

Universidad de las Ciencias Informáticas
Centro de Informatización Universitaria
CENIA



Base de datos para el Sistema de
Gestión de Investigaciones

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

Autor:

Adriel Hernández Lugo

Tutores:

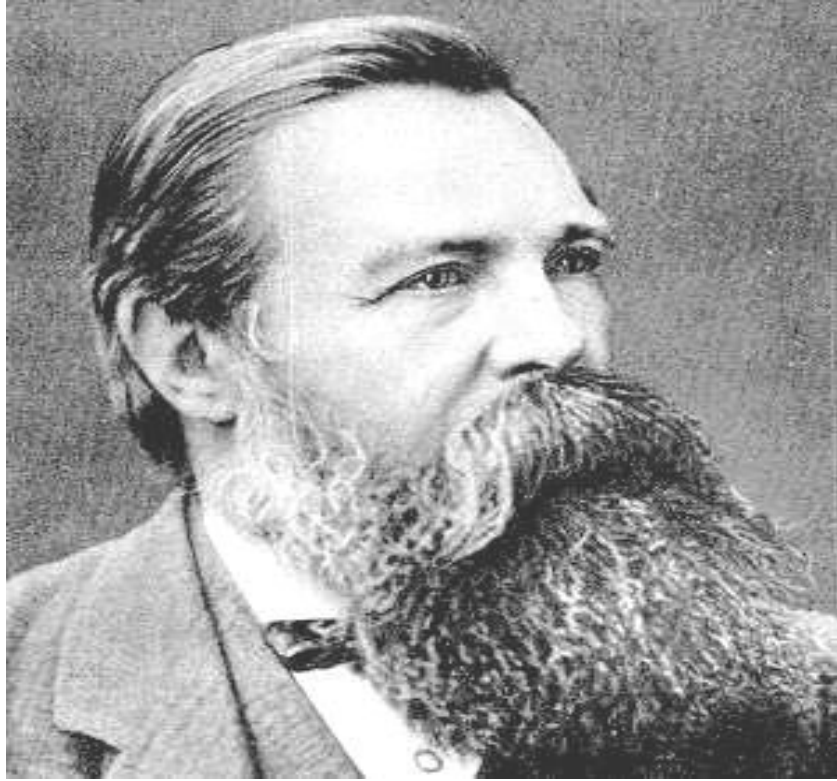
Ing. Yoan Carlos Machado Espinosa

Ing. Roberkis Terrero Galano

Ing. Irina Fuentes Viñas

La Habana, junio de 2012

“Año 54 de la Revolución”



"Una onza de acción es el valor de una tonelada de teoría."

Federico Engels.



DECLARACIÓN DE AUTORÍA

DECLARACIÓN DE AUTORÍA

Me declaro ser autor de la presente tesis y reconozco a la Universidad de Ciencias Informáticas los derechos patrimoniales de la misma con carácter no exclusivo.

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

Adriel Hernández Lugo
Autor

Ing. Roberkis Terrero Galano
Tutor

Ing. Yoan Carlos Machado Espinosa
Tutor

Ing. Irina Fuentes Viñas
Tutor



AGRADECIMIENTOS

A mi padre (el puro), por ser el motor impulsor en toda mi vida estudiantil y haber sido en ocasiones el único motivo por el cual nunca me di por vencido.

A mi mamá por ser la mejor del mundo dando cariño y mostrarme siempre los caminos que conducen a lo bien hecho y la sabiduría del alma.

A mis hermanos, en especial al mayor, Arley, por ser el faro que me guió en toda la carrera.

A mi primo David y su novia Lily, a Jorgito, a mis primas lindas de Alamar Dianet y Dayana, también a Alexis, Chago, a mi tía Carmela y al resto de mi bella familia.

A mi Honey Pot, por darme tanto amor y ser lo más bello que hay sobre la tierra para mi.

A mis amistades Eider, Kiki, Reinier, Ernesto, Orestes, Montano, a Geandys que duró poco pero marcó la diferencia, a todos en fin.



DEDICATORIA

Dedicar todo mis logros en este trabajo a mis padres y hermanos, a mi abuelita Teresa, por haberme mostrado como ser mejor persona en la vida, y lograr que yo diera el paso adelante cuando la vida me daba mil razones para que fuera hacia atrás.

A mis amistades y familiares de las afueras del país, por todo su cariño y apoyo incondicional.

A todas mis amistades que me han dado su confianza y colaboraron de una forma u otra en la realización de este sueño.



El Centro de Informatización Universitaria (CENIA) surge como resultado del desarrollo de la infraestructura productiva de la Universidad de las Ciencias Informáticas. El trabajo de este centro y sus principales actividades, se enfocan en hallar una solución de manera integral para la gestión de los procesos universitarios. Este centro desarrolla entre sus proyectos el Sistema de Gestión de Investigaciones, el cual informatiza los procesos investigativos de los que se encarga la actual Dirección de Investigaciones. Dichos procesos generan cierta variedad de datos, ya sea a la hora de gestionar eventos, premios, así como líneas y proyectos de investigación, por lo que se necesita una base de datos para el manejo de la información que se trata en el área de las investigaciones, como solución a esta problemática se desarrolla la base de datos del Sistema de Gestión de Investigaciones con el objetivo de garantizar el acceso concurrente, la integridad y seguridad en el acceso y salvaguarda de los datos relativos a los procesos investigativos. Durante la investigación, se analizaron los principales conceptos referentes a las bases de datos, los enfoques de los Sistemas Gestores de Bases de Datos (SGBD), así como su evolución y estructura. Se tiene en cuenta el proceso de normalización empleando una técnica de diseño de bases de datos, proceso resultante de una metodología de diseño previamente seleccionada, obteniéndose con esto el modelo de datos, el cual se crea después del estudio de los requisitos funcionales del sistema. También se implementan índices, llaves primarias, funciones y disparadores una vez creada la base de datos en el SGBD *PostgreSQL*. De igual manera se valida la integridad de los datos, la calidad del dato y el alto rendimiento de la solución propuesta.

PALABRAS CLAVE: base de datos, modelo de datos, normalización, Sistemas Gestores de Bases de Datos.



ÍNDICE DE CONTENIDO

Introducción	1
Capítulo 1. Fundamentación teórica.	7
1.1 Introducción.....	7
1.2 Bases de datos	7
1.3 Sistemas gestores de bases de datos	8
1.3.1 Objetivos principales de los SGBD.....	8
1.3.2 Evolución de los Sistemas Gestores de Bases de Datos	9
1.3.3 Enfoques de los Sistemas Gestores de Bases de Datos	13
1.3.4 Estructura de los Sistemas Gestores de Bases de Datos	15
1.3.4.1 Tipos de usuarios que interactúan con la BD	16
1.3.4.2 Lenguajes de BD que utilizan los usuarios	17
1.3.4.3 Lenguajes para el manejo de datos	17
1.3.4.4 Componentes de los SGBD.....	17
1.3.4.5 Estructura de datos.....	18
1.3.5 Ejemplos de Sistemas Gestores de Bases de Datos	18
1.3.5.1 Microsoft SQL Server	19
1.3.5.2 MySQL.....	20
1.3.5.3 Oracle.....	21
1.3.5.4 PostgreSQL.....	23
1.3.6 Comparación entre los diferentes SGBDR.	29
1.4 Herramientas de modelado.....	30
1.5 Conclusiones parciales	31
Capítulo 2. Diseño e implementación de la base de datos.	32



ÍNDICE DE CONTENIDO

2.1	Introducción.....	32
2.2	Diseño de la base de datos	32
2.3	Patrones de diseño de bases de datos	33
2.4	Metodología para el diseño de bases de datos	34
2.5	Técnicas para diseñar bases de datos relacionales.....	35
2.5.1	Normalización	36
2.6	Diagrama Entidad - Relación.....	38
2.6.1	Diagrama Entidad-Relación de la base de datos del Sistema de Gestión de Investigaciones.....	40
2.6.2	Descripción de las tablas.	40
2.7	Soporte arquitectónico	44
2.7.1	Seguridad y disponibilidad de los datos	44
2.7.2	Estándares de nomenclatura	45
2.7.3	Herramientas usadas.....	45
2.7.4	Tipos de datos	46
2.8	Implementación de la BD	48
2.8.1	Índices y llaves primarias	49
2.8.2	Funciones y procedimientos almacenados.....	50
2.8.3	Disparadores	56
2.9	Conclusiones parciales	56
Capítulo 3. Validación de la propuesta de solución.....		55
3.1	Introducción.....	55
3.2	Normalización de la base de datos	55



ÍNDICE DE CONTENIDO

3.3 Integridad de los datos.....	56
3.3.1 Integridad de entidades.....	57
3.3.2 Integridad referencial.....	59
3.3.3 Integridad de dominio.....	60
3.4 Calidad del dato.....	61
3.5 Alto rendimiento.....	62
3.6 Conclusiones parciales.....	64
Conclusiones generales.....	65
Recomendaciones.....	66
Referencias bibliográficas.....	67



ÍNDICE DE FIGURAS

Figura 1. Estructura de una base de datos jerárquica	13
Figura 2. Estructura de una base de datos de red.	14
Figura 3. Estructura de una base de datos relacional.	14
Figura 4. Estructura de un Sistema Gestor de Bases de Datos.....	16
Figura 5. Componentes de un sistema PostgreSQL.	24
Figura 6. Arquitectura para la integración.	53
Figura 7. Llamadas remotas entre bases de datos.	55
Figura 8. Error que se emite cuando se inserta una tupla que ya existe.	58
Figura 9. Ejemplo de error que se emite cuando se insertan valores nulos no permitidos.....	59
Figura 10. Ejemplo de error que se emite cuando se viola la integridad referencial.	60
Figura 11. Ejemplo de error que se emite cuando se viola la integridad de dominio.	61



ÍNDICE DE TABLAS

Tabla 1. Arquitectura a tres niveles.	12
Tabla 2. Comparación entre los Sistemas Gestores de Bases de Datos.....	29
Tabla 3. Tabla evento del módulo eventos.	41
Tabla 4. Tabla premio del módulo premios.....	42
Tabla 5. Tabla línea de investigación del módulo líneas y proyectos.	43
Tabla 6. Tabla proyecto de investigación del módulo líneas y proyectos.....	44
Tabla 7. Varias funciones de la base de datos.	52
Tabla 8. Ejemplo de utilización de la tecnología plproxy.	56



Introducción

La comunidad de la información no es más que el fruto de una revolución tecnológica sin antecedentes, evolucionando continuamente por el desarrollo de las telecomunicaciones, la informática y las comunicaciones, que ensanchan y hacen posible procesos socioculturales, políticos y económicos de la vida cotidiana en todo el mundo.

En nuestro país la informatización de la sociedad es el proceso de utilización ordenada y masiva de las Tecnologías de la Información y las Comunicaciones (TIC) para saciar las necesidades de información y conocimiento de toda la comunidad. Este proceso se lleva a cabo por medio de la universalización de la educación, así como la creación de los Joven Clubs con el objetivo de que la comunidad cubana tenga cierta facilidad a la hora de desarrollar sus conocimientos y lograr mayor interacción con los medios tecnológicos.

La Universidad de las Ciencias Informáticas (UCI) fue creada en el año 2002, este centro surge por la necesidad de informatizar y llevar la informática a los rincones más apartados del país, en aquel entonces fue uno de los programas más importantes de la Revolución, convirtiéndose en un grandioso centro para la formación de profesionales en esta ciencia. Con el devenir de los años en la UCI se han desarrollado una gran cantidad de proyectos productivos, en los cuales se vinculan principalmente la docencia y la producción, con el objetivo de obtener beneficios comerciales dentro y fuera del país, sin obviar la formación integral de los estudiantes y por otra parte un aumento de calidad en la mayoría de los procesos que se desarrollan en la UCI.

La universidad ha organizado el tema de la producción en diferentes perfiles productivos, por lo cual se han conformado diferentes centros, ejemplo de ello es el Centro de Informatización Universitaria (CENIA).



INTRODUCCIÓN

El CENIA es el resultado de la evolución continua de la infraestructura productiva en la UCI. Bajo la convicción de que las distintas facultades son el escenario de integración de los procesos de formación, desarrollo e investigaciones; teniendo como un importante pilar la fuerza resultante obtenida de la vinculación de estudiantes y profesores a la producción, se crea este centro, tomando como punto de partida la ya existente Dirección de Informatización. El trabajo del centro se enfoca en la informatización de la gestión universitaria y en el desarrollo y avance de la fructífera universidad digital. En la actualidad tiene como objetivo principal encontrar una solución de manera integral para la gestión de los procesos internos de la propia universidad, con proyectos que favorecen a las áreas de procesos de la formación académica de pregrado y postgrado, de las investigaciones, la cooperación, la gestión de residencia, entre otros.

Una de las actividades que se encarga de informatizar el centro es la gestión de los diferentes eventos investigativos, proceso importante que permite el control de manera integral de todas las actividades con carácter científico-investigativas dentro de la universidad, así como de aquellas con alcance nacional e internacional.

La actual Dirección de Investigaciones de la universidad realiza el proceso de gestión de las actividades investigativas de manera engorrosa, lo que acarrea que muchas veces la información correspondiente no llegue a toda la comunidad universitaria, se pierda información en ocasiones y haya demoras en la entrega de documentos, entre otras cosas. Todo esto es debido a la ausencia de un sistema que gestione los datos pertinentes de dichas actividades. A raíz de esta necesidad se determina crear una aplicación que se dedique a la gestión de todos estos procesos de investigación, surgiendo así el proyecto Sistema de Gestión de Investigaciones (SGI), el cual por su condición de haber sido creado recientemente no cuenta con un medio que se encargue de gestionar los datos



INTRODUCCIÓN

con los que va a interactuar dicho sistema, así como tampoco existe una integración de las diferentes fuentes de datos.

Para darle solución a la **situación problemática** descrita anteriormente se plantea como **problema científico**: ¿Cómo garantizar la persistencia y el acceso concurrente a la información que gestiona el SGI?

Estableciendo como **objeto de estudio** los procedimientos para el desarrollo de bases de datos y como **campo de acción** el procedimiento de desarrollo de BD dentro del Centro de Informatización Universitaria (CENIA).

Para dar solución al problema anteriormente expuesto se traza como **objetivo general**: Desarrollar una base de datos para el Sistema de Gestión de Investigaciones.

Con vista a lograr el objetivo planteado anteriormente se determinaron los siguientes **objetivos específicos**:

- Seleccionar la metodología y técnica de diseño de BD a emplear en la investigación.
- Modelar e implementar la BD del Sistema de Gestión de Investigaciones a partir de la técnica de diseño de base de datos seleccionada.
- Realizar pruebas de validación al diseño de BD del Sistema Gestión de Investigaciones.

Se define como **idea a defender** la siguiente: con el desarrollo de la base de datos del Sistema de Gestión de Investigaciones, se obtiene la persistencia y el acceso de manera concurrente a la información gestionada en el SGI.



INTRODUCCIÓN

Para el cumplimiento de los objetivos trazados se proponen las siguientes **tareas de investigación**:

- Análisis del estado del arte para fundamentar la investigación.
- Análisis de los requerimientos funcionales del sistema.
- Estudio de la metodología, patrones y técnicas de diseño de bases de datos que existen.
- Normalización de la base de datos.
- Confección del diseño de las tablas de la base de datos para el SGI.
- Estudio y desarrollo de procedimientos almacenados, funciones, vistas, disparadores, índices y llaves primarias.
- Implementación de procedimientos, funcionalidades, vistas, disparadores, índices y llaves primarias en la BD como solución al problema a resolver.
- Comprobación de la integridad referencial de los datos.
- Análisis de la base de datos en cuanto a disponibilidad, alto rendimiento y calidad del dato.

Con el objetivo de cumplir lo previsto en esta investigación se utilizan diferentes métodos científicos, con los cuales se adquiere la información necesaria referente al tema, así como cumplir los objetivos y argumentar la idea a defender.

Métodos Teóricos

- *Análisis histórico-lógico*: Permitió analizar las características de los sistemas de gestión de bases de datos más comunes utilizados para la creación de los distintos modelos existentes, realizando un estudio histórico de estos. También se evidenció a la hora de realizar un estudio detallado de las diferentes metodologías y técnicas de desarrollo de bases de datos.



INTRODUCCIÓN

- *Modelación:* Utilizado a la hora de realizar los modelos de datos del sistema a implementar.

Métodos Empíricos:

- *Revisión documental:* Se evidenció al revisar la bibliografía existente, la evolución histórica de los diferentes sistemas de gestión de datos y a la hora de consultar la información para hacer posible la realización de las tareas de investigación.
- *Entrevista:* Es una conversación planificada para obtener información. Su uso constituye un medio para el conocimiento cualitativo de los fenómenos o sobre características personales del entrevistado y puede influir en determinados aspectos de la conducta humana, por lo que es importante una buena comunicación.

Esta investigación pretende tener como **posible resultado** una base de datos para la gestión de la información que provee la persistencia y el acceso concurrente a toda la información gestionada por el SGI.

La investigación está sustentada en 3 capítulos:

Capítulo 1. Fundamentación teórica: Abarca todo el análisis bibliográfico referente al tema, se exponen las características de los diferentes sistemas de base de datos existentes. Brinda información acerca de los SGBD en general y en específico de PostgreSQL.

Capítulo 2. Diseño e implementación de la base de datos: Se evidencia la técnica de diseño de base de datos para la gestión de las actividades investigativas. También se realiza y analiza el modelo de forma general de la base de datos. Por otra parte se muestran funciones, procedimientos almacenados,



INTRODUCCIÓN

disparadores, llaves primarias e índices aplicados a la base de datos para la gestión de las actividades investigativas.

Capítulo 3. Validación de la propuesta de solución: Se exponen los resultados de las pruebas realizadas a la aplicación, tales como: alta disponibilidad, alto rendimiento, calidad del dato e integridad referencial.



Capítulo 1. Fundamentación teórica

1.1 Introducción

El presente capítulo abarca todo un estudio bibliográfico, donde se exponen las características esenciales de manera general de los Sistemas Gestores de Bases de Datos, se realiza un estudio de forma conceptual y específica donde se brindan elementos fundamentales de cada uno de estos sistemas más comunes que existen en la actualidad.

