

Universidad de las Ciencias Informáticas

Facultad 8



*Sistema de gestión
de información de la Facultad 8.
Diseño de la Base de Datos.*

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas.

AUTORES:

Arcel Labrada Batista.

Damir Góngora Mora.

TUTOR:

Ing. Mario Antonio Castelló Polo

Ciudad de La Habana, Cuba

Junio, 2007



DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Facultad ocho de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Arcel Labrada Batista.

Autor

Damir Góngora Mora.

Autor

Ing. Mario A. Castelló Polo.

Tutor



“... lo que da al hombre el poder no es el mero conocimiento que viene del uso de los sentidos, sino, ese otro conocimiento más profundo que se llama Ciencia.”

José Martí.



Arcel:

A Jesús por haberme encontrado cuando estaba perdido.

A mis padres por haberme dado todo el apoyo que pudieron y el que no, para hacer de mí lo que soy.

A mi novia, Nela por quererme tanto y pensar que soy el hombre más inteligente del mundo.

A mis abuelas Teresa y Reina que tanto amor me han dado.

A mi primo y hermano Modestico por ser un hijo para mi madre y un hermano para mí, ayudándola y cuidándola, y haciendo mis tareas de hijo mientras estoy lejos.

A mi primo y hermano Betico que aunque lejos esté, siempre lo tengo presente y recuerdo mucho nuestra infancia.

A mis tías Teresita y Beysi por quererme como a un hijo.

A mi tía Martha y sus hijos por ayudarme tanto este tiempo.

A mis amigos: Arodys, Osdalme, Karenia y Bazán, que encontré durante la carera y me han ayudado y cuidado.

A Yisel Pupo por ser mi amiga y curarme cuando he estado enfermo. Le digo gracias y fuerzas para que ella me ponga en sus agradecimientos porque se que ella llegará a esta parte también.

A mis amigos de siempre, Carlos Javier, Javier Ferrales, Luis Manuel, Raidel Muños (Primo), y Juan Carlos por ser siempre mis amigos y saber que con ellos siempre pude y puedo contar.

A mi compañero de tesis Damir por descubrir las cosas que solo no hubiera podido para el desarrollo de este trabajo.

A mi tutor Mario Antonio por ser mi guía en la asignatura que comencé a impartir y en este trabajo.

A toda mi familia, mis amigos y compañeros gracias por estar ahí.



Damir:

*A mis **padres** por ser las personas que más quiero y a quienes le debo lo que soy, por haber creído en mí y darme todo su apoyo.*

*A mi **hermano** por estar a mi lado en los buenos y malos tiempos.*

*A **Yudita** por ayudarme y darme apoyo en los momentos en que más los necesité. De que si quieres algo en la vida solo debes luchar con todas tus fuerzas por ello para conseguirlo sin importar lo que los demás digan de ti.*

*A mis **compañeros de aula** por ayudarme durante el transcurso de mi carrera.*

*A mi tutor **Mario Antonio** por ayudarme en todo lo que estuvo a su alcance.*

*A **Marianela** por aportar su granito de arena.*

*A **Arcel** mi compañero de tesis sin el que la culminación de este trabajo no hubiera sido posible.*

Además de que hacerlo solo hubiera sido bastante aburrido, el sabe por qué.

*A toda mi **familia, amigos y compañeros** que una forma u otra han contribuido a mi formación como profesional.*



Arcel:

Dedico este trabajo a mis padres, mi novia y toda mi familia que siempre supieron que yo podría.

Lo dedico también a los que algún día pasaron por mi vida y contribuyeron a formarme como profesional y como persona en este mundo.

Damir:

A mis padres y hermano que siempre confiaron en mí, a los que me ofrecieron su amistad y me brindaron su apoyo para que viera realizados mis sueños, y a los que no se encuentran aquí pero siempre creyeron que lo lograría.



El objetivo de este trabajo es el diseño de una base de datos para solucionar las deficiencias existentes en la facultad ocho de la Universidad de las Ciencias Informáticas, relacionadas con el almacenamiento y gestión de la información.

En este trabajo se expone, inicialmente, el resultado de una investigación sobre los sistemas de gestión de información en sentido general, así como el estudio de las tendencias actuales tanto a nivel internacional como nacional. Se profundiza en el conocimiento de los Sistemas Gestores de Bases de Datos, sus objetivos, ventajas y desventajas.

Posteriormente se describe la propuesta de solución, así como de la arquitectura y su fundamentación. Se realiza un análisis de optimización de consultas y de las fases por las que pasa durante la ejecución de las mismas. También se describen los requisitos funcionales y se muestran los diagramas de clases persistentes y modelo de objetos.

Finalmente se obtiene la Base de Datos y se efectúa la validación del diseño realizado.



INTRODUCCIÓN.....	2
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	6
INTRODUCCIÓN.....	6
1.1 SISTEMAS DE GESTIÓN DE INFORMACIÓN. TENDENCIAS ACTUALES.....	7
1.1.1 <i>Sistemas a nivel internacional.....</i>	<i>7</i>
1.1.2 <i>Sistemas a nivel nacional.....</i>	<i>9</i>
1.2 SISTEMAS GESTORES DE BASES DE DATOS (SGBD).....	10
1.2.1 <i>Objetivos de un SGBD.....</i>	<i>10</i>
1.2.2 <i>Ventajas y desventajas.....</i>	<i>11</i>
1.3 TECNOLOGÍAS Y HERRAMIENTAS DE DESARROLLO.....	12
1.3.1 <i>MySQL.....</i>	<i>12</i>
1.3.2 <i>Microsoft SQL Server.....</i>	<i>14</i>
1.3.3 <i>Oracle.....</i>	<i>14</i>
1.3.4 <i>PostgreSQL.....</i>	<i>15</i>
1.3.5 <i>Comparación entre los SGBD.....</i>	<i>16</i>
1.3.6 <i>Embarcadero Erwin Studio.....</i>	<i>17</i>
1.3.7 <i>Case Studio 2.....</i>	<i>17</i>
1.4 PROCESO DE DESARROLLO.....	18
1.4.1 <i>Lenguaje Unificado de Modelado (UML).....</i>	<i>19</i>
1.4.2 <i>Rational Rose.....</i>	<i>20</i>
1.5 OTRAS HERRAMIENTAS.....	21
1.5.1 <i>EMS SQL Manager para PostgreSQL.....</i>	<i>21</i>
1.5.2 <i>EMS SQL Query para PostgreSQL.....</i>	<i>22</i>
1.5.3 <i>PgAdmin 3.....</i>	<i>22</i>
CONCLUSIONES.....	23
CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.....	24
INTRODUCCIÓN.....	24
2.1 INTEGRACIÓN CON OTROS MÓDULOS Y SISTEMAS.....	24
2.2 DESCRIPCIÓN DE LA ARQUITECTURA Y FUNDAMENTACIÓN.....	24



2.3 ANÁLISIS DE OPTIMIZACIÓN DE CONSULTAS.....	26
2.4 SELECCIÓN DE LOS REQUISITOS FUNCIONALES.....	32
2.5 DIAGRAMA DE CLASES PERSISTENTES Y MODELO DE OBJETOS.	42
2.6 DESCRIPCIÓN DE LAS CLASES.....	51
2.7 DISEÑO DE LA BD. DIAGRAMAS ENTIDAD-RELACIÓN.	55
2.8 DESCRIPCIÓN DE LAS TABLAS.....	63
CONCLUSIONES.....	67
CAPÍTULO 3: VALIDACIÓN DEL DISEÑO REALIZADO	68
INTRODUCCIÓN.....	68
3.1 VALIDACIÓN TEÓRICA DEL DISEÑO.....	68
3.1.1 Integridad.....	68
3.1.2 Normalización de la base de datos.....	70
3.1.3 Análisis de redundancia de información.....	72
3.1.4 Análisis de la seguridad de la base de datos.....	73
3.1.4.1 Seguridad y control de acceso.....	73
3.1.4.2 Backups y recuperación de los datos.....	73
3.1.5 Trazabilidad de las acciones.....	75
3.2 VALIDACIÓN FUNCIONAL.....	75
3.2.1 Herramientas para un llenado voluminoso e inteligente de la base de datos.....	75
CONCLUSIONES.....	75
CONCLUSIONES	76
RECOMENDACIONES.....	77
REFERENCIAS BIBLIOGRÁFICAS.....	78
BIBLIOGRAFÍA	80
GLOSARIO	81
ANEXOS	83
ANEXO 1.- DESCRIPCIÓN DE LAS CLASES PERSISTENTES.....	83
ANEXO 2.- DESCRIPCIÓN DE LAS TABLAS DE LA BASE DE DATOS.....	94
ANEXO 3.- MODELOS DE OBJETOS.....	118



En los centros educacionales cubanos, y entre estos la Universidad de las Ciencias Informáticas (UCI), es necesario el control y manejo de la información acerca de estudiantes, profesores y trabajadores del centro, para así generar reportes y tomar decisiones.

Actualmente existe gran cantidad de información de todo tipo o categoría que se desea manejar por secretaría, directivos, profesores, entre otros; esto se torna una tarea un poco trabajosa por la cantidad de información a procesar, aunque, si bien existe una aplicación Web, (AKADEMOS), que gestiona mucha información, esta no abarca toda la que se desea manejar, atentando contra la organización, creación de reportes y toma de decisiones por parte de las personas encargadas de trabajar con dicha información.

El proceso de búsqueda, actualización y manejo de información que se utiliza actualmente en la facultad número ocho de la UCI no es el más efectivo. Algunos de los mayores problemas existentes es el hecho de que la información está descentralizada, su procesamiento es lento porque esta se encuentra en documentos de Microsoft Word, Microsoft Excel, entre otros, que no son medios profesionales para el manejo de grandes cantidades de información, no se utilizan las herramientas o medios apropiados, debido a que no existen. En la facultad número ocho de la UCI durante el proceso de investigación, se detectaron las siguientes deficiencias:

- ✓ El cúmulo de información es cada vez más grande por lo que llegará el momento en que se hará necesario tener un local preparado para almacenar toda la información (expedientes, planillas, plantillas, actas, etc.) o comenzar a desechar documentos que contengan información valiosa y que en un momento determinado pueden ser de gran utilidad para obtener un historial sobre la facultad.
- ✓ Al tener ese gran cúmulo de información almacenado en documentación convencional, la búsqueda de un dato determinado, por muy organizado que esté el establecimiento que se designe para almacenar dicha documentación, siempre tomará algunos minutos, haciendo lenta la investigación, en dependencia de la cantidad de archivos.
- ✓ La otra vía de almacenamiento en la facultad es el uso de software tales como Microsoft Word, Excel, entre otros, que si bien permiten hacer algunas búsquedas y otras operaciones con informaciones, no son flexibles para las búsquedas grandes, avanzadas, eficientes, rápidas y con opciones de actualizaciones, etc.



- ✓ La posibilidad de que todos puedan acceder a la información de manera rápida es poca pues no existe una publicación de la misma.
- ✓ Todo el personal que tiene la responsabilidad de entregar información a la facultad, debe hacer la entrega de la misma en discos de almacenamiento directamente a la persona encargada de recibir la misma, o enviarlo por correo, algo que se hace lento, crea gran cúmulo de información en los servidores de correo así como en el buzón de correo electrónico del destinatario, además se pueden perder datos en algún descuido en el momento del envío o transportación de los mismos.

Por las razones anteriormente planteadas se hace necesaria la confección de un modelo que permita la centralización, rapidez y eficiencia en cuanto al almacenamiento y manipulación de los datos de la facultad ocho de la UCI, logrando con ello realizar reportes para la toma de decisiones en el menor tiempo posible. Al mismo tiempo, la información estará disponible en la red con acceso a la información de acuerdo al rol desempeñado por el usuario, dependiendo del cargo y responsabilidades. Además debe brindar seguridad sobre la información que se almacenará.

Este trabajo surge para dar respuesta a las deficiencias existentes en la facultad, relacionados con el almacenamiento y gestión de la información, por lo que el **problema a solucionar** en él, consiste en:

¿Cómo centralizar y agilizar el proceso de gestión de la información docente y no docente de la facultad número ocho de la UCI?

Por tanto el **objeto de estudio** de este trabajo es el procesamiento y manipulación de la información de la facultad número ocho de la UCI.

De ello se deriva que el **campo de acción** que abarca este trabajo, es la informatización de la gestión y búsqueda de información de la facultad número ocho de la UCI.

Como **Idea a defender** se plantea que:

Si se desarrolla un nuevo modelo para la gestión de la información de la facultad, se piensa que es posible centralizar y agilizar la búsqueda de la información de la facultad número ocho de la UCI.

Los **aportes prácticos** que se esperan obtener son:

- ✓ Centralización de toda la información referente a los procesos que se desarrollan en la facultad.



- ✓ Rapidez en la comunicación y en las búsquedas de información por parte del personal disminuyendo su carga de trabajo, debido a la automatización de los procesos que se realizan.

El **objetivo general** de la investigación es modelar, diseñar e implementar una base de datos para la centralización y manejo de información docente y no docente de la facultad número ocho de la UCI.

Como **objetivos específicos** se plantean los siguientes:

1. Realizar un estudio e investigación de las bases de datos de otros sistemas de gestión de información de otras universidades.
2. Realizar un estudio sobre el funcionamiento de las actividades docentes y no docentes que se llevan a cabo dentro de la facultad.
3. Realizar un estudio de los procesos y que serán necesarios para lograr la consistencia e integridad de la información que se desea almacenar.
4. Diseñar un modelo que permita la manipulación de información de manera rápida y eficiente.

Para poder dar cumplimiento, de una forma completa y exitosa, a estos objetivos se ha decidido desarrollar las siguientes **tareas**:

1. Determinar y elaborar los requisitos que debe cumplir el sistema para su culminación exitosa.
2. Evaluar y proponer una solución que contenga todas las clases necesarias para el proyecto.
3. Estudio de la factibilidad de dicho sistema.
4. Realizar un estudio sobre todas las posibles herramientas a utilizar para la implementación del sistema.
5. Comparar las herramientas estudiadas y utilizar las más factibles.
6. Análisis, diseño e implementación de la Base de Datos.

El presente trabajo, estructurado en 3 capítulos, contiene la siguiente información:

Capítulo 1. Fundamentación Teórica: permite conocer el objeto de estudio, el de automatización, así como cuestiones teóricas necesarias para la comprensión del trabajo en cuestión y el conocimiento de las



posibles herramientas a usar.

Capítulo 2. Descripción y análisis de la solución propuesta: donde se tratará acerca de la estrategia de integración de la solución con otros módulos o sistemas, la descripción de la arquitectura y fundamentación, análisis de optimización de consultas, selección de los requisitos funcionales del sistema propuesto, construcción del modelo de objetos y diagrama de clases persistentes, se realizará una descripción por cada una de las clases, así como también el diagrama Entidad Relación de la Base de Datos.

Capítulo 3. Validación del diseño realizado: durante este capítulo se realizará la validación teórica del diseño, teniendo en cuenta la integridad, la normalización de la Base de Datos, el análisis de redundancia de información, un análisis de la seguridad de la base de datos, trazabilidad de la acciones. Así como la validación funcional utilizando herramientas o programas que permiten un llenado voluminoso e inteligente de la base de datos.

Al final se podrán observar las conclusiones, recomendaciones, así como las referencias bibliográficas, bibliografía, un glosario de términos y anexos necesarios que harán más comprensible muchas de las partes del documento.



Introducción.

En su todavía corta historia, la tecnología de la información ha vivido dos grandes hitos: el primero vino dado por el desarrollo del modelo relacional de bases de datos realizado por Codd en 1970 y el segundo, por la llegada de las soluciones de planificación de recursos. Antes del modelo relacional las aplicaciones definían y gestionaban su propio modelo de datos almacenando la información en ficheros externos o en soluciones más sofisticadas que utilizaban modelos de datos diversos como los jerárquicos o en red. Esta situación provocaba que diferentes aplicaciones dentro de la misma organización tuvieran replicada una gran cantidad de información con los problemas derivados de consumo de recursos, inconsistencias, repetición de tareas, falta de seguridad, etc. Con la llegada del modelo relacional y de los sistemas de gestión de bases de datos relacionales se comenzó un proceso de extracción de los datos de las aplicaciones hacia las bases de datos relacionales. Las organizaciones empezaron a diseñar un modelo de datos global para toda la organización sobre el cual se construían las aplicaciones, que acudían al gestor de bases de datos para el tratamiento de los datos.

Este cambio supuso un gran avance tanto para la gestión de los datos de las organizaciones como para el desarrollo de aplicaciones informáticas. Se disponía de un punto central de gestión de los datos, lo que permitía un mayor control en la seguridad de los mismos, una mayor eficiencia en su tratamiento y la eliminación de inconsistencias, entre otras ventajas. Las aplicaciones eran más fáciles de diseñar y más ligeras al no ser necesarios muchos módulos encargados de la gestión de datos. Las aplicaciones se comunicaban y se comunican actualmente, con la base de datos mediante un lenguaje de consulta y de definición de datos estandarizado, el SQL (Structured Query Language), lo que permite incluso no depender de un gestor de base de datos concreto, pudiendo crear una capa de interfaz entre la aplicación y la base de datos que posibilita migrar de gestor de base de datos con un esfuerzo mínimo.

La separación de los datos de las aplicaciones dio lugar a las arquitecturas de software de dos capas, una para las aplicaciones que definían las operaciones a realizar y provocaban consultas y modificaciones sobre los modelos de datos, y otra formada por la(s) base(s) de datos que daban soporte a las aplicaciones. Posteriormente, al separarse los sistemas que interactúan con el usuario/cliente de las aplicaciones surgieron modelos de tres capas.



Es aproximadamente sobre la década de los años 80 que puede hablarse del surgimiento de una nueva disciplina en el ámbito de la Información: la gestión de la información, que replantea el mercado del trabajo del profesional de la información y demanda un nuevo tipo de profesional con importantes responsabilidades en el diseño y el desarrollo de los sistemas de información en las organizaciones.

Las organizaciones están viviendo un cambio en el paradigma de desarrollo de sus sistemas de información: de los datos a los procesos. La finalidad que se persigue con ello es enfatizar los procesos de negocio para conseguir arquitecturas más ágiles y flexibles, adaptables a los continuos cambios que se producen en los mercados en los que las organizaciones desarrollan su negocio. El objetivo es independizar la gestión de los procesos de negocio de las aplicaciones, para que cualquier modificación en la lógica de negocio no afecte al código de las aplicaciones.

En los centros educacionales, se desarrollan diferentes dimensiones de la gestión. La gestión de sistemas, servicios, procesos, tecnología, recursos financieros, humanos, así como de su propia evolución o cambio que demandan la atención de todos los que se desempeñan en ellas.[1]

1.1 Sistemas de gestión de información. Tendencias actuales.

1.1.1 Sistemas a nivel internacional.

En el ámbito internacional este tipo de sistema se ha difundido bastante, sobre todo en el área latinoamericana.

AGUILA: Administración y Gestión Universitaria de información a la Labor Académica. Chile.

Permite recopilar y tratar la información de los distintos niveles institucionales, lo que relaciona la labor docente, investigativa, de creación artística y extensión, y permite conocer cuantitativa y cualitativamente las iniciativas desarrolladas por los distintos organismos de la institución.

Desde el año 2003 se realizan procesos de actualización de las Base de Datos contenidas en el SIGA.

Entre sus principales módulos se encuentran el de Personal Académico, que recoge todo la información referente a estudiantes y profesores, el de Actividad Académica que recoge todo lo referente a los programas de estudio, las carreras, programas de postgrados y postítulo, el de Proyecto y Creación Artísticas, que recoge todo lo referente a los proyectos, publicaciones, creaciones artísticas y culturales.[2]



SIU: Sistema Integral de Información Universitaria. Universidad de Veracruz, España.

El SIU automatiza métodos y procedimientos de operación de la Universidad y sistematiza el uso, la explotación y la presentación de la información universitaria. Su propósito es hacer eficiente la gestión académica y administrativa para ofrecer servicios a la comunidad universitaria, propiciando la adopción de modelos educativos más flexibles. Este sistema funciona a través de tres módulos: estudiantes, que tiene por objetivo ofrecer servicios de información y administración escolar en apoyo al quehacer de docentes, estudiantes y egresados, el de recursos humanos, proporciona información veraz y oportuna de los datos personales, profesionales y la historia laboral de los empleados y por último el de recursos financieros brinda información de carácter financiero, contable y presupuestal que sirve como base para la planeación y la toma de decisiones.[3]

SIGA: Sistema Integral de Gestión Académica. Universidad de Córdoba, España.

Las funcionalidades soportadas por el SIGA abarcan desde la gestión del acceso a la universidad hasta la tramitación de los títulos pasando por la matrícula, actas de examen, expedientes académicos, estadísticas, informes, gestión de becas, convalidaciones, etc. Entre sus principales módulos se encuentra el de Matrícula que permite la inscripción del estudiante en las asignaturas que desee y le estén permitidas. Además conforme a su expediente académico y a la normativa existente, el módulo de Expediente permite consultar y registrar cualquier información académica de un alumno relacionada con su estancia en la Universidad. El módulo de Actas el cual genera actas para las convocatorias y grupos de clases establecidos. Las calificaciones pueden ser registradas en las secretarías, por las secretarías y los profesores.[4]

Lleu: “llibre electrònic d'universitat”

Permite a los estudiantes conocer todo lo relativo a su titulación, así como personalizar sus horarios, calendarios de exámenes, fichas descriptivas de asignaturas, planes de estudios, asignaturas ofrecidas en la libre elección, planos de las instalaciones universitarias, etc.

La herramienta ayuda al encargado de titulación en el análisis de posibles errores cometidos por él. Cuando se editan calendarios de exámenes, por ejemplo, el sistema advierte de la omisión y/o repetición de convocatorias, a la vez que visualiza la información de diferentes modos: curso a curso, mezclando varios cursos, etc. En el caso de los horarios, por ejemplo, el sistema detecta la omisión de reservas para



asignaturas, la presencia de un excesivo número de horas reservadas para una asignatura, o la asignación de un aula que ya está ocupada en un rango horario concreto.[5]

SinedUC: Sistema de Información a través de Internet, Chile.

Es un sistema de información orientado a los colegios de Chile, que permite gestionar un conjunto de información. Como son: los reportes de notas, anotaciones e inasistencias; además de una agenda con horario de clases, fecha de pruebas, todo esto puede ser consultado a través de Internet.[6]

1.1.2 Sistemas a nivel nacional.

GESTACAD: Sistema de Gestión Académica.

Este sistema se estuvo usando durante un tiempo en la UCI.

El sistema está concebido por módulos, entre los que se diferencian, los módulos de actualización de datos y el sitio Web, a través del cual se muestran las diversas salidas de la aplicación.

El sistema te permite buscar un estudiante, mostrar una estadística general de cuantos hay por criterios (facultad, centro de procedencia, vía de ingreso, provincia, etc.). Además de sus notas, promedio general, etc.

AKADEMOS: Sistema automatizado para la gestión académica.

En estos momentos se está utilizando en la UCI, para la gestión de la información en el área docente, un sistema de gestión denominado Akademos y por ende, la facultad hace uso del mismo.

Este sistema brinda una serie de funcionalidades, en él se almacenan los datos, por facultad, de los resultados obtenidos en las pruebas realizadas, la matrícula con que se cuenta, asistencia a clases y causas que describen la situación del ausente, resultados de los cursos optativos, entre otras.

No obstante, el sistema presenta una serie de dificultades que no permiten cubrir las necesidades de la gestión de la información de la facultad, pues, más que esas posibilidades que brinda Akademos, se necesita un sistema que gestione otras informaciones sobre los estudiantes y profesores, tanto de su vida docente como residencial.



Se requieren otras funcionalidades como recoger todo lo referente a las organizaciones políticas y de masa, los datos personales y de interés de los trabajadores y estudiantes, el historial del recorrido de los estudiantes durante el semestre en cuanto a participación de actividades políticas o culturales así como brindar la posibilidad de observar la información sobre los resultados que estos han ido obteniendo durante el curso en la residencia, señalamientos, pases solicitados, etc.

Existen sitios Web para las organizaciones políticas y de masa, pero no brindan la posibilidad de publicar informaciones, guardar las actas de las reuniones ordinarias, controlar la asistencia de los participantes, así como otras cuestiones que son de interés para las organizaciones y para la dirección de la facultad que deben ser gestionadas y automatizadas, pues ayudan a la toma de decisiones. Además, AKADEMOS, presenta, según opiniones solicitadas a los usuarios del mismo, la navegación del sitio muy profunda y complicada.

1.2 Sistemas Gestores de Bases de Datos (SGBD).

En la actualidad, y gracias al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos tienen formato electrónico, que ofrecen un amplio rango de soluciones al problema de almacenar datos.

Los Sistemas de gestión de base de datos son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.

En la actualidad existen numerosos sistemas gestores de bases de datos, entre ellos el Microsoft Access, Oracle, MySQL, Visual Fox Pro, SQL Server y PostgreSQL, entre otros.[7]

1.2.1 Objetivos de un SGBD.

Existen distintos objetivos que deben cumplir los SGBD:

- Los SGBD ahorran a los usuarios detalles acerca del almacenamiento físico de los datos. Da lo mismo si una base de datos ocupa uno o cientos de archivos, este hecho se hace transparente al usuario. Así, se definen varios niveles de abstracción.



- La independencia de los datos consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.
- Un buen diseño de una base de datos logrará evitar la aparición de información repetida o redundante. De entrada, lo ideal es lograr una redundancia nula; no obstante, en algunos casos la complejidad de los cálculos hace necesaria la aparición de redundancias.
- En aquellos casos en los que no se ha logrado esta redundancia nula, será necesario vigilar que aquella información que aparece repetida se actualice de forma coherente, es decir, que todos los datos repetidos se actualicen de forma simultánea.
- La información almacenada en una base de datos puede llegar a tener un gran valor. Los SGBD deben garantizar que esta información se encuentra segura frente a usuarios malintencionados, que intenten leer información privilegiada; frente a ataques que deseen manipular o destruir la información; o simplemente ante las torpezas de algún usuario autorizado pero despistado. Normalmente, los SGBD disponen de un complejo sistema de permisos a usuarios y grupos de usuarios, que permiten otorgar diversas categorías de permisos.
- Se trata de adoptar las medidas necesarias para garantizar la validez de los datos almacenados. Es decir, se trata de proteger los datos ante fallos de hardware, datos introducidos por usuarios descuidados, o cualquier otra circunstancia capaz de corromper la información almacenada.
- Los SGBD deben proporcionar una forma eficiente de realizar copias de seguridad de la información almacenada en ellos, y de restaurar a partir de estas copias los datos que se hayan podido perder.
- Debe controlar el acceso concurrente a la información, que podría derivar en inconsistencias.[7]

1.2.2 Ventajas y desventajas.

Ventajas:

1. Facilidad de manejo de grandes volúmenes de información.
2. Gran velocidad en muy poco tiempo.



3. Independencia del tratamiento de información.
4. Seguridad de la información (acceso a usuarios autorizados), protección de información, de modificaciones, inclusiones, consulta.
5. No hay duplicidad de información, comprobación de información en el momento de introducir la misma.
6. Integridad referencial al terminar los registros.

Desventajas:

1. El mal diseño de esta puede originar problemas.
2. Un mal adiestramiento a los usuarios puede originar problemas.
3. Si no se encuentra un manual del sistema no se podrán hacer relaciones con facilidad.
4. Generan campos vacíos en exceso.
5. El mal diseño de seguridad genera problemas en esta.[7]

1.3 Tecnologías y Herramientas de desarrollo.

Teniendo en cuenta las necesidades vistas y las características del entorno donde se aplicará la solución propuesta, se realizó un estudio sobre las tecnologías y herramientas actuales posibles a emplear, descritas a continuación.

1.3.1 MySQL.

MySQL es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. MySQL fue creada por la empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca.



Aunque MySQL es software libre, MySQL AB distribuye una versión comercial de MySQL, que no se diferencia de la versión libre más que en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario, ya que de no ser así, se vulneraría la licencia GPL.

Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.[8]

Ventajas:

1. La velocidad a la hora de realizar las operaciones, lo que le hace uno de los gestores que ofrecen mayor rendimiento.
2. Su bajo consumo lo hacen apto para ser ejecutado en una máquina con escasos recursos sin ningún problema.
3. Las utilidades de administración de este gestor son envidiables para muchos de los gestores comerciales existentes, debido a su gran facilidad de configuración e instalación.
4. Tiene una probabilidad muy reducida de corromper los datos, incluso en los casos en los que los errores no se produzcan en el propio gestor, sino en el sistema en el que está.
5. El conjunto de aplicaciones Apache-PHP-MySQL es uno de los más utilizados en Internet en servicios de foro y de buscadores de aplicaciones.

Desventajas:

1. Carece de soporte para transacciones, rollback y subconsultas.
2. El hecho de que no maneje la integridad referencial, hace de este gestor una solución pobre para muchos campos de aplicación, sobre todo para aquellos programadores que provienen de otros gestores que sí que poseen esta característica.
3. No es viable para su uso con grandes bases de datos, a las que se acceda continuamente, ya que no implementa una buena escalabilidad.[8]



1.3.2 Microsoft SQL Server.

Microsoft SQL Server es un sistema de gestión de bases de datos relacionales (SGBD) basada en el lenguaje SQL, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea. Así de tener unas ventajas que a continuación se pueden describir.

Entre sus ventajas están:

- ✓ Soporte de transacciones.
- ✓ Gran estabilidad.
- ✓ Gran seguridad.
- ✓ Escalabilidad.
- ✓ Soporta procedimientos almacenados.
- ✓ Incluye también un potente entorno gráfico de administración, que permite el uso de comandos Lenguaje de Definición de Datos (DDL) y Lenguaje de Manipulación de Datos (DML) gráficamente.
- ✓ Permite trabajar en modo cliente-servidor donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo acceden a la información.
- ✓ Además permite administrar información de otros servidores de datos.

