

Universidad de las Ciencias Informáticas

Facultad 3



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

Título: Diseño e implementación de la base de datos del módulo Laboral para el Sistema de Informatización de Tribunales.

Autor: Nelson Alejandro Rodríguez Pérez.

Tutor: Ing. Darián González Ochoa.

Co-tutor: Ing. Yosvany Gómez Perdomo.

La Habana

Junio de 2012

“El éxito está basado en el buen juicio, el buen juicio está basado en la experiencia y la experiencia se adquiere por el mal juicio” [...]

“¡Enfóquense en el momento, aprendan de sus errores y sigan hacia adelante!”

Brian Farley

Declaración de autoría

Declaro que soy el único autor de este trabajo y autorizo a la Facultad 3 de la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Ing. Darián González Ochoa

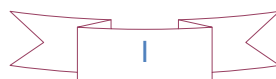
Tutor

Ing. Yosvany Gómez Perdomo

Co-tutor

Nelson Alejandro Rodríguez Pérez

Autor



Datos de contacto

Ing. Darián González Ochoa: Graduado de Ingeniería en Ciencias Informáticas, en la Universidad de las Ciencias Informáticas. Profesor Interno. Jefe del proyecto de Informatización de los Tribunales Populares Cubanos.

Correo electrónico: dochoa@uci.cu

Ing. Yosvany Gómez Perdomo: Graduado de Ingeniería en Ciencias Informáticas, en la Universidad de las Ciencias Informáticas. Profesor Interno. Arquitecto de datos y administrador de bases de datos del proyecto de Informatización de los Tribunales Populares Cubanos.

Correo electrónico: ygperdomo@uci.cu



Agradecimientos

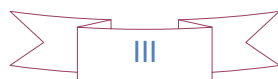
A mami, tía, Eliza, a mi papá, abuela, abuelo, Anita e Ileana por quererme y estar siempre pendientes de mí.

A mis amigos de siempre y a los de la universidad que nunca olvidaré.

A Yosvany y a todos los profesores que me ayudaron con sus sugerencias y recomendaciones.

A los profesores de la carrera como Eutimio y Hugo por marcar el camino para crecer como persona.

A mi novia Linnet por apoyarme y quererme.



Dedicatoria

A mi mamá que es la mejor madre del mundo.

A mi abuelo, abuela, Eliza, mi papá, tía, Anita e Ileana que son mi familia y los quiero mucho.

Resumen

La gestión de los datos en la Materia Laboral de los Tribunales Populares Cubanos se realiza de forma manual; esto trae consigo la intromisión de errores en los documentos, morosidad en el desarrollo de los trámites judiciales y redundancia de la información almacenada. Para resolver estos problemas se desarrolla el módulo Laboral del Sistema de Informatización de Tribunales en la Facultad 3 de la Universidad de las Ciencias Informáticas, este módulo debe gestionar un gran volumen de información y para esto se desarrolló una base de datos.

Para el diseño y la implementación de la base de datos del módulo Laboral se realizó una investigación sobre las bases de datos jurídicas y los sistemas gestores de bases de datos, los modelos de datos, el diseño de bases de datos, la integridad y la seguridad de los datos. Se diseñó el modelo de datos utilizando patrones de diseño de bases de datos. Además se normalizaron las tablas y se utilizaron índices.

Para validar la solución propuesta se realizaron pruebas de volumen, de carga y de estrés que arrojaron resultados satisfactorios.

Palabras claves

Bases de datos, diseño, modelo de datos, módulo Laboral, pruebas.

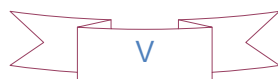


TABLA DE CONTENIDO

INTRODUCCIÓN	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.....	7
1.1 INTRODUCCIÓN	7
1.1.1 Bases de datos jurídicas	7
1.1.2 Sistemas informáticos jurídicos.....	7
1.1.3 Sistemas informáticos jurídicos en Cuba.....	9
1.2 BASES DE DATOS Y SISTEMAS GESTORES DE BASES DE DATOS.....	10
1.2.1 Bases de datos	10
1.2.2 Sistemas gestores de bases de datos	11
1.3 MODELO DE DATOS.....	11
1.3.1 Modelos conceptuales. Modelo entidad–relación	12
1.3.2 Modelos lógicos.....	13
1.4 DISEÑO DE BASES DE DATOS.....	14
1.4.1 Fases del diseño de bases de datos.....	15
1.4.2 Patrones de diseño de bases de datos	17
1.5 NORMALIZACIÓN DE BASES DE DATOS	20
1.6 INTEGRIDAD DE LOS DATOS	24
1.7 SEGURIDAD EN LAS BASES DE DATOS	24
1.8 LENGUAJES DE MODELADO. UML.....	25
1.9 HERRAMIENTAS CASE.....	26
1.9.1 ER/Studio.....	26
1.10 POSTGRESQL	27
1.11 CONCLUSIONES PARCIALES.....	28
CAPÍTULO 2. PROPUESTA DE SOLUCIÓN	30
2.1 INTRODUCCIÓN	30
2.2 ENTORNO DE DESARROLLO	30
2.3 ARQUITECTURA DE DATOS	33
2.4 CONFIGURACIONES	35
2.5 CARACTERIZACIÓN DE LA BASE DE DATOS DEL MÓDULO LABORAL	35
2.6 NOMENCLATURA	35
2.7 DISEÑO DE LA BASE DE DATOS	37
2.7.1 Modelo de datos	37
2.7.2 Patrones de diseño de bases de datos utilizados	38
2.7.3 Descripción de las principales tablas de la base de datos del módulo Laboral	39
2.8 OPTIMIZACIÓN DE LA BASE DE DATOS.....	48
2.8.1 Normalización	48
2.8.2 Índices	49