1.2 Bases de datos

Existen diversos conceptos y definiciones referentes a lo que es una base de datos según la opinión de diferentes autores y fuentes de información, a continuación se citan algunos:

- Conjunto de datos interrelacionados entre sí, almacenados con carácter más o menos permanente en la computadora. O sea, que una base de datos puede considerarse una colección de datos variables en el tiempo. (Mato García, 1999).
- Un sistema de bases de datos es básicamente un sistema computarizado cuya finalidad general es almacenar información y permitir a los usuarios recuperar y actualizar esa información. (DATE, 2001)
- Una Base de Datos es un sistema que almacena datos que están relacionados. Es un repositorio en donde guardamos información integrada que podemos almacenar y recuperar. (Tutorialesenlared, 2006).
- Un conjunto de información almacenada en memoria auxiliar que permite el acceso directo y un conjunto de programas que manipulan esos datos. (Tutorialesenlared, 2006).



CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

En resumen, se define a las bases de datos como una colección de datos estructurados, no redundantes, interrelacionados de manera lógica, que son variables en el tiempo y están organizados independientemente de su utilización y su implementación, en máquinas accesibles en tiempo real y compatible con usuarios concurrentes con necesidad de información diferente.

1.3 Sistemas gestores de bases de datos

Un Sistema Gestor de Bases de Datos (SGBD) es un programa cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos (DDL) y de un lenguaje de manipulación de datos (DML). Un SGBD permite definir los datos a distintos niveles de abstracción y manipularlos, lo que garantiza la seguridad e integridad de los mismos (CAVSI, 2008). (Date, 2001) define que *los SGBD ofrecen a los usuarios una percepción de la base de datos que está, en cierto modo, por encima del nivel del hardware y que maneja las operaciones del usuario expresadas en términos de ese nivel más alto de percepción.*

1.3.1 Objetivos principales de los SGBD

- **Abstracción de la información:** ahorran a los usuarios detalles acerca del almacenamiento físico de los datos, da lo mismo si una base de datos ocupa uno o cientos de archivos, este hecho se hace transparente al usuario (CAVSI, 2008).
- **Independencia:** capacidad de modificar el esquema físico o lógico de una BD sin tener que realizar cambios en las aplicaciones que se sirven de ella.
- **Respaldo y recuperación:** proporcionan una forma eficiente de realizar copias de respaldo de la información almacenada en ellos y de restaurar, a partir de estas copias, se recuperan datos perdidos.



CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

- **Seguridad:** garantizan que la información se encuentre segura frente a usuarios malintencionados, estos cuentan con un complejo sistema de permisos y grupos de usuarios, que permiten otorgar diversas categorías de permisos.
- **Consistencia:** en aquellos casos en los que no se ha logrado eliminar la redundancia, es necesario vigilar que aquella información que aparece repetida se actualice de forma coherente, es decir, que todos los datos repetidos se actualicen de forma simultánea.
- **Tiempo de respuesta:** es deseable minimizar el tiempo que el SGBD tarda en brindar la información solicitada y en almacenar los cambios realizados.
- **Manejo de transacciones:** se busca evitar que las transacciones interfieran unas con otras al ejecutar en paralelo, denominado "*control de paralelismo*", y garantizar que la base de datos no sea dañada en forma irreparable por caídas, ya sea del sistema en sí o de alguna de las transacciones, denominado *técnicas de recuperación* (UB, 2006).

1.3.2 Evolución de los Sistemas Gestores de Bases de Datos

La historia de las BD se inicia a mediados de los años 50, en el momento en que comenzaron a introducirse los ordenadores para automatizar la gestión de las empresas, fundamentalmente con desarrollos en *COBOL*¹.

Los Sistemas Gestores de Base Datos se han caracterizado por el uso de tecnologías orientadas a la estructuración de datos mediante modelos jerárquicos

¹ Lenguaje Común Orientado a Negocios (Common Business-Oriented Language). Fue creado en el año 1960 con el objetivo de crear un lenguaje de programación universal que pudiera ser usado en cualquier ordenador.



CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

y Codasyl de lógica procedimental, que obligan al programador a desplazarse *registro*² a registro, hecho que implica una escasa flexibilidad.

En el año 1970 se propuso el modelo relacional, basado en los trabajos del *Dr. Codd*³, básicamente el modelo matemático que dio fundamentos a la segunda generación de SGBD, caracterizada por una mayor independencia físico-lógica, dado que actúan sobre conjuntos de registros, entre ellas destacan ORACLE, DB2, INGRES, INFORMIX y SYBASE.

Codd propuso un modelo simple de datos, en el que se representarían en forma de tablas, constituidas por filas y columnas. A dichas tablas se les dio el nombre matemático de relaciones, denominándose así el sistema como relacional. También propuso dos lenguajes para manipular los datos en las tablas: álgebra y cálculo relacional, que soportan la manipulación de los datos sobre la base de operadores lógicos en lugar de los punteros físicos utilizados en los modelos jerárquicos y de red. El resultado fue la aparición de sistemas relacionales durante la última mitad de los 70, los cuales soportaban lenguajes como el Lenguaje de Consulta Estructurado (SQL por sus siglas en inglés), el Query Language (Quel) (Lenguaje de consulta, por sus siglas en español) y el Query-by-Example (QBE) (Consulta con ejemplo, por sus siglas en español).

Los trabajos de investigación que se realizaron durante la década de los 80 se centraron en la optimización de consultas, lenguajes de alto nivel, teoría de la normalización, organizaciones físicas para el almacenamiento de las relaciones,

² Un registro (también llamado fila o tupla) representa un objeto único de datos implícitamente estructurados en una tabla. Se le asigna automáticamente un número consecutivo (número de registro) que en ocasiones es usado como índice aunque lo normal y práctico es asignarle a cada registro un campo clave para su búsqueda.

³ Edgar Frank Codd (Ted Codd). Científico informático inglés (23 de agosto de 1923 - 18 de abril de 2003), conocido por sus aportes a la teoría de bases de datos relacionales.



CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

algoritmos para la gestión de memorias intermedias (buffers), técnicas de indexación para un acceso asociativo más rápido (distintas variaciones de los árboles), sistemas distribuidos, *diccionarios de datos*⁴ y gestión de transacciones. En la actualidad estas investigaciones han tenido una elevada tasa de transacciones en mucho de los productos actuales pues han permitido asegurar entornos transaccionales en línea (*Online Transaction Processing* (OLTP), por sus siglas en inglés) muy eficientes y seguros.

También cabe recordar que durante la primera mitad de los 80 se estandarizó el lenguaje SQL, el cual ofreció, al cabo de poco tiempo, prácticamente a todos los productos una interfaz SQL, aún los no relacionales (sistemas “renacidos”). La aparición y estandarización de SQL, permitió una mayor integración, multiplicó las tareas asignadas a las BD e implicó el desarrollo de sistemas de uso transparente, cuya facilidad de manejo derivó en una excepcional productividad e impresionante impacto económico.

La tercera generación de los SGBD, tiene como principal característica la optimización relacional de los sistemas en entornos multiusuario, la gestión de objetos que permite tipos de datos complejos (texto, imagen, audio, etc.), el encapsulamiento de la semántica de datos, que proporciona un soporte robusto para la recuperación automática de la información y el mantenimiento de las restricciones de integridad entre datos. Es posible clasificar los distintos sistemas como gestores de información “natural”. En la evolución de esta generación se destacan dos indicadores: **Arquitectura a tres niveles** (Tabla 1) con descripción

⁴ Es el lugar donde se deposita información sobre los datos que forman la BD. Es una guía en la que se describe la BD y los objetos que la forman.



CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

recursiva de datos (ANSI⁵, ISO⁶) como referencia; y el **Modelo relacional** (Informática Full, 2009).

La arquitectura a tres niveles tiene el propósito de separar los programas de aplicación de la base de datos física. Donde los tres esquemas no son más que las descripciones de los mismos datos pero con distintos niveles de abstracción.

Nivel	Definición
Conceptual	Nivel estructural de la BD que define su esquema lógico.
Externo	Nivel estructural de la BD que define las vistas de los usuarios
Interno	Nivel estructural de la BD que define la vista física.

Tabla 1. Arquitectura a tres niveles.

Nivel conceptual: tiene un esquema conceptual. Describe la estructura de toda la base de datos para una comunidad de usuarios. Oculta los detalles físicos de almacenamiento y trabaja con elementos lógicos como entidades, atributos y relaciones. En este nivel se puede utilizar un modelo conceptual o un modelo lógico para especificar el esquema (Martínez, 2006).

Nivel externo: posee varios esquemas externos o vistas de usuario. Cada esquema describe la visión que tiene de la base de datos a un grupo de usuarios, ocultando el resto (Martínez, 2006). En este nivel se puede utilizar un modelo conceptual o un modelo lógico para especificar los esquemas. Es el más cercano al usuario (Fundora Carrasco, 2011).

⁵ Comité con el nombre American National Standard Institute (ANSI) que propuso una arquitectura de tres niveles para los sistemas de bases de datos en el año 1975.

⁶ Organización Internacional para la Estandarización (International Organization for Standardization). Su función principal es la de buscar la estandarización de normas de productos y seguridad para las empresas u organizaciones a nivel internacional.



CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

Nivel interno o físico: tiene un esquema interno que describe la estructura física de almacenamiento de bases de datos. Emplea un modelo físico de datos y los únicos datos que existen están realmente en este nivel (Martínez, 2006). En este nivel se diseñan los archivos que contienen la información, la ubicación de los mismos y su organización, es decir, se crean los archivos de configuración (Fundora Carrasco, 2011).

La evolución de los sistemas de información acarrió a que los desarrolladores de SGBD desarrollaran potentes gestores destinados a los usuarios finales, con interfaces intuitivas-gráficas principalmente, dotadas de asistentes generadores de código.

1.3.3 Enfoques de los Sistemas Gestores de Bases de Datos

Una clasificación primaria de los SGBD permite establecer los tipos básicos según el tipo de estructura de datos que soporta:

Enfoque jerárquico: parte de una estructura de datos basada en un conjunto de registros diferentes guardados en un único archivo y jerarquizados entre sí mediante ligas. Su estructura de árbol, impone que un elemento padre puede tener varios elementos hijos, pero no su inverso. Precisa de punteros físicos (Figura 1).

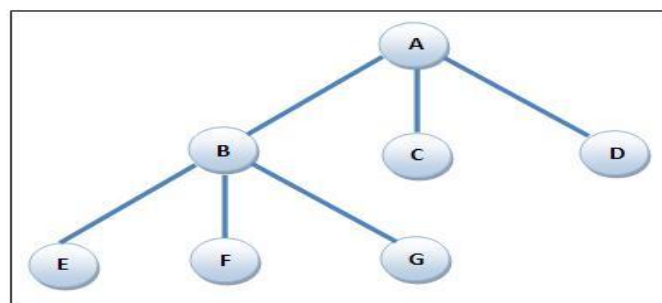


Figura 1. Estructura de una base de datos jerárquica



CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

Enfoque de red: similar al enfoque jerárquico en tanto al uso de registros y ligas, pero dentro del esquema jerárquico un elemento de inferior jerarquía puede tener varios elementos situados a un nivel superior del mismo (Figura 2). Las BD gestionadas bajo este enfoque implementan registros conectores (estructuras de datos que sirven para asociar a otras dentro de un fichero). Precisa de punteros físicos.

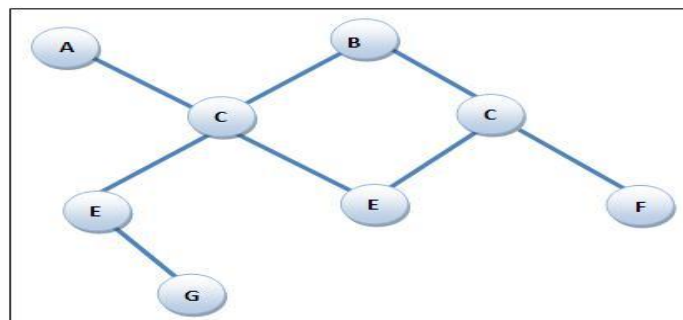


Figura 2. Estructura de una base de datos de red.

Enfoque relacional: se caracteriza por la representación de datos en forma de tablas, en las que los conjuntos de registros tienen un formato fijo e idéntica estructura (Figura 3). El enfoque relacional en BD parte del modelo relacional en matemáticas y, por tanto, son susceptibles de aplicar todas las formulaciones teóricas que este presenta.

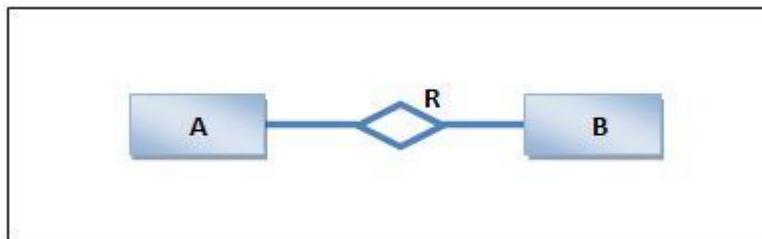


Figura 3. Estructura de una base de datos relacional.



CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

Orientado a objetos: basada en el encapsulamiento de código y datos en unidades denominadas objetos, que interactúan con el sistema a través de mensajes. El agrupamiento de objetos con métodos y variables comunes se estructura en clases jerárquicas.

Enfoque lógico: no es esencialmente distinto del enfoque relacional, pero se sustenta en la lógica de predicados de primer orden para representar y manipular los datos, con lo que se obtiene un modelo relacional flexible con capacidades para la deducción automática, a este pertenecen los SGBD deductivos (Informática Full, 2009).

Los enfoques jerárquico, de red y relacional tienen una característica en común, estos enfoques sólo aceptan datos escalares individuales tipificados y nulos. Todos estos enfoques han sido desarrollados con la capacidad de operar sobre datos “ideales”, donde se supone que la información a procesar sea exacta, correcta y bien definida.

Para la realización del modelo de datos propuesto en la investigación se selecciona el enfoque relacional, este permite la creación de entidades y sus relaciones, así como el establecimiento de restricciones de integridad. Permite además que se eviten errores a la hora de insertar y actualizar los datos, gracias a las teorías planteadas por Ted Codd en la teoría de bases de datos relacionales y su normalización.

1.3.4 Estructura de los Sistemas Gestores de Bases de Datos

Los SGBD son aplicaciones muy complicadas, en la figura 4 se muestran algunos de sus componentes, además de una representación de cómo funcionan estos de manera general, sin evidenciar la estructura real de los SGBD.



CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

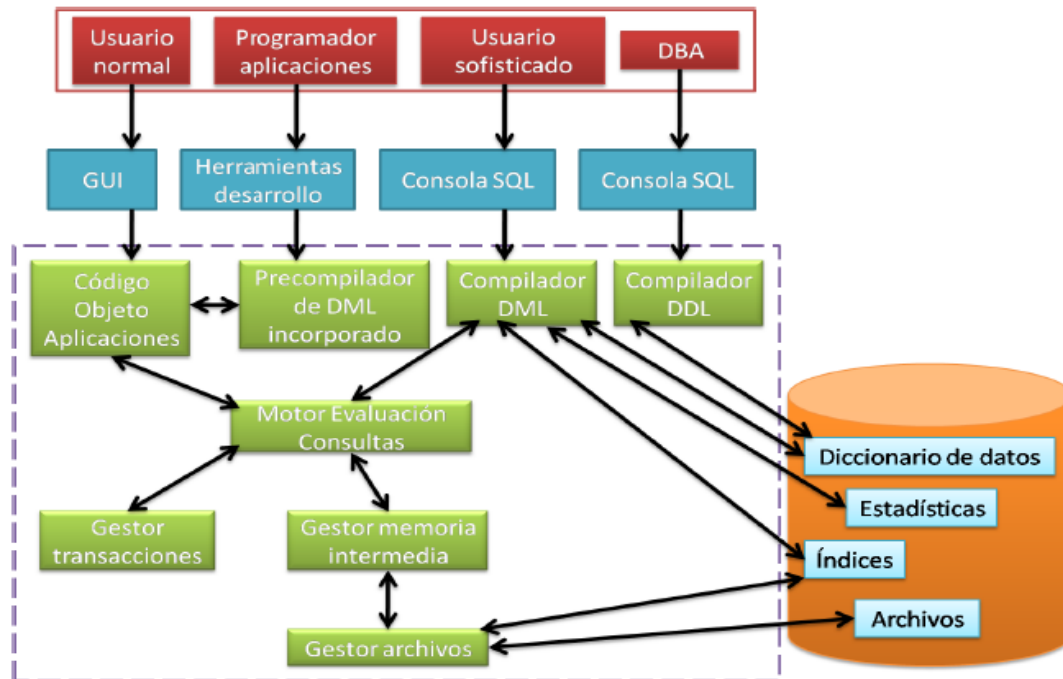


Figura 4. Estructura de un Sistema Gestor de Bases de Datos

1.3.4.1 Tipos de usuarios que interactúan con la BD

Existen diferentes tipos de usuarios que intervienen con la base de datos, estos se clasifican dependiendo de la forma en que interactúan con el SGBD.

1. *Administrador de la base de datos*: es el encargado de definir el esquema de la BD, además de su modificación en cuanto a la organización física. Es el encargado de dar la autorización para el acceso a los datos, así como la especificación de las restricciones de integridad.
2. *Usuario normal*: la comunicación con el sistema se realiza a través de programas de aplicación.
3. *Programador de aplicaciones*: Es el encargado de desarrollar las aplicaciones que utilizan la BD, estos no pueden cambiar su estructura.



CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

4. *Usuario sofisticado o programador de BD*: interactúan con la BD a través de un lenguaje, este puede cambiar su estructura.

1.3.4.2 Lenguajes de BD que utilizan los usuarios

- *Definición de Datos (DDL)*: conjunto de sentencias que permiten la definición de la estructura de la BD.
- *Manipulación de Datos (DML)*: sentencias que permiten insertar, borrar, actualizar, recuperar y organizar toda la información que se encuentra almacenada en la BD. Son instrucciones que se crean o son los llamados procedimientos.