Desventajas:

- ✓ Tiene soporte solamente en el sistema operativo Windows.
- ✓ Es un software de licencia propietaria.

1.3.3 Oracle.

Es un manejador de base de datos relacional que hace uso de los recursos del sistema informático en todas las arquitecturas de hardware, para garantizar su aprovechamiento al máximo en ambientes cargados de información.

Ventajas:



- ✓ Posee igual interacción en todas las plataformas (Windows, Unix, Macintosh y Mainframes). Ya que más del 80% de los códigos internos de Oracle son iguales a los establecidos en todas las plataformas de Sistemas Operativos.
- ✓ Oracle soporta bases de datos de todos los tamaños, desde severas cantidades de bytes y gigabytes en tamaño.
- ✓ Soporte de transacciones.
- ✓ Es mucho más fácil la realización de backups (copias de seguridad).
- ✓ Es mucho más eficaz y eficiente.
- ✓ Tiene una amplia gama de herramientas para operar con la Base de Datos tanto como usuario y como Administrador.

Desventajas:

- ✓ Elevado precio comercial.
- ✓ La seguridad de la plataforma, y las políticas de suministro de parches de seguridad, modificadas a comienzos de 2005 y que incrementan el nivel de exposición de los usuarios.
- ✓ Exige una gran cantidad de recursos de la máquina donde se encuentre instalado.

1.3.4 PostgreSQL.

PostgreSQL es el servidor de bases de datos de código abierto más potente que existe y es por tanto la alternativa a MySQL cuando se necesitan características avanzadas como transacciones, procedimientos almacenados, triggers, vistas, etc.

Es el servidor de bases de datos más utilizado por los programadores de servlets de Java y, en general, por todos aquellos que realizan aplicaciones cliente-servidor complejas o críticas en el mundo Linux/Unix.

Ventajas:



- ✓ Posee una gran escalabilidad. Es capaz de ajustarse al número de Unidades Centrales de Procesamiento (CPUs) y a la cantidad de memoria que posee el sistema de forma óptima, haciéndole capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta.
- ✓ Implementa el uso de rollback, subconsultas y transacciones, haciendo su funcionamiento mucho más eficaz.
- ✓ Tiene la capacidad de comprobar la integridad referencial, así como también la de almacenar procedimientos en la propia base de datos.
- ✓ PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos.

Desventajas:

- ✓ Tiene un límite de 8K por fila, aunque se puede aumentar a 32K, con una disminución considerable del rendimiento.
- ✓ Es lento.[9]

1.3.5 Comparación entre los SGBD.

Tabla 1 Comparación de los SGBD.

Criterios	Sistemas Gestores de Bases de Datos			
	Oracle	SQL Server	MySQL	PostgreSQL
Plataforma	Windows/Linux	Windows	Windows/Linux	Windows/Linux
Velocidad	+	+	+	-
Volumen Datos	+	+	+	+
Integridad	+	+	-	+
Potencia	+	+	+	+
Coste	-	-	+	+
Requerimientos de hardware	-	+	+	+

+ Positivo - Negativo.

Dadas las características anteriores y la tabla de comparación, el SGBD que se escogió para desarrollar el sistema fue PostgreSQL, ya que SQL Server no es multiplataforma y además hay que pagar un importe



por su licencia, así mismo Oracle, incluso su coste es más elevado que el de SQL Server a parte de que el Oracle necesita una máquina con elevados recursos (memoria RAM, velocidad de microprocesador, capacidad de disco duro), en cuanto al MySQL se puede decir que cuando se almacena gran cantidad de datos estos pueden perderse o dañarse, además de que se necesitaba un SGBD que almacenara gran cantidad de información, por lo que se decidió escoger PostgreSQL que a pesar de ser el más lento, aventaja en los demás aspectos a los otros SGBD comparados.

1.3.6 Embarcadero Erwin Studio.

Es una herramienta de modelado de datos fácil de usar y multinivel, para el diseño y construcción de bases de datos a nivel físico y lógico. Direcciona las necesidades diarias de los administradores de bases de datos, desarrolladores y arquitectos de datos que construyen y mantienen aplicaciones de bases de datos grandes y complejos.

ER/Studio está equipado para crear y manejar diseños de bases de datos funcionales y confiables. Ofrece fuertes capacidades de diseño lógico, sincronización bidireccional de los diseños físicos y lógicos, construcción automática de bases de datos, documentación y fácil creación de reportes.

ER/Studio tiene las funcionalidades de sincronización bidireccional de los diseños lógico y físico, permite la construcción automática de base de datos, hacer reingeniería inversa de la base de datos. Tiene documentación basada en HTML además de un repositorio para el modelado. Entre los SGBD que soporta están Oracle, Sybase System, Microsoft SQL Server, IBM DB/2 Universal, InterBase 4, Microsoft Access y Microsoft Visual FoxPro.

1.3.7 Case Studio 2.

Herramienta para el diseño de bases de datos. Esta aplicación permite realizar Diagramas Entidad-Relación (DER) y Diagramas de Flujos de Datos (DFD) para distintos motores de base de datos. Algunos de éstos pueden ser: Oracle, DB2, InterBase, MS SQL, MySQL, MaxDB, Firebird y PostgreSQL entre otros. Otra de las características importantes es que permite realizar ingeniería inversa, o sea, a partir del modelo de tablas llegar al modelo lógico. Reduce los errores durante el desarrollo. Verifica el trabajo



automáticamente. Permite generar informes muy detallados en HTML o en formato rtf. Trabaja con las estructuras existentes de la base de datos. Crean bases de datos más rápidamente.[10]

Al crear ERD, el programa considera las opciones individuales tales como integridad de referencial, dominios de la base de datos, triggers, los procedimientos, las funciones, los procedimientos almacenados. Además permite realizar ingeniería inversa a las estructuras de datos ya creadas.[10]

De las herramientas mencionadas anteriormente, para el desarrollo de esta investigación se utiliza para el modelado de la Base de Datos la herramienta Case Studio 2, herramienta que aunque es nueva en el mercado soporta mayor número de SGBD a diferencia de Erwin, además se parte desde la elaboración del modelo físico, eliminándose la creación del modelo lógico utilizado en Erwin para luego obtener el modelo físico.[10]

1.4 Proceso de Desarrollo.

Cada día la producción de software busca adecuarse más a las necesidades del usuario, esto trae como consecuencia que aumente en tamaño y complejidad.

Para lograr la productividad del software se necesita un proceso que integre las múltiples facetas del desarrollo del mismo.

Se hace necesario definir la metodología de ingeniería del software que guiará el proceso de automatización, se ha escogido el Proceso Unificado de Desarrollo de Software (RUP).

Los orígenes de RUP se remontan al modelo espiral original de Barry Boehm. Ken Hartman, uno de los contribuidores claves de RUP colaboró con Boehm en la investigación. En 1995 Rational Software es comprada por una compañía sueca llamada Objectory AB. El Rational Unified Process fue el resultado de una convergencia de Rational Approach y Objectory, proceso desarrollado por el fundador de Objectory Ivan Jacobson. El primer resultado de esta fusión fue el Rational Objectory Process, la primera versión de RUP, fue puesta en el mercado en 1998, siendo el arquitecto en jefe Philippe Kruchten.

RUP es un proceso de desarrollo de software, o sea, conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software. Es un marco de trabajo genérico que puede



especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyectos.

Es un proceso basado en componentes, que utiliza el Lenguaje Unificado de Modelado (UML) para preparar todos los esquemas de un sistema de software. No obstante, los verdaderos aspectos definitorios del Proceso Unificado se resumen en que está dirigido por casos de uso, este avanza a través de una serie de flujos de trabajo que parten de los casos de uso; centrado en la arquitectura y es iterativo e incremental.

Está acompañado de una herramienta que soporta cada uno de los procesos que se necesitan: Rational Rose Enterprise Edition 2003.

Además cubre el ciclo de vida de desarrollo de un proyecto y toma en cuenta las mejores prácticas a utilizar en el modelo de desarrollo de software.

1.4.1 Lenguaje Unificado de Modelado (UML).

Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

Es importante remarcar que UML es un lenguaje para especificar y no un método o un proceso, se utiliza para definir un sistema de software, para detallar los artefactos en el sistema y para documentar y construir pero no especifica en sí mismo qué metodología o proceso usar.

UML cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos de las entidades representadas. Un modelo UML indica que es lo que supuestamente hará el sistema pero no como lo hará.

De forma general las principales características son:

- ✓ Lenguaje unificado para la modelación de sistemas.



- ✓ Tecnología orientada a objetos.
- ✓ El cliente participa en todas las etapas del proyecto.
- ✓ Corrección de errores viables en todas las etapas.
- ✓ Aplicable para tratar asuntos de escala inherentes a sistemas complejos de misión crítica, tiempo real y cliente/servidor.

Existen varias herramientas CASE (Computer-Aided Systems Engineering), que dan asistencia a analistas, ingenieros de software y desarrolladores durante el ciclo de vida de desarrollo de un software, pero es Rational Rose líder en el modelado del desarrollo de los proyectos y es esta precisamente la que se utiliza en la modelación de este proyecto. La herramienta fue desarrollada por los creadores de UML, utilizando la notación estándar en la arquitectura de software. Esta herramienta integra todos los elementos que propone la metodología RUP para cubrir el ciclo de vida de un proyecto.

1.4.2 Rational Rose.

Rational Rose es la herramienta CASE desarrollada por los creadores de UML (Booch, Rumbaugh y Jacobson), que cubre todo el ciclo de vida de un proyecto: Concepción y formalización del modelo, construcción de los componentes, transición a los usuarios y certificación de las distintas fases y entregables.

Facilita el desarrollo de un proceso cooperativo en el que todos los agentes tienen sus propias vistas de información (vista de Casos de Uso, vista Lógica, vista de Componentes y vista de Despliegue), pero utilizan un lenguaje común para comprender y comunicar la estructura y la funcionalidad del sistema en construcción.

Esta herramienta propone la utilización de cuatro tipos de modelos para realizar el diseño del sistema, utilizando una vista estática y otra dinámica de los modelos del sistema, uno lógico y otro físico. Permite crear y refinar estas vistas creando de esta forma un modelo completo que representa el dominio del problema y el sistema de software.

Rational Rose utiliza un proceso de desarrollo iterativo controlado, donde el desarrollo se lleva a cabo en una secuencia de iteraciones. Cada iteración comienza con una primera aproximación del análisis, diseño



e implementación para identificar los riesgos del diseño, los cuales se utilizan para conducir la iteración, primero se identifican los riesgos y después se prueba la aplicación para que estos se hagan mínimos.

Se puede generar código en distintos lenguajes de programación a partir de un diseño en UML.

Rational Rose proporciona mecanismos para realizar la denominada Ingeniería Inversa, es decir, a partir del código de un programa, se puede obtener información sobre su diseño.

1.5 Otras herramientas.

1.5.1 EMS SQL Manager para PostgreSQL.

EMS SQL Manager for PostgreSQL es una aplicación de alto desempeño para la administración y desarrollo de PostgreSQL. El programa trabaja con cualquier versión de PostgreSQL hasta la 8.0 y soporta todas las últimas características de PostgreSQL, incluyendo espacios de tablas, nombres de argumentos en funciones y más. El programa ofrece muchas herramientas poderosas para usuarios experimentados, como un Diseñador Visual de Bases de Datos y un Constructor Visual de Consultas. [11]

Características:

- ✓ Soporte completo para PostgreSQL hasta la versión 8.1.
- ✓ Administración y navegación rápida de bases de datos.
- ✓ Administración fácil de todos los objetos PostgreSQL.
- ✓ Herramientas de manipulación avanzada de datos.
- ✓ Administración efectiva de seguridad.
- ✓ Excelente herramientas visuales y de texto para la construcción de consultas.
- ✓ Capacidades de exportación e importación de datos.
- ✓ Poderoso diseñador visual de bases de datos.
- ✓ Modo guiado para labores de mantenimiento.
- ✓ Interfaz atractiva.[11]



1.5.2 EMS SQL Query para PostgreSQL.

Es una utilidad que permite de forma rápida y simple construir consultas SQL para PostgreSQL, desde su interfaz visual pueden contruirse las consultas y es posible editarlas directamente en esta, puede conectarse a las bases de datos para seleccionar las tablas y los campos para las consultas, se puede trabajar con varias consultas a las vez y los resultados de las consultas pueden visualizados en diferentes modos.[12]

Características:

- ✓ Trabaja con varias consultas en difrentes ventanas.
- ✓ Permite la conección al mismo tiempo a varias bases de datos.
- ✓ Tiene un sistema avanzado de impresión.
- ✓ Permite calcular el tiempo de ejecución de las consultas.
- ✓ Permite la creación de diagramas basados en las consultas.
- ✓ Guarda un historial de todas las consultas hechas.[12]

1.5.3 PgAdmin 3.

Es una interfaz comprensible para el diseño y administración de una base de datos PostgreSQL, diseñada para ejecutarse en la mayoría de los Sistemas Operativos. La aplicación corre bajo GNU/Linux, FreeBSD y Windows 2000/XP.[13]

Características:

- ✓ Un agente de SQL/shell para tareas programadas.
- ✓ Soporte para PostgreSQL 8.1
- ✓ Autovacuum administration.
- ✓ Roles.
- ✓ Procedimientos almacenados.



- ✓ Transacciones preparadas.
- ✓ Permite registrar un servidor sin establecer conexion.
- ✓ Ayuda para usuarios nuevos.
- ✓ Las contraseñas pueden ser almacenadas en los ficheros .pgass/pgpass.conf.[13]

Conclusiones.

En este capítulo se determinaron las tecnologías más apropiadas para el desarrollo del sistema, donde se propone, teniendo en cuenta las tendencias actuales, la construcción de la base de datos utilizando PostgreSQL y para el modelado de los datos debido a sus prestaciones y compatibilidad con el PostgreSQL, se decidió utilizar el Case Studio 2. Para llevar a cabo el análisis y diseño se propuso el uso de la metodología RUP utilizando el lenguaje de modelado UML.



Introducción.

En el presente capítulo se explica como se lleva a cabo la comunicación entre los diferentes módulos del sistema, así como con los sistemas externos al mismo, se hará una breve descripción de la arquitectura, una selección de los principales requisitos funcionales y se determinarán las características que deberá tener la Base de Datos. Como resultado de esto se obtendrá:

- ✓ Diagrama de clases persistentes.
- ✓ Descripción de las clases.
- ✓ Modelo de objetos.
- ✓ Diagrama Entidad Relación de la Base de Datos.
- ✓ Descripción de las tablas.

2.1 Integración con otros módulos y sistemas.

La comunicación de los módulos docente, no docente, planificación docente e investigación con la base de datos va a ser a través de funciones, las mismas van a tener las funcionalidades necesarias, para que los módulos no accedan directamente a la base de datos si no a través de las mismas, en cuanto a la comunicación con otros sistemas, la base de datos va a tener la posibilidad de acceder a los datos que controla el sistema de gestión Akademos de la cual va a obtener un grupo de datos docentes y además va a tener comunicación con el sistema que gestiona los datos de las personas de la UCI, y esto va a ser a través de servicios web que brinden esos sistemas, los cuales llenaran la base de datos y la mantendrán actualizada.

2.2 Descripción de la arquitectura y fundamentación.

La Base de Datos propuesta contiene un esquema donde se encuentran 128 tablas que se utilizan en nueve módulos. Las más comunes entre todos los módulos son la tabla persona, estudiante, profesor,



trabajador y docente. El acceso a la Base de Datos será a través de peticiones que el servidor de páginas Web le hará al servidor de Base de Datos, donde existirán las funciones encargadas de dar respuesta a dichas peticiones.

A continuación un ejemplo de cómo se atienden las conexiones en un servidor de Postgres:

Ejemplo de sesiones simultáneas

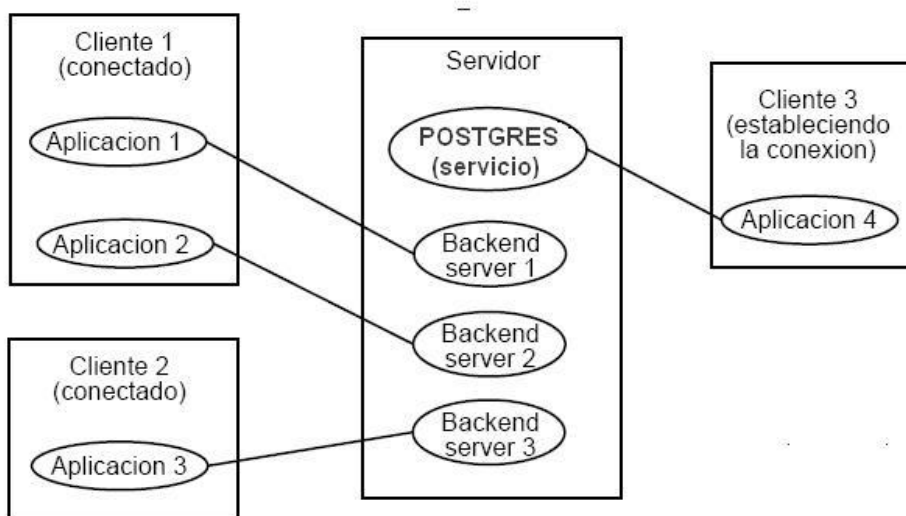


Figura 1.- Diagrama de Peticiones a un Servidor de Base de Datos Postgres.

El cliente 1 y 2 están conectados, por cada petición que se le haga a la base de datos se establecerá una conexión en espera de una petición de fin de la conexión. El cliente 3 está pidiendo un servicio de conexión, cuando dicha conexión se establezca, el servidor quedará en espera del fin de la conexión.

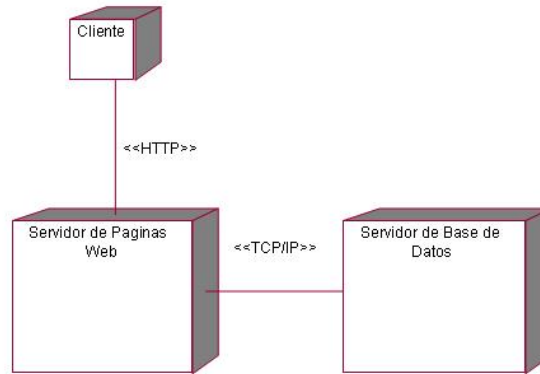


Figura 2.- Diagrama de despliegue.

2.3 Análisis de optimización de Consultas.

Las consultas son pedidos que se le hacen a la Base de Datos, para obtener información que hay guardada en ella. Durante la ejecución de las consultas se pasa por 4 fases:

1. Parser: Chequea la sintaxis de la consulta y genera el árbol de consulta (Query tree).
2. Rewriter: Procesa el árbol de consulta y comprueba la existencia de reglas reescribiendo de nuevo la consulta si es necesario.
3. Planer: Genera el plan de consulta a partir del árbol de consulta analizando todos los posibles caminos y escogiendo el de menor coste.
4. Executor: Recorre recursivamente el árbol del plan de consulta y accede al disco para obtener los datos necesarios. Realiza las uniones entre tablas, cálculos y ordenamientos necesarios para devolver las tuplas resultado de las consultas.[14]



Realización de una consulta

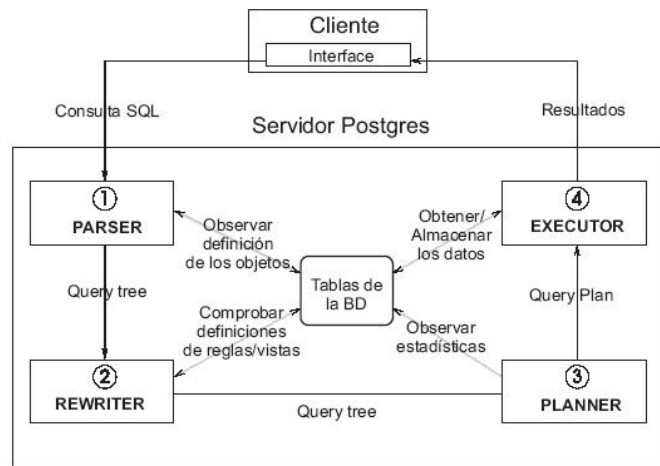


Figura 3. Diagrama de ejecución de una consulta en postgres.

La optimización de consultas es algo necesario para un mejor funcionamiento y rapidez de la Base de Datos, esto consiste en buscar las mejores opciones para realizar consultas.

Una de las medidas para obtener un mejor rendimiento en la Base de Datos es la creación de índices. La clave de un índice es un conjunto de atributos. El índice permite, dado un valor de cada uno de los atributos del conjunto, acceder todos los registros de la tabla sobre la cual se construye el índice que tienen esos valores en los atributos de la clave del índice.

A continuación se presentan algunos aspectos específicos donde se puede mejorar el rendimiento de una consulta:

1. La cláusula DISTINCT es costosa de ejecutar porque generalmente involucra un ordenamiento de las tuplas resultantes para eliminar duplicados. Es necesario analizar si realmente hace falta usar esa cláusula.
2. Las subconsultas en muchos manejadores se ejecutan ineficientemente. Una razón para ello es que al estar anidada no se utiliza algún índice relevante, al eliminar la subconsulta es posible lograr que el manejador utilice el índice apropiado.



3. En algunas consultas el usuario almacena resultados intermedios explícitamente en tablas temporales. Estas tablas pueden bajar el rendimiento de la consulta por dos razones, una porque fuerza un orden de ejecución que quizás no sea ideal y segundo, porque obliga a ejecutar actualizaciones al Diccionario de Datos (DD) lo cual tiene el peligro de convertirlo en un cuello de botella.
4. Por otro lado, las tablas temporales pueden tener un efecto positivo al reescribir consultas que contienen subconsultas correlacionadas complejas. Las subconsultas correlacionadas pueden ejecutarse ineficientemente, pero si se logra calcular primero las tuplas de la subconsulta y almacenarlas en una tabla temporal, se puede reescribir la consulta original, sin la subconsulta, accediendo a la tabla temporal.
5. Otro uso beneficioso de las tablas temporales es para evitar el uso de la cláusula ORDER BY que puede ser costosa.
6. Las condiciones de join se pueden evaluar más eficientemente contra un índice primario. Y en general, en términos de rendimiento es preferible evaluar una condición de igualdad numérica que una condición de igualdad sobre cadenas ("strings") de caracteres.
7. Es preferible no usar la cláusula HAVING si la condición deseada se puede expresar en la cláusula WHERE.
8. Es importante conocer las particularidades del manejador, esto se refiere a conocer como han sido implementadas las rutinas de procesamiento de las consultas. Por ejemplo, es importante saber, al existir una condición disyuntiva (con OR) en la consulta si se usan o no los índices existentes.
9. Otra particularidad importante del manejador que es bueno conocer es si el orden de las tablas en la cláusula FROM puede afectar la implementación del join utilizada, posiblemente esto es relevante para joins que involucran 5 tablas o más.
10. El uso de vistas pueden causar una ejecución ineficiente de consultas. Muchas veces la ejecución de consultas sobre las tablas base es más eficiente. [9]

Se trabajará con un ejemplo que utilizará un campo datetime usando la herramienta 'Analizador de Consultas SQL' que se llama explain consulta para mostrar datos acerca del tiempo de ejecución:



Con una consulta sencilla como esta:

```
SELECT
    acuerdosujc.id_acuerdo,
    acuerdosujc.numero,
    acuerdosujc.fechacumplimiento,
    acuerdosujc.responsable,
    acuerdosujc.acuerdo
FROM
    actaujc
INNER JOIN acuerdosujc ON ( actaujc.numero = acuerdosujc.numero)
WHERE
    (date_part('Year', actaujc.fecha) = '2007') AND
    (date_part('Month', actaujc.fecha) = '05')
```

Teniendo más de 100 000 registros en la tabla donde se busca, esta consulta sencilla se demoró 0.22 segundos. Ahora se probará otra forma de hacer esto.

```
SELECT
    acuerdosujc.id_acuerdo,
    acuerdosujc.numero,
    acuerdosujc.fechacumplimiento,
    acuerdosujc.responsable,
    acuerdosujc.acuerdo
FROM
    actaujc
```



```
INNER JOIN acuerdosujc ON ( actaujc.numero = acuerdosujc.numero)
```

```
WHERE
```

```
( actaujc.fecha LIKE '2007-05%')
```

Esto salió mucho peor con un tiempo de 0.28 segundos; 0.06 segundos con un peor resultado. Se probará ahora otra manera de hacer esta consulta:

```
SELECT
```

```
acuerdosujc.id_acuerdo,  
acuerdosujc.numero,  
acuerdosujc.fechacumplimiento,  
acuerdosujc.responsable,  
acuerdosujc.acuerdo
```

```
FROM
```

```
actaujc
```

```
INNER JOIN acuerdosujc ON ( actaujc.numero = acuerdosujc.numero)
```

```
WHERE
```

```
( actaujc.fecha BETWEEN '2007-05-01 00:00:00' AND '2007-05-29 23:59:59')
```

Se obtiene un resultado de 0.20 segundos

Analizando otro tipo de consulta sencilla donde la tabla acuerdosujc está indexada por su llave id_acuerdo.

```
SELECT
```

```
acuerdosujc.id_acuerdo,  
acuerdosujc.numero,  
acuerdosujc.fechacumplimiento,  
acuerdosujc.responsable,
```



```
acuerdosujc.acuerdo
```

```
FROM
```

```
    acuerdosujc
```

```
WHERE
```

```
    ( acuerdosujc.id_acuerdo >= 1000) AND
```

```
    ( acuerdosujc.id_acuerdo < 2000)
```

Se obtiene un resultado de 1000 tuplas en un tiempo de 1.34 segundos.

Para esta otra:

```
SELECT
```

```
    acuerdosujc.id_acuerdo,
```

```
    acuerdosujc.numero,
```

```
    acuerdosujc.fechacumplimiento,
```

```
    acuerdosujc.responsable,
```

```
    acuerdosujc.acuerdo
```

```
FROM
```

```
    acuerdosujc
```

```
WHERE
```

```
    ( acuerdosujc.id_acuerdo BETWEEN 1000 AND 2000)
```

Se obtienen 1000 tuplas en un tiempo 0.25 segundos.

De los resultados obtenidos se puede concluir que si se tiene un dominio bastante amplio de las maneras de hacer las consultas y con las herramientas que permitan calcular tiempos de ejecución de dichas consultas, se pueden hacer peticiones al servidor de Base de Datos con mejores resultados en cuanto al tiempo de respuesta.



2.4 Selección de los requisitos funcionales.

Requisitos funcionales de UJC:

RF 1 Gestionar Acta de la UJC.

- 1.1 Crear acta de un Comité de Base.
- 1.2 Listar actas de los Comités de Base.
 - 1.2.1 Listar actas de todos los Comités de Base de un año específico en un mes específico.
 - 1.2.2 Listar actas de todos los meses de un Comité de Base específico.
 - 1.2.3 Obtener acta de un Comité de Base específico en un mes específico.
- 1.3 Modificar acta de un Comité de Base.
- 1.4 Eliminar acta de un Comité de Base.

RF 2 Listar Acuerdos.

- 2.1 Listar acuerdos de un año específico en un mes específico.
- 2.2 Listar acuerdos de un Comité de Base específico en un mes específico.

RF 3 Listar Militantes de la UJC.

- 3.1 Listar militantes de un Comité de Base específico.

RF 4 Listar Universo Juvenil.

- 4.1 Listar universo juvenil de un Comité de Base específico.

RF 5 Listar Militantes Sancionados.

- 5.1 Listar militantes sancionados de todos los Comités de Base.
- 5.2 Listar militantes sancionados de un año específico.
- 5.3 Listar militantes sancionados de un Comité de Base específico.

RF 6 Gestionar Militante de la UJC.

- 6.1 Adicionar Militante.



6.1 Eliminar Militante.

RF 7 Gestionar Rol de los Militantes.

7.1 Asignar Rol a un militante.

7.1 Modificar Rol a un militante.

RF 8 Gestionar Militantes Sancionados.

8.1 Buscar Militante.

8.1.1 Buscar por nombre, apellidos y Comité de Base.

Requisitos funcionales del Sindicato:

RF 9 Gestionar Acta del Sindicato.

9.1 Crear acta de un Comité Sindical.

9.2 Listar actas de todos los Comité Sindical en un mes específico.

9.3 Modificar acta de un Comité Sindical.

9.4 Eliminar acta de un Comité Sindical.

RF 10 Listar Acuerdos.

10.1 Listar acuerdos de todos los Comité Sindical en un mes específico.

10.2 Listar acuerdos de un Comité Sindical específico en un mes específico.

RF 11 Gestionar Emulación.

11.1 Buscar Trabajador.

11.1.1 Buscar trabajador por nombre y apellidos.

11.1.2 Buscar trabajador por Comité Sindical.

11.2 Definir cumplidores e incumplidores en un mes.

11.3 Definir cumplidores o destacados trimestralmente.

11.4 Definir destacados semestralmente y anualmente por rubros emulativos.



11.5 Modificar cumplidores o incumplidores, mensualmente o semestralmente.

11.6 Buscar destacados por rubros.

RF 12 Gestionar Trabajador.

12.1 Adicionar trabajador.

12.2 Buscar trabajadores.

12.2.1 Buscar trabajador por nombre, apellidos y Comité Sindical.

12.3 Modificar trabajador.

12.4 Eliminar trabajador.

Requisitos funcionales de Extensión Universitaria:

RF 13 Gestionar Edificios.

13.1 Adicionar edificios.

13.2 Listar edificios.

13.2.1 Modificar datos de un edificio.

13.2.2 Eliminar edificio.

RF 14 Gestionar Apartamentos.

14.1 Adicionar apartamentos.

14.2 Listar apartamentos.

14.2.1 Modificar datos de un apartamento.

14.2.2 Eliminar apartamento.

RF 15 Gestionar ubicación de estudiantes en la residencia.

