

Universidad de las Ciencias Informáticas

Facultad 1



Título: Sistema Integrado de Cooperación Internacional.
Diseño de la base de datos.

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

Autores: Adennys Almaguer Esquivel
Arlen Tamayo Labrada

Tutores: Arturo Luis Lara Fernández.
Julio Cesar Isaza Vázquez

Ciudad de la Habana, 18 de junio del 2008.
“Año 50 de la Revolución”

DECLARACIÓN DE AUTORÍA

Por este medio declaramos que somos los únicos autores de este trabajo y que autorizamos a la Universidad de las Ciencias Informáticas (UCI) para que hagan el uso que estimen pertinente con este trabajo.

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

Adennys Almaguer Esquivel

Arlen Tamayo Labrada

Autora

Autor

Julio Cesar Isaza Vázquez

Arturo Luis Lara Fernández

Tutor

Tutor

OPINIÓN DEL TUTOR DEL TRABAJO DE DIPLOMA

Título: Sistema Informatizado de Cooperación Internacional. Diseño de la base de datos.

Autores: Adennys Almaguer Esquivel

Arlen Tamayo Labrada

Los tutores del presente Trabajo de Diploma consideran que durante su ejecución los estudiantes mostraron las cualidades que a continuación se detallan.

El proyecto que hoy se defiende constituye para la Universidad uno de los pilares fundamentales en el afán de convertirse en una ciudad digital, contar con un sistema como este permite además brindar soluciones a otras entidades y organismos nacionales que lo necesiten y con los cuales se puede empezar a trabajar.

Por trabajos como este en el cual los diplomantes han dado muestra de la calidad en su preparación y compromiso con el proyecto, se evidencia la importancia que tiene la Universidad en la informatización de la sociedad cubana.

El trabajo se desempeñó con un alto grado de independencia y con dominio del tema tratado. Tiene un alto valor creativo y originalidad, demostrado con los resultados obtenidos.

Cabe añadir que ambos tesisistas mantuvieron una actitud y disciplina excelente no solo a lo largo del desarrollo del trabajo de diploma, sino también en todo el tiempo que formaron parte del grupo de desarrollo del proyecto SICI.

Por todo lo anteriormente expresado considero que los estudiantes están aptos para ejercer como Ingenieros en Ciencias Informáticas; y propongo que se le otorgue al Trabajo de Diploma la calificación de _____ puntos.

Los resultados de este trabajo ameritan ser publicados por el tema específico en que tratan y ser presentados en eventos científicos y demás eventos afines.

Dado a los ____ días del mes de junio del año 2007

Ing. Julio Cesar Isaza

AGRADECIMIENTOS

De Adennys:

Quiero agradecer a todas las personas que me ayudaron a lograr el desarrollo de este trabajo hasta el final.

A Fidel, Raúl y a la Revolución por crear esta universidad y darme la oportunidad de formarme en ella.

A mis padres por apoyarme incondicionalmente, por darme todo el cariño que me dan y por hacerme fuerte para enfrentar esta distancia por 5 años. Por ser mis guías y mis mejores amigos.

A mis hermanos y a toda mi familia, por todo su cariño y atención. Y por darme aliento en los momentos difíciles.

A mi novio Arlen por apoyarme en todo lo que estuvo en sus manos, por su cariño y comprensión y por darme fuerzas para enfrentar todo los tropiezos en el camino. Por estar conmigo en las buenas y en las malas. Te quiero.

A todos los profesores que me formaron como futura profesional. A nuestro tutor Julio por su apoyo.

A todos mis amigos, que estuvieron ahí cuando los necesitaba para tenderme su mano, en especial a Yaimara, Yai has sido una amiga sin igual, gracias por ser mi amiga y gracias por todo, siempre tendrás un lugar especial en mi corazón.

De Arlen:

No puedo empezar sin antes agradecer a nuestro Comandante en Jefe Fidel Castro, gracias por la oportunidad de estar aquí escribiendo estas líneas. No te fallaremos nunca.

A mis queridos padres por darme todo su amor, por vivir mis preocupaciones como si fueran tuyas y prestarme sus fuerzas en cada momento. Por confiar en mí, en esto y mil cosas. Nunca lo hubiera logrado sin ustedes. Los quiero con todo el corazón.

A mi hermano Abelino que no se cansa de ser mi amigo y apoyarme. Eres el mejor hermano del mundo.

A mi novia Adennys por estar conmigo pese a todo y perdonarme mis inmadureces. Por repetirme siempre que soy único hasta hacer que me lo crea de verdad. Por quererme tan bonito y regalarme ese rincón de felicidad que es su abrazo. Mi niña como te dije un día: te quisiera aun sin haberte conocido.

A todos los profesores que me han brindado su ayuda y a nuestro tutor por su apoyo.

A todos mis amigos que en una medida u otra han estado conmigo a lo largo de estos 5 años compartiendo alegrías y tristezas y en especial a Ramón por su amistad sincera.

RESUMEN

El tema de la tesis es “Sistema Informatizado de Cooperación Internacional (SICI). Diseño de la Base de Datos”.

La investigación está dirigida a realizar el diseño de la Base de Datos que posteriormente permitirá almacenar toda la información que se maneja en la Dirección de Cooperación Internacional (DCI). Esta dirección se encarga de las actividades de cooperación internacional, relaciones públicas y de gestionar los trámites de las personas que viajan al exterior como parte de misiones orientadas por la Universidad de las Ciencias Informáticas (UCI), así como los de las personas que entran a la universidad a razón de algún intercambio de índole profesional.

En la DCI se almacena un gran volumen de información producto de las labores que en ella se realizan, mucha de esta información se guardan en archivos físicos o carpetas y en algunos casos en documentos Word y Excel. Además actualmente cuenta con una pequeña base de datos en un software propietario, que representa un modelo viejo del sistema, con reglas del negocio que no representan las necesidades reales de la dirección. Todo esto provoca problemas a la hora de guardar y recuperar la información dentro de la dirección para realizar alguna operación.

De ahí que el objetivo de la investigación sea Diseñar una Base de Datos que nos permita satisfacer las necesidades de almacenamiento, disponibilidad, integridad y seguridad de la información que se maneja en la DCI. Lo que nos da la medida de la importancia de la investigación puesto que en la dirección se maneja mucha información sensible para la universidad la cual requiere de un buen control y seguridad.

Los resultados relevantes de la investigación son el diseño de la base de datos. La reducción del empleo de recursos gastables y el tiempo de ejecución de las actividades que se llevan a cabo dentro de la DCI, además contribuir al logro de una mayor organización de la información, permitiendo un rápido acceso a ella. Este trabajo favorecerá también al proceso de informatización que se está llevando a cabo en la UCI.

Palabras claves: Base de datos, Cooperación internacional, Diseño, SICI.

Índice

Agradecimientos	III
Resumen.....	V
Introducción.....	1
Capítulo1. Fundamentación teórica.....	9
1.1 Introducción.....	9
1.2 Surgimiento histórico de las bases de datos	9
1.3 Actualidad de las bases de datos	10
1.3.1 Actualidad de las bases de datos en el mundo	10
1.3.2 Actualidad de las bases de datos en Cuba	11
1.3.3 Actualidad de las bases de datos en la UCI	13
1.4 Conceptos de base de datos	14
1.5 Metodologías	15
1.5.1 Metodología de diseño de bases de datos	15
1.5.2 Metodologías de desarrollo y lenguaje de modelado	16
1.6 Conceptos de sistema gestor de base de datos	18
1.7 Arquitectura de un SGBD	19
1.7.1 Arquitectura de las bases de datos relacionales	19
1.7.2 Componentes de la arquitectura de las bases de datos relacionales	20
1.7.3 Arquitectura del SGBD PostgreSQL.....	21
1.8 Características y objetivos del enfoque de los SGBD.....	21
1.9 Ventajas de utilizar un SGBD	23
1.10 Desventajas de utilizar un SGBD	24
1.11 Clasificación de los SGBD	25
1.12 Herramientas analizadas	26
1.12.1 Sistemas gestores de bases de datos	26
1.12.2 Herramientas de modelado	29
1.12.3 Sistemas de gestión de contenidos	30
1.13 Justificación de la propuesta de solución	31

1.14 Conclusiones.....	32
Capitulo 2. Descripción y análisis de la solución propuesta	33
2.1 Requisitos funcionales y no funcionales.....	33
2.1.1 Requisitos funcionales	33
2.1.2 Requisitos no funcionales	36
2.2 Estrategia de integración de la solución con otros módulos y sistemas	38
2.3 Descripción de la arquitectura y fundamentación	39
2.4 Diagrama de clases persistentes.....	40
2.5 Diagramas Entidad-Relación	44
2.5.1 Descripción de las tablas de la base de datos.	62
2.6 Análisis de optimización de consultas	67
2.7 Conclusiones.....	71
Capitulo 3. Validación del diseño realizado	72
3.1 Introducción.....	72
3.2 Validación teórica del diseño	72
3.2.1 Integridad.....	72
3.2.2 Normalización de la base de datos.....	76
3.2.3 Análisis de redundancia de información	80
3.2.4 Análisis de la seguridad de la base de datos	81
3.2.5 Trazabilidad de las acciones.....	83
3.3 Validación funcional	84
3.3.1 Generación de código de programas para un llenado voluminoso e inteligente de la base de datos	84
3.4 Conclusiones.....	85
Conclusiones.....	86
Recomendaciones	87
Bibliografía referenciada	88
Bibliografía consultada.....	89
Glosario de términos	90

Índice de figuras

Figura 1. 1 Arquitectura de un sistema gestor de bases de datos	19
Figura 2. 1 Diagrama de clases persistentes del módulo de Trámites.....	40
Figura 2. 2 Diagrama de clases persistentes del módulo de Economía	41
Figura 2. 3 Diagrama de clases persistentes del módulo de Cooperación	42
Figura 2. 4 Diagrama de clases persistentes del módulo de Relaciones Públicas.....	43
Figura 2. 5 Diagrama entidad relación del módulo de Trámites	44
Figura 2. 6 Diagrama entidad relación del módulo de Trámites continuación	45
Figura 2. 7 Diagrama entidad relación del módulo de Trámites continuación	46
Figura 2. 8 Diagrama entidad relación del módulo de Trámites continuación	47
Figura 2. 9 Diagrama entidad relación del módulo de Trámites fin.....	48
Figura 2. 10 Diagrama entidad relación del módulo de RRPP	49
Figura 2. 11 Diagrama entidad relación del módulo de RRPP continuación	50
Figura 2. 12 Diagrama entidad relación del módulo de RRPP fin.....	51
Figura 2. 13 Diagrama entidad relación del módulo de Economía.....	52
Figura 2. 14 Diagrama entidad relación del módulo de Economía continuación	53
Figura 2. 15 Diagrama entidad relación del módulo de Economía continuación	54
Figura 2. 16 Diagrama entidad relación del módulo de Economía continuación	55
Figura 2. 17 Diagrama entidad relación del módulo de Economía fin.....	56
Figura 2. 18 Diagrama entidad relación del módulo de Cooperación Internacional	57
Figura 2. 19 Diagrama entidad relación del módulo de Cooperación Internacional	58
Figura 2. 20 Diagrama entidad relación del módulo de Cooperación Internacional	59
Figura 2. 21 Diagrama entidad relación del módulo de Cooperación Internacional	60
Figura 2. 22 Diagrama entidad relación del módulo de Cooperación Internacional fin	61
Figura 3. 1 Ejemplo de transformación de atributos multivaluados	79
Figura 3. 2 Añadir un nuevo usuario mediante el gestor de usuario de postgresQL	82

Índice de tablas

Tabla 1 Operacionalización de las variables de la investigación	4
Tabla 2 Cronograma de trabajo de la investigación	7
Tabla 3 Descripción de la tabla visita	62
Tabla 4 Descripción de la tabla misionero	62
Tabla 5 Descripción de la tabla misión.....	64
Tabla 6 Descripción de la tabla persona	65
Tabla 7 Descripción de la tabla extranjero	65
Tabla 8 Descripción de la tabla visa	66
Tabla 9 Descripción de la tabla pasaporte	66
Tabla 10 Descripción de la tabla visitante.....	67
Tabla 11 Descripción de la tabla actividad.....	67

INTRODUCCIÓN

Desde tiempos muy remotos el hombre ha encontrado la manera de preservar la información que ha considerado vital, ya sea por considerarla como una simple actividad o por tener en cuenta la importancia que tiene. En todas las partes del mundo se puede encontrar información que pasa de generación en generación hasta que se pierde, pero no toda la información se considera como un simple recuerdo, también se puede hallar información del pasado que dice como eran aquellos tiempos o hasta información de gran impacto para todo aquel que viva en este planeta. Sin duda alguna, la información a hecho crecer como personas a la humanidad entera, gracias a ella se sabe de donde provenimos y como se ha estado evolucionando siempre apoyados en la información que nos dice lo que se puede o no hacer para lograr el éxito que se busca, de igual forma, toda organización por pequeña que sea, necesita saber su pasado, la situación actual en la que se encuentra y con que cuenta para afrontar el futuro. La información es un recurso vital y el buen manejo de esta puede significar la diferencia entre el éxito o el fracaso para todos los proyectos que se emprendan dentro de un organismo que busca el crecimiento y el éxito.

En la actualidad es evidente el desarrollo tecnológico y la necesidad de la informática en la vida del hombre, este desarrollo va en aumento. En todas partes del mundo se difunde ampliamente el uso de las tecnologías de la información y las comunicaciones (TICs) siendo esta la fuente de solución a disímiles problemas y para muchos ya constituye el día a día, debido a que es notable en el mundo el uso de sistemas informáticos para el desempeño de cualquier rol laboral.

Cuba desde los primeros años de la Revolución y quizás un poco antes, a pesar de ser un país subdesarrollado y actualmente bloqueado ha hecho un gran esfuerzo por propiciar e incentivar el uso de la informática. En años recientes han comenzado a tener un rápido y masivo auge en el país el estudio y la aplicación de las ciencias informáticas como parte de una estrategia que incluye desde extender a todos los sectores poblacionales del país, conocimientos básicos de esta temática hasta la creación de una universidad de excelencia como es la UCI.

La UCI está adscrita al Ministerio de la Informática y las Comunicaciones y tiene la misión de ser una universidad innovadora de excelencia científica, académica y productiva, que forma profesionales integrales comprometidos con la patria; creada para convertirse en soporte de la informatización del país y la competitividad internacional de la Industria Cubana del Software. En su estructura organizativa, cuenta con la DCI, que tiene como función principal la gestión de las actividades de colaboración internacional y relaciones públicas. Además esta dirección se ocupa específicamente de tramitar las salidas de las personas de la universidad por algún tipo de misión al extranjero y las

entradas de todas aquellas personas que a razón de un intercambio profesional o cultural lo requieren. Atienden las relaciones con otras instituciones, controla lo relativo a las becas, convenios, proyectos y cursos entre otros. La DCI pertenece a la Vicerrectoría de Investigaciones y Cooperación Internacional y está compuesta por estos tres grupos:

El Grupo de Cooperación Internacional tiene entre sus funciones gestionar la estrategia de colaboración internacional de la Universidad, coordinar la participación de la UCI en organizaciones internacionales de excelencia, diseñar y gestionar los vínculos inter-institucionales y contribuir a la identificación y gestión de programas de cooperación internacional.

El Grupo de Relaciones Públicas se encarga de dirigir todo lo relacionado con la proyección de la imagen institucional, la promoción y divulgación de las actividades de interés de la UCI. Además, este Grupo, en conjunto con el de Cooperación Internacional, está encargado de dirigir metodológica y funcionalmente el sistema de colaboración internacional de la UCI, gestionar y conformar la información de apoyo a los intercambios interinstitucionales, así como garantizar las acciones necesarias para el éxito de las misiones en el exterior y de sus visitantes en la universidad.

El Grupo de Trámites dirige los asuntos migratorios y de extranjería en la UCI. Se encarga de la documentación necesaria para que una persona pueda viajar al exterior o un extranjero pueda entrar al país por motivos de trabajo relacionados con la Universidad. Se elaboran los reportes estadísticos y económicos de las personas que viajan y las misiones, además de los partes sobre el estado de estas y del proceso de tramitación de los misioneros. Se controlan y archivan los documentos y la información obtenida.

En la DCI se maneja un gran volumen de información que se obtiene de cada uno de los grupos que la componen. Esta información que se genera se guarda en archivos físicos y carpetas, que pasado un tiempo la misma persona que la guardó tiene que hacer un gran esfuerzo para saber donde la ha guardado. Con la información digital pasa lo mismo, el personal de los grupos de trabajo tiene problemas a la hora de buscar los datos que necesita en el momento que los necesita. Además la documentación necesaria para que una persona de la universidad pueda viajar al extranjero o un extranjero pueda entrar a la universidad por razones de trabajo se llena de forma manual lo que provoca demora en la realización de las tareas. También al realizar estas prestaciones de forma manual no se tiene un control claro de la información a la hora de brindar los servicios al cliente y provoca que se incurran en errores muchas veces no detectables.

En sus inicios en la DCI no existía ninguna base de datos para almacenar toda esta información, por tal razón se guardaba en documentos de Office o herramientas similares, tampoco existía un sistema

que automatizara las funciones como brindar reportes estadísticos y económicos en la tramitación, lo cual producía información duplicada o con errores, lo que supone un gasto elevado de tiempo y recursos, además de lo engorroso a la hora de buscar algún dato.

Actualmente la DCI cuenta con una pequeña base de datos en un software propietario, que representa un modelo viejo del sistema, con reglas del negocio que no representan las necesidades reales del negocio. Por estas razones surge la necesidad de crear un sistema de almacenamiento mejor, que represente realmente el negocio y que satisfaga todas las necesidades de almacenamiento, disponibilidad, control y consistencia de la información que se maneja.

Debido a esta situación en la Facultad¹ se creó el proyecto Sistema Informatizado de Cooperación Internacional (SICI) el cual tiene la misión de darle solución a la problemática anteriormente planteada. Lo antes expuesto permitió identificar el siguiente **problema científico**: ¿Como diseñar un medio de almacenamiento que satisfaga las necesidades de almacenamiento, disponibilidad, integridad y seguridad de la información que se maneja en la DCI?

El **objeto de estudio** de la investigación son los procesos de almacenamiento y recuperación de la información en la DCI.

El **campo de acción** es el diseño del medio de almacenamiento y recuperación de la información en la DCI.

Para resolverlo se planteó como **objetivo** diseñar una Base de Datos (BD) para almacenar correctamente la información que se maneja en la DCI garantizando su disponibilidad, integridad y seguridad.

Partiendo de la **hipótesis** de que si se construye una BD que almacene correctamente la información, entonces se podrá garantizar la disponibilidad, integridad y seguridad de la información que se genera en la DCI.

Variables de la investigación

* Independiente: Base de Datos.

* Dependientes: Disponibilidad, integridad y seguridad de la información

Operacionalización de las variables de la investigación

Variable Conceptual	Dimensión	Indicador
Disponibilidad de la información.	-Factibilidad	-Tiempo de acceso. -Gestión de operaciones. -Tiempo de ejecución.
Integridad de la información.	-Acceso a la información.	-Complejidad de los datos. -Unicidad de los datos. -Irreductibilidad de los datos. -Transacciones válidas. -Eliminación de anomalías.
Seguridad de la información	-Protección de la información.	-Alteración de los datos. -Validación de las entradas -Confiability. -Importancia de los datos. -Control de accesos.
Base de Datos	-Almacenamiento	-Control de la información. -Confiability de la información. -Consistencia de la información.

Tabla 1 Operacionalización de las variables de la investigación

Para dar cumplimiento al objetivo trazado se plantean las siguientes **tareas**:

- 1- Revisar bibliografía referente al tema de las bases de datos.
- 2- Estudiar las herramientas que se utilizara para el diseño de la base de datos.
- 3- Investigar los distintos Sistemas Gestores de Bases de Datos (SGBD) para analizar sus características y así escoger el más óptimo para realizar este trabajo.

- 4- Confeccionar el modelo conceptual de la BD.
- 5- Confeccionar el modelo lógico de la BD.
- 6- Confeccionar diseño físico de la BD.
- 7- Realizar la validación teórica del diseño:
 - Normalización de la Base de datos.
 - Análisis de redundancia de información
- 8- Analizar y valorar los resultados obtenidos según el diseño de la BD.

Métodos científicos de investigación

Métodos teóricos:

* Histórico-Lógico:

Para la realización de esta investigación se hizo un análisis en profundidad de la trayectoria histórico lógica del modo de almacenamiento de la información y el efecto que ha traído consigo en el trabajo de la DCI.

* Análisis-Síntesis:

Se realizó un análisis de los temas referentes a las bases de datos y los procesos de almacenamiento de información en la DCI desde su aspecto más simple para después sintetizar el resultado.

* Inductivo-Deductivo:

Con este método se logró inferir conocimientos, permitiendo el tránsito de lo particular a lo general y también de aseveraciones generales a otras particulares mediante pasos que no tienen contradicciones lógicas en lo referente a los procesos de almacenamiento de información en la DCI.

* Modelación:

Se aplicó la modelación para la construcción de diagramas, logrando un mejor entendimiento del negocio, ya que el diseño de BD requiere modelar diversos aspectos relacionados con la información que se quiere almacenar.

* Genético:

Permitió establecer cierto campo de acción que convirtió en célula a los procesos de almacenamiento de información de la DCI y que contenía todas sus características esenciales y hacía imposible su división. Para su aplicación se combina el análisis histórico lógico.

* Dialéctico:

Se utilizó para descubrir las contradicciones que dan lugar al auto movimiento de los procesos de almacenamiento de información en la DCI, una vez determinada su estructura y dinámica para provocar los cambios cuantitativos en cualitativos, para dar lugar a un nuevo objeto.

Métodos empíricos:

* Observación:

Permite el estudio de los procesos de almacenamiento de información mediante la percepción y registro planificado y sistemático de su comportamiento en la DCI.

Análisis financiero

El proyecto cuenta con 30 estudiantes y tres profesores líderes del mismo a los que se les paga mensualmente un estipendio y salario respectivamente.

El estipendio de los estudiantes debe verse por separado porque en el proyecto hay estudiantes de tres años distintos 3ro, 4to y 5to, el de primero y tercero es de \$50 (3 estudiantes de 3ro y 1 de 1ro), el de cuarto es de \$75 (12 estudiantes) y el de 5to es de \$100 (14 estudiantes). El salario de los profesores es de 349 (el básico) más un adecuado de 402 si salen bien en la evaluación mensual que se les hace, por lo que se necesitará un respaldo económico de \$65400 mensuales para pagarles tanto a los estudiantes como a los profesores, por aproximadamente 3 años (duración aproximada del proyecto).

Además se usan 8 computadoras para el desarrollo de la investigación y la futura obtención del producto. Dichas computadoras son material de la UCI en las cuales se analizó su depreciación por el uso y el gasto de energía eléctrica. Además se analizó el uso de materiales gastables de oficina.

Realizando un estudio de cada factor se puede observar que el costo general del proyecto es de \$70 600 aproximadamente.

Cronograma para la ejecución de la investigación

No.	Acciones a realizar	Responsable	Participantes	Fecha	
				Inicio	Fin

1	Revisar bibliografía referente al tema de las BD	Arlen Tamayo Adennys Almaguer		Octubre	Enero
2	Estudiar las herramientas que se utilizara para el diseño de la BD	Arlen Tamayo Adennys Almaguer		Octubre	Enero
3	Investigar los distintos SGBD para escoger el más óptimo.	Arlen Tamayo Adennys Almaguer		Enero	Febrero
4	Confeccionar el modelo conceptual de la BD.	Arlen Tamayo Adennys Almaguer	Julio Isaza	Enero	Abril
5	Confeccionar el modelo lógico de la BD.	Arlen Tamayo Adennys Almaguer	Julio Isaza	Enero	Abril
6	Confeccionar diseño físico de la BD.	Arlen Tamayo Adennys Almaguer		Abril	Mayo
7	Realizar la validación teórica del diseño: - Normalización de la Base de datos. - Análisis de redundancia de información	Arlen Tamayo Adennys Almaguer		Abril	Mayo
8	Analizar y valorar los resultados obtenidos según el diseño de la BD.	Arlen Tamayo Adennys Almaguer		Abril	Mayo

Tabla 2 Cronograma de trabajo de la investigación

Aportes prácticos del trabajo:

El aporte práctico de esta investigación es el diseño de una base de datos, para almacenar la información que se utiliza o genera en la DCI, dirigiendo este diseño a satisfacer las necesidades de almacenamiento de dicha dirección, además de tener en cuenta la disponibilidad, seguridad e integridad de la información.

Esto reducirá en gran medida el empleo de recursos gastables y el tiempo de ejecución de las tareas y actividades que se llevan a cabo en la DCI, además contribuirá a lograr una mayor organización de la información, permitiendo un rápido acceso a ella.

Este trabajo favorecerá también a la informatización de la UCI.

El presente documento está compuesto por 3 capítulos:

Capítulo 1:

En el capítulo 1 se trata la fundamentación teórica de la investigación, incluyendo un estado del arte referente al tema de las bases de datos a varios niveles (Internacional, Nacional y de la Universidad). Además se aborda las tendencias actuales sobre el tema y se hace un análisis de las distintas herramientas, metodologías y técnicas que son necesarias para el desarrollo de una base de datos.

Capítulo 2:

En el capítulo 2 se hace la descripción y análisis de la solución propuesta, en la que se exponen aspectos relacionados con el diseño de la base de datos, además de la descripción de las entidades involucradas. Además se hace un análisis de la arquitectura y la estrategia de integración con otros módulos o sistemas. También se analiza el tema de optimización de consultas.

Capítulo 3:

En el capítulo 3 se aborda la validación del diseño realizado, tanto teórica como funcional. Donde se analizan temas como la integridad de los datos, la normalización y redundancia de la información. También se realiza una valoración de los resultados obtenidos con la investigación. En las páginas finales de la tesis encontrarán las secciones de las conclusiones, bibliografía, anexos y glosario de términos.

CAPÍTULO1. FUNDAMENTACIÓN TEÓRICA.

1.1 Introducción

Las bases de datos se han constituido como unas de las herramientas más ampliamente difundidas en la actual sociedad de la información, utilizadas como fuentes secundarias en cuanto a recuperación y almacenamiento de información en todos los campos ha nivel científico, social, económico, político y cultural.

En los últimos años, el software de bases de datos ha experimentado un auge extraordinario, a raíz de la progresiva informatización de casi la totalidad de las empresas.

