQL en un equipo de sobremesa y el servidor MySQL en otro equipo al que se establecerá la conexión.

3. Por último, su equipo de sobremesa puede ser cualquier ordenador que se conecte a otro equipo con un cliente MySQL instalado, que a su vez se conectará al servidor MySQL, situado en el mismo equipo o en otro.

2.4 Maguma Studio

Maguma Studio es un IDE (Integrated Development Environment) (ambiente integrado del desarrollo) enfocado al PHP y a los lenguajes de programación de códigos abierto del lado del servidor. Maguma Studio es para ambiente Windows con un diseño XP que incluye complementación de la sintaxis de código HTML y PHP [14].

Maguma es algo más que un simple editor de programación; Es una completa herramienta integrada de desarrollo enfocado a PHP (pero con soporte para más lenguajes) que proporciona a programadores y desarrolladores un entorno de trabajo atractivo y funcional.

El programa facilita al usuario su tarea mediante diversas funcionalidades tales como código de colores para comandos de sintaxis, plantillas para diversos lenguajes de programación auto completado de código, librería de snippets, palabras clave PHP, tags de HTML, organizador de proyectos y mucho más.

2.4.1 Características de Maguma

- Completación de Sintaxis en HTML, PHP.
- Manejador de Archivos, Proyectos, etc.

- Previsualización Interna y Externa con capacidad de correr Scripts PHP.
- Debug.
- Capacidad de usar un Wizard.
- Capacidad de FTP.

Tiene la capacidad de funciones así como la posibilidad de correr archivos PHPs sin la necesidad de tener instalado Apache u otro servidor Web, solo se necesita configurar la ruta del php.exe; además, crea una lista de los includes que llama el archivo PHP en edición.

CAPÍTULO

3

**DESCRIPCIÓN DEL SISTEMA DE
INFORMACIÓN**

CAPÍTULO III

DESCRIPCIÓN DEL SISTEMA DE INFORMACIÓN

En este capítulo se expone el proceso de creación de los modelos. El método empleado contempla: la determinación de las entidades, los pasos para eliminar las entidades no específicas, la realización del primer modelo IDEF, sus modificaciones, hasta terminar con el modelo final. El modelo IDEF1x desarrollado es posteriormente programado en un motor de base de datos específico.

3.1 Construcción del modelo IDEF0.

Uno de los aspectos de IDEF0 más importante es que como concepto de modelación va introduciendo gradualmente más y más niveles de detalle a través de la estructura del modelo. De esta manera, la comunicación se produce dando al lector un tema bien definido con una cantidad de información detallada disponible para profundizar en el modelo.

3.1.1 Diagramas y componentes

Los diagramas utilizados en IDEF0, comúnmente llamados ICOM'S o diagramas A-0 son utilizados para el modelado de decisiones, acciones y actividades de una organización, cuentan con una serie de componentes elementales, representados en la *Figura 3.1* y los cuales se enlistan a continuación:

- ❖ Función 0
- ❖ Entrada (input)
- ❖ Control

- ❖ Salida (output)
- ❖ Mecanismo
- ❖ Llamada

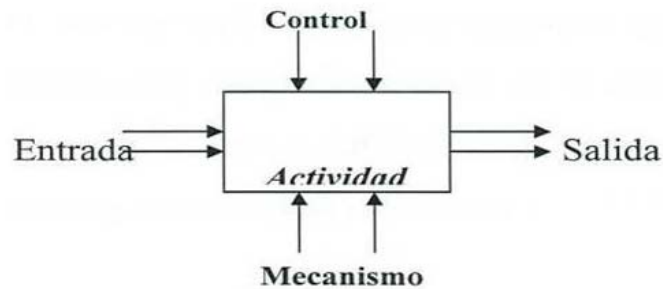


Figura 3.1. Elementos básicos del ICOM (A-n)

3.1.2 Reglas de sintaxis de los diagramas.

Para la construcción de un ICOM se tiene una serie de reglas de sintaxis que deberán acatarse con el fin de llevar un control sobre la construcción de los mismos y la identificación de sus componentes.

1. Los diagramas contextos deben tener numero de nodo A-n, donde n es igual o mayor a cero.
2. El modelo debe tener un diagrama de contexto A-0 que contenga solo una caja.
3. El número de caja de la caja única del diagrama de contexto A-0 debe ser 0.
4. Un diagrama que no sea el de contexto debe tener entre 3 y 6 cajas.
5. Cada caja de un diagrama que no sea de contexto debe enumerarse en su esquina inferior derecha desde 1 hasta 6.

6. Cada caja que ha sido detallada debe tener la expresión de la referencia detallada de su diagrama hijo escrito bajo la esquina inferior derecha de la caja.
7. Las flechas deben dibujarse con trazos horizontales y verticales, nunca diagonales.
8. Cada caja debe de tener un mínimo de una flecha de control y una flecha de output.
9. Una caja puede tener cero o más flechas de input.
10. Una caja puede tener cero o más flechas de no llamada de mecanismos.
11. Una caja puede tener 0 ó 1 flechas de llamada.
12. El extremo no conectado de las flechas de límite deberá tener un código ICOM propio que especifique su conexión a la caja padre (parental) en caso que no se encuentre secuencia.
13. Las flechas de límite como final abierto que representan el mismo dato u objeto deben conectarse mediante flechas ramificadas a todas las zonas afectadas, a menos que esto haga que el diagrama sea incomprensible.
14. Los nombres de flechas y cajas no deben consistir únicamente en palabras tales como: función, actividad, proceso, entrada, salida, control o mecanismo.

3.1.3 Operaciones en cadena

Algunas funciones en un modelo pueden ser desarrolladas en cadena si las condiciones necesarias se han satisfecho. El output de una caja puede proveer algunos o todos los datos y objetos necesarios para la activación de una o varias cajas.

Cuando la salida (output) de una caja proporciona parte o todas las entradas (input), controles o mecanismos necesarios para otra caja, la activación de la caja última dependerá del desarrollo secuencial. Sin embargo, distintas activaciones de la misma caja con distintos requisitos, pueden operar en cadena.

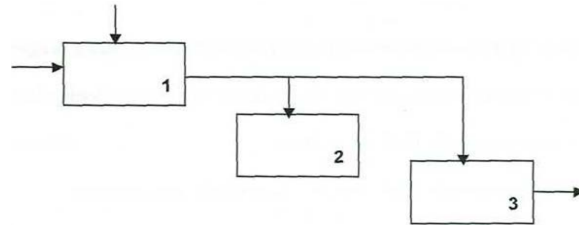


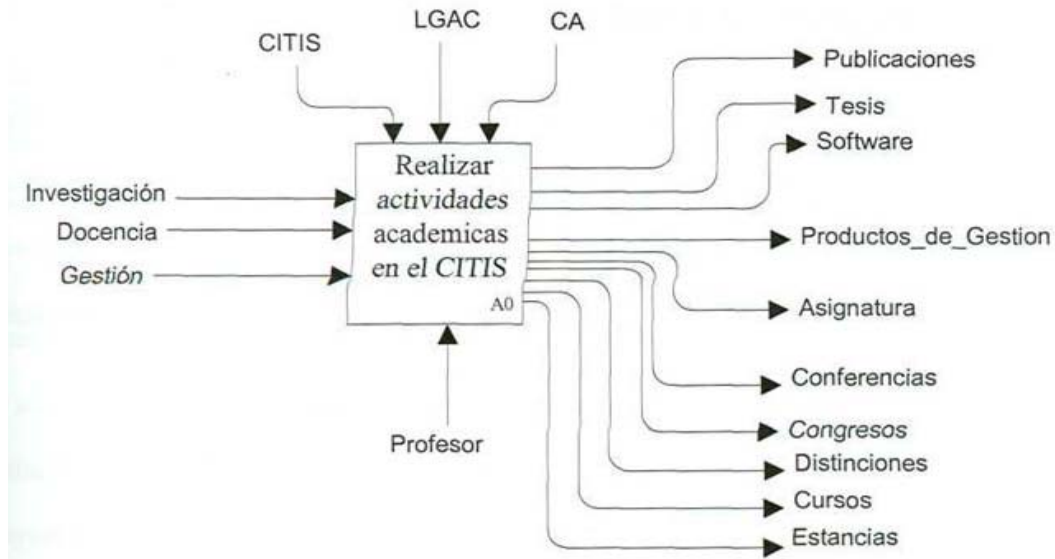
Figura 3.2. Operación en cadena

Una vez que el dato u objeto ha sido provisto, las funciones 2 y 3 pueden operar en cadena.

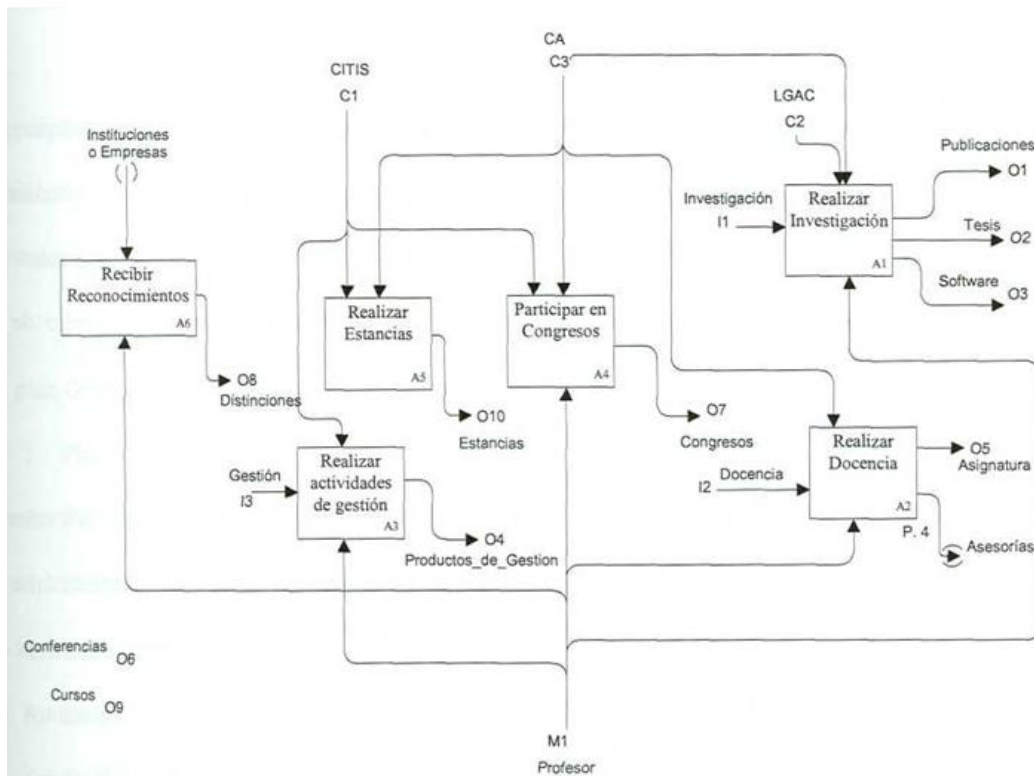
Los párrafos anteriores ofrecen una breve descripción de IDEF0. Sin embargo IDEF0 es utilizado en la realización de un análisis funcional, que realiza el dominio experto envolvente y los consensos de toma de decisiones a través de recursos gráficos simplificados, además de que como concepto de modelización va introduciendo gradualmente más y más niveles de detalle a través de la estructura del modelo. Se puede encontrar más información sobre IDEF0 en [12].

Como podemos darnos cuenta en el siguiente diagrama IDEF0 que nos muestra el producto de las actividades realizadas dentro del CITIS, como son la investigación, docencia, gestión, estas actividades son realizadas por los profesores, a través de los Cuerpos Académicos y las LGAC que estos conforman. Ver *Figura 3.3.*

El diagrama de procesos detallados, como lo dice su nombre, detalla más a fondo las actividades realizadas por los profesores y así mismo los productos de estas actividades realizadas. *Figura 3.4.*



*Figura 3.3.*Diagrama de Procesos de Alto Nivel.



*Figura 3.43.*Diagrama de Procesos Detallados.

3.2 Construcción del modelo IDEF1x.

El modelo de datos debe ser apoyado por la documentación apropiada, la cual sirve como un registro en donde se plasma la justificación de ciertos esquemas así como la definición de todos los atributos, entidades y relaciones, logrando con ello un entendimiento del uso del modelo final. El modelo IDEF1x resultante para el sistema de información se construyó siguiendo las etapas descritas por Andrew Kusiak [6].

3.2.1 Etapa cero.

El modelo IDEF1x debe ser definido en términos de sus metas y limitaciones. Estos objetivos incluyen, para el sistema propuesto, lo siguiente:

Definición del proyecto: Diseño, desarrollo e implantación de un sistema de información que consiste en una base de datos que permitirá a los profesores del CITIS agregar y modificar todo referente a sus LCAC, cuerpos académicos a los que pertenecen, las proyectos de investigación que tienen a cargo, entre otras cosas más, darlos a conocer por medio de la Internet.

Material fuente: Se analizó el modelo IDEF0 e IDEF1x que sirven para posteriormente interpolar los modelos a un motor cualquiera de una base de datos, para que posteriormente se programe con lo que es Apache, html y PHP con la finalidad de que sea observada a través de Internet. Para realizar todos estos pasos ha sido necesario analizar los documentos de PROMEP, así como hacerles

un análisis detallado junto con el director del centro, el coordinador y algunos profesores más.

Convenciones de autor: Se asume que todas las entidades deberán ser escritas en mayúsculas con tipo de letra Arial del número 12, todos los atributos con el mismo tipo y tamaño de letra en minúsculas, las claves primarias serán resaltadas con negritas, las fechas serán escritas en el orden de (día / mes / año), y la numeración será con números arábigos.

Los miembros del equipo IDEF1x: Son; el jefe del proyecto; que es quien lo revisa y aprueba, y un comité de revisión que cumplirá el rol de modeladores y expertos en el tema.

El esfuerzo de modelado comienza con una sentencia sobre el propósito y alcance del modelo incluyendo también el alcance en tiempo, es decir, si es un modelo del sistema actual o de un sistema futuro. El alcance debe indicar los datos de interés, delimitando los datos que aporten uno ó más usuarios o actores.

El propósito de nuestro modelo IDEF1x es incorporar todas las actividades relacionadas para la transformación de conocimiento acerca del sistema en un modelo que describa los procesos llevados a cabo dentro de nuestro sistema de información.

El plan del modelo consiste en los siguientes pasos:

1. Planificación del proyecto.
2. Recolección de datos.

3. Definición de entidades.
4. Definición de relaciones.
5. Definición de atributos clave.
6. Definición de atributos no clave.
7. Validación del modelo
8. Revisión y aceptación.

3.2.2 Etapa 1.

La etapa 1 consiste en la determinación de las entidades a ser modeladas.

La *Tabla 3.1* muestra la lista inicial de entidades.

Tabla 3.1. Determinación de las entidades iniciales a ser modeladas

Número	Nombre de la entidad
E-1	Profesor
E-2	Cuerpo-académico
E-3	LGAC
E-4	Actividad-académica
E-5	Tutoría
E-6	Curso-actualización
E-7	Estancia
E-8	Investigación
E-9	Gestión
E-10	Docencia
E-11	Asesorías
E-12	Asignatura
E-13	Alumno
E-14	Proyecto_investigación
E-15	Producto_investigación
E-16	Artículo
E-17	Tesis
E-18	Libro
E-19	Congreso
E-20	Conferencia

Otro componente de la etapa 1 es el glosario de entidades. Este glosario incluye:

Nombre de la entidad.- Este nombre debe ser descriptivo y con significado.

Sinónimos de la entidad.- Esto debe representar la misma definición y el nombre, pero con otra palabra.

Definición de la entidad: Se debe escribir el significado de la entidad de acuerdo a lo que el modelo desea reflejar.

Estos 3 conceptos se deben agrupar en una tabla como la que se muestra a continuación.

Tabla 3.2. Glosario de entidades

Nombre de la entidad	Sinónimo	Definición de entidades
Profesor	Investigador	Es aquel profesor de la U.A.E.H. que desarrolla proyectos de investigación de acuerdo a las diferentes líneas de investigación que se manejen.
Cuerpo-académico	Grupo de investigación y docencia	Son grupos de profesores de tiempo completo que comparten una o varias líneas de generación en temas disciplinares o multidisciplinares y un conjunto de objetivos y metas académicas.
LGAC	Línea de investigación	Campo de estudio o área específica de estudio y desarrollo de proyectos.
Actividad-académica	Responsabilidad académica	Cualquier tarea que tenga que ver con la generación (investigación), aplicación (resolución de problemas), y difusión del conocimiento.
Tutoría	Guía para el estudiante	Se presta individualmente a todos los estudiantes desde su incorporación a la Universidad, según la necesidad que cada estudiante tenga de ella.
Curso-actualización	Curso de actualización continúa	Curso de formación continúa.
Estancia	Visita académica	Periodo en el que un PTC realiza trabajo de investigación en una institución de educación superior externa.

Investigación	Generación del conocimiento	Es la búsqueda, desarrollo y orden de un tema específico.
Gestión	Actividades de administración o dirección	Definir los modelos y los instrumentos de enseñanza-aprendizaje más adecuados para cada objetivo educativo.
Docencia	Enseñanza	Asesoría especializada sobre los cursos que se imparten.
Asesorías	Enseñanza individualizada	Son clases que se les imparten a los alumnos que lo soliciten sobre una asignatura en especial.
Asignatura	Materia	Son las materias que se les imparten a los alumnos en cada semestre.
Alumno	Estudiante	Es el sujeto inscrito en alguno de los programas de estudio de la U.A.E.H. que ingresa al sistema desde la red universitaria o de cualquier equipo conectado a Internet, el cual mediante previa identificación obtendrá información.
Proyecto_investigación	Proyecto de generación y aplicación del conocimiento	Es una investigación académica, acerca de un tema en específico.
Producto_investigación	Entregable	Es el resultado final de una investigación.
Artículo	Escrito en revista	Publicación que aparece en revistas científicas de alto impacto.
Tesis	Trabajo final	Documento que todo alumno debe elaborar para obtener su título o grado correspondiente.
Libro	Libro	Publicación impresa elaborada bajo el auspicio y responsabilidad de compañías editoriales.
Congreso	Evento científico	Son una serie de conferencias académicas acerca de temas específicos.
Conferencia	Ponencia	Es una ponencia académica que se les imparte tanto a alumnos, como profesores, acerca de un tema específico.

3.2.3 Etapa 2.

La etapa 2 implica la identificación y definición de las relaciones entre entidades.

Por tratarse de etapas iniciales algunas relaciones pueden ser no específicas y tendrán que ser refinadas.

Tabla 3.3. Matriz de identificación de relaciones.

	Profesor	Cuerpo_académico	LGAC	Actividad_académica	Tutoría	Curso_actualización	Estancia	Investigación	Gestión	Docencia	Asesorías	Asignatura	Alumno	Proyecto_investigación	Producto_investigación	Artículo	Tesis	Libro	Congreso	Conferencia
Profesor		X	X	X																
Cuerpo_académico	X		X																	
LGAC	X	X																		
Actividad_académica	X				X	X	X	X	X	X	X									
Tutoría				X																
Curso_actualización				X																
Estancia				X																
Investigación				X										X						
Gestión				X																
Docencia				X								X								
Asesorías				X								X	X				X			
Asignatura										X	X		X							
Alumno											X	X		X						
Proyecto_investigación								X					X		X					
Producto_investigación														X		X	X	X	X	X
Artículo															X					
Tesis													X		X					
Libro															X					
Congreso															X					
Conferencia															X					

3.2.3.1 Identificación de relaciones.

La identificación de relaciones entre entidades se lleva a cabo mediante una matriz como la que se muestra en la *Tabla 3.3*. Una X colocada en intersección significa que existe una relación entre las entidades en que se intersectan.