15.1 Ubicar uno o varios estudiantes en un apartamento.

15.2 Eliminar ubicación de uno o varios estudiantes en un apartamento.

RF 16 Generar reporte de ubicación de estudiantes.



- 16.1 Generar reporte de ubicación de estudiantes por edificio.
- 16.2 Generar reporte de ubicación de estudiantes por paso de escalera.
- 16.3 Generar reporte de ubicación de estudiantes por apartamento.

RF 17 Realizar encuesta a estudiantes.

- 17.1 Realizar encuesta a estudiantes sobre los deportes que practica.
- 17.2 Realizar encuesta a estudiantes sobre las manifestaciones culturales que practica.

RF 18 Generar reporte de estudiantes.

- 18.1 Generar reporte de estudiantes que practican un deporte determinado.
- 18.2 Generar reporte de estudiantes que practican una manifestación cultural determinada.

RF 19 Buscar estudiante.

- 19.1 Buscar estudiante.
 - 19.1.1 Obtener datos personales.

RF 20 Gestionar reporte de la guardia estudiantil.

- 20.1 Adicionar reporte.
- 20.2 Listar reportes.
 - 20.2.1 Consultar datos de un reporte.
 - 20.2.2 Eliminar reporte.

RF 21 Gestionar caracterización del estudiante.

- 21.1 Listar estudiantes.
 - 21.1.1 Registrar participación del estudiante en actividades políticas, de la residencia y extensión universitaria.
 - 21.1.2 Modificar participación del estudiante en actividades políticas, de la residencia y extensión universitaria.

RF 22 Obtener caracterización del estudiante.



22.1 Listar estudiantes.

22.1.1 Obtener caracterización del estudiante.

Requisitos funcionales de Planificación Docente:

RF 23 Gestionar horarios.

23.1 Buscar horario de grupos y profesores.

23.2 Modificar horario de grupos.

23.3 Crear horario de grupos.

23.4 Enviar horario.

23.5 Registrar asignatura.

RF 24 Gestionar Afectaciones

24.1 Registrar afectaciones.

24.2 Buscar afectaciones.

24.3 Modificar afectaciones.

24.4 Eliminar Afectaciones.

RF 25 Gestionar P1

25.1 Registrar P1.

25.2 Modificar P1.

25.3 Buscar P1.

RF 26 Gestionar Local

26.1 Modificar Local.

26.2 Buscar Horario Local.

26.3 Registrar Local.

RF 27 Mostrar Horario



RF 28 Mostrar P1.

RF 29 Mostrar Afectaciones.

RF 30 Mostrar Horario Local.

Requisitos funcionales de Investigación:

RF 31 Gestionar Evento Científico.

- 31.1 Insertar datos de un Evento.
- 31.2 Modificar datos de un Evento.
- 31.3 Guardar resultados de los participantes de un evento.

RF 32 Gestionar Comisión.

- 32.1 Insertar datos de una comisión.
- 32.2 Modificar datos de una comisión.
- 32.3 Eliminar datos de una comisión.

RF 33 Gestionar participantes.

- 33.1 Insertar participantes en un evento.
- 33.2 Eliminar participantes en un evento.

RF 34 Gestionar Trabajo.

- 34.1 Insertar trabajo en un evento.
- 34.2 Modificar trabajo de un evento.

RF 35 Gestionar Publicación

- 35.1 Insertar datos de una publicación.
- 35.2 Modificar datos de una publicación.

RF 36 Mostrar Reporte de publicaciones.

- 36.1 Mostrar publicaciones según el nombre del autor.



36.2 Mostrar publicaciones según algún criterio de búsqueda.

RF 37 Mostrar Reportes de Eventos.

37.1 Mostrar los eventos en que ha participado una persona.

37.2 Mostrar listado de eventos según un criterio de búsqueda.

Requisitos funcionales de Proyectos Productivos:

RF 38 Gestionar Línea de Investigación.

38.1 Insertar datos de una Línea de Investigación.

38.2 Modificar datos de una Línea de Investigación.

RF 39 Gestionar Proyecto.

39.1 Insertar datos de un Proyecto.

39.2 Modificar datos de un Proyecto.

39.3 Insertar datos del cliente de un proyecto.

39.4 Modificar datos del cliente de un proyecto.

39.5 Asignar estudiantes a un proyecto.

39.6 Dar baja a un estudiante de un proyecto.

39.7 Asignar laboratorio a un proyecto.

RF 40 Gestionar Grupos de Trabajo.

40.1 Insertar Grupos de Trabajo.

40.2 Modificar Grupos de Trabajo.

RF 41 Gestionar Tarea.

41.1 Insertar datos de una Tarea.

41.2 Modificar datos de una Tarea.

RF 42 Gestionar Tesis.



42.1 Insertar datos de una Tesis.

42.2 Modificar datos de una Tesis.

RF 43 Generar Reportes de una Línea de Investigación.

43.1 Listar proyectos que integran una línea de investigación.

RF 44 Generar Reportes de proyectos.

44.1 Mostrar los datos de un proyecto.

44.2 Listar locales de producción de un proyecto.

44.3 Listar grupos de trabajos que trabajan en un proyecto.

44.4 Listar clientes que se relacionen con proyectos productivos.

44.5 Listar datos de estudiantes en proyectos.

44.6 Listar tesis que se encuentran en un proyecto específico.

RF 45 Generar reportes de Grupos de trabajos.

45.1 Listar tareas que realiza un Grupo en específico.

45.2 Listar datos de estudiantes que se encuentren en grupos de trabajos.

RF 46 Generar reportes de tesis.

46.1 Listar datos de los estudiantes y sus tesis.

Requisitos funcionales de Cursos Optativos:

RF 47 Gestionar matrícula.

47.1 Matricular en un curso optativo.

47.2 Matricular en un perfil.

47.3 Vaciar la capacidad del curso optativo una vez terminado para volverlo a ofertar.

RF 48 Gestionar curso.

48.1 Actualizar curso.



- 48.2 Modificar curso.
- 48.3 Listar los cursos.
- 48.4 Permitir ver los detalles de un curso.
- 48.5 Mostrar cantidad de cursos.

RF 49 Gestionar perfil.

- 49.1 Actualizar perfil.
- 49.2 Insertar perfil.
- 49.3 Eliminar perfil.
- 49.4 Listar perfil.
- 49.5 Mostrar cantidad de perfiles.

RF 50 Gestionar información.

- 50.1 Listar cursos y total de cursos por estudiante.
- 50.2 Listar cursos y total de cursos por grupo.
- 50.3 Listar cursos y total de cursos por profesor.
- 50.4 Listar estudiantes y total de estudiantes por curso.
- 50.5 Listar estudiantes y total de estudiantes con n cursos.
- 50.6 Listar estudiantes recibiendo un curso determinado.
- 50.7 Mostrar si el estudiante con cierto curso lo tiene acreditado.
- 50.8 Listar perfil y asignaturas del mismo por estudiante.
- 50.9 Listar perfil y asignaturas del mismo por grupo.
- 50.10 Mostrar el por ciento de estudiantes que han culminado su perfil por grupo.
- 50.11 Mostrar el por ciento de estudiantes que han culminado su perfil por año.
- 50.12 Mostrar el por ciento de estudiantes que han culminado su perfil por facultad.



50.13 Listar estudiantes y total de estudiantes con un perfil vencido.

Requisitos funcionales de Docencia.

RF 51 Gestionar departamentos docentes.

51.1 Registrar departamentos.

51.2 Listar departamentos.

51.3 Listar personal de los departamentos.

51.4 Modificar datos de los departamentos.

51.5 Eliminar departamentos.

RF 52 Gestionar evaluación del personal docente.

52.1 Registrar evaluación.

52.2 Listar evaluación.

52.3 Modificar evaluación del profesor.

RF 53 Gestionar plan de trabajo.

53.1 Registrar plan de trabajo.

53.2 Listar plan de trabajo

53.3 Modificar plan de trabajo.

RF 54 Gestionar visitas a clases.

54.1 Registrar visita a turno de clase.

54.2 Registrar criterios de visita a clases.

54.3 Registrar evaluación de la visita a clases.

54.4 Listar visitas a clases.

54.5 Modificar visitas a clases.

RF 55 Gestionar resoluciones decanales.



55.1 Registrar resoluciones decanales.

55.2 Listar resoluciones.

55.3 Modificar resoluciones.

55.4 Eliminar resoluciones decanales.

RF 56 Gestionar aplicación de resoluciones decanales.

56.1 Registrar aplicación de las resoluciones.

56.2 Listar personal afectado.

56.3 Modificar resolución aplicada.

2.5 Diagrama de clases persistentes y modelo de objetos.

Debido a la cantidad de clases que conformaban el diagrama principal y para una mayor comprensión del mismo, se decidió hacer un diagrama por cada módulo.

Los modelos de datos se encuentran en el Anexo 3.

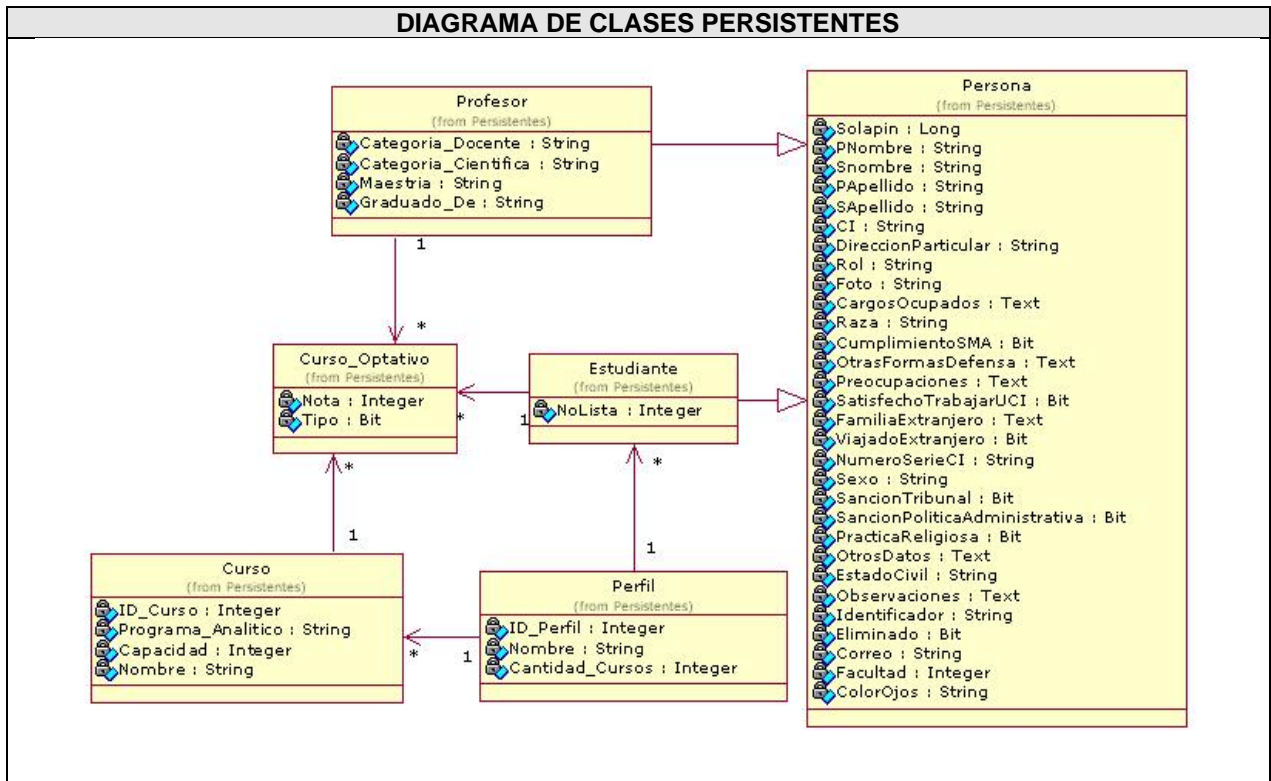


Figura 3.- Diagrama de clases persistentes del módulo Cursos Optativos.



DIAGRAMA DE CLASES PERSISTENTES

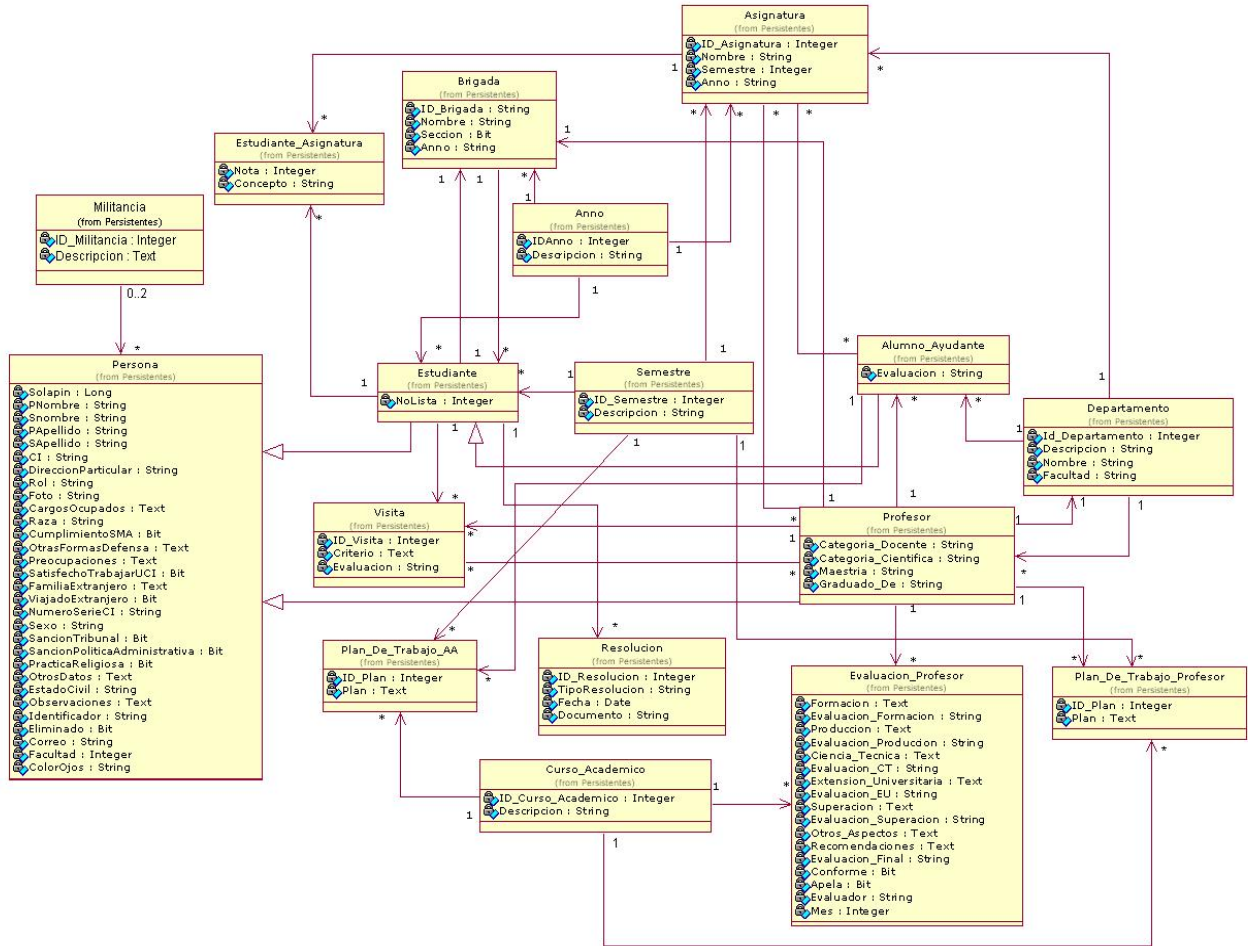


Figura 4.- Diagrama de clases persistentes del módulo Docencia.

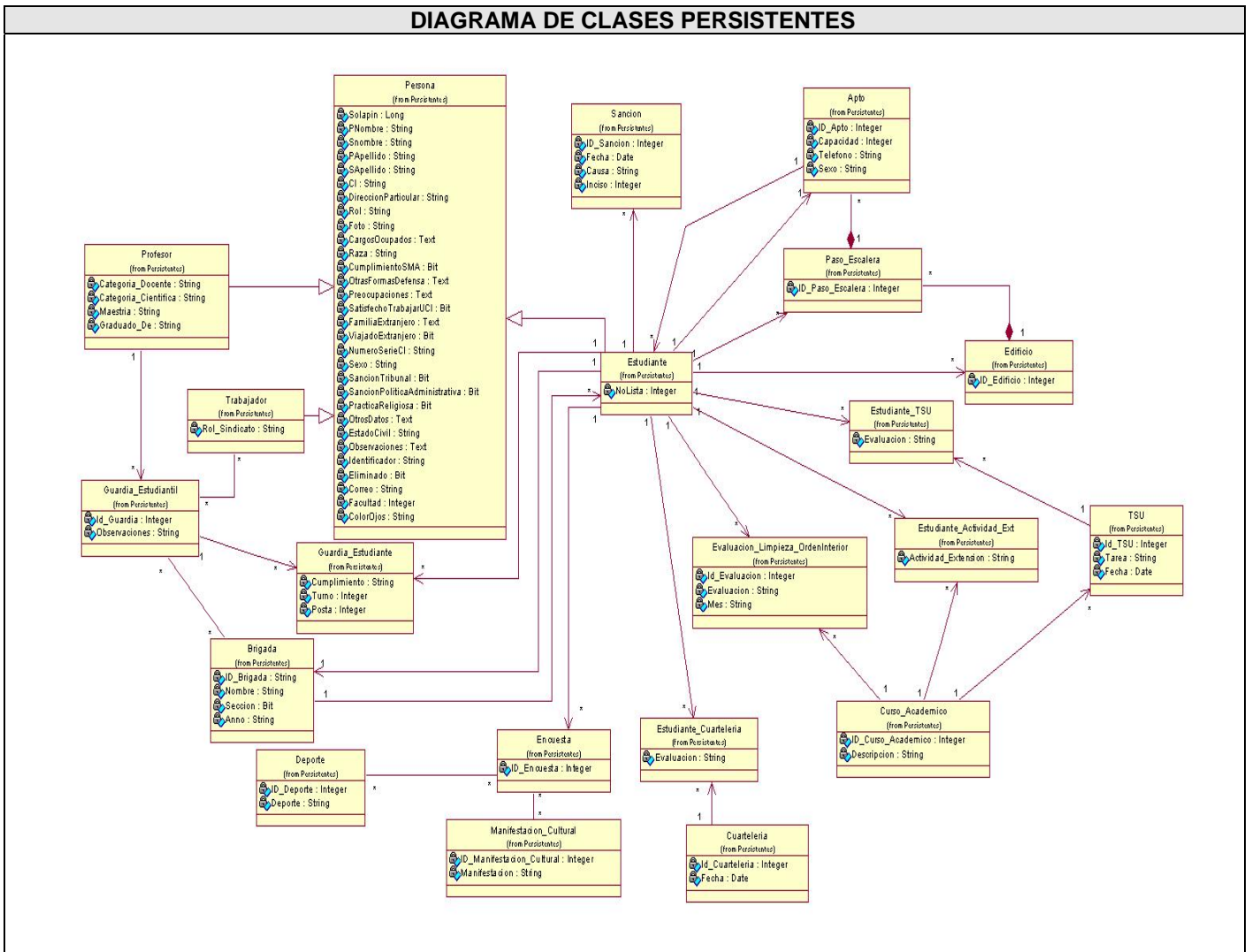


Figura 5.- Diagrama de clases persistentes del módulo Extensión Universitaria.



CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

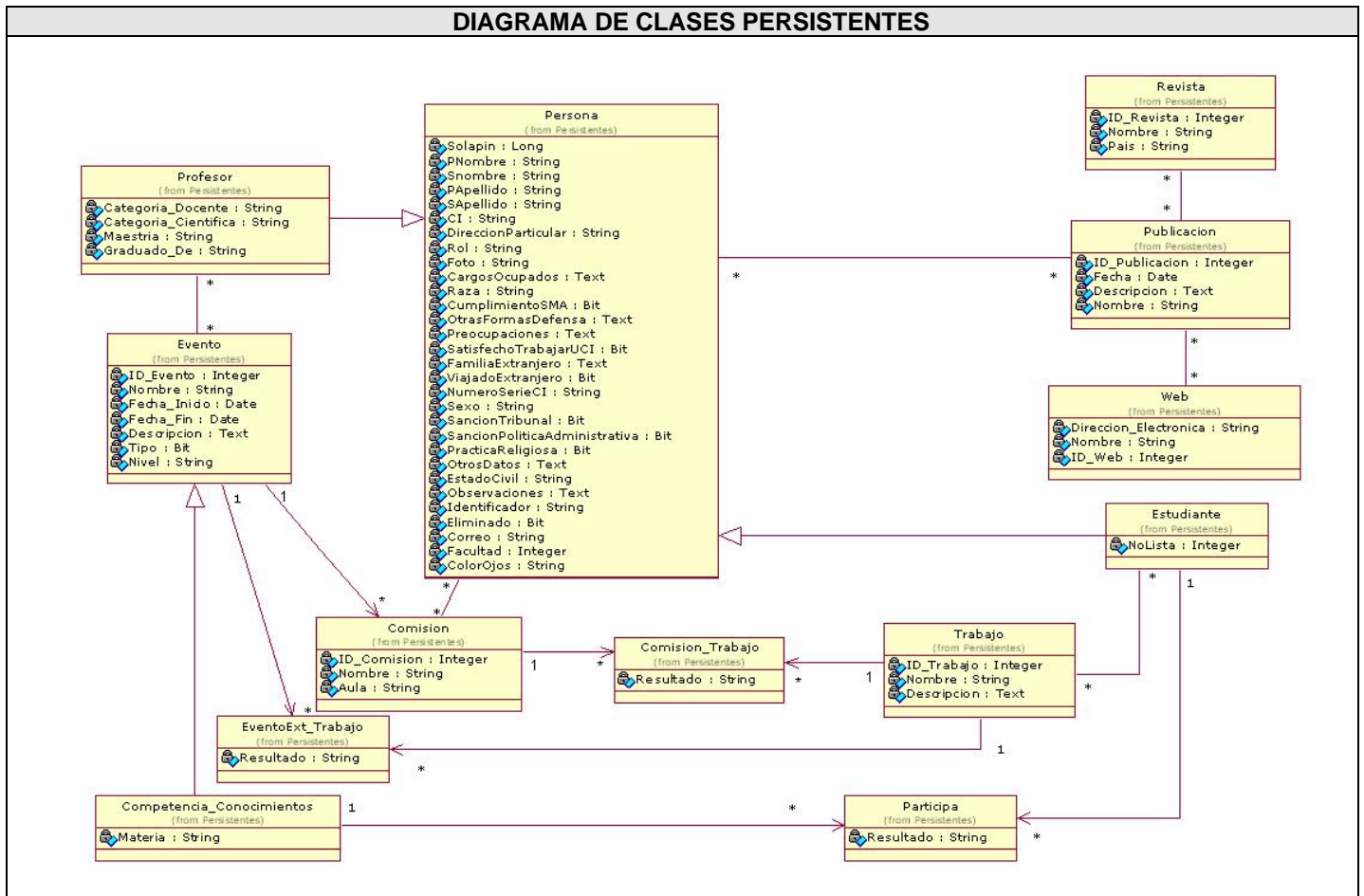


Figura 6.- Modelo de clases persistentes del módulo de Investigación.

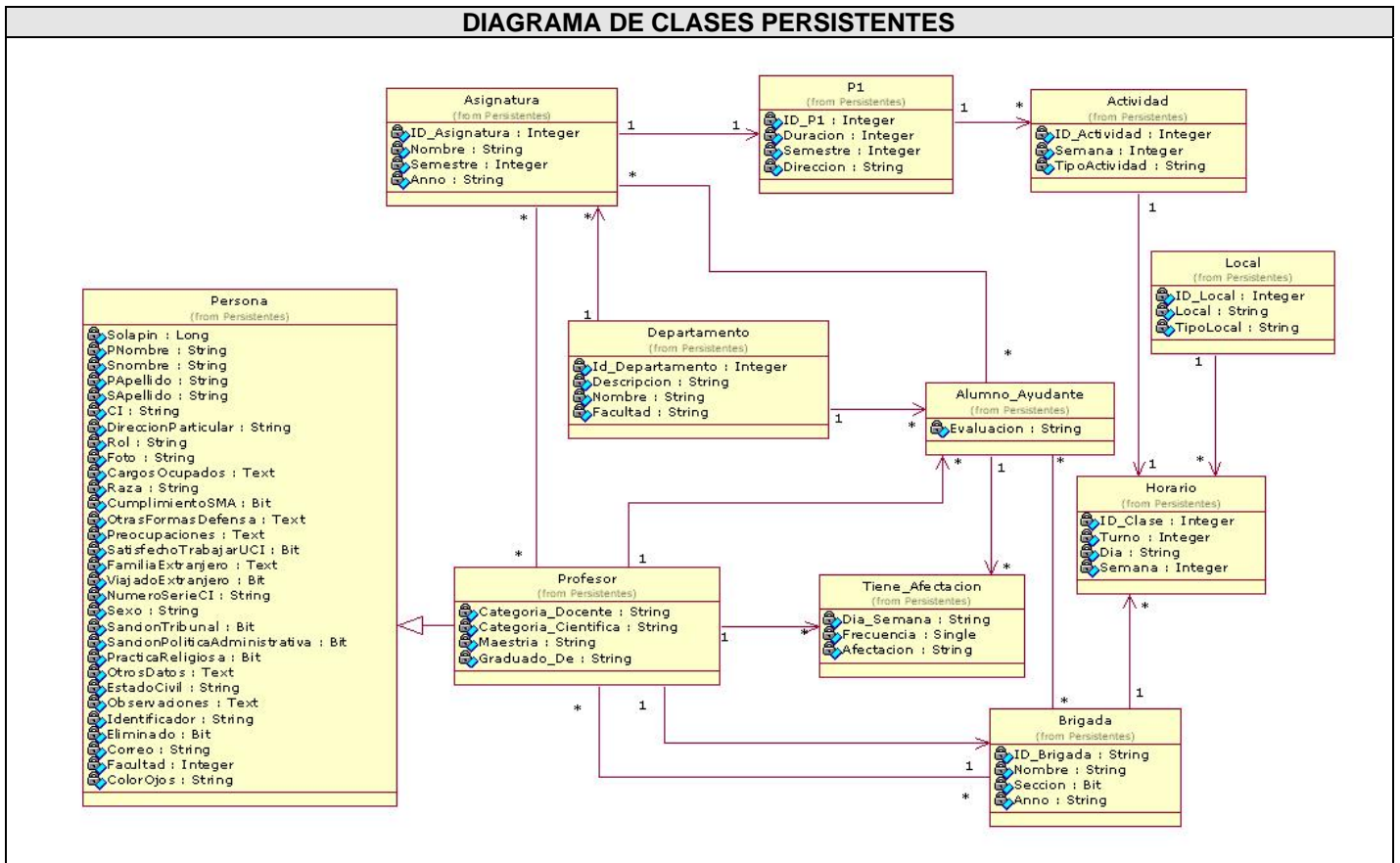


Figura 7.- Diagrama de clases persistentes del módulo Planificación Docente.

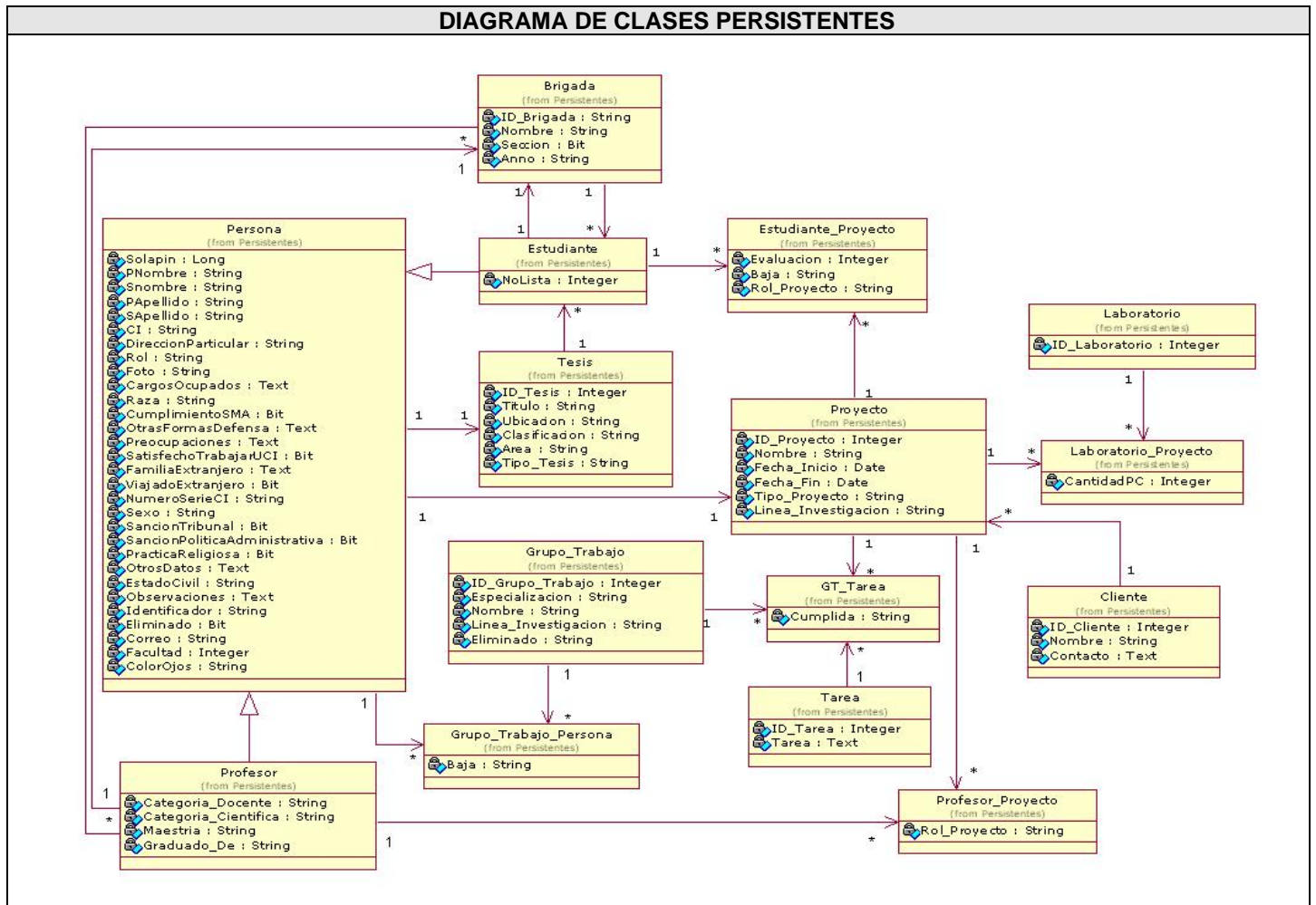


Figura 8.- Diagrama de clases persistentes del módulo Proyectos Productivos.