1.2 Surgimiento histórico de las bases de datos

Rapidez, efectividad en los procesos y los grandes flujos de información están como primera necesidad a la hora de optimizar servicios y productos. Ante esta notable demanda de soluciones informáticas han surgido multitud de gestores de bases de datos, siendo estos los programas que permiten manejar la información de modo sencillo y que prestan servicios para el desarrollo y el manejo de bases de datos.

Al estudiar el desarrollo del procesamiento automatizado de datos y desde el punto de vista del aseguramiento matemático y en particular, del aseguramiento de programas, algunos autores reconocen tres generaciones que están caracterizadas por los siguientes puntos:

- Solución de tareas aisladas.
- Integración de tareas aisladas en sistemas particulares.
- Integración de sistemas particulares en sistemas automatizados de dirección.

Este proceso de integración ocurre paralelamente, aunque no simultáneamente, en dos esferas:

- Integración de los programas.

Ha estado facilitada por el uso de lenguajes de programación cada vez más sofisticados y de redactores que permiten el acoplamiento de módulos escritos en lenguajes diferentes.

- Integración de los datos.

En la integración de los datos se han producido tres categorías de técnicas para su manipulación:

1. Sistemas orientados a los dispositivos.

Programas y archivos son diseñados y empleados de acuerdo con las características físicas de la unidad central y los periféricos. Cada programa está altamente interconectado con sus archivos, por lo que la integración de datos de diferentes sistemas es prácticamente imposible.

2. Sistemas orientados a los archivos.

La lógica de los programas depende de las técnicas de organización de los archivos (secuencial, directo). Cada usuario organiza su archivo de acuerdo con sus necesidades y las relaciones entre los elementos se establecen a través de los programas de aplicación.

Esta forma de trabajo implica redundancia de datos que trae aparejada mayor gasto de memoria y complica las operaciones de actualización (modificar un dato donde quiera que aparezca). Esto aumenta el tiempo de tratamiento y atenta contra la integridad de la información.

3. Sistemas orientados a BD

En estos sistemas se separa sutilmente la lógica de la aplicación de los datos que componen la información. Presentan varias ventajas como son:

- Abstracción de datos.
- Independencia física y lógica entre aplicaciones y datos.
- Múltiples vistas: Cada usuario puede estar interesado sólo en una porción o perspectiva particular de la BD.
- Información actualizada: ya que cuando alguna información es guardada cualquier usuario puede acceder a ella inmediatamente.

Las primeras bases de datos que surgieron en el mundo respondían al modelo jerárquico y luego al modelo reticular. A medida que aumentó el uso de las bases de datos surgió el modelo relacional el cual ha sido adoptado ampliamente en todo el mundo, pero este modelo no es el último, es decir, el desarrollo ha continuado y han surgido nuevos modelos como el modelo orientado a objetos. (GARCÍA, ROSA MARÍA MATO)

1.3 Actualidad de las bases de datos

1.3.1 Actualidad de las bases de datos en el mundo

Desde los años setenta, los sistemas de bases de datos han ido reemplazando a los sistemas de ficheros en los sistemas de información de las empresas. Al mismo tiempo, se ha ido reconociendo la gran importancia que tienen los datos que estas manejan, convirtiéndose en uno de sus recursos más importantes.

Debido al auge de la utilización de las BD han surgido nuevas aplicaciones para estas como son:

- Bases de datos de multimedia: con dibujos, videos, sonidos.
- Bases de datos orientadas a objetos.
- Bases de datos distribuidas.
- Sistemas de información geográfica: con mapas, datos del tiempo, imágenes de satélite.
- Almacenes de datos (o Data Warehouses) y Sistemas de Proceso Analítico On-Line (OLAP): analizan y extraen información útil para la toma de decisiones.
- Base de datos activas y de tiempo real: útiles en procesos de control industrial y fabricación.

Casos de estudio:

Greenplum es uno de los proveedores principales de inteligencia comercial y almacén de datos. Su producto insignia, la BD de Greenplum, es un sistema de base de datos paralelo innovador basado en PostgreSQL que entrega a la industria escalabilidad y manejabilidad para apoyar las demandas de toma de decisiones en ambientes grandes. Esta BD impulsa el aparato de almacén de datos de la empresa Sun. Diseñada para apoyar la toma de decisiones este sistema funciona más rápidamente que las tecnologías de almacén de datos tradicionales. (GROUP 2008)

Skype usa PostgreSQL como el SGBD principal para la mayoría de sus de necesidades comercial de salida y han tenido bastante éxito creciendo como compañía en cuanto a: el número de usuarios a los que posee y en la complejidad de servicios que ofrece.(ZONE 2008)

La aplicación de técnicas de búsqueda en BD a la web está mejorando la eficiencia de las búsquedas en este entorno.(ELMASRI/NAVATHE 2002)

1.3.2 Actualidad de las bases de datos en Cuba

A partir de la utilización ordenada y masiva de las tecnologías de la información y las comunicaciones en todas las esferas de la sociedad cubana, se ha incrementado el uso de las bases de datos como la alternativa más adecuada para gestionar los datos de las empresas e instituciones cubanas.

El criterio principal que se utiliza para clasificar los SGBD es el modelo lógico en que se basan. Los modelos lógicos empleados con mayor frecuencia en los SGBD actuales son el relacional, el de red y el jerárquico, aunque algunos más modernos se basan en modelos orientados a objetos. Dentro de los SGBD relacional más usados específicamente en las instituciones cubanas se encuentran: MySQL y Access.

A pesar del férreo bloqueo económico y social que sufre la isla, el desarrollo de la informática en general y de las tecnologías de bases de datos en particular, el estudio detallado de algunos sitios cubanos destaca el uso del gestor de bases de datos Access que viene con el paquete de Office de Windows como herramienta para la elaboración de las bases de datos de las instituciones cubanas, por su facilidad de uso y la poca complejidad que presenta. Esta tendencia se observa sobre todo en sitios con bases de datos de tamaño reducido, ejemplo de esto son:

La base de datos para el control de la documentación del Banco de Germoplasma de Arroz de Cuba que cuenta con un diseño relacional para el control de la documentación del Banco de Germoplasma de Arroz en Cuba. Fue realizada utilizando Microsoft Access versión 8.0 Copyright 1991–1996.

Además existen otros casos como el Instituto Nacional de Recursos Hidráulicos (INRH) que cuenta con una base de datos elaborada en Microsoft Access para recopilar los datos hidrológicos y de la calidad del agua. También podemos observar casos como el Acuario Nacional de Cuba (ANC), el Centro de Inspección y Control Ambiental (CICA) y el Jardín Botánico Nacional entre muchos otros que usan este gestor de bases de datos.(DÍAZ 2008)

Sin embargo como consecuencia del antes mencionado bloqueo económico se ha visto una progresiva migración de la informática cubana hacia el software libre. En el ámbito de las bases de datos se presenta un uso ya habitual de gestores de bases de datos como MySQL el cual presenta licencia de código abierto y está a disposición del público de forma gratis. La investigación en sitios cubanos arroja que este gestor de bases de datos es usado en:

Sitio Web para la aplicación de un Sistema de Gestión Tecnológica Hospitalaria:

Este sitio Web pasará a llenar el vacío que existe actualmente en la red de salud cubana y de América Latina relacionado con el tema de la Gestión Tecnológica. Este producto propuesto por el Centro de Bioingeniería (CEBIO) del Instituto Superior Politécnico “José Antonio Echeverría” (ISPJAE) usa como lenguaje para manipular los datos en páginas dinámicas PHP por sus cualidades de lenguaje multiplataforma con código sencillo basado en Perl y C y que constituye un excelente manipulador de ficheros. Esta elección unido a la de MySQL como gestor de bases de datos constituye una combinación efectiva, veloz y robusta empleada en la actualidad, sobretodo para servidores Linux y Unix, considerados uno de los mejores hoy.

Centro Provincial de Información de Ciencias Médicas Las Tunas:

Con el principal objetivo de dotar a la Biblioteca Médica Provincial en Las Tunas de un espacio Web que mostrará los servicios, acontecer provincial, nacional e internacional en materia de información

médica; además de posibilitar el incentivo por parte de los usuarios a la colaboración, fue confeccionado Biblioweb.

En la elaboración de la plataforma Web, siempre se pensó en la utilización de la Arquitectura Cliente – Servidor por la amplia gama de posibilidades que brinda. La infraestructura de software utilizada fue: el paquete Wamp5 que dispone de un servidor Apache, MySQL como sistema gestor de bases de datos y PHP como lenguaje de programación.

Centro Nacional de Áreas Protegidas (CNAP):

Mediante el Sistema de Información Nacional para la Gestión de las Áreas Protegidas (SIGAP) gestiona el manejo, control y seguimiento de la información relacionada con la infraestructura y biodiversidad existente en las áreas protegidas de Cuba. Este sistema cuenta con una base de datos alfanumérica y espacial confeccionada en MySQL, la cual posee 30 000 registros, soporta georeferenciación local e individual de especies, tiene una estructura cliente-servidor y sobre todo se basa en una licencia GPL.

1.3.3 Actualidad de las bases de datos en la UCI

La UCI representa el prototipo de informatización de la sociedad cubana y como parte del proceso de informatización que se esta llevando a cabo ha alcanzado un alto nivel de desarrollo en cuanto a tecnologías de la información y las comunicaciones. En la UCI como parte de su estrategia de producción están conformados varios proyectos productivos que se encargan de automatizar la universidad y de producir software para beneficio del país.

En sincronización con las tendencias actuales existe un amplio uso de las bases de datos como herramienta ideal de almacenamiento de la información, tal es así que como parte de la formación académica que se les da a los estudiantes en la UCI se imparte el aprendizaje de Microsoft SQLServer en la asignatura de bases de datos para fomentar el aprendizaje de esta tecnologías novedosas. No obstante la tendencia actual en la producción es la utilización de herramientas de software libre de acuerdo a las limitaciones presentes, por lo tanto se presenta una utilización general de gestores de bases de datos libres como PostgreSQL, MySQL o InterBase 6.1. Muchos aseguran que PostgreSQL es la solución pero los criterios aparecen divididos en la mayoría de los casos en cuanto a la utilización de MySQL o PostgreSQL en la elaboración de las bases de datos.

También se realizó una investigación acerca del diseño de bases de datos en la UCI, esta arrojó lo siguiente:

En el proyecto Correos de Cuba de la facultad 1 existe una base de datos para la gestión de las operaciones en una oficina multiservicios, utiliza como SGBD PostgreSQL y como lenguaje de programación PHP.

EL Proyecto Plataforma de Cómputo Distribuida de la facultad 6 usa como SGBD MySQL 5.0 y lenguaje de programación Java.

El Proyecto Kainos de la facultad 1 tiene una base de datos relacional, usa PostgreSQL como SGBD y php y java como lenguajes de alto nivel de programación. Este proyecto utiliza herramientas para gestionar el acceso a datos como Drupal en el caso de php e Hibernate para java.

El Proyecto Identidad de la facultad 1 utiliza como SGBD Oracle 9i. Y para programar en la BD PLSQL. Los diseños de estas bases de datos de cada uno de los proyectos analizados están dirigidos a los requerimientos de cada uno de sus problemas específicos. El proyecto SICI está dirigido a dar cumplimiento a las necesidades de la DCI, en una primera etapa se realizó una BD en software propietario específicamente en Microsoft SQL Server la cual no cumple con la necesidades reales. Por esta razón se necesita de un sistema de almacenamiento mejor, que represente realmente el negocio y que esté confeccionado con herramientas potentes y libres. Para llevar a cabo esta tarea se utilizará como gestor de bases de datos PostgreSQL, la programación de la aplicación se hará en PHP, lenguaje que presenta una adecuada integración con el SGBD antes mencionado.

1.4 Conceptos de base de datos

Una base de datos es un conjunto de datos interrelacionados entre sí, almacenados con carácter más o menos permanente en la computadora. O sea, que puede considerarse una colección de datos variables en el tiempo.(GARCÍA, LIC. ROSA MARÍA MATO 1999)

Una base de datos es un conjunto de datos que tiene las siguientes propiedades implícitas. (DATE 2003):

1. Representa algún aspecto del mundo real, llamado mini mundo o universo de discurso. Las modificaciones del mini mundo se reflejan en la BD.
2. Es un conjunto de datos lógicamente coherentes, con un cierto significado inherente. Una colección aleatoria de datos no puede considerarse propiamente una BD.

Una BD se diseña, construye y puebla con datos para propósito específico. Está dirigida a un grupo de usuarios y tiene ciertas aplicaciones preconcebidas que interesan a distintos usuarios. O sea, una BD tiene (DATE 2003):

1. Una fuente de la cual se derivan los datos.

2. Cierta grado de interacción con los hechos del mundo real.
3. Un público activamente interesado en el contenido de la BD.
4. Su tamaño es variado.
5. Debe ser posible buscar, obtener y actualizar los datos siempre que sea necesario.

Base de datos: colección de datos relacionados entre sí.

Datos: hechos conocidos que pueden registrarse y que tienen un significado implícito (nombres, teléfonos, direcciones).(ELMASRI 2002).

Según Henry F. Korth autor del libro "Fundamentos de Bases de Datos" se define una base de datos como una serie de datos organizados y relacionados entre sí, y un conjunto de programas que permitan a los usuarios acceder y modificar esos datos.

1.5 Metodologías

1.5.1 Metodología de diseño de bases de datos

El diseño de una base de datos es un proceso complejo que abarca decisiones a muy distintos niveles. La complejidad se controla mejor si se descompone el problema en subproblemas y se resuelve cada uno de estos subproblemas independientemente, utilizando técnicas específicas. Así, el diseño de una base de datos se descompone en diseño conceptual, diseño lógico y diseño físico. (C. BATINI 1994)

El diseño conceptual parte de las especificaciones de requisitos de usuario y su resultado es el esquema conceptual de la base de datos. Un esquema conceptual es una descripción de alto nivel de la estructura de la base de datos, independientemente del SGBD que se vaya a utilizar para manipularla. Un modelo conceptual es un lenguaje que se utiliza para describir esquemas conceptuales. El objetivo del diseño conceptual es describir el contenido de información de la base de datos y no las estructuras de almacenamiento que se necesitarán para manejar esta información. (C. BATINI 1994)

El objetivo es comprender:

- La perspectiva que cada usuario tiene de los datos.
- La naturaleza de los datos, independientemente de su representación física.
- El uso de los datos a través de las áreas de aplicación.

El diseño conceptual se puede utilizar para que el diseñador transmita lo que ha entendido sobre la información que se maneja. Para ello, debe estar familiarizado con la notación utilizada en este esquema. La más popular es la notación del modelo entidad–relación. El diseño conceptual es una fuente de información para el diseño lógico de la base de datos.

El diseño lógico es el proceso de construir un esquema de la información ha utilizar, basándose en un modelo de base de datos específico, independiente del SGBD concreto que se vaya a utilizar y de cualquier otra consideración física.

En esta etapa, se transforma el esquema conceptual en un esquema lógico que utilizará las estructuras de datos del modelo de base de datos en el que se basa el SGBD que se vaya a utilizar, como puede ser el modelo relacional, el modelo de red, el modelo jerárquico o el modelo orientado a objetos. Conforme se va desarrollando el esquema lógico, este se va probando y validando con los requisitos de usuario.

La normalización es una técnica que se utiliza para comprobar la validez de los esquemas lógicos basados en el modelo relacional, ya que asegura que las relaciones (tablas) obtenidas no tienen datos redundantes.

El esquema lógico es una fuente de información para el diseño físico. Además, juega un papel importante durante la etapa de mantenimiento del sistema, ya que permite que los futuros cambios que se realicen sobre los programas de aplicación o sobre los datos, se representen correctamente en la base de datos.

El diseño físico parte del esquema lógico y da como resultado un esquema físico. Un esquema físico es una descripción de la implementación de una base de datos en memoria secundaria: las estructuras de almacenamiento y los métodos utilizados para tener un acceso eficiente a los datos. Por ello, el diseño físico depende del SGBD concreto y el esquema físico se expresa mediante su lenguaje de definición de datos.(C. BATINI 1994)

Los objetivos del diseño de la base de datos son:

- Representar los datos que requieren las principales áreas de aplicación y los grupos de usuarios, y representar las relaciones entre dichos datos.
- Proporcionar un modelo de datos que soporte las transacciones que se vayan a realizar sobre los datos.
- Especificar un esquema que alcance las prestaciones requeridas para el sistema.

1.5.2 Metodologías de desarrollo y lenguaje de modelado

1.5.2.1 Rational Unified Process (RUP)

RUP es un proceso para el desarrollo de un proyecto de un software que define claramente quien, cómo, cuándo y qué debe hacerse en el proyecto. Sus tres características esenciales son que está dirigido por los casos de uso: orientan el proyecto a la importancia para el usuario y lo que este quiere,

está centrado en la arquitectura: relaciona la toma de decisiones que indican cómo tiene que ser construido el sistema y en qué orden, y como última característica esencial RUP es iterativo e incremental: divide el proyecto en mini proyectos donde los casos de uso y la arquitectura cumplen sus objetivos de manera más depurada.

En RUP se han agrupado las actividades en grupos lógicos definiéndose nueve flujos de trabajo principales. Los seis primeros son conocidos como flujos de ingeniería y los tres últimos como de apoyo.

Flujos de trabajo:

- Modelamiento del negocio
- Requerimientos
- Análisis y diseño
- Implementación
- Prueba
- Administración del proyecto
- Administración de configuración y cambios
- Ambiente

Fases:

- Conceptualización (Concepción o Inicio)
- Elaboración
- Construcción
- Transición

1.5.2.2 Unified Modeling Language (UML):

UML es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema de software orientado a objetos. Se ha convertido en el estándar de facto de la industria, debido a que ha sido impulsado por los autores de los tres métodos más usados de orientación a objetos: Grady Booch, Ivar Jacobson y Jim Rumbaugh.

Estos autores fueron contratados por la empresa Rational Software Co. para crear una notación unificada en la que basar la construcción de sus herramientas CASE. En el proceso de creación de UML han participado, no obstante, otras empresas de gran peso en la industria como Microsoft, Hewlett-Packard, Oracle o IBM, así como grupos de analistas y desarrolladores.

Esta notación ha sido ampliamente aceptada debido al prestigio de sus creadores y debido a que incorpora las principales ventajas de cada uno de los métodos particulares en los que se basa (principalmente Booch, OMT y OOSE). UML ha puesto fin a las llamadas “guerras de métodos” que se han mantenido a lo largo de los 90, en las que los principales métodos sacaban nuevas versiones que incorporaban las técnicas de los demás. Con UML se fusiona la notación de estas técnicas para formar una herramienta compartida entre todos los ingenieros software que trabajan en el desarrollo orientado a objetos. Uno de los objetivos principales de la creación de UML era posibilitar el intercambio de modelos entre las distintas herramientas CASE orientadas a objetos del mercado.(CLIKER.COM 2004)

Es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software. UML es un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos.(BOOCH 2000.)

1.6 Conceptos de sistema gestor de base de datos

El software que permite la utilización y/o la actualización de los datos almacenados en una (o varias) base(s) de datos por uno o varios usuarios desde diferentes puntos de vista y a la vez, se denomina sistema de gestión de bases de datos (SGBD). (GARCÍA, LIC. ROSA MARÍA MATO 1999)

Un Sistema Gestor o Sistema Manejador de Bases de Datos (SGBD o DBMS por sus siglas en inglés) es un conjunto de programas que permite a los usuarios crear y mantener una BD, por lo tanto, el SGBD es un software de propósito general que facilita el proceso de definir, construir y manipular la BD para diversas aplicaciones. Pueden ser de propósito general o específico. (DATE 2003)

Un DBMS es una colección de numerosas rutinas de software interrelacionadas, cada una de las cuales es responsable de una tarea específica.(MARQUÉS 2004)

SGBD: Conjunto de programas que permite a los usuarios crear y mantener una BD.

– De propósito general

– Permite definir, construir, manipular datos para aplicaciones diversas

Sistema de BD: conjunto formado por la BD y el software.(ELMASRI 2002)

1.7 Arquitectura de un SGBD

1.7.1 Arquitectura de las bases de datos relacionales

La arquitectura relacional se puede expresar en términos de tres niveles de abstracción: nivel interno, conceptual y de visión o externo. El objetivo de esta arquitectura es el de separar los programas de aplicación de la base de datos física. En esta arquitectura, el esquema de una base de datos se define en tres niveles de abstracción distintos:

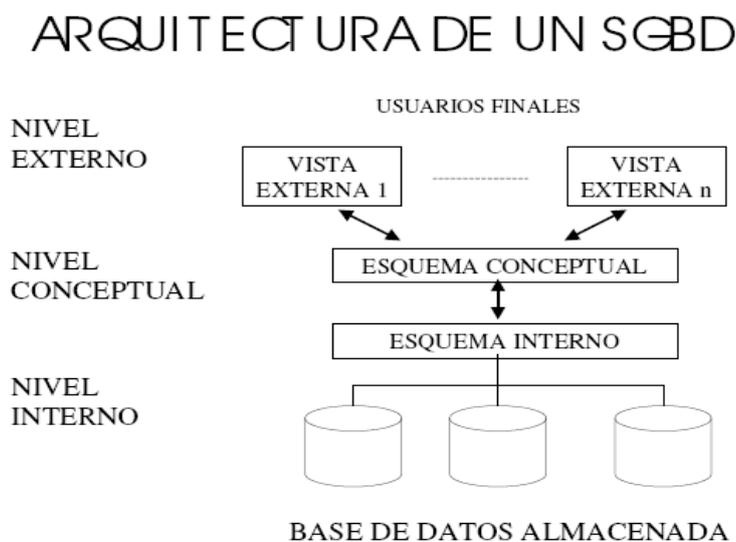


Figura 1. 1 Arquitectura de un sistema gestor de bases de datos

1. En el nivel interno se describe la estructura física de la base de datos mediante un esquema interno. Este esquema se especifica mediante un modelo físico y describe todos los detalles para el almacenamiento de la base de datos, así como los métodos de acceso.
2. En el nivel conceptual se describe la estructura de toda la base de datos para una comunidad de usuarios (todos los de una empresa u organización), mediante un esquema conceptual. Este esquema oculta los detalles de las estructuras de almacenamiento y se concentra en describir entidades, atributos, relaciones, operaciones de los usuarios y restricciones. En este nivel se puede utilizar un modelo conceptual o un modelo lógico para especificar el esquema.
3. En el nivel externo se describen varios esquemas externos o vistas de usuario. Cada esquema externo describe la parte de la base de datos que interesa a un grupo de usuarios determinados y ocultos a ese grupo el resto de la base de datos. En este nivel se puede utilizar un modelo conceptual o un modelo lógico para especificar los esquemas.

La arquitectura de tres niveles es útil para explicar el concepto de independencia de datos que podemos definir como la capacidad para modificar el esquema en un nivel del sistema sin tener que modificar el esquema del nivel inmediato superior. Se pueden definir dos tipos de independencia de datos:

- La independencia lógica es la capacidad de modificar el esquema conceptual sin tener que alterar los esquemas externos ni los programas de aplicación. Se puede modificar el esquema conceptual para ampliar la base de datos o para reducirla. Si, por ejemplo, se reduce la base de datos eliminando una entidad, los esquemas externos que no se refieran a ella no deberán verse afectados.
- La independencia física es la capacidad de modificar el esquema interno sin tener que alterar el esquema conceptual (o los externos). Por ejemplo, puede ser necesario reorganizar ciertos ficheros físicos con el fin de mejorar el rendimiento de las operaciones de consulta o de actualización de datos. Dado que la independencia física se refiere sólo a la separación entre las aplicaciones y las estructuras físicas de almacenamiento, es más fácil de conseguir que la independencia lógica.

1.7.2 Componentes de la arquitectura de las bases de datos relacionales

La arquitectura relacional consta de los siguientes componentes:

- Modelo relacional de datos:

En el nivel conceptual, el modelo relacional de datos está representado por una colección de relaciones almacenadas. Cada registro de tipo conceptual en un modelo relacional de datos se implanta como un archivo almacenado distinto.

- Submodelo de datos:

Los esquemas externos de un sistema relacional se llaman submodelos relacionales de datos; cada uno consta de uno a más escenarios (vistas) para describir los datos requeridos por una aplicación dada. Un escenario puede incluir datos de una o más tablas de datos. Cada programa de aplicación está provisto de un buffer ("Área de trabajo de usuario") donde el DBMS puede depositar los datos recuperados de la base para su procesamiento, o puede guardar temporalmente sus salidas antes de escribirlas en la base de datos.

- Esquema de almacenamiento:

En el nivel interno, cada tabla base se implanta como un archivo almacenado. Para las recuperaciones sobre las claves principal o secundaria se pueden establecer uno o más índices para acceder a un archivo almacenado.

- Sublenguaje de datos:

Es un lenguaje de manejo de datos para el sistema relacional, el álgebra relacional y cálculo relacional, ambos lenguajes son "relacionalmente completos", esto es, cualquier relación que pueda derivarse de una o más tablas de datos, también se puede derivar con un solo comando del sublenguaje. Por tanto, el modo de operación de entrada/salida en un sistema relacional se puede procesar en la forma: una tabla a la vez en lugar de: un registro a la vez; en otras palabras, se puede recuperar una tabla en vez de un solo registro con la ejecución de un comando del sublenguaje de datos.

1.7.3 Arquitectura del SGBD PostgreSQL

La arquitectura cliente/servidor separa las funciones de una aplicación en componentes que establecen diálogos entre sí para intercambiar información, servicios o recursos con el objeto de realizar una tarea común. Cada componente puede estar en un ordenador diferente. El proceso que inicia el diálogo o solicita recursos se denomina cliente y suele ser la aplicación que el usuario está ejecutando. El proceso que responde a las solicitudes se denomina servidor.

Esta arquitectura se basa, al igual que el caso anterior, en varias plataformas interconectadas: una de las cuales actúa como "servidor" de la BD en la que los datos están físicamente localizados y centraliza las funciones de administración. Las plataformas denominadas "clientes" realizan funciones de manejo de los interfaces de usuario, lógica de aplicación, etc.

La arquitectura cliente/servidor no exige requisitos especialmente complejos a los SGBD ya que, aunque estén involucrados varios ordenadores, la base de datos en sí está normalmente centralizada en un ordenador y su mantenimiento es igual de sencillo que en una arquitectura centralizada clásica. Para esta arquitectura es importante que el SGBD soporte sistemas de comunicación normalizados ya que tendrá que recibir peticiones de diversos clientes operando con máquinas y protocolos distintos.