3.2.3.2 Definición de relaciones.

El siguiente paso es definir las relaciones que se acaban de determinar. Estas relaciones se deben examinar en ambas direcciones para decidir si la relación es específica o no específica. Por ejemplo una LGAC puede ser manejada por uno o más profesores. ¿Puede un profesor en particular manejar una o más LGAC? Por lo tanto, ésta es una relación no específica, es decir, una relación muchos a muchos. Sin embargo un profesor puede pertenecer a un sólo cuerpo académico. En este caso la relación es específica debido a que en un lado se tiene cardinalidad de 1.

Una vez que las dependencias se han determinado, el modelador debe darle un nombre a la relación y definir su significado para las entidades involucradas, por ejemplo, la relación LGAC/Profesor se puede llamar “cultiva” de modo que se lea “Un profesor puede cultivar una o más LGAC”. Las relaciones no específicas tienen dos nombres separados cada uno con una diagonal. Los nombres de las relaciones deben ser específicos, concisos y con un significado claro. La *Tabla 3.4* muestra los resultados de este paso.

Tabla 3.4. Definición de relaciones entre entidades.

Entidad	Específica/ No específica	Nombre de la relación	Definición de la relación
Profesor/Cuerpo_académico	E	Pertenece a	Un profesor pertenece a un cuerpo académico
Profesor/LGAC	NE	Cultiva	Un profesor cultiva una o varias LGAC.
Profesor/Actividad_académica	E	Realiza	Un profesor realiza una o varias actividades académicas.
Cuerpo_académico/Profesor	E	Se compone de	Un cuerpo académico se compone de uno o varios profesores.
Cuerpo_académico/LGAC	E	comparten	Un cuerpo académico comparte una o varias LGAC.
LGAC/Profesor	NE	Es cultivada por	Una LGAC es cultivada por uno o varios profesores.
LGAC/Cuerpo_académico	E	Definen	Una o varias LGAC se definen por un cuerpo académico
Actividad_académica/Profesor	E	Son ejecutadas	Una o varias actividades académicas son ejecutadas por uno o varios profesores
Actividad_académica/Tutoría	E	Es una	Una tutoría es una actividad académica
Actividad_académica/Investigación	E	Es una	Una investigación es una actividad académica
Actividad_académica/Curso_actualización	E	Es una	Un curso de actualización es una actividad académica.
Actividad_académica/Estancia	E	Es una	Una estancia es una actividad académica.
Actividad_académica/Gestión	E	Es una	Una gestión es una actividad académica.
Actividad_académica/Docencia	E	Es una	La docencia es una actividad académica.
Actividad_académica/Asesorías	E	Es una	Una asesoría es una actividad académica.
Docencia/Asignatura	E	Tiene asignada	La actividad de docencia consiste en impartir una o varias asignaturas.

Asignatura/Docencia	E	Pertenece a	Una o varias asignaturas pertenece a la actividad de docencia.
Asignatura/Alumno	NE	Se le imparte	Una o varias asignaturas se le imparte a uno o varios alumnos.
Asignatura/Asesorías	E	Es apoyada por	Una asignatura es apoyada por una o varias asesorías.
Asesorías/Alumno	NE	Se le imparte	Una o varias asesorías se le imparten a uno o varios alumnos.
Asesoría/Asignatura	E	Se apoya	Una o varias asesorías es apoyada por una o varias asignaturas.
Asesorías/Tesis	E	Apoya	Una o varias asesorías apoya a una tesis
Alumno/Asesorías	NE	Recibe	Uno o varios alumnos reciben una o varias asesorías
Alumno/Proyecto_investigación	NE	Participa en	Uno o varios alumnos participan en un proyecto de investigación.
Alumno/Asignatura	NE	Toma	Uno o varios alumnos toman una o varias asignaturas.
Investigación/Proyecto_investigación	E	Se realiza por medio de	Una investigación se realiza por medio de uno o varios proyectos de investigación.
Proyecto_investigación/Alumno	NE	Acepta	Un proyecto de investigación acepta uno o más alumnos.
Proyecto_investigación/Investigación	E	Establece	Un proyecto de investigación establece una o más investigaciones.
Proyecto_investigación/Producto_investigación	E	Se realiza con	Un proyecto de investigación se realiza con uno o más productos de investigación.
Producto_investigación/Artículo	E	Es un	Un artículo es un producto de investigación.
Producto_investigación/Tesis	E	Es un	Una tesis es un producto de investigación.
Producto_investigación/Libro	E	Es un	Un libro es un producto de investigación.

3.2.3.3 Construcción del diagrama inicial.

Se deben dibujar todas las entidades en forma de cajas cuadradas, es preferente tener todas las entidades en un diagrama que tener muchas entidades. El ejemplo del diagrama en esta etapa se muestra en la *Figura 3.5*.

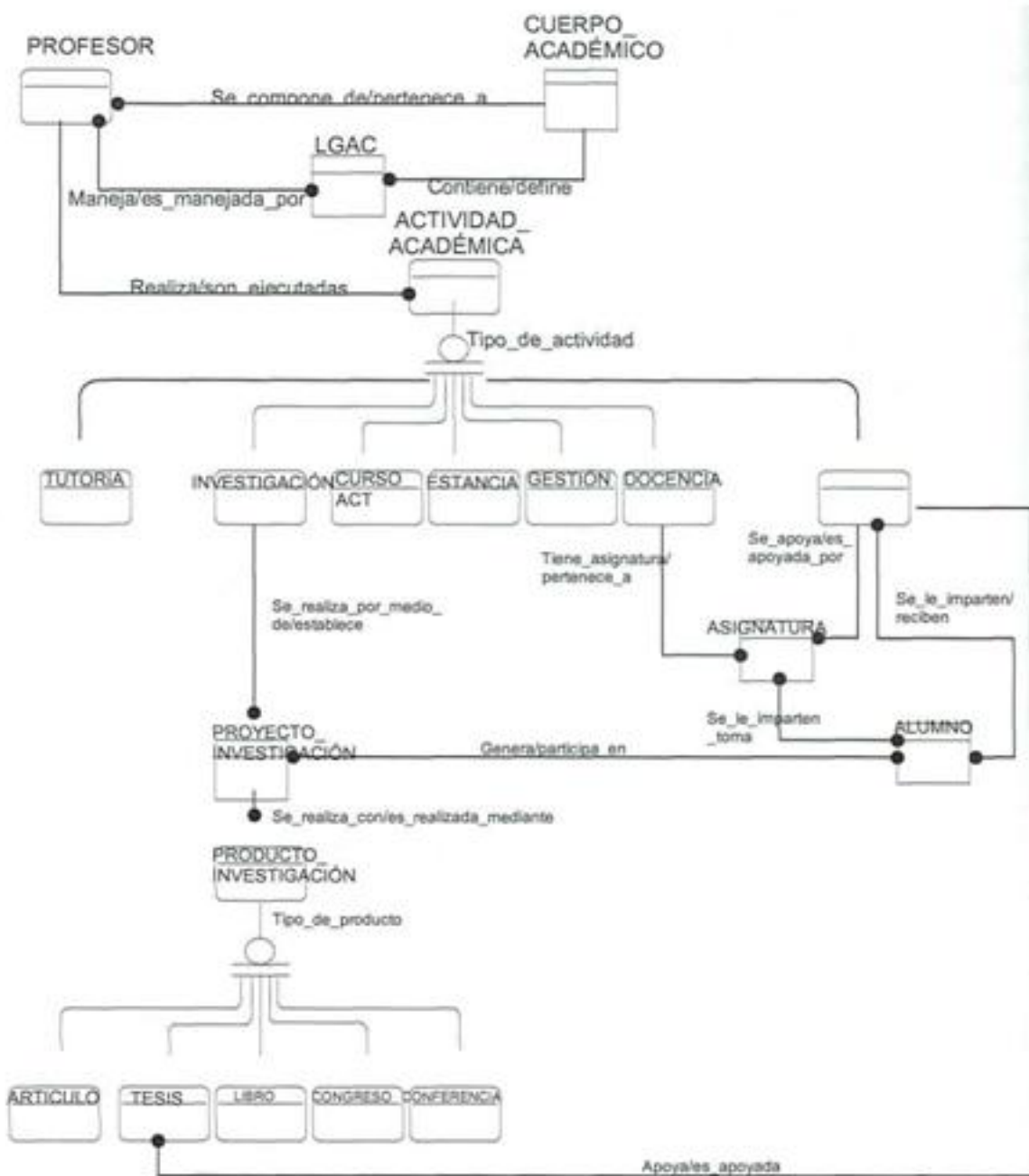


Figura 3.5. Construcción del diagrama inicial IDEF1x

3.2.4 Etapa 3

En la etapa 3 las relaciones no específicas deben de ser refinadas, se deben definir también los atributos clave de cada entidad, se migran las claves primarias para formar claves externas en las entidades hijo, por último las relaciones y las claves se validan.

3.2.4.1 Resolución de relaciones no específicas.

La etapa 3 requiere que solamente se usen relaciones padre e hijo o de categorización. Las relaciones no específicas se convierten en 2 relaciones específicas creando para ello una nueva entidad. En la *Figura 3.2* se denota que una LGAC puede ser manejada por uno o más profesores y que un profesor puede manejar una o más LGAC. Después de la transformación cada miembro de la entidad auxiliar LGAC-Profesor se relaciona exactamente con 1 miembro de las entidades LGAC y Profesor.

Esta entidad auxiliar se conoce como entidad asociativa. La entidad asociativa resuelve las relaciones no específicas al contener un subconjunto del producto cruz de las dos entidades padre.

Este proceso de refinación consiste de 4 pasos:

- El desarrollo de 1 ó más alternativas para cada relación no específica.
- La selección de una alternativa.
- La actualización de la información de la etapa 1 para incluir a las nuevas entidades resultantes.

- La actualización de la información de la etapa 2 para definir las relaciones asociadas con las nuevas entidades.

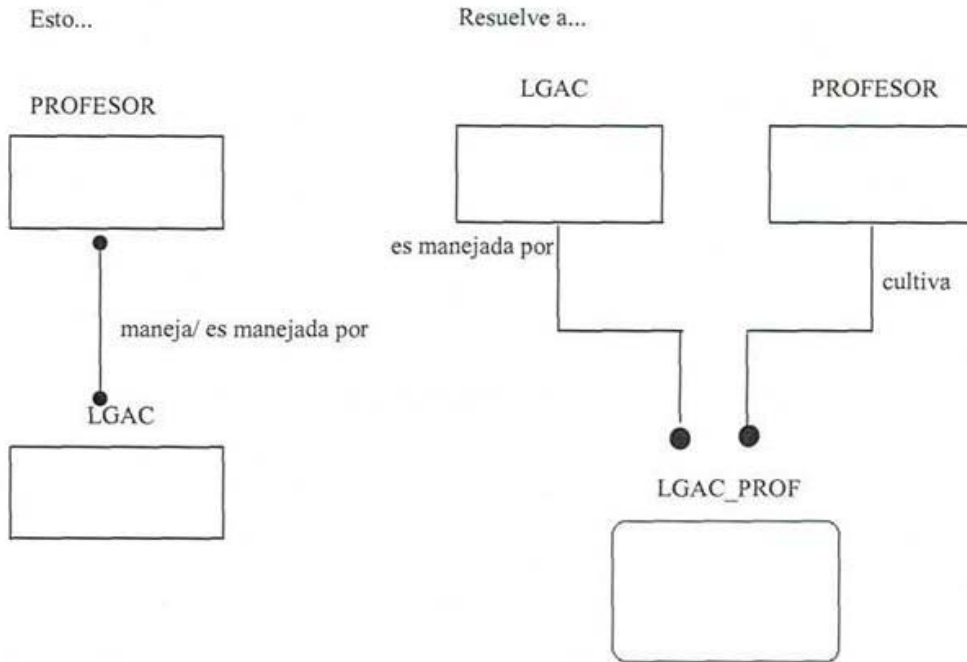


Figura 3.6a. Resolución de relaciones no específicas

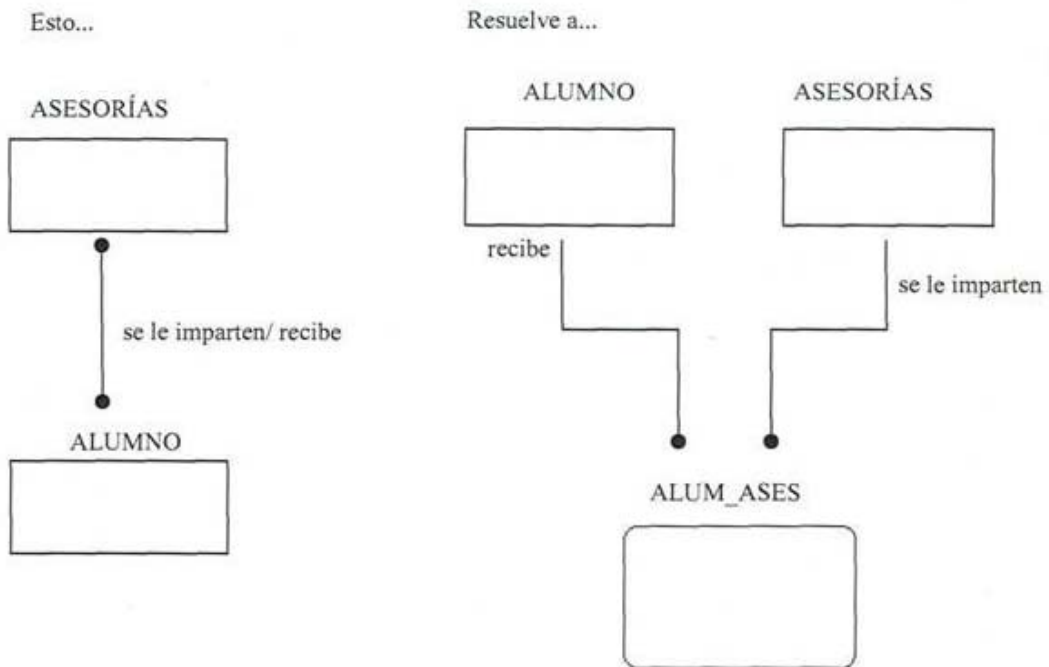


Figura 3.6b. Resolución de relaciones no específicas

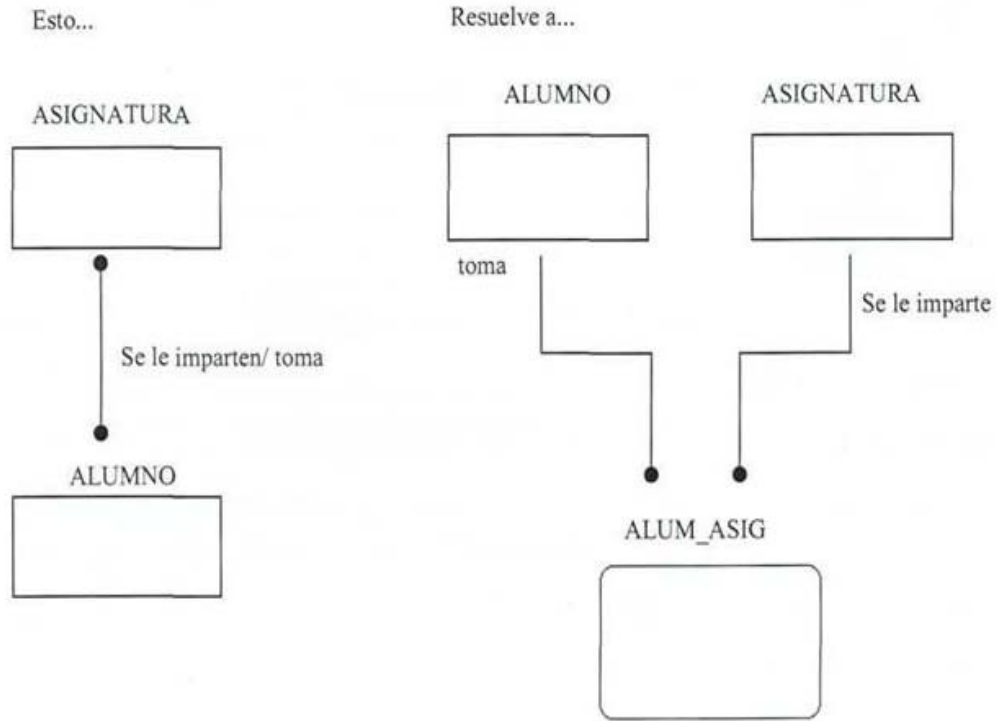


Figura 3.6c. Resolución de relaciones no específicas

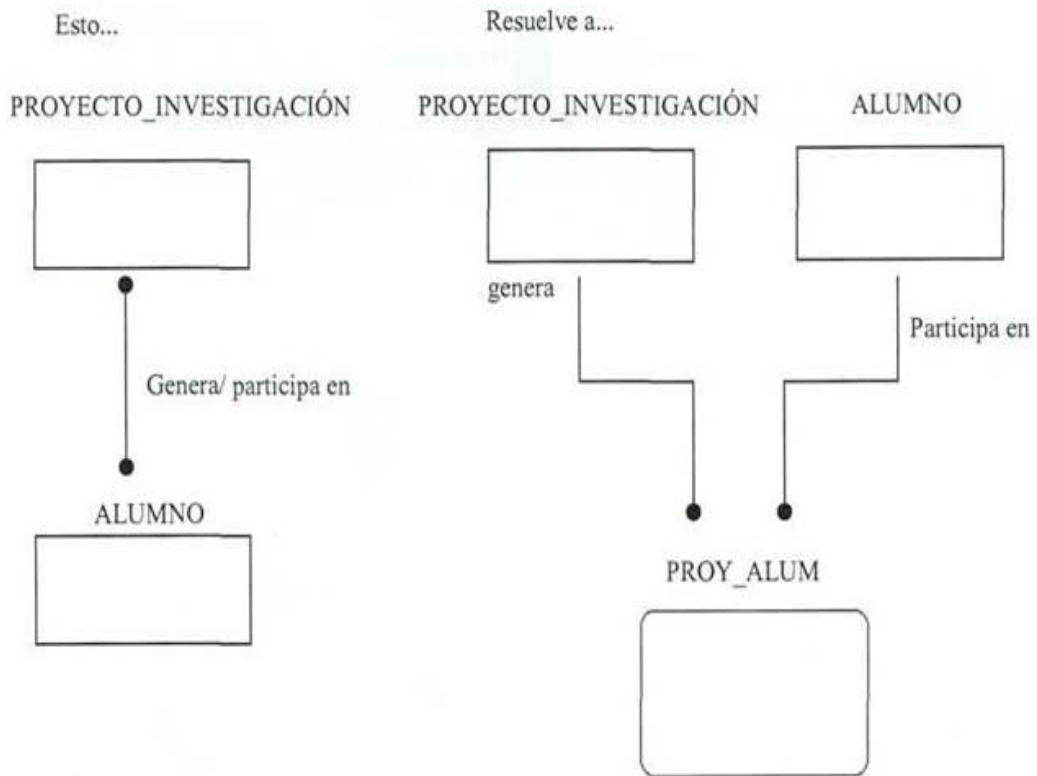


Figura 3.6d. Resolución de relaciones no específicas

Actualización de la información de la etapa 1.