2.9 INTEGRIDAD Y SEGURIDAD DE LA BASE DE DATOS.....	52
2.10 CONCLUSIONES PARCIALES.....	53
CAPÍTULO 3. VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.....	55
3.1 INTRODUCCIÓN.....	55
3.2 PRUEBAS DE VOLUMEN.....	55
3.3 PRUEBAS DE RENDIMIENTO.....	57
3.4 PRUEBAS DE CARGA Y ESTRÉS.....	61
3.5 CONCLUSIONES PARCIALES.....	62
CONCLUSIONES GENERALES.....	64
RECOMENDACIONES.....	65
REFERENCIAS BIBLIOGRÁFICAS.....	66
BIBLIOGRAFÍA.....	68
ÍNDICE DE TABLAS	
Tabla 1. Herramientas para el entorno de desarrollo.....	30
Tabla 2. Diccionario de datos.....	33
Tabla 3. Tabla del modelo de datos: lab_dpersona.....	40
Tabla 4. Tabla del modelo de datos: lab_dpersonanatural.....	40
Tabla 5. Tabla del modelo de datos: lab_dpersonajurídica.....	41
Tabla 6. Tabla del modelo de datos: lab_ddireccion.....	41
Tabla 7. Tabla del modelo de datos: lab_dtramite.....	42
Tabla 8. Tabla del modelo de datos: lab_dexpediente.....	43
Tabla 9. Tabla del modelo de datos: lab_ddiligencia.....	44
Tabla 10. Tabla del modelo de datos: lab_ddiligenciatribunal.....	44
Tabla 11. Tabla del modelo de datos: lab_demplazamiento.....	45
Tabla 12. Tabla del modelo de datos: lab_descrito.....	46
Tabla 13. Tabla del modelo de datos: lab_dresolucionjudicial.....	47
Tabla 14. Tablas y sus atributos indexados.....	52
Tabla 15. Entidades y sus tuplas en la base de datos.....	58
Tabla 16. Número de hilos para las pruebas de estrés 1 y 2.....	61
Tabla 17. Resultados de las pruebas 1 y 2.....	62
ÍNDICE DE FIGURAS	
Figura 1. Entorno de desarrollo para el modelado de datos.....	31
Figura 2. Distribución de la información.....	33
Figura 3. Modelo de datos del módulo Laboral del SIT.....	37
Figura 4. Tabla lab_nDPA.....	38

Índice

Figura 5. Tablas relacionadas con la tabla lab_dPersona.	39
Figura 6. Tablas lab_dTramite, lab_dExpediente y lab_dExpedienteTramite.	42
Figura 7. Principales tablas relacionadas con la tabla lab_dDiligencia.	43
Figura 8. Tabla lab_dEscrito y sus nomencladores.	45
Figura 9. Tabla lab_dResoluciónJudicial y sus nomencladores.	47
Figura 10. Prueba de volumen para la tabla lab_dtramite.	57
Figura 11. Resultados de la prueba de carga utilizando la herramienta PGAdmin III.....	59
Figura 12. Resultados de la prueba con el atributo “numero” indexado.	60
Figura 13. Resultados de la prueba con el atributo “numero” sin indexar.....	60

Introducción

El desarrollo de las tecnologías de la información y las comunicaciones (TICs) ha permitido una gran transformación tecnológica que hoy posibilita hacer realidad lo que parecía un eslogan: el gobierno electrónico¹ (e-government o también conocido como gobierno en línea). Gradualmente se ha establecido la disponibilidad de las herramientas que han cambiado drásticamente el mundo de las tecnologías de la información implementando un nuevo paradigma, logrado gracias al avance de las comunicaciones y el software junto a la masificación de estándares para el uso y almacenamiento electrónico de documentos.

La incorporación de sistemas informáticos al sistema judicial constituye un contacto directo entre las tecnologías y la ciencia del derecho, un ahorro de esfuerzo y tiempo y trae consigo una mayor calidad en el desarrollo de los procesos judiciales para lograr una justicia rápida; de ahí que “la informática jurídica puede ser definida como la tecnología aplicada a la sistematización y automatización de la información jurídica.” (Pelaez, 2002)

Cuba ha mostrado avances en la producción de software con instituciones como la Universidad de las Ciencias Informáticas (UCI) a la vanguardia y se encuentra constantemente en la búsqueda y aplicación de las TICs en la gestión hacia adentro (control de la entidad) y hacia fuera (servicios de cara al cliente) de los procesos que se llevan a cabo en los departamentos de la administración pública.

La UCI es una universidad de nuevo tipo en el país, fue creada con el objetivo de acelerar el desarrollo de la informática en Cuba, está dedicada a la docencia y a la producción y tiene entre sus prioridades la informatización de disímiles áreas sociales, políticas y económicas del país y dedica además esfuerzos y recursos al software de exportación.

La facultad 3 de la UCI, específicamente el Centro de Gobierno Electrónico, ha venido trabajando en la informatización de algunas entidades de orden jurídico en aras de contribuir al continuo desarrollo del gobierno electrónico en el país. El Sistema de Informatización de Tribunales (SIT) surge en esta universidad como necesidad del Tribunal Supremo Popular de la República de Cuba de cumplir con una

¹ **Gobierno electrónico:** incluye todas aquellas actividades basadas en las tecnologías informáticas, en particular Internet, que el Estado desarrolla para incrementar la eficacia de la gestión pública, mejorar los servicios ofrecidos a los ciudadanos y proveer a las acciones de gobierno un marco transparente.

Introducción

de sus proyecciones estratégicas para el año 2009: continuar desarrollando la infraestructura tecnológica de los tribunales populares y asegurar su gestión, administración y explotación; basándose en los principios de la seguridad informática.

El Tribunal Supremo Popular tiene como misión administrar justicia en nombre del pueblo de Cuba de manera ética, eficaz y humanista para la preservación y desarrollo de nuestra sociedad socialista; su visión es contribuir notablemente a la seguridad jurídica del país y a la realización plena de los derechos al actuar con profesionalidad y sentido de justicia alcanzando alta credibilidad, autoridad y reconocimiento social. Bajo estos principios del Tribunal Supremo, el Sistema de Informatización de Tribunales pretende asistir la toma de decisiones por operadores jurídicos y por las necesidades de organismos superiores, asume además la tarea de incrementar la eficiencia y la eficacia de los procesos que se ejecutan en los Tribunales Populares Cubanos en cualquiera de sus instancias, para la realización de los servicios que prestan en un marco de transparencia y dirigido al sistema de tribunales exclusivamente por la características propias de los procesos judiciales enmarcados en la Ley de la República de Cuba. (Betancourt Vázquez & Montané Izaguirre, 2010)

Los Tribunales Populares Cubanos están estructurados por instancias y materias. Existen tres instancias: tribunal municipal, tribunal provincial y tribunal supremo, cada uno de los cuales cuenta con diferentes procesos distribuidos en cinco materias:

- Materia Administrativa.
- Materia Civil.
- Materia Económica.
- Materia Penal.
- Materia Laboral.