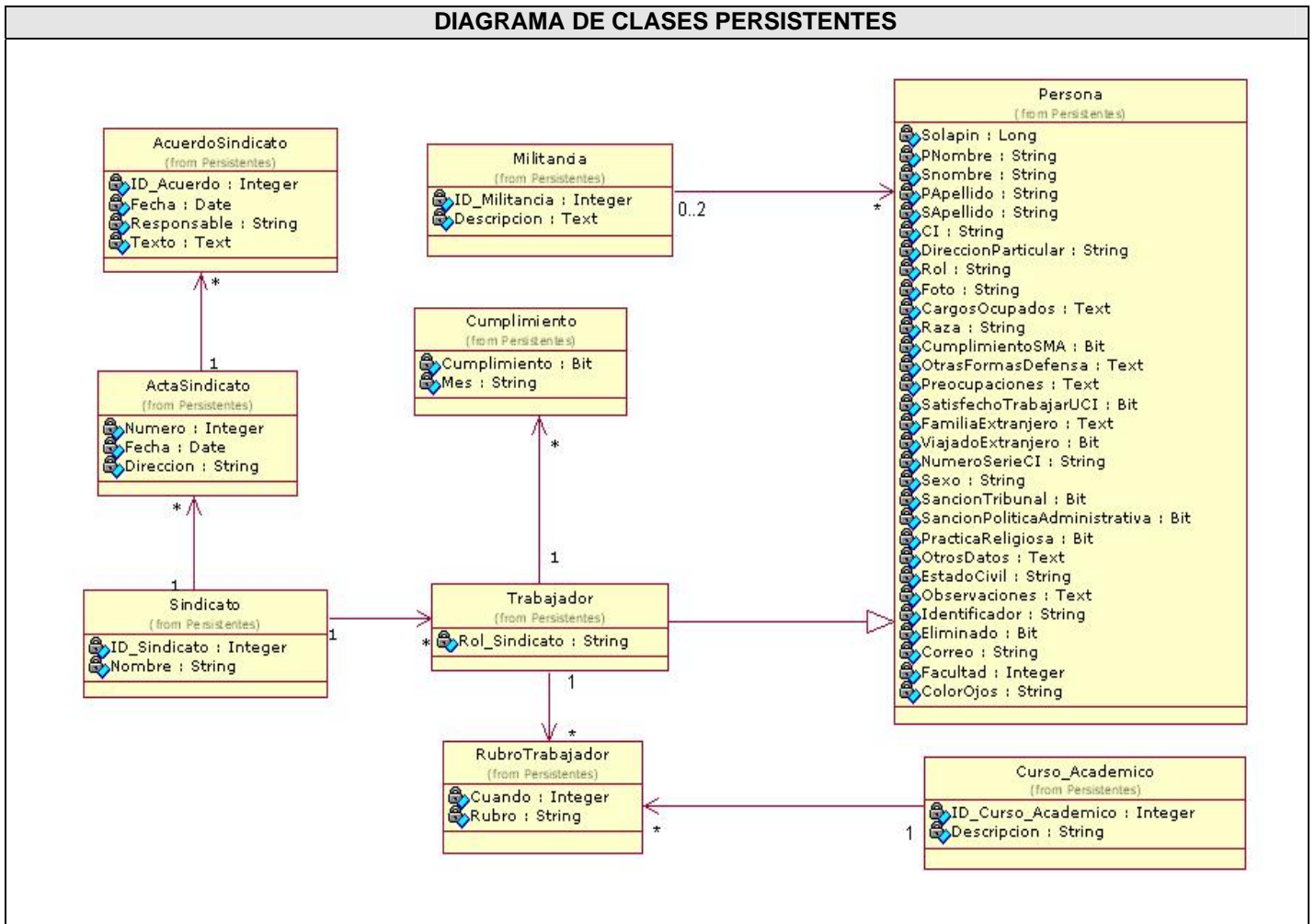


Figura 9.- Diagrama de clases persistentes del módulo Sindicato.

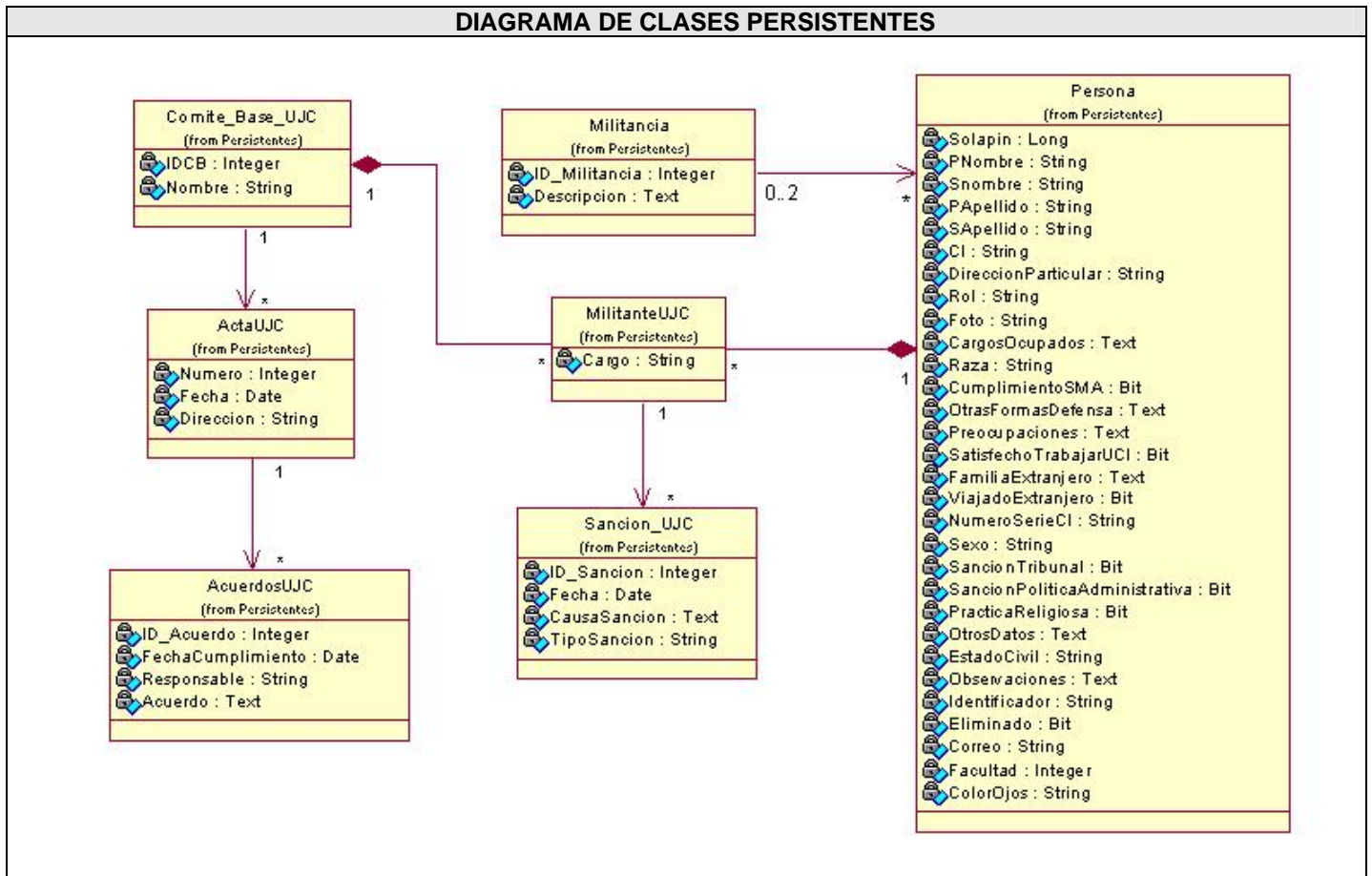


Figura 10.- Diagrama de clases persistentes del módulo UJC.

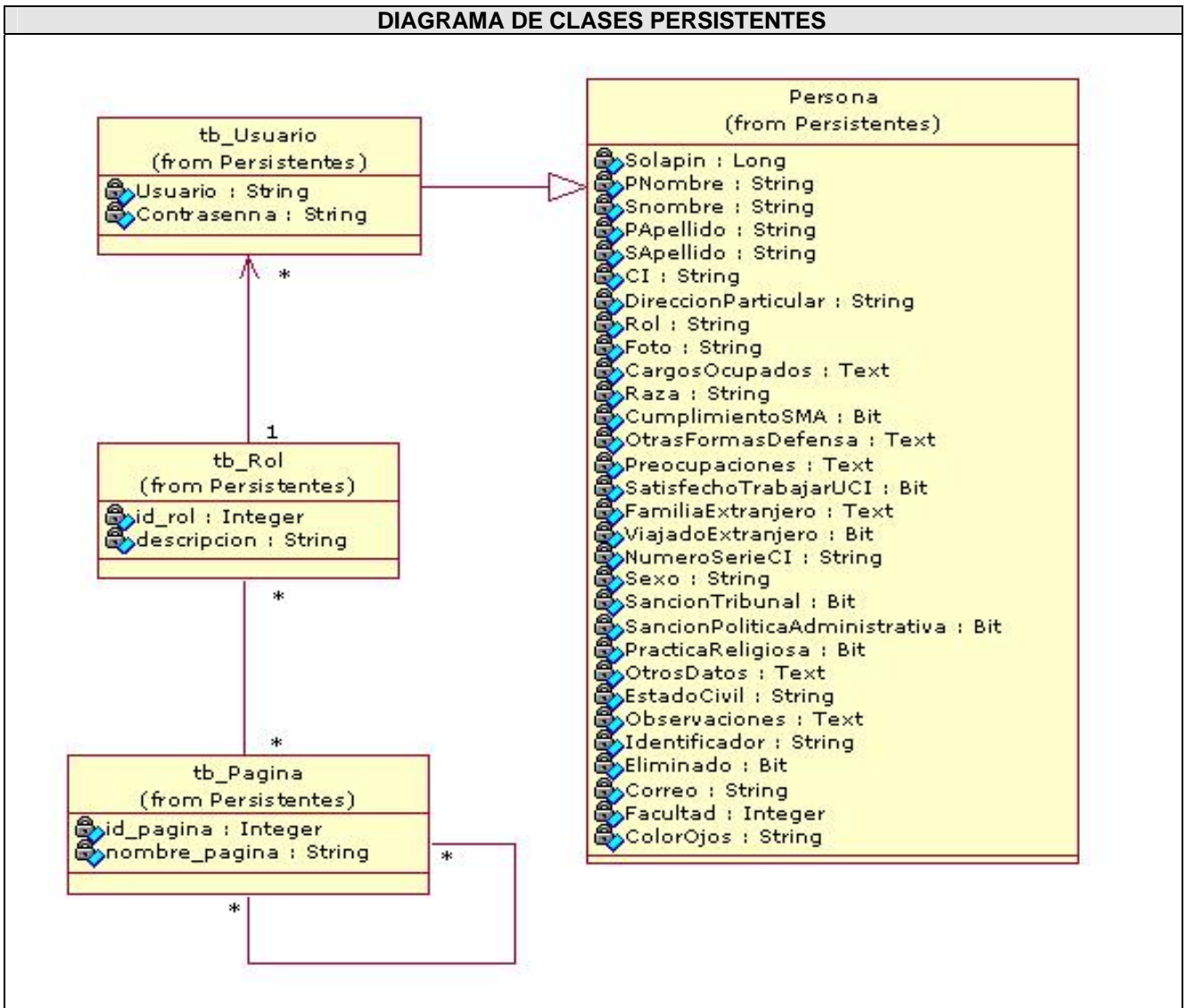


Figura 11.- Diagrama de clases persistentes del módulo Seguridad.

2.6 Descripción de las clases.

Las clases persistentes poseen un conjunto de atributos que se describirán en las tablas que se muestran en el Anexo 1. Cada clase tiene como métodos:



- ✓ Un constructor: se llama igual que la clase y contiene cada uno de los atributos de la misma, excepto el Id en alguna de ellas que será auto-numérico en la BD (las que cumplan la condición se identificarán con un *), ejemplo: para la clase Actividad el constructor sería Actividad (int aSemana, int aTipoActividad).
- ✓ Métodos get, que retornan el valor del atributo: se pondrán con minúscula, seguidos del nombre del atributo que tiene asociado, ejemplo: getSemana().
- ✓ Métodos set, que modifican el valor del atributo: se pondrán con minúscula, seguidos del nombre del atributo que tiene asociado, ejemplo: setSemana (int aSemana).

Los métodos no se pondrán en las tablas por cuestión de legibilidad. A continuación se muestran las principales clases, las demás se pueden encontrar en el Anexo 1.

Tabla 2 Clase Actividad (*).

Nombre: Actividad	
Atributo	Tipo
Id_Actividad	integer
Semana	integer
TipoActividad	string

Tabla 3.- Clase Alumno_Ayudante.

Nombre: Alumno_Ayudante	
Atributo	Tipo
Evaluacion	string

Tabla 4.- Clase Brigada.

Nombre: Brigada	
Atributo	Tipo
ID_Brigada	integer
Nombre	string
Seccion	bit
Anno	string

Tabla 5.- Clase Curso_Academico (*).

Nombre: Curso_Academico	
Atributo	Tipo
ID_Curso_Academico	integer
Descripcion	string



Tabla 6.- Clase Estudiante.

Nombre: Estudiante	
Atributo	Tipo
NoLista	integer

Tabla 7.- Clase Persona.

Nombre: Persona	
Atributo	Tipo
Solapin	long
PNombre	string
SNombre	string
PApellido	string
SApellido	string
CI	string
DireccionParticular	string
Foto	string
CargosOcupados	text
Raza	string
CumplimientoSMA	bit
OtrasFormasDefensa	string
Preocupaciones	string
SatisfechoTrabajarUCI	bit
FamiliaExtranjero	text
ViajadoExtranjero	bit
NumeroSerieCI	string
Sexo	string
SancionTribunal	bit
SancionPoliticaAdministrativa	bit
PracticaReligiosa	bit
OtrosDatos	text
Observaciones	text
Identificador	string
Eliminado	bit
Correo	string
Facultad	integer
ColorOjos	string
Rol	string
EstadoCivil	string

Tabla 8.- Clase Profesor.

Nombre: Profesor	
Atributo	Tipo



Categoría_Docente	string
Categoría_Cientifica	string
Maestria	string
Graduado_De	string

Tabla 9.- Clase Trabajador.

Nombre: Trabajador	
Atributo	Tipo
Rol_Sindicato	string



2.7 Diseño de la BD. Diagramas Entidad-Relación.

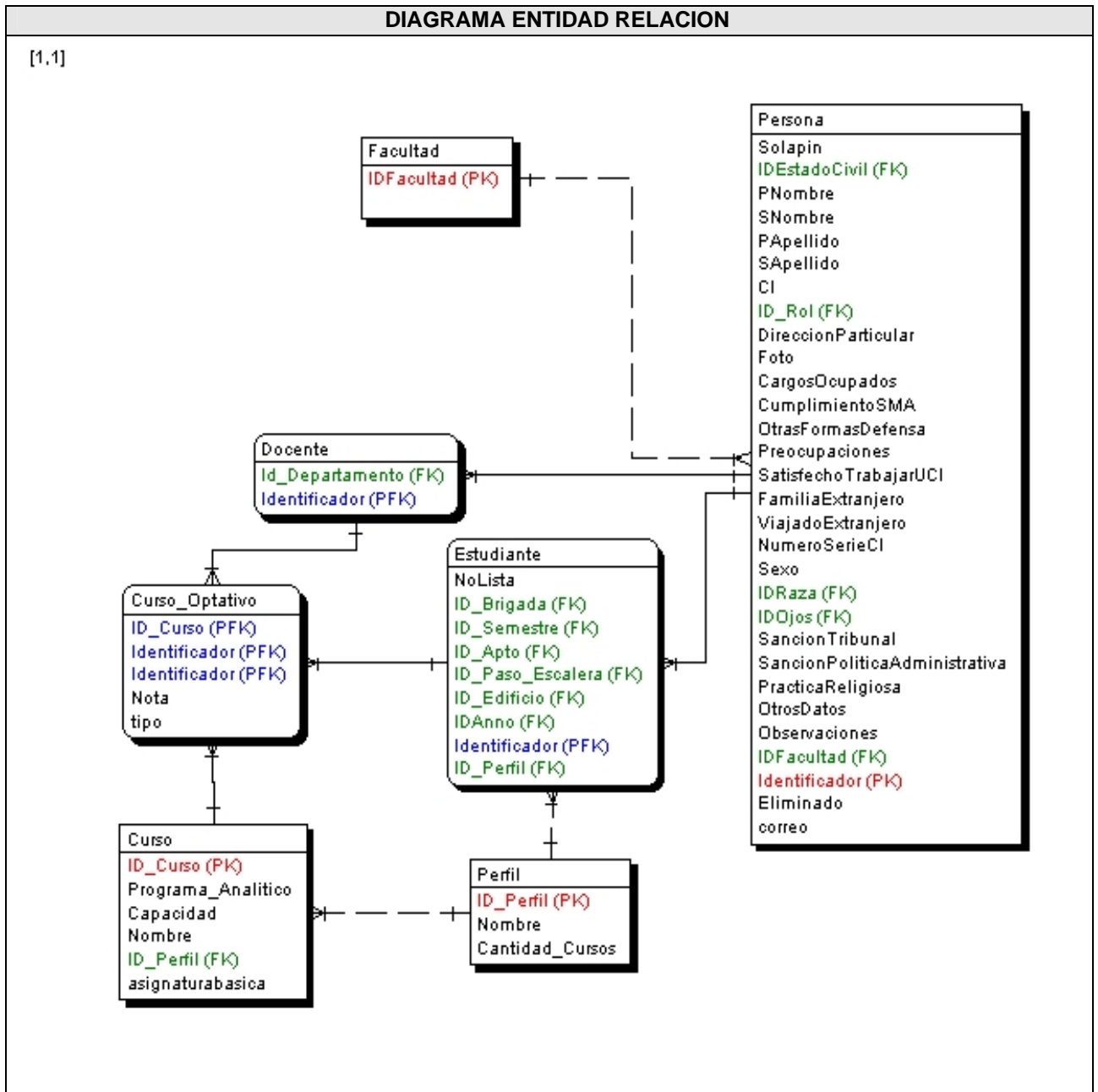


Figura 12.- Diagrama Entidad-Relación del módulo Cursos Optativos.



CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

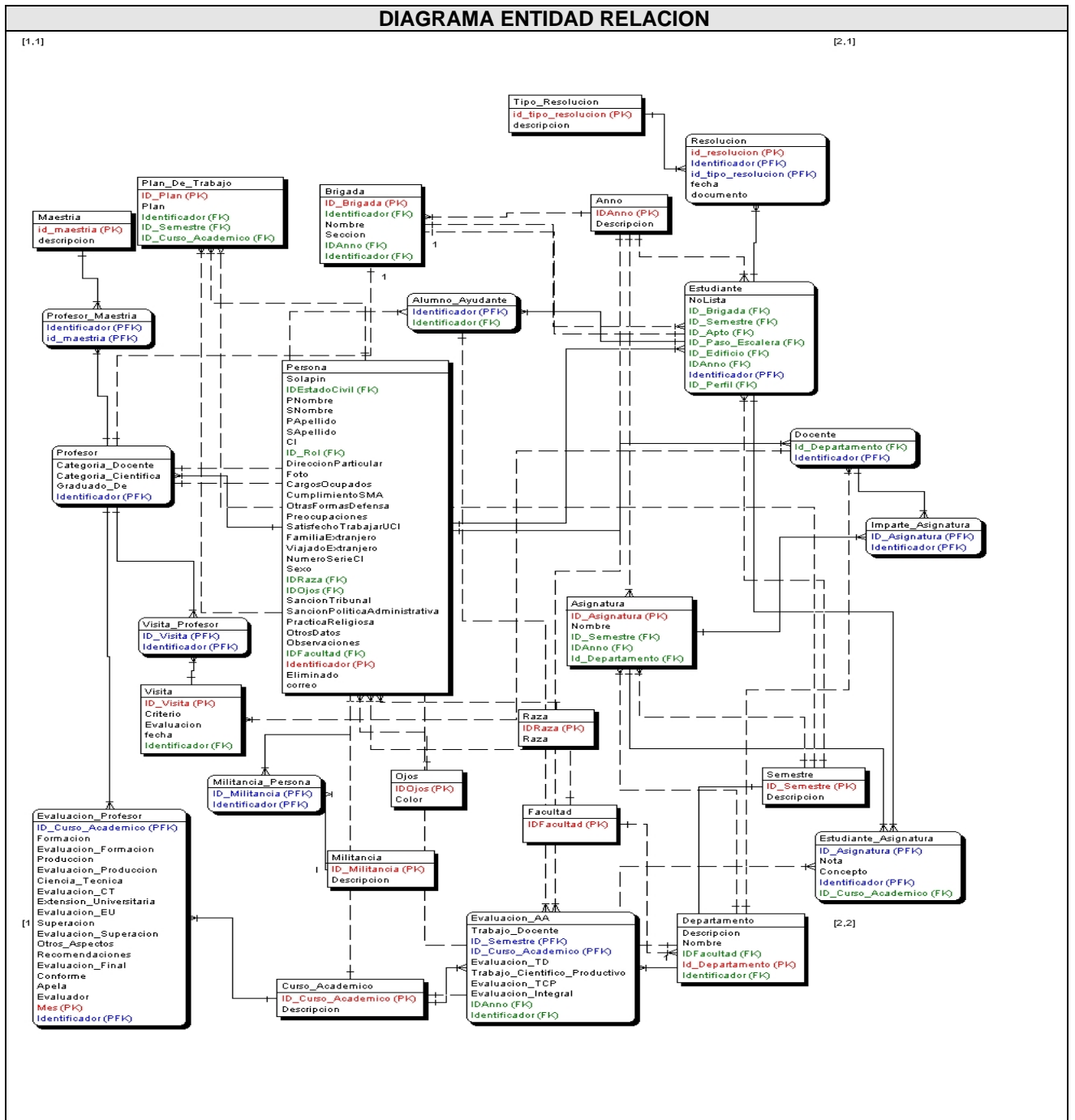


Figura 13.- Diagrama Entidad-Relación del módulo Docencia.



CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

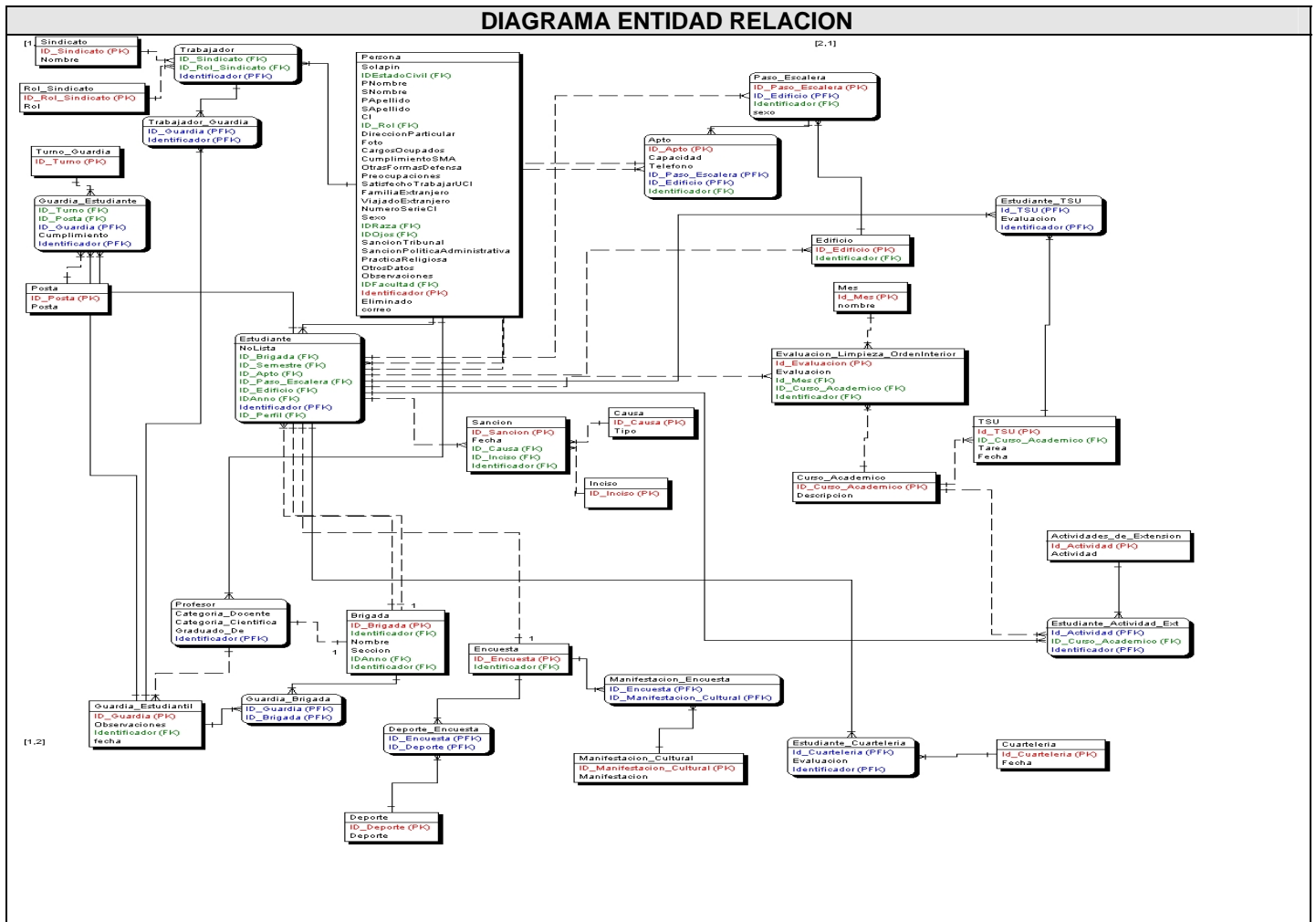


Figura 14.- Diagrama Entidad-Relación del módulo Extensión Universitaria.

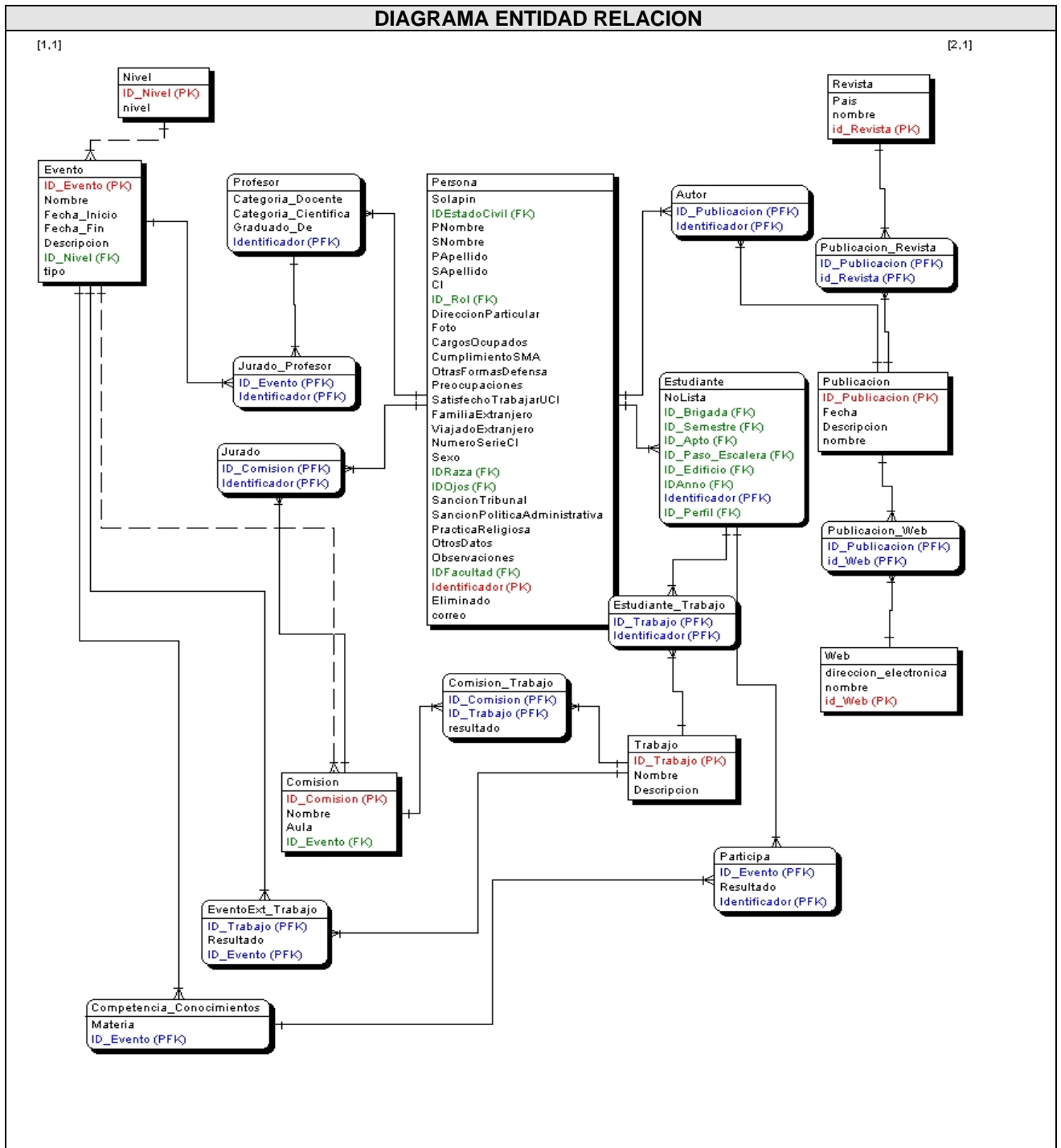


Figura 15.- Diagrama Entidad-Relación del módulo Investigación.



CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

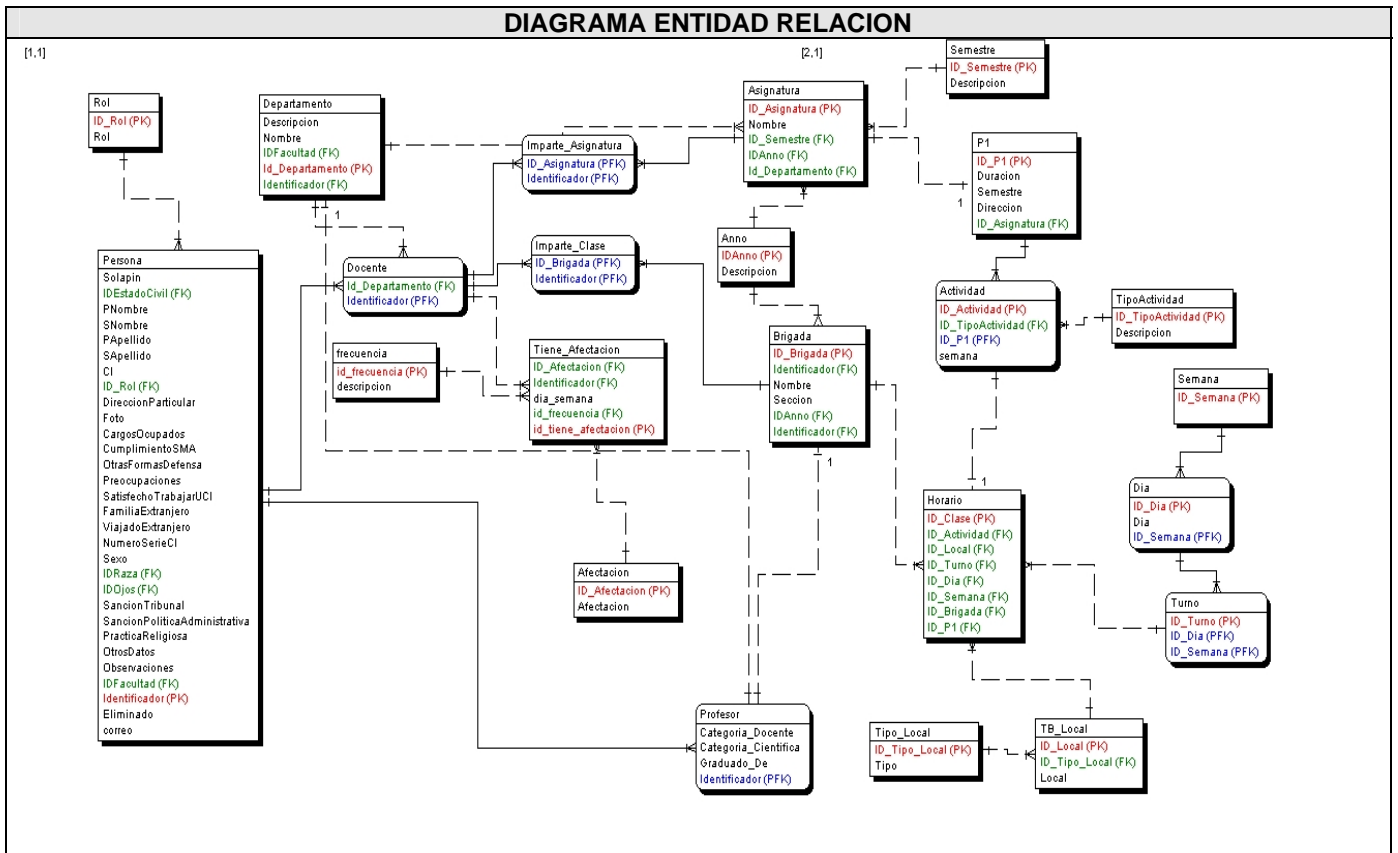


Figura 16.- Diagrama Entidad-Relación del módulo Planificación Docente.



CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

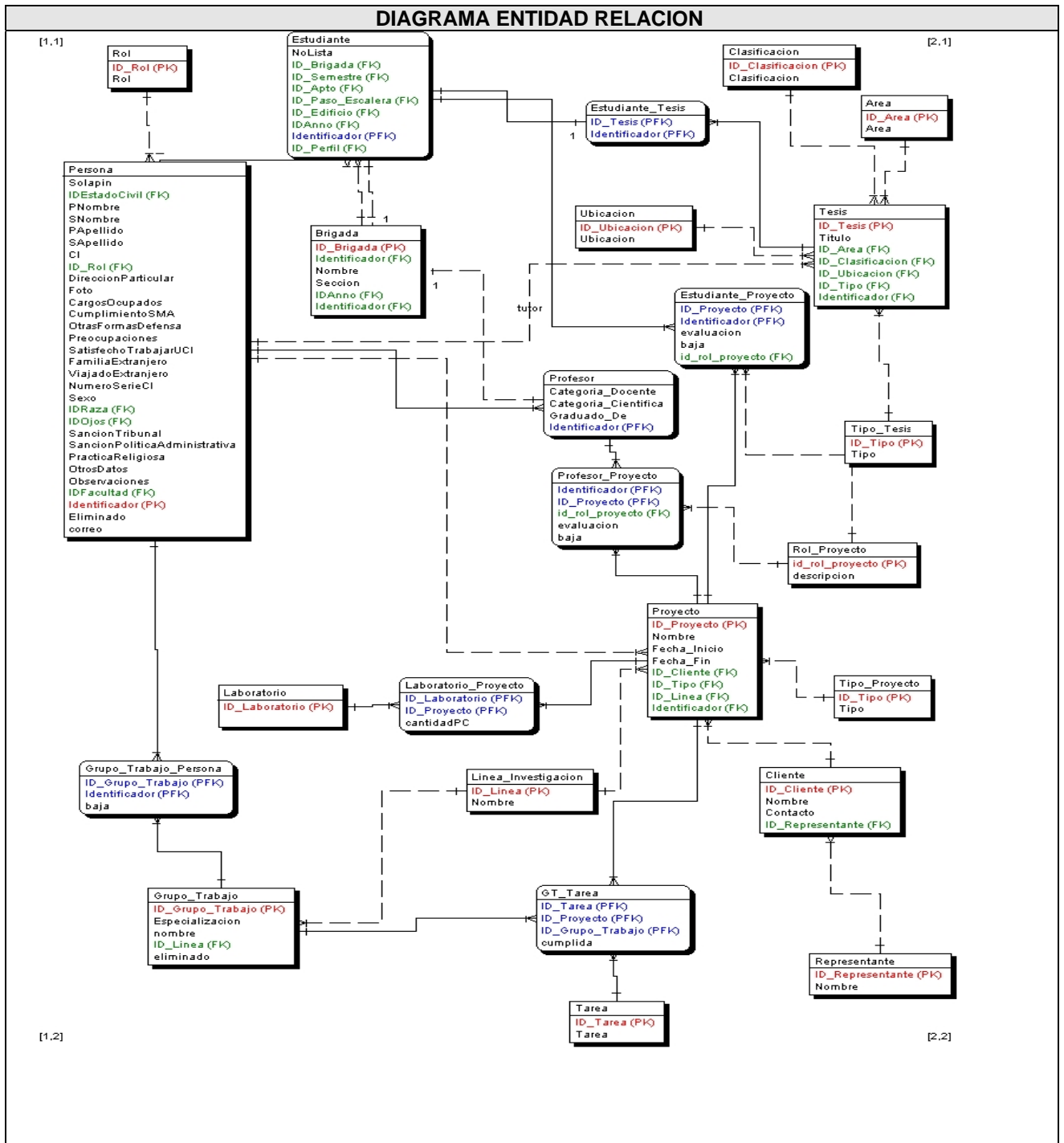


Figura 17.- Diagrama Entidad-Relación del módulo de Proyectos Productivos.

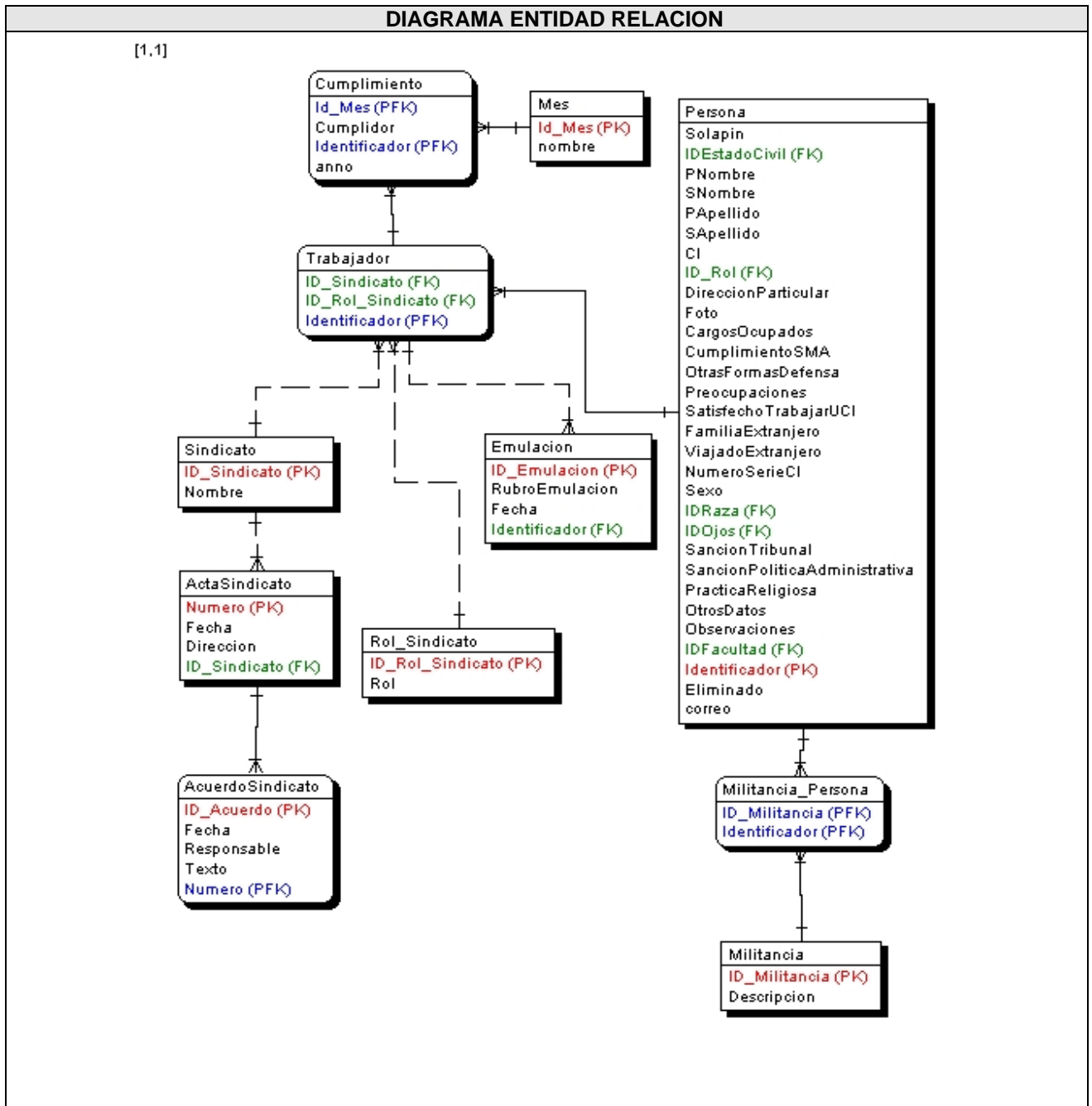


Figura 18.- Diagrama Entidad-Relación del módulo Sindicato.

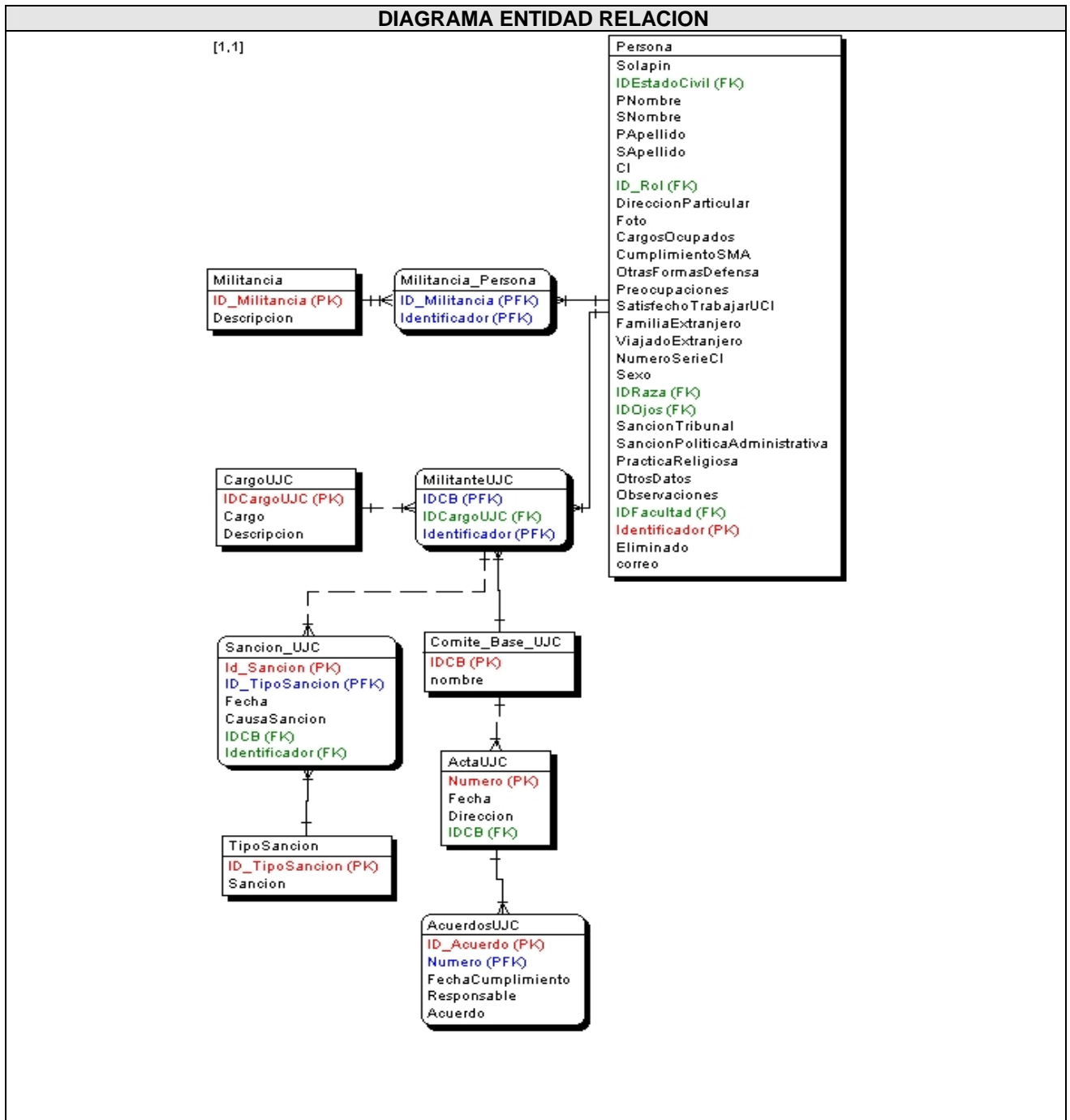


Figura 19.- Diagrama Entidad-Relación del módulo UJC.

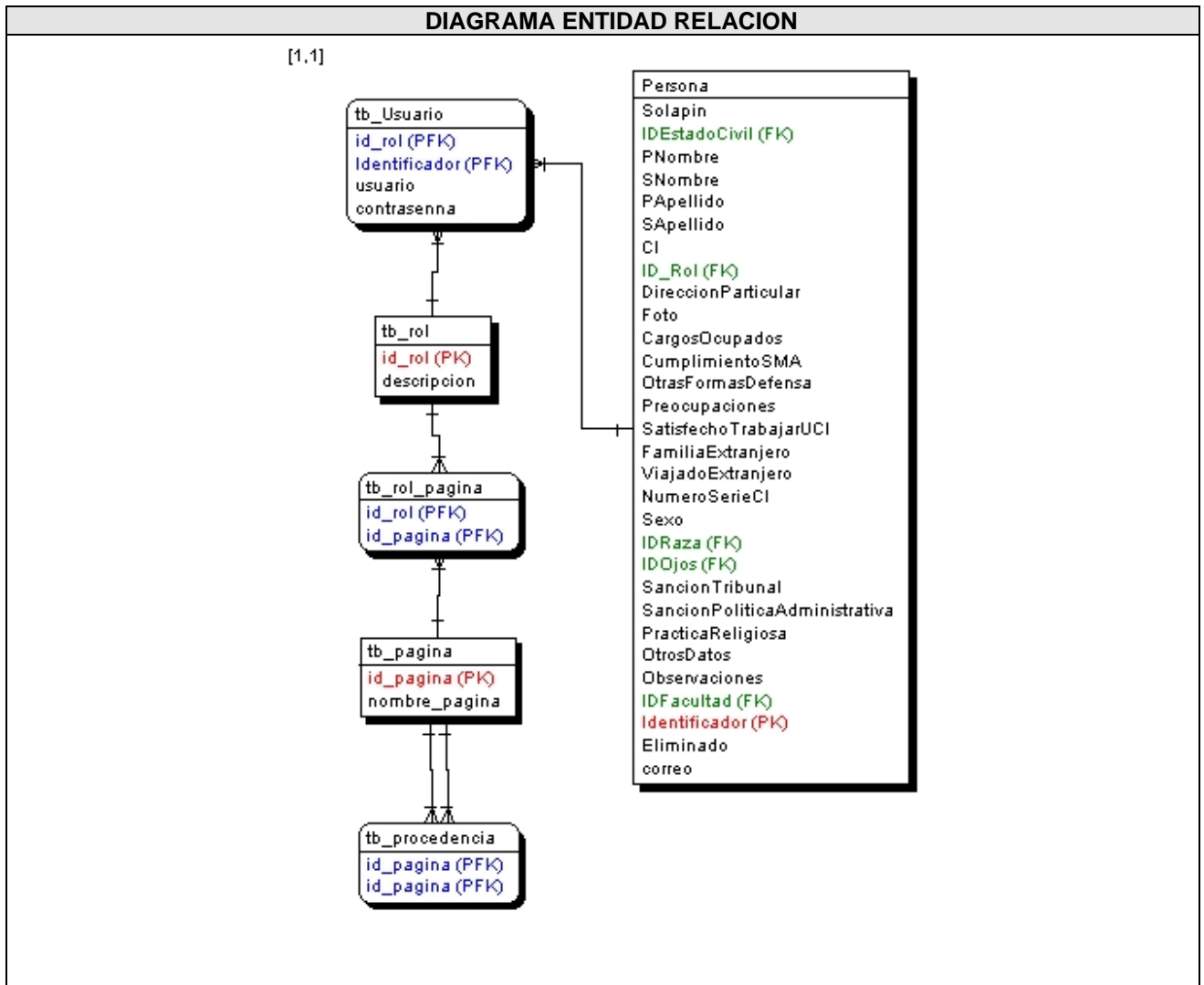


Figura 20.- Diagrama Entidad-Relación del módulo de Seguridad.

2.8 Descripción de las tablas.

Se partió de definir las clases persistentes. Las clases simples se representaron como tablas y en las relaciones de 1..n se añadió una llave extranjera al extremo n, que se corresponde con la llave de la clase del extremo 1.



En el caso de las relaciones n...m se crea una nueva tabla que tiene como llave las clases que conforman la relación.

El gestor de base de datos a utilizar es el PostgreSQL y por tanto los tipos de los campos son los que se corresponden con el lenguaje plpgsql.

A continuación se muestran las principales tablas que se utilizan en la Base de Datos. Las demás tablas se encuentran en el Anexo 2.

Tabla 10.- Alumno_Ayudante.

Nombre: Alumno_Ayudante.		
Descripción: Almacena los datos del alumno ayudante.		
Atributo	Tipo	Descripción
IdentificadorTutor	string	El id de profesor.
Identificador	string	El id de estudiante.

Tabla 11.- Asignatura.

Nombre: Asignatura.		
Descripción: Almacena los datos de la asignatura de un estudiante.		
Atributo	Tipo	Descripción
ID_Asignatura	integer	El id de Asignatura.
Nombre	string	El nombre de la asignatura.
ID_Semestre	integer	El id de semestre.
IDAnno	integer	El id de Anno.
Id_Departamento	integer	El id de departamento.

Tabla 12.- Brigada.

Nombre: Brigada.		
Descripción: Almacena los datos de la brigada.		
Atributo	Tipo	Descripción
ID_Brigada	string	El id de Brigada.
Identificador	string	El id del estudiante jefe de grupo.
Nombre	string	Nombre de la brigada.
Seccion	bit	Sesión de clases de esa brigada.
IDAnno	integer	El id de Anno.
Identificador	string	El id del profesor guía.

Tabla 13.- Docente.

Nombre: Docente.		
Descripción: Almacena los datos de un docente.		



CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

Atributo	Tipo	Descripción
ID_Departamento	integer	El id de Departamento.
Identificador	string	El id de la Persona.

Tabla 14.- Estudiante.

Nombre: Estudiante.		
Descripción: Almacena los datos de los estudiantes.		
Atributo	Tipo	Descripción
NoLista	integer	El numero de la lista.
Identificador	string	El id de la Persona que es estudiante.
ID_Brigada	string	El id de Brigada.
ID_Semestre	integer	El id de Semestre.
ID_Apto	integer	El id de Apto.
ID_Paso_Escalera	integer	El id de Paso_Escalera
ID_Edificio	integer	El id de Edificio.
IDAnno	integer	El id de Anno.
ID_Perfil	integer	El id de Perfil.

Tabla 15.- Horario.

Nombre: Horario.		
Descripción: Almacena los datos de los horarios de los estudiantes.		
Atributo	Tipo	Descripción
ID_Clase	integer	El id de Horario.
ID_Actividad	integer	El id de Actividad.
ID_Local	integer	El id de Local.
ID_Turno	integer	El id de Turno.
ID_Dia	integer	El id de Dia.
ID_Semana	integer	El id de Semana.
ID_Brigada	string	El id de Brigada.
ID_P1	integer	El id de P1.

Tabla 16.- Persona.

Nombre: Persona.		
Descripción: Almacena los datos de las personas.		
Atributo	Tipo	Descripción
Solapin	bigint	El numero de solapin.
IDEstadoCivil	integer	El id de EstadoCivil.
PNombre	string	Primer nombre.
SNombre	string	Segundo nombre.
PApellido	string	Primer apellido.
SApellido	string	Segundo apellido.
CI	string	El numero del carnet de identidad.



CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

ID_Rol	integer	El id de Rol.
DireccionParticular	string	La dirección particular.
Foto	string	La dirección donde se encuentra la foto.
CargosOcupados	text	Los cargos ocupados.
CumplimientoSMA	bit	Si cumplió o no el servicio militar.
OtrasFormasDefensa	text	Otras formas de defensa.
Preocupaciones	text	Las preocupaciones.
SatisfechoTrabajarUCI	bit	Si esta satisfecho de trabajar en la UCI.
FamiliaExtranjero	text	Si tiene familia en el extranjero.
ViajadoExtranjero	bit	Si ha viajado al extranjero.
NumeroSerieCI	string	El numero de serie del carnet de identidad.
IDRaza	integer	El id de Raza.
IDOjos	integer	El id de Ojos.
SancionTribunal	bit	Si tiene sanciones.
SancionPoliticaAdministrativa	bit	Si tiene sanciones políticas administrativas.
PracticaReligiosa	bit	Si realiza prácticas religiosas.
OtrosDatos	text	Otros datos de interés.
Observaciones	text	Observaciones pertinentes.
IDFacultad	integer	El id de Facultad.
Identificador.	string	El id de Persona.
Eliminado	string	Si no se encuentra activo.
Correo	string	El correo de cada persona.

Tabla 17.- Profesor.

Nombre: Profesor.		
Descripción: Almacena los datos de los profesores.		
Atributo	Tipo	Descripción
Categoria_Docente	string	La categoría docente del profesor.
Categoria_Cientifica	string	La categoría científica del profesor.
Graduado_De	string	Graduado de.
Identificador	string	El id de Persona.

Tabla 18.- Proyecto.

Nombre: Proyecto.		
Descripción: Almacena los datos de los proyectos.		
Atributo	Tipo	Descripción
ID_Proyecto	integer	El id de Proyecto.
Nombre	string	El nombre del proyecto.
Fecha_Inicio	date	La fecha de inicio del proyecto.
Fecha_Fin	date	La fecha de fin del proyecto.
ID_Cliente	integer	El id de Cliente.
ID_Tipo	integer	El id de Tipo_Proyecto.



ID_Linea	integer	El id de Linea_Investigacion.
Identificador	string	El id de Persona, jefe del proyecto.

Conclusiones.

Se finaliza la etapa de descripción y análisis, se confeccionaron los diagramas que fueron necesarios utilizando el Rational Rose y el Case Studio 2, para obtener una mejor visión de la solución del problema presentado; además se presentó el diseño de la base de datos en la cual se van a almacenar los datos persistentes resultados de la gestión de la información del sistema.



Introducción.

En este capítulo se verá lo referente a la validación teórica del diseño, la integridad, así como la normalización y el análisis de la redundancia. También se tratará acerca de la seguridad y el control de acceso, la realización y recuperación de backups de la base de datos, la trazabilidad de acciones y la validación funcional.

3.1 Validación teórica del diseño.

3.1.1 Integridad.

En un momento dado, los valores de los datos en una base de datos son una representación de un fragmento de la realidad. Es decir, si se tiene una tabla con los atributos de personas y entre ellos el peso o la edad, estos no pueden ser negativos, porque en el mundo real, esto no es posible. Si se añade una restricción de este tipo a una base de datos, se está incluyéndole una *regla de integridad*; por ejemplo, si hay una base de datos alumnos, profesores y asignaturas para una escuela o facultad, algunas reglas de integridad serían:

- ✓ Un profesor no puede dar más de dos materias.
- ✓ Un profesor no puede dar la misma materia más de dos semestres seguidos.
- ✓ Un estudiante que no aprueba una materia en la segunda oportunidad será dado de baja.
- ✓ Las calificaciones no pueden tomar valores fuera del rango 2-5.

Algunas de estas reglas son arbitrarias y para fines de ejemplificar el concepto y es inmediato notar que se aplican a tablas en específico.

Sin embargo, las bases de datos relacionales, tienen dos reglas *generales* de integridad que se aplican a las llaves primarias y a las llaves foráneas y estas son:

- ✓ Integridad de las entidades. Ningún componente de la llave primaria de una relación puede aceptar valores nulos.



- ✓ Integridad referencial. La base de datos no debe contener valores de llaves ajenas sin concordancia.

La otra regla es acerca de la integridad transaccional, proporcionada por las características Durabilidad, Aislamiento, Consistencia e Indivisibilidad (ACID). En bases de datos se denomina ACID a la propiedad de una base de datos para realizar transacciones seguras.

- ✓ Indivisibilidad: es la propiedad que asegura que la operación se ha realizado o no, y por lo tanto ante un fallo del sistema no puede quedar a medias.
- ✓ Consistencia: es la propiedad que asegura que sólo se empieza aquello que se puede acabar. Por lo tanto se ejecutan aquellas operaciones que no van a romper la reglas y directrices de integridad de la base de datos.
- ✓ Aislamiento: es la propiedad que asegura que una operación no puede afectar a otras. Esto asegura que dos transacciones sobre la misma información nunca generarán ningún tipo de error.
- ✓ Durabilidad: es la propiedad que asegura que una vez realizada la operación, ésta persistirá y no se podrá deshacer aunque falle el sistema.

En PostgreSQL esto es posible gracias a los conjuntos de acciones.

BEGIN WORK;	BEGIN WORK;
.....
COMMIT WORK;	ROLLBACK WORK;

También están los “obligadores” (constraints), las reglas y los “disparadores” (triggers), los que tienen el objetivo de mantener la Base de Datos íntegra y consistente previniendo contra entrada de datos inválidas.