Tabla 3.1.1 Actualización de la Tabla 3.1

(Determinación de las nuevas entidades a ser modeladas)

Nombre de la entidad	Sinónimo	Definición de entidades
LGAC_PROF	No hay	Es la entidad que surge de la refinación de la relación no específica entre las entidades Profesor y LGAC.
ALUM_ASES	No hay	Es la entidad que surge de la refinación de la relación no específica entre las entidades Alumno y Asesorías.
ALUM_ASIG	No hay	Es la entidad que surge de la refinación de la relación no específica entre las entidades Alumno y Asignatura.
PROY_ALUM	No hay	Es la entidad que surge de la refinación de la relación no específica entre las entidades Proyecto_ Investigación y Alumno.

Tabla 3.2.1 Actualización de la Tabla 3.2

(Glosario de las nuevas entidades)

Número	Nombre de la entidad	Número de origen
E-21	LGAC_PROF	Etapa 3 EDEF1x
E-22	ALUM_ASES	Etapa 3 EDEF1x
E-23	ALUM_ASIG	Etapa 3 EDEF1x
E-24	PROY_ALUM	Etapa 3 EDEF1x

Actualización de la información de la etapa 2.

Tabla 3.3.1 Actualización de la *Tabla 3.3*
(Matriz de identificación de relaciones).

	Profesor	Cuerpo académico	LGAC	Actividad académica	Tutoría	Curso_ actualización	Estancia	Investigación	Gestión	Docencia	Asesorías	Asignatura	Alumno	Proyecto investigación	Producto investigación	Artículo	Tesis	Libro	Congreso	Conferencia	LGAC_PROF	ALUM_ASES	ALUM_ASIG	PROY_ALUM
Profesor		X	X	X																	X			
Cuerpo académico	X		X																					
LGAC	X	X																			X			
Actividad académica	X				X	X	X	X	X	X	X													
Tutoría				X																				
Curso actualización				X																				
Estancia				X																				
Investigación				X										X										
Gestión				X																				
Docencia				X								X												
Asesorías				X								X	X				X					X		
Asignatura										X	X		X										X	
Alumno											X	X		X								X	X	X
Proyecto investigación								X					X		X									X
Producto investigación														X		X	X	X	X	X				
Artículo															X									
Tesis													X		X									
Libro															X									
Congreso															X									
Conferencia															X									
LGAC_PROF	X		X																					
ALUM_ASES											X		X											
ALUM_ASIG												X	X											
PROY_ALUM													X	X										

Tabla 3.4.1. Actualización de la Tabla 3.4
(Definición de relaciones entre entidades).

Entidad	Específica / No específica	Nombre de la relación	Definición de la relación
Profesor/LGAC_Prof	E	Da lugar	Un profesor da lugar a una relación lgac_prof; donde un profesor puede manejar una o varias LGAC
LGAC/LGAC_Prof	E	Da lugar	Una lgac da lugar a una relación lgac_prof; donde una lgac puede ser manejada por uno o varios profesores.
Alumno/Alum_Ases	E	Da lugar	Un alumno da lugar a una relación Alum_Ases, donde un alumno puede recibir una o más asesorías.
Asesorias/Alum_Ases	E	Da lugar	Una asesoría da lugar a una relación Alum_Ases, donde una asesoría puede ser impartida a uno o más alumnos.
Alumno/Alum_Asig	E	Da lugar	Un alumno da lugar a una relación Alum_Asig, donde un alumno puede tomar una o más asignaturas.
Asignatura/Alum_Asig	E	Da lugar	Una asignatura da lugar a una relación Alum_Asig, donde una asignatura puede ser impartida a uno o más alumnos.
Proyecto_Investigación/Proy_Alum	E	Da lugar	Un proyecto de investigación da lugar a una relación Proy_Alum; donde un proyecto de investigación puede generar una o varios proyectos para uno o más alumnos.
Alumno/Proy_Alum	E	Da lugar	Un alumno da lugar a una relación Proy_Alum; donde un alumno puede participar en un proyecto de investigación.

3.2.4.2 Vistas de función

En las etapas iniciales del proceso de modelado puede ser posible dibujar todas las relaciones en un solo diagrama. Cuando éste crece es necesario dividirlo en subdiagramas llamados vistas de función. La vista se restringe a una cantidad que pueda ser vista, validada y entendida. En nuestro caso particular no hay vistas de función.

3.2.4.3 Identificación del atributo clave.

En este paso de la etapa 3 se identifica el o los atributos de cada entidad que sirven para identificar de un modo único cada instancia de la entidad. Este atributo es conocido como la clave primaria de la entidad. Las claves primarias potenciales se conocen como claves candidatas. Solamente las claves primarias se migran a las entidades hijo. La *Figura 3.3* muestra las claves primarias de las entidades de nuestro diagrama.

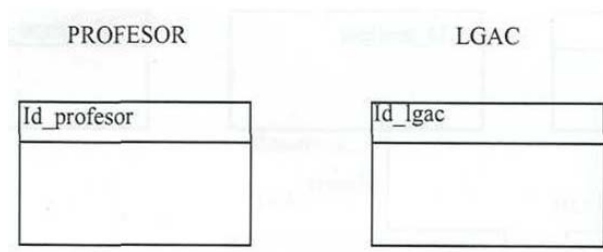


Figura 3.7. Atributos clave

3.2.4.4 Migración de claves.

La migración de claves radica en copiar la clave primaria de una entidad a otra entidad llamada comúnmente entidad hijo. La clave primaria migrada se conoce como clave externa en la entidad hijo.

Existen 3 reglas para llevar a cabo la migración de claves:

1. La migración siempre ocurre desde la entidad padre hacia la entidad hijo, o desde la entidad genérica a la entidad de categorización.
2. El o los atributos que forman la clave primaria se deben migrar en cada relación que sigue en cada relación que así lo necesite.
3. Las claves alternas y los demás atributos nunca se migran.

La *Figura 3.4* muestra algunos ejemplos de migraciones de clave. En el segundo ejemplo la clave primaria de la entidad *Cuerpo_Academico* migra hacia la entidad *Profesor*. Sin embargo debido a que las instancias de la entidad *Profesor* se pueden identificar de un modo único por su propia clave primaria, la clave externa "Id_lgac" pasa a ser un atributo heredado y no un miembro de la clave primaria.

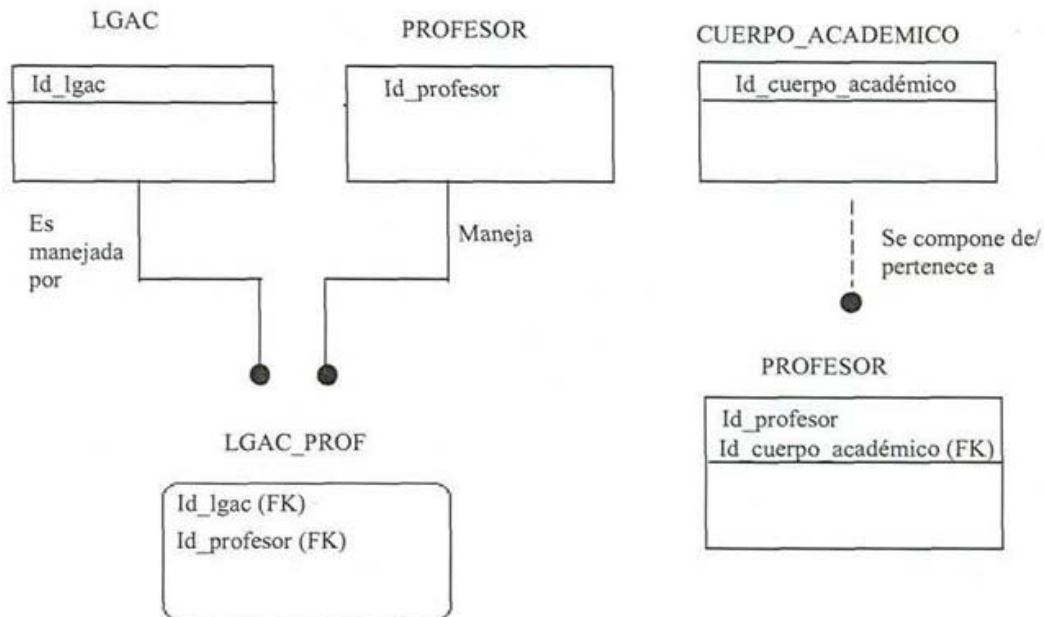


Figura 3.8. Migración de identificadores dependientes e identificadores independientes de entidades.

3.2.4.5 Validación de claves y relaciones.

Se emplean las siguientes reglas para la identificación y migración de claves:

- ❖ Está prohibido el uso de relaciones no específicas.
- ❖ Es obligatoria la migración de claves.
- ❖ Está prohibido el uso de atributos y que almacenen el mismo dato para una entidad.
- ❖ Está prohibido el uso de atributos que no tengan datos.

- ❖ Las entidades con claves compuestas no pueden ser divididas en entidades diversas con claves más pequeñas.

Considere la entidad profesor mostrada en la *Figura 3.5*. En esta forma la entidad profesor violaría la regla número 3 debido a que los atributos `id_profesor` y `id_lgac` formarían la clave primaria, esto significa que debería existir una entidad, lo cual no corresponde con la realidad. El segundo arreglo mostrado en la *Figura 3.5* representa adecuadamente la relación entre profesor y lgac.

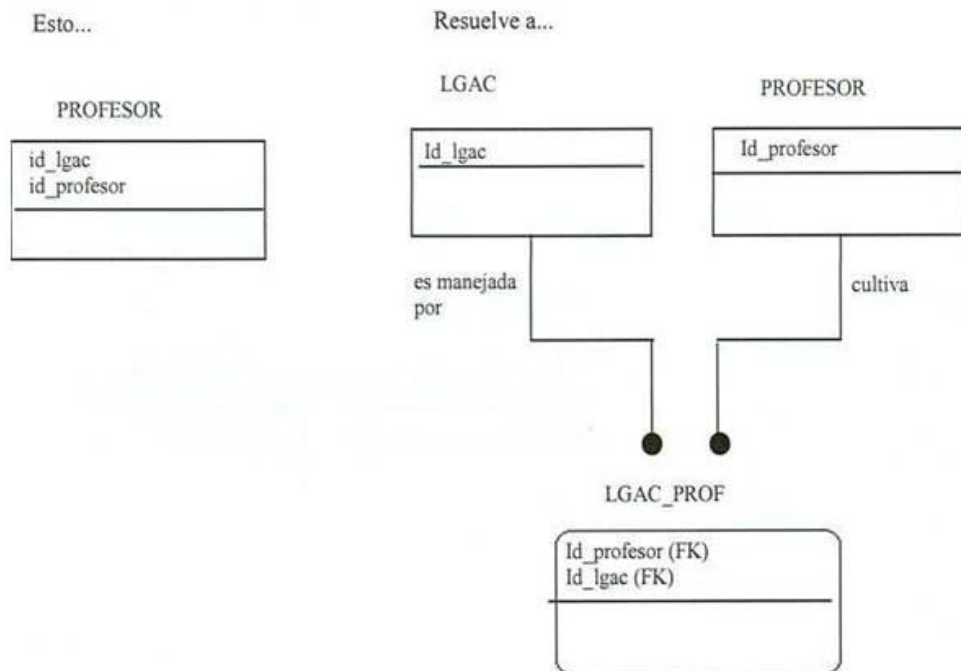


Figura 3.9a. No-repetición de claves.

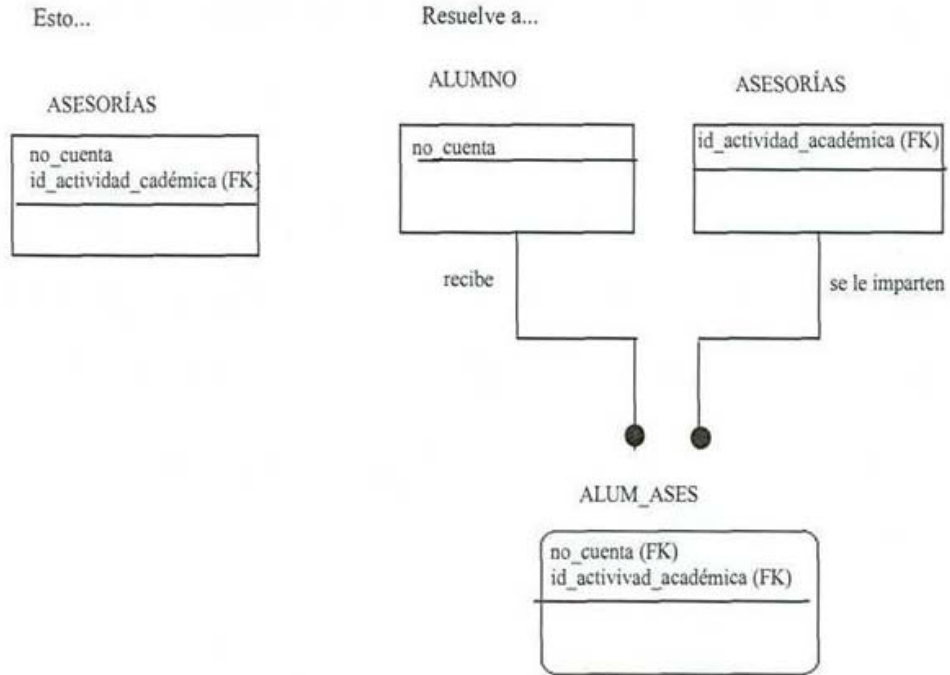


Figura 3.9b. No-repetición de claves.

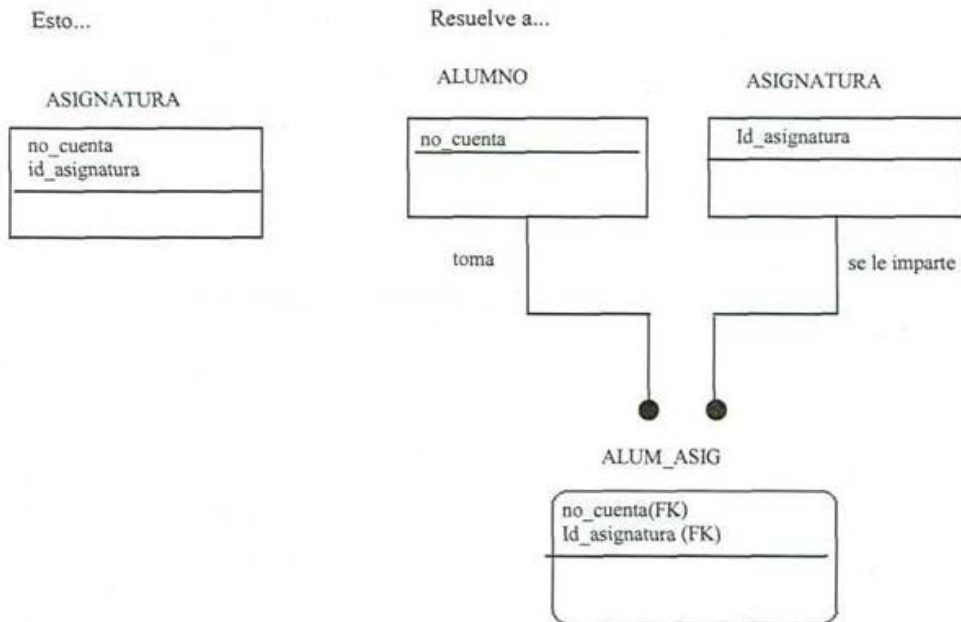


Figura 3.9c. No-repetición de claves.



Figura 3.9d. No-repetición de claves.

Enseguida se deben analizar las entidades con claves compuestas para garantizar la regla número 5. Esta regla garantiza la cuarta y quinta formas normales de Codd descritas en la teoría relacional. En nuestro caso particular estas reglas no se rompen.

Se debe analizar el esquema para las relaciones duales. Una relación dual existe cuando una entidad hijo tiene dos relaciones para una entidad padre. Cuando esto ocurre se necesita una afirmación para determinar si los caminos son iguales, desiguales o indeterminados. Si uno de los caminos consiste de una relación sencilla entonces el camino es redundante. En nuestro diagrama no contamos con relaciones duales.

Se utiliza una matriz entidad-atributo para analizar el uso de los atributos en el modelo. En esta matriz se utilizan los siguientes códigos para describir a los

atributos y a las entidades: “O” = dueño; “K” = clave primaria; “I” = heredado Un ejemplo de esta matriz se muestra en la *Tabla 3.5*.

Tabla 3.5. Matriz Entidad/ Atributo

Entidad/ Atributo	Código de LGAC	Clave del Profesor
Profesor		OK
Cuerpo-académico	IK	IK
LGAC	OK	
Actividad-académica		IK
Tutoría		IK
Curso-actualización		IK
Estancia		IK
Investigación		IK
Gestión		IK
Docencia		IK
Asesorías		IK
Asignatura		IK
Alumno		
Proyecto_investigación		IK
Producto_investigación		IK
Artículo		IK
Tesis		IK
Libro		IK
Congreso		IK
Conferencia		IK
LGAC__Prof	IK	IK
Entidad/ Atributo	Código de Alumno	Clave del Asesorías
Profesor		IK
Cuerpo-académico		
LGAC		
Actividad-académica		IK
Tutoría	IK	
Curso-actualización		
Estancia		
Investigación		
Gestión		
Docencia		

Asesorías		OK
Asignatura	IK	IK
Alumno	OK	
Proyecto_investigación		
Producto_investigación	IK	
Artículo		
Tesis	IK	IK
Libro		
Congreso		
Conferencia		
Alum_Ases	IK	IK
Entidad/ Atributo	Código de Alumno	Clave del Asignatura
Profesor		IK
Cuerpo-académico		
LGAC		
Actividad-académica		IK
Tutoría	IK	
Curso-actualización		
Estancia		
Investigación		
Gestión		
Docencia		
Asesorías	IK	
Asignatura		OK
Alumno	OK	
Proyecto_investigación		
Producto_investigación	IK	
Artículo		
Tesis	IK	
Libro		
Congreso		
Conferencia		
Alum_Asig	IK	IK
Entidad/ Atributo	Código de Proyecto_Investigación	Clave del Alumno
Profesor	IK	

Cuerpo-académico	IK	
LGAC		
Actividad-académica	IK	
Tutoría		IK
Curso-actualización		
Estancia		
Investigación	IK	
Gestión		
Docencia		
Asesorías		
Asignatura		
Alumno		OK
Proyecto_investigación	OK	
Producto_investigación	IK	IK
Artículo	IK	
Tesis	IK	IK
Libro	IK	
Congreso	IK	
Conferencia	IK	
Pory_Alum	IK	IK

Códigos:

OK: Owner of the Key - Dueño de la Clae

IK: Inherited key - Clave heredada

3.2.4.6 Definición del atributo clave

El mismo procedimiento que se empleó para la definición de entidades se utiliza para la definición de claves, es decir, se debe definir el nombre del atributo, la definición del atributo y los sinónimos para el atributo clave. Los atributos se asocian con la entidad a la que pertenecen.

3.2.4.7 Resultados de la etapa 3.

Las vistas de función de la etapa 3 deben contener:

- Atributos clave, claves alternas y claves externas.
- Entidades independientes y entidades dependientes.
- Relaciones de identificación y de no-identificación.