1.3.4.3 Lenguajes para el manejo de datos

- *Procedurales*: controlan la BD registro a registro, las operaciones se deben realizar para obtener los datos como resultado.
- *No procedurales*: Operan sobre determinados conjuntos de registros, se especifica qué datos deben obtenerse pero sin decir cómo hacerlo. El lenguaje no procedural más utilizado es el SQL, el cual es un estándar y el lenguaje de los SGBD relacionales.

La diferencia entre estos dos tipos de lenguajes esta dada por el modo en que acceden a los datos.

1.3.4.4 Componentes de los SGBD

- *Gestor de archivos (GA)*: gestiona la asignación de memoria en disco y las estructuras de datos.
- *Gestor de base de datos (GBD)*: se encarga de la interfaz con los datos de bajo nivel y programas de aplicación y consultas.



CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

- *Procesador de consultas (PC)*: traduce las consultas realizadas a lenguaje de bajo nivel.
- *Precompilador de DML (PDML)*: convierte las sentencias de DML en llamadas a procedimientos.
- *Compilador de DDL (CDDL)*: convierte las sentencias de DDL en conjuntos de tablas.

1.3.4.5 Estructura de datos

Un *dato* no es más que una cierta cantidad de caracteres que describe algo de nuestra realidad. No todos los datos que se encuentran en el proceso de análisis de un fenómeno, constituye información. Se denomina información a la parte de los datos que influye en las acciones o en la toma de decisiones, o aquellos datos que su ausencia determina en la toma de decisiones o en la realización de las acciones dentro de un fenómeno en particular. Para el almacenamiento de la información en una computadora se establecen ciertas estructuras de datos que se corresponden con las características de la información almacenada. Entre las estructuras de datos en las que se almacena la información se encuentran tres tipos fundamentales de archivos:

- Archivo de datos: para almacenar los datos.
- Diccionarios de datos: para almacenar información sobre los datos que se almacenan en la BD.
- Índice: proporciona un acceso rápido a los datos (Monografias.com, 2006).

1.3.5 Ejemplos de Sistemas Gestores de Bases de Datos

Entre los SGBD que más se destacan se encuentran Oracle, MySQL (ambos actualmente de la Empresa Oracle), Microsoft SQL Server y PostgreSQL. A continuación se exponen algunas de sus características.



CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

1.3.5.1 Microsoft SQL Server

Microsoft SQL Server permite la gestión de un entorno de BD relacional, abarca tanto el área de diseño como la de administración. Es nombrado SQL porque hace uso de este lenguaje para la definición y manejo de los datos, y se llama Server porque dispone de una parte servidora que es responsable de atender a los procesos clientes, que son aquellos que realizan las peticiones al servidor, cumpliendo con una arquitectura cliente-servidor.

Ventajas

- Arquitectura de almacenamiento en disco, que permite la escalabilidad desde BD de equipos portátiles hasta BD empresariales de tamaño de terabytes.
- Permite el acceso y la realización de consultas desde URL⁷ a través de HTTP⁸, donde el optimizador de consultas con múltiples fases busca el plan óptimo de consultas para mejorar el rendimiento de consultas complejas.
- Soporta consultas en múltiples servidores.
- Simplifica la configuración y gestión de un clúster⁹ de caídas.
- Permite que las BD permanezcan en línea durante la mayoría de las operaciones.
- Activa salvallas instantáneas.
- Los servicios de transformación de datos sirven para importar, exportar y transformar datos heterogéneos.

⁷ Localizador Uniforme de Recursos, comúnmente denominado URL (sigla en inglés de Uniform Resource Locator), es una secuencia de caracteres, de acuerdo a un formato modélico y estándar, que se usa para nombrar recursos en Internet para su localización o identificación.

⁸ Protocolo de Transferencia de Hipertexto, comúnmente denominado HTTP (siglas en inglés de Hypertext Transfer Protocol), es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor.

⁹ Se aplica a los conjuntos o conglomerados de computadoras construidos mediante la utilización de componentes de hardware comunes y que se comportan como si fuesen una única computadora.



CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

- Soporte enriquecido con Lenguajes de Marcas Extensibles (XML por sus siglas en inglés) (Fundora Carrasco, 2011).

Desventajas

- Es un SGBD propietario.
- Las funciones definidas por el usuario tienen algunas restricciones.
- No todas las sentencias SQL son válidas dentro de una función (Fundora Carrasco, 2011).

1.3.5.2 MySQL

MySQL es un SGBD cliente-servidor, que usa programación multihilo y permite la conexión de múltiples usuarios de manera concurrente. Se compone de un servidor SQL, varios programas clientes y bibliotecas, herramientas administrativas, y una gran variedad de interfaces de programación (API¹⁰). Se puede obtener también como una biblioteca multihilo que se enlaza dentro de otras aplicaciones para obtener un producto más pequeño, más rápido, y más fácil de manejar (Pozo Conrado, 2005).

Ventajas:

- Dispone de borrados multitablas.
- Mejores utilidades de administración (salvas y recuperación de errores), contando con un sistema de replicación multihilo en los servidores esclavos.

¹⁰ Del inglés Application Programming Interface (API), es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.



CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

- Los registros de replicación contienen los marcadores de transacción necesarios para asegurarse que las transacciones son replicadas apropiadamente.
- Recuperación automática ante fallas.

Desventajas:

- No es viable para su uso con grandes BD, a las que se acceda continuamente, ya que no implementa una buena escalabilidad.
- Carece de soporte para transacciones, subconsultas y revertimiento de cambios realizados.
- Para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa propietaria una licencia específica que les permita este uso.

1.3.5.3 Oracle

Es un SGBD Relacional (RDBMS por el acrónimo en inglés de Relational Data Base Management System) que utiliza arquitectura cliente-servidor, fabricado por la corporación Oracle (Oracle Corporation). Es considerado uno de los sistemas de BD más completos, destacando su soporte de transacciones, estabilidad, escalabilidad y el ser multiplataforma (Masip, 2002).

Ventajas:

- Es un manejador de BD relacional que hace uso de los recursos del sistema informático en todas las arquitecturas de hardware, para garantizar su aprovechamiento al máximo en ambientes cargados de información.
- Corre automáticamente en más de ochenta arquitecturas de hardware y software distintas, sin tener la necesidad de cambiar una sola línea de código, esto se debe a que más del ochenta por ciento de los códigos



CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

internos de Oracle son iguales a los establecidos en todas las plataformas de sistemas.

- Está disponible en múltiples plataformas como Windows, Linux y todas las versiones de Unix, lo que lo convierte en una verdadera solución empresarial.
- Puede ser usado tanto para el manejo de información personal, como para gigantescas bibliotecas multimedia, y corre en ordenadores portátiles, computadoras personales (PC por sus siglas en inglés), microcomputadoras, mainframes y computadoras con procesamiento paralelo masivo.
- Soporta la mayoría de los lenguajes de computación al igual que veintiséis idiomas diferentes.
- Soporta datos alfanuméricos ubicados en las tradicionales "filas y columnas" de las BD, además de textos sin estructura, imágenes, audio y video. Incluye mejoras de rendimiento y de utilización de recursos.
- Soporta aplicaciones de Procesamiento de Transacciones Online (OLTP, por el acrónimo en inglés de Online Transaction Processing) y de Depósito de Datos (Data Warehousing, en inglés) mayores y más exigentes. Soporta ambientes de clúster en modo activo-pasivo, es decir, un sólo nodo utiliza la BD mientras el(los) otro(s) nodo(s) está(n) pendiente(s) de entrar en funcionamiento en el momento que el servidor primario tenga una falla.
- Permite tener copias de la BD productiva en lugares lejanos a la ubicación principal, a través de réplicas, que pueden estar en modo de sólo lectura. A partir de la versión Oracle11g R2 en adelante los bloques se recuperan de forma automática en ambientes de BD de reserva o espera (Fundora Carrasco, 2011).

Desventajas:



CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

- Es un software propietario.
- Tiene un elevado costo de soporte técnico.

1.3.5.4 PostgreSQL

PostgreSQL es un sistema de gestión de BD objeto-relacional (ORDBMS, por sus siglas en inglés) basado en Postgres versión 4.2, desarrollado en la Universidad de California en el departamento de Informática de Berkeley. Postgres es pionero en muchos conceptos que sólo llegaron a estar disponibles mucho después en sistemas de bases de datos comerciales. PostgreSQL es un descendiente Open Source (Código abierto en español) del código original Berkeley. Soporta una gran parte del SQL estándar y ofrece muchas características modernas como:

- Consultas complejas.
- Claves foráneas.
- Disparadores.
- Vistas.
- Integridad transaccional.
- Control de concurrencia (multiversión).

Este SGBD también puede ser ampliado por el usuario de muchas formas, por ejemplo, añadiendo nuevos:

- Tipos de datos
- Funciones
- Operadores
- Funciones agregadas
- Métodos de indexación
- Lenguajes procedurales



CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

Debido a que la licencia es libre, PostgreSQL puede ser usado, modificado y distribuido por cualquier persona con cualquier propósito, ya sea privado, comercial o académico (PostgreSQL 9.0.1 Documentation). Este utiliza un modelo cliente-servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afecta al resto y el sistema continúa funcionando.

En la Figura 5 se muestra de manera general los componentes más importantes en un sistema PostgreSQL.

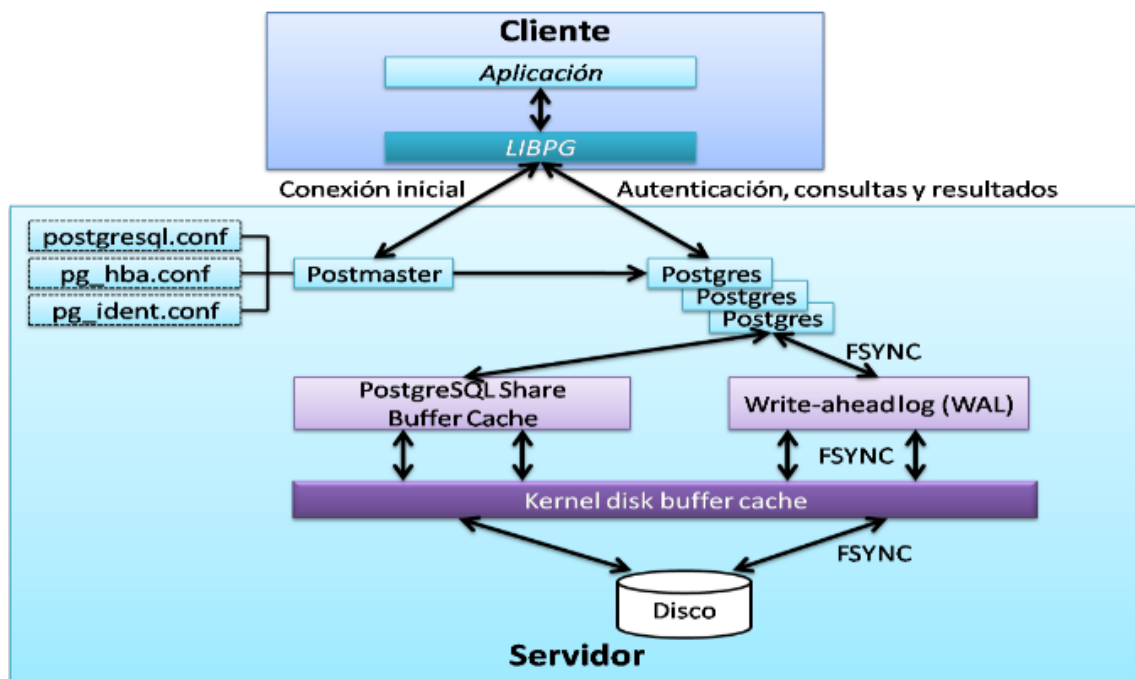


Figura 5. Componentes de un sistema PostgreSQL.

- La conexión puede ocurrir vía TCP/IP¹¹ o sockets¹² locales.

¹¹ TCP/IP es un conjunto de protocolos. Las siglas TCP/IP significan "Protocolo de control de transmisión/Protocolo de Internet". Provee conectividad de extremo a extremo especificando cómo los datos deberían ser formateados, direccionados, transmitidos, enrutados y recibidos por el destinatario.



CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

- **Demonio postmaster:** este es el proceso principal de PostgreSQL. Es el encargado de escuchar por un puerto/socket, conexiones entrantes de clientes. También es el encargado de crear los procesos hijos que se responsabilizan de autenticar estas peticiones, gestionar las consultas y mandar los resultados a las aplicaciones clientes.
- **Ficheros de configuración:** los tres ficheros principales de configuración utilizados por PostgreSQL son: postgresql.conf, pg_hba.conf y pg_ident.conf.
- **Procesos hijos PostgreSQL:** procesos hijos que se encargan de autenticar a los clientes, de gestionar las consultas y mandar los resultados a las aplicaciones clientes.
- **PostgreSQL share buffer cache:** memoria compartida usada por PostgreSQL para almacenar datos en caché.
- **Write-Ahead Logging (WAL):** Escritura anticipada de registro, es el componente del sistema encargado de asegurar la integridad de los datos.
- **Kernel disk buffer cache:** caché de disco del sistema operativo.
- **Disco:** disco físico donde se almacenan los datos y toda la información necesaria para que PostgreSQL funcione (PostgreSQL-Es, 2009).

Principales características

¹² Socket o "Enchufe" en español, designa un concepto abstracto por el cual dos programas (posiblemente situados en computadoras distintas) pueden intercambiar cualquier flujo de datos, generalmente de manera fiable y ordenada.



CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

- Soporta casi toda la sintaxis SQL, tiene soporte total para llaves foráneas, uniones, vistas, disparadores y procedimientos almacenados (en múltiples lenguajes).
- Integridad transaccional, obedece completamente a la especificación ACID¹³.
- Acceso concurrente multi-versión, Control de Concurrencia Multi-Versión (MVCC por sus siglas en inglés), no se bloquean las tablas, ni siquiera las filas, cuando un proceso escribe. Es la tecnología que PostgreSQL usa para evitar bloqueos innecesarios. Mediante el uso de MVCC, PostgreSQL evita el problema de que los procesos lectores estén esperando a que se termine de escribir. En su lugar, PostgreSQL mantiene una ruta a todas las transacciones realizadas por los usuarios de la BD. PostgreSQL es capaz entonces de manejar los registros sin necesidad de que los usuarios tengan que esperar a que los registros estén disponibles.
- Cliente-servidor: PostgreSQL usa una arquitectura proceso-por-usuario cliente-servidor. Esta es similar al método del Apache 1.3.x para manejar procesos. Hay un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que se intente conectar a PostgreSQL.
- Write Ahead Logging (WAL): la característica de PostgreSQL conocida como Write Ahead Logging (Escritura anticipada de registro, según su significado en español) incrementa la dependencia de la BD al registro de cambios antes de que estos sean escritos en la misma. Esto garantiza que en el hipotético caso de que la BD se caiga, existirá un registro de las transacciones a partir del cual se podrá restaurar la BD. Esto puede ser enormemente beneficioso en el caso

¹³ ACID es un acrónimo de Atomicity, Consistency, Isolation and Durability: Atomicidad, Consistencia, Aislamiento y Durabilidad en español. Es un conjunto de características necesarias para que una serie de instrucciones puedan ser consideradas como una transacción.



CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

de caída, ya que cualquiera de los cambios que no fueron escritos en la BD pueden ser recuperados usando el dato que fue previamente registrado. Una vez que el sistema ha quedado restaurado, un usuario puede continuar trabajando desde el punto en que lo dejó cuando cayó la BD.

- Lenguajes procedurales: PostgreSQL tiene soporte para lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL/pgSQL. Este lenguaje es comparable al lenguaje procedural de Oracle, PL/SQL. Otra ventaja de PostgreSQL es su habilidad para usar Perl, Python, o TCL como lenguaje procedural embebido.
- Interfaces con lenguajes de programación: la flexibilidad del API de PostgreSQL ha permitido a los vendedores proporcionar soporte al desarrollo fácilmente para el RDBMS PostgreSQL. Estas interfaces incluyen Object Pascal, Python, Perl, PHP, ODBC, Java/JDBC, Ruby, TCL, C/C++ y Pike.
- Herencia de tablas.
- Incluye la mayoría de los tipos de datos SQL92 y SQL99 (INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL y TIMESTAMP), soporta almacenamiento de objetos grandes binarios, además de tipos de datos y operaciones geométricas.
- Puntos de recuperación a un momento dado, tablespaces, replicación asincrónica, transacciones jerarquizadas (savepoints¹⁴) y backups en línea.

¹⁴ Un savepoint (en español, punto de recuperación) es una forma de implementar subtransacciones dentro de un ORDBMS indicando un punto dentro de una transacción de BD que puede ser "rolled back" (devuelta) sin afectar a cualquier trabajo realizado en la transacción antes de que el punto de recuperación fuera creado.



CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

- Un sofisticado analizador/optimizador de consultas.
- Soporta juegos de caracteres internacionales y codificación de caracteres multibyte.

Ventajas

- Aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas.
- Usa una arquitectura proceso por usuario cliente-servidor: hay un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intente conectarse a PostgreSQL.
- Altamente extensible: soporta los tipos de datos base y otros como los de fechas, monetarios, elementos gráficos, datos sobre redes (MAC, IP) y cadenas de bits. Además operadores, funciones, métodos de acceso y tipos de datos definidos por el usuario.
- Integridad referencial: es utilizada para garantizar la validez de los datos de la BD.
- Tiene soporte para lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL/pgSQL. Además tiene habilidad para usar Perl, Python, o TCL como lenguaje procedural embebido.
- Control de Concurrencia Multi-Versión: es usado para evitar bloqueos innecesarios, permite la lectura sin que sea bloqueada por los que escriben y actualizan registros.
- Incrementa la dependencia de la BD al registro de cambios antes de que estos sean escritos en la BD. Esto garantiza que en caso de que la BD se



CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

caiga, exista un registro de las transacciones a partir del cual se pueda restaurar la BD desde el punto en que se quedó (Fundora Carrasco, 2011).

Desventaja

- Consume muchos recursos y carga con facilidad el sistema.