Constraints:

El check que restringe la entrada de datos chequeando determinada condición.

```
CREATE TABLE Persona (  
PNombre    varchar(50)  CHECK(lenght(PNombre)>=2),
```



Edad integer CHECK(Edad>0),
Sexo varchar(1) CHECK(Sexo IN ('M','F')))

Reglas:

Permiten modificar los datos cuando se accede a una tabla. Modifican los efectos de SELECT, INSERT, UPDATE, y DELETE.

Triggers:

Permiten chequear o modificar una tupla antes de que esta sea añadida a la base de datos. Llamam funciones especiales que se ejecutan en el servidor cada vez que una tupla es añadida o modificada.[14]

3.1.2 Normalización de la base de datos.

La Normalización de la Base de Datos es una técnica desarrollada inicialmente por E.F. Codd en 1972, para diseñar la estructura lógica de una base de datos en el modelo relacional.

La Normalización es un proceso en el cual se va comprobando el cumplimiento de una serie de reglas, o restricciones, por parte de un esquema de relación; cada regla que se cumple aumenta el grado de normalización del esquema de relación; si una regla no se cumple, el esquema de relación se debe descomponer en varios esquemas de relación que sí la cumplan por separado.

Existen varios niveles de normalización:

- ✓ Primera Forma Normal (1FN).
- ✓ Segunda Forma Normal (2FN).
- ✓ Tercera Forma Normal (3FN).
- ✓ Forma Normal Boyce-Codd.
- ✓ Cuarta Forma Normal.
- ✓ Quinta Forma Normal o Forma Normal de Proyección-Unión.
- ✓ Forma Normal de Proyección-Unión Fuerte.
- ✓ Forma Normal de Proyección-Unión Extra Fuerte y Forma Normal de Clave de Dominio.



Primera Forma Normal.

La primera forma normal se definió para prohibir los atributos multivaluados, los atributos compuestos y sus combinaciones. Esto indica que todos los atributos deben tener valores atómicos.

La Base de Datos del Sistema de Gestión de Información de la Facultad número ocho (SGIF) cumple con la primera forma normal, todos los datos de las relaciones involucradas son atómicos, o sea, no son ni multivaluados ni compuesto ni ninguna combinación de estos, excepto por la tabla profesor que tiene un campo.

Por ejemplo:

La tabla Profesor tiene los campos categoría_científica, categoría_docente, graduado_de y maestría que es multivaluado pues un profesor puede tener varias maestrías.

La manera de solucionar este problema es creando una relación que se llamará maestría que tendrá como atributos un id_maestria y una descripción. Se crea una relación de mucho a mucho entre esta nueva relación y la relación profesor generando otra tabla para la base de datos denominada profesor_maestria.

Segunda Forma Normal.

La segunda forma normal plantea que si todos los atributos que no forman parte de la llave primaria, dependen de manera total de los atributos que forman la llave primaria. Una relación que está en primera forma normal y que tiene como llave primaria un solo atributo, entonces está en segunda. La segunda forma normal solo es aplicable a las relaciones que tienen llaves compuestas por dos o más atributos.

La Base de Datos de SGIF cumple con la segunda forma normal, pues ya está en primera forma normal y no presenta en ninguna de sus tablas, atributos que dependan solo de parte de la llave primaria de esa tabla.

Tercera Forma Normal.

La tercera forma normal plantea que la relación debe estar en segunda forma normal y que cada atributo de la relación que no está contenido dentro de la llave primaria dependa solo de la llave primaria y no de ningún otro atributo.



La Base de Datos de SGIF cumple con la tercera forma normal, pues no presenta en ninguna de sus tablas, atributos que dependan de otros atributos que no formen parte de la llave primaria, además están en segunda forma normal.

Solo se llegará hasta la tercera forma normal, debido a que un esquema relacional muy grande como el que se está desarrollando es muy difícil llevarlo a las otras formas normales y durante el proceso se pueden crear características no deseadas.[9]

3.1.3 Análisis de redundancia de información.

Con un buen diseño de una base de datos se logrará evitar la aparición de información repetida o redundante. Lo ideal es lograr una redundancia nula; no obstante, en algunos casos la complejidad de los cálculos hace necesaria la aparición de redundancias.

Por ejemplo en la Base de Datos de SGIF existe redundancia que es necesaria, en cuanto a la militancia de las personas en cada una de las organizaciones. Existe la tabla militancia, donde se guardará cada una de las organizaciones como Unión de Jóvenes Comunistas (UJC), Partido Comunista de Cuba (PCC), etcétera. También existe la tabla militanteujc, donde se registrarán todos los militantes de la UJC. Como se podrán dar cuenta, cada militante de la UJC estará registrado en las tablas militanteujc y militancia_persona que es la tabla que se crea por la relación de mucho a mucho que existe entre persona y militancia, pero se considera que es necesaria esta redundancia, pues se necesitan datos del militante si es de la UJC que no es posible relacionarlos con la tabla militancia debido a que para los otros tipos de militancia no se necesitan estos datos.

Existe también redundancia en el módulo docente en cuanto a los planes de trabajo de los alumnos ayudante y los profesores, los datos que se recogen son los mismos para ambos, por lo que se puede eliminar esta redundancia creando una tabla denominada docente donde se registrarán los alumnos ayudantes y los profesores, esta relación sería una especialización de persona que se relacionaría con los planes de trabajo.

Existe también redundancia en el módulo de planificación en cuanto a las tablas alumnos ayudante y profesores que imparten asignaturas, le dan clase a las brigadas y tienen afectaciones, en todos los casos tiene relaciones de mucho a mucho con las tablas asignatura, brigada y tiene afectación respectivamente,



donde se crean nuevas tablas dobles, solo se diferenciarían en el nombre. Esto se resuelve con la tabla docente que se creó y relacionándola con las tablas asignatura, brigada y tiene afectación, ahora se crearían solo tres nuevas tablas y la información no estaría repetida eliminándose así la redundancia.

3.1.4 Análisis de la seguridad de la base de datos.

La información que se almacena en la base de datos debe protegerse contra el acceso no autorizado, la destrucción o alteración con fines indebidos y la introducción accidental de inconsistencia.

Es más fácil evitar la pérdida accidental de consistencia de la información que permitir el acceso malintencionado a la base de datos. No es posible proteger de manera absoluta a la base de datos contra abusos ilícitos, pero puede hacerse que el coste sea tal alto que frustré casi todos los intentos de lograr acceso sin autorización a la base de datos.

3.1.4.1 Seguridad y control de acceso.

En PostgreSQL existen tres niveles de acceso:

El nivel 0 es el que se encarga sobre las máquinas (host) y los usuarios. Es decir es el permite configurar que máquina/s y/o usuario/s se pueden conectar a la Base de Datos. Utilizando las opciones del fichero de configuración `pg_hba.conf`.

El nivel 1 es el que se encarga de los usuarios y las Bases de Datos. Es el que permite configurar a que Bases de Datos se pueden conectar. Utilizando las opciones del fichero de configuración `pg_ident.conf`.

El nivel 2 es el que se encarga de las tablas. Es el que permite configurar a que tablas pueden acceder. Utilizando los comandos `GRANT` para dar permisos y `REVOKE` para eliminar los permisos.[14]

3.1.4.2 Backups y recuperación de los datos.

Existen dos formas de realizar el aseguramiento de los datos.

- 1.-Compactado del directorio de datos.



- ✓ Detener el servicio de Postgres.
- ✓ Realizar la copia en *.rar o *.zip del directorio data de la carpeta de instalación de Postgres.
- ✓ Iniciar el servicio de Postgres.
- ✓ Copiar el *.rar o *.zip en algún soporte. (CD, disco.....).

Para recuperar la copia: descomprimir en el directorio data de la carpeta de instalación de Postgres.

Ventajas:

- ✓ Es rápido y sencillo.

Desventajas:

- ✓ Recomendable parar el servicio Postgres.
- ✓ Incompatibilidad entre versiones.

2.- Volcado (dump).

Para el volcado se utiliza el comando pg_dump.

```
pg_dumpall --d >todas.dump
```

Para recuperar la recuperación se utiliza ps_restore.

```
pg_restore -e template1 <todas.dump
```

Ventajas:

- ✓ Se puede hacer con el servicio corriendo.
- ✓ Se puede hacer parcial.
- ✓ Compatibilidad entre versiones.

Desventajas:

- ✓ Gran lentitud con Base de Datos grandes.
- ✓ Consumo de recursos del sistema.



3.1.5 Trazabilidad de las acciones.

La trazabilidad es un conjunto de medidas, acciones y procedimientos que permiten registrar e identificar cada consulta desde su origen hasta su destino final. Es la capacidad para reconstruir la historia, recorrido o aplicación de un a determinada consulta.

El servidor de base de datos almacena en la carpeta de los registros ficheros con la información del día, hora, usuario, máquina de origen, consulta, datos y tiempo que invirtió en la consulta, lo que permite llevar un control estricto de que se está haciendo en la Base de Datos.

3.2 Validación funcional.

3.2.1 Herramientas para un llenado voluminoso e inteligente de la base de datos.

En el mundo existen algunas herramientas para el llenado voluminoso de bases de datos. Algunas personas utilizan scripts encontrados en Internet para llenar sus bases de datos y realizarles pruebas, aunque no son recomendables debido a que se debe tener conocimientos de programación para esto.

EMS Data Generator para PostgreSQL es una de estas herramientas para el llenado voluminoso de la base de datos para la realización de pruebas. Se usa para llenar varias tablas o todas las tablas de una base de datos al mismo tiempo, respetando la integridad referencial. La aplicación permite definir rangos de valores, máscaras, que permiten llenar la base de datos de forma sencilla y rápida. La generación de los datos es de forma aleatoria e incremental. Permite la utilización de resultados de consultas para generar listas de valores.

Conclusiones.

Se finaliza con la etapa de validación del diseño, se realizó un análisis sobre la validación teórica y funcional, en la se aborda sobre la integridad, seguridad, y sobre el control de las acciones de los usuarios hacia la Base de Datos.



El presente trabajo una vez terminado y certificado se pondrá en explotación en la facultad ocho como parte del sistema de gestión y control de la información docente, que actualmente se desarrolla de forma manual haciendo uso de documentos en Microsoft Word o empleando tablas Excel. En el mismo se reflejó el estudio realizado, el cual permitió obtener las bases para la generación de una propuesta de solución que permitiera realizar los procesos de gestión de una forma rápida y eficiente.

Se considera que la idea a defender planteada al inicio ha sido debidamente demostrada aportando la centralización de toda la información referente a los procesos que se desarrollan en la facultad así como la rapidez en la comunicación y en las búsquedas de información por parte del personal disminuyendo así su carga laboral. Las metas propuestas fueron alcanzadas y se logró el objetivo fundamental de la investigación, el análisis, diseño e implementación de la Base de Datos.



1. Que se realice la estandarización de la Base de Datos, según las normas establecidas en la Universidad para este tipo de sistema.
2. Que se ponga en práctica la utilización de este diseño de Base de Datos.
3. Que se implementen los servicios web que van a brindar los sistemas de la escuela, para el llenado de la Base de Datos.



REFERENCIAS BIBLIOGRÁFICAS

1. Anónimo. *Base de Datos*. 2006 mayo, 2007 [cited; Available from: http://es.wikipedia.org/wiki/Base_de_datos].
2. Anónimo. *Administración y Gestión Universitaria de información a la Labor Académica*. 2005 [cited marzo del 2007]; Available from: <http://alpes.stg.uchile.cl/~srvinfo/infogest/cc61afinal/index.htm#inicio>.
3. Anónimo. *Sistema Integral de Información Universitaria*. 2005 [cited; Available from: <http://www.uv.mx/universidad/doctosofi/4informe2004/10/contenido.htm#siuu>].
4. Anónimo. *Sistema Integral de Gestión Académica*. 2005 [cited; Available from: <http://www.gestion.uco.es/gestion/aplicaciones/siga/>].
5. Ibáñez, J.D. and A. Marzal. *Proyecto Lleu*. Noviembre, 2000 [cited; Available from: http://es.tldp.org/Presentaciones/200002hispalinux/conf-20/20-tml/libro_electronico_universid.html].
6. Anónimo. *¿Qué es SinedUC?* 2005 [cited; Available from: <http://www.sineduc.cl/>].
7. Anónimo. *Sistemas de gestión de Base de Datos*. 2007 [cited; Available from: <http://es.wikipedia.org/wiki/SGBD>].
8. Anónimo. *MySQL 5.0 Reference Manual*. 2007 [cited; Available from: <http://dev.mysql.com/doc/refman/5.0/es/index.html>].
9. Mota, S.A. (abril-junio, 2005) *Entonación de Bases de Datos*.
10. Anónimo. *CASE Studio 2*. 2006 [cited; Available from: http://www.programacionphp.net/recursos-programas/programas-de-bases-de-datos/CASE-Studio-2-2.18_11.html].
11. Anónimo. *EMS SQL Manager for PostgreSQL*. 2006 [cited mayo de 2007]; Available from: <http://www.chilewarez.org/index.php?showtopic=73002>.
12. Anónimo. *EMS SQL Query for PostgreSQL*. 2007 [cited mayo de 2007]; Available from: <http://www.yonimh.net/Foro/index.php?action=profile;u=50;sa=showPosts>.



13. Anónimo. Usando PgAdmin. 2007 [cited junio de 2007]; Available from: <http://www.todo-linux.com/modules.php?name=News&file=article&sid=3421>.
14. Quintela, J.A.R., *Puesta en marcha de un servidor de bases de datos utilizando PostgreSQL*. mayo, 2001.



BIBLIOGRAFÍA

1. ANÓNIMO. *ER/Studio User Guide*, 2006.
2. ANÓNIMO. *Oracle Universal Content Management*, 2005.
3. ANÓNIMO. *PostgreSQL 8.2 Documentation*, 2006. [Disponible en: <http://www.postgresql.org/docs/8.2/interactive/index.html>].
4. BOGGS, W. and M. BOGGS. *UML with Rational Rose 2002*, 2002.
5. HANSEN, G. W. and J. H. HANSEN. *Diseño y Administración de Bases de Datos*, 2007.
6. JACOBSON, I.; G. BOOCH, et al. *El proceso unificado de desarrollo de software*, 2000.
7. PERDUE, T. *MySQL and PostgreSQL Compared*, 2000. [Disponible en: <http://www.phpbuilder.com/columns/tim20000705.php3?page=1>].
8. RIORDAN, R. M. *Aprenda programación en SQL Server 2000 Ya*, 2001.



Atributo: Es cada una de las cualidades, propiedades o características de un elemento.

Automatización: Ejecución automática de tareas industriales, administrativas o científicas haciendo más ágil y efectivo el trabajo y ayudando al ser humano.

Backups (Copias de seguridad): Copia de datos de tal forma que estas copias adicionales puedan restaurar un sistema después de una pérdida de información.

Base de datos: Una base de datos es un conjunto de datos que pertenecen al mismo contexto, almacenados sistemáticamente para su uso posterior. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta.

Centralización: Hacer que varias cosas dependan de un poder central. Reunir varias cosas en un centro común.

Escalabilidad: La capacidad del sistema informático de cambiar su tamaño o configuración para adaptarse a las circunstancias cambiantes.

Interfaz: Una interfaz es la parte de un programa informático que permite a éste comunicarse con el usuario o con otras aplicaciones permitiendo el flujo de información.

Normalización: Proceso de reducción sobre una estructura de datos que procura aumentar la integridad, disminuir la redundancia y las dependencias funcionales de esa estructura.

Prototipo: Un prototipo también se puede referir a un objeto diseñado para una demostración de cualquier tipo.

Redundancia: Repetición de una información ya dada en el mensaje.

Rol: Papel desempeñado por las personas en la sociedad.



Rollback: Es el proceso automático que se ejecuta durante una transacción si un error ocurre la base de datos regresa al estado en que se encontraba desde el última transacción.

Servicios Web: Un servicio Web es una colección de protocolos y estándares que sirve para intercambiar datos entre aplicaciones.

Scripts: Un conjunto de comandos escritos en un lenguaje interpretado para automatizar ciertas tareas de aplicación.

Trazabilidad: Aptitud de reconstruir la historia, la utilización o la localización de un producto por medio de identificaciones registradas.

Triggers: Una acción que causa la invocación automática de un procedimiento, por ejemplo conservar la integridad referencial.

Tupla: Es una hilera o fila en una tabla

**Anexo 1.- Descripción de las clases persistentes.**

Tabla 19.- Clase ActaSindicato (*).

Nombre: ActaSindicato	
Atributo	Tipo
Numero	integer
Fecha	date
Direccion	string

Tabla 20.- Clase ActaUJC (*).

Nombre: ActaUJC	
Atributo	Tipo
Numero	integer
Fecha	date
Direccion	string

Tabla 21.- Clase AcuerdoSindicato (*).

Nombre: AcuerdoSindicato	
Atributo	Tipo
ID_Acuerdo	integer
Fecha	date
Responsable	string
Texto	text

Tabla 22.- Clase AcuerdosUJC (*).

Nombre: AcuerdosUJC	
Atributo	Tipo
ID_Acuerdo	integer
Acuerdo	string
FechaCumplimiento	date
Responsable	string

Tabla 23.- Clase Anno (*).

Nombre: Anno	
Atributo	Tipo
IDAnno	integer
Descripcion	string



Tabla 24.- Clase Apto.

Nombre: Apto	
Atributo	Tipo
ID_Apto	integer
Capacidad	integer
Telefono	string
Sexo	string

Tabla 25.- Clase Asignatura (*).

Nombre: Asignatura	
Atributo	Tipo
ID_Asignatura	integer
Nombre	string
Semestre	integer
Anno	string

Tabla 26.- Clase Cliente (*).

Nombre: Cliente	
Atributo	Tipo
ID_Cliente	integer
Nombre	string
Contacto	text

Tabla 27.- Clase Comision (*).

Nombre: Comision	
Atributo	Tipo
ID_Comision	integer
Nombre	string
Aula	string

Tabla 28.- Clase Comision_Trabajo.

Nombre: Comision_Trabajo	
Atributo	Tipo
Resultado	string

Tabla 29.- Clase Comite_Base_UJC (*).

Nombre: Comite_Base_UJC	
Atributo	Tipo
IDCB	integer
Nombre	string



Tabla 30.- Clase Competencia_Conocimientos.

Nombre: Competencia_Conocimientos	
Atributo	Tipo
Materia	string

Tabla 31.- Clase Cuarteleria (*).

Nombre: Cuarteleria	
Atributo	Tipo
Id_Cuarteleria	integer
Fecha	date

Tabla 32.- Clase Cumplimiento.

Nombre: Cumplimiento	
Atributo	Tipo
Cumplimiento	bit
Mes	string

Tabla 33.- Clase Curso (*).

Nombre: Curso	
Atributo	Tipo
ID_Curso	integer
Programa_Analitico	string
Capacidad	integer
Nombre	string

Tabla 34.- Clase Curso_Optativo.

Nombre: Curso_Optativo	
Atributo	Tipo
Nota	integer
Tipo	bit

Tabla 35.- Clase Departamento (*).

Nombre: Departamento	
Atributo	Tipo
Id_Departamento	integer
Descripcion	string
Nombre	string
Facultad	integer

Tabla 36.- Clase Deporte (*).



Nombre: Deporte	
Atributo	Tipo
ID_Deporte	integer
Deporte	string

Tabla 37.- Clase Edificio.

Nombre: Edificio	
Atributo	Tipo
ID_Edificio	integer

Tabla 38.- Clase RubroTrabajador.

Nombre: RubroTrabajador	
Atributo	Tipo
Cuando	integer
Rubro	string

Tabla 39.- Clase Encuesta (*).

Nombre: Encuesta	
Atributo	Tipo
ID_Encuesta	integer

Tabla 40.- Clase Estudiante_Actividad_Ext.

Nombre: Estudiante_Actividad_Ext	
Atributo	Tipo
Actividad_Extension	string

Tabla 41.- Clase Estudiante_Asignatura.

Nombre: Estudiante_Asignatura	
Atributo	Tipo
Nota	integer
Concepto	string

Tabla 42.- Clase Estudiante_Cuarteleria.

Nombre: Estudiante_Cuarteleria	
Atributo	Tipo
Evaluacion	integer

Tabla 43.- Clase Estudiante_Proyecto.

Nombre: Estudiante_Proyecto	
Atributo	Tipo
Evaluacion	integer



Baja	string
Rol_Proyecto	string

Tabla 44.- Clase Estudiante_TSU.

Nombre: Estudiante_TSU	
Atributo	Tipo
Evaluacion	string

Tabla 45.- Clase Evaluacion_Limpieza_OrdenInterior (*).

Nombre: Evaluacion_Limpieza_OrdenInterior	
Atributo	Tipo
Id_Evaluacion	integer
Evaluacion	string
Mes	string

Tabla 46.- Clase Evaluacion_Profesor.

Nombre: Evaluacion_Profesor	
Atributo	Tipo
Formacion	text
Evaluacion_Formacion	string
Produccion	text
Evaluacion_Produccion	string
Ciencia_Tecnica	text
Evaluacion_CT	string
Extensión_Universitaria	text
Evaluacion_EU	string
Superacion	text
Evaluacion_Superacion	string
Otros_Aspectos	text
Recomendaciones	text
Evaluacion_Final	string
Conforme	bit
Apela	bit
Evaluador	string
Mes	integer

Tabla 47.- Clase Evento (*).

Nombre: Evento	
Atributo	Tipo
ID_Evento	integer
Nombre	string



Fecha_Inicio	date
Fecha_Fin	date
Descripcion	text
Tipo	bit
Nivel	string

Tabla 48.- Clase EventoExt_Trabajo.

Nombre: EventoExt_Trabajo	
Atributo	Tipo
Resultado	string

Tabla 49.- Clase Grupo_Trabajo (*).

Nombre: Grupo_Trabajo	
Atributo	Tipo
ID_Grupo_Trabajo	integer
Especializacion	string
Nombre	string
Linea_Investigacion	string
Eliminado	string

Tabla 50.- Clase Grupo_Trabajo_Persona.

Nombre: Grupo_Trabajo_Persona	
Atributo	Tipo
Baja	string

Tabla 51.- Clase GT_Tarea.

Nombre: GT_Tarea	
Atributo	Tipo
Cumplida	string

Tabla 52.- Clase Guardia_Estudiante.

Nombre: Guardia_Estudiante	
Atributo	Tipo
Cumplimiento	string
Turno	integer
Posta	integer

Tabla 53.- Clase Guardia_Estudantil (*).

Nombre: Guardia_Estudantil	
Atributo	Tipo
Id_Guardia	integer



Observaciones	string
---------------	--------

Tabla 54.- Clase Horario.

Nombre: Horario	
Atributo	Tipo
ID_Clase	integer
Turno	integer
Dia	string
Semana	integer

Tabla 55.- Clase Laboratorio.

Nombre: Laboratorio	
Atributo	Tipo
ID_Laboratorio	integer

Tabla 56.- Clase Laboratorio_Proyecto.

Nombre: Laboratorio_Proyecto	
Atributo	Tipo
CantidadPC	integer

Tabla 57.- Clase Local (*).

Nombre: Local	
Atributo	Tipo
ID_Local	integer
Local	string
Capacidad	integer
TipoLocal	string

Tabla 58.- Clase Manifestacion_Cultural (*).

Nombre: Manifestacion_Cultural	
Atributo	Tipo
ID_Manifestacion_Cultural	integer
Manifestacion	string

Tabla 59.- Clase Militancia (*).

Nombre: Militancia	
Atributo	Tipo
ID_Militancia	integer
Descripcion	string

Tabla 60.- Clase MilitanteUJC.



Nombre: MilitanteUJC	
Atributo	Tipo
Cargo	string

Tabla 61.- Clase P1 (*).

Nombre: P1	
Atributo	Tipo
ID_P1	integer
Duracion	integer
Semestre	integer
Direccion	string

Tabla 62.- Clase Participa.

Nombre: Participa	
Atributo	Tipo
Resultado	string

Tabla 63.- Clase Paso_Escalera.

Nombre: Paso_Escalera	
Atributo	Tipo
ID_Paso_Escalera	integer

Tabla 64.- Clase Perfil (*).

Nombre: Perfil	
Atributo	Tipo
ID_Perfil	integer
Nombre	string
Cantidad_Cursos	integer

Tabla 65.- Clase Plan_De_Trabajo_AA (*).

Nombre: Plan_De_Trabajo_AA	
Atributo	Tipo
ID_Plan	integer
Plan	text

Tabla 66.- Clase Plan_De_Trabajo_Profesor (*).

Nombre: Plan_De_Trabajo_Profesor	
Atributo	Tipo
ID_Plan	integer
Plan	text



Tabla 67.- Clase Profesor_Proyecto.

Nombre: Profesor_Proyecto	
Atributo	Tipo
Rol_Proyecto	string

Tabla 68.- Clase Proyecto (*).

Nombre: Proyecto	
Atributo	Tipo
ID_Proyecto	integer
Nombre	string
Fecha_Inicio	date
Fecha_Fin	date
Tipo_Proyecto	string
Linea_Investigacion	string

Tabla 69.- Clase Publicación (*).

Nombre: Publicación	
Atributo	Tipo
ID_Publicacion	integer
Fecha	date
Descripcion	text
Nombre	string

Tabla 70.- Clase Revista (*).

Nombre: Revista	
Atributo	Tipo
ID_Revista	integer
Nombre	string
Pais	string

Tabla 71.- Clase Sancion (*).

Nombre: Sancion	
Atributo	Tipo
ID_Sancion	integer
Fecha	date
Causa	string
Inciso	integer

Tabla 72.- Clase Sancion_UJC (*).

Nombre: Sancion_UJC



Atributo	Tipo
ID_Sancion	integer
Fecha	date
CausaSancion	text
TipoSancion	string

Tabla 73.- Clase Semestre (*).

Nombre: Semestre	
Atributo	Tipo
ID_Semestre	integer
Descripcion	string

Tabla 74.- Clase Sindicato (*).

Nombre: Sindicato	
Atributo	Tipo
ID_Sindicato	integer
Nombre	string

Tabla 75.- Clase Tarea (*).

Nombre: Sindicato	
Atributo	Tipo
ID_Tarea	integer
Tarea	text

Tabla 76.- Clase Tesis (*).

Nombre: Tesis	
Atributo	Tipo
ID_Tesis	integer
Titulo	string
Ubicacion	string
Clasificacion	string
Area	string
Tipo_Tesis	string

Tabla 77.- Clase Tiene_Afectacion.

Nombre: Tiene_Afectacion	
Atributo	Tipo
Dia_Semana	string
Frecuencia	string
Afectacion	string

Tabla 78.- Clase Trabajo (*).



Nombre: Trabajo	
Atributo	Tipo
ID_Trabajo	integer
Nombre	string
Descripcion	text

Tabla 79.- Clase TSU (*).

Nombre: TSU	
Atributo	Tipo
Id_TSU	integer
Tarea	string
Fecha	date

Tabla 80.- Clase Visita (*).

Nombre: Visita	
Atributo	Tipo
ID_Visita	integer
Criterio	text
Evaluacion	string

Tabla 81.- Clase Web (*).

Nombre: Web	
Atributo	Tipo
ID_Web	integer
Nombre	string
Direccion_Electronica	string

Tabla 82.- Resolucion (*).

Nombre: Resolucion.	
Atributo	Tipo
ID_Resolucion	integer
Fecha	date
Tipo_Resolucion	string
Documento	string

Tabla 83.- tb_Usuario.

Nombre: tb_Usuario.	
Atributo	Tipo
Usuario	string
Contrasenna	string

Tabla 84.- tb_Rol (*).



Nombre: tb_Rol.	
Atributo	Tipo
Id_Rol	integer
Descripcion	string

Tabla 85.- tb_Pagina (*).

Nombre: tb_Pagina.	
Atributo	Tipo
Id_Pagina	integer
Nombre_Pagina	string

Anexo 2.- Descripción de las tablas de la Base de Datos.

Tabla 86.- ActaSindicato.

Nombre: ActaSindicato		
Descripción: Almacena los datos de las actas de los reuniones de un sindicato.		
Atributo	Tipo	Descripción
Numero	integer	El id de ActaPCC.
Fecha	date	La fecha en que se confecciona.
Direccion	string	La dirección donde se encuentra el archivo del acta.

Tabla 87.- ActaUJC

Nombre: ActaUJC		
Descripción: Almacena los datos de las actas de los reuniones de un comité de la UJC..		
Atributo	Tipo	Descripción
Numero	integer	El id de ActaPCC.
Fecha	date	La fecha en que se confecciona.
Direccion	string	La dirección donde se encuentra el archivo del acta.

Tabla 88.- Actividad.

Nombre: Actividad		
Descripción: Almacena los datos de las actividades.		
Atributo	Tipo	Descripción
ID_Actividad	integer	El id de Actividad.
ID_TipoActividad	date	El id del tipo de actividad.
ID_P1	string	El id del P1
Semana	integer	Semana en que se realiza la actividad.

Tabla 89.- Actividades_de_Extension.



Nombre: Actividades_de_Extension.		
Descripción: Almacena los datos de las actividades de extensión.		
Atributo	Tipo	Descripción
ID_Actividad	integer	El id de Actividades_de_Extension
Actividad	string	El texto de la actividad.

Tabla 90.- AcuerdoSindicato.

Nombre: AcuerdoSindicato.		
Descripción: Almacena los acuerdos de las de las actas del sindicato de las reuniones.		
Atributo	Tipo	Descripción
ID_Acuerdo	integer	El id de Acuerdo.
Numero	integer	El id de ActaSindicato.
Texto	text	El contenido del acuerdo.
Fecha	date	La fecha de cumplimiento del acuerdo.
Responsable	string	El responsable del acuerdo.

Tabla 91.- AcuerdosUJC.

Nombre: AcuerdosUJC.		
Descripción: Almacena los acuerdos de las de las actas de la UJC de las reuniones.		
Atributo	Tipo	Descripción
ID_Acuerdo	integer	El id de Acuerdo.
Numero	integer	El id de ActaUJC.
Acuerdo	text	El contenido del acuerdo.
FechaCumplimiento	date	La fecha de cumplimiento del acuerdo.
Responsable	string	El responsable del acuerdo.