La documentación que debe contener la información del modelo incluye:

1. La definición de las entidades. Ver Tabla 3.6.
2. La lista de las claves primarias, claves alternas y claves externas. Ver Tabla 3.6.
3. La lista de relaciones para las cuales la entidad es una entidad de categorización
4. La lista de relaciones de identificación para las cuales la entidad actúa como padre. Ver Tabla 3.7.
5. La lista de las entidades de identificación para las cuales la entidad actúa como hijo. Ver Tabla 3.8.
6. La lista de las relaciones de no-identificación para las cuales la entidad actúa como padre. Ver Tabla 3.9.
7. La lista de las entidades de no-identificación para las cuales la entidad actúa como hijo. Ver Tabla 3.10.

La documentación correspondiente a la información del modelo IDEF1x al final de la etapa 3 se presenta en las tablas siguientes:

Tabla 3.6. Documentación correspondiente a la etapa 3.

Nombre de la entidad	Definición de la entidad	Clave Primaria (PK)	Clave alterna (AK)	Clave externa (FK)
Profesor	Es aquel profesor de la U.A.E.H. que desarrolla proyectos de investigación de acuerdo a las diferentes líneas de investigación que se manejen.	id_profesor	/	id_cuerpo_academico
Cuerpo-académico	Son grupos de profesores de tiempo completo que comparten una o varias líneas de generación en temas disciplinares o multidisciplinarios y un conjunto de objetivos y metas académicas.	id_cuerpo_academico	/	/
LGAC	Campo de estudio o área específica de estudio y desarrollo de proyectos.	id_LGAC	/	id_cuerpo_academico
Actividad-académica	Cualquier tarea que tenga que ver con la generación (investigación), aplicación (resolución de problemas), y difusión del conocimiento.	id_actividad_academica	/	id_profesor id_cuerpo_academico
Tutoría	Se presta individualmente a todos los estudiantes desde su incorporación a la universidad, según la necesidad que cada estudiante tenga de ella.	/	/	id_profesor id_cuerpo_academico id_actividad_academica
Curso-actualización	Curso de formación continúa.	/	/	id_profesor id_cuerpo_academico id_actividad_academica
Estancia	Periodo en el que un PTC realiza trabajo de investigación en una institución de educación superior externa.	/	/	id_profesor id_cuerpo_academico id_actividad_academica
Investigación	Es la búsqueda, desarrollo y orden de un tema específico.	/	/	id_profesor id_cuerpo_academico id_actividad_academica
Gestión	Definir los modelos y los instrumentos de enseñanza-aprendizaje más adecuados para cada objetivo educativo.	/	/	id_profesor id_cuerpo_academico id_actividad_academica
Docencia	Asesoría especializada sobre los cursos que se imparten.	/	/	id_profesor id_cuerpo_academico id_actividad_academica
Asesorías	Son clases que se les imparten a los alumnos que lo soliciten sobre una asignatura en especial.	/	/	id_profesor id_cuerpo_academico id_actividad_academica
Asignatura	Son las materias que se les imparten a los alumnos en su semestre.	Id_asignatura	/	id_profesor id_cuerpo_academico id_actividad_academica

Tabla 3.6 (Continuación)

Alumno	Es el sujeto inscrito en alguno de los programas de estudio de la U.A.E.H. que ingresa al sistema desde la red universitaria o de cualquier equipo conectado a Internet, el cual mediante previa identificación obtendrá información.	no_cuenta	/	/
Proyecto_investigación	Es una investigación académica, acerca de un tema en específico.	id_proyecto_investigación	/	id_profesor id_cuerpo_academico id_actividad_academica
Producto_investigación	Es el resultado final de una investigación.	Id_producto_investigación	/	id_proyecto_investigación
Artículo	Publicación que aparece en revistas científicas de alto impacto.	/	/	id_proyecto_investigación id_producto_investigación
Tesis	Documento que todo alumno debe elaborar para obtener su título o grado correspondiente.	/	/	id_proyecto_investigación id_producto_investigación
Libro	Publicación impresa elaborada bajo el auspicio y responsabilidad de compañías editoriales.	/	/	id_proyecto_investigación id_producto_investigación
Congreso	Son una serie de conferencias académicas acerca de temas específicos.	/	/	id_proyecto_investigación id_producto_investigación
Conferencia	Es una ponencia académica que se les imparte tanto a alumnos, como a profesores, acerca de un tema específico.	/	/	id_proyecto_investigación id_producto_investigación
LGAC_PROF	Es la entidad que surge de la refinación de la relación no específica entre las entidades LGAC y Profesor.	/	/	id_profesor id_cuerpo_academico id_lgac
ALUM_ASES	Es la entidad que surge de la refinación de la relación no específica entre las entidades Alumno y Asesorías.	/	/	no_cuenta id_profesor id_acatividad_academica id_cuerpo_académico
ALUM_ASIG	Es la entidad que surge de la refinación de la relación no específica entre las entidades Alumno y Asignatura.	/	/	id_asignatura no_cuenta
PROY_ALUM	Es la entidad que surge de la refinación de la relación no específica entre las entidades Proyecto_Investigación y Alumno.	/	/	id_proyecto_investigación no_cuenta

Tabla 3.7. Lista de relaciones de identificación para las cuales la entidad actúa como padre.

Entidad Padre	Entidad Hijo
LGAC	LGAC_Profesor
Profesor	LGAC_Profesor
Alumno	Alum_Ases
Asesorías	Alum_Ases
Alumno	Alum_Asig
Asignatura	Alum_Asig
Proyecto_Investigación	Proy_Alum
Alumno	Proy_Alum
Cuerpo_Academico	Profesor
Profesor	Actividad_Académica

Tabla 3.8. Lista de entidades de identificación para las cuales la entidad actúa como hijo.

Entidad Hijo	Entidad Padre
LGAC_Profesor	LGAC
LGAC_Profesor	Profesor
Alum_Ases	Alumno
Alum_Ases	Asesorías
Alum_Asig	Alumno
Alum_Asig	Asignatura
Proy_Alum	Proyecto_Investigación
Proy_Alum	Alumno
Profesor	Cuerpo_Academico
Actividad_Académica	Profesor

Tabla 3.9. Lista de relaciones de no-identificación para las cuales la entidad actúa como padre.

Entidad Padre	Entidad Hijo
Cuerpo_Academico	LGAC

Tabla 3.10. Lista de entidades de no-identificación para las cuales la entidad actúa como hijo.

Entidad hijo	Entidad padre
LGAC	Cuerpo_Academico

La Figura 3.10. Muestra el modelo IDEF1x al final de la etapa 3.

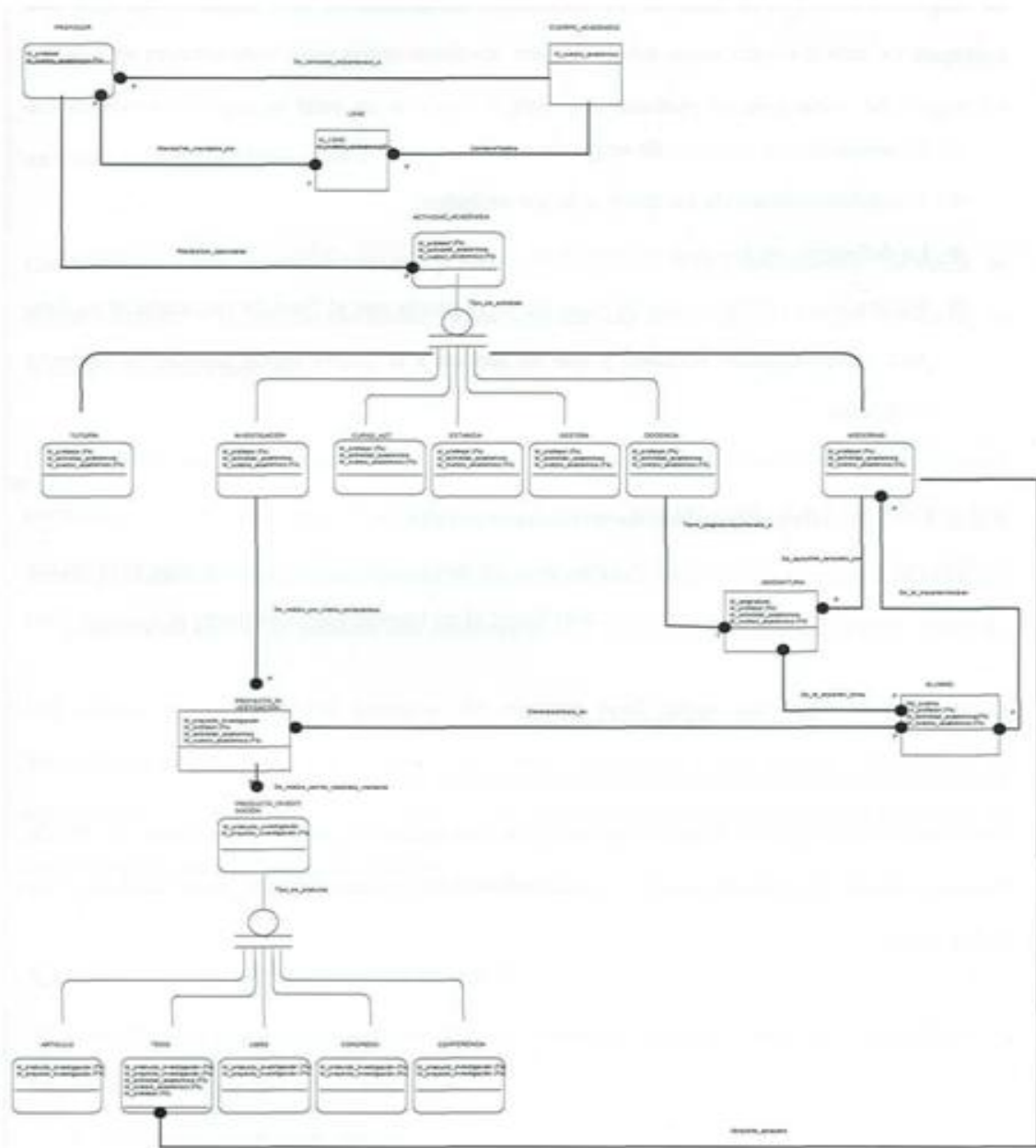


Figura 3.10. Modelo IDEF1x al final de la etapa 3

3.2.5 Etapa 4.

La etapa 4 constituye el final del procedimiento de modelado. Los objetivos en esta fase incluyen:

- ❖ El desarrollo de una lista de atributos
- ❖ El establecimiento de los dueños de los atributos
- ❖ La definición de los atributos no clave
- ❖ Validación y refinamiento del modelo. Se pretende que al final de esta etapa el modelo esté completamente refinado y que se adecúe a la quinta forma normal de la teoría relacional.

3.2.5.1 Identificación de atributos no clave.

Se enlistan los atributos y se les asigna un número de identificación único indicando la fuente de donde fueron extraídos. En la Tabla 3.11 se presenta una muestra de la lista de atributos.

3.2.5.2 Dueños de los atributos.

Cada atributo no clave debe ser contenido en una entidad.

Para los atributos que no tengan un dueño claro es necesario examinar la fuente de donde fueron extraídos después del análisis se determinará la pertinencia o no de un atributo. Ver Tabla 3.11.

3.2.5.3 Refinación del modelo.

Las reglas descritas en la etapa 3 se aplican tanto a los atributos clave como a los atributos no clave. Este proceso puede dar como resultado más entidades cuyas claves deben ser migradas del mismo modo que se hizo en la etapa 3. Para ello también es necesario checar que los atributos no tengan valores nulos.

Conforme se crean nuevas entidades y relaciones éstas se deben documentar, es decir, se deben adicionar a la lista de entidades, se les debe dar una definición, deben ser colocadas en la matriz de relaciones, etc.

Las entidades con claves compuestas deben obedecer la regla de la dependencia funcional completa, la cual indica que una entidad no clave en una relación, debe ser identificada únicamente por la clave compuesta completa y no sólo por una parte de ella. Esta regla es equivalente a la segunda forma normal en la teoría relacional.

Del mismo modo todos los atributos del modelo final deben satisfacer la regla de la dependencia no transitiva. Esta regla es equivalente a la tercera forma normal. El resultado de aplicar estas dos reglas es que un atributo no clave debe ser dependiente de su clave, de su clave entera, y nada más que de su clave

3.2.5.4. Resultados de la etapa 4

Al final de esta etapa los diagramas deben mostrar el modelo refinado y completo.

La documentación completa está formada por:

- ❖ La definición de cada entidad
- ❖ Una lista de los atributos clave, de las claves alternas y de las claves externas
- ❖ La definición para cada atributo clave y no clave.
- ❖ Para cada entidad, una lista de las relaciones en las que participa, definiendo también su papel.
- ❖ Una referencia cruzada de relaciones y entidades

La documentación correspondiente a la información del modelo IDEF1x al final de la etapa 4 está contenida en las tablas desarrolladas a lo largo de este capítulo.

Tabla 3.11. Documentación correspondiente a la etapa 3.

Número	Nombre del atributo	Dueño del atributo
1	nombre_profesor	Profesor
2	grado	Profesor
3	especialidad	Profesor
4	nombre_ca	Cuerpo_Academico
5	Estatus	Cuerpo_Academico
6	fecha_creación	Cuerpo_Academico
7	tipo_cuerpo_academico	Cuerpo_Academico
8	nombre_lgac	LGAC
9	Descripción	LGAC
10	fecha_inicio	LGAC
11	nombre_tutoría	Tutoría
12	horas_impartición	Tutoría
13	número_alumnos	Tutoría
14	lugar_impartición	Tutoría
15	especialidad_impartición	Tutoría
16	nombre_curso	Curso_Act
17	fecha_curso	Curso_Act
18	nombre_estancia	Estancia

19	lugar_estancia	Estancia
20	fecha_inicio_estancia	Estancia
21	fehca_term_estancia	Estancia
22	puesto_tiene	Gestión
23	Responsabilidad	Gestión
24	horas_imperte	Docencia
25	programa_educativo	Docencia
26	horas_imparticion	Asesorías
27	tema_asesorias	Asesorías
28	tipo_investigación	Investigación
29	descripción_inv	Investigación
30	nombre_proyecto	Proyecto_Investigación
31	nombre_participantes	Proyecto_Investigación
32	número_participantes	Proyecto_Investigación
33	jefe-proyecto	Proyecto_Investigación
34	grado_avance_proyecto	Proyecto_Investigación
35	financiamiento_proyecto	Proyecto_Investigación
36	fecha_inicio_proy	Proyecto_Investigación
37	fecha-fin_proy	Proyecto_Investigación
38	nombre_articulo	Articulo
39	nombre_autor	Articulo
40	lugar_publicación	Articulo
41	nombre_revista	Articulo
42	año_articulo	Articulo
43	volumen_rev	Articulo
44	num_revista	Articulo
45	issn_art	Articulo
46	nombre_tesis	Tesis
47	fecha_inicio_tesis	Tesis
48	fecha_termino_tesis	Tesis
49	grado_avance_tesis	Tesis
50	Nivel	Tesis
51	Carrera	Tesis
52	nombre_alumno	Tesis
53	nombre_coasesor	Tesis
54	nombre_libro	Libro
55	motivo_libro	Libro
56	año_publicación	Libro
57	Editorial	Libro
58	número_paginas	Libro
59	Isbn	Libro
60	Autor	Libro
61	Coautor	Libro
62	nombre_congreso	Congreso
63	fecha_congreso	Congreso
64	tipo_congreso	Congreso
65	nombre_articulo	Congreso
66	año_congreso	Congreso
67	organiza_congreso	Congreso
68	isbn_congreso	Congreso

69	volumen_congreso	Congreso
70	nombre_conferencia	Conferencia
71	lugar_conferencia	Conferencia
72	tipo_conferencia	Conferencia
73	nombre_asignatura	Asignatura
74	nivel_asig	Asignatura
75	especialidad_alumno	Alumno
76	nombre_alumno	Alumno

El diagrama que muestra el modelo final se presenta en la *Figura 3.11*.