La solución informática planteada para los Tribunales Populares Cubanos está conformada por siete subsistemas o módulos como comúnmente son conocidos, entre estos se encuentra el módulo Laboral cuyo objetivo fundamental es solucionar las reclamaciones o inconformidades sobre la imposición de medidas disciplinarias o por la vulneración de los derechos laborales regulando la participación efectiva de los trabajadores, administraciones de entidades laborales y el Órgano de Justicia Laboral de Base.

Actualmente el trabajo en los tribunales se realiza de forma manual y con gran dependencia del papel,

Introducción

esto presenta los siguientes inconvenientes:

- El registro de documentos, el asiento en los libros, la creación y archivo de los expedientes, posibilitan que se cometan errores de escritura, tachaduras, repetición de palabras, borrones y pérdida de documentos.
- Los expedientes se deterioran con el tiempo en los archivos del tribunal por la humedad, el tamaño de estos y el insuficiente espacio que existe; haciéndose ilegible estos documentos.
- La información registrada es redundante y el tiempo de tramitación de los procesos es bastante prolongado.
- Los reportes estadísticos se emiten en plazos muy largos para la alta dirección, por lo que no se pueden tomar decisiones operativas basadas en información actualizada.

Un aspecto a resaltar es que el procesamiento de la información suele ser muy engorroso y lento, teniendo en cuenta que no existe un sistema que permita que estas operaciones se realicen con la rapidez y la calidad requerida. Otro problema es que no se reutiliza la información existente, fundamentalmente en la creación de los documentos de los expedientes, dígame resoluciones, diligencias, citaciones y notificaciones, pues cada vez que se genera un documento hay que escribir nuevamente información que con anterioridad fue registrada. Además se suma que los datos no se encuentran centralizados lo que conlleva al descontrol y a la desorganización de los mismos y se invierte gran cantidad de tiempo en la búsqueda y consulta de esta información.

Para resolver los problemas relacionados con la gestión de datos e información en la Materia Laboral, se decidió realizar un módulo que gestione sus procesos. El módulo debe ser capaz de manejar gran volumen de información que debe estar estructurada y organizada para facilitar el acceso a la misma, debe ser posible gestionarla a partir de los cambios que se realicen así como permitir el registro de nuevos datos, pero no se cuenta con una estructura que permita realizar estas acciones.

A partir de la problemática descrita anteriormente se identifica el siguiente **problema a resolver**: ¿Cómo proveer al módulo Laboral de una estructura que permita el almacenamiento y la gestión de la información judicial?

Se define como **objeto de estudio**: el diseño y la implementación de bases de datos relacionales.

Introducción

Para dar solución al problema previamente planteado se traza como **objetivo general**: realizar el diseño e implementación del modelo de datos para el almacenamiento y la gestión de la información en el módulo Laboral.

El **campo de acción** del presente trabajo se precisa en el diseño e implementación del modelo de datos del módulo Laboral.

Una vez analizados los elementos anteriores se plantea la siguiente **idea a defender**: con el desarrollo del diseño y la implementación del modelo de datos del módulo Laboral se contribuirá al almacenamiento y gestión de los datos que generan los procesos de la Materia Laboral.

Objetivos específicos:

- Elaborar el marco teórico de la investigación.
- Desarrollar el diseño e implementación del modelo de datos del módulo Laboral de la solución informática Sistema de Informatización de Tribunales.
- Realizar la validación de la propuesta de solución.

Tareas de la investigación:

- Análisis del estado del arte que fundamenta el objeto de la investigación.
- Caracterización de la herramienta para el diseño de bases de datos utilizada en el Sistema de Informatización de Tribunales.
- Caracterización del gestor de base de datos utilizado en el Sistema de Informatización de Tribunales.
- Análisis de los diferentes patrones de diseño de bases de datos existentes.
- Diseño lógico y físico del modelo de datos del módulo Laboral.
- Implementación del diseño físico y configuración de roles y permisos de la base de datos del módulo Laboral.
- Refinamiento de los documentos Entorno de Desarrollo, Configuraciones y la Vista de Datos del documento de Arquitectura.
- Establecimiento de una correlación adecuada entre la normalización y la desnormalización de la base de datos del módulo Laboral.

Introducción

- Realización de pruebas de volumen, rendimiento y estrés a la base de datos del módulo Laboral.

Para realizar la investigación se utilizaron los siguientes **métodos de investigación**:

Métodos teóricos:

- Histórico-lógico: se empleó para analizar la trayectoria y evolución de los sistemas de bases de datos jurídicos.
- Analítico-sintético: permitió el procesamiento de la información y arribar a las conclusiones prácticas y teóricas de la investigación, así como precisar las herramientas utilizadas para el diseño y la implementación de la base de datos.
- Modelación: se utilizó para la creación de abstracciones que explican la realidad, este método es un instrumento de la investigación que permite la creación de modelos, descubrir y estudiar nuevas relaciones y cualidades del objeto de estudio.

Método empírico:

- Entrevista: se realizaron entrevistas a especialistas con experiencia en el diseño y la implementación de bases de datos, con el objetivo de profundizar en las diferentes temáticas a tratar.

Breve descripción de la estructuración del contenido:

Capítulo 1: ubica al lector en las tendencias del uso de bases de datos, abordándose las características y el funcionamiento de las mismas. En este capítulo se desarrollan temas relacionados con los modelos de datos y las características del diseño de bases de datos. Además se analizan herramientas relacionadas con el modelado, la administración y la gestión de bases de datos.

Capítulo 2: en este capítulo se describen las características de la base de datos del módulo Laboral del Sistema de Informatización de Tribunales. Además se describe el estándar de nomenclatura utilizado, se muestran las características generales del diseño de la base de datos, se ejemplifican los patrones de diseño de bases de datos utilizados, se presenta el modelo físico de la base de datos, se utiliza la normalización y se especifica la utilización de índices. También se describen acciones tomadas para asegurar la integridad y la seguridad de la base de datos.