1.8 Características y objetivos del enfoque de los SGBD

Existen muchas formas de organizar las bases de datos, pero hay un conjunto de objetivos generales que deben cumplir todas los SGBD, de modo que faciliten el proceso de diseño de aplicaciones y que los tratamientos sean más eficientes y rápidos, dando la mayor flexibilidad posible a los usuarios. Los objetivos fundamentales de los SGBD son:

- Independencia de los datos y los programas de aplicación:

Independencia de los datos:

Inmunidad de las aplicaciones a los cambios en la estructura de almacenamiento y en la estrategia de acceso y constituye el objetivo fundamental de los SGBD, o sea, es la capacidad de modificar el esquema en un nivel del sistema de BD sin tener que modificar el nivel inmediato superior. (DATE 2003)

Independencia lógica con respecto a los datos:

Es la capacidad de modificar el esquema conceptual sin tener que alterar los esquemas externos, ni los programas de aplicación. Podemos modificar el esquema conceptual para ampliar la BD o para reducirla. Las modificaciones no deberán afectar los esquemas externos que sólo se refieren a los datos restantes. Además, las restricciones podrán ser modificadas en el esquema conceptual sin afectar los esquemas externos.

Independencia física respecto a los datos:

Es la capacidad de modificar el esquema interno sin tener que alterar el esquema conceptual o los externos. Se refiere sólo a la separación entre las aplicaciones y la estructura física de almacenamiento, es más fácil de lograr que la independencia lógica de los datos.

En un SGBD sería indeseable la existencia de aplicaciones y datos dependientes entre sí, por dos razones fundamentales:

1. Diferentes aplicaciones necesitarán diferentes aspectos de los mismos datos.
2. Se debe poder modificar la estructura de almacenamiento o el método de acceso según los cambios en el fenómeno o proceso de la realidad sin necesidad de modificar los programas de aplicación para buscar mayor eficiencia.

- Minimización de la redundancia:

Con los ficheros tradicionales se produce redundancia de la información. Uno de los objetivos de los SGBD es minimizar la redundancia de los datos. Se dice disminuir la redundancia, no eliminarla, pues aunque se definen las BD como no redundantes, en realidad existe redundancia en un grado no significativo para disminuir el tiempo de acceso a los datos o para simplificar el método de direccionado. Lo que se trata de lograr es la eliminación de la redundancia superflua.

- Integridad de los datos:

Consiste en garantizar la no contradicción entre los datos almacenados de modo que, en cualquier momento los datos almacenados sean correctos, es decir, que no se detecte inconsistencia entre los datos. Está relacionada con la minimización de redundancia, ya que es más fácil garantizar la integridad si se elimina la redundancia, entre otros aspectos.

Seguridad y protección de los datos:

Protección: garantizar el acceso autorizado a los datos, de forma de interrumpir cualquier intento de acceso no autorizado, ya sea por error del usuario o por mala intención.

Seguridad: que el sistema de bases de datos disponga de métodos que garanticen la restauración de las BD al producirse alguna falla técnica.

- **Facilidad de manipulación de la información:**

Los usuarios de una BD pueden referirse a ella con las solicitudes para resolver muchos problemas diferentes. El Sistema de Bases de datos (SBD) debe contar con la capacidad de una búsqueda rápida por diferentes criterios, permitir que los usuarios planteen sus demandas de una forma simple, aislándolo de las complejidades del tratamiento de los ficheros y del direccionado de los datos. Los SBD actuales brindan lenguajes de alto nivel con diferentes grados de facilidad para el usuario no programador que garantizan este objetivo.

- **Control centralizado:**

Uno de los objetivos más importantes de los SBD es garantizar el control centralizado de la información. Permite controlar de manera sistemática y única los datos que se almacenan en la BD, así como el acceso a ella. Lo anterior implica que debe existir una persona o conjunto de personas que tenga la responsabilidad de los datos operacionales: el administrador de la BD, que puede considerarse parte integrante del SBD.

1.9 Ventajas de utilizar un SGBD

- **Control de la redundancia:**

- Cada dato se almacena en un solo lugar para evitar inconsistencias.

- Alternativa: permitir duplicar datos controlándolos para evitar que se produzcan inconsistencias.

- **Restricción de accesos no autorizados a la BD:**

- Cuentas de acceso a la BD.

- Restricción a determinados datos.

- Restricción sobre las operaciones que se pueden realizar.

- Control sobre el software de creación de cuentas y concesión de permisos.

- Persistencia

- Es la posibilidad de recuperar el valor de una variable de un programa tras su ejecución (aunque contenga una estructura de datos compleja).

- Problema de incompatibilidad de impedancia: se da en SGBD tradicionales, cuando sus estructuras de datos son incompatibles con las del lenguaje de programación.

- Inferencias y acciones mediante reglas:

- BD deductivas: permiten definir reglas de deducción para deducir información nueva a partir de la almacenada.

- BD activas: permiten definir reglas que inician acciones automáticamente.

- Múltiples interfaces: como lenguajes de consulta, interfaces del lenguaje de programación, formularios, menús, lenguaje natural.

- Representación de relaciones complejas de datos mutuamente relacionados:

Que se obtengan y actualicen con rapidez.

- Restricciones de integridad: el SGBD permite definir las y es responsable de que se cumplan.

- Algunas restricciones pueden precisar su verificación en los programas de actualización de la BD.

- Subsistema de copias de seguridad (backup) y recuperación: si el sistema falla durante su ejecución este subsistema del SGBD restaura la BD al estado previo a la ejecución del programa (o reanuda la ejecución desde el punto en el que fue interrumpido).

1.10 Desventajas de utilizar un SGBD

- Coste de software y hardware: Además del coste de comprar el software, probablemente haya que ampliar o subir de versión el hardware para que el programa pueda almacenarse y funcione correctamente; además, si no, el sistema podría degradarse y los tiempos de respuesta subir drásticamente por la sobrecarga que añade el SGBD al tener que implementar la seguridad, integridad y datos compartidos.

- Coste de migración de aplicaciones: Un coste adicional, también, es la migración de las aplicaciones para pasar a un entorno integrado.

- Las operaciones de respaldo y recuperación de los datos en caso de fallo son complejas, debido al acceso concurrente de múltiples usuarios.
- Criticidad del SGBD: La centralización también significa que los datos de la organización, están almacenados en un único lugar, en la base de datos, con lo que cualquier fallo de seguridad u operación del SGBD puede afectar a toda la organización.

1.11 Clasificación de los SGBD

Según modelos conceptuales:

* Relacional:

Los modelos relacionales se han convertido en los modelos de base de datos de mayor demanda y su popularidad va en aumento. En el modelo relacional, cada fila de una tabla representa una entidad de datos, mientras que las columnas de la tabla representan los atributos.

Cada atributo solo puede tomar ciertos valores. Los valores permisibles para dichos atributos se denominan el dominio. El dominio para un atributo en particular indica cuales valores se pueden colocar en cada una de las columnas de la tabla relacional.

* Redes:

Un modelo de red es una ampliación del modelo jerárquico, sin embargo, en lugar de tener solo varios niveles de relaciones de uno a muchos, el modelo de red es una relación de propietario-miembro en la cual un miembro puede tener muchos propietarios.

En una base de datos estructurada como un modelo de red a menudo se puede tener acceso a un elemento de datos en particular a través de más de una ruta.

* Jerárquico:

En un modelo jerárquico de base de datos, los datos se organizan en una estructura descendente, o en forma de un árbol invertido. Consideramos un proyecto característico.

El modelo jerárquico es más conveniente para situaciones en donde las relaciones lógicas entre los datos se pueden presentar en forma apropiada con el enfoque de un padre-muchos hijo.

Si tiene acceso a los datos en forma lógica pasando a través de la "generaciones" apropiadas de padres para llegar al elemento de datos que se buscan, y solo hay una ruta de acceso a cualquier elemento particular de datos.

* Orientado a Objetos:

Las bases de datos orientadas a objetos (BDOO) almacenan y manipulan información que puede ser digitalizada por objetos, proporcionan una estructura flexible con acceso ágil, rápido, con gran capacidad de modificación. Las BDOO permiten el desarrollo y mantenimiento de aplicaciones complejas ya que se puede utilizar un mismo modelo conceptual y así aplicarlo al análisis, diseño y programación, esto reduce el problema entre los diferentes modelos a través de todo el ciclo de vida, con un costo significativamente menor.

Además combina las mejores cualidades de los archivos planos, las bases jerárquicas y relacionales. En las tendencias actuales de modelación de BD las BDOO representan el siguiente paso en la evolución de las bases de datos para soportar análisis, diseño y programación orientada a objetos.

* Otros

- Según el número de usuarios:
 - Monousuario.
 - Multiusuario.
- Según el número de sitios:
 - Centralizado.
 - Distribuido.
- Según el uso de Software:
 - Homogéneo.
 - Heterogéneo.
- En cuanto al propósito:
 - De propósito general.
 - De propósito específico (OLTP reservaciones de aerolíneas).

1.12 Herramientas analizadas

1.12.1 Sistemas gestores de bases de datos

PostgreSQL

Destaca por su amplísima lista de prestaciones que lo hacen capaz de competir con cualquier SGBD comercial:

- Está desarrollado en C, con herramientas como Yacc y Lex.

- La API de acceso al SGBD se encuentra disponible en C, C++, Java, Perl, PHP, Python y TCL, entre otros.
- Cuenta con un rico conjunto de tipos de datos, permitiendo además su extensión mediante tipos y operadores definidos y programados por el usuario.
- Su administración se basa en usuarios y privilegios.
- Sus opciones de conectividad abarcan TCP/IP, sockets Unix y sockets NT, además de soportar completamente ODBC.
- Los mensajes de error pueden estar en español y hacer ordenaciones correctas con palabras acentuadas o con la letra 'ñ'.
- Es altamente confiable en cuanto a estabilidad se refiere.
- Puede extenderse con librerías externas para soportar encriptación, búsquedas por similitud fonética (soundex).
- Control de concurrencia multi-versión, lo que mejora sensiblemente las operaciones de bloqueo y transacciones en sistemas multi-usuario.
- Implementación de algunas extensiones de orientación a objetos. En PostgreSQL es posible definir un nuevo tipo de tabla a partir de otra previamente definida.
- Soporte para vistas, claves foráneas, integridad referencial, disparadores, procedimientos almacenados, subconsultas y casi todos los tipos y operadores soportados en SQL92 y SQL99.

Postgres es completamente ACID en el mundo de las bases de datos se le llama ACID a la propiedad de un gestor de base de datos para realizar transacciones seguras. En concreto ACID es un acrónimo de Atomicity, Consistency, Isolation and Durability: Atomicidad, Consistencia, Aislamiento y Durabilidad en español.

- Atomicidad (Indivisible) 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.

MySQL

Es uno de los SGBD más populares, desarrollado bajo la filosofía de código abierto. La desarrolla y mantiene la empresa MySQLAB pero puede utilizarse gratuitamente y su código fuente está disponible.

Es un servidor de bases de datos multiusuario, la más rápida y robusta, de su categoría, tanto para volúmenes de datos grandes como pequeños en entornos Web. Tiene una implementación cliente-servidor que consiste en un demonio mysqld y varios programas clientes y librerías. Su principal objetivo de diseño fue la velocidad y se sacrificaron algunas características esenciales con este fin.

Otra característica importante es que consume muy pocos recursos.

Licencia GPL a partir de la versión 3.23.19.

Ventajas:

- Mayor rendimiento.
- Mayor velocidad tanto al conectar con el servidor como al servir selects y demás.
- Mejores utilidades de administración (backup, recuperación de errores, etc.).
- Aunque se cuelgue, no suele perder información ni corromper los datos.
- Mejor integración con PHP.
- No hay límites en el tamaño de los registros.
- Mejor control de acceso, en el sentido de que usuarios tienen acceso a que tablas y con que permisos.

Inconvenientes:

- No soporta transacciones, "roll-backs" ni subselects.
- No considera las claves ajenas.
- Ignora la integridad referencial, dejándola en manos del programador de la aplicación.

SQL Server

Es el SGBD de elección para una amplia gama de clientes corporativos y proveedores independientes de software que construyen aplicaciones de negocios.

Es una aplicación poderosa, robusta, que permite gran seguridad de los datos, ostenta marcas de referencia en cuanto a escalabilidad y confiabilidad, que son críticas para el éxito de bases de datos de gran tamaño. Permite además lograr una gran velocidad en el procesamiento de transacciones, y agilidad en todas sus operaciones.

Las necesidades y requerimientos de los clientes han llevado a la creación de innovaciones de producto significativas para facilitar la utilización, escalabilidad, confiabilidad y almacenamiento de datos. La estrategia de Microsoft es la de hacer que SQL Server sea la base de datos más fácil de utilizar para construir, administrar e implementar aplicaciones de negocios.

SQL Server disminuye el costo total de propiedad a través de características como administración multi-servidor y con una sola consola; ejecución y alerta de trabajos basadas en eventos; seguridad integrada; y scripting administrativo. Libera al administrador de base de datos para aspectos más sofisticados del trabajo al automatizar las tareas de rutina. Al combinar estos poderosos servicios de administración con las nuevas características de configuración automática, es la elección ideal de automatización de sucursales y aplicaciones de base de datos insertadas.

Ofrece un motor de base datos único que escala desde una computadora portátil que ejecuta Windows® 95 o Windows 98, hasta clústeres de procesadores múltiples simétricos de terabyte que ejecutan Windows NT Server Enterprise Edition. Todos estos sistemas mantienen la seguridad y confiabilidad que exigen los sistemas de negocios de misión crítica.

Las otras características tales como bloqueo a nivel de línea dinámico, el paralelismo interno de las consultas y mejoras para las bases de datos muy grandes (VLDB) hacen que sea la elección ideal para sistemas OLTP de alta tecnología y sistemas de data warehousing.

1.12.2 Herramientas de modelado

Visual Paradigm

Es una herramienta CASE que utiliza “UML” como lenguaje de modelaje. Facilita las organizaciones visuales, diseña, integra y despliega su misión en las aplicaciones y sus bases de datos subyacentes y permite crear tipos diferentes de diagramas en un ambiente totalmente visual. Esta herramienta es un producto de calidad, soporta aplicaciones Web, es muy fácil de instalar y actualizar, genera código para varios lenguajes, pero se debe señalar que una de sus desventajas es que las imágenes y reportes generados no son de muy buena calidad. El Visual Paradigm ofrece además:

- Un entorno de creación de diagramas para UML 2.0.
- Diseño centrado en casos de uso y enfocado al negocio generando un software de mayor calidad.
- Uso de un lenguaje común para todo el equipo de desarrollo facilitando la comunicación.
- Capacidades de ingeniería directa e inversa.

- Disponibilidad de múltiples versiones, para cada necesidad.
- Disponibilidad en múltiples plataformas

Rational Rose

Es una herramienta CASE que proporciona mecanismos para realizar ingeniería directa e inversa, es decir, a partir del código de un programa, se puede obtener información sobre su diseño. Brinda facilidades para elaborar la documentación del software que se está desarrollando y posee un gran número de estereotipos predefinidos que agilizan el proceso de modelación. Posibilita capacidades avanzadas de modelado visual para bases de datos. Además permite a los desarrolladores ver cómo accederá la aplicación a la base de datos, de forma que los problemas se pueden escalar antes del desarrollo.

Esta herramienta brinda la posibilidad de construir un modelo de casos de usos bien definido, identificar los objetos y representar cómo interactúan con los diagramas de secuencia y colaboración, organizar los componentes de software y su despliegue para diseminarlos por los distintos nodos de una arquitectura de red, describir la estructura de las clases, generar el código fuente de las clases definidas en el diseño.

También se debe decir que una de sus limitaciones es que una vez generado el diagrama de clases persistentes a partir del cual se genera la base de datos del sistema, no existe la posibilidad de que el mismo exporte ese modelo hacia algún SGBD.

1.12.3 Sistemas de gestión de contenidos

Los sistemas de gestión de contenidos (Content Management Systems o CMS) son un software que se utiliza principalmente para facilitar la gestión web, ya sea en Internet o en una intranet, y por eso también son conocidos como gestores de contenido web (Web Content Management o WCM). Hay que tener en cuenta, sin embargo, que la aplicación de los CMS no se limita sólo al entorno web.

Los CMS son una interfaz que administran una o varias bases de datos donde se encuentra el contenido del sitio web. Permiten modificar el diseño del sitio en cualquier momento sin preocuparse del contenido, o viceversa, puesto que manejan ambos de manera independiente.

Lo más trascendental de los CMS es que son aplicaciones prefabricadas, esto tiene la ventaja de que con conocimientos técnicos muy básicos o casi ninguno, se puede lograr un resultado de mucha calidad. Por otra parte permiten a los desarrolladores más avanzados interactuar con el mismo

profundamente, ya que los CMS son una plataforma muy escalable y mediante el uso de plug-ins que pueden ser desarrollados por cualquier desarrollador, se puede extender ampliamente su funcionalidad.

CMS Drupal

Es un software que permite de una manera sencilla a una persona o una comunidad de usuarios publicar, gestionar y organizar una gran variedad de contenido en un sitio web. Decenas de miles de personas y organizaciones han utilizado Drupal para establecer diferentes tipos de sitios web. Drupal es un software de código abierto bajo la licencia GPL, y es mantenido y desarrollado por una comunidad de miles de usuarios y desarrolladores.

Requerimientos:

- Lenguaje de programación: PHP.
- Sistema gestor de base de datos: MySQL ó PostgreSQL.
- Servidor web: Apache, IIS.

Licencia: GPL.

1.13 Justificación de la propuesta de solución

De acuerdo a lo analizado a lo largo de este capítulo para la propuesta de solución, atendiendo al rol que corresponde al mismo (diseñador de base de datos), para la base de datos de SICI se define la siguiente propuesta:

En primer lugar se decide el empleo RUP como metodología de desarrollo de software; es una metodología que dará buen soporte al desarrollo por sus cualidades antes mencionadas. Teniendo en cuenta que RUP hace uso de UML como lenguaje de modelado para sus diagramas y plantillas, se utilizará la herramienta CASE Visual Paradigm (versión 6.0 Enterprise Edition), esta además cumple perfectamente con todos los estándares UML.

La plataforma de desarrollo será el CMS Drupal (versión 5.2), por su potencia y flexibilidad, y a la vez sencillez lo que permitirá construir un sistema de alta calidad y excelentes prestaciones que den respuesta a los requerimientos de los clientes, además por su vasta documentación y amplio conocimiento de uso por desarrolladores de la Universidad y del equipo del proyecto.

Esta versión de Drupal utiliza PHP 5 como lenguaje de programación, - lenguaje muy potente con amplias funciones - con vistas a incorporar nuevas funciones a Drupal o modificar las ya existentes, se utilizará el Entorno de Desarrollo Integrado (IDE, siglas en inglés de Integrated Development

Environment) Zend Studio, disponible para desarrolladores profesionales que agrupa todos los componentes de desarrollo necesarios para ciclo de desarrollo de aplicaciones PHP.

Como sistema gestor de base de datos se optará por PostgreSQL, se integra perfectamente con PHP y posee un rendimiento muy alto, actualmente es la opción de miles de sitios web de prestigiosas compañías profesionales que atienden millones de usuarios y recogen una basta información en su sistema.

1.14 Conclusiones

En este capítulo se analizaron temas como la actualidad de las bases de datos en el mundo, en Cuba y en la UCI. También se estudiaron las herramientas relacionadas con el diseño y modelado de bases de datos relacionales y se hizo una rigurosa investigación entre algunos SGBD para escoger el más adecuado según sus características y nuestras necesidades. Y se presenta la propuesta de solución para realizar el diseño de la BD.

CAPITULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

En el actual capítulo se presentará la descripción de la solución propuesta, prestando especial interés tanto a los requisitos funcionales como a los no funcionales. Se presentara el diagrama de clases persistentes, obtenido a partir del diagrama de clases del diseño realizado por los analistas y se realizará la descripción detallada de todas las clases que involucra. También se analizará el diseño de la base de datos, incluyendo el diagrama entidad relación de la base de datos y una descripción de aquellas tablas importantes en el diseño de la misma. Además veremos la descripción de la arquitectura utilizada y su fundamentación, así como la optimización de consultas.

2.1 Requisitos funcionales y no funcionales.

¿Qué es un requisito o un requerimiento?

Según la IEEE Standard Glossary of Software Engineering Terminology (Glosario de Terminología Estándar de la Ingeniería de Software) define un requerimiento como:

1. Condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo.
2. Condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente.
3. Una representación documentada de una condición o capacidad como en 1 o 2.

2.1.1 Requisitos funcionales

1) Subsistema de Trámite

RF 1 Gestionar datos de pasaporte.

RF 2 Gestionar datos de visa.

RF 3 Gestionar datos de pasaje.

RF 4 Gestionar datos de persona.

RF 5 Validar datos introducidos

RF 6 Registrar trámites de misionero.

RF 7 Gestionar estado de trámite de Misionero.

RF 8 Capturar entrega de documentos.

RF 9 Capturar recibo de documentos.

RF 10 Capturar fecha de salida de misiones.

- RF 11** Capturar fecha de salida de un / varios misionero(s).
- RF 12** Capturar fecha de llegada de misiones.
- RF 13** Modificar estado
- RF 14** Capturar fecha de llegada de un / varios misionero(s)
- RF 15** Consultar estado de permisos de salida.
- RF 16** Registrar trámites de Visitante.
- RF 17** Gestionar estado de trámite de Visitante.
- RF 18** Generar reportes de Trámite.
- RF 19** Generar documentos
- RF 20** Registrar valoración del MININT.
- RF 21** Gestionar datos de permiso de salida o habilitación

2) Subsistema de cooperación internacional

- RF 1** Autenticar usuarios.
- RF 2** Realizar solicitud misión.
- RF 3** Realizar visita.
- RF 4** Recibir información del viajero.
- RF 5** Gestionar información.
- RF 6** Gestionar datos de país.
- RF 7** Gestionar datos de embajada.
- RF 8** Obtener información de trámites.
- RF 9** Gestión de administración.

3) Subsistema de economía

- RF 1** Permitir la autenticación de un usuario.
- RF 2** Gestionar presupuesto de visita.
- RF 3** Gestionar presupuesto de misión.
- RF 4** Gestionar carta de solicitud de gastos.
- RF 5** Gestionar presupuesto de salón VIP.
- RF 6** Reportes de visitas.
- RF 7** Reportes de misiones.
- RF 8** Reportes de cheques.
- RF 9** Gestionar presupuesto de regalo.

- RF 10** Reportes de regalos.
- RF 11** Registrar facturas de misión.
- RF 12** Registrar facturas de visitas.
- RF 13** Registrar facturas de regalos.
- RF 14** Inventariar medios básicos.

4) Subsistema de relaciones públicas

- RF 1** Adicionar datos de la visita.
- RF 2** Editar datos de la visita
- RF 3** Eliminar datos de la visita
- RF 4** Buscar datos de la visita.
- RF 5** Adicionar datos del visitante
- RF 6** Editar datos del visitante
- RF 7** Eliminar visitante
- RF 8** Buscar datos del visitante
- RF 9** Adicionar programa de visita
- RF 10** Editar programa de visita
- RF 11** Eliminar programa de visita
- RF 12** Buscar programa de visita
- RF 13** Adicionar programa de atenciones
- RF 14** Editar programa de atenciones
- RF 15** Eliminar programa de atenciones
- RF 16** Buscar programa de atenciones.
- RF 17** Adicionar informe de visita
- RF 18** Editar informe de visita
- RF 19** Eliminar informe de visita
- RF 20** Buscar informe de visita

2.1.2 Requisitos no funcionales

Un producto software puede cumplir con todas las funcionalidades requeridas, pero si no es un software seguro y no cumple con las propiedades no funcionales, se puede asegurar que no es un software confiable y por tanto los clientes no quedan satisfechos con los resultados, para resolver estas incongruencias se establecen los requerimientos no funcionales, pues estos son propiedades o cualidades que el producto de software debe tener, ejemplo de ello son las restricciones del entorno o de la implementación, rendimiento, dependencias de la plataforma, facilidad de mantenimiento, entre otras.

Entre los requerimientos no funcionales del sistema propuesto se encuentran:

Apariencia o interfaz externa:

- Diseño orientado a llamar la atención del usuario y con una navegación sencilla.
- Construcción de enlaces rápidos o anclas para los documentos muy largos.
- Identificación de colores y formatos para resaltar la vigencia de los pasaportes y los permisos de salida.

Usabilidad:

- La aplicación garantiza una fácil interacción entre cliente y PC, de tal forma que no haya conflictos de usabilidad entre ambos.
- La aplicación contiene una ayuda la cual es muy importante para usuarios que no han interactuado aun con la aplicación o para aquellos que no tengan dominio de sistemas informáticos.
- El sistema podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de la computadora y de un ambiente Web en sentido general.

Rendimiento:

- Con la aparición del sistema informático, se podrán reducir el consumo de materias primas y obtener informaciones en un tiempo reducido.

Soporte:

- Para el servidor de base de datos:

- Se requiere un servidor de bases de datos con las siguientes características:
 - Soporte para grandes volúmenes de datos y velocidad de procesamiento.
 - Tiempo de respuesta rápido en accesos concurrentes.
- Para el servidor de aplicaciones:
 - Se requiere que esté instalado un intérprete de ficheros PHP.
 - Versión de PHP 4.3.0 o superior.
 - Servidor Web Apache 1.3.x o superior.
- Por parte del cliente:
 - Se requiere un navegador capaz de interpretar Java Script.

Portabilidad:

- Necesidad de que el sistema sea multiplataforma.
- El sistema deberá ser compatible con el sistema operativo UNIX (Linux).
- El sistema deberá ser compatible con el sistema operativo Windows (Versiones como 2000 y XP).

Seguridad:

- Para un mejor mantenimiento de los datos, los mismo deben de viajar por la red de forma encriptado.
- Para la entrada de usuarios al sistema debe ser verificado si el mismo ya esta autenticado, si no lo esta brindarle el servicio de autenticación.
- Macromedia Dreamweaver MX 2004.
- Adobe PhotoShop CS.
- MySQL 2.3.x.

- En las computadoras de los clientes solo se requiere de un navegador (Internet Explorer versión 4.0 o superior, Mozilla Firefox versión 1.5 o superior).
- En caso de que el usuario no contara con los recursos suficientes para que la aplicación funcione con la arquitectura descrita entonces la computadora tiene que tener instalados todos los programas antes mencionados.

Legales:

- El sistema se basa por manuales que se llevan a cabo en la DCI.
- La plataforma escogida para el desarrollo de la aplicación, está basada en la licencia GNU/GPL.

Confiabilidad:

- La información estará protegida para cualquier intruso que quiera realizar daños.
- La herramienta de implementación a utilizar tiene soporte para recuperación ante fallos y errores.
- Validar en todo momento que los datos introducidos tengan el formato correcto.