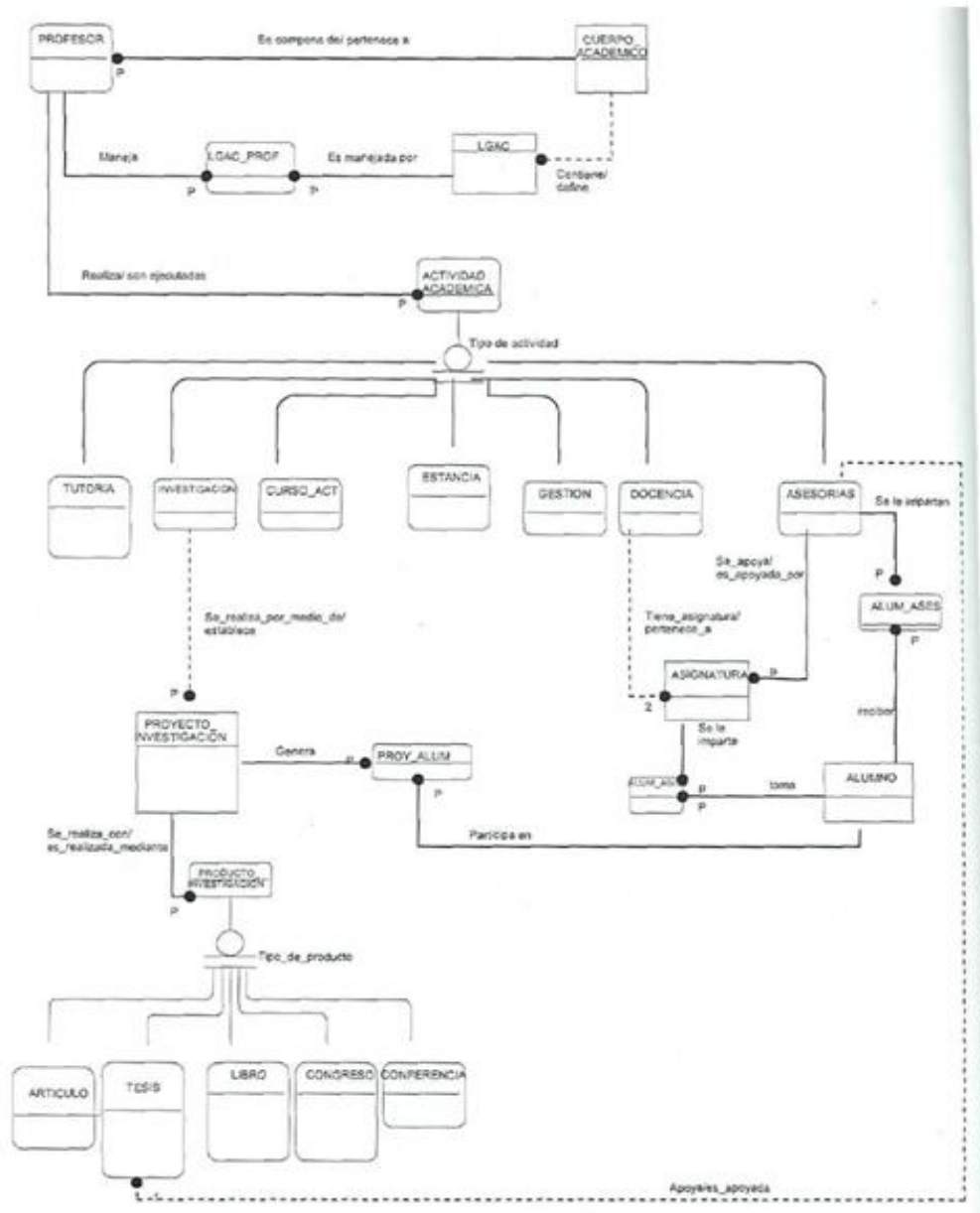
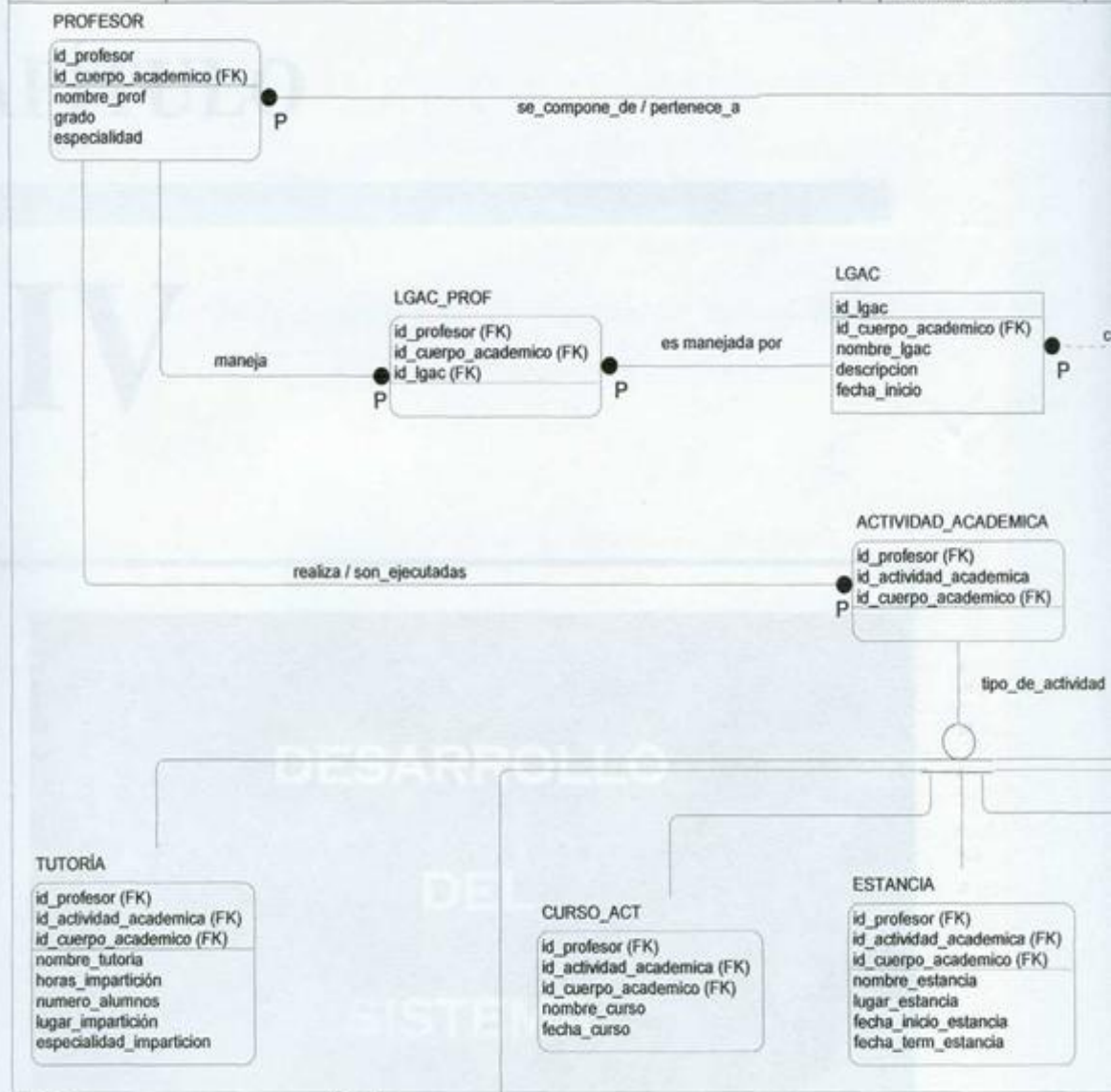


Figura 3.11. Modelo IDEF1x al final de la etapa 4 (ESTA EN OTRA HOJA LLAMADA MOD_IDEF)

USED AT:	AUTHOR:	DATE: 03/03/06	x	WORKING	RE
	PROJECT:	REV:		DRAFT	
	NOTES: 1 2 3 4 5 6 7 8 9 10			RECOMMENDED	
				PUBLICATION	



NODE: IDEF1X	TITLE:
--------------	--------

Figura a

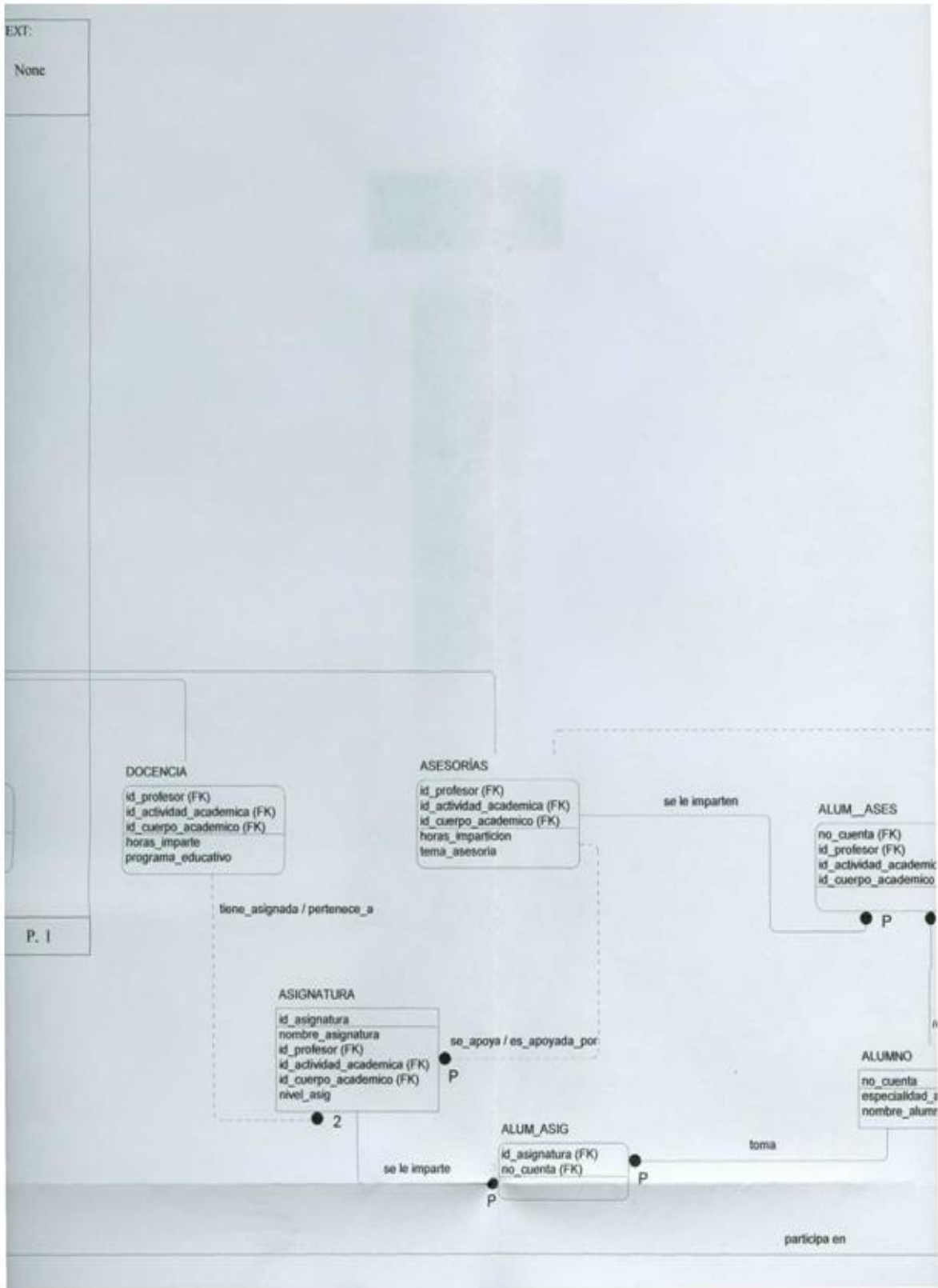


Figura a.1

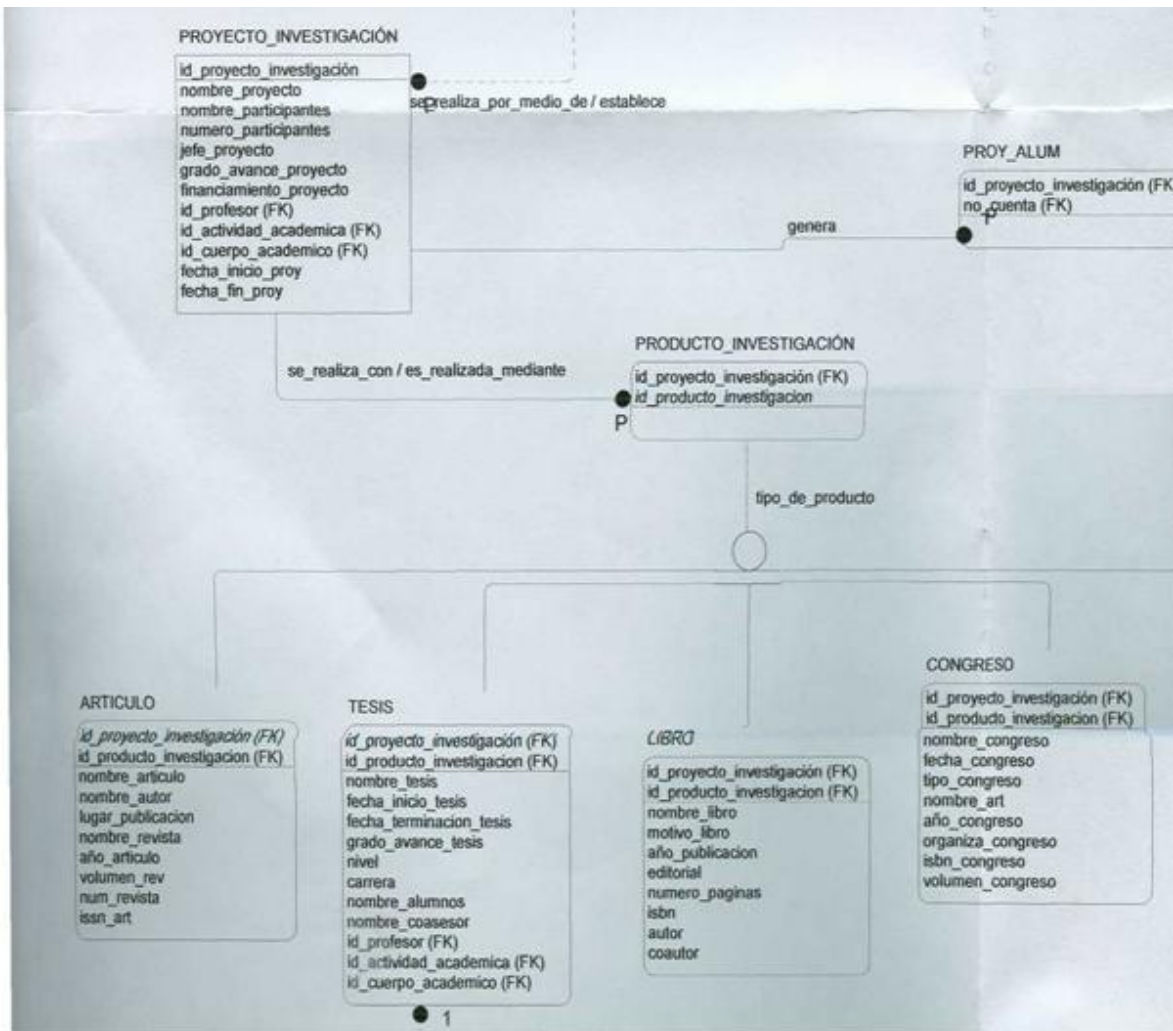


Figura a.2

CAPÍTULO

4

**DESARROLLO
DEL
SISTEMA**

CAPÍTULO IV. DESARROLLO DEL SISTEMA

4.1 Requerimientos de los Usuarios.

Los usuarios del sistema se dividen en dos grupos: Profesores y Administradores.

Los profesores tienen derecho a consultar y agregar información, a través de un acceso seguro. Se les asigna, por lo tanto, un nombre y una contraseña.

Como segunda y principal instancia tenemos a los administradores, los cuales tienen derecho a consultar, modificar y eliminar la información necesaria. Igualmente requieren de un nombre y de una contraseña.

Los demás usuarios sólo tendrán derecho a consultar la información general, los programas académicos con los que ofrecen, las líneas de investigación que se tienen y el plan docente que se tiene en el CITIS. Este tipo de usuarios no son usuarios registrados en el sistema.

4.2 Descripción del funcionamiento del sistema

El sistema funciona sobre la web. Cada usuario registrado podrá acceder al sistema a través de un navegador. El profesor tendrá permiso para consultar los datos de los profesores, como son: nombre, grado, especialidad y nombre del cuerpo académico al que pertenece. Al igual podrá consultar el estatus del cuerpo académico. Por otro lado, también podrá consultar el nombre y la descripción de

las LGAC a las que pertenecen tanto el profesor que consulta como los demás profesores.

Los administradores tienen el privilegio de modificar la información, al igual que de consultar y agregar. El usuario en el rol de administrador puede consultar toda la información, agregar información de los profesores, cuerpos académicos y LGAC's. En el papel de administrador, se puede eliminar la información que ya no es útil.

4.3 Modelado del funcionamiento del sistema

El modelado de funcionamiento del sistema se realiza por medio de diagramas de flujo. Éstos representan las interacciones entre el usuario y el sistema, y sirven de base para la implantación del mismo.

En el diagrama AGREGAR, tenemos que seleccionar “agregar datos generales del profesor”, y el sistema nos responde solicitándonos el nombre, clave, grado y especialidad. Al agregarlos, el sistema responde si los datos ya fueron agregados previamente o no. Si ya fueron agregados, el sistema nos mostrará una leyenda donde dice que los “datos ya existen”; en caso contrario, agregará, en la tabla “profesor” los datos solicitados. El sistema buscará en la tabla “cuerpo_academico” el *id* del cuerpo académico, según el nombre del cuerpo académico que nosotros le enviamos. Al encontrarlo, agregará dicho atributo en la tabla profesor. Finalmente, regresará a las opciones del administrador, como se muestra en el diagrama de flujo de la Fig. 4.1.

El siguiente diagrama de flujo muestra cómo CONSULTAR los datos del profesor. En primer lugar se selecciona “consultar datos generales del profesor”. Enseguida, el sistema solicita que se seleccione el nombre del profesor. Al pulsar la opción de consulta y si los datos del profesor están en la base de datos, nos muestra su nombre, grado, especialidad, nombre del cuerpo académico y LGAC a la que pertenece el profesor. En caso de que los datos del profesor no existan en la base de datos, se muestra una leyenda indicando que los datos del profesor no existen. También se genera la liga para agregar los datos del profesor (ver Fig. 4.2).

Continuamos con el último diagrama de flujo que es el de ELIMINAR datos del profesor. Como primer paso, se selecciona el botón eliminar, posteriormente el sistema nos solicita que seleccionemos el nombre del profesor. Si éste se encuentra en la base de datos lo elimina, enviando una leyenda “DATOS ELIMINADOS”, nos muestra un botón “CONTINUAR”el cual se selecciona y nos envía a la pagina opciones_administrador. En caso de que los datos del profesor no se encuentren en la base de datos nos da una opción para registrar sus datos, enviándonos a la pagina generales_agregar. (Ver Fig. 4.3)