Introducción

Capítulo 3: en este capítulo se realizan pruebas de volumen para analizar el comportamiento de la base de datos al almacenar grandes volúmenes de datos, pruebas de rendimiento para conocer el tiempo de respuesta de la base de datos ante las consultas realizadas y pruebas de estrés donde se realizan consultas simulando varias conexiones de manera concurrente.

Capítulo 1. Fundamentación Teórica

Capítulo 1. Fundamentación teórica

1.1 Introducción

En este capítulo se realiza un estudio sobre las bases de datos jurídicas y los sistemas informáticos jurídicos y se abordan conceptos referentes a las bases de datos, los sistemas gestores de bases de datos, los modelos de datos, el diseño de bases de datos y las herramientas utilizadas.

1.1.1 Bases de datos jurídicas

Las bases de datos jurídicas son documentos jurídicos básicos (legislación, jurisprudencia, interpelaciones parlamentarias, entre otros) almacenados en soportes digitales y susceptibles a ser tratados, recuperados y transmitidos mediante procedimientos y medios informáticos, que con la aplicación de técnicas derivadas de la utilización de la información jurídica pretenden ser utilizados con una finalidad divulgadora pública y generalizada de su contenido. (Alvite Díez, abril,2001)

Estas bases de datos presentan particularidades respecto al resto: emplean distintas unidades documentales, contienen un enorme volumen de documentación almacenada con diferente grado de vigencia, precisan una permanente actualización, su carácter exhaustivo es la garantía de la seguridad jurídica, su aplicabilidad está delimitada a un ámbito jurisdiccional concreto, existe interconexión de la documentación jurídica y cuentan con sistemas de tratamiento de la información adaptados a las necesidades de los profesionales del mundo jurídico. (Alvite Díez, abril,2001)

El desarrollo de las bases de datos jurídicas ha estado marcado por los avances tecnológicos en este campo concreto de la informática. En cualquier caso, la opción tradicional para la construcción de estos productos informáticos se ha venido basando en la abstracción documental de los textos jurídicos, extrayendo de ellos la información sustancial y estructurándola en una serie de campos de interés para su recuperación, con ello se crean registros que dan soporte a los modelos relacionales o documentales de la base de datos elaborada. (Nogales Flores, Martín Galán, & Arellano Pardo, 2003)

1.1.2 Sistemas informáticos jurídicos

El uso del software para el campo jurídico tiene una posición adelantada en varios países pues las

Capítulo 1. Fundamentación Teórica

tecnologías de la información agilizan y modernizan los procesos llevados a cabo por las organizaciones en este campo.

Ejemplos de Sistemas informáticos jurídicos:

Infolex es un software de gestión jurídica utilizado en España. Este software es desarrollado por la empresa Jurisoft, líder en el sector de la informática jurídica, la cual se dedica a la creación de software aplicado al mundo del Derecho, lo que le permite conocer perfectamente las peculiaridades técnicas y legales de la información jurídica. Infolex se enfoca a brindar un servicio integral, para lograrlo cubre todas las necesidades surgidas en la actividad diaria del profesional del derecho: bases de datos documentales, bibliografía, infraestructura informática, servicios de Internet, software de gestión, seguridad informática. Es compatible con el sistema operativo Windows. Utiliza como sistema gestor de bases de datos Microsoft SQL Express 2005 (motor SQL de Microsoft de uso gratuito que se incluye con la instalación) y se recomienda el uso del sistema gestor de bases de datos Microsoft SQL Server 2005 o superior en clientes que posean un gran volumen de datos (más de 10 000 expedientes). (Departamento de Documentación de Jurisoft, marzo, 2009)

El sistema Soft Class para Abogados es otro software utilizado en España para la gestión jurídica y permite llevar un completo control de todas las tareas relacionadas con los abogados. Soft Class para Abogados es un sistema integral de gestión para bufetes de abogados y despachos o asesorías jurídicas que permite llevar un completo control de todas las tareas relacionadas con los mismos, además es un software adaptable a otros ámbitos. La principal función de este software es la administración de expedientes, con varios apartados en el que se gestionan independiente o conjuntamente los datos que lo conforman. El sistema tiene una base de datos en formato MS Access (programa, utilizado en los sistemas operativos Windows de la compañía Microsoft, para la gestión de bases de datos creado y modificado por Microsoft y orientado a ser usado en entorno personal o en pequeñas organizaciones). Eso da garantía de poder recuperar los datos en cualquier momento, y de poder efectuar exportaciones. Del mismo modo, se puede leer los datos de la base de datos para efectuar una combinación o realizar una selección de registros. (Tomás Diago, 2004)

Capítulo 1. Fundamentación Teórica

Además de Infolex y Soft Class para Abogados existen varios sistemas dedicados a la informática jurídica pero la mayoría son propietarios, tienen costos muy elevados y no se ajustan a las necesidades de los Tribunales Populares Cubanos que no cuentan con una infraestructura que los soporte.

1.1.3 Sistemas informáticos jurídicos en Cuba

En el campo de la informática jurídica de gestión el país ha comenzado a dar sus primeros pasos, con el objetivo de automatizar los procesos de las diferentes áreas en cada una de las instancias existentes para lograr una mayor rapidez al tramitar la información.

Cuba no deja escapar la oportunidad de informatizar sus procesos judiciales, en dos ocasiones ha realizado el intento, primero la Materia Penal y luego la Económica. Sin embargo estos sistemas no han logrado satisfacer las expectativas.

Ejemplos de sistemas informáticos jurídicos en Cuba:

SisProp1: es un sistema informático desarrollado en la provincia de Villa Clara. La propuesta inicial fue que abarcara la instancia suprema y provincial de la Materia Penal, desarrollándose solamente la tramitación de los procesos. Esta aplicación al no registrar ningún dato de la tramitación no emite informes estadísticos, no elimina las barreras del papel, quedando por debajo de sus expectativas.