Tabla 92.- Afectación.

Nombre: Afectacion.		
Descripción: Almacena las afectaciones de los profesores y alumnos ayudantes.		
Atributo	Tipo	Descripción
ID_Afectacion	integer	El id de Afectacion.
Afectacion	string	El contenido de la afectación.

Tabla 93.- Anno.

Nombre: Anno.		
Descripción: Almacena los datos de los años.		
Atributo	Tipo	Descripción
IDAnno	integer	El id de Anno.
Descripcion	string	El nombre del año.

Tabla 94.- Apto.



Nombre: Apto.		
Descripción: Almacena los datos de los apartamentos.		
Atributo	Tipo	Descripción
ID_Apto	integer	El id de Apto
Capacidad	integer	La capacidad del apartamento.
Telefono	string	El teléfono del apartamento.
ID_Paso_Escalera	integer	El id de Paso Escalera.
ID_Edificio	integer	El id de Edificio.
Identificador	string	El id de estudiante jefe de apto.

Tabla 95.- Area

Nombre: Area.		
Descripción: Almacena los datos de las áreas en que se realizan las tesis.		
Atributo	Tipo	Descripción
ID_Area	integer	El id de Area.
Area	string	La descripción del área.

Tabla 96.- Autor.

Nombre: Autor.		
Descripción: Almacena los datos del autor de la publicación.		
Atributo	Tipo	Descripción
ID_Publicacion	integer	El id de Publicacion.
Identificador	string	El id de la Persona.

Tabla 97.- CargoUJC

Nombre: CargoUJC.		
Descripción: Almacena los datos de los cargos de un comité de base UJC.		
Atributo	Tipo	Descripción
IDCargoUJC	integer	El id de CargoUJC.
Cargo	string	El nombre del cargo.
Descripción	string	La descripción del cargo.

Tabla 98.- Causa.

Nombre: Causa.		
Descripción: Almacena las causas de la sanción de un estudiante.		
Atributo	Tipo	Descripción
ID_Causa	integer	El id de Causa.
Tipo	string	La causa de la sanción.

Tabla 99.- Clasificacion.

Nombre: Clasificacion.



Descripción: Almacena los datos de la clasificación de una tesis.		
Atributo	Tipo	Descripción
ID_Clasificacion	integer	El id de Clasificacion.
Clasificacion	string	El nombre de la clasificación.

Tabla 100.- Cliente.

Nombre: Cliente.		
Descripción: Almacena los datos del cliente.		
Atributo	Tipo	Descripción
ID_Cliente	integer	El id de Cliente.
Nombre	string	El nombre del cliente.
Contacto	text	Forma de contacto del cliente.
ID_Representante	integer	El id de Representante.

Tabla 101.- Comision.

Nombre: Comision.		
Descripción: Almacena los datos de las comisiones.		
Atributo	Tipo	Descripción
ID_Comision	integer	El id de Comision.
Nombre	string	El nombre de la comisión.
Aula	string	Aula en que sesiona la comisión.

Tabla 102.- Comision_Trabajo.

Nombre: Comision_Trabajo.		
Descripción: Almacena los datos de las comisiones de trabajo.		
Atributo	Tipo	Descripción
ID_Comision	integer	El id de Comision.
ID_Trabajo	Integer	El id de Trabajo.
Resultado	string	El resultado de la comisión en un trabajo.

Tabla 103.- Comite_Base_UJC

Nombre: Comite_Base_UJC.		
Descripción: Almacena los datos de los comités de base de la UJC.		
Atributo	Tipo	Descripción
IDCB	integer	El id de Comite_Base_UJC.
Nombre	string	El nombre del comité de base UJC.

Tabla 104.- Competencia_Conocimientos.

Nombre: Competencia_Conocimientos.		
Descripción: Almacena los datos de las competencias de conocimientos.		
Atributo	Tipo	Descripción



Materia	string	La materia de esa competencia de conocimientos.
ID_Evento	integer	El id de Evento.

Tabla 105.- Cuarteleria.

Nombre: Cuarteleria.		
Descripción: Almacena los datos del autor de la publicación.		
Atributo	Tipo	Descripción
Id_Cuarteleria	integer	El id de Cuarteleria.
Fecha	date	La fecha en que se realiza la cuartelaría.

Tabla 106.- Cumplimiento.

Nombre: Cumplimiento.		
Descripción: Almacena los datos de si un trabajador es cumplidor.		
Atributo	Tipo	Descripción
ID_Mes	integer	El id de Mes.
IDAnno	integer	El id de Anno.
Cumplidor	bit	Si es cumplidor o no.
Identificador	string	El id del trabajador.

Tabla 107.- Curso.

Nombre: Curso.		
Descripción: Almacena los datos de los cursos optativos.		
Atributo	Tipo	Descripción
ID_Curso	integer	El id de Curso.
Programa_Analitico	string	El programa analítico del curso optativo.
Capacidad	integer	La capacidad del curso.
Nombre	string	El nombre del curso.
ID_Perfil	integer	El id de Perfil.
AsignaturaBasica	string	Indica si es asignatura básica ese curso.

Tabla 108.- Curso_Academico

Nombre: Curso_Academico.		
Descripción: Almacena los datos de los cursos académicos.		
Atributo	Tipo	Descripción
ID_Curso_Academico	integer	El id de Curso_Academico.
Descripción	string	El nombre del curso académico.

Tabla 109.- Curso_Optativo.

Nombre: Curso_Optativo.		
Descripción: Almacena los datos de los cursos optativos.		



Atributo	Tipo	Descripción
ID_Curso	integer	El id de Curso.
Identificador	string	El id de la Estudiante que recibe el curso optativo.
Identificador	string	El id del Docente que imparte el curso optativo.
Nota	integer	La nota en ese curso optativo.
Tipo	integer	El tipo de curso optativo.

Tabla 110.- Departamento.

Nombre: Departamento.		
Descripción: Almacena los datos de un departamento.		
Atributo	Tipo	Descripción
Id_Departamento	integer	El id de Departamento.
Descripcion	string	La descripción del departamento.
Nombre	string	El nombre del departamento.
IDFacultad	integer	El id de Facultad.
Identificador	string	El identificador de Profesor que es jefe de departamento.

Tabla 111.- Deporte.

Nombre: Deporte.		
Descripción: Almacena los datos de un deporte.		
Atributo	Tipo	Descripción
ID_Deporte	integer	El id de Deporte.
Deporte	string	El nombre del deporte.

Tabla 112.- Deporte_Encuesta.

Nombre: Deporte_Encuesta.		
Descripción: Almacena los datos de las encuestas de deporte.		
Atributo	Tipo	Descripción
ID_Encuesta	integer	El id de Encuesta.
ID_Deporte	integer	El id de la Deporte.

Tabla 113.- Dia.

Nombre: Dia.		
Descripción: Almacena los datos de los días de la semana.		
Atributo	Tipo	Descripción
ID_Dia	integer	El id de Dia.
Dia	string	El día de la semana.
ID_Semana	integer	El id de Semana.

Tabla 114.- Edificio.



Nombre: Edificio.		
Descripción: Almacena los datos de un edificio.		
Atributo	Tipo	Descripción
ID_Edificio	integer	El id de Edificio.
Identificador	string	El id del Estudiante jefe de edificio.

Tabla 115.- Rubro.

Nombre: Rubro.		
Descripción: Almacena los datos de los rubros.		
Atributo	Tipo	Descripción
ID_Rubro	integer	El id de Rubro.
RubroEmulacion	string	La descripción del rubro.

Tabla 116.- Encuesta.

Nombre: Encuesta.		
Descripción: Almacena los datos de la encuesta.		
Atributo	Tipo	Descripción
ID_Encuesta	integer	El id de Encuesta.
Identificador	string	El id del Estudiante.

Tabla 117.- EstadoCivil.

Nombre: EstadoCivil.		
Descripción: Almacena los datos del estado civil de una persona.		
Atributo	Tipo	Descripción
IDEstadoCivil	integer	El id de EstadoCivil.
Estado	string	El estado civil.

Tabla 118.- Estudiante_Actividad_Ext.

Nombre: Estudiante_Actividad_Ext.		
Descripción: Almacena los datos de las actividades de extensión.		
Atributo	Tipo	Descripción
ID_Actividad	integer	El id de Estudiante_Actividad_Ext.
ID_Curso_Academico	integer	El id de Curso_Academico
Identificador	string	El id de Estudiante.

Tabla 119.- Estudiante_Asignatura.

Nombre: Estudiante_Asignatura.		
Descripción: Almacena los datos de las asignaturas de los estudiantes.		
Atributo	Tipo	Descripción
ID_Asignatura	integer	El id de Asignatura.
Nota	integer	La nota de la asignatura.



Concepto	string	El concepto de la nota.
Identificador	string	El id de Estudiante.

Tabla 120.- Estudiante_Cuarteleria.

Nombre: Estudiante_Cuarteleria.		
Descripción: Almacena los datos de la cuartería de los estudiantes.		
Atributo	Tipo	Descripción
Id_Cuarteleria	integer	El id de Estudiante_Cuarteleria.
Evaluacion	string	La evaluación en la cuartería.
Identificador	string	El id de Estudiante.

Tabla 121.- Estudiante_Proyecto.

Nombre: Estudiante_Proyecto.		
Descripción: Almacena los datos de la estudiantes en proyecto.		
Atributo	Tipo	Descripción
ID_Proyecto	integer	El id de Proyecto.
Evaluacion	integer	La evaluación en el proyecto.
Baja	string	Si causo baja del proyecto.
Identificador	string	El id de Estudiante.
Id_rol_proyecto	integer	El id de Rol_Proyecto.

Tabla 122.- Estudiante_Tesis.

Nombre: Estudiante_Tesis.		
Descripción: Almacena los datos de la tesis de los estudiantes.		
Atributo	Tipo	Descripción
ID_Tesis	integer	El id de Tesis.
Identificador	string	El id de Estudiante.

Tabla 123.- Estudiante_Trabajo.

Nombre: Estudiante_Trabajo.		
Descripción: Almacena los datos de los trabajos de los estudiantes.		
Atributo	Tipo	Descripción
ID_Trabajo	integer	El id de Trabajo.
Identificador	string	El id de Estudiante.

Tabla 124.- Estudiante_TSU.

Nombre: Estudiante_TSU.		
Descripción: Almacena los datos de los TSU de los estudiantes.		
Atributo	Tipo	Descripción
Id_TSU	integer	El id de TSU.
Identificador	string	El id de Estudiante.



Evaluacion	string	La evaluación en el TSU.
------------	--------	--------------------------

Tabla 125.- Evaluacion_AA.

Nombre: Evaluacion_AA.		
Descripción: Almacena los datos de evaluación de los alumnos ayudantes.		
Atributo	Tipo	Descripción
Trabajo_Docente	text	El trabajo docente desempeñado.
Identificador	string	El id de Estudiante.
ID_Semestre	integer	El id de Semestre.
ID_Curso_Academico	integer	El id de Curso_Academico
Evaluacion_TD	text	La evaluación del trabajo docente.
Trabajo_Cientifico_Productivo	text	El trabajo científico productivo desempeñado.
Evaluacion_TCP	string	Evaluación del trabajo científico productivo.
Evaluacion_Integral	string	Evaluación integral del alumno ayudante.
IDAnno	integer	El id de Anno.

Tabla 126.- Evaluacion_Limpieza_OrdenInterior.

Nombre: Evaluacion_Limpieza_OrdenInterior.		
Descripción: Almacena los datos de la evaluación de la limpieza en los aptos.		
Atributo	Tipo	Descripción
Id_Evaluacion	Integer	El id de Evaluacion_Limpieza_OrdenInterior.
Identificador	string	El id de Estudiante.
Id_Mes	integer	El id de Mes.
ID_Curso_Academico	integer	El id de Curso_Academico
Evaluacion	string	La evaluación en la limpieza.

Tabla 127.- Evaluacion_Profesor.

Nombre: Evaluacion_Profesor.		
Descripción: Almacena los datos de evaluación de los profesores.		
Atributo	Tipo	Descripción
Formacion	text	La formación del profesor.
Identificador	string	El id de Profesor.
ID_Curso_Academico	integer	El id de Curso_Academico
Evaluacion_Formacion	string	La evaluación de la formación.
Producción	text	La producción del profesor.
Evaluacion_Produccion	string	La evaluación de la producción.
Ciencia_Tecnica	text	El desempeño en la ciencia y técnica.
Evaluacion_CT	string	Evaluación de la ciencia y técnica.
Extensión_Universitaria	text	El desempeño en la extensión universitaria.
Evaluacion_EU	string	La evaluación en la extensión universitaria.
Superacion	text	El desempeño en la superación.
Evaluacion_Superacion	string	La evaluación de la superación.



Otros_Aspectos	text	Otros aspectos a destacar.
Recomendaciones	text	Recomendaciones.
Evaluacion_Final	string	Evaluación final.
Conforme	bit	Si esta conforme o no con su evaluación.
Apela	bit	Si desea apelar o no a la evaluación.
Evaluador	string	El que realizo la evaluación.
Mes	integer	El id de Evaluacion_Profesor.

Tabla 128.- Evento

Nombre: Evento.		
Descripción: Almacena los datos de los eventos.		
Atributo	Tipo	Descripción
ID_Evento	integer	El id de Evento.
Nombre	string	El nombre del evento.
Fecha_Inicio	date	La fecha de inicio del evento.
Fecha_Fin	date	La fecha de fin del evento.
Descripcion	text	La descripción del evento.
ID_Nivel	integer	El id de Nivel.
Tipo	string	El tipo de evento.

Tabla 129.- EventoExt_Trabajo

Nombre: EventoExt_Trabajo.		
Descripción: Almacena los datos de los trabajos presentados en los eventos externos.		
Atributo	Tipo	Descripción
ID_Trabajo	integer	El id de Trabajo.
Resultado	string	El resultado del trabajo.
ID_Evento	integer	El id de Evento.

Tabla 130.- Facultad

Nombre: Facultad.		
Descripción: Almacena los datos del número de las facultades.		
Atributo	Tipo	Descripción
IDFacultad	integer	El id de Facultad.

Tabla 131.- Frecuencia.

Nombre: Frecuencia.		
Descripción: Almacena los datos de las frecuencias de las afectaciones.		
Atributo	Tipo	Descripción
id_frecuencia	integer	El id de Frecuencia.
Descripcion	string	La frecuencia de la afectación..

Tabla 132.- Grupo_Trabajo.



Nombre: Grupo_Trabajo.		
Descripción: Almacena los datos de los grupos de trabajo.		
Atributo	Tipo	Descripción
ID_Grupo_Trabajo	integer	El id de Grupo_Trabajo.
Especializacion	string	La especialización del grupo de trabajo.
Nombre	string	El nombre del grupo de trabajo.
ID_Linea	integer	El id de Linea_Investigacion.

Tabla 133.- Grupo_Trabajo_Persona.

Nombre: Grupo_Trabajo_Persona.		
Descripción: Almacena los datos de las personas que están en grupos de trabajo.		
Atributo	Tipo	Descripción
ID_Grupo_Trabajo	Integer	El id de Grupo_Trabajo.
Identificador	string	El id de Persona.
Baja	string	Si resultado baja del grupo de trabajo.

Tabla 134.- GT_Tarea.

Nombre: GT_Tarea.		
Descripción: Almacena los datos de las tareas de los grupos de trabajo.		
Atributo	Tipo	Descripción
ID_Proyecto	integer	El id de Proyecto.
ID_Grupo_Trabajo	Integer	El id de Grupo_Trabajo.
ID_Tarea	integer	El id de Tarea.
Cumplida	string	Si se cumplió la tarea.

Tabla 135.- Guardia_Brigada.

Nombre: Guardia_Brigada.		
Descripción: Almacena los datos de las guardias de las brigadas.		
Atributo	Tipo	Descripción
ID_Guardia	integer	El id de Guardia_Estudiantil.
ID_Brigada	string	El id de Brigada.

Tabla 136.- Guardia_Estudiente.

Nombre: Guardia_Estudiente.		
Descripción: Almacena los datos de las guardias de los estudiantes.		
Atributo	Tipo	Descripción
ID_Turno	integer	El id de Turno.
Identificador	string	El id de Estudiante.
ID_Posta	integer	El id de Posta.
ID_Guardia	integer	El id de Guardia_Estudiantil.
Cumplimiento	string	Si cumplió o no el estudiante su guardia.



Tabla 137.- Guardia_Estudiantil.

Nombre: Guardia_Estudiantil.		
Descripción: Almacena los datos de las guardias.		
Atributo	Tipo	Descripción
ID_Guardia	integer	El id de Guardia_Estudiantil.
Identificador	string	El id de Profesor.
Observaciones	string	Las observaciones de la guardia.

Tabla 138.- Imparte_Asignatura.

Nombre: Imparte_Asignatura.		
Descripción: Almacena los datos de los docentes que imparten una asignatura.		
Atributo	Tipo	Descripción
ID_Asignatura	integer	El id de Asignatura.
Identificador	string	El id de Docente.

Tabla 139.- Imparte_Clase.

Nombre: Imparte_Clase.		
Descripción: Almacena los datos de los docentes que imparten a una brigada.		
Atributo	Tipo	Descripción
ID_Asignatura	integer	El id de Asignatura.
Identificador	string	El id de Docente.

Tabla 140.- Inciso.

Nombre: Inciso.		
Descripción: Almacena los datos de los incisos de las sanciones.		
Atributo	Tipo	Descripción
ID_Inciso	char	El id de Inciso.

Tabla 141.- Jurado.

Nombre: Jurado.		
Descripción: Almacena los datos de los jurados de las comisiones.		
Atributo	Tipo	Descripción
ID_Comision	integer	El id de Comision.
Identificador	string	El id de Persona.

Tabla 142.- Jurado_Profesor.

Nombre: Jurado_Profesor.		
Descripción: Almacena los datos de los profesores que son jurados del evento.		
Atributo	Tipo	Descripción
ID_Evento	integer	El id de Evento.



Identificador	string	El id de Profesor.
---------------	--------	--------------------

Tabla 143.- Laboratorio.

Nombre: Laboratorio.		
Descripción: Almacena los datos de los laboratorios.		
Atributo	Tipo	Descripción
ID_Laboratorio	integer	El id de Laboratorio.

Tabla 144.- Laboratorio_Proyecto.

Nombre: Laboratorio_Proyecto.		
Descripción: Almacena los datos de los laboratorios de proyectos.		
Atributo	Tipo	Descripción
ID_Laboratorio	integer	El id de Laboratorio.
ID_Proyecto	integer	El id de Proyecto.
cantidadPC	integer	Cantidad de máquinas.

Tabla 145.- Linea_Investigacion.

Nombre: Linea_Investigacion.		
Descripción: Almacena los datos de las líneas de investigación de las tesis.		
Atributo	Tipo	Descripción
ID_Linea	integer	El id de Linea_Investigacion.
Nombre	string	El nombre de la línea de investigación.

Tabla 146.- Local.

Nombre: Local.		
Descripción: Almacena los datos de los locales de recibir clases.		
Atributo	Tipo	Descripción
ID_Local	integer	El id de Local.
ID_Tipo_Local	integer	El id de Tipo_Local.
Local	String	El nombre del local.

Tabla 147.- Manifestacion_Cultural.

Nombre: Manisfestacion_Cultural.		
Descripción: Almacena los datos de las manifestaciones culturales.		
Atributo	Tipo	Descripción
ID_Manifestacion_Cultural	integer	El id de Manisfestacion_Cultural.
Manifestacion	string	El nombre de la manifestación.

Tabla 148.- Manisfestacion_Encuesta.

Nombre: Manifestacion_Encuesta.		
Descripción: Almacena los datos de las encuestas de las manifestaciones culturales.		



Atributo	Tipo	Descripción
ID_Encuesta	integer	El id de Encuesta.
ID_Manifestacion_Cultural	integer	El id de Manifestación_Cultural.

Tabla 149.- Mes.

Nombre: Mes.		
Descripción: Almacena los datos de los meses.		
Atributo	Tipo	Descripción
Id_Mes	integer	El id de Mes.
Nombre	string	El nombre del mes.

Tabla 150.- Militancia.

Nombre: Militancia.		
Descripción: Almacena los datos de la militancia.		
Atributo	Tipo	Descripción
ID_Militancia	integer	El id de Militancia.
Descripcion	string	La descripción.

Tabla 151.- Militancia_Persona.

Nombre: Militancia_Persona.		
Descripción: Almacena los datos de la militancia de las personas.		
Atributo	Tipo	Descripción
ID_Militancia	integer	El id de Militancia.
Identificador	string	El id de Persona.

Tabla 152.- MilitanteUJC.

Nombre: MilitanteUJC.		
Descripción: Almacena los datos de los militantes UJC.		
Atributo	Tipo	Descripción
IDCB	integer	El id de Comite_Base_UJC.
IDCargoUJC	integer	El id de CargoUJC.
Identificador	string	El id de Persona.

Tabla 153.- Nivel.

Nombre: Nivel.		
Descripción: Almacena los datos del nivel de los eventos.		
Atributo	Tipo	Descripción
ID_Nivel	integer	El id de Nivel.
Nivel	string	El nivel del evento.

Tabla 154.- Ojos.



Nombre: Ojos.		
Descripción: Almacena los datos del color de los ojos de las personas.		
Atributo	Tipo	Descripción
IDOjos	integer	El id de Ojos.
Color	string	El color de los ojos.

Tabla 155.- P1

Nombre: P1.		
Descripción: Almacena los datos de los P1 de las asignaturas.		
Atributo	Tipo	Descripción
ID_P1	integer	El id de P1.
Duracion	integer	La duración del P1.
Semestre	integer	El semestre en que se va a impartir el P1.
Direccion	string	La dirección del archivo del P1.
ID_Asignatura	integer	El id de Asignatura.

Tabla 156.- Participa.

Nombre: Participa.		
Descripción: Almacena los datos de la participación en la competencia de conocimientos.		
Atributo	Tipo	Descripción
ID_Evento	integer	El id de Competencia_Conocimientos.
Resultado	string	El resultado en la competencia.
Identificador	string	El id de Estudiante.

Tabla 157.- Paso_Escalera.

Nombre: Paso_Escalera.		
Descripción: Almacena los datos de los pasos de escalera de los edificios de los estudiantes.		
Atributo	Tipo	Descripción
ID_Paso_Escalera	integer	El id de Paso_Escalera.
ID_Edificio	integer	El id de Edificio.
Identificador	string	El id de Estudiante.
Sexo	bit	El sexo de los estudiantes de ese paso de escalera.

Tabla 158.- Perfil.

Nombre: Perfil.		
Descripción: Almacena los datos de los perfiles de los cursos optativos.		
Atributo	Tipo	Descripción
ID_Perfil	integer	El id de Perfil.
Nombre	string	El nombre del perfil.
Cantidad_Cursos	integer	La cantidad de cursos de ese perfil.



Tabla 159.- Plan_De_Trabajo.

Nombre: Plan_De_Trabajo.		
Descripción: Almacena los datos de los planes de trabajo.		
Atributo	Tipo	Descripción
ID_Plan	integer	El id de Plan_De_Trabajo.
Plan	text	El contenido del plan.
Identificador	string	El id de Persona.
ID_Semestre	integer	El id de Semestre.
ID_Curso_Academico	integer	El id de Curso_Academico.

Tabla 160.- Posta.

Nombre: Posta.		
Descripción: Almacena los datos de las postas en las guardias.		
Atributo	Tipo	Descripción
ID_Posta	integer	El id de Posta.
Posta	string	Ubicación de la posta.

Tabla 161.- Profesor_Proyecto.

Nombre: Profesor_Proyecto.		
Descripción: Almacena los datos de los profesores en proyecto.		
Atributo	Tipo	Descripción
ID_Proyecto	integer	El id de Proyecto.
Evaluacion	string	La evaluación en ese proyecto.
id_rol_proyecto	integer	El id de Rol_Proyecto.
Identificador	string	El id de Profesor.

Tabla 162.- Publicación.

Nombre: Publicacion.		
Descripción: Almacena los datos de las publicaciones.		
Atributo	Tipo	Descripción
ID_Publicacion	integer	El id de Publicación.
Fecha	date	La fecha de la publicación.
Descripción	text	La descripción de la publicación.
Nombre	string	El nombre de la publicación.

Tabla 163.- Publicacion_Revista.

Nombre: Publicación_Revista.		
Descripción: Almacena los datos de las publicaciones de las revistas.		
Atributo	Tipo	Descripción
ID_Publicacion	integer	El id de Publicacion.
ID_Revista	integer	El id de Revista.



Tabla 164.- Publicacion_Web.

Nombre: Publicación_Web.		
Descripción: Almacena los datos de las publicaciones de las webs.		
Atributo	Tipo	Descripción
ID_Publicacion	integer	El id de Publicacion.
ID_Web	integer	El id de Web.

Tabla 165.- Raza.

Nombre: Raza.		
Descripción: Almacena los datos de las razas.		
Atributo	Tipo	Descripción
IDRaza	integer	El id de Raza.
Raza	string	El nombre de la raza.

Tabla 166.- Representante.

Nombre: Representante.		
Descripción: Almacena los datos de los representantes.		
Atributo	Tipo	Descripción
ID_Representante	integer	El id de Representante.
Nombre	string	El nombre del representante.

Tabla 167.- Revista.

Nombre: Revista.		
Descripción: Almacena los datos de las revistas.		
Atributo	Tipo	Descripción
Pais	string	El país de la publicación de la revista.
ID_Revista	integer	El id de Revista.
Nombre	string	El nombre de la revista.

Tabla 168.- Rol.

Nombre: Rol.		
Descripción: Almacena los datos de los roles.		
Atributo	Tipo	Descripción
ID_Rol	integer	El id de Rol.
Rol	string	El nombre del rol.

Tabla 169.- Rol_Proyecto

Nombre: Rol_Proyecto.		
Descripción: Almacena los datos de los roles de proyecto.		
Atributo	Tipo	Descripción



id_rol_proyecto	integer	El id de Rol_Proyecto.
Descripcion	string	La descripción del rol de proyecto.

Tabla 170.- Rol_Sindicato.

Nombre: Rol_Sindicato.		
Descripción: Almacena los datos de los roles del sindicato.		
Atributo	Tipo	Descripción
ID_Rol_Sindicato	integer	El id de Rol_Sindicato.
Rol	integer	El nombre del rol del sindicato.

Tabla 171.- Sancion.

Nombre: Sancion.		
Descripción: Almacena los datos de las sanciones a los estudiantes.		
Atributo	Tipo	Descripción
ID_Sancion	integer	El id de Sancion.
Fecha	date	La fecha de la sanción.
ID_Causa	integer	El id de Causa.
ID_Inciso	char	El id de Inciso.
Identificador.	string	El id de Estudiante.

Tabla 172.- Sancion_UJC.

Nombre: Sancion_UJC.		
Descripción: Almacena los datos de las sanciones de la UJC de los militantes.		
Atributo	Tipo	Descripción
Id_Sancion	integer	El id de Sancion_UJC.
ID_TipoSancion	integer	El id de Tipo_Sancion.
Fecha	date	La fecha de la sanción.
CausaSancion	text	La causa de la sanción.
IDCB	integer	El id de MilitanteUJC.
Identificador	string	El id de MilitanteUJC.

Tabla 173.- Semana.

Nombre: Semana.		
Descripción: Almacena los datos de las semanas.		
Atributo	Tipo	Descripción
ID_Semana	integer	El id de Semana.

Tabla 174.- Semestre.

Nombre: Semestre.		
Descripción: Almacena los datos del semestre.		
Atributo	Tipo	Descripción



ID_Semestre	integer	El id de Semestre.
Descripcion	string	La descripción del semestre.

Tabla 175.- Sindicato.

Nombre: Sindicato.		
Descripción: Almacena los datos del sindicato.		
Atributo	Tipo	Descripción
ID_Sindicato	integer	El id de Sindicato.
Nombre	string	El nombre del sindicato.

Tabla 176.- Tarea.

Nombre: Tarea.		
Descripción: Almacena los datos de las tareas de los grupos de trabajo.		
Atributo	Tipo	Descripción
ID_Tarea	integer	El id de Tarea.
Tarea	text	La descripción de la tarea.

Tabla 177.- Tesis.

Nombre: Tesis.		
Descripción: Almacena los datos de las tesis de las personas.		
Atributo	Tipo	Descripción
ID_Tesis	integer	El id de Tesis.
Titulo	string	El título de los tesis.
ID_Area	integer	El id de Area.
ID_Clasificacion	integer	El id de Clasificacion.
ID_Ubicacion	integer	El id de Ubicacion.
ID_Tipo	integer	El id de Tipo_Tesis.
Identificador	string	El id de persona.

Tabla 178.- Tiene_Afectacion.

Nombre: Tiene_Afectacion.		
Descripción: Almacena los datos de las afectaciones de los profesores y alumnos ayudantes.		
Atributo	Tipo	Descripción
ID_Afectacion	integer	El id de Afectacion.
Id_frecuencia	integer	El id de Frecuencia.
Identificador	string	El id de Docente.
dia_semana	string	El día se la afectación.
id_tiene_afectacion	integer	El id de Tiene_Afectacion.

Tabla 179.- Tipo_Local.

Nombre: Tipo_Local.		
----------------------------	--	--



Descripción: Almacena los datos de los tipos de locales.		
Atributo	Tipo	Descripción
ID_Tipo_Local	integer	El id de Tipo_Local.
Tipo	string	El tipo de local.

Tabla 180.- Tipo_Proyecto.

Nombre: Tipo_Proyecto.		
Descripción: Almacena los datos de los tipos de proyecto.		
Atributo	Tipo	Descripción
ID_Tipo	integer	El id de Tipo_Proyecto.
Tipo	string	El tipo de proyecto.

Tabla 181.- Tipo_Tesis.