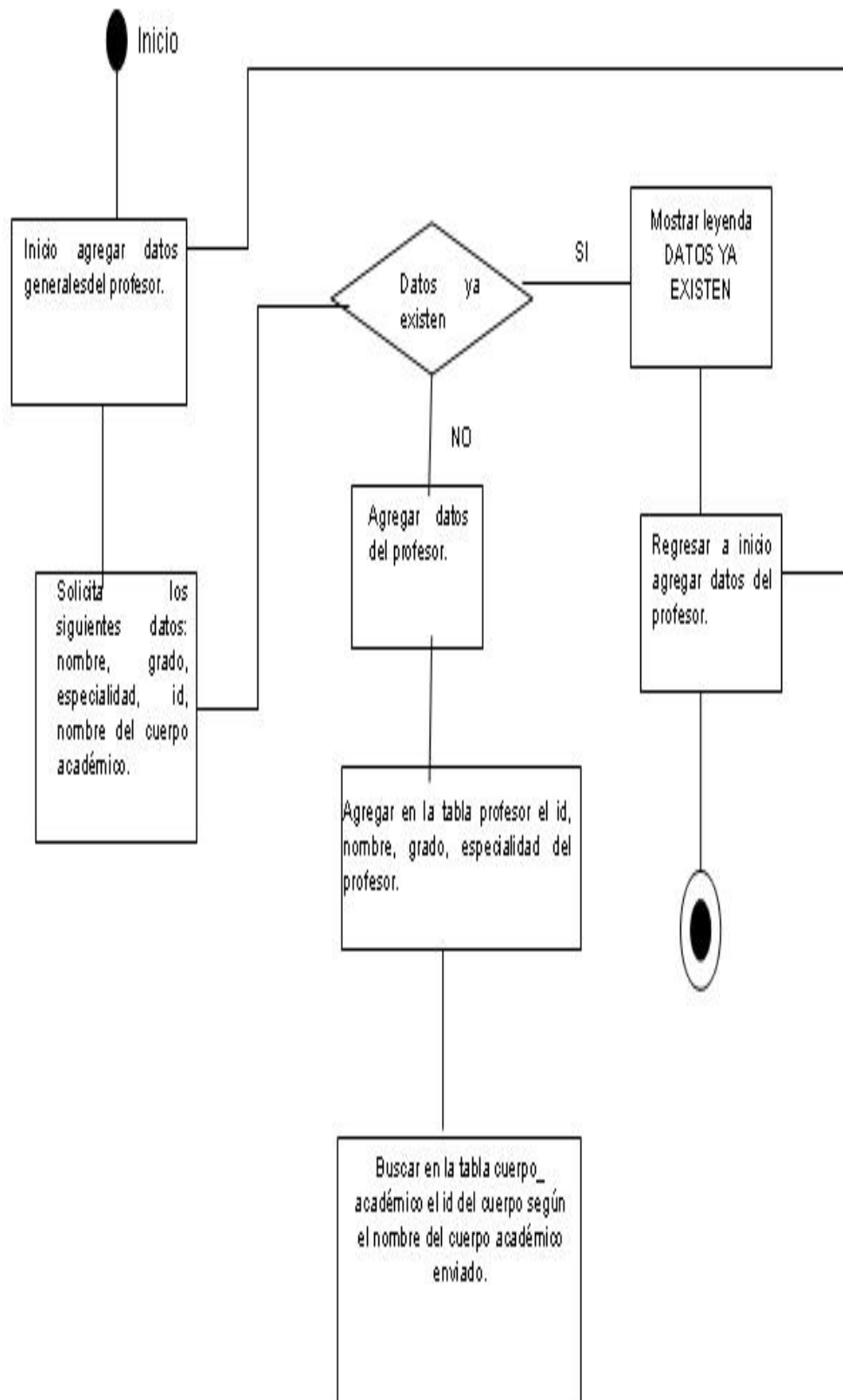


Fig. 4.1 Diagrama de Flujo-Agregar

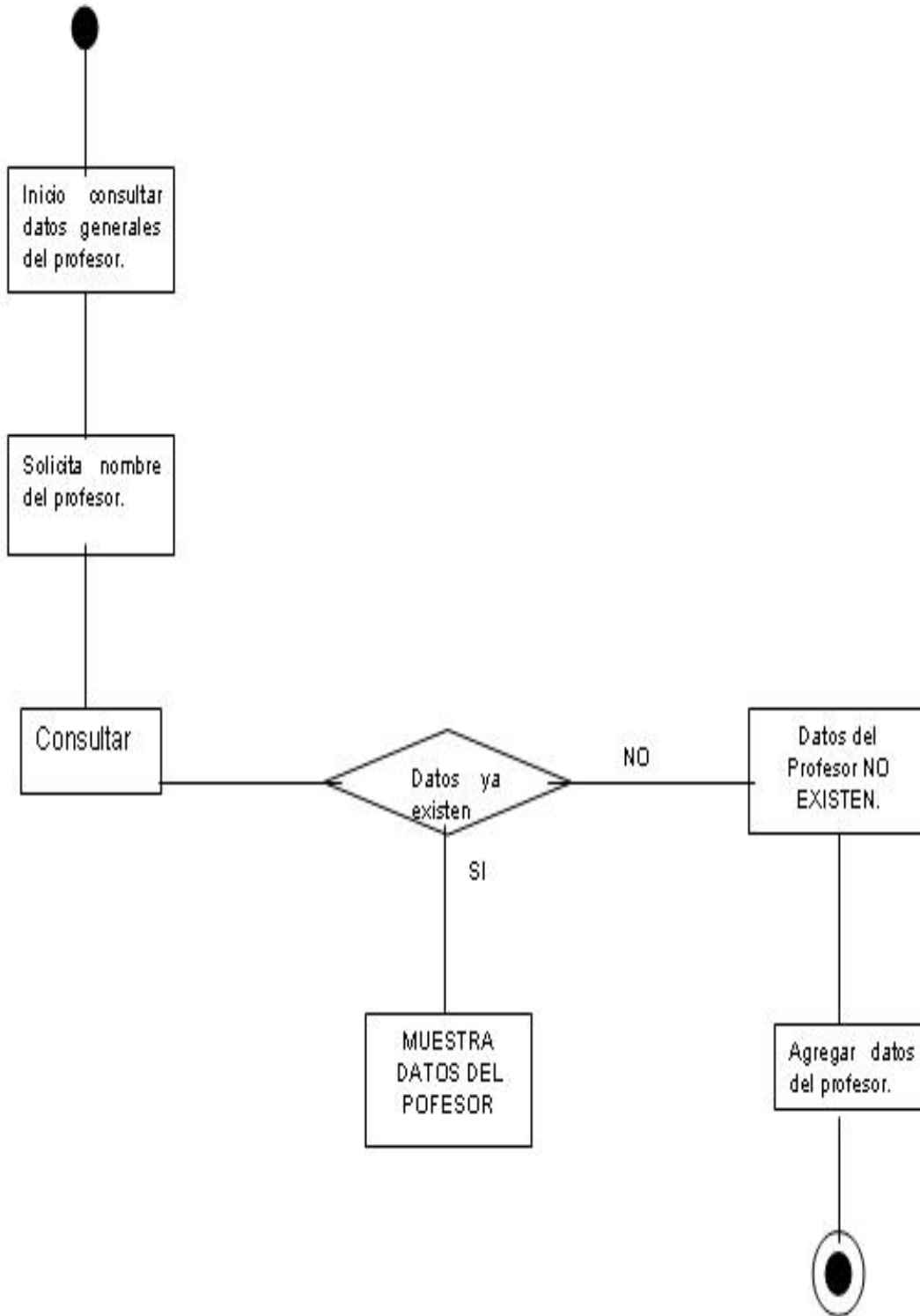


Fig. 4.2 Diagrama de Flujo-Consultar

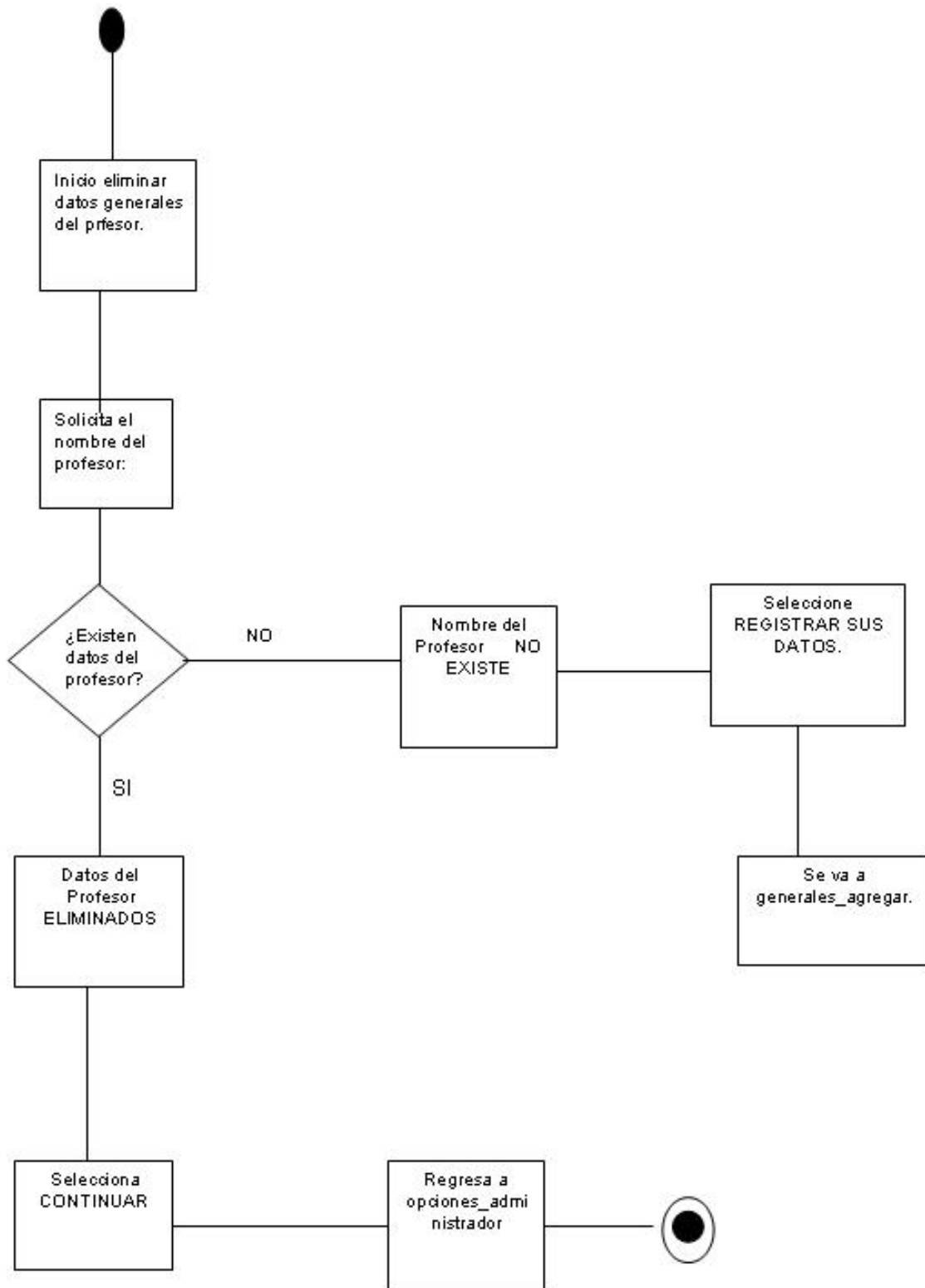


Fig. 4.3 Diagrama de Flujo-Eliminar

4.4 Modelos Navegacionales del Sistema

El siguiente modelo navegacional nos muestra las relaciones, permisos y restricciones de cada uno de los usuarios, dependiendo cual sea éste, profesor del CITIS o alumno. *Figura 4.4.*

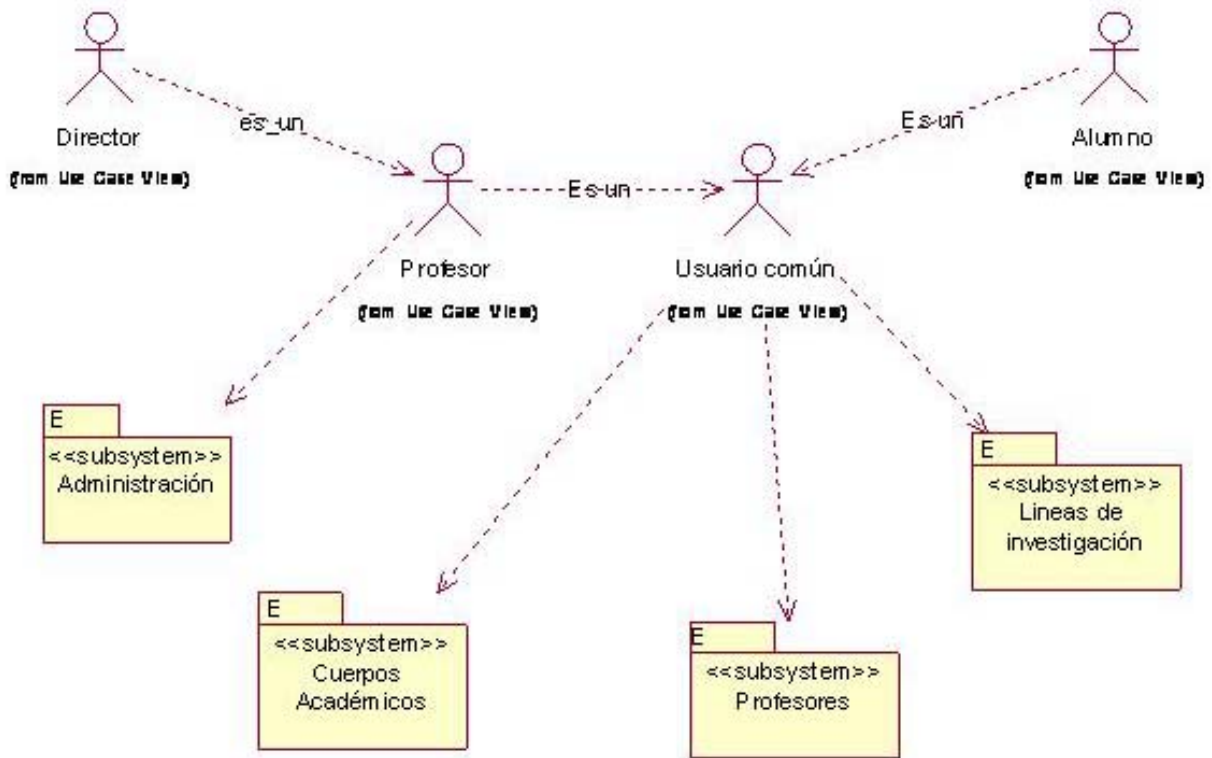


Figura 4.4 Vista General del Modelo Navegacional

En la *figura 4.5* vemos más detalladamente los permisos que tiene un usuario común, como es el de acceder a los datos generales para consultar los proyectos de investigación, las publicaciones que existen, las conferencias, estancias, asesorías, docencia, gestión, congresos, cursos y tesis que hay y por qué profesores son otorgadas.

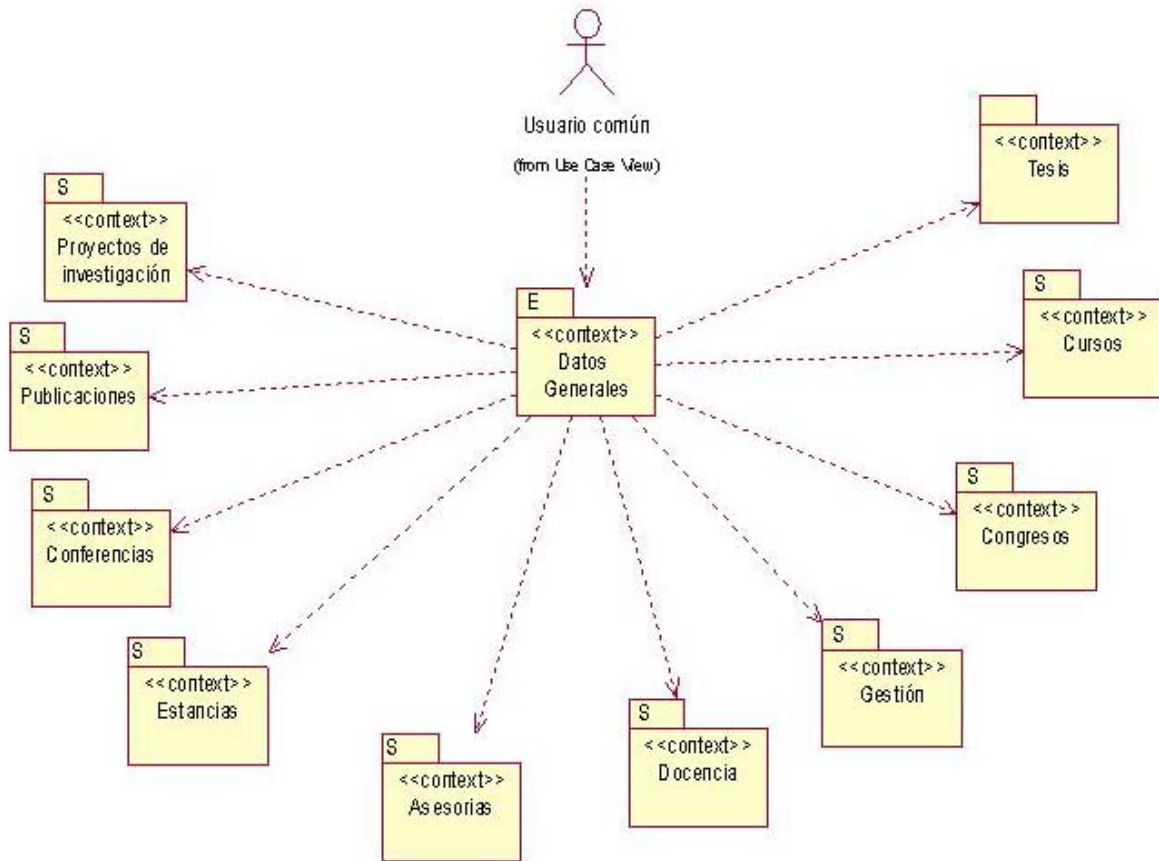


Figura 4.5 Modelo Navegacional de Usuario no Registrado

En la *figura 4.6* podemos observar cómo el director y el profesor tienen diferentes permisos. El profesor puede acceder a todos los productos de investigación, además de acceder también a algunas actividades académicas; en cambio el director tiene aún más permisos, como el de acceder a los cuerpos académicos, líneas de generación y aplicación del conocimiento y a profesores, además de acceder a todo el sistema si es necesario.

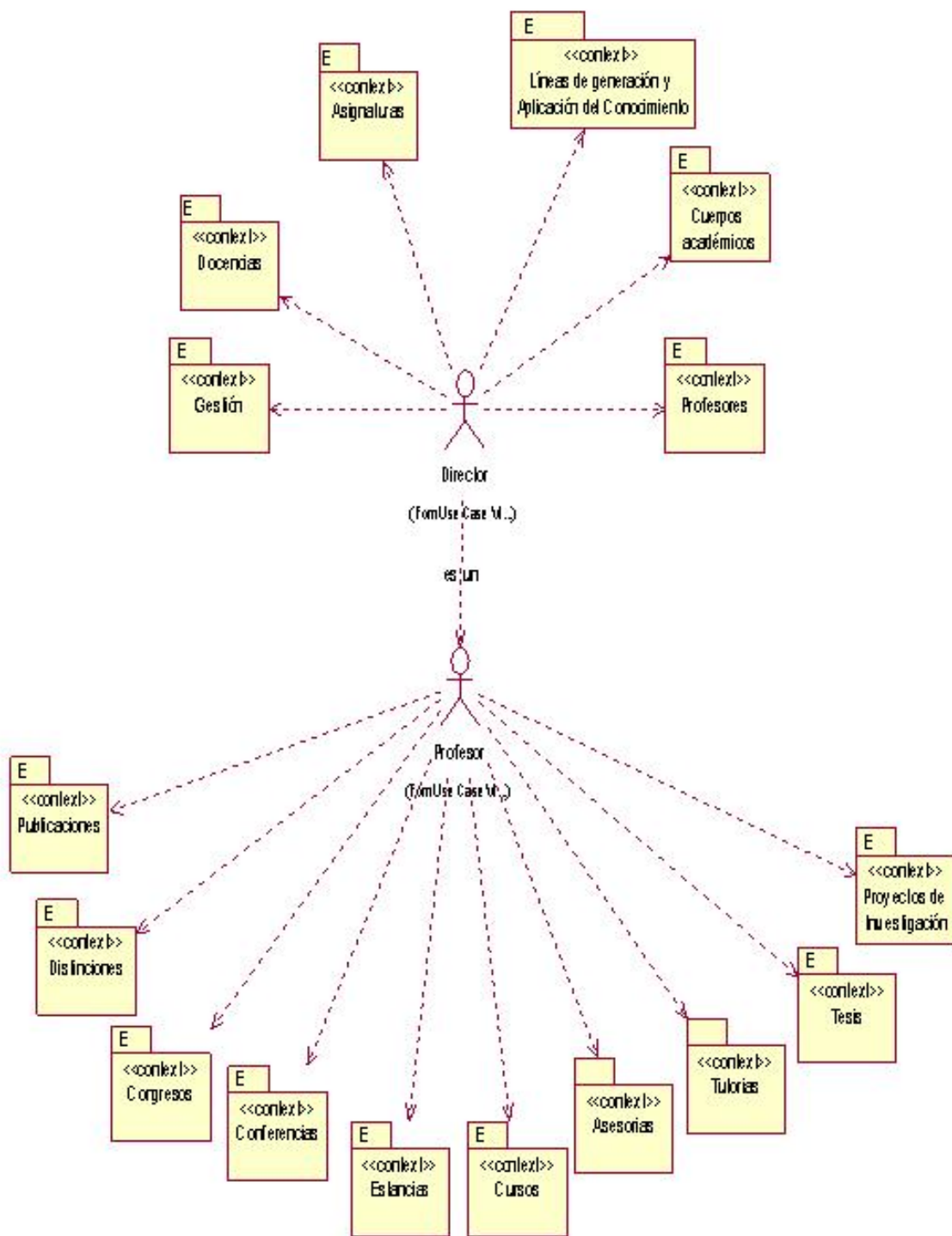


Figura 4.6 Modelo Navegacional de Usuario Registrado.

4.5 Implantación del sistema

El sistema es fácil de manejar, pues al entrar a la página principal, ésta cuenta con tres marcos o frames, de los cuales el de la parte superior sólo tiene un nombre, los dos de la parte inferior, son los que contienen toda la información importante, el de la parte izquierda tiene cinco anclas, de las cuales cuatro de ellas contienen información general acerca del centro de investigación, y la última ancla (SGIC) nos lleva al sistema.