1.3.6 Comparación entre los diferentes SGBDR.

Para tener una visión del porque se toma PostgreSQL para el desarrollo de la base de datos se realiza un estudio comparativo de los diferentes SGBD que se mencionaron anteriormente como se muestra en la Tabla 2:

Criterios	SQLSERVER	ORACLE	MySQL	PostgreSQL
Plataforma	Windows	Windows GNU/LINUX	Windows GNU/LINUX	Windows GNU/LINUX
Requerimiento de hardware	+	-	+	+
Coste	-	-	+	+
Potencia	+	+	+	+
Escalabilidad	+	+	-	+
Velocidad	+	+	+	-
Leyenda	Aspecto positivo:		+	
	Aspecto negativo:		-	

Tabla 2. Comparación entre los Sistemas Gestores de Bases de Datos.

Después de haber realizado un análisis de los diferentes SGBD, se puede llegar a la conclusión de que se concuerda con que el SGBD PostgreSQL es el mejor para llevar a cabo el desarrollo de la BD del Sistema de Gestión de Investigaciones, ya que SQL Server no es multiplataforma y así como Oracle para su uso es necesario



CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

pagar importes por su licencia, lo cual no cumple con las políticas de la Universidad y del país de migrar al software libre.

Por otra parte MySQL y PostgreSQL cumplen con dichas políticas, no obstante, se utiliza PostgreSQL debido a que en comparación con MySQL posee una gran escalabilidad, lo que lo hace más conveniente en cuanto a usabilidad, además puede ajustarse a la cantidad de computadoras que se conecten a la BD, así como a la memoria que posee el sistema de forma óptima, esto hace que sea adecuado para soportar mayor cantidad de peticiones simultáneas de manera correcta y permite tres veces más la carga que podría soportar MySQL.

PostgreSQL implementa el uso de rollback's¹⁵, subconsultas y transacciones, lo que hace su funcionamiento mucho más eficaz y brinda soluciones en campos en los que MySQL no podría.

En la actualidad el uso de PostgreSQL es cada vez mayor en las empresas que buscan un servidor de base de datos altamente sofisticado, con alto rendimiento, estable y capacitado para lidiar con grandes volúmenes de datos. El hecho de ser un producto de código abierto, sin costos de licencia, convierte al PostgreSQL en una alternativa extremadamente atractiva para las empresas que buscan un ahorro significativo de costos en activos. Con la apreciación de este estudio el grupo CENIA establece PostgreSQL para la modelación y administración de las bases de datos.

1.4 Herramientas de modelado

Las herramientas *CASE (Computer Aided Software Engineering)* sirven de ayuda en todos los aspectos del ciclo de vida del desarrollo de un software, en tareas

¹⁵ En tecnologías de base de datos, un rollback es una operación que devuelve a la base de datos a algún estado previo. Los Rollbacks son importantes para la integridad de la base de datos, a causa de que significan que la base de datos puede ser restaurada a una copia limpia incluso después de que se han realizado operaciones erróneas.



CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores. (Barranco de Areba, 2001).

Visual Paradigm.

El Visual Paradigm es una herramienta CASE (acrónimo de Computer Aided Software Engineering, en español, Ingeniería de Software Asistida por Computadora) que brinda un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. Visual Paradigm ha sido concebido para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas. Constituye una herramienta de software libre de probada utilidad. Fue diseñado para una amplia gama de usuarios interesados en la construcción de sistemas de software de forma fiable a través de la utilización de un enfoque Orientado a Objetos (EcuRed, 2008).

1.5 Conclusiones parciales

En este capítulo se profundizó en el estudio de los conceptos de los SGBD, así como de las características de los mismos que existen en la actualidad ratificando que PostgreSQL es el indicado para el desarrollo y gestión de la BD. Se determinó que para la realización del modelo de datos se tuvo en cuenta el enfoque relacional, que permite la creación de entidades y sus relaciones, así como el establecimiento de restricciones de integridad. Además se fundamentó acerca de la herramienta CASE Visual Paradigm.



CAPÍTULO 2. DISEÑO E IMPLEMENTACIÓN DE LA BASE DE DATOS

Capítulo 2. Diseño e implementación de la base de datos

2.1 Introducción

En el presente capítulo se expone información relacionada a la metodología empleada para diseñar bases de datos relacionales, también se brinda información de las más comunes técnicas de diseño de bases de datos para poder seleccionar una de estas, la cual se aplicará posteriormente con el objetivo de obtener los modelos de datos necesarios. Se brindan aspectos acerca del soporte arquitectónico de la BD. Posteriormente se realiza un análisis del modelo de la BD para el almacenamiento de la información generada de la gestión de los procesos de investigaciones y por último se fundamenta acerca de la implementación de dicha BD con elementos tales como: funciones, disparadores, índices y llaves primarias con el propósito de llevar a cabo los objetivos propuestos en la investigación.

2.2 Diseño de la base de datos

El diseño de BD se convierte en una actividad crucial en el manejo de sistemas complejos, este se simplifica cuando se utilizan modelos. Los modelos son abstracciones simplificadas de eventos y condiciones del mundo real, que permiten explorar las características de entidades y las relaciones que se pueden crear entre ellas. Si los modelos no son lógicamente buenos, no se lograrán diseños de BD funcionales que permitan obtener información útil. Los buenos modelos dan como resultado correctos diseños de bases de datos que son la base para aplicaciones funcionales. (Rojas Santiesteban & Boloy Boado, 2010).



CAPÍTULO 2. DISEÑO E IMPLEMENTACIÓN DE LA BASE DE DATOS

2.3 Patrones de diseño de bases de datos

Se puede definir un patrón como el fragmento de un modelo que es recurrente. Un patrón es una solución a un problema específico que se ha mantenido a pesar del tiempo. Algunas definiciones para patrones de que se pueden encontrar según diferentes fuentes son:

- Un patrón de arquitectura de software describe un problema particular de diseño recurrente en contextos específicos y presenta un esquema genérico probado para su solución.
- Un patrón es una plantilla. Esta plantilla se utiliza para dar solución a un problema recurrente de la realidad. Es una plantilla para una solución, no una solución en sí.
- Un patrón es una plantilla de objetos que interactúan, que puede ser utilizado una y otra vez por analogía.
- Un patrón de diseño nombra y explica un diseño general que se ocupa de un problema de diseño se repite... En él se describe el problema, la solución, cuándo aplicar la solución y sus consecuencias.
- Un patrón es una solución recurrente a un problema estándar... Los patrones tienen un contexto en el cuál son aplicables.
- En general, un patrón es un binomio problema-solución en un contexto dado. Un patrón no sólo documenta "cómo" una solución resuelve un problema, sino también "por qué" se resuelve, es decir, el fundamento de esta solución en particular. (EVA, 2012).

Durante la investigación se evidenció el uso de algunos de los patrones más conocidos, para lograr un mejor diseño en aras de hacer más fácil la



CAPÍTULO 2. DISEÑO E IMPLEMENTACIÓN DE LA BASE DE DATOS

implementación del sistema. Los cuales se exponen a continuación, con su respectiva descripción.

- Árboles fuertemente codificados (Hardcoded tree, por sus siglas en inglés): En este caso a cada nivel del árbol se le asocia una entidad. Normalmente constituyen relaciones de 1 a muchos (n). Normalmente utilizado para representar jerarquías donde es bien conocida la estructura y es importante representar la correspondencia, por ejemplo las estructuras organizacionales. Es importante señalar que este patrón debe utilizarse sólo en los casos en que los cambios en la estructura a representar sean poco probables.
- Árboles simples: Patrón normalmente utilizado cuando el árbol es la representación de una estructura de datos. Los elementos a almacenar son del mismo tipo, es decir, pueden ser almacenados en la misma entidad. No pueden existir ciclos, es decir, un hijo no puede ser su propio padre.
- Patrón de llaves subrogadas: Este patrón es muy utilizado pues facilita la interacción con la BD en un futuro. El mismo plantea que se genere una llave primaria única para cada entidad, en vez de usar un atributo identificador en el contexto dado. Normalmente se usan enteros en columnas identity (identidad, en español) o GUID (Global UniqueIdentifier, por sus siglas en inglés) que están demostradas que no se repiten o con una probabilidad extremadamente baja. Esto permite que las tablas sean más fáciles de consultar a partir del identificador, pues todos tienen el mismo tipo en cada una de las tablas.

2.4 Metodología para el diseño de bases de datos

La metodología para el diseño de la BD, consta de los siguientes pasos:



CAPÍTULO 2. DISEÑO E IMPLEMENTACIÓN DE LA BASE DE DATOS

- Determinación de entidades y atributos.
- Normalización de entidades.
- Determinación de relaciones.
- Obtención del modelo lógico global de los datos.
- Diseño físico de la BD.

Cuando se realiza el diseño de la BD para un sistema determinado, es necesario establecer los datos a tomar en cuenta y las dependencias funcionales existentes entre ellos. Rigurosamente, esto se obtiene luego de realizada la etapa de análisis del sistema y partiendo de lo obtenido en esta (Mato García, 1999).

2.5 Técnicas para diseñar bases de datos relacionales

Existen dos escuelas que suceden dos grupos de estrategias a la hora de aplicar la Teoría de Normalización: los métodos de análisis o descomposición y por otra parte los procedimientos de síntesis, que seguidamente se detallan.

Diseño descendente, análisis o descomposición.

Este método es el que se emplea mediante un proceso denominado “normalización”. El mismo parte de una relación llamada Universal que contiene todos los atributos del universo del discurso y las dependencias funcionales existentes entre ellas. Por medio de descomposiciones sucesivas y cumpliendo determinados principios de conservación de la información y de las dependencias, resultan esquemas de menor grado en formas normales cada vez más avanzadas, por lo que se puede garantizar que se reducen las anomalías. Se pueden aplicar estos métodos también a cada una de las relaciones obtenidas a partir de un esquema conceptual en un modelo de datos de alto nivel (Modelo Entidad-Relación) y luego transformar el esquema conceptual a un conjunto de relaciones, empleando un procedimiento de transformación. También se puede aplicar el



CAPÍTULO 2. DISEÑO E IMPLEMENTACIÓN DE LA BASE DE DATOS

método a cada una de las relaciones que se obtienen de la transformación del Diagrama Entidad-Relación al modelo relacional. El objetivo inicial que se pretende con estos métodos es separar la información referente a un objeto o entidad diferente.

Síntesis relacional o Procedimientos de síntesis.

Este es un método de diseño alternativo a la descomposición, resulta ser un enfoque más purista. Implica contemplar el diseño de esquemas de BD relacionales estrictamente en términos de dependencias funcionales, y otros tipos especificados para los atributos de la BD. Aquí los esquemas de relación en tercera forma normal o forma normal de Boyce-Codd se sintetizan gradualmente agrupando los atributos apropiados. A partir de conjuntos de atributos y dependencias funcionales se obtienen relaciones. Recorre un camino inverso a la descomposición, es decir, busca agrupar atributos, a fin de tener en una relación toda la información correspondiente a un objeto o entidad (EVA, 2011).

2.5.1 Normalización

Normalización es un grupo de reglas que sirven para ayudar a los diseñadores a elaborar un esquema que minimice los problemas de lógica. Cada regla está basada en la que le antecede. La normalización se adoptó porque el viejo estilo de poner todos los datos en un solo lugar, como un archivo o una tabla de la base de datos, era ineficiente y conducía a errores de lógica cuando se trataba de manipular los datos. Evitando así la aparición de anomalías de actualización.

El concepto de normalización fue introducido por primera vez por E.D. Codd y fue tomado en cuenta a la hora de aplicarse a sistemas relacionales. No obstante, tiene muchas más aplicaciones. La normalización es la expresión formal del estilo de realizar un buen diseño. Provee los medios necesarios para describir la



CAPÍTULO 2. DISEÑO E IMPLEMENTACIÓN DE LA BASE DE DATOS

estructura lógica de los datos en un sistema de información. Debido a este proceso que se le realiza a la BD, aparecen las siguientes ventajas:

- Evita anomalías en la actualización.
- Mejora la independencia de los datos, permite realizar extensiones de la BD y afecta muy poco, o nada, a los programas de aplicación existentes que accedan a la BD.

La normalización consta de varias fases que se realizan en un determinado orden. Para la realización de la segunda fase debe haberse concluido la primera y así sucesivamente. Tras culminar cada fase se dice que la relación está en:

- ✓ Primera Forma Normal (1FN): una relación está en primera forma normal si los valores de sus atributos son atómicos, los dominios no tienen elementos que a su vez sean conjuntos y la relación no incluye ningún grupo repetitivo.
- ✓ Segunda Forma Normal (2FN): debe estar en 1FN y además define que los atributos no llaves son funcionales y completamente dependientes de la llave primaria, por lo general compuesta.
- ✓ Tercera Forma Normal (3FN): debe estar en 2FN y señala que no pueden existir dependencias transitivas entre los atributos no llaves, los que son independientes de cualquier otro atributo no llave primaria.
- ✓ Forma Normal de *Boyce-Codd* (FNBC): además de cumplir con lo de las tres formas normales anteriores, debe satisfacer que cada determinante es una llave (candidata o primaria).
- ✓ Cuarta Forma Normal (4FN): debe estar en FNBC y que además existan tres o más atributos formando parte de la llave primaria. Debe cumplir que



CAPÍTULO 2. DISEÑO E IMPLEMENTACIÓN DE LA BASE DE DATOS

no existan dos o más atributos independientes con dependencia respecto a un conjunto de atributos, formando parte de la llave junto a dicho conjunto, es decir, no dependencia entre atributos llaves.

- ✓ Quinta Forma Normal (5FN): cumple que toda dependencia de agregación (*join dependency*, por sus siglas en inglés) es implicada por las llaves candidatas. Diseñada para reducir la redundancia en relaciones que almacenan hechos multi-evaluados a través del aislamiento de relaciones semánticamente relacionadas. Una relación está en 5FN cuando su contenido no puede ser reconstruido a partir de un conjunto de tablas, se excluye la posibilidad de tablas con los mismos atributos llaves.
- ✓ FN de dominio/llave (*DKNF*, por sus siglas en inglés): no depende de las FN anteriores. Declara de manera explícita todas las reglas de negocios como reglas de la BD. Con ella se evitan las anomalías no temporales a la hora de insertar y eliminar datos o entidades completas con relaciones de dependencias. Define como “llave” al identificador único de una tupla y como “dominio” a los valores de un atributo, que son permitidos (tanto física como semánticamente). Dentro de sus características fundamentales se define el proceso más simple: no se necesita saber que se elimina una dependencia transitiva. Sólo se deben aplicar las restricciones que surgen como consecuencia lógica de las definiciones de llave y del dominio.

2.6 Diagrama Entidad - Relación

Un Modelo o Diagrama Entidad Relación (en ocasiones denominado por su siglas en inglés, “E-R” (Entity Relationship), o también “DER” Diagrama Entidad Relación), es una herramienta usada para el modelado de datos de un determinado sistema de información. Estos modelos expresan entidades



CAPÍTULO 2. DISEÑO E IMPLEMENTACIÓN DE LA BASE DE DATOS

relevantes, así como sus interrelaciones y propiedades. Propuesto por *Peter Chen* en 1976, mediante este diagrama lo que se quiere es visualizar los elementos que pertenecen a una BD, que reciben el nombre de entidades, las cuales se corresponden con el concepto de clase de la Programación Orientada a Objeto (POO). Los cuatro elementos fundamentales de un Modelo Entidad Relación (MER) son: las entidades, los atributos, las interrelaciones y el dominio.

Entidad: es el objetivo básico representado en el MER. Representa elementos u objetos de la vida real con existencia necesaria en la BD.

- ✓ Entidad débil: es el tipo de entidad que no puede existir por sí sola, su existencia depende de otra entidad.
- ✓ Atributos: es la unidad menor de información sobre un objeto almacenado en la BD. Son propiedades que poseen las entidades, por ejemplo un color, un nombre o cualquier otro dato que ofrezca información sobre la entidad.
- ✓ Dominio: conjunto de valores posibles que puede tomar un atributo determinado.
- ✓ Interrelación: relación, vínculo o correspondencia entre entidades.
- ✓ Llave primaria: atributo (o conjunto de atributos) que identifica a una entidad como un objeto único e irrepitible dentro de la base de datos.
- ✓ Llave foránea: conjunto de atributos común a dos entidades que sirve como relación entre las mismas. No es un atributo de la entidad relacionada, pero es la llave primaria de la entidad con la cual esta se relaciona.



CAPÍTULO 2. DISEÑO E IMPLEMENTACIÓN DE LA BASE DE DATOS

- ✓ Llaves secundarias: no pueden identificar en forma única a un registro. Las llaves secundarias pueden usarse para seleccionar un grupo de registros que pertenecen a un conjunto.

2.6.1 Diagrama Entidad-Relación de la base de datos del Sistema de Gestión de Investigaciones.

El diagrama entidad-relación representa el estado final de las tablas de la base de datos. En la propuesta el modelo de datos va estar formado por tres módulos, estos darán lugar a tres esquemas de la base de datos, logrando con esto una mayor organización en el diseño. El diseño fue realizado manteniendo una total correspondencia con los requisitos funcionales y las restricciones de la BD, elementos obtenidos después del estudio de los requisitos funcionales y las reglas del negocio del sistema identificadas por las analistas del sistema.

El modelo físico se obtuvo luego de hacerse un modelo lógico utilizando el *Visual Paradigm*, sincronizando este último a dicho modelo físico. El modelo físico de la BD se podrá encontrar detallado en las figuras del anexo 2; a continuación se muestra un parte de la descripción de las tablas de la BD, el resto de estas se encuentra en el anexo 3, todo esto haciendo referencia al diccionario de datos del propio sistema. El diseño de la BD que se elaboró dio lugar a la aplicación que se obtuvo como resultado de la investigación, luego de generarse el script de la BD en el Visual Paradigm a partir del modelo físico elaborado previamente en dicha herramienta.

2.6.2 Descripción de las tablas.

Módulo eventos:

Nombre: tb_devento



CAPÍTULO 2. DISEÑO E IMPLEMENTACIÓN DE LA BASE DE DATOS

Descripción: Almacena los datos de los eventos investigativos que se realicen.		
Atributo	Tipo	Descripción
id_evento	Entero	Identificador del evento.
id_estado_evento	Entero	Identificador del estado del evento.
id_alcance_evento	Entero	Identificador del alcance del evento.
nombre_evento	Cadena	Nombre del evento.
objetivos	Texto	Objetivos del evento.
url	Texto	URL del sitio del evento.
otorga_premio	Boolean	Indica si el evento otorga premios.
descripción_rechazo	Texto	Descripción del posible rechazo.
id_solicitante	Entero	Identificador del solicitante.