SisEco1: sistema informático desarrollado en La Habana en el 2002, concebido para el área económica. Este sistema no cumplió las expectativas iniciales de su creación. Es lento, cualidad que no va aparejada con la estadística en tiempo real. Prácticamente no informatiza nada, pues la radicación se realiza de forma manual, las salvas se guardan en disquetes, que en la actualidad ya no son utilizados, la información almacenada es borrada cada cinco años, algo incomprensible para un sistema judicial, ya que la información es el mecanismo principal para la justicia.

De forma general los programas Sisprop1 y SisEco1 presentan las siguientes deficiencias:

- No ofrecen reportes estadísticos.
- Nula comunicación entre instancias.
- Arraigados a los registros tradicionales (libros).
- Utilización de software propietario.

Capítulo 1. Fundamentación Teórica

- Permiten hacer cualquier acción sobre el expediente sin tener en cuenta el estado del mismo.

1.2 Bases de datos y sistemas gestores de bases de datos

1.2.1 Bases de datos

Debido a la gran importancia que tiene la información en estos tiempos para la mayoría de las organizaciones de la sociedad, los científicos informáticos han desarrollado un amplio conjunto de conceptos, técnicas, tecnologías, diagramas y herramientas para la gestión de los datos. Las bases de datos y sus tecnologías tienen un impacto decisivo con el creciente uso de los ordenadores y desempeñan un papel crucial en casi todas las áreas de aplicaciones informáticas.

Varios autores han aportado distintos conceptos sobre el término base de datos y en general se plantea la idea de que las bases de datos son un conjunto de datos interrelacionados entre sí, almacenados con carácter más o menos permanente. O sea, que una base de datos puede considerarse una colección de datos variables en el tiempo. (Mato García, 2005)

Una base de datos es un conjunto de datos que tiene las siguientes propiedades implícitas: (Pérez Rojas & Montaña Arango, junio, 2007)

- Representa algún aspecto del mundo real, llamado minimundo o universo de discurso. Las modificaciones del minimundo se reflejan en la base de datos.
- Es un conjunto de datos lógicamente coherentes, con un cierto significado inherente. Una colección aleatoria de datos no puede considerarse propiamente una base de datos.
- Una base de datos se diseña, construye y puebla con datos para un propósito específico. Está dirigida a un grupo de usuarios y tiene ciertas aplicaciones preconcebidas que interesan a distintos usuarios.

Por tanto una base de datos es una colección de información relacionada sobre un aspecto del mundo real. Para su definición y manipulación se utilizan los sistemas gestores de bases de datos cuyo principal objetivo es proporcionar un entorno que sea tanto conveniente como eficiente para las personas que los utilizan para la recuperación y almacenamiento de la información. (Mato García, 2005)

Capítulo 1. Fundamentación Teórica

1.2.2 Sistemas gestores de bases de datos

Los sistemas gestores de bases de datos (SGBD) permiten la utilización y la actualización de los datos almacenados en una o varias bases de datos por uno o varios usuarios desde diferentes puntos de vista. Además permiten almacenar y acceder a los datos de las bases de datos de forma rápida y estructurada. Sirven de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. (Mato García, 2005)

Los SGBD son una capa intermedia entre los programas que el usuario utiliza y el sistema por tanto son los encargados de establecer la comunicación entre ellos, deben presentar los siguientes servicios: (Zambrano Ramírez, 2008)

- Definir y crear bases de datos.
- Manipular los datos utilizando consultas.
- Brindar acceso controlado a los datos mediante mecanismos de seguridad de acceso a los usuarios.
- Mantener la integridad de los datos.
- Controlar la concurrencia a las bases de datos.
- Poseer mecanismos de copias de respaldo y recuperación para restablecer la información en caso de fallos en el sistema.

1.3 Modelo de datos

Una de las características fundamentales de los sistemas de bases de datos es que proporcionan cierto nivel de abstracción de datos, al ocultar las características sobre el almacenamiento físico que la mayoría de usuarios no necesita conocer. Los modelos de datos son el instrumento principal para ofrecer dicha abstracción. (Marqués, enero, 2009)

Un modelo de datos es un conjunto de conceptos que sirven para describir la estructura de una base de datos: los datos, las relaciones entre los datos y las restricciones que deben cumplirse sobre los datos. En general, un modelo no es capaz de expresar todas las propiedades de una realidad determinada, por lo que hay que añadir afirmaciones que complementen el esquema. Los modelos de datos contienen también un conjunto de operaciones básicas para realizar lecturas y actualizaciones de los datos.

Capítulo 1. Fundamentación Teórica

Además, los modelos de datos más modernos incluyen conceptos para especificar comportamiento, permitiendo especificar un conjunto de operaciones definidas por el usuario. (Marqués, enero, 2009)

Los modelos de datos se pueden clasificar dependiendo de los tipos de conceptos que ofrecen para describir la estructura de la base de datos. Los modelos de datos de alto nivel, o modelos conceptuales, disponen de conceptos muy cercanos al modo en que la mayoría de los usuarios percibe los datos, mientras que los modelos de datos de bajo nivel, o modelos físicos, proporcionan conceptos que describen los detalles de cómo se almacenan los datos en el ordenador: el formato de los registros, la estructura de los ficheros (desordenados, ordenados, entre otros) y los métodos de acceso utilizados. Los conceptos de los modelos físicos están dirigidos al personal informático, no a los usuarios finales. Entre estos dos extremos se encuentran los modelos lógicos, cuyos conceptos pueden ser entendidos por los usuarios finales, aunque no están demasiado alejados de la forma en que los datos se organizan físicamente. Los modelos lógicos ocultan algunos detalles de cómo se almacenan los datos, pero pueden implementarse de manera directa en un ordenador. (García Chávez, 2005)

1.3.1 Modelos conceptuales. Modelo entidad–relación

El diseño conceptual parte de las especificaciones de requisitos de los usuarios y su resultado es el esquema conceptual de la base de datos. Una opción para recoger los requisitos consiste en examinar los diagramas de flujo de datos, que se pueden haber producido previamente, para identificar cada una de las áreas funcionales. La otra opción consiste en entrevistar a los usuarios, examinar los procedimientos, los informes y los formularios, y también observar el funcionamiento de la empresa.