Funcionalidad:

- Reducir al mínimo el tiempo en que carga la aplicación.
- Guardar en caché páginas de contenido para agilizar la navegación de la página.

Software:

- Navegador compatible o superior con Internet Explorer 4, o Netscape Navigator.
- Se utilizará tecnología Apache 1.3.x.49.
- El sistema se desarrollará con tecnología PHP 4.3.4 o superior.

2.2 Estrategia de integración de la solución con otros módulos y sistemas.

SICI está estructurado en cuatro subsistemas, los cuales abarcan las funcionalidades principales para el desarrollo de las actividades en la DCI, los datos que provienen de dichos subsistemas es necesario que se almacenen en la BD para luego utilizarlos. Estos subsistemas tendrían una interacción de forma

indirecta con la BD física, todas las peticiones son manejadas por el CMS Drupal a través de su capa de abstracción de datos que provee a los desarrolladores con la posibilidad de construir dinámicamente las consultas a la base de datos. El objetivo de esta capa es conservar la sintaxis y poder del lenguaje de consultas SQL tanto como sea posible. Es preciso mencionar que las consultas no se construirán ni se guardaran en la base de datos lo que posibilita que si se decide cambiar el SGBD no se deban construir de nuevo, además a través de la ya antes mencionada capa de acceso a datos de Drupal se le pueden realizar consultas a múltiples bases de datos.

Los subsistemas encargados de la gestión de la información que se maneja en la DCI posteriormente se integraran en un solo Sistema Global.

Subsistemas de SICI:

- Subsistema de Trámite
- Subsistema de Cooperación Internacional
- Subsistema de Relaciones Públicas
- Subsistema de Economía

Para la futura implementación y operatividad de cada unos de los subsistemas fue necesario realizar un diseño de la base de datos en el que se reduce la redundancia de la información, logrando la integridad y control de los datos, además de lograr que la información guardada sea fácil de manipular.

2.3 Descripción de la arquitectura y fundamentación

El diseño de una base de datos es un proceso complejo que abarca decisiones a muy distintos niveles. La complejidad se controla mejor si se descompone el problema en subproblemas y se resuelve cada uno de ellos independientemente, utilizando técnicas específicas. Así, el diseño de una base de datos se descompone en diseño conceptual, diseño lógico y diseño físico.

La BD propuesta contiene 157 tablas en su modelo físico, para su funcionamiento drupal crea 77 tablas y las restantes 80 tablas almacenan la información para el funcionamiento básico de los cuatro subsistemas del proyecto: Cooperación, Trámite, Relaciones Públicas y Economía. Algunas de las tablas principales son misionero, misión, persona, pasaporte, extranjero, visa, visitante, solicitud, presupuesto, actividad; las cuales recogen la información de mayor peso. Por otra parte utilizamos como SGBD PostgreSQL y como plataforma de desarrollo el Sistema Manejador de Contenidos (CMS) Drupal.