Tabla 3. Tabla evento del módulo eventos.

Módulo premios:

Nombre: tb_dpremio		
Descripción: Almacena los datos de los premios.		
Atributo	Tipo	Descripción
id_premio	Entero	Identificador del premio.
id_alcance	Entero	Identificador del alcance de un premio.



CAPÍTULO 2. DISEÑO E IMPLEMENTACIÓN DE LA BASE DE DATOS

id_institucion_promotora	Entero	Identificador de la institución promotora.
id_frecuencia	Entero	Identificador de la frecuencia.
nombre_premio	Cadena	Nombre del premio.
descripcion	Texto	Descripción del premio.
fecha_entrega	Fecha	Fecha en que se entrega el premio.

Tabla 4. Tabla premio del módulo premios.

Módulo líneas y proyectos:

Nombre: tb_dlinea_investigacion		
Descripción: Almacena las líneas de investigación.		
Atributo	Tipo	Descripción
id_linea_investigacion	Entero	Identificador de la línea de investigación.
id_estado_linea	Entero	Identificador del estado de la línea.
id_persona_lider	Entero	Identificador de la persona líder.
id_persona_admon	Entero	Identificador del administrador de la línea.
id_clasificacion_linea	Entero	Identificador de la clasificación de la línea.



CAPÍTULO 2. DISEÑO E IMPLEMENTACIÓN DE LA BASE DE DATOS

nombre	Cadena	Nombre de la línea.
decripcion	Texto	Descripción de la línea.
estructura_entidad	Entero	Estructura de la entidad.
cronograma	Texto	Cronograma de la línea.
activo	Boolean	Si esta activa la línea.

Tabla 5. Tabla línea de investigación del módulo líneas y proyectos.

Nombre: tb_dproyecto_investigacion		
Descripción: Almacena los proyectos de investigación.		
Atributo	Tipo	Descripción
id_proyecto_investigacion	Entero	Identificador del proyecto.
id_persona_jefe	Entero	Identificador de la persona jefe.
id_tematica	Entero	Identificador de la temática.
id_estado_desarrollo	Entero	Identificador del estado de desarrollo.
id_estado_revision	Entero	Identificador del estado de revisión.
id_clasificacion_proyecto	Entero	Identificador de la clasificación del proyecto.
titulo	Texto	Título del proyecto.



CAPÍTULO 2. DISEÑO E IMPLEMENTACIÓN DE LA BASE DE DATOS

activo	Boolean	Si esta activo el proyecto.
observacion	Texto	Observaciones pertinentes.

Tabla 6. Tabla proyecto de investigación del módulo líneas y proyectos.

2.7 Soporte arquitectónico

A continuación se tratan aspectos relacionados al soporte arquitectónico, el cual cumple con los requisitos no funcionales de la BD.

2.7.1 Seguridad y disponibilidad de los datos

La seguridad de la BD está definida a nivel de roles, con el propósito de garantizar la integridad de los datos con vista al acceso que le corresponde a cada usuario. El sistema presenta un grupo de políticas de accesibilidad a las distintas funcionalidades de este dependiendo del nivel de autorización que posea un usuario determinado. Los servicios web que brinde la aplicación deben estar restringidos a grupos de usuarios predeterminados y aprobados con anterioridad.

El sistema debe tener la capacidad de registrar las acciones del usuario, así como permitir auditorías y chequeos de las trazas, ya sea en tiempo real como en históricos. Es necesaria la existencia de un monitor de incidencias para la visualización y tratamiento de dichas trazas.

El servicio de BD se brinda mediante el SGBD PostgreSQL versión 8.4 o superior bajo el sistema operativo Ubuntu Server 11.10. Como cliente de administración en GNU/Linux se sugiere utilizar pgAdmin III, versión 1.10. El servidor de aplicaciones web que se utiliza es Apache 2.2 o superior.



CAPÍTULO 2. DISEÑO E IMPLEMENTACIÓN DE LA BASE DE DATOS

La responsabilidad de protección a la BD está delegada al sistema operativo bajo el uso de NetFilter/IPTables¹⁶. El control de acceso a la BD es entregado al SGBD a través de sus archivos de configuración: postgresql.conf y pg_hba.conf. El método de cifrado md5 es el empleado para la conexión a la BD.

Se usan servidores de BD centralizados para el desarrollo y las pruebas, debido a que es menester impedir una posible pérdida del avance del desarrollo individual si se daña la máquina de desarrollo. Por otra parte, debe evitarse la duplicación innecesaria de tablas existentes en esquemas adyacentes.

2.7.2 Estándares de nomenclatura

La investigación se ajusta a una nomenclatura definida por el centro, esto no es más que una serie de nombres de los objetos con textos descriptivos, de manera que el nombre de los objetos den a entender a qué se hace referencia. En el anexo 1 se muestra una tabla donde se explica dicha nomenclatura.

2.7.3 Herramientas usadas

Para la creación, la modificación y las relaciones de las tablas, se debe utilizar siempre una herramienta capaz de mantener los objetos físicos que esté relacionado con el modelo lógico, de modo que se logre la documentación.

Con el objetivo de lograr este propósito, la herramienta a utilizar para la creación y sincronización hacia o desde la BD es el *Visual Paradigm* versión 8.0, esta herramienta permite el trabajo con *PostgreSQL* versión 8.4. Hay una licencia adquirida por la universidad para su uso y funciona en la plataforma *GNU/Linux*. *Visual Paradigm* para *UML (Unified Modeling Language*, por sus siglas en inglés, Lenguaje Unificado de Modelado), es fácil de usar y soporta la última notación

¹⁶ Netfilter es un *framework* disponible en el núcleo de *Linux* que permite interceptar y manipular paquetes de red. El componente más popular construido sobre *Netfilter* es *iptables*, una herramienta de cortafuegos que permite no solamente filtrar paquetes, sino también realizar traducción de direcciones de red para IPv4 o mantener registros de *log*.



CAPÍTULO 2. DISEÑO E IMPLEMENTACIÓN DE LA BASE DE DATOS

UML 2.1, ingeniería inversa, generación de código, importación desde *Rational Rose*, exportación/importación XML (del inglés *eXtensible Markup Language*, lenguaje de marcas extensible), generador de informes, editor de figuras, integración con *MS Visio*, *plug-in*¹⁷, integración *IDE*¹⁸ con Visual Studio, Eclipse, NetBeans y otros. Entre sus nuevas características se incluyen el modelado colaborativo con CVS y Subversion, e interoperabilidad con modelos UML a través de XML.

Como aplicación gráfica para administrar el SGBD PostgreSQL se hace uso del PgAdmin III, la herramienta más popular de código abierto de administración y plataforma de desarrollo de PostgreSQL.

PgAdmin III está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples hasta desarrollar BD complejas. La interfaz gráfica soporta todas las características de PostgreSQL y facilita enormemente la administración. La aplicación también incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor, un agente para lanzar scripts programados, soporte para el motor de replicación Slony-I y mucho más. La conexión al servidor puede hacerse mediante TCP/IP o *Unix Domain Sockets* (en plataformas *Unix*), y puede encriptarse mediante *SSL* para mayor seguridad (PgAdmin PostgreSQL Tools, 2008).

2.7.4 Tipos de datos

Tipo de datos "serial"

¹⁷ Un *plug-in* o complemento es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica. Esta aplicación adicional es ejecutada por la aplicación principal e interactúan por medio de la API.

¹⁸ IDE (en inglés *integrated development environment*, entorno de desarrollo integrado) es un programa informático compuesto por un conjunto de herramientas de programación.



CAPÍTULO 2. DISEÑO E IMPLEMENTACIÓN DE LA BASE DE DATOS

El tipo de datos “serial” es un tipo numérico que se incrementa automáticamente admitiendo el dinamismo para la generación de las claves primarias de las entidades, proporcionando números enteros consecutivos a cada tupla que se registra. En el caso de las claves primarias de las personas, su uso obstruye el curso normal de réplica de datos. Cuando se refiere a la replicación de información surge la tendencia a la duplicación de datos ya existentes, o a la no inclusión de los mismos por existir su llave en el nodo destino. A causa de ello se propone el uso de un mecanismo interno para la generación de las claves, que está basado en una “variante” del md5. Es decir, dado que el md5 es un algoritmo usado para la generación de un texto encriptado, se usa para generar las claves a partir de la información recogida en uno de los campos de la tabla en cuestión, o puede ser a partir de la combinación lineal de varios campos.

Tipo de datos “integer”

Se utiliza para los campos con valores enteros positivos.

Tipo de datos “timestamp”

Se utiliza para los campos donde es guardada la información referente a fechas y tiempos. Dicho tipo no se acompaña de la zona horaria, o sea, en su definición se añade “without time zone”. En el caso de los campos donde se desea describir momentos de inserciones o modificaciones se utiliza la función *now()* para dicho fin.

Tipo de datos “boolean”

Se utiliza para los campos con valores condicionales.

Tipo de datos “varchar(n)”



CAPÍTULO 2. DISEÑO E IMPLEMENTACIÓN DE LA BASE DE DATOS

Los tipos de datos varchar almacenan datos compuestos de caracteres en mayúsculas y minúsculas, como por ejemplo: a, b y C; números como 1, 2 y 3; y caracteres especiales como el signo de arroba (@), "y" comercial (&) y el signo de exclamación de cierre (!). Además este tipo de datos es de longitud variable, n puede ser un valor entre 1 y 8000. En el trabajo con la BD se tiene en cuenta lo siguiente:

- varchar (11) representa los campos del carné de identidad.
- varchar (35) representa campos cortos.
- varchar (50) representa campos medianos.
- varchar (100) representa campos largos.

Tipo de datos "text"

En el caso de los campos que son una descripción o que no se puede restringir la cantidad de caracteres se define como *text*.

2.8 Implementación de la BD

Las tablas con la información correspondiente, no son los únicos objetos que conforman una BD y su existencia no puede estar separada de otros elementos que brindan facilidades para el trabajo con las mismas. El lenguaje *PL/pgSQL* es uno de los lenguajes soportado por el SGBD *PostgreSQL*. Dentro de sus principales características se incluye agrupar bloques de sentencias con una funcionalidad específica dentro de objetos denominados **funciones**, que tienen sus características y para el desarrollo de estas se utilizan estructuras y herramientas incluidas en el propio lenguaje. Otros elementos importantes son los **índices y llaves primarias**, los cuales optimizan el acceso a los datos y mejoran



CAPÍTULO 2. DISEÑO E IMPLEMENTACIÓN DE LA BASE DE DATOS

el rendimiento, acelerando los tiempos de respuesta de las consultas así como otras operaciones. Los **disparadores** son un elemento imprescindible, estos son empleados para mejorar la administración de la BD, obviando la necesidad de contar con que el usuario ejecute la sentencia de *SQL*. Todos estos elementos fueron aplicados en la implementación de la BD del Sistema de Gestión Investigaciones. Posteriormente se fundamenta el empleo de los mismos.

2.8.1 Índices y llaves primarias

El **índice** de una BD es una estructura de datos que mejora la rapidez de las operaciones realizadas y permite un rápido acceso a los registros de una tabla en una BD. Al incrementar de manera drástica la velocidad de acceso, se suelen usar en aquellos campos a los que se les hacen frecuentes búsquedas. Tiene un funcionamiento similar al índice de un libro, dado a que guarda parejas de elementos: el que se desea indexar y su posición en la BD. A la hora de buscar un elemento que esté indexado, sólo hay que buscarlo en el índice, con el objetivo de que cuando sea localizado, se devuelva el registro que se encuentre en la posición marcada por el índice. Los índices pueden ser creados usando una o más columnas, lo que proporciona el punto de partida tanto para búsquedas rápidas al azar, como para un acceso ordenado y eficiente a registros. Existen diversos tipos de índices que se usan dependiendo de las necesidades y requisitos, a continuación se detalla acerca de algunos de estos.

Primary key (del español llave primaria) es el tipo de índice más usado dentro de las BD relacionales. Lo más común es definir una llave primaria de tipo numérico que se incremente de manera automática. Esto garantiza que, independientemente de si las filas preexistentes se eliminan posteriormente, cada fila tiene una única entrada de clave principal. Los valores indexados deben ser



CAPÍTULO 2. DISEÑO E IMPLEMENTACIÓN DE LA BASE DE DATOS

únicos y además no pueden ser nulos. Una tabla solamente puede tener una sola clave primaria. Este tipo de índice es utilizado en todas las tablas de la BD del Sistema de Gestión de Investigaciones, pues cada una de dichas tablas se identifica inequívocamente por su correspondiente llave primaria, en la mayoría de los casos se definen incrementales y en otros se usa un mecanismo interno para la generación de las claves, basado en una variante del md5 a partir de la combinación lineal de varios campos. Cuando se define una columna que no permite valores repetidos se crea implícitamente un índice. **Unique** (del español, único) crea un índice para los cuales, los valores deben ser únicos y diferentes. A la hora de agregar un registro con un valor ya existente aparece un mensaje de error. Permite valores nulos y pueden definirse varios por tabla. En la implementación de la BD del Sistema de Gestión de Investigaciones este tipo de índice se aplica a los atributos que se refieren a los nombres de los registros de las tablas, ejemplo: el atributo nombre_frecuencia de la tabla tb_nfrecuencia del esquema sq_premios. También se utiliza para hacer única la combinación de atributos que referencian llaves foráneas en tablas de relaciones cuando lo requieran. Cuando se crea una llave foránea también se construye un índice. Las llaves foráneas son el resultado de la adición a una tabla de atributos ya existentes, en la mayoría de los casos es una clave primaria de otra tabla. Este tipo de índice se utiliza frecuentemente en la implementación de la BD del Sistema de Gestión de Investigaciones, en consecuencia quedan establecidas de esta forma las relaciones entre tablas.

2.8.2 Funciones y procedimientos almacenados

Las funciones son objetos de la BD que están implementadas por medio de un lenguaje procedural específico, estos objetos pueden facilitar la administración de la BD, así como la visualización de la información contenida en ella.



CAPÍTULO 2. DISEÑO E IMPLEMENTACIÓN DE LA BASE DE DATOS

Estas funciones se pueden ejecutar desde una aplicación. Por otra parte, en el lenguaje definido para su implementación es posible emplear variables, funciones internas del SGBD, así como estructuras para implementar ejecuciones condicionales y ciclos. En la implementación de la BD del Sistema de Gestión de Investigaciones se incluyen varias funciones, en su totalidad imprescindibles para el correcto desarrollo de la aplicación.

Esta es una de las funciones que se implementa para obtener la información de las personas, dicha información se encuentra en la base de datos BD_sgu_nucleo. De igual manera el resto de las funciones se utilizan tanto para enviar como para solicitar información, en este caso, desde la base de datos BD_sgu_investigaciones hacia la BD_sgu_nucleo. En la Tabla 3 se muestran algunas de estas funciones.

Funciones	Esquemas	Detalles
fn_buscar_persona(character varying, character varying, character varying, integer, integer)	sq_master_integracion	Parámetros: criterio, condiciones o filtros, columna de orden, dirección del orden, fila comienzo, cantidad de filas.
fn_buscar_persona_avanzada(character varying, character varying, character varying, integer, integer)	sq_master_integracion	Parámetros: criterio, condiciones o filtros, columna de orden, dirección del orden, fila comienzo, cantidad de filas.
fn_buscar_persona_personalizada(character varying, character varying, character varying, character varying, integer, integer)	sq_master_integracion	Parámetros: atributos, criterio, condiciones o filtros, columna de orden, dirección del orden, fila comienzo, cantidad de filas.



CAPÍTULO 2. DISEÑO E IMPLEMENTACIÓN DE LA BASE DE DATOS

fn_obtener_ciudadania(character varying)	sq_master_integracion	Parámetro: id_persona
fn_obtener_pais(character varying)	sq_master_integracion	Parámetro: id_persona

Tabla 7. Varias funciones de la base de datos.

Partiendo de la idea de unir en una misma solución la gestión de los procesos universitarios, surge la necesidad de integrar los diferentes subsistemas, tales como: el Sistema de Cooperación Internacional, el Sistema de Residencia, el Sistema de Pregrado, el Sistema de Investigaciones, entre otros, en el Sistema de Gestión Universitaria. Después de un análisis a la información que necesita un sistema de otro, debido a la densidad de carga de resultados en la petición de la información de las personas resulta disminuir el rendimiento a nivel de software, por tanto la aplicación se sustenta de una estrategia de integración a nivel de datos para peticiones de la información de personas. Esta estrategia trae en consecuencia algunas dificultades, debido a que en un subsistema se pueden modificar datos que para otros son sensibles, pero aún así son mayores los beneficios que las dificultades que puedan surgir.

La Figura 6 muestra cómo queda estructurada la arquitectura de algunas de las BD del centro, teniendo en cuenta la integración de las mismas.



CAPÍTULO 2. DISEÑO E IMPLEMENTACIÓN DE LA BASE DE DATOS

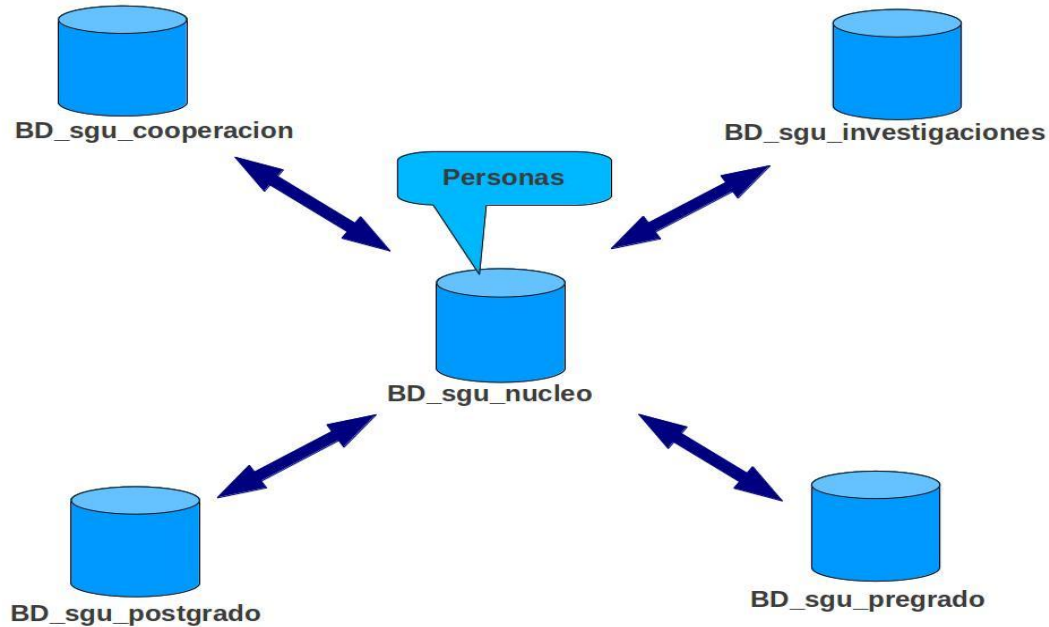


Figura 6. Arquitectura para la integración.