Los modelos conceptuales utilizan conceptos como entidades, atributos y relaciones. Una entidad representa un objeto o concepto del mundo real, un atributo representa alguna propiedad de interés de una entidad y una relación describe una interacción entre dos o más entidades. (García Chávez, 2005)

Un esquema conceptual es una descripción de alto nivel de la estructura de la base de datos, independientemente del sistema gestor de bases de datos que se vaya a utilizar para manipularla. Para especificar los esquemas conceptuales se utilizan modelos conceptuales. Los modelos conceptuales se utilizan para representar la realidad a un alto nivel de abstracción. Mediante los modelos conceptuales se puede construir una descripción de la realidad fácil de entender. En el diseño de bases de datos se usan

Capítulo 1. Fundamentación Teórica

primero los modelos conceptuales para lograr una descripción de alto nivel de la realidad, y luego se transforma el esquema conceptual en un esquema lógico.

Modelo entidad-relación.

El modelo entidad-relación (MER) es el modelo conceptual más utilizado para el diseño conceptual de bases de datos. Fue propuesto por Peter Chen en 1976 y ha encontrado una amplia aceptación como instrumento para modelar el mundo real en el proceso de diseño de las bases de datos. El modelo entidad-relación está formado por un conjunto de conceptos que permiten describir la realidad mediante un conjunto de representaciones gráficas y lingüísticas. El MER opera con los conceptos de entidad (estas generalmente se extraen de las especificaciones de requisitos de usuario y en estas especificaciones se buscan los nombres o los sintagmas nominales que se mencionan y representan un objeto o concepto del mundo real) y las relaciones que se establecen entre entidades. (Mato García, 2005)

1.3.2 Modelos lógicos

Los modelos lógicos empleados con mayor frecuencia en los SGBD actuales son el relacional, el de red y el jerárquico. Algunos SGBD más modernos se basan en modelos orientados a objetos.

El modelo relacional se basa en el concepto matemático denominado "relación", que gráficamente se puede representar como una tabla. En el modelo relacional, los datos y las relaciones existentes entre los datos se representan mediante estas relaciones matemáticas, cada una con un nombre que es único y con un conjunto de columnas. (García Chávez, 2005)

En el modelo relacional la base de datos es percibida por el usuario como un conjunto de tablas. Esta percepción es sólo a nivel lógico pues a nivel físico puede estar implementada mediante distintas estructuras de almacenamiento. (García Chávez, 2005)

En el modelo de red los datos se representan como colecciones de registros y las relaciones entre los datos se representan mediante conjuntos, que son punteros en la implementación física. Los registros se organizan como un grafo: los registros son los nodos y los arcos son los conjuntos. (García Chávez, 2005)

El modelo jerárquico es un tipo de modelo de red con algunas restricciones. De nuevo los datos se

Capítulo 1. Fundamentación Teórica

representan como colecciones de registros y las relaciones entre los datos se representan mediante conjuntos. Sin embargo, en el modelo jerárquico cada nodo puede tener un solo padre. Una base de datos jerárquica puede representarse mediante un árbol: los registros son los nodos, también denominados segmentos, y los arcos son los conjuntos. (García Chávez, 2005)

El modelo orientado a objetos define una base de datos en términos de objetos, sus propiedades y sus operaciones. Los objetos con la misma estructura y comportamiento pertenecen a una clase, y las clases se organizan en jerarquías o grafos acíclicos. Las operaciones de cada clase se especifican en términos de procedimientos predefinidos denominados métodos. (García Chávez, 2005)

La mayoría de los SGBD comerciales actuales están basados en el modelo relacional, mientras que los sistemas más antiguos estaban basados en el modelo de red o el modelo jerárquico. Estos dos últimos modelos requieren que el usuario tenga conocimiento de la estructura física de la base de datos a la que se accede, mientras que el modelo relacional proporciona una mayor independencia de los datos. El modelo relacional es declarativo (especifica qué datos se han de obtener) y los modelos de red y jerárquico son navegacionales (especifican cómo se deben obtener los datos). (García Chávez, 2005)

1.4 Diseño de bases de datos

El diseño de bases de datos es el proceso que determina la organización de la base de datos y por ende su estructura, contenido y las aplicaciones que se han de desarrollar. El diseño de una base de datos es un proceso complejo que abarca decisiones a muy distintos niveles. La complejidad se controla mejor si se descompone el problema en subproblemas y se resuelve cada uno de estos subproblemas independientemente, utilizando técnicas específicas. (García Chávez, 2005)

En el diseño de una base de datos se debe realizar un modelo de datos que ayude a entender el significado de los datos y que facilite la comunicación en cuanto a los requisitos de información. La primera etapa es el diseño conceptual, en donde se construye un esquema de la información que se maneja en una organización, independientemente de todas las consideraciones físicas. Después se realiza el diseño lógico, en el que el esquema anterior se transforma según el modelo de base de datos que se vaya a utilizar para implementar el sistema. Por último, en la etapa del diseño físico, se produce una descripción de la implementación de la base de datos. (García Chávez, 2005)

Capítulo 1. Fundamentación Teórica

Los objetivos del diseño de bases de datos son: (García Chávez, 2005)

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

El diseño de bases de datos debe proporcionar un acceso fácil y rápido a los datos evitando duplicar la información innecesariamente, ya que requiere malgastar el espacio e incrementar la posibilidad de que se produzcan errores e incoherencias. Se debe almacenar sólo la información necesaria y que la misma sea correcta y completa es un punto importante garantizando así la exactitud e integridad de ésta. Además el diseño debe soportar las operaciones que los usuarios necesiten realizar con los datos ajustándose a las especificaciones de los mismos. (García Chávez, 2005)

1.4.1 Fases del diseño de bases de datos

A continuación se describen cada una de las etapas del diseño de bases de datos: diseño conceptual, diseño lógico y diseño físico.

Diseño conceptual.

En esta etapa se debe construir un esquema de la información que se usa en la organización, independientemente de cualquier consideración física. A este esquema se le denomina esquema conceptual. Al construir el esquema, los diseñadores descubren la semántica (significado) de los datos de la empresa: encuentran entidades, atributos y relaciones. El objetivo es comprender: (Guerra Cabrera, 2010)

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

El esquema conceptual se puede utilizar para que el diseñador transmita a la organización lo que ha entendido sobre la información que esta maneja. Para ello, ambas partes deben estar familiarizadas con la notación utilizada en el esquema, un ejemplo de notación es la utilizada en el modelo entidad-relación.