Nombre: Tipo_Tesis.		
Descripción: Almacena los datos de los tipos de tesis.		
Atributo	Tipo	Descripción
ID_Tipo	integer	El id de Tipo_Tesis.
Tipo	string	El tipo de tesis.

Tabla 182.- TipoActividad.

Nombre: TipoActividad.		
Descripción: Almacena los datos de los tipos de actividades.		
Atributo	Tipo	Descripción
ID_TipoActividad	integer	El id de TipoActividad.
Descripcion	string	La descripción del tipo de actividad.

Tabla 183.- TipoSancion.

Nombre: TipoSancion.		
Descripción: Almacena los datos de los tipos de sanciones.		
Atributo	Tipo	Descripción
ID_TipoSancion	integer	El id de TipoSancion.
Sancion	string	El tipo de sanción.

Tabla 184.- Trabajador.

Nombre: Trabajador.		
Descripción: Almacena los datos de los trabajadores.		
Atributo	Tipo	Descripción
ID_Sindicato	integer	El id de Sindicato.
ID_Rol_Sindicato	integer	El id de Rol_Sindicato.
Identificador	string	El id de Persona.

Tabla 185.- Trabajador_Guardia.



Nombre: Trabajador_Guardia.		
Descripción: Almacena los datos de los trabajadores de guardia.		
Atributo	Tipo	Descripción
ID_Guardia	integer	El id de Guardia_Estudiantil.
Identificador	string	El id de Trabajador.

Tabla 186.- Trabajo.

Nombre: Trabajo.		
Descripción: Almacena los datos de los trabajos de los eventos.		
Atributo	Tipo	Descripción
ID_Trabajo	integer	El id de Trabajo.
Nombre	string	El nombre de los trabajos.
Descripción	text	La descripción de los trabajos.

Tabla 187.- TSU.

Nombre: TSU.		
Descripción: Almacena los datos de los TSU de los estudiantes.		
Atributo	Tipo	Descripción
Id_TSU	integer	El id de TSU.
ID_Curso_Academico	integer	El id de Curso_Academico.
Tarea	string	La descripción de la tarea.
Fecha	date	La fecha del TSU.

Tabla 188.- Turno.

Nombre: Turno.		
Descripción: Almacena los datos de los turnos de clases.		
Atributo	Tipo	Descripción
ID_Turno	integer	El id de Turno.
ID_Dia	integer	El id de Dia.
ID_Semana	integer	El id de Semana.

Tabla 189.- Turno_Guardia.

Nombre: Turno_Guardia.		
Descripción: Almacena los datos de los turnos de guardia.		
Atributo	Tipo	Descripción
ID_Turno	integer	El id de Turno_Guardia.

Tabla 190.- Ubicacion.

Nombre: Ubicacion.		
Descripción: Almacena los datos de las ubicaciones de las tesis.		
Atributo	Tipo	Descripción



ID_Ubicacion	integer	El id de Ubicacion.
Ubicacion	string	La ubicación.

Tabla 191.- Visita.

Nombre: Visita.		
Descripción: Almacena los datos de las visitas a clases.		
Atributo	Tipo	Descripción
ID_Visita	integer	El id de Visita.
Criterio	text	El criterio de la visita.
Evaluacion	string	La evaluación de la visita.
Fecha	date	La fecha de la visita
Identificador	string	El id de Docente, del visitado.

Tabla 192.- Visita_Profesor.

Nombre: Visita_Profesor.		
Descripción: Almacena los datos de las visitas de los profesores.		
Atributo	Tipo	Descripción
ID_Visita	integer	El id de Visita.
Identificador	string	El id de Profesor, que visita.

Tabla 193.- Web.

Nombre: Web.		
Descripción: Almacena los datos de las webs.		
Atributo	Tipo	Descripción
id_Web	integer	El id de Web.
Nombre	string	El nombre de la web.
Direccion_electronica	string	La dirección electrónica de la web.

Tabla 194.- Tipo_Resolucion.

Nombre: Tipo_Resolucion.		
Descripción: Almacena los datos de los tipos de resolución decanal.		
Atributo	Tipo	Descripción
id_tipo_resolucion	integer	El id de Tipo_Resolucion.
Descripcion	string	La descripción del tipo de resolución.

Tabla 195.- Resolucion.

Nombre: Resolucion.		
Descripción: Almacena los datos de las resoluciones decanales aplicadas a los estudiantes.		
Atributo	Tipo	Descripción
id_resolucion	integer	El id de Resolucion.
Identificador	string	El id de Estudiante.



Id_tipo_resolucion	integer	El id de Tipo_Resolucion.
Fecha	date	La fecha en que se aplicó la resolución.

Tabla 196.- Profesor_Maestria.

Nombre: Profesor_Maestria.		
Descripción: Almacena los datos de las maestrías por profesores.		
Atributo	Tipo	Descripción
Identificador	string	El id de Profesor.
Id_Maestria	integer	El id de Maestria

Tabla 197.- Maestria.

Nombre: Maestria.		
Descripción: Almacena los datos de las maestrías.		
Atributo	Tipo	Descripción
id_Maestria	integer	El id de Maestria.
Descripcion	string	La descripción de la maestría.

Tabla 198.- tb_Usuario.

Nombre: tb_Usuario.		
Descripción: Almacena los datos de los usuarios.		
Atributo	Tipo	Descripción
id_rol	integer	El id de tb_Rol.
Identificador	string	El id de Persona.
Usuario	string	El usuario de la persona.
Contraseña	string	La contraseña del usuario.

Tabla 199.- tb_Rol.

Nombre: tb_Rol.		
Descripción: Almacena los roles de los usuarios.		
Atributo	Tipo	Descripción
id_Rol	integer	El id de tb_Rol
Descripcion	string	La descripción del rol.

Tabla 200.- tb_Rol_Pagina.

Nombre: tb_Rol_Pagina.		
Descripción: Almacena los datos de las páginas a los que tienen acceso los roles.		
Atributo	Tipo	Descripción
id_Rol	integer	El id de Rol.
id_Pagina	integer	El id de tb_Pagina.

Tabla 201.- tb_Pagina.



Nombre: tb_Pagina.		
Descripción: Almacena los datos de las páginas.		
Atributo	Tipo	Descripción
id_Pagina	integer	El id de tb_Pagina.
Nombre_Pagina	string	El nombre de la página.

Tabla 202.- tb_Procedencia.

Nombre: tb_Procedencia.		
Descripción: Almacena los datos de las relaciones entre páginas.		
Atributo	Tipo	Descripción
id_Pagina	integer	El id de tb_Pagina.
id_Pagina	integer	El id de tb_Pagina.

Tabla 203.- RubroTrabajador.

Nombre: RubroTrabajador.		
Descripción: Almacena los datos de los rubros de los trabajadores.		
Atributo	Tipo	Descripción
Identificador	string	El id de Trabajador.
ID_Rubro	integer	El id de Rubro.
Cuando	integer	El semestre en que lo recibió.
ID_Curso_Academico	integer	El id de Curso_Academico



Anexo 3.- Modelos de objetos.

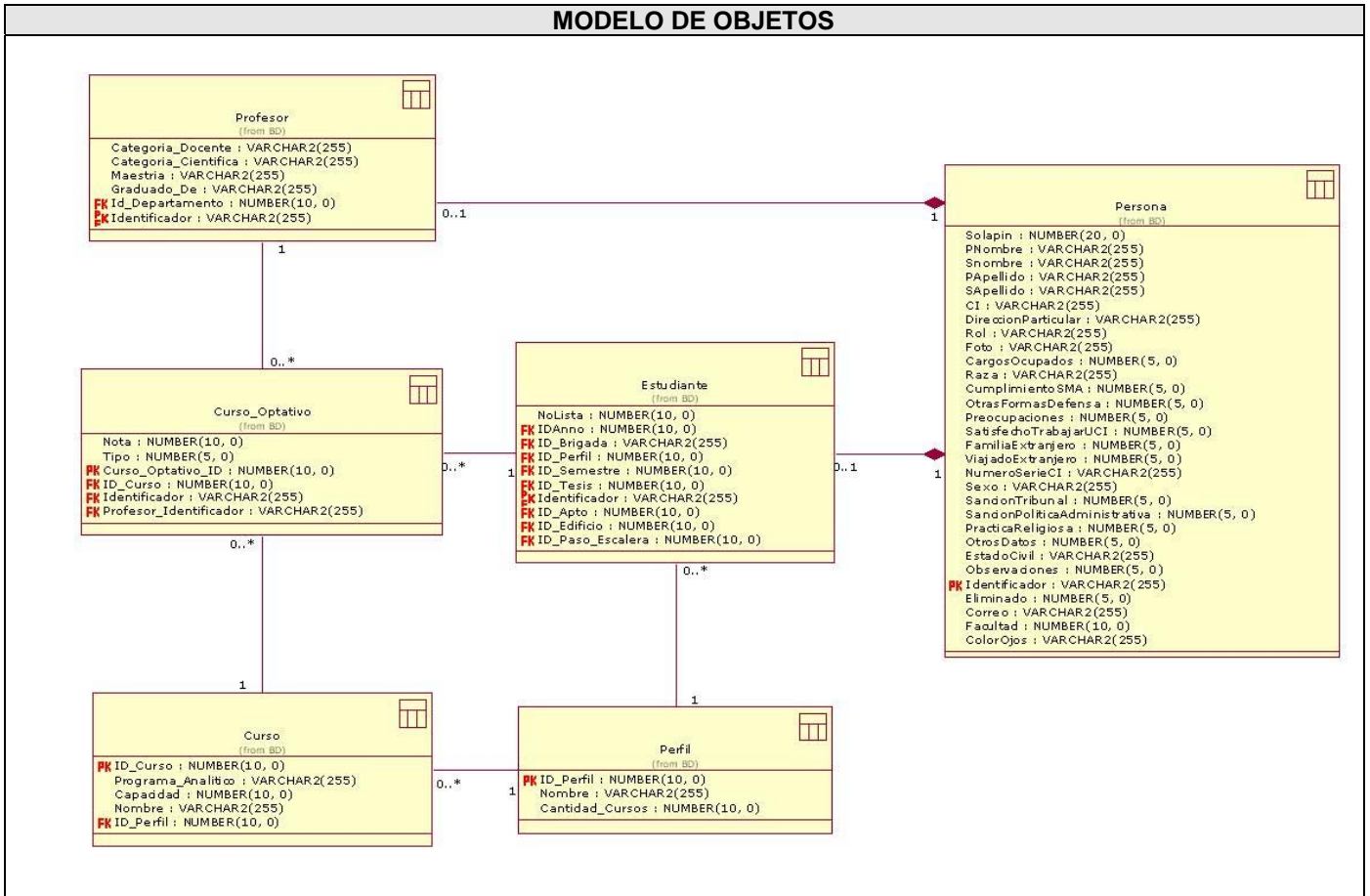


Figura 21.- Modelo de Objetos del módulo de Cursos Optativos.

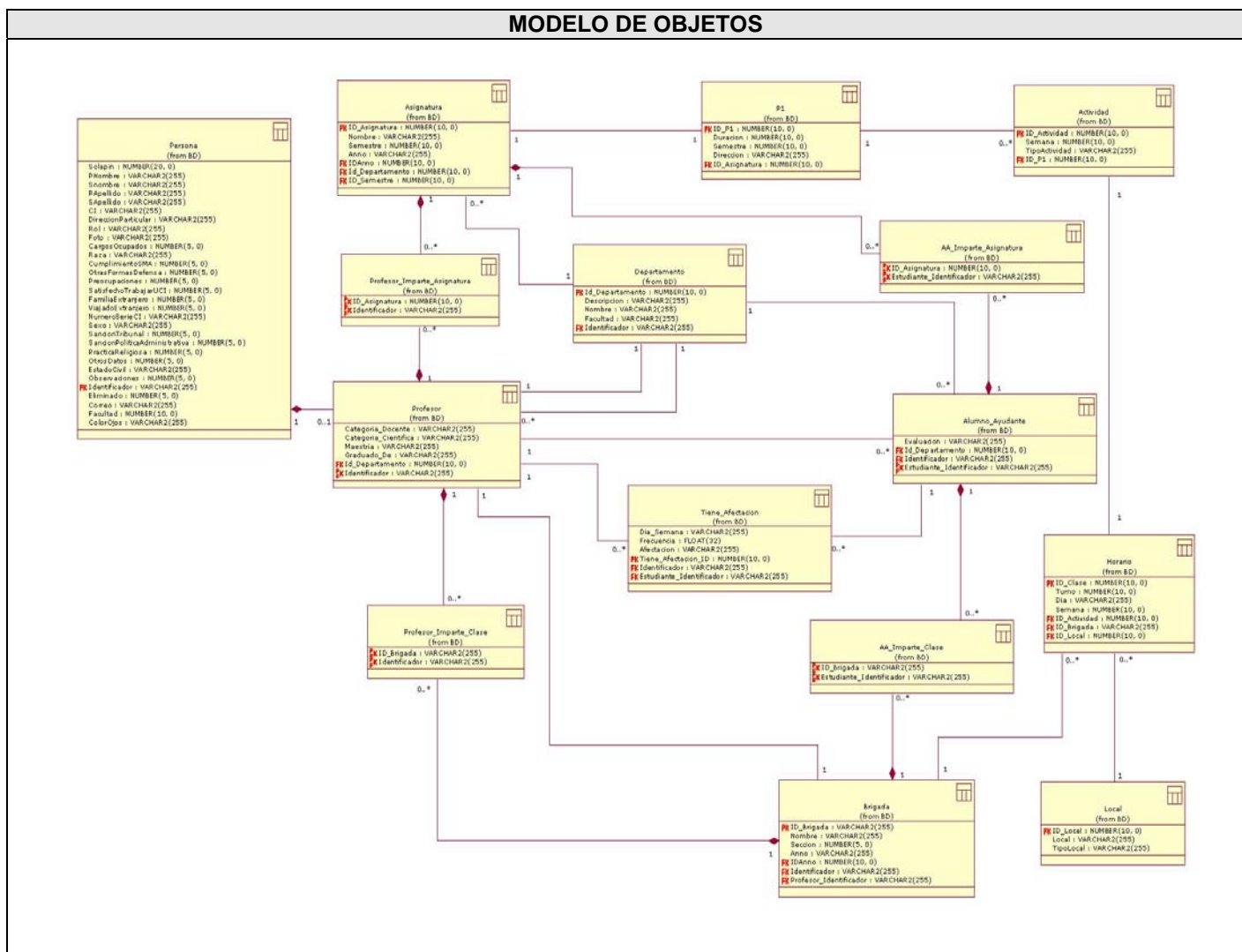


Figura 25.- Modelo de Objetos del módulo Planificación Docente.

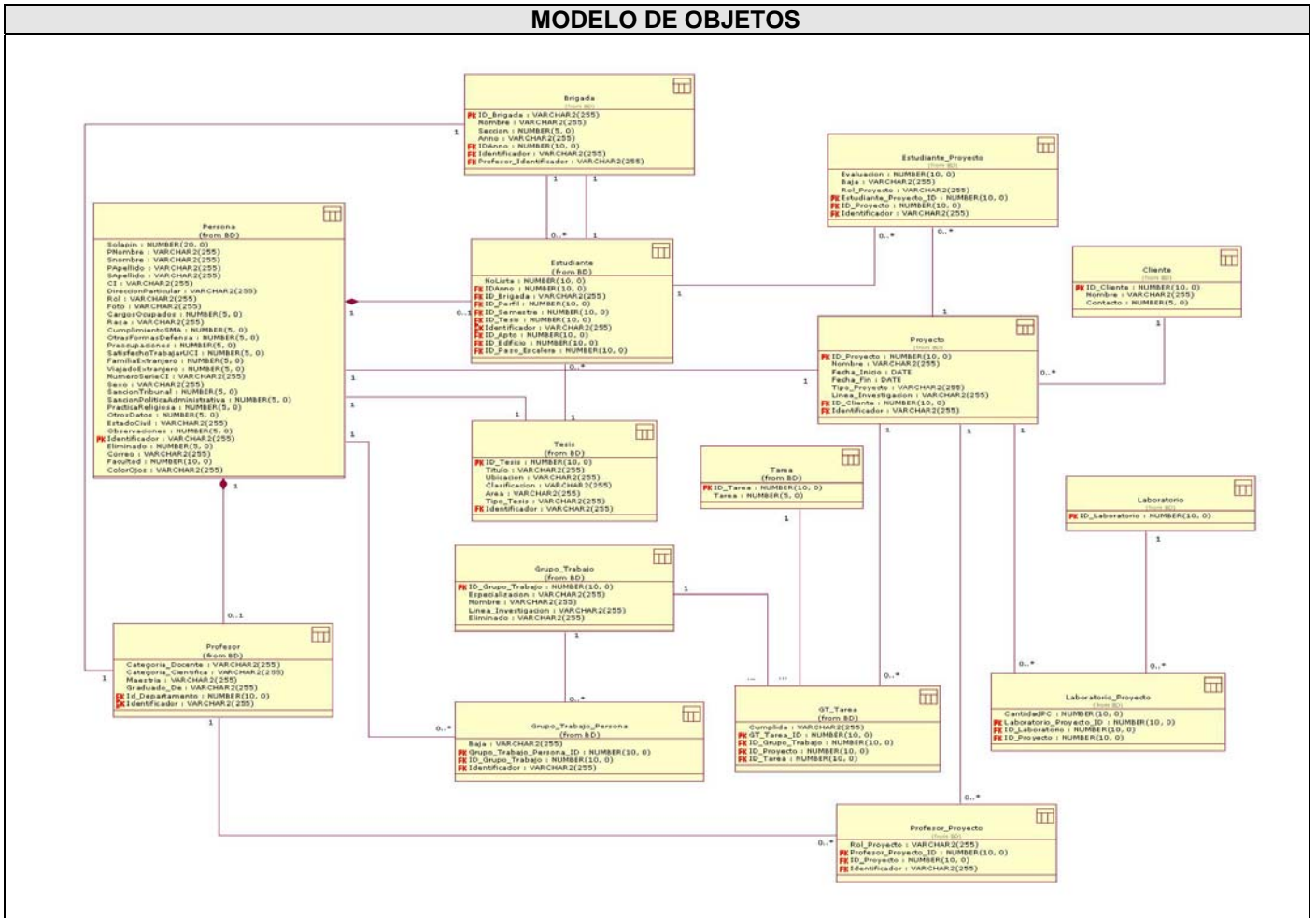


Figura 26.- Modelo de Objetos del módulo Proyectos Productivos.

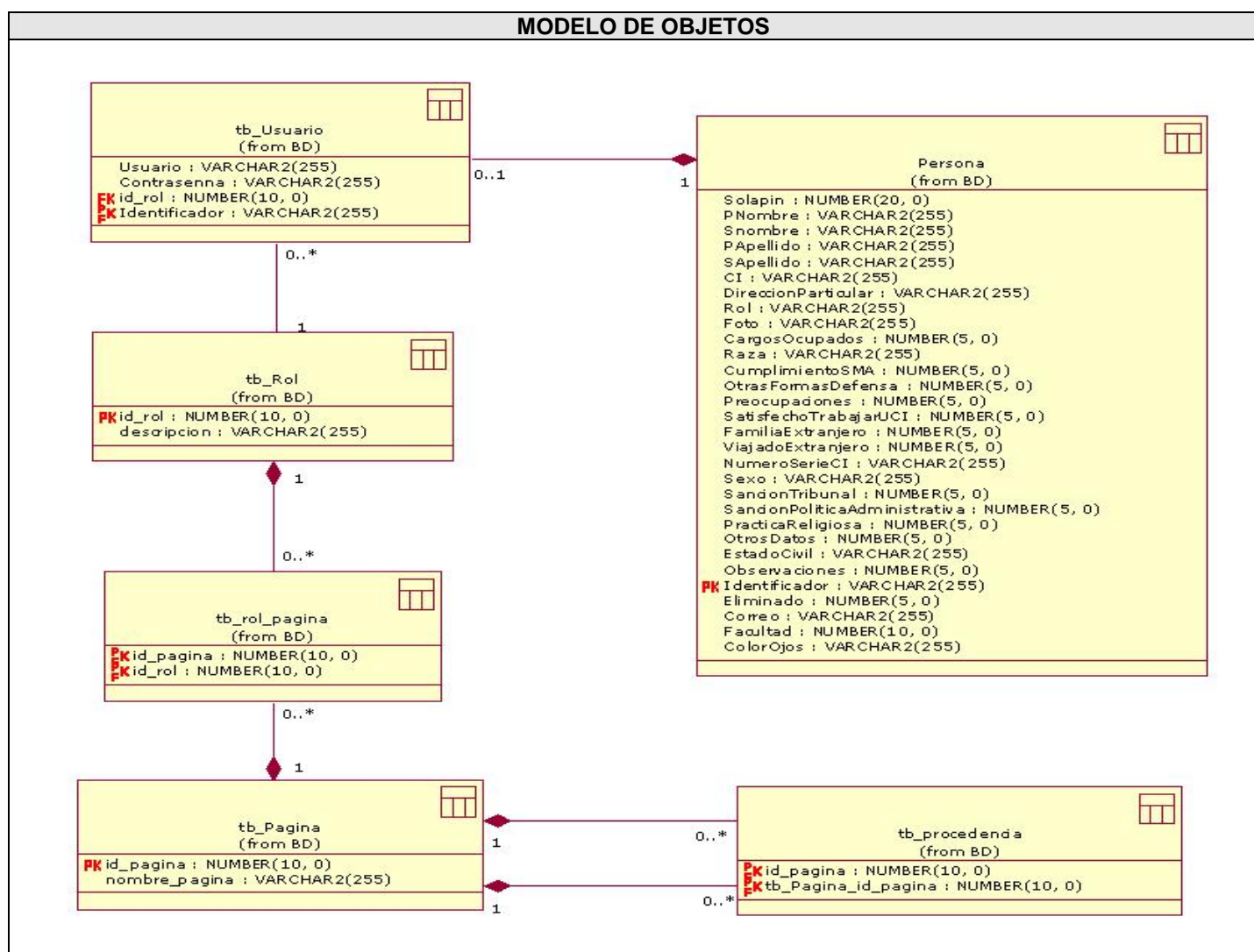


Figura 27.- Modelo de Objetos del módulo Seguridad.

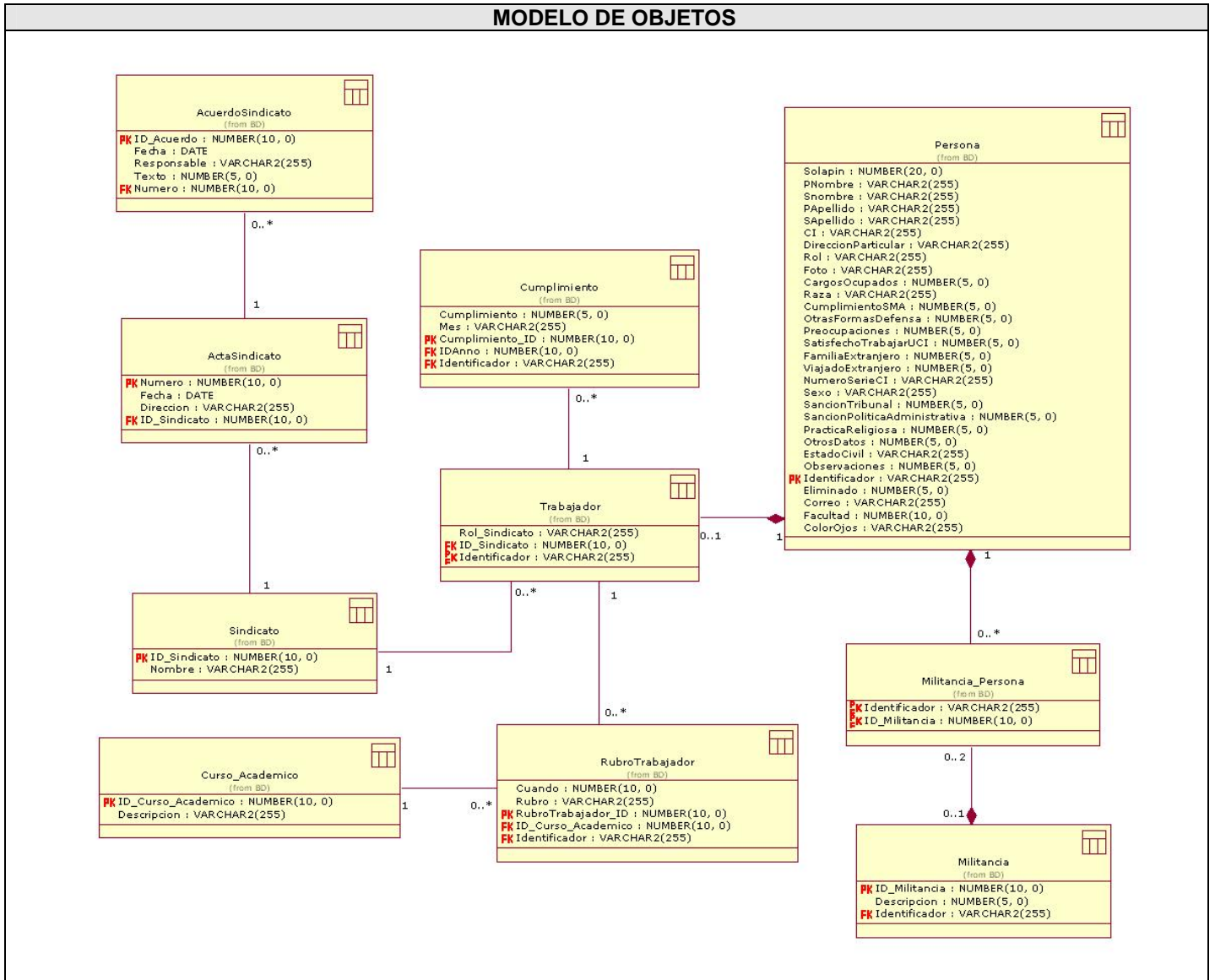


Figura 28.- Modelo de Objetos del módulo Sindicato.

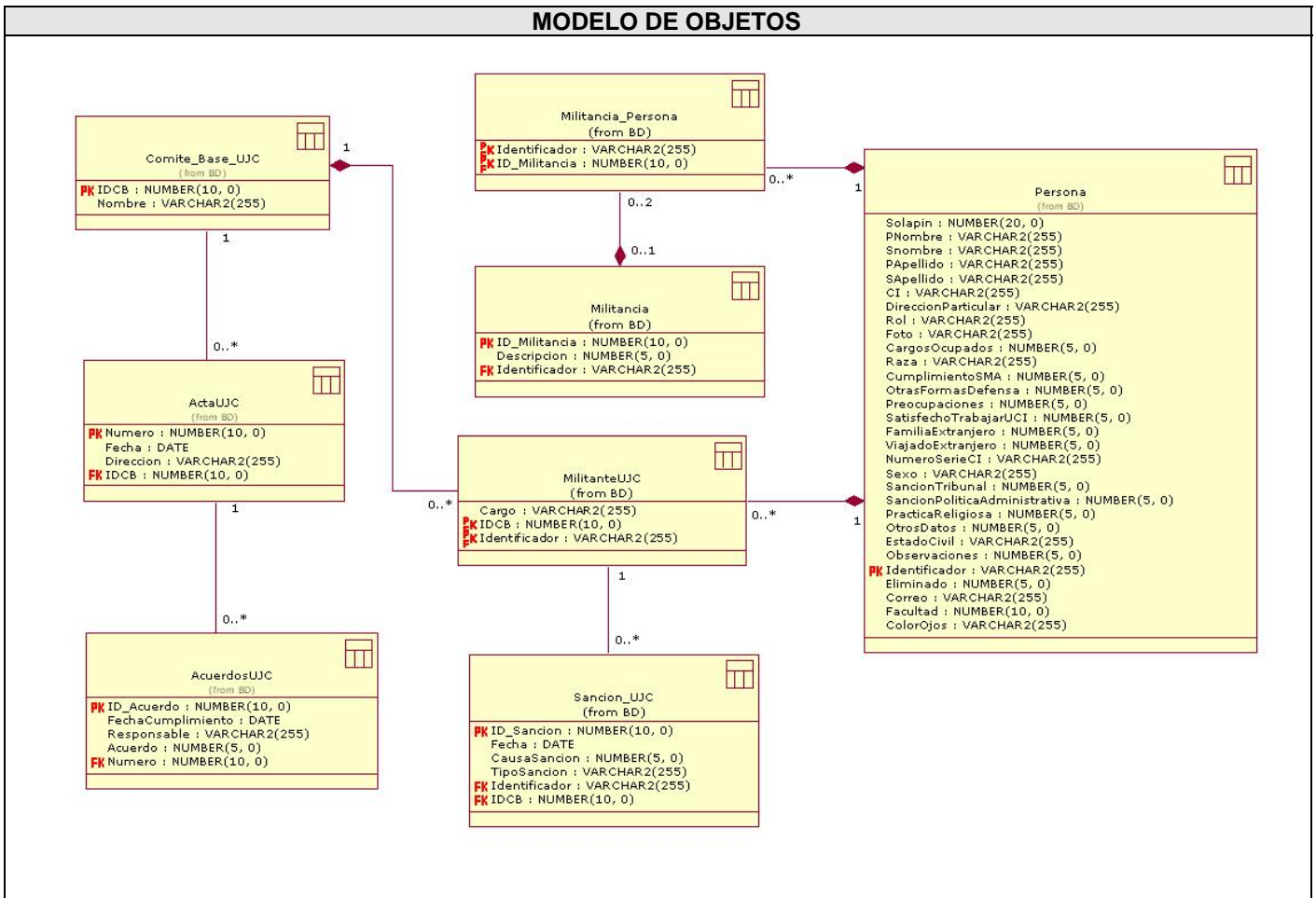


Figura 29.- Modelo de Objetos del módulo UJC.