Se dispone del esquema nombrado `sq_gestion_personal` de la base de datos `BD_sgu_nucleo`, esta contiene tablas y campos que permiten el manejo de los datos primarios de las diferentes personas (primer nombre, segundo nombre, primer apellido, segundo apellido, carné de identidad, serie del carné de identidad, entre otros), también datos físicos (color de ojos, color de piel y sexo), información social (ciudadanía y estado civil) y la organización a la que pertenece la persona. En esta BD se almacenan todas las personas que se registran desde cualquier sistema. Es por ello que en la BD del Sistema de Gestión de Investigaciones (`BD_sgu_Investigaciones`) se implementan varias funciones y disparadores que permiten el correcto y eficiente funcionamiento de las BD integradas, las cuales se nutren unas de otras en dependencia de las características de estas. En la base



CAPÍTULO 2. DISEÑO E IMPLEMENTACIÓN DE LA BASE DE DATOS

de datos `BD_sgu_investigaciones`, al realizarse la investigación, se almacenan los datos de las personas que interactúan con los procesos de investigaciones, contando con lo específico de las mismas y lo general, mediante el uso de funciones que permiten unir todos los datos de una persona.

Por lo general, las funciones se encuentran en el esquema `sq_master_integracion` de ambas BD; este esquema está definido por la arquitectura propia con el objetivo de que funcione a modo de túnel o vía principal de conexión a la `BD_sgu_nucleo`. Las funciones de la base de datos de `BD_sgu_investigaciones` llaman a sus funciones homólogas en la base de datos `BD_sgu_nucleo`. Esta funcionalidad se realiza mediante el uso de la tecnología `plproxy`, que se utiliza para hacer llamadas remotas desde una BD a otra; existe otra tecnología que realiza esta funcionalidad con el mismo propósito llamada `dblink`, pero se tiene en cuenta el uso de `plproxy` debido a que esta maneja conexiones automáticas en lo que `dblink` necesita que el usuario las maneje por su propia cuenta, por otra parte, la primera tecnología tiene cierta facilidad para ejecutar una consulta en varios servidores remotos trabajando paralelamente, la segunda también tiene una interfaz que hace esto posible, pero su uso es muy complejo. En fin, la tecnología empleada para hacer llamadas remotas es más eficiente en muchos aspectos que la otra. En la siguiente figura y la Tabla 4 se expone cómo se realiza lo anteriormente expuesto.



CAPÍTULO 2. DISEÑO E IMPLEMENTACIÓN DE LA BASE DE DATOS

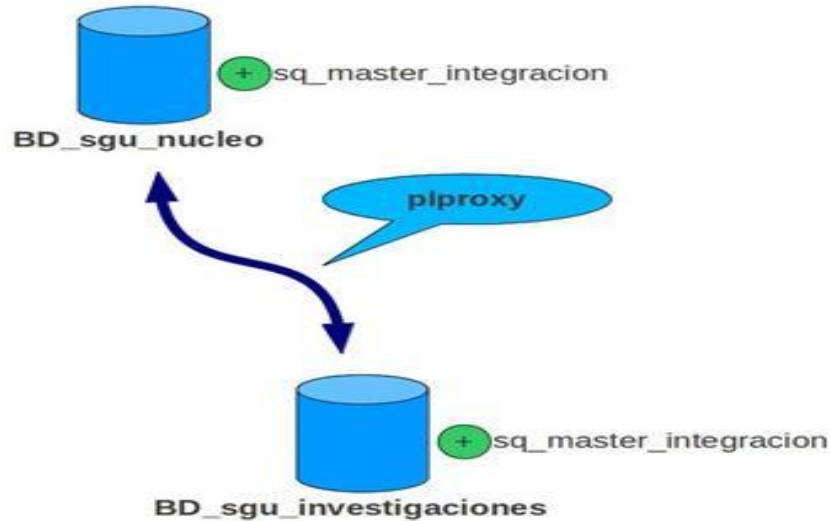


Figura 7. Llamadas remotas entre bases de datos.

Función en la base de datos BD_sgu_nucleo	Función en la base de datos BD_sgu_investigaciones
<pre> CREATE OR REPLACE FUNCTION sq_master_integracion.fn_obtener_ciudadania(character varying) RETURNS SETOF tipo_ciudadania AS \$BODY\$ declare _where text := ""; ex_query text; begin if(length(\$1) > 0) then _where = ' WHERE ' \$1; end if; ex_query = 'SELECT id_ciudadania, nombre_ciudadania, descripcion, fecha_registro, activo FROM sq_gestion_personal.tb_nciudadania ' _where; return QUERY EXECUTE ex_query; end \$BODY\$ </pre>	<pre> CREATE OR REPLACE FUNCTION sq_master_integracion.fn_obtener_ciudadania(c haracter varying) RETURNS SETOF tipo_ciudadania AS \$BODY\$ CONNECT 'dbname=BD_sgu_nucleo'; \$BODY\$ LANGUAGE plproxy VOLATILE COST 100 ROWS 1000; ALTER FUNCTION sq_master_integracion.fn_obtener_ciudadania(c haracter varying) OWNER TO desarrollo; </pre>



CAPÍTULO 2. DISEÑO E IMPLEMENTACIÓN DE LA BASE DE DATOS

```
LANGUAGE plpgsql VOLATILE
COST 100
ROWS 1000;
ALTER FUNCTION
sq_master_integracion.fn_obtener_ciudadania(character
varying) OWNER TO postgres;
```

Tabla 8. Ejemplo de utilización de la tecnología plproxy.

2.8.3 Disparadores

Una de las funcionalidades disponibles en *PostgreSQL* son los denominados disparadores. Un disparador no es más que una acción definida en una tabla de la BD y ejecutada automáticamente por una función. Esta acción se activa, según se especifique, cuando se realice un *INSERT* (insertar), un *UPDATE* (modificar) o un *DELETE* (eliminar) en la propia tabla.

2.9 Conclusiones parciales

En este capítulo se abordó acerca de la metodología usada para el diseño de BD y el proceso de normalización proveniente de la técnica empleada para diseñar bases de datos relacionales. Por otra parte se detallaron aspectos correspondientes a la descripción del soporte arquitectónico, se trataron temas de seguridad y disponibilidad de los datos, estándares de nomenclatura, herramientas usadas y tipos de datos usados en el sistema. Se confeccionó el Modelo Entidad-Relación y se evidenciaron procedimientos almacenados, funciones, disparadores, índices y llaves primarias, todo implementado en la BD, lo cual se tuvo en cuenta para hacer posible el desarrollo de la BD del SGI.



CAPÍTULO 3. VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN

Capítulo 3. Validación de la propuesta de solución

3.1 Introducción

En el siguiente capítulo se exponen temas relacionados con la normalización de la base de datos, así como aspectos relacionados con la validación de la propuesta, con el objetivo de garantizar esto, se tienen en cuenta aspectos referentes a la integridad de los datos, la calidad del dato y el alto rendimiento.

3.2 Normalización de la base de datos

El proceso de normalización ayuda a diseñar una correcta estructura lógica que permita mayor organización y rendimiento en la BD, de modo tal que sea escalable fácilmente. La función de la normalización es favorecer la integridad de los datos, tratar de evitar la posibilidad de introducir datos que no sean razonables. (Rojas Santiesteban & Boloy Boado, 2010).

A la hora de realizar el proceso de normalización en la investigación, se verifica la forma normal de las tablas que existen en la base de datos producto de la integración a la que se ajusta la aplicación, también se chequea que cada entidad cumpla con un conjunto de reglas basadas en la clave primaria, así como las dependencias funcionales parciales con respecto a la llave primaria y las dependencias transitivas de sus atributos no llave respectivamente. Posteriormente a realizar un análisis y llevar a cabo el proceso de normalización se evidencia que el modelo de datos está entre la segunda y la 3FN; esto es debido a que la tabla que se utiliza en el esquema sq_gestion_personal de la BD, nombrada tb_dpersona_data_plus, se encuentra en segunda forma normal a propósito, con el objetivo de agilizar el proceso de búsqueda de información relativa a las personas; esta tabla tiene como objetivo garantizar el trabajo con el



CAPÍTULO 3. VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN

versionado de la información de las personas, de modo que cada vez que se modifique o inserte un dato a una persona en específico este cambio se actualice en la BD núcleo de la arquitectura. Por otra parte en lo que respecta al negocio que responde la BD, el modelo se encuentra en 3FN. A continuación se evidencia con un ejemplo parte del proceso de normalización:

evento(id_evento, estado, alcance, nombre, url, edicion)

Ahora se procede a efectuar los cambios pertinentes para que la relación universal expuesta esté en primera forma normal:

evento(id_evento, nombre, url)

evento_estado(id_estado, nombre, descripcion)

evento_alcance(id_alcance, nombre, descripcion)

edicion_evento(id_edicion, numero, fecha_inicio, fecha_fin)

edicion_nivel(id_nivel, id_edicion)

Ya con este paso, se evidencia que la relación esta en segunda forma normal y en la tercera también, debido a la dependencia de los atributos no llaves respecto a la llave primaria, y la no existencia de dependencias transitivas entre los mismos. Por lo que se puede decir que la relación esta en tercera forma normal finalmente.

3.3 Integridad de los datos

La integridad de los datos se define como un conjunto de reglas, que utilizan la mayoría de las BD relacionales para asegurarse que los registros de tablas relacionadas son válidos, y que no se borren o cambien datos relacionados de forma accidental, produciendo errores de integridad (Carralero Alcalde, 2010).



CAPÍTULO 3. VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN

Algunas de las restricciones presentes en el modelo propuesto se basan en la integridad referencial, integridad de entidad e integridad de dominio.

Se realizan pruebas con el objetivo de demostrar que un buen diseño evita malas prácticas y errores en los datos. Las pruebas de integridad verifican que los datos sean exactos, completos, coherentes y autorizados, estas pruebas nos indican si existen errores en los controles de entrada o procesamiento.

3.3.1 Integridad de entidades

La integridad de una entidad define una fila como entidad única para una tabla específica. Exige la integridad de las columnas de los identificadores o la clave principal de una tabla mediante índices y restricciones UNIQUE (única), o restricciones PRIMARY KEY (llave primaria). En la validación de integridad de las entidades almacenadas, se chequea que todas posean una clave principal, que define a cada registro de modo exclusivo respecto al resto de estos, esto da lugar a que ninguna de las llaves primarias puede tomar valores repetidos o valores nulos, como se puede observar en las figuras que a continuación se muestran.



CAPÍTULO 3. VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN

	id_nivel [PK] serial	descripcion text	activo boolean	fecha_regi timestamp	nombre_ni character v
1	2		TRUE	2012-04-	Nacional
2	4		TRUE	2012-04-	Municipa
3	5		TRUE	2012-04-	Provinci
*	5		TRUE		Provinci

pgAdmin III

Ha ocurrido un error:

ERROR: duplicate key value violates unique constraint "tb_dnivel_pkey"
DETAIL: Key (id_nivel)=(5) already exists.

Aceptar

Figura 8. Error que se emite cuando se inserta una tupla que ya existe.



CAPÍTULO 3. VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN

	id_nivel [PK] serial	descripcion text	activo boolean	fecha_registro timestamp with time zone	nombre_nivel character varying
1	2		TRUE	2012-04-06	Nacional
2	4		TRUE	2012-04-06	Municipal
3	5		TRUE	2012-04-06	Provincial
*	6		TRUE	2012-04-23	

Ha ocurrido un error:
ERROR: null value in column "nombre_nivel" violates not-null constraint

Aceptar

Figura 9. Ejemplo de error que se emite cuando se insertan valores nulos no permitidos

3.3.2 Integridad referencial.

La integridad referencial es un sistema de reglas que utilizan la mayoría de las bases de datos relacionales para asegurarse que los registros de tablas relacionadas son válidos y que no se borren o cambien datos relacionados de forma accidental produciendo errores de integridad (aulaclic, 2000). Contempla la integridad de las relaciones entre varias entidades de la BD, específicamente cuando se crean, modifican o eliminan filas completas. Esta es la categoría más importante, dado que determina la consistencia de los datos existentes en cada tabla. Se logra a través de las restricciones de *FOREIGN KEY* (Llave foránea, en español) y los modos de propagación de cambios.

Con las validaciones de integridad referencial se busca asegurar que las filas relacionadas entre tablas no dejen de estarlo, o cambie esta relación cuando se



CAPÍTULO 3. VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN

modifiquen los datos correspondientes. Con este tipo de integridad se limita la actividad que puede realizar un usuario sobre la BD. En la siguiente figura se muestra un error referente a una situación que se evita con este tipo de integridad.

The screenshot shows the pgAdmin III interface. At the top, there is a toolbar with various icons and a dropdown menu set to 'Sin límite'. Below the toolbar is a table with the following data:

	id_alcance [PK] serial	id_tipo_alcance integer	codigo_pais integer
1	6	1	
2	7	1	
*	8	2	

Below the table, an error dialog box is displayed. The dialog box has a title bar that says 'pgAdmin III' and a red 'X' icon. The main text of the dialog box reads:

Ha ocurrido un error:
ERROR: insert or update on table "tb_dalcance" violates foreign key constraint "tb_dalcance_id_tipo_alcance_fkey"
DETAIL: Key (id_tipo_alcance)=(2) is not present in table "tb_nalcance_evento".

At the bottom right of the dialog box, there is an orange button labeled 'Aceptar'.

Figura 10. Ejemplo de error que se emite cuando se viola la integridad referencial.

3.3.3 Integridad de dominio

La integridad de dominio está dada por la validez de las entradas para una columna determinada. Las pruebas de integridad de dominio sirven para controlar la información que se guarda en la BD, verificando que los datos que se insertan



CAPÍTULO 3. VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN

están acordes con sus definiciones. Garantizan también que los datos tienen un valor legítimo. Las validaciones son a nivel de campo, es por ello que se toman varios atributos, todos referentes a distintos dominios de datos, y se intenta introducir valores de tipos diferentes a los definidos para comprobar que se garantiza la integridad de dominio. En la Figura 11 se muestra lo ocurrido en este caso.

The screenshot shows the pgAdmin III interface. At the top, there is a toolbar with various icons and a dropdown menu set to 'Sin límite'. Below the toolbar is a table with the following data:

	id_nivel [PK] serial	descripcion text	activo boolean	fecha_registro timestamp	nombre_nivel character v
1	2		TRUE	2012-04-	Nacional
2	4		TRUE	2012-04-	Municipa
3	5		TRUE	2012-04-	Provinci
*	e		TRUE		Provinci

Below the table, an error dialog box is displayed with the following text:

```
Ha ocurrido un error:  
ERROR: invalid input syntax for integer: "e"  
LINE 1: ...tb_nnivel(id_nivel, activo, nombre_nivel)  
VALUES ('e':integ...
```

The dialog box has an 'Aceptar' button at the bottom right.

Figura 11. Ejemplo de error que se emite cuando se viola la integridad de dominio.

3.4 Calidad del dato.

La calidad del dato es el grado en que los datos corresponden con la verdadera información. Las dimensiones de la calidad del dato vienen dadas por:



CAPÍTULO 3. VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN

- **Exactitud:** mide el grado en que la información refleja lo que está pasando con el evento, o sea, el dato es correcto para lo que está representando.
- **Integralidad:** es el grado en que las BD cuentan con toda la información crítica requerida para el evento.
- **Oportunidad:** que esté disponible cuando se requiere para tomar una decisión, o sea que el dato esté actualizado y la fecha de actualización es apropiada para la tarea a realizar.
- **Consistencia:** el dato pasa todos los controles de aceptabilidad del formato, la longitud, entre otras características.

Después de analizar estos factores, y ejecutados en la investigación, se puede decir que la aplicación cumple con la calidad del dato. Para la realización del diseño y la implementación de la BD del Sistema de Gestión de Investigaciones se hicieron reuniones y entrevistas con las analistas del sistema, donde se tuvo en cuenta los requisitos funcionales y las reglas del negocio, quedando bien identificados las tablas, relaciones y campos necesarios para el almacenamiento de toda la información con la precisión necesaria. En la actualidad, la aplicación en desarrollo trabaja con la BD, lo que evidencia que la misma refleja los rasgos necesarios que aseguran la calidad del dato.

3.5 Alto rendimiento.

A la hora de comprobar el alto rendimiento que posee el diseño de la BD desarrollada se realiza una prueba de manera funcional del mismo. Para esta prueba se configura un servidor con el SGBD utilizado en la investigación, PostgreSQL, y para la carga de datos hacia las tablas de la base de datos se utiliza el EMS Data Generator for PostgreSQL 2010, esta herramienta posibilita la generación de datos para una o varias tablas concurrentemente. A través de esta aplicación se verifica la integridad referencial de los datos que se quieren



CAPÍTULO 3. VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN

introducir, dado a que permite para cada atributo, especificar los valores admisibles en dependencia del tipo de dato que está previamente definido. Por otra parte permite introducir la cantidad de tuplas que se desean generar para la realización de la prueba.

Se generaron 167 tuplas entre las tablas `tb_npremio_resultado`, `tb_dedicion_premio`, `tb_dpremio` y `tb_dalcance`. Este llenado de las tablas, con una restricción de 100 tuplas por tabla, se realizó en un aproximado a 45 segundos. Posteriormente se realizaron algunas de las más comunes e importantes consultas que se le realizan frecuentemente a la BD, estas consultas fueron hechas en la herramienta pgAdminIII, con la idea de comprobar el correcto funcionamiento de dicha base de datos, así como los tiempos de ejecución de las mismas.