Capítulo 1. Fundamentación Teórica

El esquema conceptual se construye utilizando la información que se encuentra en la especificación de los requisitos de usuario. El diseño conceptual es completamente independiente de los aspectos de implementación, como puede ser el SGBD que se vaya a usar, los programas de aplicación, los lenguajes de programación, el hardware disponible o cualquier otra consideración física. Durante todo el proceso de desarrollo del esquema conceptual éste se prueba y se valida con los requisitos de los usuarios. El esquema conceptual es una fuente de información para el diseño lógico de la base de datos. (Guerra Cabrera, 2010)

Diseño lógico.

El diseño lógico es el proceso de construir un esquema de la información que utiliza la empresa, basándose en un modelo de base de datos específico, independiente del sistema gestor de bases de datos que se vaya a utilizar y de cualquier otra consideración física. (Guerra Cabrera, 2010)

En esta etapa, se transforma el esquema conceptual en un esquema lógico que utilizará las estructuras de datos del modelo de base de datos en el que se basa el sistema gestor de bases de datos que se vaya a utilizar, como puede ser el modelo relacional, el modelo de red, el modelo jerárquico o el modelo orientado a objetos. Conforme se va desarrollando el esquema lógico, éste se va probando y validando con los requisitos de usuario. (Guerra Cabrera, 2010)

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

Tanto el diseño conceptual, como el diseño lógico, son procesos iterativos, tienen un punto de inicio y se van refinando continuamente. Ambos se deben ver como un proceso de aprendizaje en el que el diseñador va comprendiendo el funcionamiento de la empresa y el significado de los datos que maneja. El diseño conceptual y el diseño lógico son etapas claves para conseguir un sistema que funcione correctamente. Si el esquema no es una representación fiel de la empresa, será difícil, sino imposible, definir todas las vistas de usuario (esquemas externos), o mantener la integridad de la base de datos. También puede ser difícil definir la implementación física o el mantener unas prestaciones aceptables del sistema. Además, hay que tener en cuenta que la capacidad de ajustarse a futuros cambios es un sello que identifica a los buenos diseños de bases de datos. Por todo esto, es fundamental dedicar el tiempo y

Capítulo 1. Fundamentación Teórica

las energías necesarias para producir el mejor esquema que sea posible. (Guerra Cabrera, 2010)

Diseño físico.

El diseño físico es el proceso de producir la descripción de la implementación de la base de datos en memoria secundaria: estructuras de almacenamiento y métodos de acceso que garanticen un acceso eficiente a los datos.

Para llevar a cabo esta etapa, se debe haber decidido cuál es el sistema gestor de bases de datos a utilizar, ya que el esquema físico se adapta a él. Entre el diseño físico y el diseño lógico hay una retroalimentación, ya que algunas de las decisiones que se tomen durante el diseño físico para mejorar las prestaciones, pueden afectar a la estructura del esquema lógico. (Guerra Cabrera, 2010)

En general, el propósito del diseño físico es describir cómo se va a implementar físicamente el esquema lógico obtenido en la fase anterior. Concretamente, en el modelo relacional, esto consiste en: (Guerra Cabrera, 2010)

- Obtener un conjunto de tablas, relaciones entre ellas y las restricciones que se deben cumplir sobre ellas.
- Determinar las estructuras de almacenamiento y los métodos de acceso que se van a utilizar para conseguir unas prestaciones óptimas.
- Diseñar el modelo de seguridad del sistema.

1.4.2 Patrones de diseño de bases de datos

Un patrón es una plantilla que ya ha sido evaluada como la responsable de resolver un problema, es una guía para apoyarse en realizar el trabajo. Los patrones de diseño de bases de datos permiten crear una base de datos más fortalecida y ofrecen una guía que especifica cómo debe quedar organizada esta. El diseño y construcción de una base de datos requiere del mayor esfuerzo y análisis posible ya que a partir de este se crean las bases de datos que en la actualidad suelen ser muy grandes y utilizar patrones de diseño en su modelado asegura un mejor resultado. (Blaha, 2010)

En términos de base de datos las relaciones que existen entre las tablas pueden ser de uno a uno, de uno a muchos y de muchos a muchos, en este caso las relaciones son de uno a muchos. En los patrones de

Capítulo 1. Fundamentación Teórica

árboles y de grafos a cada nodo se le asocia una entidad.

Patrón: árbol fuertemente codificado. (Hardcoded tree).

El árbol fuertemente codificado es utilizado para representar jerarquías donde es bien conocida la estructura. Este patrón debe utilizarse sólo en los casos en que los cambios en la estructura a representar sean poco probables. El patrón admite tantos niveles como requiera la jerarquía que se vaya a representar. (Blaha, 2010)

Patrón: árbol simple.

Este patrón es 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. (Blaha, 2010)

Patrón: árbol estructurado.

Este modelo es usado cuando se necesita diferenciar los nodos hojas, de aquellos que generan una nueva rama, porque ambos tipos de nodos tienen diferentes atributos, relaciones y/o semántica. No pueden existir ciclos, es decir, un hijo no puede ser su propio padre. La generalización tiene cubrimiento total y exclusivo, cada elemento de la entidad nodo, debe tener su correspondiente elemento en la entidad nodo hoja o en la entidad rama. (Blaha, 2010)

Patrón: grafo dirigido simple.

El conjunto de modelos o patrones de grafos es similar al de los árboles, solo que las relaciones de uno a mucho son sustituidas por relaciones de muchos a muchos, surgiendo así una nueva entidad producto de la relación. El patrón grafo dirigido simple se utiliza cuando todos los nodos contienen el mismo tipo de datos. Es similar al árbol simple, la diferencia es que en este caso la relación recursiva sobre los nodos tiene cardinalidad de muchos a muchos. (Blaha, 2010)

Patrón: grafo dirigido estructurado.