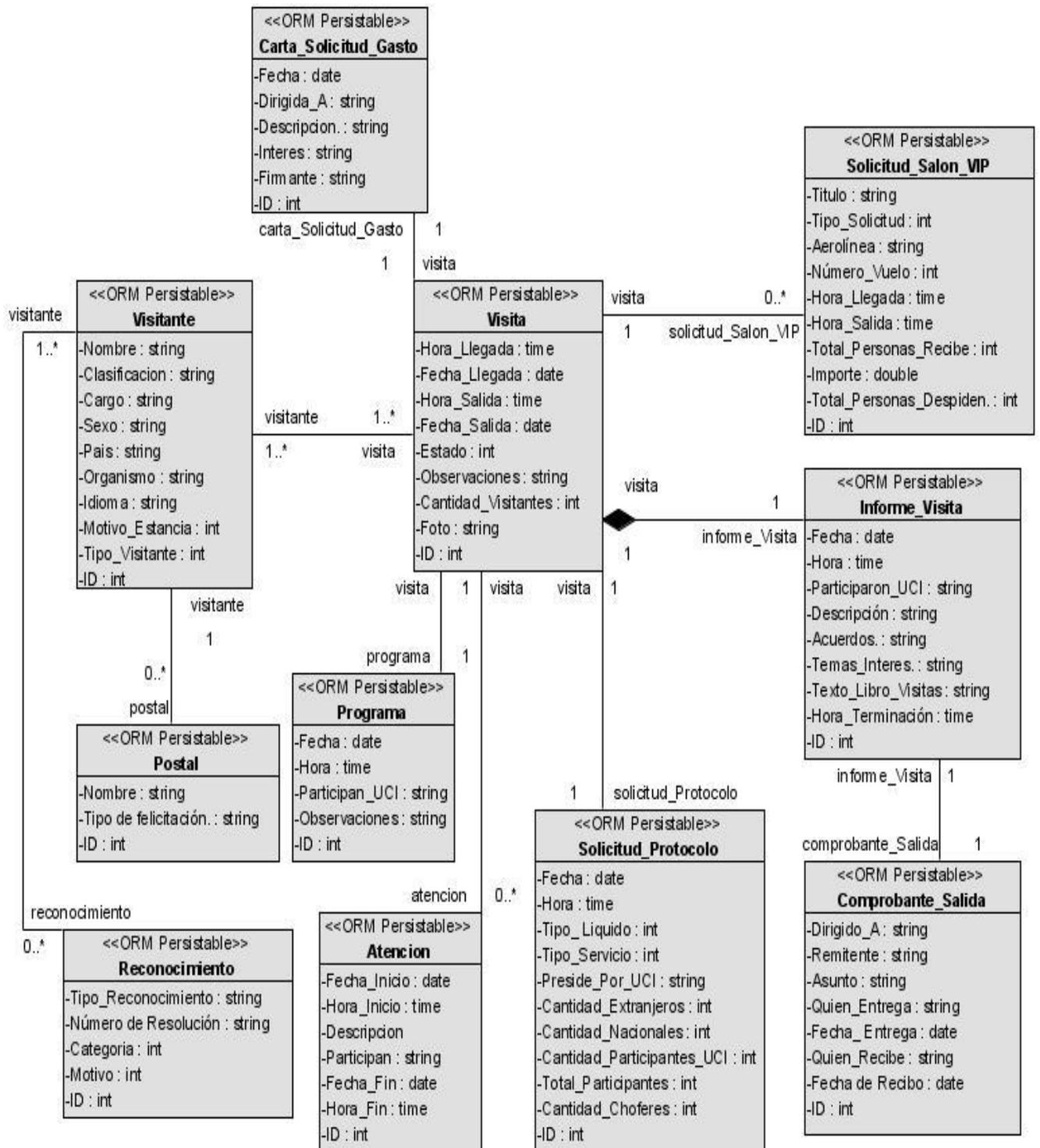


Figura 2. 2 Diagrama de clases persistentes del módulo de Economía

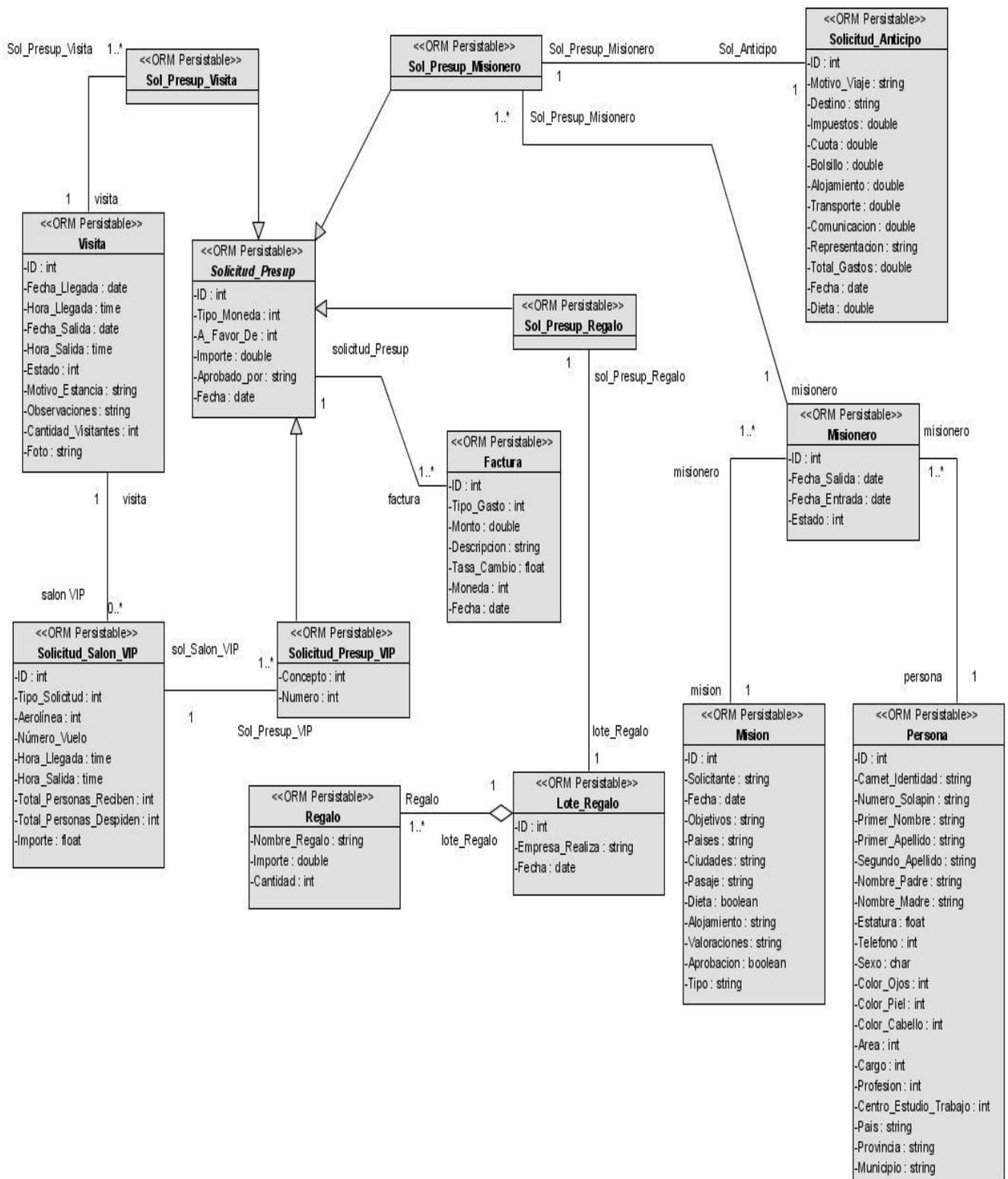


Figura 2. 3 Diagrama de clases persistentes del módulo de Cooperación

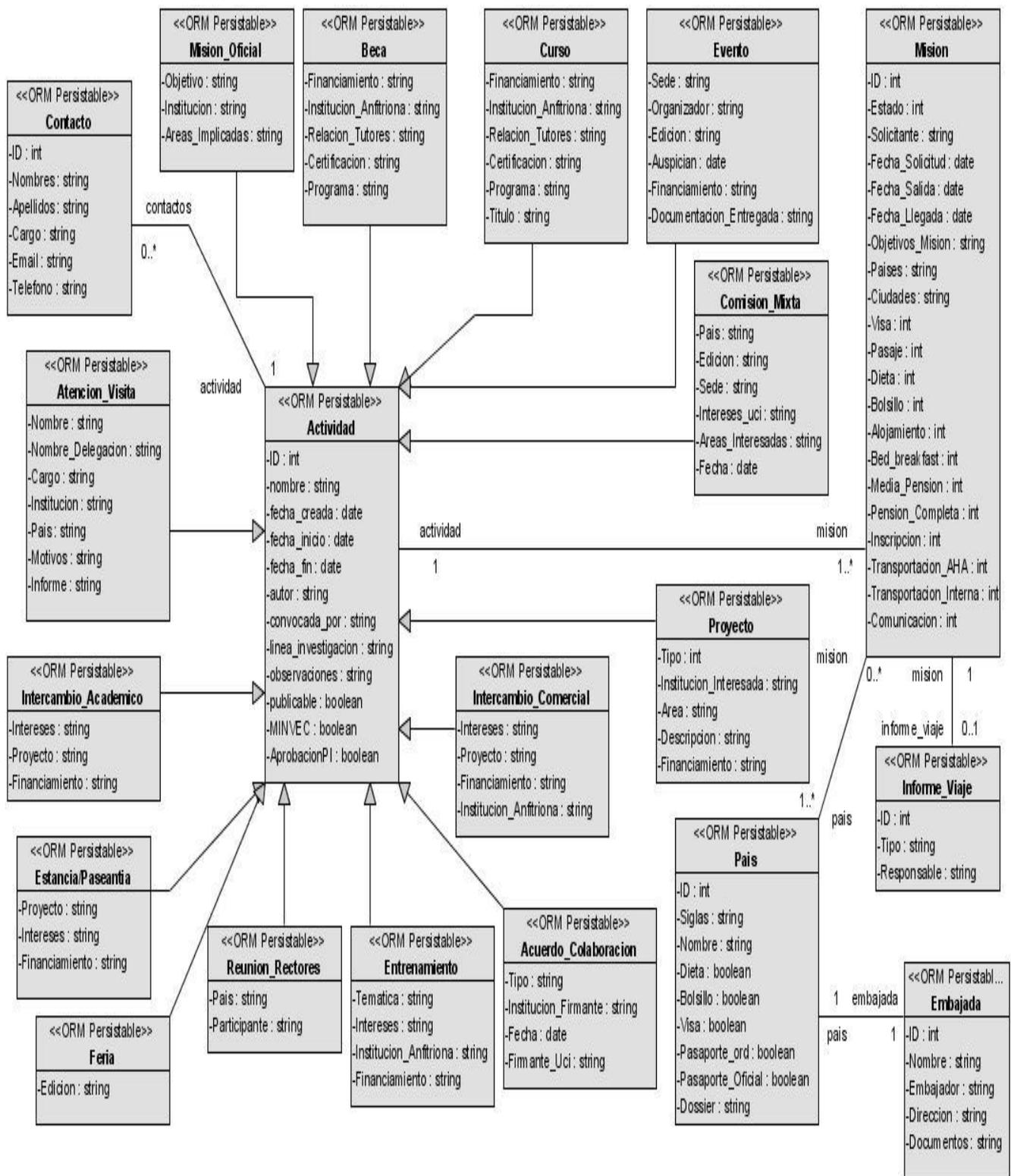


Figura 2. 4 Diagrama de clases persistentes del módulo de Relaciones Públicas

2.5 Diagramas Entidad-Relación

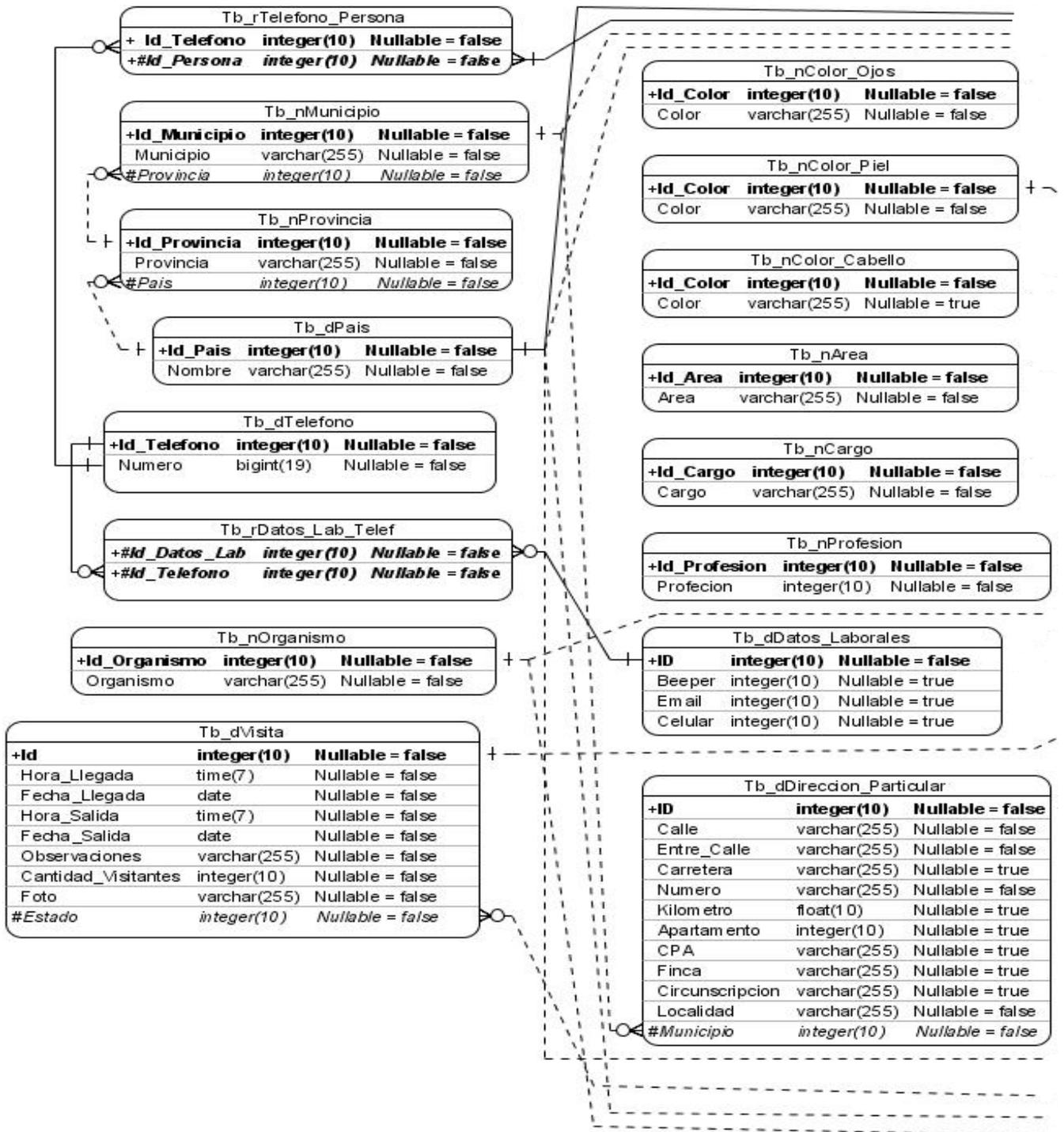


Figura 2. 5 Diagrama entidad relación del módulo de Trámites

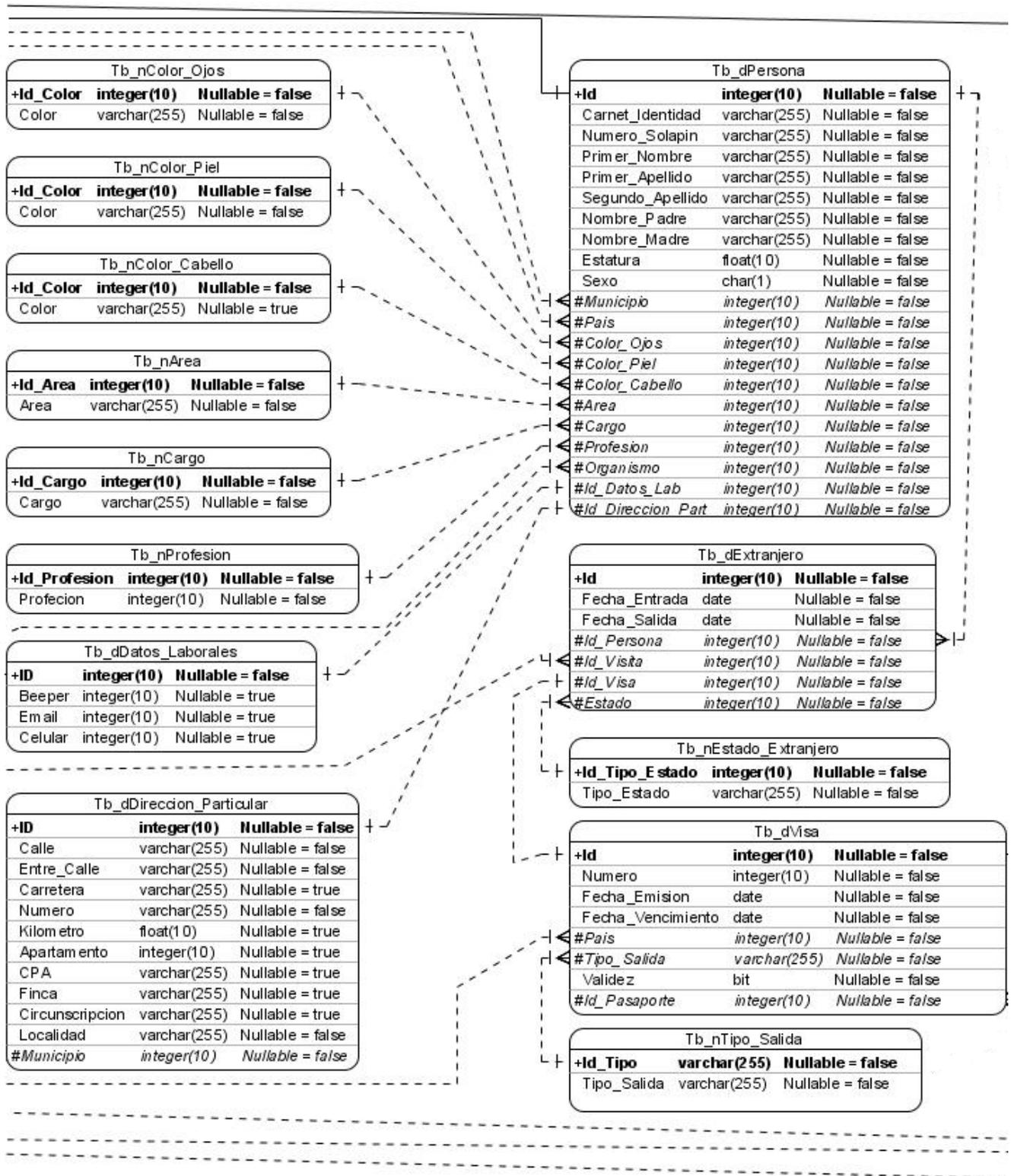


Figura 2. 6 Diagrama entidad relación del módulo de Trámites continuación

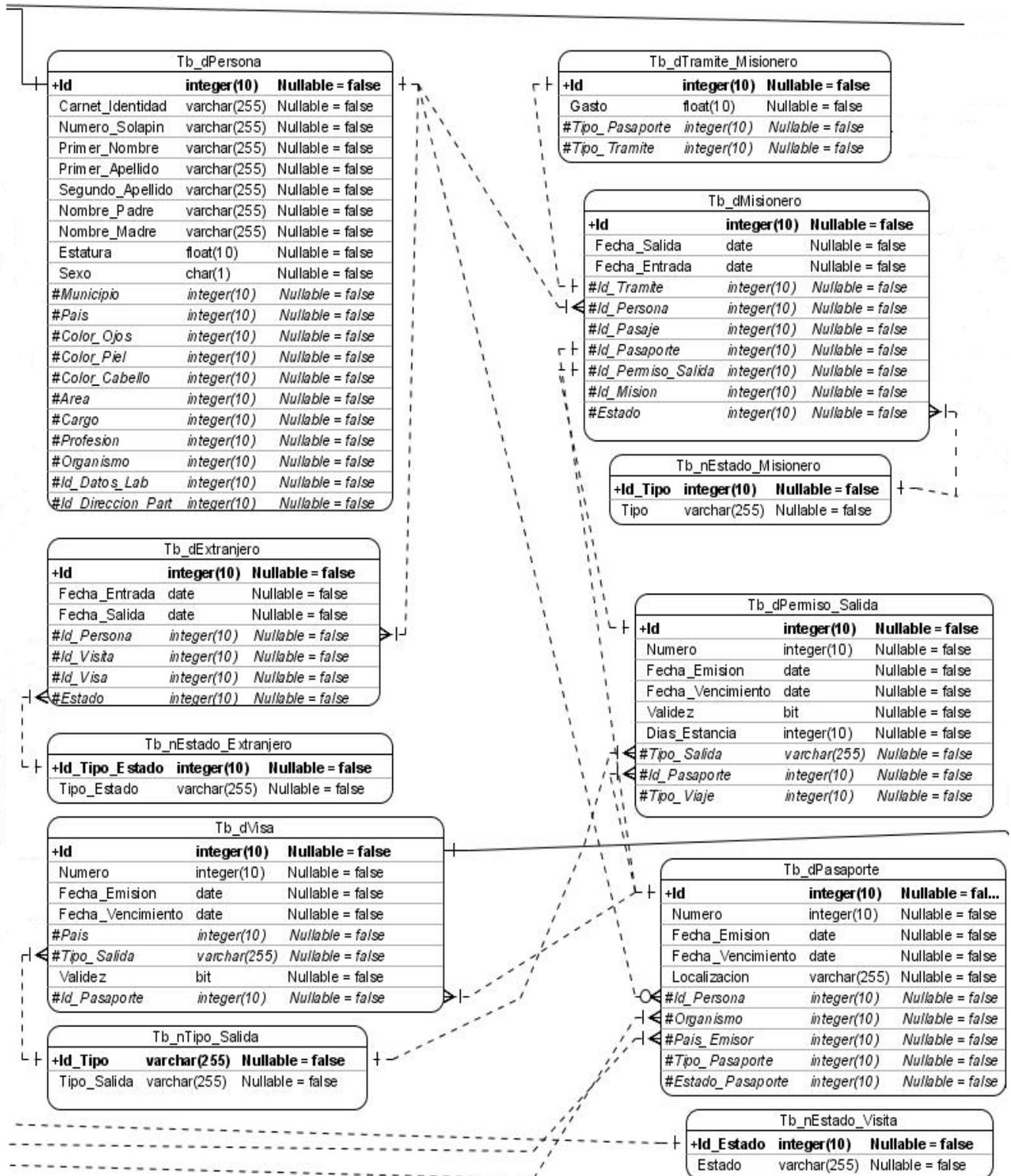


Figura 2. 7 Diagrama entidad relación del módulo de Trámites continuación

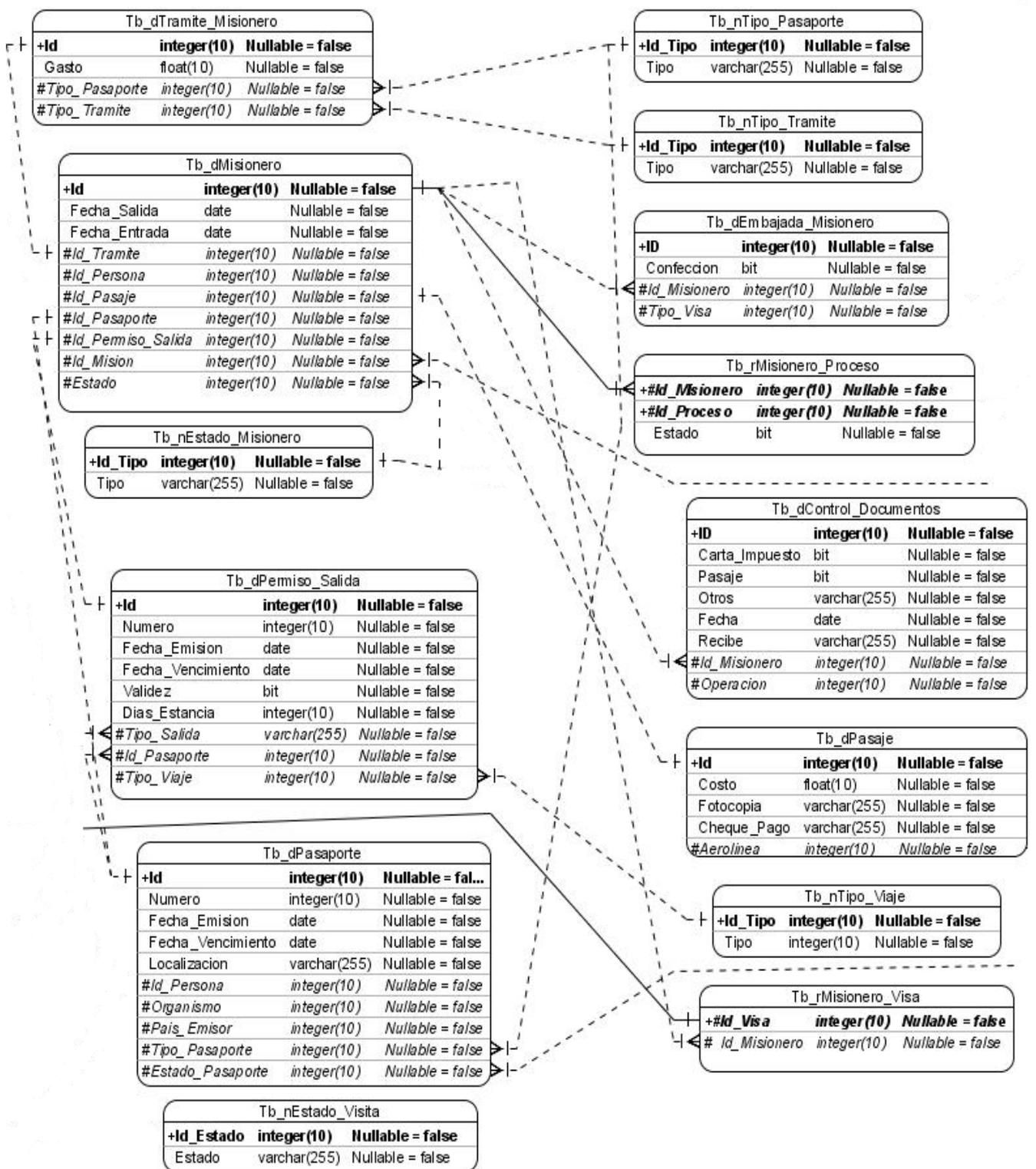


Figura 2. 8 Diagrama entidad relación del módulo de Trámites continuación

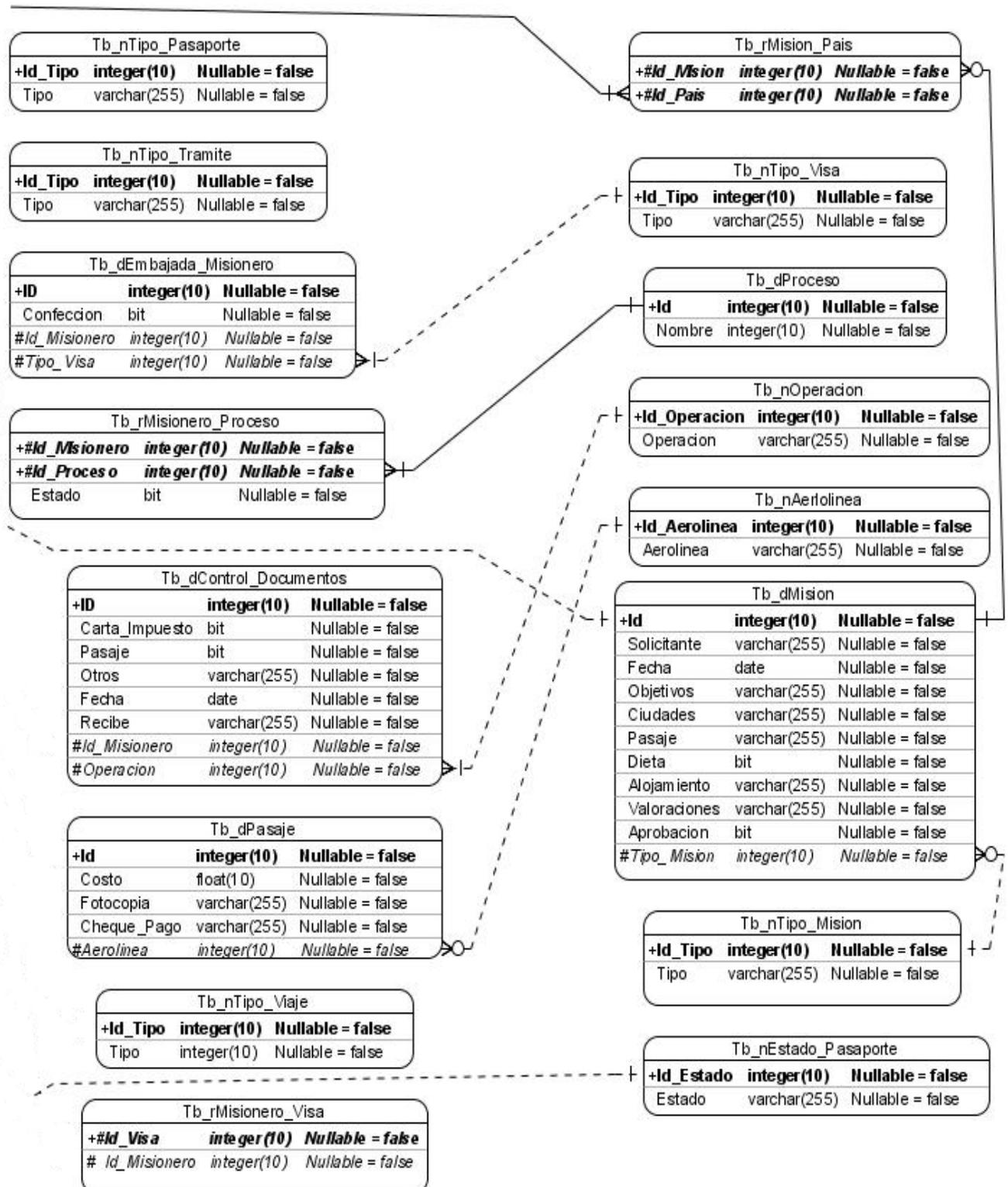


Figura 2. 9 Diagrama entidad relación del módulo de Trámites fin

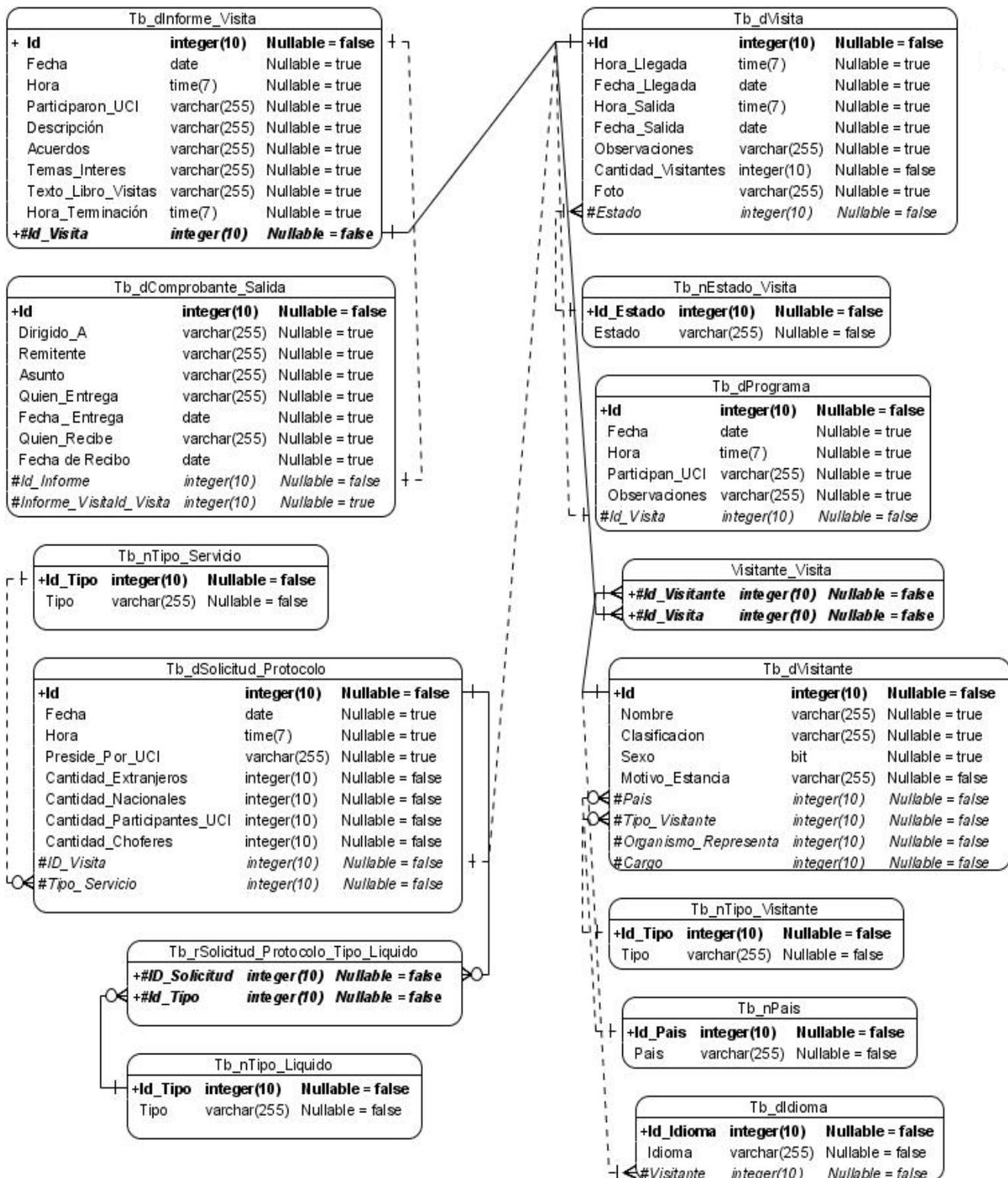


Figura 2. 10 Diagrama entidad relación del módulo de RRPP

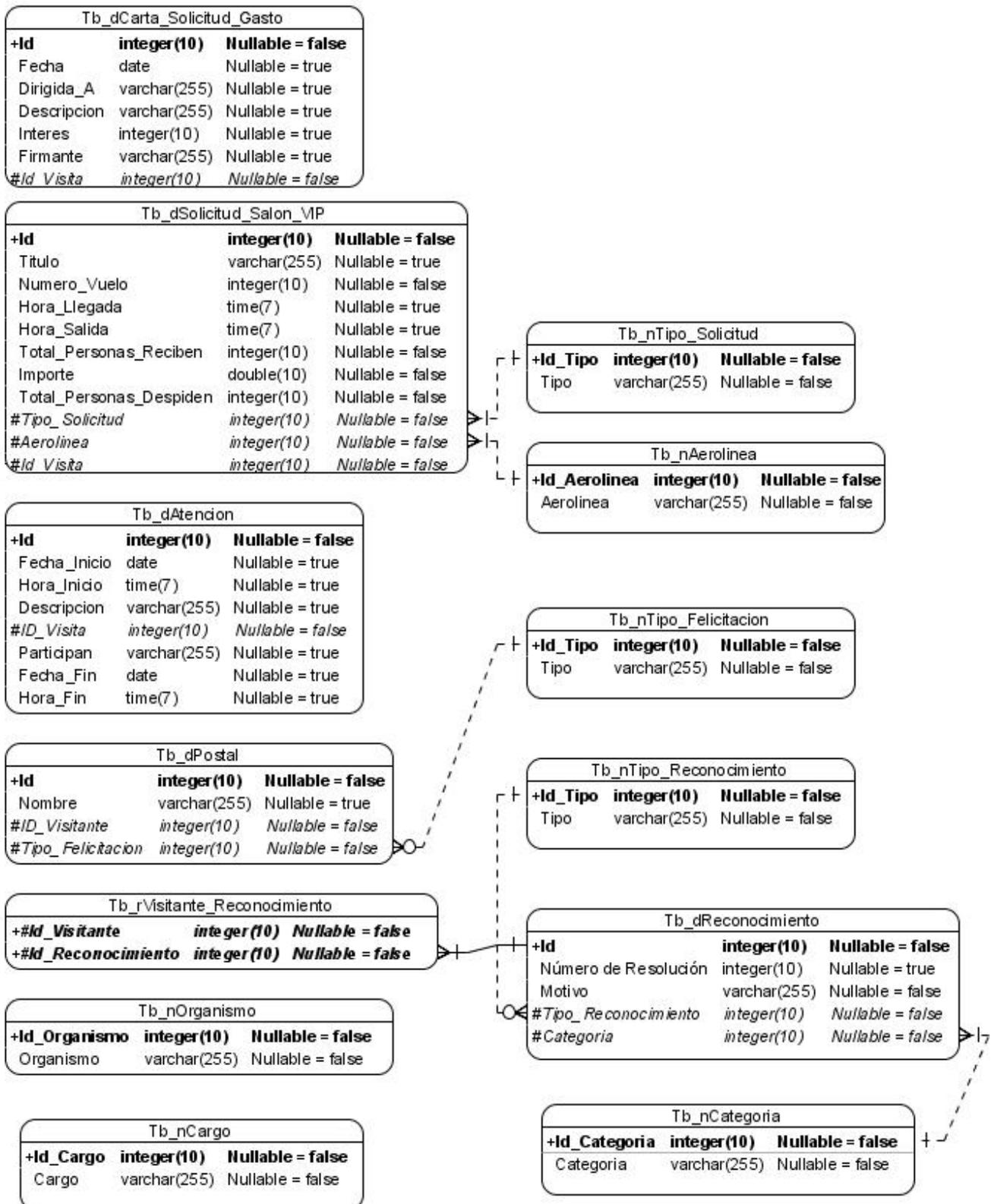


Figura 2. 12 Diagrama entidad relación del módulo de RRPP fin

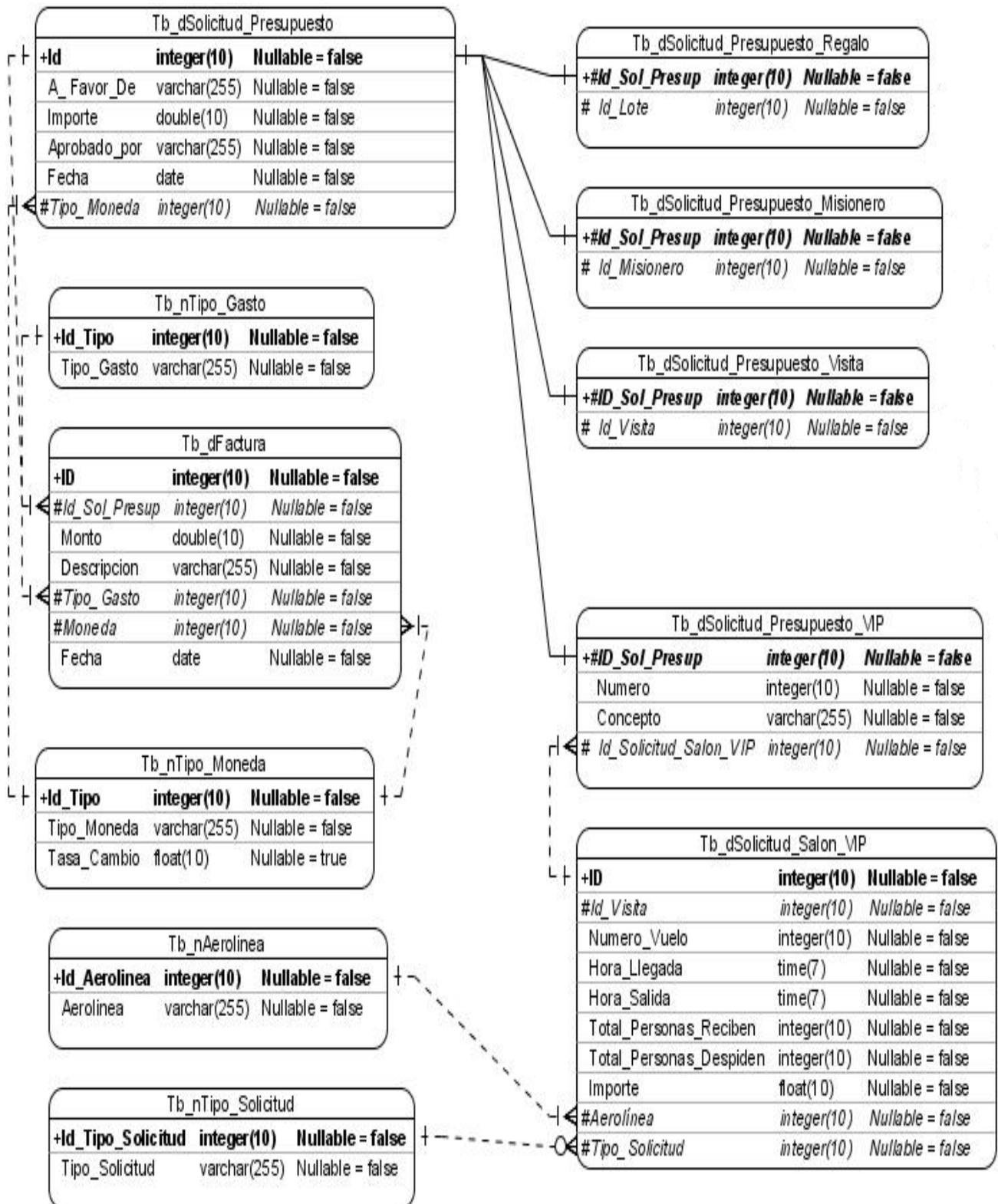


Figura 2. 13 Diagrama entidad relación del módulo de Economía

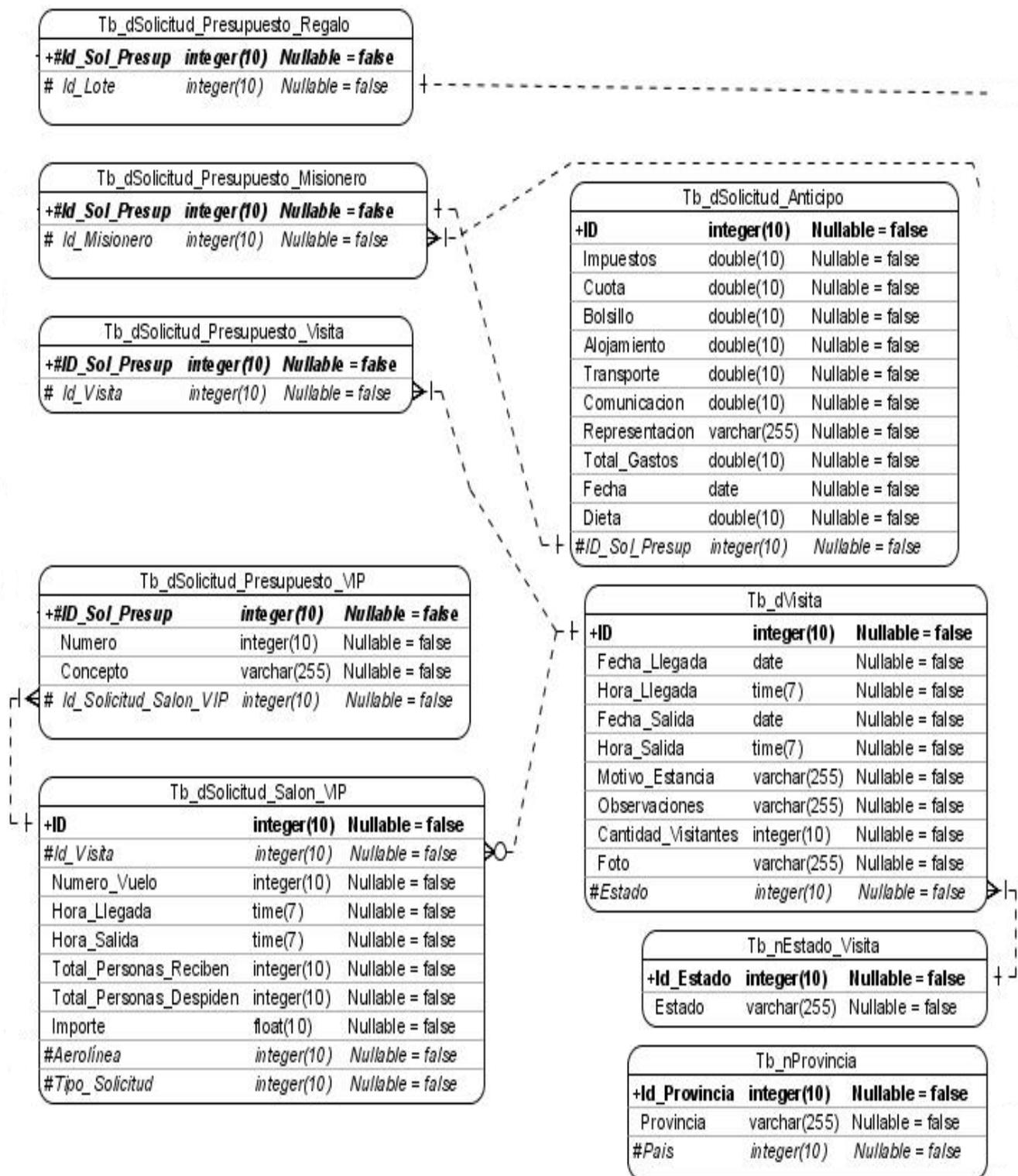


Figura 2. 14 Diagrama entidad relación del módulo de Economía continuación

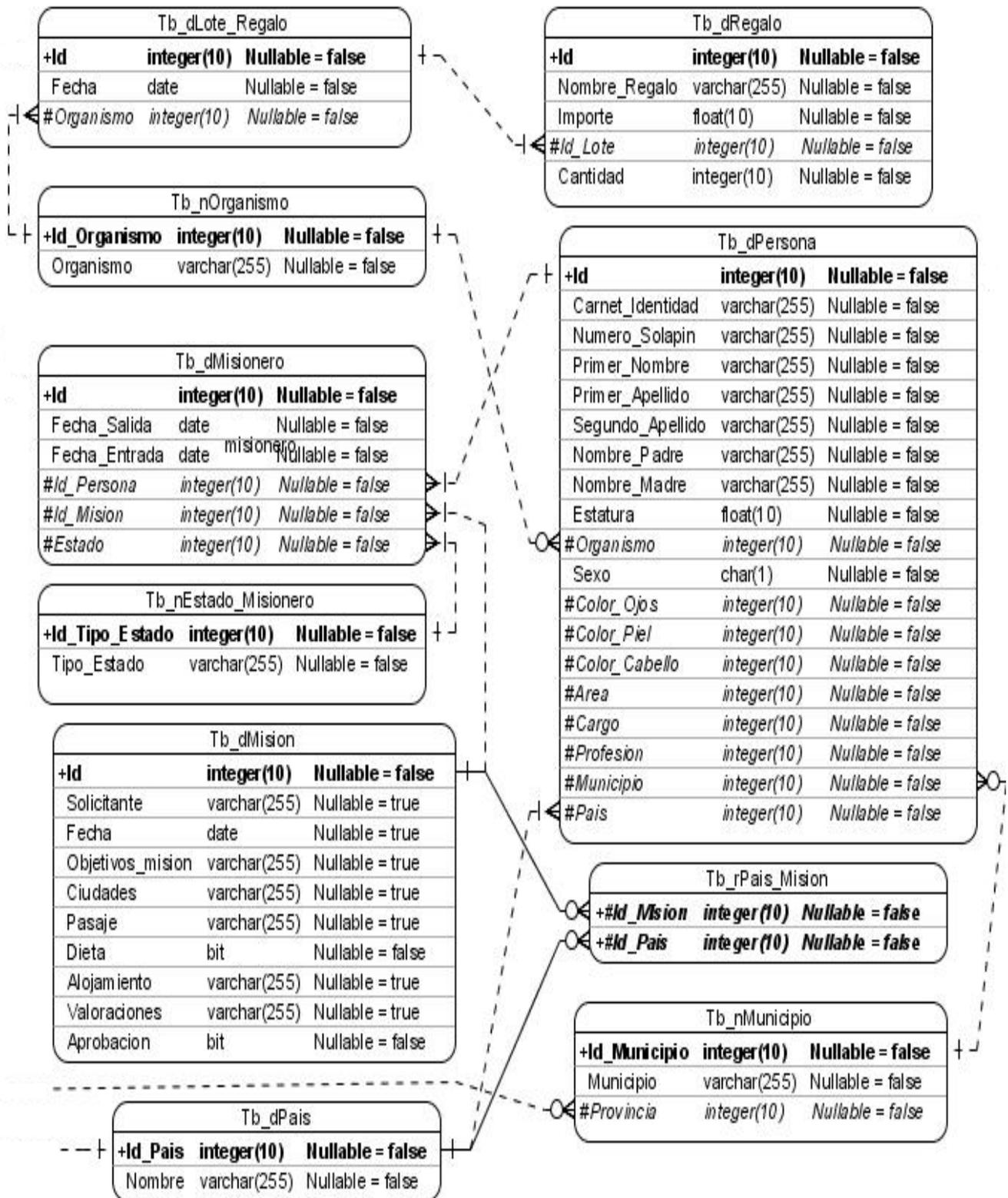


Figura 2. 16 Diagrama entidad relación del módulo de Economía continuación

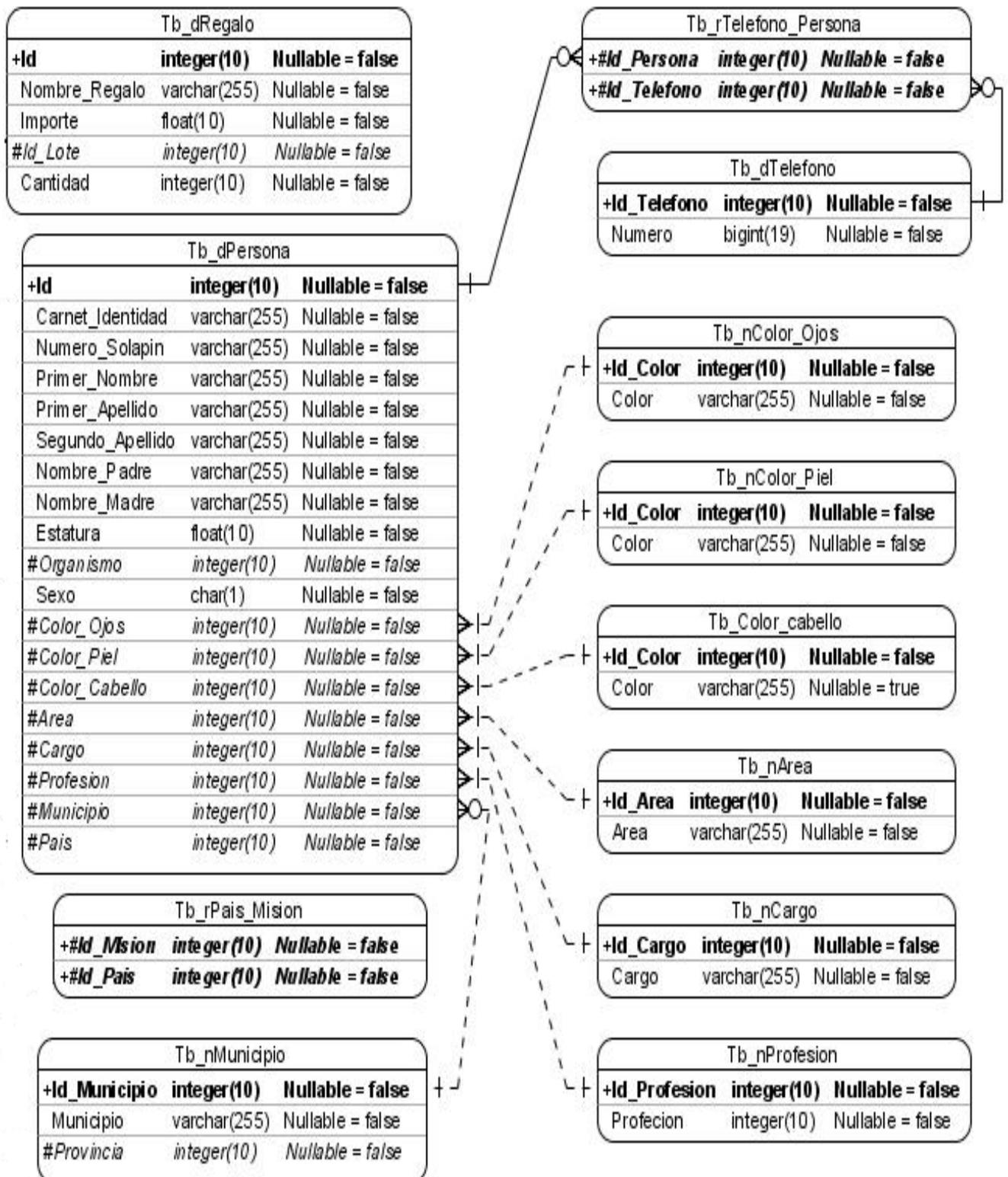


Figura 2. 17 Diagrama entidad relación del módulo de Economía fin

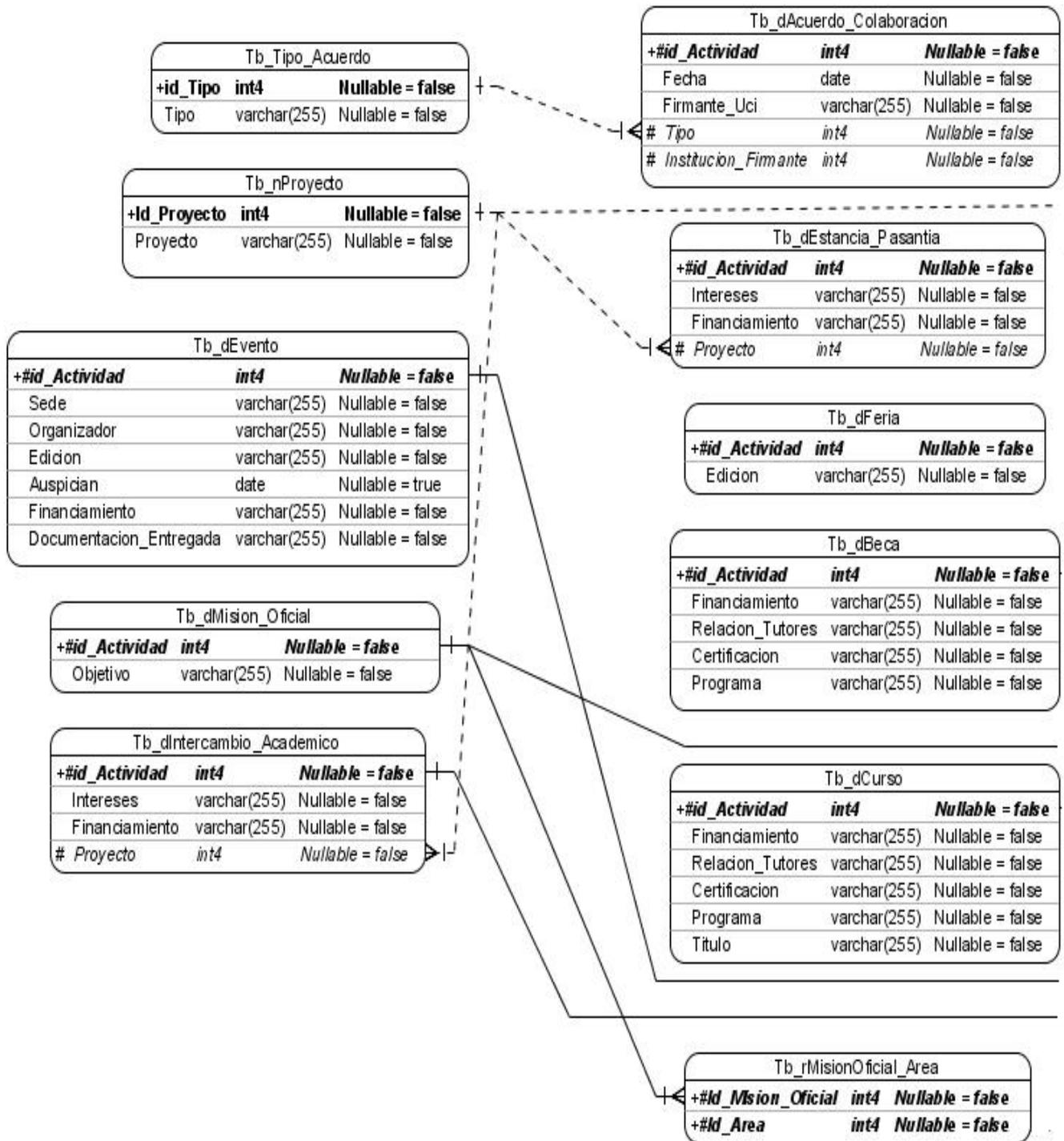


Figura 2. 18 Diagrama entidad relación del módulo de Cooperación Internacional

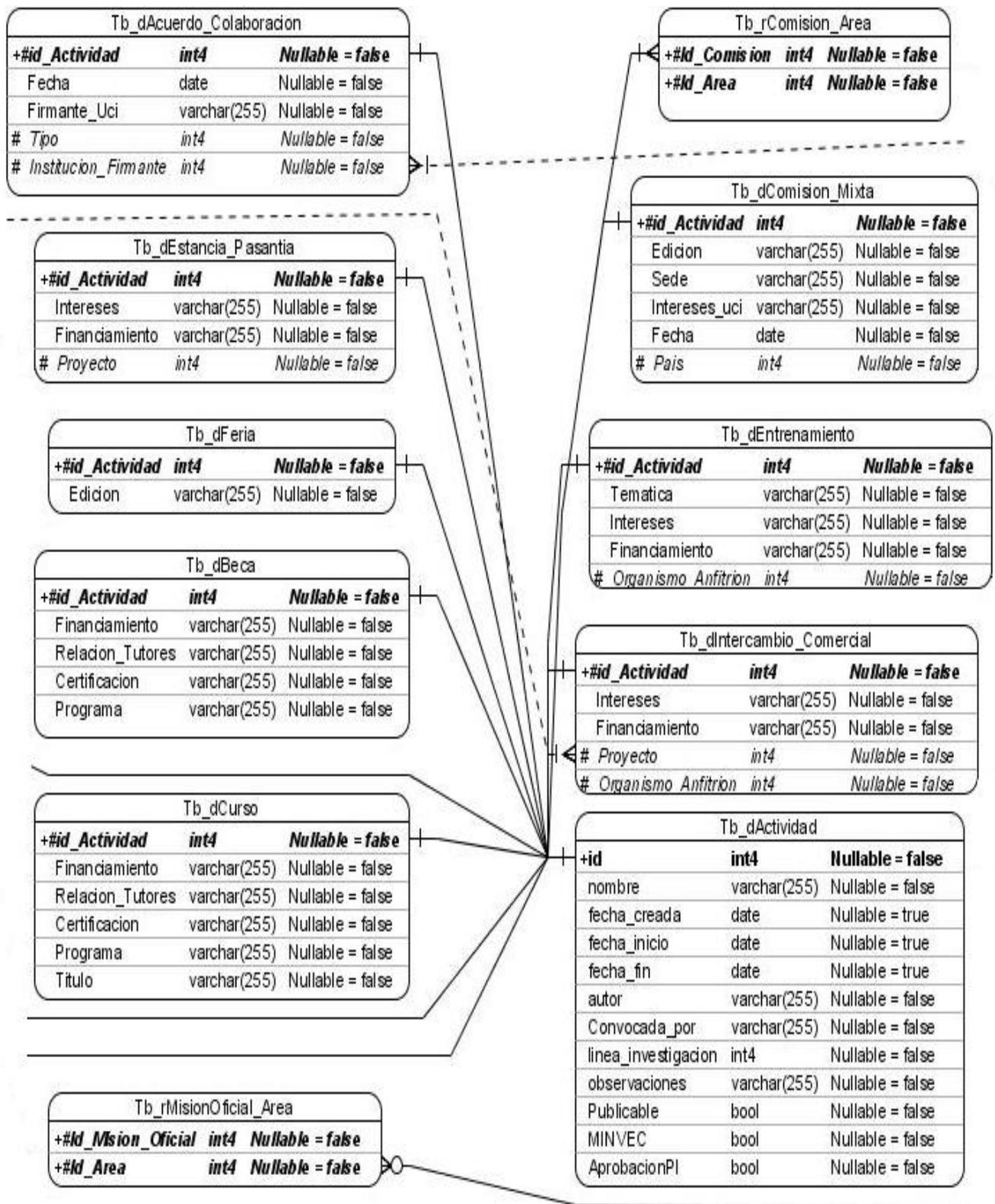


Figura 2. 19 Diagrama entidad relación del módulo de Cooperación Internacional continuación

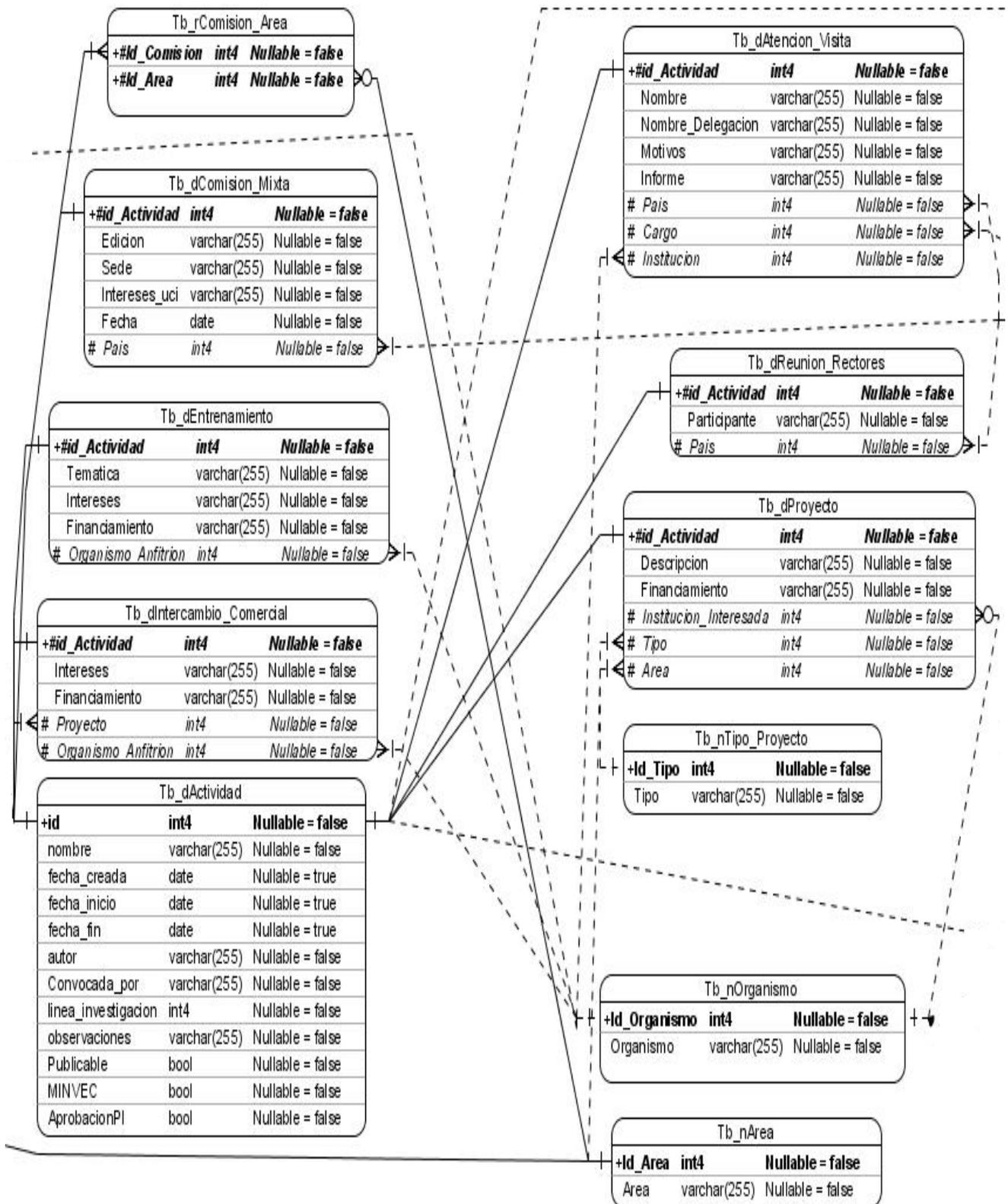


Figura 2. 20 Diagrama entidad relación del módulo de Cooperación Internacional continuación

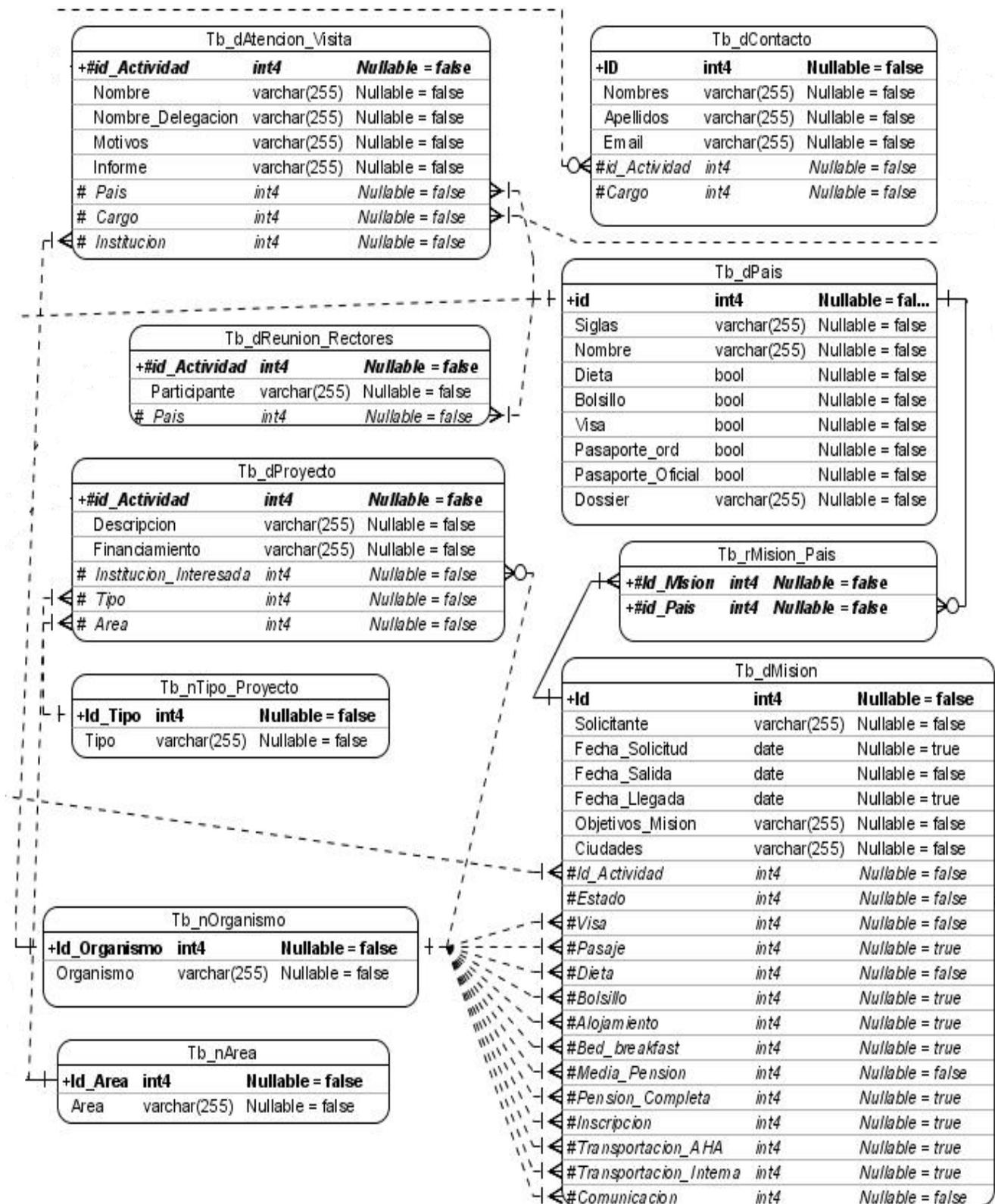


Figura 2. 21 Diagrama entidad relación del módulo de Cooperación Internacional continuación

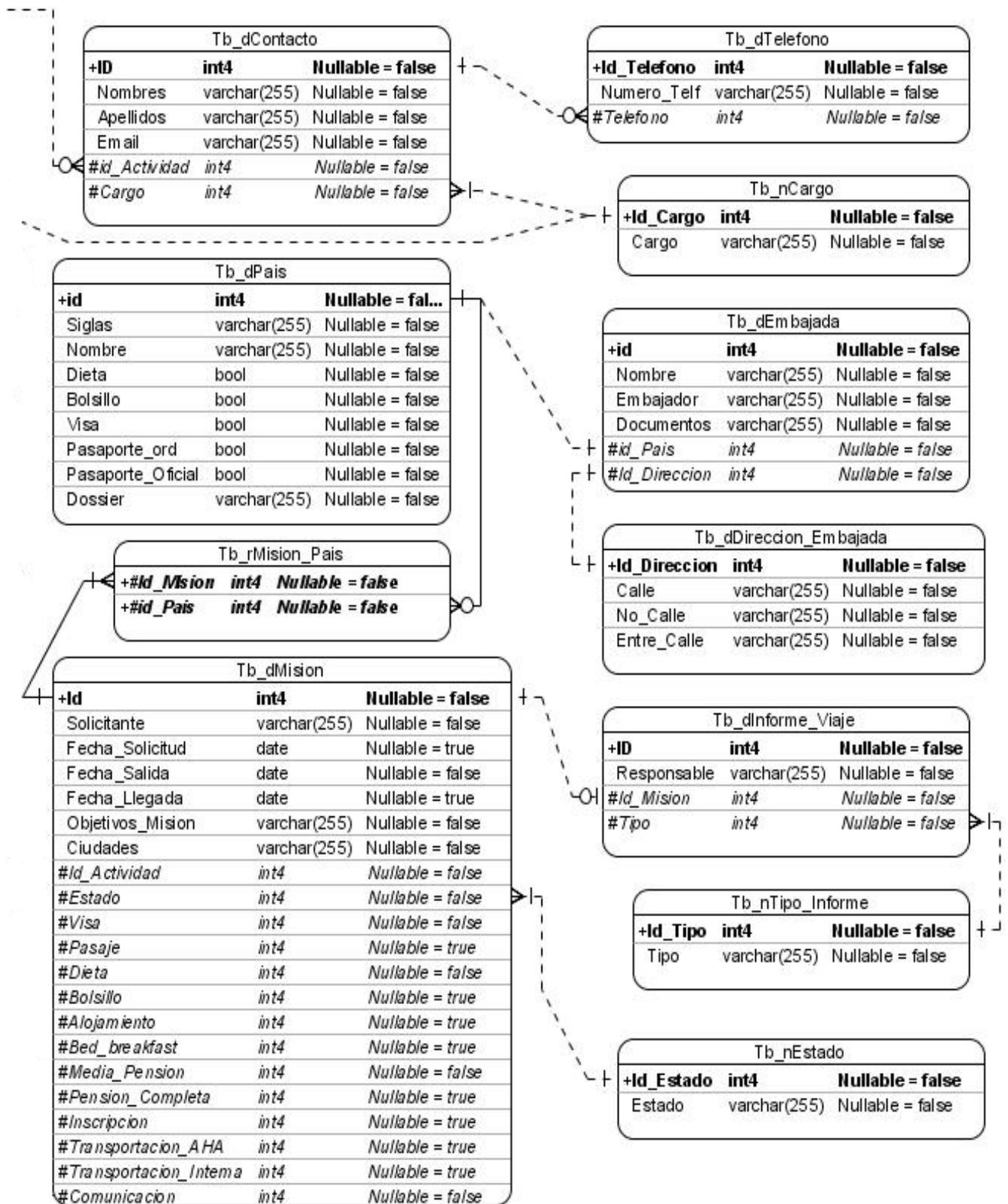


Figura 2. 22 Diagrama entidad relación del módulo de Cooperación Internacional fin

2.5.1 Descripción de las tablas de la base de datos.

Nombre: content_type_visita		
Descripción: Representa a la entidad visita, donde se guardaran sus datos		
Atributo	Tipo	Descripción
vid	int_unsigned	Identificador de la tabla de valor entero
nid	int_unsigned	Identificador de la tabla node
field_hora_llegada_visita_value	time	Hora de llegada de la visita
field_fecha_llegada_visita_value	date	Fecha de llegada de la visita
field_hora_salida_visita_value	time	Hora de salida de la visita
field_fecha_salida_visita_value	date	Fecha de salida de la visita
field_observ_visita_value	text	Observaciones
field_cant_visitantes_visita_value	integer	Cantidad de visitantes de la visita
field_foto_visita_title	varchar(255)	Dirección de la foto de la visita
field_estado_visita_nid	integer	Identificador del estado que presenta la visita

Tabla 3 Descripción de la tabla visita

Nombre: content_type_misionero		
Descripción: Representa a la entidad misionero, donde se guardaran sus datos		
Atributo	Tipo	Descripción
vid	int_unsigned	Identificador de la tabla de valor entero
nid	int_unsigned	Identificador de la tabla node
field_fecha_salida_misionero_value	date	Fecha de salida del misionero
field_fecha_entrada_misionero_value	date	Fecha de entrada del misionero