Ejemplo: Esta consulta devuelve todos los resultados de las ediciones de un premio, tanto con alcance nacional como internacional:

```
SELECT tb_npremio_resultado.id_edicion_premio,  
       tb_npremio_resultado.nombre_premio_resultado,  
       tb_npremio_resultado.descripcion  
FROM sq_premios.tb_npremio_resultado JOIN sq_premios.tb_dedicion_premio  
ON sq_premios.tb_npremio_resultado.id_edicion_premio =  
sq_premios.tb_dedicion_premio.id_edicion_premio  
JOIN sq_premios.tb_dpremio ON sq_premios.tb_dedicion_premio.id_premio =  
sq_premios.tb_dpremio.id_premio  
JOIN sq_premios.tb_nalcance ON sq_premios.tb_dpremio.id_alcance =  
sq_premios.tb_nalcance.id_alcance;
```



CAPÍTULO 3. VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN

La consulta mostrada anteriormente devuelve un resultado de 20 filas en 78 milisegundos.

Otro ejemplo de una consulta común es la siguiente, la cual muestra los premios a una publicación dada, en dependencia de una edición de este premio especificada:

```
SELECT tb_dpremio_publicacion.id_edicion_premio
FROM sq_premios.tb_dpremio_publicacion JOIN sq_premios.tb_dedicion_premio
ON sq_premios.tb_dpremio_publicacion.id_edicion_premio =
sq_premios.tb_dedicion_premio.id_edicion_premio;
```

Esta consulta devolvió un total de 26 filas en 56 milisegundos.

Los juegos de datos insertados en la BD son un estimado bastante aceptable en dependencia a la aproximación real de rendimiento a la que se someterá la aplicación. En el momento en que el Sistema de Gestión de Investigaciones se comience a utilizar, se deben tener en cuenta la cantidad de usuarios que estarán conectados al sistema y el grado de concurrencia de las solicitudes realizadas por estos. Partiendo de esto la mejor prueba a la que es sometida la base de datos es la interacción con el usuario final, los resultados que se obtienen de la misma pueden servir en la realización de nuevas versiones de la base de datos buscando mejoras, o resolver posibles deficiencias que se puedan detectar durante su funcionamiento.

3.6 Conclusiones parciales.

En este capítulo se realizaron pruebas de integridad al modelo de datos desarrollado. De esta forma se demostró la presencia en la aplicación en cuanto a



CAPÍTULO 3. VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN

la integridad referencial, de dominio y de entidad, lo cual evita la inconsistencia de los datos. También, se efectuó la comprobación de la calidad del dato y el alto rendimiento del diseño de la BD del Sistema de Gestión de Investigaciones.



Conclusiones generales

Se pudieron identificar las necesidades de información y las mejoras necesarias a tener en cuenta a la hora de realizar el diseño de la base de datos del Sistema de Gestión de Investigaciones. Con el análisis bibliográfico y la caracterización del estado actual de los SGBD, se fundamentó la utilización del SGBD empleado para la manipulación de los datos relativos a los procesos de investigación, lo que facilitó la profundización de los conocimientos acerca del SGBD PostgreSQL. Posteriormente se obtuvo el modelo de datos del sistema, el cual dio lugar a un diseño funcional que comprende toda la información gestionada en el sistema. Se logró implementar la base de datos del Sistema de Gestión de Investigaciones y en consecuencia, el acceso a la base de datos para el desarrollo de la aplicación web de dicho sistema. Con la comprobación de la integridad de los datos, el análisis del diseño de la base de datos del SGI en cuanto a alto rendimiento y calidad del dato, se logró obtener resultados de la misma y esto permitió valorar cierto grado de eficiencia del diseño.



Recomendaciones

- ✓ Continuar con la mejora del proceso de modelado de la base de datos e incluir esta en la misma para lograr una mejor gestión de los datos del sistema.
- ✓ Realizar la implantación de la base de datos que se propone con un enfoque de mejora continua, para garantizar la flexibilidad del diseño en cuanto a la incorporación de nuevas formas organizativas.
- ✓ Realizar pruebas de stress a la BD para analizar su funcionamiento ante algunas condiciones anormales que puedan producirse.



Referencias bibliográficas

Aulacltic (2000). Recuperado el 10 de marzo de 2012, de

<http://www.aulacltic.es/sql/b811.htm>

Barranco de Areba, J. (2001). *Metodología del análisis estructurado de sistemas*.

Univ Pontifica de Comillas.

Carralero Alcalde, A. L. (2010). *Diseño e implementación de la base datos del*

Sistema Unificado de la Fuerza de Trabajo Calificada. Ciudad de La Habana:

Universidad de las Ciencias Informáticas.

CAVSI. (2008). Recuperado el 20 de octubre de 2011, de

<http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sqbd/>

Conferencia de Ingeniería de Software. (2011). *Introducción a la Disciplina de*

Requisitos de RUP (pág. 5). La Habana: Universidad de las Ciencias Informáticas.

Date, C. J. (2001). *Introducción a los Sistemas de Bases de Datos*. Méjico:

UNIVERSIDAD FEDERICO SANTA MARIA.

EcuRed. (2008). Recuperado el 4 de abril de 2011, de

http://www.ecured.cu/index.php/Visual_Paradigm

EVA. (2011). Recuperado el 10 de noviembre de 2011, de Conferencia de

normalización:

http://eva.uci.cu/file.php/624/2._Clases/Semana_5/2da_frecuencia/MApoyo/C4Normaliz.pdf.



REFERENCIAS BIBLIOGRÁFICAS

EVA. (2012). Recuperado el 24 de abril de 2012, de http://eva.uci.cu/file.php/180/2._Clases/Semana_2/1ra_frecuencia/MApoyo/Patrones_de_diseño_de_BD.pdf

Fundora Carrasco (2011). Diseño e Implementación de la base de datos del Sistema de Gestión de Postgrado. Ciudad de La Habana.

Informática Full. (2009). Recuperado el 29 de noviembre de 2011, de <http://informatica-full2.blogspot.com/2009/05/evolucion-de-los-sgbd.html>

Martínez, L. (2007). Monografias.com. Recuperado el 27 de noviembre de 2011, de Monografias.com.SA: <http://www.monografias.com/trabajos37/arquitectura-de-sistemas/arquitectura-de-sistemas.shtml>

Masip, D. (2002). Qué es Oracle. DesarrolloWeb. Recuperado el 21 de noviembre de 2011, de <http://www.desarrolloweb.com/articulos/840.php>

Monografias.com. (2006). Recuperado el 23 de enero de 2011, de <http://www.monografias.com/trabajos80/sistemas-gestion-bases-datos>

PgAdmin PostgreSQL Tools. (2008). Recuperado el 5 de marzo de 2012, de Introduction pgAdmin III: <http://www.pgadmin.org/docs/dev/using.html>

PostgreSQL 9.0.1 Documentation.

PostgreSQL-Es. (2009). Recuperado el 20 de enero de 2012, de http://www.postgresql-es.org/sobre_postgresql

Pozo Coronado, S. (2005). MySQL con Clase. Recuperado el 20 de febrero de 2012, de Gestión de bases de datos: http://www.angelfire.com/droid/jacksman/curso/mysql/curso_mysql.pdf



REFERENCIAS BIBLIOGRÁFICAS

Rojas Santiesteban, E., & Boloy Boado, A. E. (2010). *Arquitectura y diseño de la Base de Datos Única de los Sistemas de Identificación, Inmigración y Extranjería de la República de Cuba*. Ciudad de La Habana.

Tutorialesenlared.com. (2006). Introducción a los sistemas de bases de datos.

Recuperado el 20 de septiembre de 2011, de

<http://www.tutorialesenlared.com/manual3202.html>

Universidad de Belgrano. (2006). Recuperado el 5 de diciembre de 2011, de

<http://www.ub.edu.ar/catedras/ingenieria/Datos/capitulo1/cap16.htm>



Glosario de términos

1. **Base de datos:** desde el punto de vista informático, la base de datos es un sistema formado por un conjunto de datos almacenados en discos que permiten el acceso directo a ellos y un conjunto de programas que manipulen ese conjunto de datos.
2. **Base de datos relacional:** es una base de datos en donde todos los datos visibles al usuario están organizados estrictamente como tablas de valores, y en donde todas las operaciones de la base de datos se realizan sobre estas tablas. Estas bases de datos son percibidas por los usuarios como una colección de relaciones normalizadas de diversos grados que varían con el tiempo.
3. **Clave primaria:** conjunto de atributos de su esquema que son elegidos para servir de identificador unívoco de sus tuplas.
4. **Claves foráneas:** expresan relaciones entre objetos representados, incluyendo en el esquema de una relación atributos de otra. Utilizado para relacionar tablas.
5. **Dependencia funcional:** es una relación entre atributos de una misma relación (tabla).
6. **Entidad:** se refiere a cualquier concepto del mundo real con una existencia independiente.
7. **Gestor de base de datos:** Es un tipo de software específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.



GLOSARIO DE TÉRMINOS

8. **Integridad:** En términos de datos se refiere a la corrección y completitud de los datos en una base de datos.
9. **Querys:** consiste en una cadena de consulta, normalmente se utilizan para: insertar, actualizar o editar valores de la base de datos.
10. **Servidor:** en informática, un servidor es un tipo de software que realiza ciertas tareas en nombre de los usuarios. El término servidor ahora también se utiliza para referirse al computador en el cual funciona ese software, una máquina cuyo propósito es proveer datos de modo que otras máquinas puedan utilizar esos datos.



Anexo 1. Nomencladores de la base de datos

Objeto	Prefijo	Observación	Ejemplo
Base de Datos	BD_sgu_<texto>		BD_sgu_Investigaciones
Esquemas	sq_+<texto>		sq_premios
Tabla de datos	tb_d+<texto>		tb_dpersona
Tabla de nomencladores	tb_n+<texto>		tb_nestado
Tablas de relación	tb_r+<texto>	Usualmente se usan para interrelacionar conceptos o claves de negocio, y con dicha relación representar otros conceptos o bien reglas de negocio.	tb_rpremio_persona
Campo de llave primaria	id_+<texto>	El texto debe estar refiriendo el concepto o clave de negocio representado	id_premio



		en esta tabla.	
Campo de llave foránea	<campo_referenciado>	Clave foránea	En una tabla que hereda de persona como la tabla participante, el campo que referencia la tabla persona se nombre: id_persona.
Campo de nombre	Nombre_+<texto>	El texto debe estar refiriendo el concepto o clave de negocio representado en esta tabla	nombre_evento
Campo de descripción	Descripción		descripción
Campo de activación	activo	Son campos usados en la estrategia conservadora de la no eliminación de información.	activo
Campos de fecha de registros	fecha_registro	Son campos usados en la estrategia de auditorías a la	fecha_registro



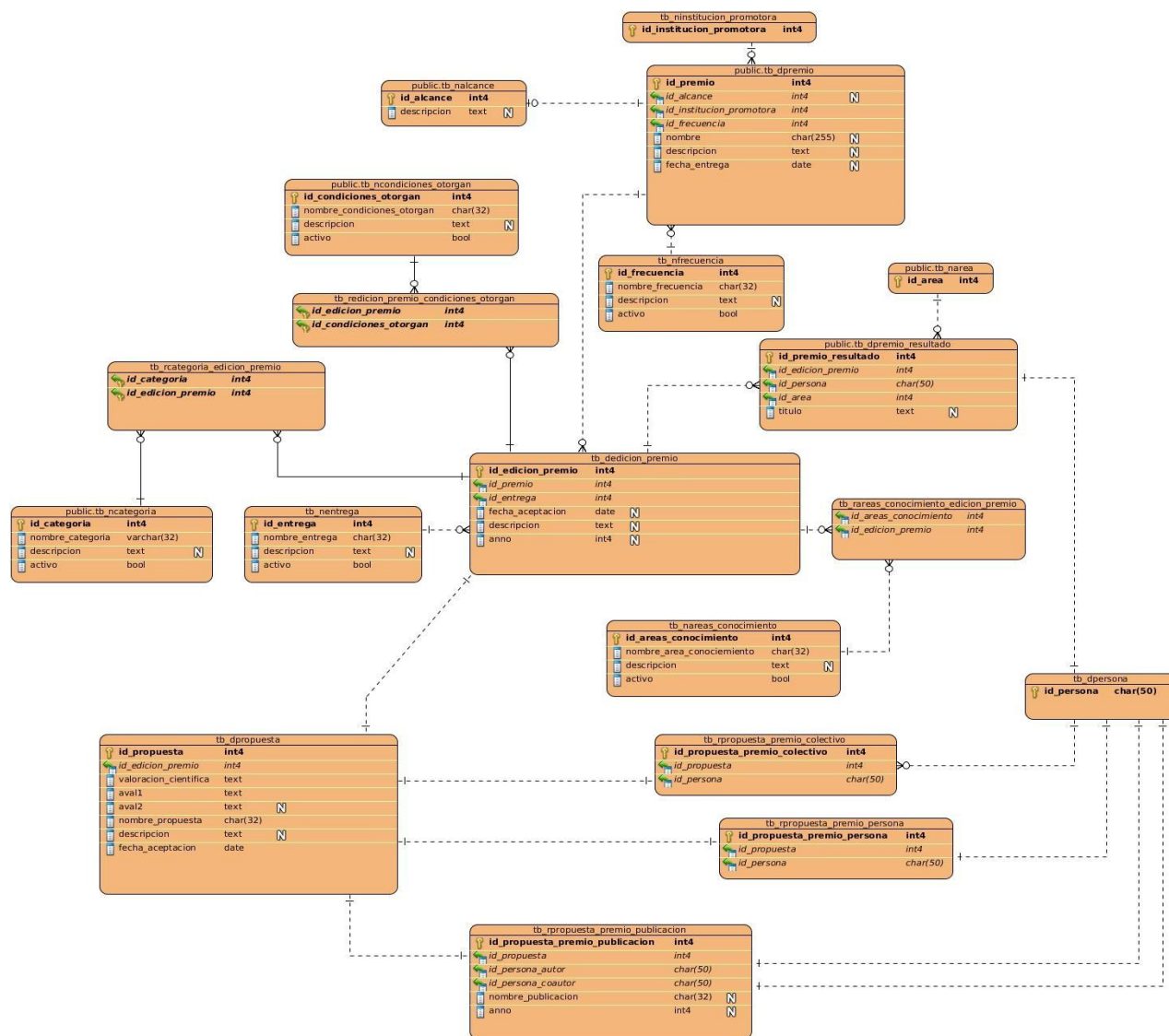
ANEXOS

		información.	
Función	fn_+<funcionalidad> <entidad>_<operación>		fn_obtener_país
Disparador	tr_+<funcionalidad> <entidad>_<operación>	trigger	



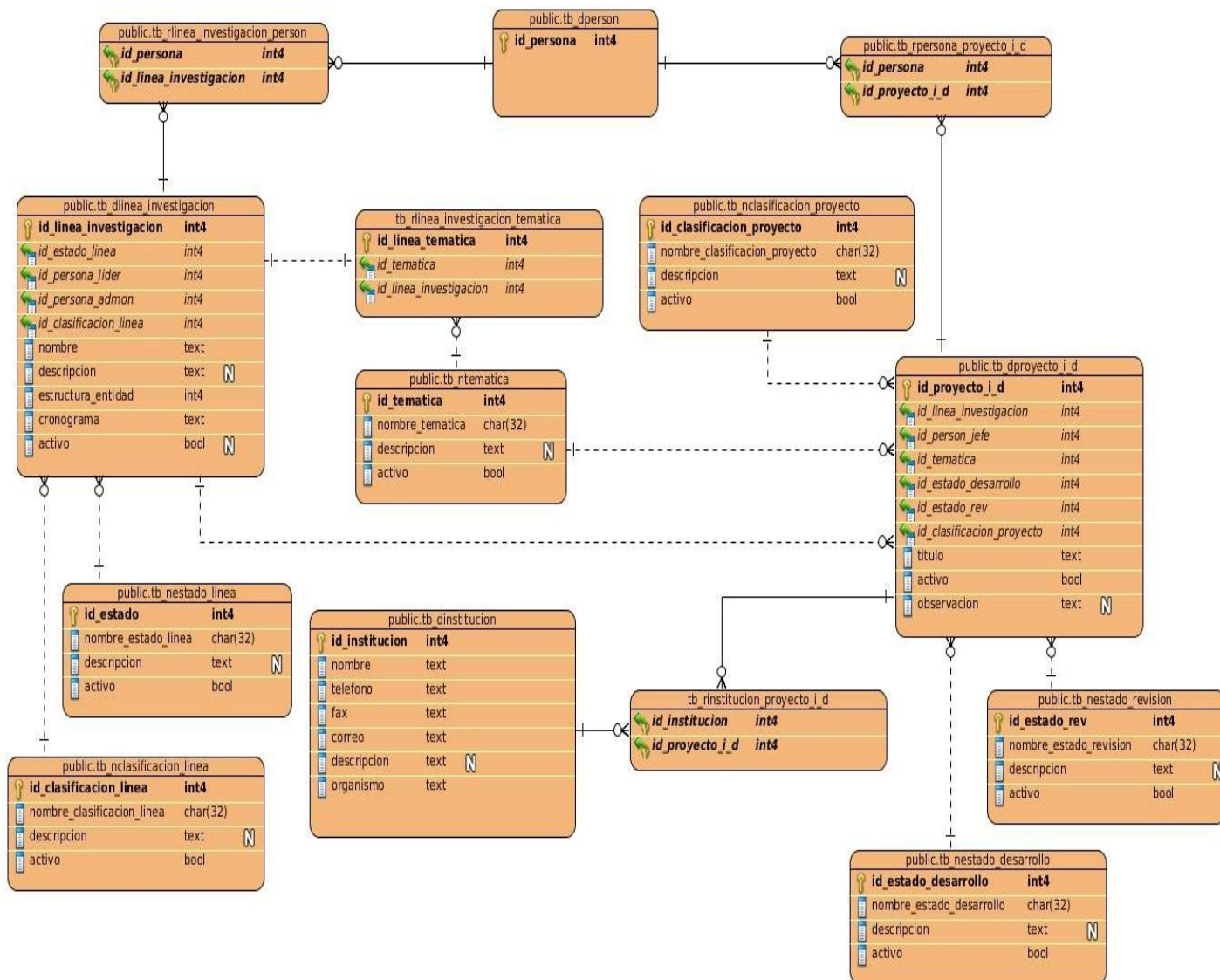
Anexo 2. Diagrama Entidad-Relación de la BD.