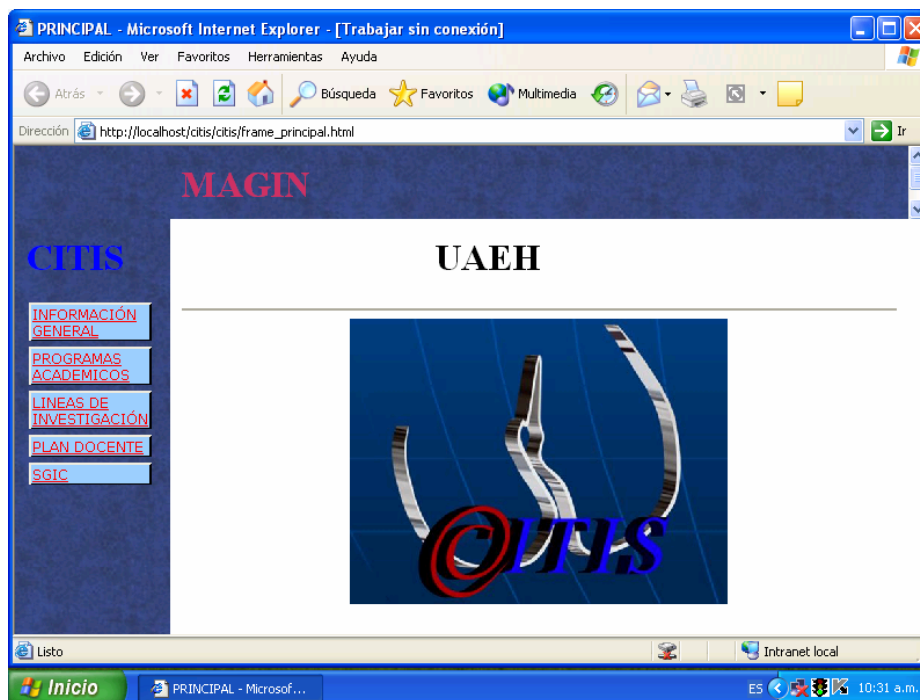


Fig. 4.7 PAGINA PRINCIPAL

Al seleccionar el botón (SGIC), éste nos lleva a que seleccionemos una opción, dependiendo de quien sea el usuario podrá seleccionar una de las dos opciones, pues para poder ingresar a la administración se tiene que contar con una clave, a diferencia de los profesores, como veremos en la siguiente figura.

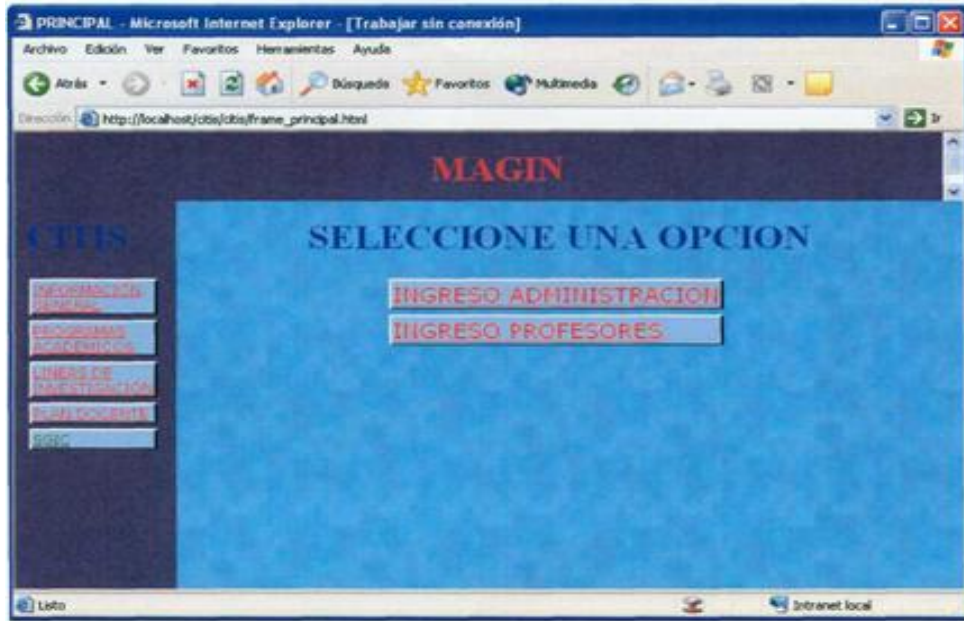


Fig. 4.8 SELECCIÓN DE INGRESO

Como podemos observar en la siguiente figura, sólo son tres los profesores que tienen permiso para ingresar como administradores, los cuales cuentan cada uno con su respectiva clave. Se tiene que seleccionar el nombre e ingresar la clave del administrador.



Fig. 4.9 INGRESO DE ADMINISTRADOR

En caso que la contraseña sea incorrecta, nos enviará una leyenda diciéndonos que el password es incorrecto como se muestra en la siguiente figura.

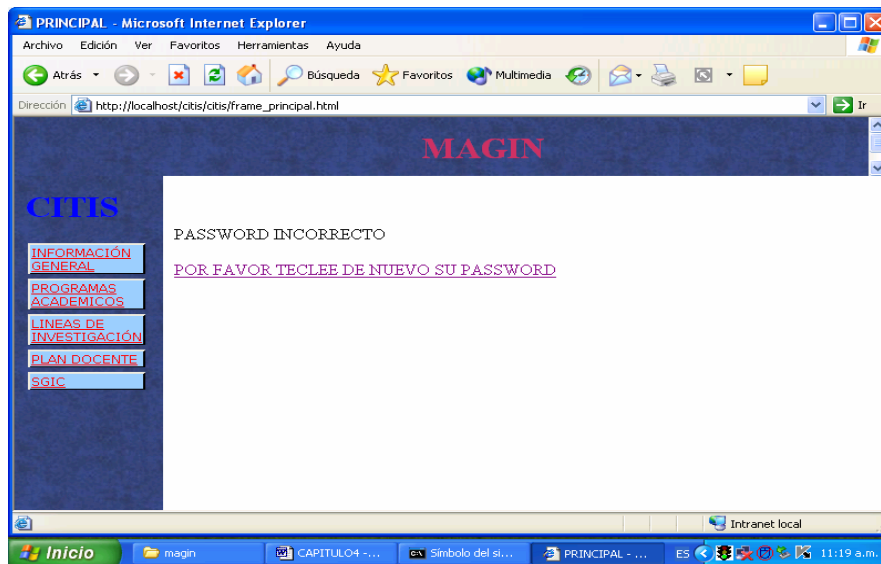


Fig. 4.10 PASSWORD INCORRECTO

Al mostrarnos la pantalla anterior seleccionamos el ancla que nos da (POR FAVOR TECLEE DE NUEVO SU PASSWORD) para volver a la pantalla anterior y así volver a seleccionar el nombre del administrador y teclear nuevamente la clave para poder ingresar al sistema.

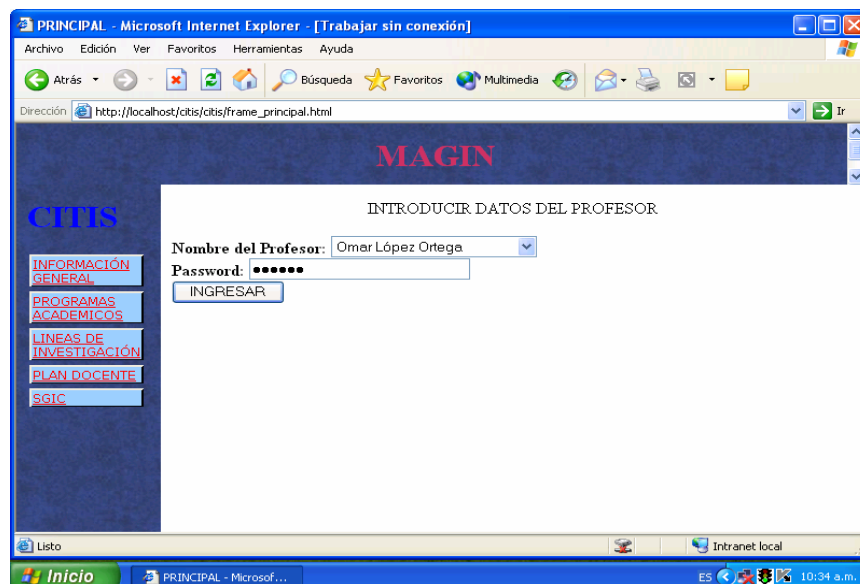


Fig. 4.11 PERMISO ADMINISTRADORES

Al teclear la clave correcta el sistema automáticamente nos manda a la selección de una de las tres opciones a las cuales el administrador puede acceder como se muestra en la siguiente figura.

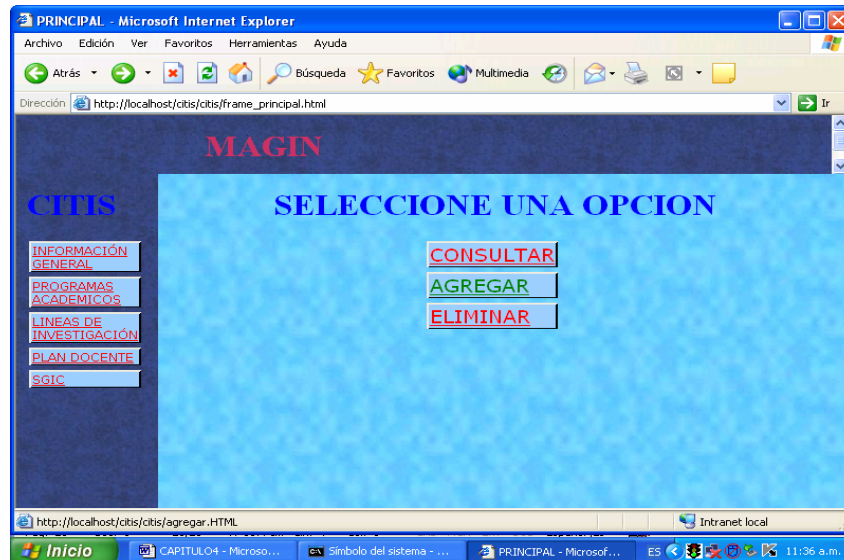


Fig. 4.12 OPCIONES DEL ADMINISTRADOR

Si se selecciona el botón agregar, automáticamente nos envía a otra opción de selección, pues aquí tendremos que seleccionar el botón al cual queremos agregar, ya sea en generales, cuerpo académico o lgac.



Fig. 4.13 OPCIONES PARA AGREGAR

Si seleccionamos el botón generales, éste nos envía a otra pantalla en la cual nos pide que agreguemos los datos de los profesores, y ya agregados los datos, seleccionamos el botón AGREGAR, para ingresar la información a la base de datos, como vemos en la siguiente figura.

The screenshot shows a web browser window titled "PRINCIPAL - Microsoft Internet Explorer". The address bar displays "http://localhost/citis/citis/frame_principal.html". The main content area has a dark blue header with the word "MAGIN" in red. Below the header, there is a sidebar on the left with the "CITIS" logo and several menu items: "INFORMACIÓN GENERAL", "PROGRAMAS ACADÉMICOS", "LINEAS DE INVESTIGACIÓN", "PLAN DOCENTE", and "SGIC". The main content area is titled "INTRODUCIR DATOS DEL PROFESOR" and contains a form with the following fields: "Nombre del Profesor:" (a dropdown menu with "Félix Agustín Castro Espinoza" selected), "Clave:" (a text input field with "*****"), "Grado:" (a dropdown menu with "Maestría" selected), "Especialidad:" (a text input field with "Ingeniería de Software"), and "Nombre del CA:" (a dropdown menu with "Soft Computing" selected). At the bottom of the form is a button labeled "AGREGAR". The browser's status bar at the bottom shows "Listo" and "Intranet local".

Fig. 4.14 DATOS A AGREGAR

Al oprimir el botón AGREGAR, el sistema envía una pantalla en donde se muestra la clave del cuerpo académico. También muestra la clave del profesor, y aparecen las LGAC de acuerdo al nombre del Cuerpo Académico seleccionado. De las LGAC existentes, se selecciona una o más a las cuales pertenece el profesor.

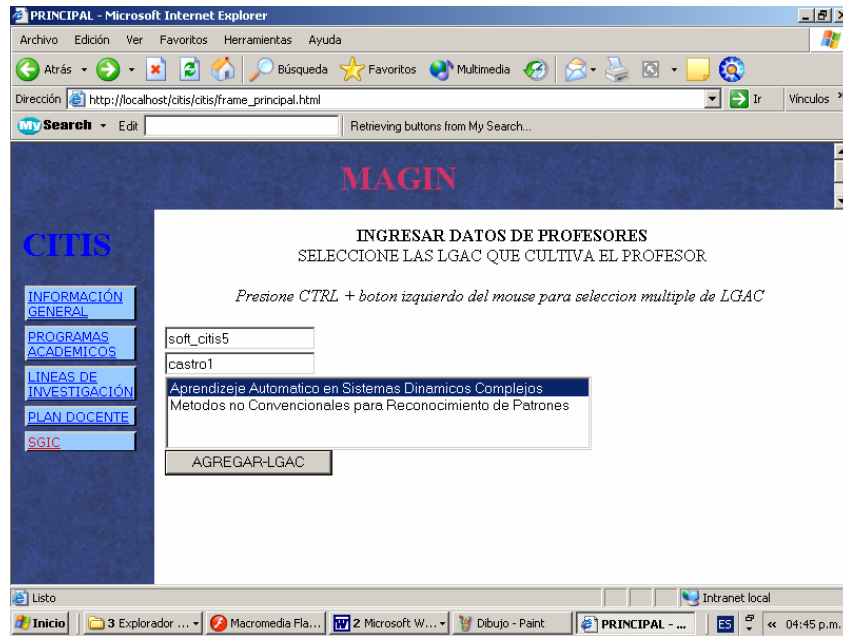


Figura 4.15 AGREGAR LGAC

Cuando se pulsa el botón AGREGAR-LGAC, el sistema genera otra pantalla en donde se muestra la clave del profesor, al igual que el nombre y la clave de la LGAC.

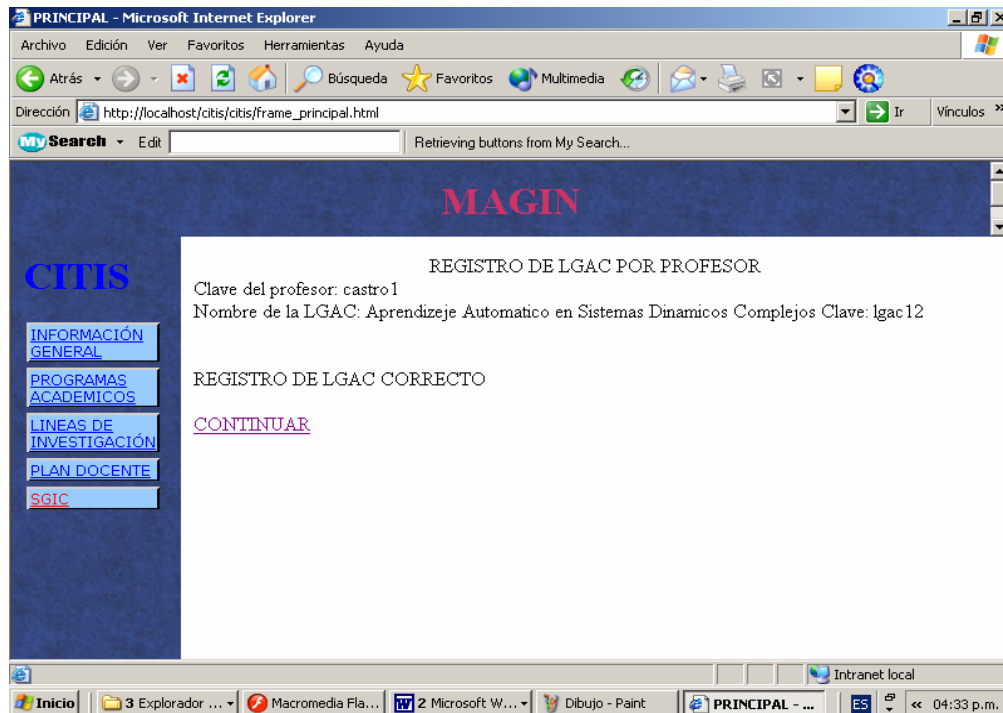


Figura 4.16 REGISTRO DE LGAC

En caso que los datos agregados ya estuvieran almacenados en la base de datos, el sistema nos envía una leyenda como la que se muestra en la figura 4.12. Seleccionamos el vínculo y éste nos conduce a la página anterior, con el fin de agregar los datos del profesor.

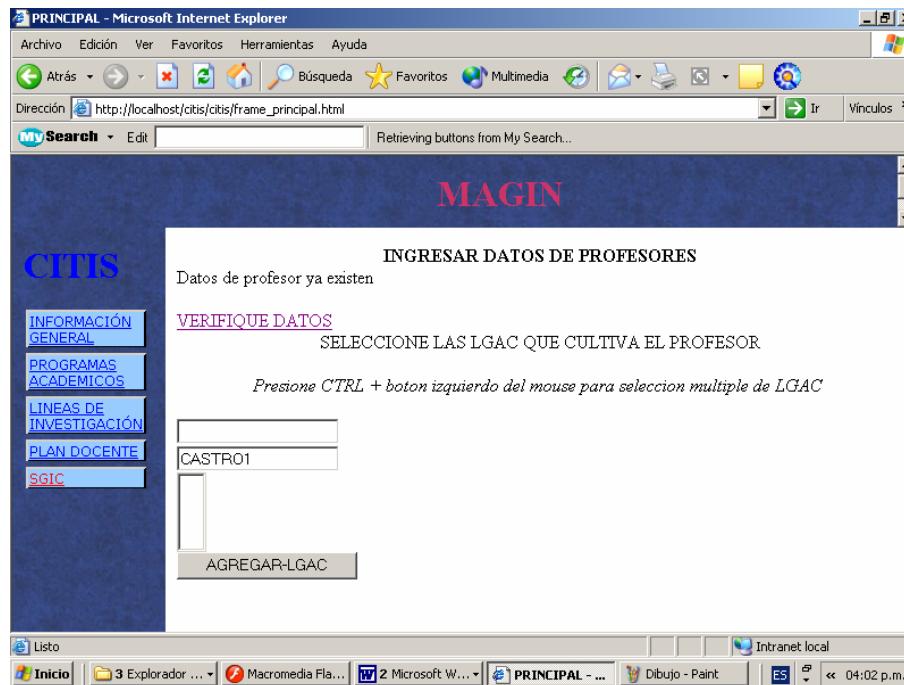


Fig. 4.17 DATOS EXISTENTES

Si los datos agregados no se encuentran en la base de datos al seleccionar el botón AGREGAR, el sistema conduce a otra página en la que indica la clave del cuerpo académico al que pertenece, su contraseña y además nos pide que seleccionemos la LGAC del profesor. Ya seleccionada ésta, se pulsa el botón AGREGAR-LGAC para que agregar la información.