field_persona_misionero_nid	integer	Identificador de la tabla persona
field_mision_misionero_nid	integer	Identificador de la tabla misión
field_estadom_misionero_nid	integer	Identificador del nomenclador estado del misionero
field_pasaje_misionero_nid	integer	Identificador de la tabla pasaje con el que viaja el misionero
field_pasaporte_misionero_nid	integer	Identificador de la tabla pasaporte con el que viaja el misionero
field_permisosal_misionero_nid	integer	Identificador de la tabla permiso de salida con que viaja el misionero
field_tramite_misionero_nid	integer	Tramite que se le definió al misionero

Tabla 4 Descripción de la tabla misionero

Nombre: content_type_mision		
Descripción: Representa a la entidad misión, donde se guardaran sus datos		
Atributo	Tipo	Descripción
vid	int_unsigned	Identificador de la tabla de valor entero
nid	int_unsigned	Identificador de la tabla node
field_solicitante_mision_value	text	Nombre del solicitante
field_fechas_mision_value	date	Fecha de solicitud de la misión
field_fechass_mision_value	date	Fecha de salida de la misión
field_fechasll_mision_value	date	Fecha de llegada de la misión
field_ciudades_mision_value	text	Ciudades que involucra la misión
field_actividad_mision_nid	integer	Identificador de la tabla actividad a la que corresponde la misión
field_estadom_mision_nid	integer	Identificador de la tabla nomenclador estado de la misión
field_visa_mision_nid	integer	Identificador de la tabla nomenclador organismo, representa el organismo que corre con el gasto de la visa en la misión
field_pasaje_mision_nid	integer	Identificador de la tabla nomenclador organismo, representa el organismo que corre con el gasto de la pasaje en la misión
field_dieta_mision_nid	integer	Identificador de la tabla nomenclador organismo, representa el organismo que corre con el gasto de la dieta en la misión
field_bolsillo_mision_nid	integer	Identificador de la tabla nomenclador organismo, representa el organismo que corre con el gasto del dinero de bolsillo en la misión
field_alojamiento_mision_nid	integer	Identificador de la tabla nomenclador organismo, representa el organismo que corre con el gasto del alojamiento en la misión
field_camadesayuno_mision_nid	integer	Identificador de la tabla nomenclador organismo, representa el organismo que corre con el gasto de cama y desayuno en la misión
field_medpension_mision_nid	integer	Identificador de la tabla nomenclador organismo, representa el organismo que corre con el gasto de media pensión en la misión

field_pensioncomp_mision_nid	integer	Identificador de la tabla nomenclador organismo, representa el organismo que corre con el gasto de pensión completa en la misión
field_inscripcion_mision_nid	integer	Identificador de la tabla nomenclador organismo, representa el organismo que corre con el gasto de inscripción en la misión
field_transpaha_mision_nid	integer	Identificador de la tabla nomenclador organismo, representa el organismo que corre con el gasto de transportación AHA en la misión
field_traspint_mision_nid	integer	Identificador de la tabla nomenclador organismo, representa el organismo que corre con el gasto de transportación interna en la misión
field_comunicacion_mision_nid	integer	Identificador de la tabla nomenclador organismo, representa el organismo que corre con el gasto de comunicación en la misión
field_valoracion_mision_nid	text	Valoraciones de la misión
field_aprobacion_mision_nid	bit	Estado de aprobación de la misión