Módulo premios del esquema físico:





Módulo líneas y proyectos del esquema físico:





Anexo 3. Descripción de las tablas de la BD.

Modulo eventos:

Nombre: tb_dedicion		
Descripción: Almacena los datos de una edición de un evento.		
Atributo	Tipo	Descripción
id_evento	Entero	Identificador del evento al que pertenece.
id_organizacion	Entero	Identificador de la organización.
numero	Entero	Número de la edición.
contacto_organizadores	Texto	Datos de contacto de los organizadores.
memoria	Boolean	Si la edición contiene memoria.
codigo_pais	Entero	Código del país donde se crea la edición
activa	Boolean	Si está activa la edición.
fecha_aceptacion	Fecha	Fecha en que se acepta el evento.
fecha_inicio	Fecha	Fecha en que se inicia el evento.
fecha_fin	Fecha	Fecha en que culmina el evento.
ciudad	Cadena	Ciudad en que se desarrolla el evento.

Tabla 9. Tabla de edición.



Nombre: tb_dtaller		
Descripción: Almacena los talleres que se desarrollen.		
Atributo	Tipo	Descripción
id_taller	Entero	Identificador del taller.
id_edicion	Entero	Identificación de la edición perteneciente.
nombre	Cadena	Nombre del taller.
fecha_inicio	Fecha	Fecha en que se inicia el taller.
fecha_fin	Fecha	Fecha en que culmina el taller.

Tabla 10. Tabla de taller.

Nombre: tb_dtrabajo		
Descripción: Almacena los trabajos relacionados a los eventos.		
Atributo	Tipo	Descripción
id_trabajo	Entero	Identificador del trabajo.
id_taller	Entero	Identificador del taller perteneciente.
id_edicion	Entero	Identificador de la edición perteneciente.
id_tipo_premio	Entero	Identificador del tipo de premio.
id_persona_autor_principal	Entero	Identificador del autor principal.
trabajo	Texto	Trabajo a almacenar.



titulo	Cadena	Titulo del trabajo.
--------	--------	---------------------

Tabla 11. Tabla de trabajo.

Nombre: tb_redicion_nivel		
Descripción: Almacena los niveles de las ediciones de un evento.		
Atributo	Tipo	Descripción
id_edicion_nivel	Entero	Identificador de la relación.
id_edicion	Entero	Identificador de la edición perteneciente.
id_nivel	Entero	Identificador del nivel perteneciente.
taller	Boolean	Especifica si un nivel tiene talleres.

Tabla 12. Tabla de relación edición-nivel.

Nombre: tb_redicion_taller		
Descripción: Almacena los talleres y trabajos de una edición de un evento.		
Atributo	Tipo	Descripción
id_edicion_taller	Entero	Identificador de la relación.
id_edicion	Entero	Identificador de la edición.
id_taller	Entero	Identificador del taller.
id_trabajo	Entero	Identificador del trabajo.

Tabla 13. Tabla de relación edición-taller.



Nombre: tb_redicion_nivel_taller		
Descripción: Almacena los talleres y trabajos de un nivel de una edición de un evento.		
Atributo	Tipo	Descripción
id_edicion_nivel_taller	Entero	Identificador de la relación.
id_nivel	Entero	Identificador del nivel perteneciente.
id_taller	Entero	Identificador del taller que contenga.
id_trabajo	Entero	Identificador del trabajo que contenga.

Tabla 14. Tabla de relación edición nivel-taller.

Nombre: tb_redicion_nivel_trabajo		
Descripción: Almacena los trabajos de un nivel de una edición de un evento.		
Atributo	Tipo	Descripción
id_edicion_nivel_trabajo	Entero	Identificador de la relación.
id_edicion_nivel	Entero	Identificador del nivel perteneciente.
id_trabajo	Entero	Identificador del trabajo que contenga.

Tabla 15. Tabla de relación edición nivel-trabajo.



Nombre: tb_nestado_evento		
Descripción: Almacena los estados de un evento.		
Atributo	Tipo	Descripción
id_estado_evento	Entero	Identificador del estado de un evento.
nombre_estado_evento	Cadena	Nombre del estado de un evento.
descripcion	Texto	Descripción del estado de un evento.
activo	Boolean	Si está activo el estado.

Tabla 16. Tabla de estado de evento.

Nombre: tb_alcance_evento		
Descripción: Almacena los estados de un evento.		
Atributo	Tipo	Descripción
id_alcance_evento	Entero	Identificador del alcance de un evento.
nombre_alcance_evento	Cadena	Nombre del alcance de un evento.
descripcion	Texto	Descripción del alcance de un evento.



ANEXOS

activo	Boolean	Si está activo el alcance.
--------	---------	----------------------------

Tabla 17. Tabla de alcance de evento.

Nombre: tb_dnivel		
Descripción: Almacena los niveles de una edición.		
Atributo	Tipo	Descripción
id_nivel	Entero	Identificador del nivel de una edición.
nombre_nivel	Cadena	Nombre del nivel de una edición.
descripcion	Texto	Descripción del nivel de una edición.
activo	Boolean	Si está activo el nivel.

Tabla 18. Tabla de nivel.

Nombre: tb_norganizacion		
Descripción: Almacena las organizaciones pertenecientes de una edición de un evento.		
Atributo	Tipo	Descripción
id_organizacion	Entero	Identificador de la organización.
nombre_organización	Cadena	Nombre de la organización.
descripcion	Texto	Descripción de la organización.
activo	Boolean	Si está activa una organización.

Tabla 19. Tabla de organización.



Nombre: tb_rpersona_trabajo		
Descripción: Almacena los trabajos de las personas.		
Atributo	Tipo	Descripción
id_persona_trabajo	Entero	Identificador de la relación.
id_trabajo	Entero	Identificador del trabajo que tenga.
id_persona	Entero	Identificador de la persona.

Tabla 20. Tabla de relación persona-trabajo.

Nombre: tb_ntipo_premio		
Descripción: Almacena los tipos de premios que pueden tener un trabajo.		
Atributo	Tipo	Descripción
id_tipo_premio	Entero	Identificador del tipo de premio.
nombre_tipo_premio	Cadena	Nombre del tipo de premio.
descripcion	Texto	Descripción del tipo de premio.
activo	Boolean	Si está activo el tipo de premio.

Tabla 21. Tabla de tipo de premio.



Módulo premios:

Nombre: tb_dedicion_premio		
Descripción: Almacena las ediciones de un premio.		
Atributo	Tipo	Descripción
id_edicion_premio	Entero	Identificador de la edición de un premio
id_premio	Entero	Identificador del premio al que pertenece.
id_entrega	Entero	Identificador del tipo de entrega.
fecha_aceptacion	Fecha	Fecha en que se acepta una edición.
descripcion	Texto	Descripción de la edición.
anno	Entero	Año de la edición.

Tabla 22. Tabla edición.

Nombre: tb_dpremio_resultado		
Descripción: Almacena los resultados de una edición de un premio.		
Atributo	Tipo	Descripción
id_premio_resultado	Entero	Identificador del resultado.
id_edicion_premio	Entero	Identificador de la edición.
id_persona	Entero	Identificador de la persona premiada.
id_area	Entero	Identificador del área.



titulo	Cadena	Título del resultado de un premio.
--------	--------	------------------------------------

Tabla 23. Tabla premio resultado.

Nombre: tb_rpropuesta_premio_publicacion		
Descripción: Almacena una propuesta de un premio a una publicación.		
Atributo	Tipo	Descripción
id_propuesta	Entero	Identificador de la propuesta.
id_persona_autor	Entero	Identificador del autor de la publicación.
id_persona_coautor	Entero	Identificador del coautor de la publicación.
nombre_publicacion	Cadena	Nombre de la publicación.
anno	Entero	Año en que se crea la publicación.

Tabla 24. Tabla relación propuesta-premio publicación.

Nombre: tb_rpropuesta_premio_colectivo		
Descripción: Almacena las propuestas a un premio colectivo.		
Atributo	Tipo	Descripción
id_propuesta_premio_colectivo	Entero	Identificador de la relación.
id_propuesta	Entero	Identificador de la propuesta.
id_persona	Cadena	Identificador de las personas premiadas.

Tabla 25. Tabla premio colectivo.



Nombre: tb_rpropuesta_premio_persona		
Descripción: Almacena las propuestas a un premio colectivo.		
Atributo	Tipo	Descripción
id_propuesta_premio_persona	Entero	Identificador de la relación.
id_propuesta	Entero	Identificador de la propuesta.
id_persona	Cadena	Identificador de la persona premiada.

Tabla 26. Tabla premio persona.

Nombre: tb_dpropuesta		
Descripción: Almacena las propuestas de una edición.		
Atributo	Tipo	Descripción
id_propuesta	Entero	Identificador de la propuesta.
id_edicion_premio	Entero	Identificador de la edición perteneciente.
nombre_propuesta	Cadena	Nombre de la propuesta.
descripcion	Texto	Descripción de la propuesta.
fecha_aceptacion	Fecha	Fecha en que se acepta la propuesta.
valoración_científica	Texto	Documento de valoración científica.
aval	Texto	Documento de avales de la propuesta.

Tabla 27. Tabla propuesta.



Nombre: tb_rcategoria_edicion_premio		
Descripción: Almacena las categorías de una edición de un premio.		
Atributo	Tipo	Descripción
id_categoria	Entero	Identificador de la categoría.
id_edicion_premio	Entero	Identificador de la edición del premio.

Tabla 28. Tabla relación categoría edición-premio.

Nombre: tb_dcategoria		
Descripción: Almacena las categorías de una edición de un premio.		
Atributo	Tipo	Descripción
id_categoria	Entero	Identificador de la categoría.
nombre_categoria	Cadena	Nombre de la categoría.
descripcion_categoria	Texto	Descripción de la categoría.

Tabla 29. Tabla categoría.

Nombre: tb_nalcance		
Descripción: Almacena el alcance de un premio.		
Atributo	Tipo	Descripción
id_alcance	Entero	Identificador del alcance.



ANEXOS

nombre_alcance	Cadena	Nombre del alcance.
descripcion	Texto	Descripción del alcance.
activo	Boolean	Si está activo el alcance.

Tabla 30. Tabla alcance.

Nombre: tb_nintitucion_promotora		
Descripción: Almacena la institución promotora de un premio.		
Atributo	Tipo	Descripción
id_intitucion_promotora	Entero	Identificador de la institución.
nomobre_intitucion_promotora	Cadena	Nombre de la institución.
descripcion	Texto	Descripción de la institución promotora.
activo	Boolean	Si está activa la institución promotora.

Tabla 31. Tabla institución promotora.

Nombre: tb_nfrecuencia		
Descripción: Almacena la frecuencia de un premio.		



Atributo	Tipo	Descripción
id_frecuencia	Entero	Identificador de la frecuencia.
nombre_frecuencia	Cadena	Nombre de la frecuencia.
descripcion	Texto	Descripción de la frecuencia.
activo	Boolean	Si está activa la frecuencia.

Tabla 32. Tabla frecuencia.

Nombre: tb_ncondiciones_otorgan		
Descripción: Almacena las condiciones que se otorgan a una edición de un premio.		
Atributo	Tipo	Descripción
id_condiciones_otorgan	Entero	Identificador de las condiciones otorgadas.
nombre_condiciones_otorgan	Cadena	Nombre de las condiciones otorgadas.
descripcion	Texto	Descripción de las condiciones otorgadas.
activo	Boolean	Si está activa la condición otorgada.

Tabla 33. Tabla condiciones que se otorgan.

Nombre: tb_redicion_premio_condiciones_otorgan
Descripción: Almacena los identificadores de la relación.



Atributo	Tipo	Descripción
id_edicion_premio	Entero	Identificador de la edición de un premio.
idcondiciones_otorgan	Entero	Identificador de la condición otorgada.

Tabla 34. Tabla relación edición premio-condiciones otorgan.

Nombre: tb_rareas_conocimiento_edicion_premio		
Descripción: Almacena los identificadores de la relación.		
Atributo	Tipo	Descripción
id_areas_conocimiento	Entero	Identificador del área del conocimiento.
id_edicion_premio	Entero	Identificador de la edición de un premio.

Tabla 35. Tabla relación aéreas conocimiento-edición premio.

Nombre: tb_nentrega		
Descripción: Almacena los datos de una entrega de una edición de un premio.		
Atributo	Tipo	Descripción
id_entrega	Entero	Identificador de la entrega.
nombre_entrega	Cadena	Nombre de la entrega.
descripcion	Texto	Descripción de la entrega.
activo	Boolean	Si está activa la entrega.

Tabla 36. Tabla entrega.



Nombre: tb_narea		
Descripción: Almacena los datos de un área en la que se encuentre el resultado de un premio.		
Atributo	Tipo	Descripción
id_area	Entero	Identificador del área.
nombre_area	Cadena	Nombre del área.
descripcion	Texto	Descripción del área.
activo	Boolean	Si está activa el área.

Tabla 37. Tabla área.

Nombre: tb_nareas_conocimiento		
Descripción: Almacena los datos de un área del conocimiento de una edición de un premio.		
Atributo	Tipo	Descripción
id_area_conocimiento	Entero	Identificador del área del conocimiento.
nombre_area_conocimiento	Cadena	Nombre del área del conocimiento.
descripcion	Texto	Descripción del área del conocimiento.



activo	Boolean	Si está activa el área del conocimiento.
--------	---------	--

Tabla 38. Tabla áreas del conocimiento.

Módulo líneas y proyectos:

Nombre: tb_rlinea_investigacion_persona		
Descripción: Almacena los identificadores de la relación.		
Atributo	Tipo	Descripción
id_persona	Entero	Identificador de la persona.
id_linea_investigacion	Entero	Identificador de la línea de investigación.

Tabla 39. Tabla relación línea de investigación-persona.

Nombre: tb_rpersona_proyecto_investigacion		
Descripción: Almacena los identificadores de la relación.		
Atributo	Tipo	Descripción
id_persona	Entero	Identificador de la persona.
id_proyecto_investigacion	Entero	Identificador del proyecto de investigación.

Tabla 40. Tabla relación persona-proyecto de investigación.



Nombre: tb_rlinea_investigacion_tematica		
Descripción: Almacena los identificadores de la relación.		
Atributo	Tipo	Descripción
id_tematica	Entero	Identificador de la temática.
id_linea_investigacion	Entero	Identificador de la línea de investigación.

Tabla 41. Tabla relación línea de investigación-temática.

Nombre: tb_rinstitucion_proyecto_investigacion		
Descripción: Almacena los identificadores de la relación.		
Atributo	Tipo	Descripción
id_institucion	Entero	Identificador de la institución.
id_proyecto_investigacion	Entero	Identificador del proyecto de investigación.

Tabla 42. Tabla relación institución-proyecto de investigación.

Nombre: tb_dinstitucion		
Descripción: Almacena los datos de una institución.		
Atributo	Tipo	Descripción
id_institucion	Entero	Identificador de la institución.
nombre	Texto	Nombre de la institución.



telefono	Texto	Teléfono de la institución.
fax	Texto	Fax de la institución.
correo	Texto	Correo de la institución.
descripcion	Texto	Descripción de la institución.
organismo	Texto	Organismo al que pertenece la institución.

Tabla 43. Tabla institución.

Nombre: tb_ntematica		
Descripción: Almacena los datos de una temática de un proyecto o línea de investigación.		
Atributo	Tipo	Descripción
id_ tematica	Entero	Identificador de la temática.
nombre_ tematica	Cadena	Nombre de la temática.
descripcion	Texto	Descripción de la temática.
activo	Boolean	Si está activa la temática.

Tabla 44. Tabla temática.



Nombre: tb_nclasificacion_proyecto		
Descripción: Almacena los datos de una clasificación de un proyecto de investigación.		
Atributo	Tipo	Descripción
id_clasificacion_proyecto	Entero	Identificador de la clasificación.
nombre_clasificacion_proyecto	Cadena	Nombre de la clasificación.
descripcion	Texto	Descripción de la clasificación.
activo	Boolean	Si está activa la clasificación.

Tabla 45. Tabla clasificación de proyecto.

Nombre: tb_nestado_linea		
Descripción: Almacena los datos de un estado de una línea de investigación.		
Atributo	Tipo	Descripción
id_estado_linea	Entero	Identificador del estado.
nombre_estado_linea	Cadena	Nombre del estado.
descripcion	Texto	Descripción del estado.
activo	Boolean	Si está activo el estado.

Tabla 46. Tabla estado de una línea.



Nombre: tb_nclasificaion_linea		
Descripción: Almacena los datos de una clasificación de una línea de investigación.		
Atributo	Tipo	Descripción
id_clasificaion_linea	Entero	Identificador de la clasificación de una línea.
nombre_clasificaion_linea	Cadena	Nombre de la clasificación de una línea.
descripcion	Texto	Descripción de la clasificación de una línea.
activo	Boolean	Si está activa la clasificación.

Tabla 47. Tabla de clasificación de una línea.

Nombre: tb_nestado_revision		
Descripción: Almacena los datos de un estado de revisión de un proyecto de investigación.		
Atributo	Tipo	Descripción
id_estado_revision	Entero	Identificador del estado de revisión.
nombre_estado_revision	Cadena	Nombre del estado de revisión.
descripcion	Texto	Descripción del estado de revisión.
activo	Boolean	Si está activo el estado.

Tabla 48. Tabla de estado de revisión.



Nombre: tb_nestado_desarrollo		
Descripción: Almacena los datos de un estado de desarrollo de un proyecto de investigación.		
Atributo	Tipo	Descripción
id_estado_desarrollo	Entero	Identificador del estado de desarrollo.
nombre_estado_desarrollo	Cadena	Nombre del estado de desarrollo.
descripcion	Texto	Descripción del estado de desarrollo.
activo	Boolean	Si está activo el estado.

Tabla 49. Tabla de estado de desarrollo.