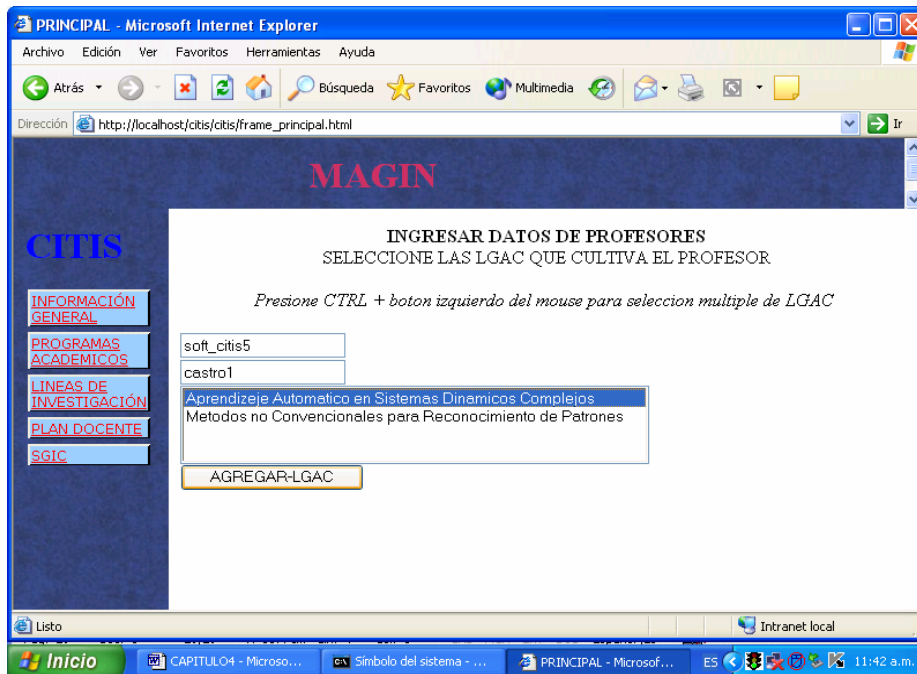


Fig. 4.18 LGAC INGRESADA

A seleccionar el botón AGREGAR-LGAC, el sistema automáticamente nos conduce a otra pantalla, la cual nos muestra la clave del profesor, el nombre de la LGAC con su respectiva clave y nos manda una leyenda diciéndonos que el registro de la LGAC fue correcto, así como se muestra en la figura siguiente.

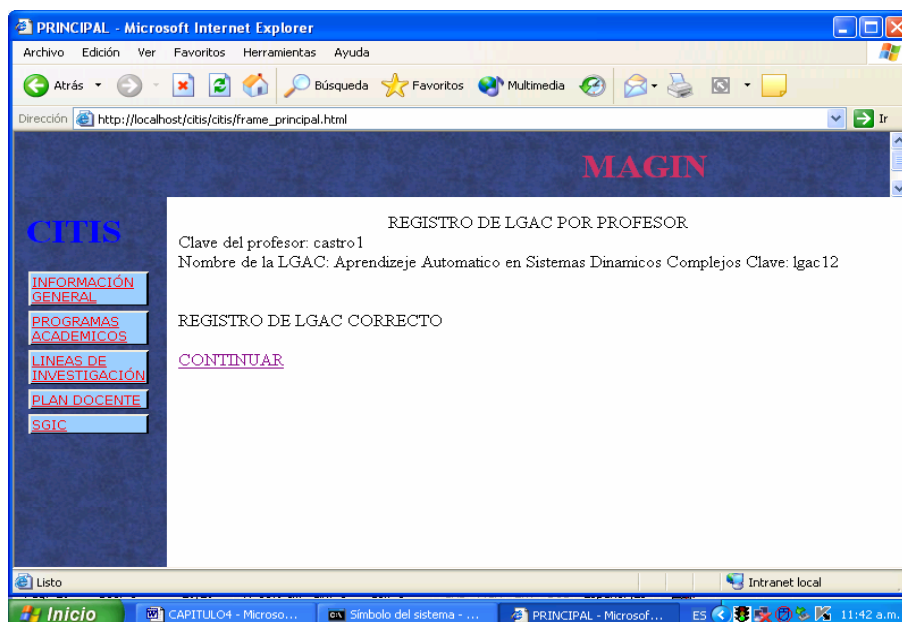


Fig. 4.19 LGAC AGREGADA

Como nos podemos percatar, la pantalla anterior tiene un botón para seleccionar y poder continuar. Al seleccionarlo, el sistema nos lleva nuevamente a la pantalla de opciones del administrador (ver Fig. 4.12). En esta página tenemos que seleccionar una de las tres opciones. Si se selecciona el botón CONSULTAR, el usuario es conducido a una página de opciones para consultar (ver Fig. 4.13). En esta página seleccionamos nuevamente generales, la cual nos envía una página en la cual nos pide que seleccionemos el nombre del profesor que deseamos consultar, como podemos ver en la Fig. 4.20.

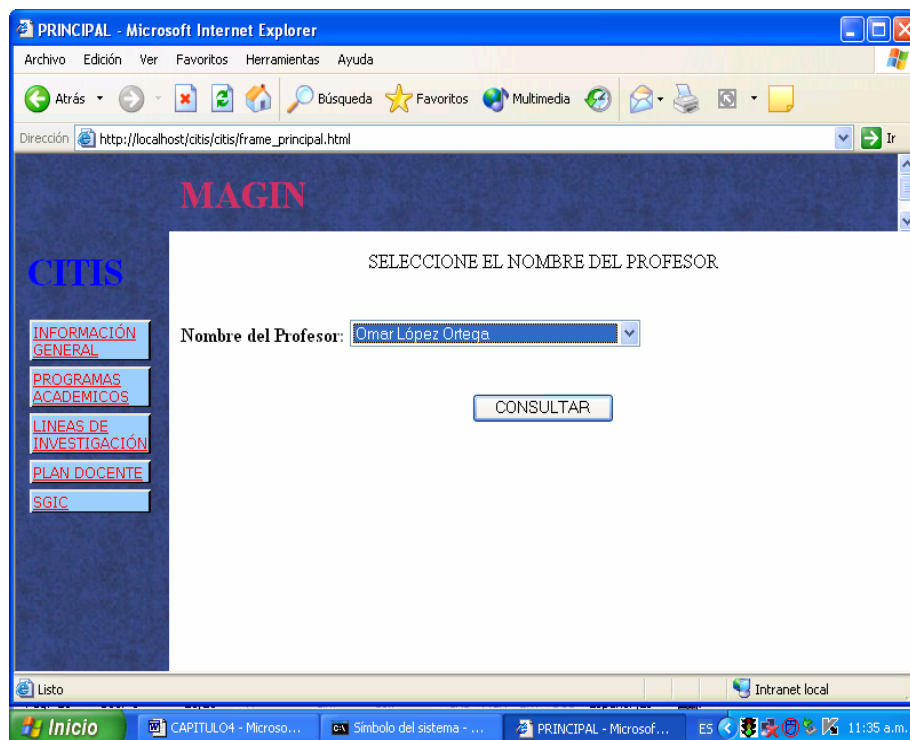


Fig. 4.20 DATOS A CONSULTAR

Ya seleccionado el nombre del profesor a consultar, seleccionamos el botón CONSULTAR el cual nos lleva a otra página para mostrarnos la información del profesor que solicitamos, en este caso nos muestra el nombre, grado y

especialidad del profesor, así mismo nos muestra el nombre del cuerpo académico y las lgac a las que pertenece, como podemos ver en la Fig. 4.21.



Fig. 4.21 DATOS DE CONSULTA

Al seleccionar CONTINUAR, ésta nos vuelve a llevar a las opciones del administrador (ver Fig. 4.12), en esta página seleccionamos una de las tres opciones. Si se selecciona el botón ELIMINAR, éste nos lleva a una página de opciones para eliminar. En esta página seleccionamos nuevamente la opción “GENERALES”, la cual nos envía una página en la cual nos pide que seleccionemos el nombre del profesor que deseamos eliminar como podemos ver en la Fig. 4.22.

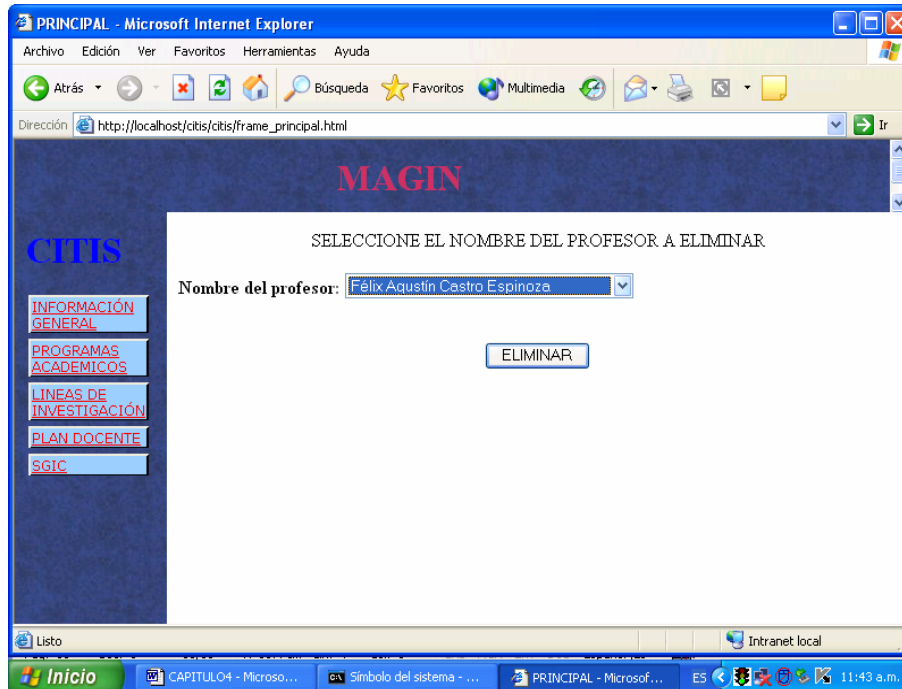


Fig. 4.22 DATOS A ELIMINAR

Ya que seleccionamos el nombre del profesor que deseamos eliminar, seleccionamos el botón eliminar, para que podamos eliminar la información, y así nos envía una pantalla, la cual nos dice que los datos del profesor han sido eliminados, como se muestra en la Fig. 4.23.

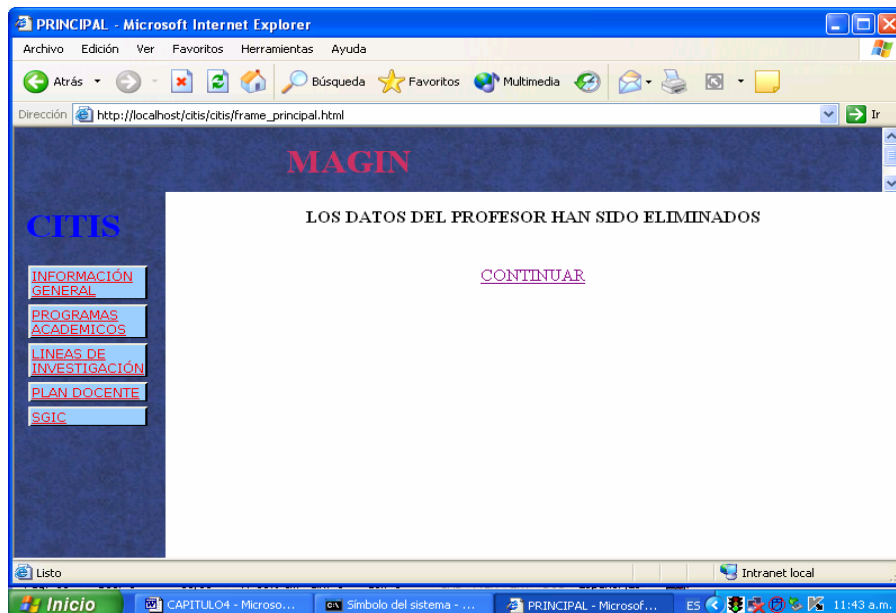


Fig. 4.23 DATOS ELIMINADOS

En esta pantalla hay un vínculo, a través del botón CONTINUAR. Esta opción conduce al usuario a las opciones del administrador (*ver Fig. 4.12*). En esta pantalla, si queremos podemos seguir agregando, consultando o eliminando la información que nosotros necesitemos, o de lo contrario salirnos del sistema.

Conclusiones Generales

En esta tesis se ha hablado acerca de lo importante que es para una organización la integración de los procesos internos en el diseño de sistemas de información, y de cómo la gestión de una organización influye en el diseño y estructura de los sistemas de información. De igual forma se describió que los avances en las tecnologías de la información influyen en el (re)diseño de los procesos internos de la organización. Este marco conceptual sirvió de base para el desarrollo del presente trabajo de tesis. Particularmente, se hizo énfasis en el modelado de datos y procesos empleando los lenguajes IDEF0 e IDEF1x. Las conclusiones del trabajo se describen a continuación, de acuerdo con el objetivo original de la tesis, i. e. realizar un sistema de información orientado a la web tomando como caso de uso al Centro de Investigaciones en Tecnologías de Información y Sistemas (CITIS), a través de los modelos de procesos y modelos de datos. Se logró modelar el sistema de información de acuerdo a la metodología IDEF. En este sentido, los modelos IDEF1x sirvieron para describir la estructura lógica de las bases de datos, la cual fue implantada en el SGBD MySQL. La traducción de los modelos de datos hacia la descripción física de las bases de datos fue realizada coherentemente. Esto es debido a que IDEF1x captura las entidades empleadas en los procesos descritos en IDEF0. El modelo de datos se realizó de un modo estricto, realizando diferentes entrevistas y recavando información de los PTC acerca de su trabajo desempeñado dentro del CITIS y de sus diferentes actividades que éstos desarrollan, y plasmando estas actividades de acuerdo con el método seguido.

Otro objetivo planteado fue la realización de modelos para la implementación de mapas navegacionales. En este caso no se utilizaron los modelos de casos de uso, tal como lo describe la metodología original. En nuestro caso se adaptó a la metodología y se ocuparon los diagramas IDEF1x que reflejan de modo más fiel la estructura de base de datos subyacente.

Se cumplió el objetivo de implantar el modelo en un sistema de gestión de base de datos relacional. El modelo IDEF1x desarrollado fue implantado en MySQL sin pérdidas de información y de un modo transparente. Esto sirvió de base para integrar la base de datos hacia un sistema orientado a la Web. o anterior permitió cumplir con el objetivo de desarrollar el sistema orientado a la Web, utilizando PHP. Los diagramas IDEF1x junto con los mapas navegacionales permiten definir qué información se maneja en la interacción del usuario con el sistema, cuando éste se programa hacia la Web.

El objetivo de validación del sistema, se hizo utilizando una porción de la base de datos de acuerdo con la información que la dirección del CITIS aportó. Por otro lado como trabajo futuro hay que poner en práctica el sistema. Para ello se requiere modificar los procesos internos del CITIS con respecto al procedimiento de actualización de resultados por parte de los profesores. Este cambio organizacional, aunque no corresponde al presente trabajo, puede llevarse a cabo tomando como base los diagramas IDEF0 presentados en esta tesis.

Glosario de términos

ATRIBUTO: Es un conjunto de valores característicos para determinada tabla, dado que el valor de un atributo es el contenido particular para el atributo de la entidad.

ENTIDAD: Es una clase de objetos que comparten características comunes.

FORMAS NORMALES: Son reglas de normalización que están diseñadas para prevenir anomalías de actualización e inconsistencia de datos.

HTML: Siglas en inglés HyperText Markup Language, "Lenguaje de etiquetado de hipertexto"; es el lenguaje que se utiliza para crear las páginas Web, un lenguaje muy sencillo que permite combinar gráficos, textos y enlaces.

ICOM: Los diagramas utilizados en IDEF0, también llamados diagramas A-0, son utilizados para el modelado de decisiones, acciones y actividades de una organización.

IDEF: Identification definition. Es un lenguaje de modelado de procesos y procesos.

IDEF0: Es un método diseñado para el modelado de decisiones, acciones y actividades de una organización o sistema.

IDEF1x: El lenguaje IDEF1x se emplea para el modelado de datos que nos permite establecer las relaciones existentes entre cada una de las entidades y los atributos correspondientes a cada una de ellas.

INTERFAZ: Sistema de comunicación de un programa con su usuario; la interfaz comprende las pantallas y los elementos que informan al usuario sobre lo que puede hacer, o sobre lo que está ocurriendo.

INTERNET: La intercomunicación de un número ingente de redes diversas y computadoras bajo el protocolo TCP/IP. Constituye con mucho la mayor red de la historia y comunica a millones de personas y proporciona multitud de servicios.

INTRANET: Atributo que se refiere a todo aquello relacionado con la red que está conectado a Internet pero mantiene su separación, para distinguirlo de aquellas completamente integradas.

MODELADO DE PROCESOS: Se emplea para incorporar todas las actividades relacionadas para la transformación de conocimiento acerca del sistema en modelos que describan los procesos llevados a cabo por las organizaciones.

MODELADO DE SISTEMAS DE INFORMACIÓN: Se usa para describir todos aquellos enfoques que pretenden hacer que nuestras abstracciones de los sistemas de información reproduzcan a los sistemas reales que representan.

RELACIÓN DE CONEXIÓN: Es una asociación entre dos entidades. Cada instancia de la primera entidad (la entidad padre) de algún modo esta relacionada con 0, 1, ó más de una instancia de la segunda entidad (la entidad hijo).

SGBD: Sistema de Gestión de Base de Datos.

SQL: El Structured Query Lenguaje es un lenguaje de programación de base de datos, hablamos por tanto de un lenguaje normalizado que nos permite trabajar con cualquier tipo de lenguaje normalizado (ASP, HTML, C++) en combinación con cualquier tipo de base de datos.

SOLICITANTE: Puede ser un PTC o un investigador o alumno, interno o externo a la UAEH.

USUARIO: Un usuario puede considerarse como un sinónimo de solicitante.

Apéndice

Se anexa un disco compacto que contiene:

- El modelo IDEF1x
- El script de la generación de la base de datos
- El código html
- El código php.

Referencias

- [1] Elmasri, Navathe, "Sistemas de Bases de Datos", 2ª edición, Ed. Pearson Educación.
- [2] Chen, P.P-S... 1976, The entity-relationship model-toward a unified view of data. ACM Transaction on Database Systems, I, 9-36
- [3] Codd, E. F., 1970 "A relational model of data for large shared data banks. Communications of the ACM", 13,377-783
- [4] Lucas, Henry, "La tecnología de la información y la paradoja de la productividad", Editorial Oxford, 2000.
- [5] Hammer, Michel y James Champy, "Reingeniería", Editorial Norma, 1994.
- [6] Kusiak, Andrew; Letsche, Terry and Zacarian, Armen; "Data modeling with IDEF1x"; *Int. J. Computer Entegrated Manufacturing*, 1997, Vol.10, No. 6, pp.470-486.
- [7] López Ortega, Omar; Núñez, Esquer, Gustavo; Acosta Samperio, Karina; Vera Guerrero, Julio; "Modelo de un sistema de información unificando Casos de Uso e IDEF1x", Tercer Congreso Internacional de Ingeniería Electromecánica y de Sistemas, Noviembre 25-29, 2002, México, D.F.
- [8] Acosta Samperio, Karina; Vera Guerrero, Julio César; "Desarrollo de un Sistema de Información Unificando Casos de Uso e IDEF1x", TESIS para obtener el título de Ingeniero Industrial, UAE, Agosto 2003, Pachuca, Hgo.
- [9] MySQL Ian Gilfillan Editorial Anaya Multimedia Ed. 2003.
- [10] <http://www.apache.org>

[11] <http://www.gnu.org>

[12] <http://idef.com>

[13] <http://hugles.com.au>

[14] <http://programacion.com>

[15] <http://www ldc.usb.ve/~vtheok/webmaestro>

[16] <http://geneura.ugr.es/~maribel/php/>

[17] http://es.tldp.org/Manuales-LuCAS/manual_PHP/manual_PHP/

[18] <http://www.aqa.es/doc/>