Tabla 5 Descripción de la tabla misión

Nombre: content_type_persona		
Descripción: Representa a la entidad persona, donde se guardarán sus datos		
Atributo	Tipo	Descripción
vid	int_unsigned	Identificador de la tabla de valor entero
nid	int_unsigned	Identificador de la tabla node
field_ci_persona_value	text	Carné de identidad de la persona
field_nosolapin_persona_value	text	Número de solapín de la persona
field_pnombre_persona_value	text	Primer nombre de la persona
field_papellido_persona_value	text	Primer apellido de la persona
field_snombre_persona_value	text	Segundo nombre de la persona
field_sapellido_persona_value	text	Segundo apellido de la persona
field_npadre_persona_value	text	Nombre del padre de la persona
field_nmadre_persona_value	text	Nombre de la madre de la persona
field_estatura_persona_value	float	Estatura de la persona
field_organismo_persona_nid	integer	Identificador de la tabla nomenclador organismo
field_sexo_persona_value	char	Sexo de la persona

field_cojos_persona_nid	integer	Identificador de la tabla nomenclador color de los ojos
field_cpiel_persona_nid	integer	Identificador de la tabla nomenclador color de la piel
field_ccabello_persona_nid	integer	Identificador de la tabla nomenclador color del cabello
field_area_persona_nid	integer	Identificador de la tabla nomenclador área
field_cargo_persona_nid	integer	Identificador de la tabla nomenclador cargo
field_profesion_persona_nid	integer	Identificador de la tabla nomenclador profesión
field_municipio_persona_nid	integer	Identificador de la tabla nomenclador municipio
field_pais_persona_nid	integer	Identificador de la tabla país
field_direccion_persona_nid	integer	Identificador de la tabla dirección de persona
field_datoslab_persona_nid	integer	Identificador de la tabla datos laborales

Tabla 6 Descripción de la tabla persona

Nombre: content_type_extranjero		
Descripción: Representa a la entidad extranjero, donde se guardaran sus datos		
Atributo	Tipo	Descripción
vid	int_unsigned	Identificador de la tabla de valor entero
nid	int_unsigned	Identificador de la tabla node
field_fecha_e_extranjero_value	date	Fecha de entrada del extranjero
field_fecha_s_extranjero_value	date	Fecha de salida del extranjero
field_persona_extranjero_nid	integer	Identificador de la tabla persona
field_visita_extranjero_nid	integer	Identificador de la tabla visita
field_visa_extranjero_nid	integer	Identificador de la tabla visa
field_estado_e_extranjero_nid	integer	Identificador de la tabla nomenclador estado del extranjero

Tabla 7 Descripción de la tabla extranjero

Nombre: content_type_visa		
Descripción: Representa a la entidad visa, donde se guardaran sus datos		
Atributo	Tipo	Descripción
vid	int_unsigned	Identificador de la tabla de valor entero
nid	int_unsigned	Identificador de la tabla node
field_numero_visa_value	integer	Número de la visa

field_fecha_e visa_value	date	Fecha de emisión de la visa
field_fecha_v visa_value	date	Fecha de vencimiento de la visa
field_pais_visa_nid	integer	Identificador de la tabla país al que pertenece la visa
field_pasaporte_visa_nid	integer	Identificador de la tabla pasaporte al que corresponde la visa
field_tsalida_visa_nid	integer	Identificador de la tabla nomenclador tipo de salida
field_validez_visa_value	bit	Si es válida o no

Tabla 8 Descripción de la tabla visa

Nombre: content_type_pasaporte		
Descripción: Representa a la entidad pasaporte, donde se guardaran sus datos		
Atributo	Tipo	Descripción
vid	int_unsigned	Identificador de la tabla de valor entero
nid	int_unsigned	Identificador de la tabla node
field_numero_pasaporte_value	integer	Número del pasaporte
field_fecha_e_pasaporte_value	date	Fecha de emisión del pasaporte
field_fecha_v_pasaporte_value	date	Fecha de vencimiento del pasaporte
field_persona_pasaporte_nid	integer	Identificador de la tabla persona al que pertenece el pasaporte
field_paise_pasaporte_nid	integer	Identificador de la tabla país que emite el pasaporte
field_tpasaporte_pasaporte_nid	integer	Identificador de la tabla nomenclador tipo de pasaporte
field_epasaporte_pasaporte_nid	integer	Identificador de la tabla nomenclador estado del pasaporte
field_local_pasaporte_value	text	Localización del pasaporte en archivos

Tabla 9 Descripción de la tabla pasaporte

Nombre: content_type_visitante		
Descripción: Representa a la entidad visitante, donde se guardaran sus datos		
Atributo	Tipo	Descripción
vid	int_unsigned	Identificador de la tabla de valor entero
nid	int_unsigned	Identificador de la tabla node

field_clasificacion_visitante_value	text	Clasificación del visitante
field_sexo_visitante_value	char	Sexo del visitante
field_idioma_visitante_value	text	Idioma del visitante
field_motivo_visitante_value	text	Motivo de estancia del visitante
field_nombre_visitante_value	text	Nombre del visitante
field_pais_visitante_nid	integer	Identificador de la tabla nomenclador país
field_cargo_visitante_nid	integer	Identificador de la tabla nomenclador cargo
field_institucion_visitante_nid	integer	Identificador de la tabla nomenclador organismo
field_tipo_visitante_nid	integer	Identificador de la tabla nomenclador tipo de visitante

Tabla 10 Descripción de la tabla visitante

Nombre: content_type_actividad		
Descripción: Representa a la entidad actividad, donde se guardaran sus datos		
Atributo	Tipo	Descripción
vid	int_unsigned	Identificador de la tabla de valor entero
nid	int_unsigned	Identificador de la tabla node
field_nombre_value	text	Nombre de la actividad
field_fecha_creda_value	date	Fecha de creación de la actividad
field_fecha_inicio_value	date	Fecha de inicio de la actividad
field_fecha_fin_value	date	Fecha de fin de la actividad
field_autor_value	text	Autor de la actividad
field_convocada_por_values	text	Por quién es convocada la actividad
field_linea_investigacion_value	integer	Línea de investigación de la actividad
field_observaciones_value	text	Observaciones de la actividad
field_publicable_value	bit	Si se puede publicar o no la actividad
field_minvec	bit	Si lleva o no aprobación del minvec
field_aporbacionpi_value	bit	Si lleva o no aprobación PI

Tabla 11 Descripción de la tabla actividad

2.6 Análisis de optimización de consultas

La optimización de consultas es una consideración importante en una base de datos, ya que la diferencia en tiempo de ejecución entre una buena y una mala estrategia puede ser sustancial. Las

consultas relacionales pueden manipular de una vez conjuntos enteros de registros. Es, por lo tanto, posible y deseable optimizar de forma automática las consultas.

En las bases de datos relacionales el lenguaje de consultas: Lenguaje de Consulta Estructurado (SQL [/esecuele/ en español, /sicuél/ en inglés] Structured Query Language) es el más utilizado por los programadores y desarrolladores para obtener información desde las BD. La complejidad que pueden alcanzar algunas consultas puede ser tal, que su diseño puede tomar un tiempo considerable, obteniendo no siempre una respuesta óptima.

Como en el caso de los más modernos lenguajes relacionales, SQL está basado en el álgebra y el cálculo relacional de tuplas permitiendo lanzar consultas con el fin de recuperar -de una forma sencilla- información de interés de una base de datos, así como también hacer cambios sobre la misma. Como resultado, toda consulta formulada utilizando el cálculo relacional de tuplas (o su equivalente, el álgebra relacional) se puede formular también utilizando SQL.

Cuando hablamos de optimización de consultas nos referimos a mejorar los tiempos de respuesta en un sistema de gestión de bases de datos relacional, pues la optimización es el proceso de modificar un sistema para mejorar su eficiencia, el uso de los recursos disponibles y la eficiencia en la estrategia de ejecución para procesar cada consulta.

Para llevar a cabo el proceso de optimización se deben tener en cuenta una serie de medidas que aseguran su realización como por ejemplo:

Es preferible no usar la cláusula **HAVING** si la condición deseada se puede expresar en la cláusula **WHERE**.

Crear índices en tantos campos como sea posible. Debe tenerse en cuenta que una tabla con demasiados índices baja el rendimiento de la BD.

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.

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.

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.

Utilizar para almacenar los datos el tipo de dato de menor tamaño que pueda contenerlos.

Evitar operadores costosos, como el **NOT LIKE**. Algunos operadores, tanto en los joins como en los predicados del **WHERE**, tienden a producir operaciones muy costosas. El operador **LIKE** con un valor encerrado entre los comodines ('valor') casi siempre produce un barrido de la tabla.

Consultar únicamente las filas y columnas requeridas. No abusar del "**SELECT * FROM...**". Se debe restringir las búsquedas. No hacerlo ralentiza las consultas muchos más que si se retornaran solo los datos necesarios y no todos.

Consulta de ejemplo, primera variante:

Seleccionar todas las personas en una misión dada.

```
SELECT * FROM content_type_persona INNER JOIN content_type_misionero
ON content_type_persona.nid = content_type_misionero.field_persona_misionero_nid
WHERE content_type_misionero.field_mision_misionero_nid = 3050.
```

Esta consulta fue analizada con el comando **EXPLAIN** con el cual PostgreSQL muestra el plan de ejecución que su planificador genera para la consulta dada. El plan de ejecución muestra la manera en que serán escaneadas las tablas referenciadas. La parte más crítica de la presentación es el costo estimado de ejecución de la consulta, que es la suposición del planificador sobre el tiempo que tomará correr la consulta (medido en unidades de captura de páginas de disco). Después de analizar la consulta se obtuvo un costo de 0.180 segundos.

Escribiendo una variante de la consulta anterior de manera tal que se restrinjan los campos a buscar y haciendo uso una de las ventajas que ofrece PostgreSQL como lo son las subconsultas se obtiene un costo de 0.125 segundos. En el resultado se puede apreciar la disminución del costo de realización de la consulta, demostrando una mejor optimización.

Consulta de ejemplo, segunda variante:

```
SELECT content_type_persona.nid
FROM content_type_persona
WHERE content_type_persona.nid IN
(
SELECT content_type_misionero.field_persona_misionero_nid
FROM content_type_misionero
```

WHERE content_type_misionero.field_mision_misionero_nid = 3050

)

Evitar el uso de funciones explícitas o implícitas en el predicado de la cláusula **WHERE**. Las columnas en esta cláusula son vistas como expresiones en lugar de columnas.

Es recomendable a la hora de utilizar varias tablas dentro de una misma consulta tener cuidado con el orden empleado en la cláusula **FROM** y no usar **WHERE** en vez de **ON** por ejemplo:

Para saber cuántas Personas han participado en una Misión determinada en una fecha indicada se puede escribir:

```
SELECT count (content_type_persona.nid)
FROM content_type_misionero, content_type_persona
WHERE content_type_persona.nid = content_type_misionero.field_persona_misionero_nid
AND content_type_misionero.field_fecha_entrada_misionero_value = '10/12/1984'
AND content_type_misionero.field_mision_misionero_nid = 3050
```

Mediante esta construcción el gestor recorrerá todas las personas que coincidan en la tabla misionero para después hacerle un filtrado por las condiciones que deben cumplir y luego sumar la cantidad y devolver el resultado.

Si por el contrario se escribe:

```
SELECT count (content_type_persona.nid)
FROM content_type_misionero INNER JOIN content_type_persona
ON content_type_misionero.field_fecha_entrada_misionero_value = '10/12/1984'
WHERE content_type_misionero.field_mision_misionero_nid = 3050
AND content_type_persona.nid = content_type_misionero.field_persona_misionero_nid
```

El gestor filtra primero los misioneros, busca la misión, busca las coincidencias en la tabla personas y finalmente suma la cantidad de personas, disminuyendo el tiempo de la consulta de 0.28 a 0.22.

PostgreSQL cuenta con un comando denominado **VACUUM**, que limpia y analiza una Base de Datos. El **VACUUM** da la posibilidad de imprimir un reporte detallado de la actividad de análisis para cada tabla y de actualizar las estadísticas de columnas usadas por el optimizador para determinar la manera

más eficiente de ejecutar una consulta. Las estadísticas representan la dispersión de los datos en cada columna.

La ejecución de **VACUUM** periódicamente aumenta la velocidad de la base de datos al procesar las consultas del usuario. El comando **VACUUM** puede ser ejecutado en cualquier momento, particularmente, después de copiar una clase grande en PostgreSQL o después de borrar un gran número de registros. Esto actualizará los catálogos del sistema con todos los cambios recientes, y permitirá al organizador de consultas de PostgreSQL tomar las mejores decisiones al planear las consultas de los usuarios.

2.7 Conclusiones

En este capítulo se presentaron los requisitos funcionales y no funcionales, el diagrama de clases persistentes, del cual se detallaron cada una de sus clases, mostrando así sus atributos y tipos de estos atributos, además se vio el diagrama entidad relación. Otro aspecto muy importante que se consideró en este capítulo fue la optimización de consultas de la base de datos, lo que permitió hacer un análisis de cómo se podría reducir el tiempo de ejecución, es decir el tiempo que se demora en dar una respuesta el sistema gestor de bases de datos. Con esto lo que buscamos es mejorar la eficiencia en la estrategia de ejecución para procesar las consultas y la disponibilidad de los recursos. También se abordó el tema de la arquitectura utilizada.

CAPITULO 3. VALIDACIÓN DEL DISEÑO REALIZADO

3.1 Introducción

En este capítulo se analizarán aspectos relacionados con la validación teórica y funcional del diseño realizado a la base de datos para SICI, haciendo referencia al proceso de normalización aplicado, las restricciones de integridad impuestos para lograr un correcto funcionamiento, así como la aplicación de mecanismos de eliminación de redundancia de la información teniendo en cuenta el diseño físico de la base de datos. También se muestran los análisis de la seguridad que presenta la base de datos, y su respuesta a las distintas transacciones que se ejecutan en ella con el objetivo de alcanzar un correcto funcionamiento.

3.2 Validación teórica del diseño

3.2.1 Integridad

En las aplicaciones actuales de las metodologías de desarrollo de base de datos se puede apreciar que los modelos conceptuales son cada vez más ricos semánticamente hablando y esto implica que son capaces de recoger con mayor precisión las especificaciones del entorno o del universo de discurso de los problemas que se quieran resolver. Las complicaciones para los diseñadores de bases de datos surgen cuando quieren mantener la semántica de la vida real en su diseño, sobre todo en la fase lógica; donde la mayoría de las metodologías coinciden en la utilización del modelo relacional para transformar el esquema conceptual en un esquema cercano a la implementación. Para ello se aplican un conjunto de reglas de transformación. Estas reglas básicas sólo especifican la transformación de los elementos más simples y sencillos del modelo conceptual. En general, esto implica una pérdida de semántica significativa que lleva al diseñador a controlar y chequear las restricciones de integridad de la base de datos

Las restricciones de integridad garantizan que el contenido de la base de datos es conforme con las reglas establecidas para presentar el universo del discurso. La integridad de una base de datos significa la existencia de dos componentes importantes que son la exactitud y la completitud. Es decir, que la integridad de base de datos garantiza que todos los datos sean correctos, válidos y relevantes. Se considera que la tarea de garantizar estas dos componentes es difícil porque en esa fase todavía no existe ningún modelo de datos que puede capturar la semántica completa que se requiere modelar, o la base de datos puede aceptar algunos datos que son válidos para algunos hechos, sin embargo estos datos podrían causar la presencia de otros que no tengan ninguna relación con el dominio de la aplicación. La falta de exactitud y completitud en la base de datos puede llevar a deducir hechos que

no son reales. Por eso, la mayoría de las bases de datos necesitan apoyarse de distintos mecanismos automáticos para vigilar y garantizar el contenido de las mismas, ellos son:

- ✓ Restricciones de unicidad
- ✓ Restricciones de valor no nulo
- ✓ Restricciones de clave primaria
- ✓ Restricciones de dominio.
- ✓ Restricciones de integridad de entidades.
- ✓ Restricciones de integridad referencial

Restricciones de unicidad:

Mediante esta regla se restringen los valores de un conjunto de atributos para que no puedan repetirse en una relación. Esta restricción permite la definición de claves alternativas o candidatas. En la base de datos se aplicó esta restricción en muchas tablas que necesitaban reflejar la propiedad de que algunos de sus campos cumplieran la condición de ser únicos.

```
CREATE TABLE content_type_postal (  
  vid int_unsigned NOT NULL,  
  nid int_unsigned NOT NULL,  
  field_nombre_postal_value text NOT NULL,  
  field_felicitation_postal_value text NOT NULL,  
  field_visitante_postal_nid integer DEFAULT 0 NOT NULL,  
  CONSTRAINT content_type_postal_nid_key UNIQUE ("nid"),  
  CONSTRAINT content_type_postal_pkey PRIMARY KEY ("vid")  
)
```

En el ejemplo anterior se definió una restricción de unicidad, la cual se muestra a continuación:

```
CONSTRAINT content_type_postal_nid_key UNIQUE ("nid")
```

Para ello se usó la cláusula **constraint** para definir una restricción sobre el campo nid que asegurará la unicidad de este valor mediante la instrucción **unique**. Esta regla definida se podrá eliminar en cualquier momento sin tener que borrar la columna nid.

Restricción de valor no nulo:

Establece que algunos atributos deben contener valores en todo momento, es decir, no admiten nulos. En el ejemplo anterior se muestra una regla de este tipo sobre el campo tipo de felicitación de la tabla postal para evidenciar que una postal debe tener en todo momento el tipo de felicitación que se envió. A continuación se muestra la secuencia completa de la restricción.

```
field_felicitation_postal_value text NOT NULL
```

Al poner la cláusula **NOT NULL** como se muestra en el ejemplo se asegura que el campo no puede contener valores nulos.

Restricciones de clave primaria:

En general un esquema puede tener más de una clave. En tal caso, cada una de ellas se denomina clave candidata. Es común designar una de las claves candidatas como clave primaria de la relación o simplemente llave de la relación. Una clave primaria de una relación es un conjunto de atributos de su esquema que son elegidos para servir de identificador unívoco de sus tuplas. Para eso debe cumplir algunas condiciones como son: debe ser única, es decir; su valor no puede repetirse en la tabla, no puede contener valores nulos y debe ser minimal; no debe tener ningún subconjunto propio que pueda ser a su vez clave primaria.

Restricciones de dominio:

Todos los atributos tienen un dominio asociado, que es el conjunto de valores que cada atributo puede tomar. Con esta restricción se limitan los valores que puede tomar un atributo respecto a su dominio. Se pueden definir dominios en cuanto a los valores posibles que los datos pueden tomar o al tipo de datos que pueden ser. Por ejemplo, en el caso de la BD UCI_DRI que es la que se analiza en esta investigación se han definido las tablas tomando decisiones sobre qué atributos incluir.

Por ejemplo, los valores del campo nombre del regalo (field_nombre_regalo_value) de la tabla Regalo serán cadenas de caracteres, mientras que el campo importe por unidad (field_importe_regalo_value) almacenará datos de tipo float.

Con la asignación de tipos de datos, además seleccionamos un dominio para un atributo. El dominio del campo nombre del regalo, es el conjunto de todas las cadenas de caracteres de longitud ≤ 20 (este es el tipo SQL para cadenas de caracteres de longitud ≤ 20), mientras el dominio de importe por unidad es el conjunto de todos los números float. Ejemplo:

```
CREATE TABLE content_type_regalo (
```

```
vid int_unsigned NOT NULL,
nid int_unsigned NOT NULL,
field_nombre_regalo_value varchar(20) NOT NULL,
field_importe_regalo_value float NOT NULL,
field_loter_regalo_nid integer NOT NULL,
field_cantidad_regalo_value integer NOT NULL,
CONSTRAINT "content_type_salon_vip_pkey" PRIMARY KEY ("vid")
)
```

Restricciones de integridad de entidades:

Establece que ningún valor de clave primaria puede ser nulo y siempre deberá ser única. Un ejemplo de las buenas prácticas lo constituye la elección del campo de la tabla que será o formará parte de la llave primaria, partiendo de la obligación de que todos los valores del atributo son únicos y nunca nulos (**NULL**).

Restricciones de integridad referencial:

Esta restricción se especifica entre dos relaciones y sirve para mantener la consistencia entre tuplas de las dos relaciones. Establece que una tupla en una relación que haga referencia a otra relación deberá referirse a una tupla existente en esa relación.

La integridad referencial dice que si una clave ajena tiene un valor, si es no nula, ese valor debe ser uno de los valores de la clave primaria a la que referencia. Hay varios aspectos a tener en cuenta sobre las claves ajenas para lograr que se cumpla la integridad referencial.

1. ¿Admite nulos la clave ajena? Cada clave ajena expresa una relación. Si la participación de la entidad hijo en la relación es total, entonces la clave ajena no admite nulos; si es parcial, la clave ajena debe aceptar nulos.

2. ¿Qué hacer cuando se quiere borrar una ocurrencia de la entidad padre que tiene algún hijo? O lo que es lo mismo, ¿qué hacer cuando se quiere borrar una tupla que está siendo referenciada por otra tupla a través de una clave ajena? Hay varias respuestas posibles:

Restringir: no se pueden borrar tuplas que están siendo referenciadas por otras tuplas.

Propagar: se borra la tupla deseada y se propaga el borrado a todas las tuplas que le hacen referencia.

Anular: se borra la tupla deseada y todas las referencias que tenía se ponen, automáticamente, a nulo (esta respuesta sólo es válida si la clave ajena acepta nulos).

Valor por defecto: se borra la tupla deseada y todas las referencias toman, automáticamente, el valor por defecto (esta respuesta sólo es válida si se ha especificado un valor por defecto para la clave ajena).

No comprobar: se borra la tupla deseada y no se hace nada para garantizar que se sigue cumpliendo la integridad referencial.

3.2.2 Normalización de la base de datos

La normalización es una técnica para refinar la estructura lógica de los datos de un sistema de información en el modelo relacional, desarrollada por E. F. Codd en 1972. Es una estrategia de diseño de abajo a arriba: se parte de los atributos y estos se van agrupando en relaciones (tablas) según su afinidad. La normalización se debe utilizar en una etapa posterior a la traducción del esquema conceptual al esquema lógico, de forma tal que permita eliminar las dependencias entre atributos no deseadas. Las ventajas de la normalización son las siguientes:

- ✓ Evita anomalías en inserciones, modificaciones y borrados.
- ✓ Mejora la independencia de datos.
- ✓ No establece restricciones artificiales en la estructura de los datos.
- ✓ Reduce o elimina el almacenaje de datos duplicados

Uno de los conceptos fundamentales en la normalización es el de dependencia funcional. Una dependencia funcional es una relación entre atributos de una misma relación.

La dependencia funcional es una noción semántica. Si hay o no dependencias funcionales entre atributos no lo determina una serie abstracta de reglas, sino, más bien, los modelos mentales del usuario y las reglas de negocio de la organización o empresa para la que se desarrolla el sistema de información. Cada dependencia funcional es una clase especial de regla de integridad y representa una relación de uno a muchos. (Merche Marques abril 2004 Introducción a las bases de datos)

En el proceso de normalización se debe ir comprobando que cada tabla cumple una serie de reglas que se basan en la clave primaria y las dependencias funcionales de los demás atributos o campos de la tabla. Cada regla que se cumple aumenta el grado de normalización. Si una regla no se cumple, la relación se debe descomponer en varias relaciones que si la cumplan. La normalización se lleva a cabo en una serie de pasos. Cada paso corresponde a una forma normal que tiene propiedades y conforme se va avanzando, las relaciones tienen un formato más estricto, más fuerte y, por lo tanto, son menos vulnerables a las anomalías de actualización.

El modelo relacional solo requiere un conjunto de relaciones en primera forma normal. Las restantes formas normales son opcionales. Sin embargo, para evitar las anomalías de actualización, es recomendable llegar al menos a la tercera forma normal. Tampoco es siempre una buena idea tener una BD conformada en el nivel más alto de normalización, puede llevar a un nivel de complejidad que pudiera ser evitado si estuviera en un nivel más bajo de normalización.

(Date C. J. Introducción a los sistemas de Bases de Datos, tercera parte. La Habana, Editorial Félix Varela, 2003.)

Existen varios niveles de normalización:

- Primera Forma Normal (1NF)
- Segunda Forma Normal (2NF)
- Tercera Forma Normal (3NF)
- 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.
- Forma Normal de Clave de Dominio.

Primera forma normal (1FN):

La primera forma normal esta definida para prohibir los atributos multivaluados, los atributos compuestos y sus combinaciones. Cuando un esquema de relación no está en primera forma normal hay que eliminar de ella los grupos repetitivos. Un grupo repetitivo seria el atributo o grupo de atributos que tiene múltiples valores para cada tupla de la relación. Hay dos formas de eliminar los grupos repetitivos. En la primera, se repiten los atributos con un solo valor para cada valor del grupo repetitivo. De este modo, se introducen redundancias ya que se duplican valores, pero estas redundancias se eliminarían después mediante las restantes formas normales. La segunda forma de eliminar los grupos repetitivos consiste en poner cada uno de ellos en una relación aparte, heredando la clave primaria de la relación en la que se encontraban.

Segunda forma normal (2FN):

Una relación está en segunda forma normal si, y solo si, está en 1FN y, además, cada atributo que no está en la clave primaria es completamente dependiente de la clave primaria.

La segunda forma normal (2FN) sólo se aplica a los esquemas de relación que tienen claves primarias compuestas por dos o más atributos. Si un esquema de relación está en primera forma normal (1FN) y su clave primaria es simple entonces está en segunda forma normal (2FN). Las relaciones que no están en 2FN pueden sufrir anomalías cuando se realizan actualizaciones.

Para la transformación de una relación de 1FN a 2FN hay que eliminar las dependencias parciales de su clave primaria. Para ello, se eliminan los atributos que son funcionalmente dependientes y se ponen en una nueva relación con una copia de los atributos de la clave primaria de los que dependen.

Tercera forma normal (3FN):

Un esquema de relación está en tercera forma normal (3FN) si, y sólo si, está en segunda forma normal (2FN) y, además cada atributo del esquema de relación que no está en la clave primaria sólo depende funcionalmente de la clave primaria, y no de ningún otro atributo.

Aunque las relaciones en 2FN tienen menos redundancias que las relaciones en 1FN, todavía pueden sufrir anomalías frente a las actualizaciones. Para pasar una relación de 2FN a 3FN hay que eliminar las dependencias transitivas. Para ello, se eliminan los atributos que dependen transitivamente y se ponen en una nueva relación con una copia del atributo o atributos no clave de los que dependen.

Forma normal de Boyce–Codd (BCFN):

Una relación está en la forma normal de Boyce–Codd si, y solo si, está en tercera forma normal (3FN) y todo determinante es una clave candidata. La BCFN es más fuerte que la 3FN, por lo tanto, toda relación en BCFN está en 3FN.

La violación de la BCFN es poco frecuente ya que se da bajo ciertas condiciones que raramente se presentan. Se debe comprobar si una relación viola la BCFN si tiene dos o más claves candidatas compuestas que tienen al menos un atributo en común. Esta forma normal es una de las formas normales más deseables para un buen diseño de una base de datos.

Teniendo en cuenta los puntos analizados hasta aquí acerca del proceso de normalización se puede decir que el diseño propuesto en este trabajo para la base de datos de SICI está en la forma normal de Boyce-Codd ya que cumple rigurosamente con todas las formas normales hasta llegar a ella, por ejemplo: no presenta campos compuestos ni multivaluados. Las tablas que presentaban campos con esa situación se les transformaron del mismo modo de la siguiente manera: los atributos o campos compuestos como es el caso del atributo dirección particular de la tabla persona se descompuso en una nueva relación formada por sus atributos simples como son: calle, número de calle y entre calle entre otros.

Con los atributos multivaluados que estaban presentes en la BD se les descompuso es relaciones individuales para cada uno siguiendo los siguientes criterios: Para los atributos multivaluados con un dominio corto y conocido (se cita dominio refiriéndose a los posibles valores que puede tener no al tipo de datos), se pusieron cada uno de ellos en una relación aparte, heredando la clave primaria de la relación en la que se encontraban definiéndolos como nomencladores. Para los atributos multivaluados con un dominio amplio y desconocido, es decir que no se conozcan sus posibles valores, se les separo en tablas individuales las cuales tienen como llave primaria el identificador de la entidad a la cual el atributo pertenecía y el identificador del atributo multivaluado.

Ejemplo de transformación de atributos multivaluados:

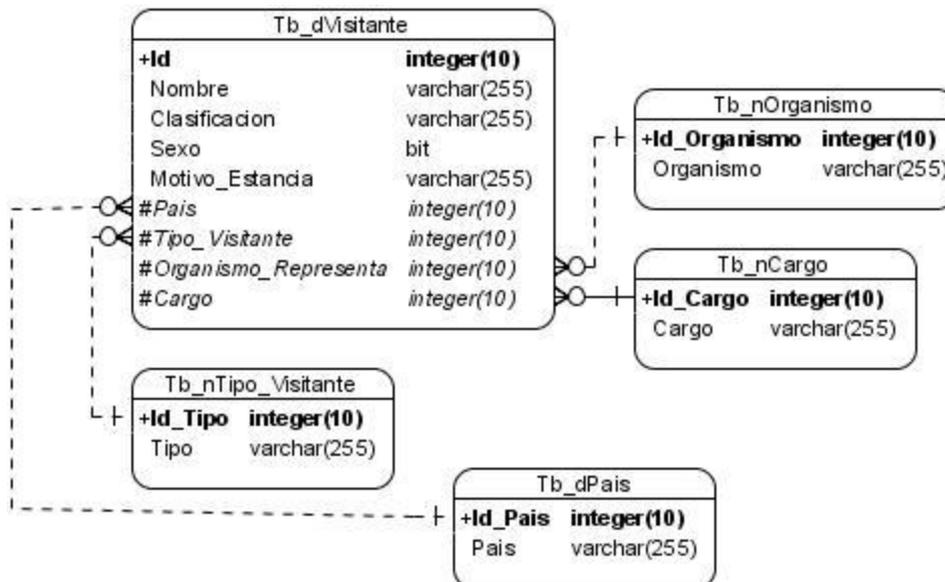


Figura 3. 1 Ejemplo de transformación de atributos multivaluados

Definición de nomencladores para atributos multivaluados con valores de un dominio corto y conocido.

Ejemplo de transformación de atributos multivaluados en modelo relacional:

EQUIPO (Cod, nombre, color)

EQUIPO (Cod, nombre)

EQUIPO_COLOR (Cod, Color)

En el ejemplo se puede observar que el primer esquema no está en 1FN al contrario del segundo esquema.

Además de cumplir con la 1FN la base de datos que se analiza no presenta dependencias parciales cumpliendo así con la 2FN. Tampoco presenta dependencias transitivas por lo que puede asegurarse que su diseño cumple con la 3FN. Para llevar el diseño a FNBC se analizaron las dependencias funcionales en todas las relaciones en las cuales no había solapamiento entre las llaves candidatas por lo que se puede asegurar que el diseño cumple con la forma normal de Boyce-Codd.

3.2.3 Análisis de redundancia de información

La redundancia de información se presenta debido a la duplicación de información en las bases de datos. Cuando se almacena información repetida generalmente se sufren de anomalías de inserción, actualización y eliminación por lo cual el diseño realizado pierde efectividad. Además esta información repetida supone mayor espacio en disco, y una disminución en el rendimiento de la BD reflejándose en los costos de ejecución de las consultas o peticiones. La normalización es un paso importante hacia el proceso de eliminación de la redundancia y el logro de una mayor integridad y consistencia en la información que se almacena.

Introducción de redundancias controladas:

En ocasiones puede ser conveniente relajar las reglas de normalización introduciendo redundancias de forma controlada, con objeto de mejorar las prestaciones del sistema. En la etapa del diseño lógico se recomienda llegar, al menos, hasta la tercera forma normal para obtener un esquema con una estructura consistente y sin redundancias. Pero, a menudo, sucede que las bases de datos así normalizadas no proporcionan la máxima eficiencia, con lo que es necesario volver atrás y desnormalizar algunas relaciones, sacrificando los beneficios de la normalización para mejorar las prestaciones.

Es importante hacer notar que la desnormalización solo debe realizarse cuando se estime que el sistema no puede alcanzar las prestaciones deseadas. Y, desde luego, la necesidad de desnormalizar en ocasiones no implica eliminar la normalización del diseño lógico: la normalización obliga al diseñador a entender completamente cada uno de los atributos que se han de representar en la base de datos.

3.2.4 Análisis de la seguridad de la base de datos

En la DCI se maneja mucha información sensible y de gran importancia para la universidad y el país. Es por lo tanto una necesidad crucial asegurar la protección de los datos que se van a almacenar contra lecturas no autorizadas, cambios, destrucción o cualquier acción que altere la integridad de los datos.

Para llevar a cabo el arduo trabajo de asegurar la seguridad de la base de datos, se analizaron varias posibilidades y se aplicaron en la medida de las necesidades. Algunas de ellas son:

Protección mediante la aplicación:

Mediante el CMS Drupal, con el cual se esta desarrollando la aplicación se han definido los usuarios y roles con sus respectivos privilegios de acceso a distintos niveles para acceder a la aplicación y por inferencia a la información almacenada. Este administrador de contenido dispone de distintos módulos que le permiten asegurar el acceso a la base de datos mediante diversos criterios. Inicialmente todo usuario debe estar registrado y correctamente autenticado para poder tener una vista de los datos que desee ver. Para lograr este importante aspecto se utiliza un módulo que permite la integración con servidores LDAP lo cual permite que los usuarios del dominio UCI.CU puedan acceder a la aplicación con su usuario del dominio.

Algunos módulos de Drupal que garantizan niveles de acceso y seguridad son:

Modulo Paranoia:

Deshabilita la posibilidad de inclusión de código php en drupal, evitar que se pueda eliminar la edición de usuarios mediante cambios en la aplicación.

Modulo Taxonomy Access Control:

Control de acceso a la taxonomía del sitio mediante el uso de roles

Modulo Captcha:

Verifica que una persona y no una máquina traten de crear cuentas desde una dirección IP específica.

Protección mediante el SGBD:

PostgreSQL esta provisto de excelentes mecanismos de seguridad. Primeramente dispone de un Gestor de Usuario (User Manager) para añadir, eliminar usuarios y editar los privilegios globales de dichos usuarios. También cuenta con el Gestor de Privilegios (Grant Manager) para el otorgamiento de privilegios de bases de datos seleccionadas, tablas y atributos.

Los privilegios globales de usuario definen los derechos de acceso de los usuarios a todas las tablas de la base de datos. Los privilegios de otorgamiento de las bases de datos seleccionadas, tablas o

atributos permiten al usuario la ejecución de operaciones determinadas (select, update, insert) sobre todas las tablas de la base de datos seleccionada, sobre algunas tablas seleccionadas o sobre los atributos seleccionados solamente.

Añadir un nuevo usuario mediante el gestor de usuario de PostgreSQL:

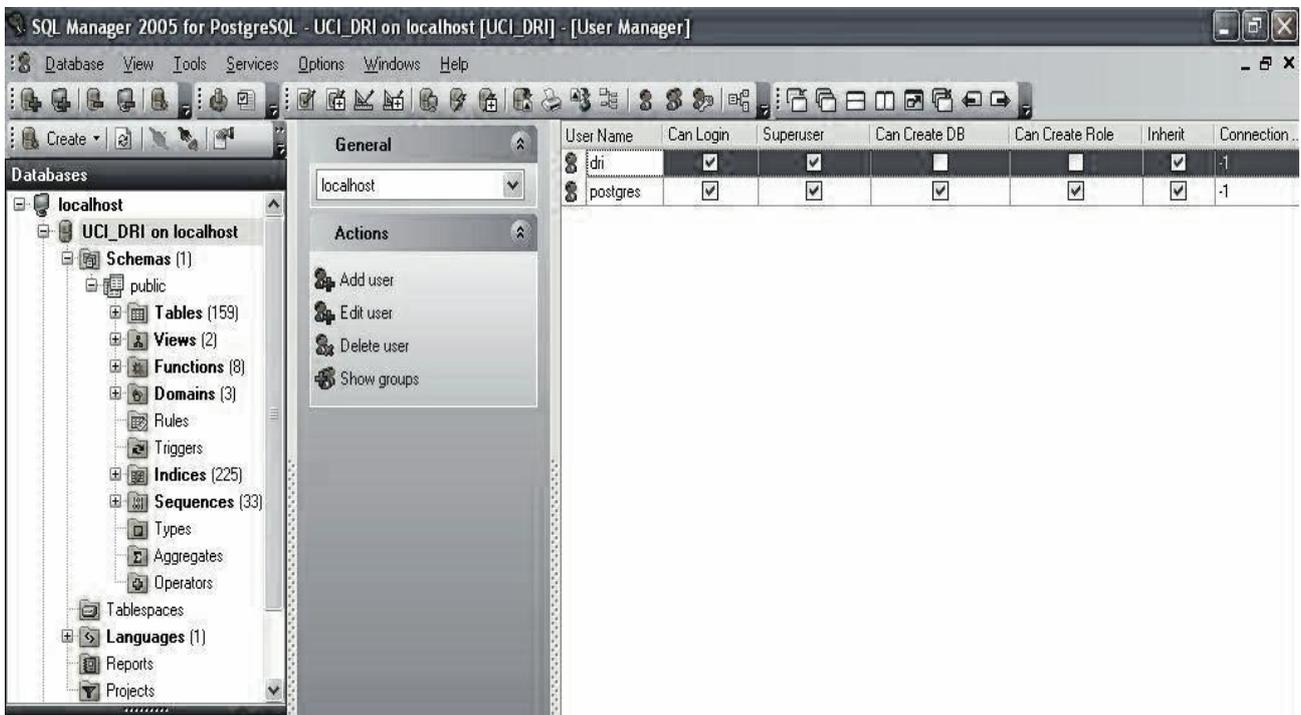


Figura 3. 2 Añadir un nuevo usuario mediante el gestor de usuario de PostgreSQL

En la BD se realizaron además configuraciones para controlar los permisos de los usuarios, utilizando tres niveles de acceso.

En un primer nivel (nivel 0) se configuraron los permisos de conexión para los host y los usuarios a la BD, datos que se recogen en el archivo `pg_hba.conf` en el cual se definen que computadoras (dirección o direcciones IP) tendrán acceso, y el modo en que podrán conectarse, que puede ser: conexión sin contraseña, validando el usuario y la contraseña para conectarse, o que rechace cualquier conexión desde el IP o rangos IP y usuarios seleccionados.

Como método de acceso se selecciono el md5 el cual es un algoritmo de reducción criptográfica que permite encriptar las contraseñas almacenadas de los usuarios.

Ejemplo de configuración de acceso solo de forma local encriptadas en el archivo pg_hba.conf:

```
# TYPE DATABASE  USER    CIDR-ADDRESS  METHOD
# "local" is for UNIX domain socket connections only
local  all      all                md5
# IPv4 local connections:
hostssl all      all  127.0.0.1/32    md5
# IPv6 local connections:
hostssl all      all  ::1/128         md5
```

Ejemplo de encriptación mediante md5:

```
INSERT INTO public.content_type_presente VALUES (1, 2, md5 ('hola') ,3)
```

Al hacer una consulta seleccionando los datos de la tabla content_type_presente se obtendrán los datos:

vid	nid	field_resp_presente_value	field_dest_presente_nid
1	2	4d186321c1a7f0f354b297e8914ab240	3

Nótese el valor del campo field_resp_presente_value encriptado.

En un segundo nivel (nivel 1), se configuró PostgreSQL para permitir el acceso de determinados usuarios, utilizando las opciones del archivo pg_ident.conf.

Por último existe un tercer nivel (nivel 2), que permite configurar dentro de las BD los accesos a las tablas, utilizando los comandos GRANT y REVOKE para permitir o denegar los permisos, respectivamente. Con estos comandos se definieron los permisos por usuarios y grupos de usuarios a la BD.

3.2.5 Trazabilidad de las acciones

La trazabilidad es la "capacidad para rastrear los antecedentes, la aplicación o la ubicación de una entidad por medio de identificaciones registradas." También se puede expresar como la capacidad para reconstruir la historia, recorrido o aplicación de un determinado producto.

Aplicando este concepto al ámbito de bases de datos, se puede hablar de la trazabilidad de las acciones que se realizan en una base de datos. Siempre se tiene que tener un control de las transacciones que se realizan en todo momento sobre usuarios, tablas o incluso hasta en errores o conflictos. Para ello el gestor de bases de datos utilizado para construir la base de datos para SICI, dispone de herramientas para el manejo de los logs que son archivos donde se guarda el historial de las operaciones realizadas a la base de datos. En los archivos logs de PostgreSQL se guardan los datos como la fecha, hora y acción realizada en las base de datos.

Específicamente los logs son mecanismos que tienen los SGBD para seguir las trazas en bases de datos. Se usan para registrar información sobre (quién, qué, cuándo, dónde) convirtiéndose en un búsqueda oficial de eventos en cualquier momento.

3.3 Validación funcional

3.3.1 Generación de código de programas para un llenado voluminoso e inteligente de la base de datos

Para el llenado automático de la base de datos con el objetivo de realizar pruebas en diseño y análisis de rendimiento se utilizaron varias herramientas, entre ellas está el PgDataGen y el EMS Data Generator For PostgreSQL 2005 las cuales tienen una buena integración con PostgreSQL y permiten generar hasta 10 000 valores por columna, generan datos para una o varias tablas a la vez, definiendo para cada campo el rango de valores admisibles, dependiendo del tipo, además de la cantidad de tuplas que se desean generar, validando de forma automática la integridad referencial.

Se debe mencionar además la migración de los datos desde SQL Server 2000 hacia PostgreSQL ya que la DCI contaba con una BD en este SGBD para guardar la información vital para su funcionamiento. En la realización de la migración se utilizó la herramienta Data Pump 2006 con la cual se exportaron los datos entre los dos sistemas gestores de bases de datos.

Selección de algunas consultas:

```
SELECT count (public.content_type_persona.vid) as cantidad misioneros
```

```
FROM public.content_type_persona JOIN public.content_type_misionero
```

```
ON (public.content_type_persona.nid = public.content_type_misionero.field_persona_misionero_nid)
```

```
JOIN public.content_type_mision
```

ON (public.content_type_mision.nid = public.content_type_misionero.field_mision_misionero_nid)

Esta consulta muestra la cantidad de personas que son misioneros.

```
SELECT content_type_persona.nid
FROM content_type_persona
WHERE content_type_persona.nid IN
(
SELECT content_type_misionero.field_persona_misionero_nid
FROM content_type_misionero
WHERE content_type_misionero.field_mision_misionero_nid = 3050
)
```

Mostrar los identificadores de todas las personas en una misión dada.

Para las pruebas realizadas se analizaron aspectos como el rendimiento de la base de datos ante diferentes transacciones. Se analizó también la optimización de consultas en especial los costos de ejecución de las mismas. Se le prestó especial interés al diseño de las transacciones de recuperación, de actualización y mixtas que tienen como base las especificaciones de requerimientos de los usuarios. El objetivo de este aspecto es validar el diseño local para garantizar que pueda soportar las transacciones requeridas por los correspondientes usuarios.

3.4 Conclusiones

En este capítulo se analizaron puntos esenciales para el correcto diseño de bases de datos. Se examinaron diferentes consideraciones como la integridad referencial para la cual se mostró como se cumplen en el diseño las reglas establecidas, se expone el proceso de normalización realizado, además de los mecanismos de seguridad establecidos y el análisis de la redundancia de la información. Todos estos aspectos en su conjunto aseguran el control de las operaciones que se realizan en la nueva base de datos para la DCI.

CONCLUSIONES

La tecnología de bases de datos ha sido, sin lugar a duda, uno de los desarrollos claves en la evolución de la informática ya que ha aportado la capacidad de integrar la información permitiendo que múltiples usuarios y aplicaciones puedan compartirla en un ambiente que garantiza su seguridad e integridad.

Sin embargo, el completo aprovechamiento de las posibilidades que ofrece esta tecnología requiere un correcto diseño de las estructuras de datos que conformarán el contenido de las bases de datos. De ahí se desprende la importancia de este trabajo, pues el correcto diseño de la base de datos, que es en esencia el objetivo de esta investigación, logrará satisfacer en buena medida las necesidades reales y latentes de almacenamiento, disponibilidad, integridad y seguridad de la información de la Dirección de Cooperación Internacional de la UCI.

Los resultados obtenidos demuestran que se alcanzan ganancias significativas por parte de la Universidad y en particular por la DCI en cuanto a organización de la información, mejor rendimiento en lo que a rapidez en la gestión de los procesos se refiere, control de la información y al aseguramiento de funcionalidades.

En este trabajo se tuvieron en cuenta aspectos como:

- Las tendencias actuales de las bases de datos a nivel internacional, nacional y de la UCI.
- Metodologías de desarrollo de software y de diseño de bases de datos.
- Conceptos y definiciones de BD, SGBD así como sus arquitecturas ventajas y desventajas.
- Herramientas utilizadas para el diseño de bases de datos.
- Descripción de las clases persistentes y modelo entidad relación de la base de datos
- Optimización de consultas
- Validación teórica y funcional de la base de datos.

Como resultado final se obtuvo un diseño de base de datos que refleja los requerimientos del usuario y sus necesidades reales. Esta base de datos cuyo diseño se ha expuesto en este trabajo esta destinada a la construcción de un sistema informatizado que gestione todas las tareas que se llevan a cabo en la Dirección de Cooperación Internacional.

RECOMENDACIONES

Elaborar una apropiada estrategia de mantenimiento y seguridad a la base de datos asegurando la realización periódica de copias de seguridad con el fin de mantener la fiabilidad, rendimiento y funcionamiento adecuado de la misma.

Extender el diseño de la base de datos para otros subsistemas del proyecto SICI como el de Salida por proyecto y Logística que aun se encuentran en la fase de modelamiento del negocio.

Se recomienda la ejecución del comando **vacuum** mensualmente para mantener las estadísticas relativamente actualizadas. Particularmente porque después de insertar, eliminar o actualizar grandes volúmenes de información como los que se manejan en la DCI el SGBD registra todas las transacciones realizadas almacenando cada número de tupla y de tabla lo cual provoca un pobre rendimiento. El comando **vacuum** limpia los registros de transacciones ya pasadas y actualiza las estadísticas en los catálogos del sistema.

BIBLIOGRAFÍA REFERENCIADA

- BOOCH, G. R., J. Y JACOBSON, I. . “*El Lenguaje Unificado de Modelado*”. 2000. p.
- C. BATINI, S. C., S.B. NAVATHE. *Diseño conceptual de Bases de Datos. Un enfoque de entidades-interrelaciones.*, 1994. p.
- CLIKEAR.COM. *Desarrollo Orientado a Objetos con UML*, 2004. [Disponible en: <http://www.clikear.com/manuales/uml/>]
- DATE, C. J. *Introducción a los sistemas de bases de datos*. La Habana, Félix Varela, 2003. p.
- DÍAZ, I. V. R., 2008. [Disponible en: <http://www.sld.cu/mainsearch.php?q=bases+de+datos&d=2>]
- ELMASRI, N. *Conceptos y arquitectura de un sistema de bases de datos*. 2002. p.
- ELMASRI/NAVATHE. *Bases de datos y usuarios de bases de datos*. 2002. p.
- GARCÍA, L. R. M. M. *Diseño de Base de Datos*. 1999. p.
- GARCÍA, R. M. M. *Sistemas de Bases de Datos*
- GROUP, P. G. D. *Cases Studies*, 2008. [Disponible en: <http://www.postgresql.org/about/users>]
- MARQUÉS, M. *Introducción a las Bases de Datos*. 2004. p.
- ZONE, S. D. *PostgreSQL at Skype*, 2008. [Disponible en: <https://developer.skype.com/SkypeGarage/DbProjects/SkypePostgresqlWhitepaper>]

BIBLIOGRAFÍA CONSULTADA

- AMESCUA, A. *Diseño de Bases de Datos. Problemas Resueltos*. 2000. p.
- ADORACIÓN DE MIGUEL CASTAÑO, M. G. P. V., ESPERANZA MARCOS MARTÍNEZ. *Diseño de Bases de Datos Relacionales*. 2000. p.
- HERMAN E. DOLDER, E. L. *Sistema Experto para el Diseño de Bases de Datos*, 1988.
- WILLIAM BRAVO CABALLERO, R. Á. T. *Diseño de la base de datos para el módulo de IPC de la Oficina Nacional de Estadísticas*, 2007. p.
- YENEIRYS HERNÁNDEZ CASTELLANOS, W. P. G. *Diseño de Bases de Datos para la Intranet 2*, 2007. p.
- DATE C. J. *Introducción a los sistemas de bases de datos*, Capítulos 1, 2, 13, Editorial Félix Varela, La Habana, 2003.
- ECHARRI, P. A. M. V. R. *La metodología de la Investigación Educativa para la Formación del Profesional*. 2002. p.
- PUJOL, J. C. *Reglas de transformación del Diagrama ERE al Modelo Relacional*. 2007. p.

GLOSARIO DE TÉRMINOS

- UCI: Universidad de las Ciencias Informáticas
- DCI: Dirección de Cooperación Internacional
- ID: Identificador
- MINVEC: Ministerio de la Inversión extranjera y la Colaboración
- MIC: Ministerio de la Informática y las Comunicaciones.
- Misión: Actividad gestionada por el Grupo de Cooperación Internacional. Se puede tratar de un viaje al extranjero de una o más personas por intereses de la Universidad, la firma de un documento oficial o la atención a un visitante.
- MINREX: Ministerio de Relaciones Exteriores.
- MINVEC: Ministerio de la Inversión extranjera y la Colaboración.
- RRPP: Relaciones Públicas.
- Viajero: Persona que viaja al extranjero.
- VIP: Very Important Person.
- Reunión de Coordinación: Reunión que se desarrolla cada lunes, donde participan todos los factores (Vicerrectores, Decanos, PCC, UJC, FEU y Sindicato), para dar la propuesta de programa de atención de la visita.
- Informe de la visita: Documento elaborado por un especialista de RRPP o de la DCI. Se hace una explicación de forma general del desarrollo de la visita, se detallan las preguntas formuladas por los visitantes, se escribe textualmente lo que escriben en el libro de visitantes; en el caso de otros idiomas se hace la traducción y se escribe la hora de finalizada la vista
- Comprobante de salida: Comprobante de salida de documentos ordinarios, este debe ser firmado por el oficial que recibe en el CE y envía la copia para tener constancia de la entrega, en el se especifica la fecha de entrega.
- Programa de Atención a visita: Documento que se le entrega al visitante, proponiéndole las actividades a realizar en su estancia.
- Carta de solicitud de Gastos: Documento que se le presenta al Rector donde se le informa los posibles gastos del programa de atención a visitas, para obtener su aprobación.
- Visitantes: Personas que participan en determinadas actividades educativas o de producción relacionadas con la Universidad, pueden ser visitantes extranjeros o cubanos.
- Solicitud a Protocolo: Documento que se realiza para el pedido de algún alimento a Protocolo.
- Solicitante: Organismo o persona que realiza la solicitud de una visita a la Universidad

- Base de Datos Relacional.- Tipo de base de datos o sistema de administración de bases de datos, que almacena información en tablas (filas y columnas de datos) y realiza búsquedas utilizando los datos de columnas especificadas de una tabla para encontrar datos adicionales en otra tabla.
- DBMS: Data Base Management System (SISTEMA DE MANEJO DE BASE DE DATOS).- Consiste de una base de datos y un conjunto de aplicaciones (programas) para tener acceso a ellos.
- Modelo de Datos.- es un conjunto de herramientas conceptuales para describir los datos, las relaciones entre ellos, su semántica y sus limitantes.
- Redundancia.- Esta se presenta cuando se repiten innecesariamente datos en los archivos que conforman la base de datos.
- Aislamiento de los Datos.- Se refiere a la dificultad de extender las aplicaciones que permitan controlar a la base de datos, como pueden ser, nuevos reportes, utilerías y demás debido a la diferencia de formatos en los archivos almacenados.
- Anomalías en el Acceso Concurrente.- Ocurre cuando el sistema es multiusuario y no se establecen los controles adecuados para sincronizar los procesos que afectan a la base de datos. Comúnmente se refiere a la poca o nula efectividad de los procedimientos de bloqueo.