Este patrón se utiliza cuando todos los nodos contienen el mismo tipo de datos. Es similar al árbol

Capítulo 1. Fundamentación Teórica

estructurado, la diferencia es que en este caso la relación sobre los nodos tiene cardinalidad de muchos a muchos. Una aplicación práctica para este patrón sería la representación de las listas de contactos en un servicio web de correo electrónico, donde se permita la creación de listas anidadas, así como incluir a un mismo contacto o una misma lista dentro de varias listas. (Blaha, 2010)

Patrones de flujo de trabajo.

Existen además de los patrones de árboles y los patrones de grafos los patrones de flujo de trabajo que surgen cuando se encuentra la necesidad de representar y persistir flujos de trabajo. Aunque el lenguaje de modelado unificado (UML por sus siglas en inglés) tiene su notación para la representación de un flujo de trabajo, en el modelado bases de datos es recurrente el tema de cómo representar esto en un modelo. Entre estos patrones se encuentran los patrones de máquina de estado para entidades y para escenarios: (Blaha, 2010)

Máquina de estado para un tipo de entidad.

Este patrón se emplea para representar los posibles cambios de estado por los que puede atravesar un tipo de entidad, no refleja el almacenamiento de las ocurrencias sino las propiedades del flujo de trabajo.

Máquina de estado para escenarios (control de flujo).

Este modelo representa la ocurrencia del cambio de estado en un escenario de una entidad dada, por lo tanto considera el tiempo y la persistencia del mismo en las tablas resultantes. También representa la ocurrencia de un estímulo en una fecha y los estados por los que ha pasado.

Patrón modelo entidad-atributo-valor.

El modelo entidad-atributo-valor es la representación de un modelo flexible donde se pueden representar objetos con sus atributos, es un acercamiento al modelo orientado a objetos representado en el modelo relacional, donde se representan las entidades clases, los atributos de las clases, los objetos que representan las instancias de las clases y los valores de cada atributo para cada objeto dado.

Patrón de llaves subrogadas.

Capítulo 1. Fundamentación Teórica

Este patrón es muy utilizado pues se decide generar una llave primaria para cada entidad que no guarda relación con el negocio en vez de usar un atributo identificador en el contexto dado, generalmente se presentan valores enteros de forma incremental que se repiten con una probabilidad extremadamente baja. Permite agilizar las consultas que se realicen por el identificador dado. (Blaha, 2010)

Por su importancia para el diseño de las bases de datos, se utilizan en el diseño de la base de datos del módulo Laboral el patrón de llaves subrogadas y el de árbol simple.

1.5 Normalización de bases de datos

La normalización es la expresión formal del modo de realizar un buen diseño. Provee los medios necesarios para describir la estructura lógica de los datos en un sistema de información. (Mato García, 2005)

La normalización de las tablas en una base de datos tiene entre sus ventajas que evita anomalías en la actualización de los datos, mejora la independencia de los datos disminuyendo la redundancia² en estos y protege la integridad de los datos. Involucra varias fases que se realizan en orden. La realización de la segunda fase supone que se ha concluido la primera y así sucesivamente, las fases de la normalización que se explican a continuación son: primera, segunda y tercera forma normal, la forma normal de Boyce Codd y existen además la cuarta y la quinta forma normal.

Primera forma normal (1FN).

Una relación está en la primera forma normal si, y solo si, satisface la limitación de que contenga solamente valores atómicos. En otras palabras, una relación está en primera forma normal si, y solo si, no incluye ningún grupo repetitivo (un grupo repetitivo es un atributo que contiene un conjunto de valores y no un único valor). (Mato García, 2005)

Segunda forma normal (2FN).

Una relación está en 2FN si, y solo si, está en 1FN y cada atributo no llave es totalmente dependiente de

² **Redundancia:** repetición de la información. La redundancia en las bases de datos hace referencia al almacenamiento de los mismos datos varias veces en diferentes lugares.

Capítulo 1. Fundamentación Teórica

la llave primaria. En otras palabras, una relación R se dice que está en 2FN si está en 1FN y si, y solo si, los atributos no llaves (ni primarias, ni candidatas) de R son funcional y completamente dependientes de la llave primaria de R. (Mato García, 2005)

Las únicas relaciones que pueden violar la 2FN son las que tienen llaves compuestas pues no es posible que en una relación cuya llave primaria sea simple (compuesta por un solo atributo) haya atributos que dependan de parte de la llave primaria. Una relación que esté en 1FN y que tenga una llave primaria simple ya está en 2FN. (Mato García, 2005)

Tercera forma normal (3FN).

Una relación R está en 3FN si, y solo si, está en 2FN y los atributos no llaves son independientes de cualquier otro atributo no llave primaria. En otras palabras, para que una relación esté en 3FN se deben eliminar las dependencias transitivas de atributos no llaves respecto a la llave primaria; estando ya la relación en 2FN. (Mato García, 2005)

Para aclarar el concepto de dependencia transitiva es necesario decir que: sean A, B y C conjuntos de atributos de una relación R. Si B es dependiente funcionalmente de A y C lo es de B, entonces C depende transitivamente de A. (Mato García, 2005)

Forma normal Boyce Codd (FNBC).

Una relación R está en FNBC si y solo si cada determinante³ de la relación es una súper llave de R. Para que una relación no esté en FNBC se deben cumplir la condición necesaria mas no suficiente: (Guerra Cabrera, 2010)

La relación tiene llaves candidatas múltiples, estas llaves candidatas son compuestas y las llaves candidatas se solapan (tienen al menos un atributo en común).

Cuarta forma normal (4FN).

La cuarta forma normal es una forma normal usada en la normalización de bases de datos. La 4NF se asegura de que las dependencias multivaluadas independientes estén correcta y eficientemente

³**Determinante:** atributo o conjunto de atributos del cual depende completamente otro atributo (Mato García, 2005).

Capítulo 1. Fundamentación Teórica

representadas en un diseño de base de datos. La 4NF es el siguiente nivel de normalización después de la forma normal de Boyce Codd. (Pérez, Valero, & Zavala, 2011)

Quinta forma normal (5FN).

La quinta forma normal, también conocida como forma normal de proyección-uniión, es un nivel de normalización de bases de datos designado para reducir redundancia en las bases de datos relacionales que guardan hechos multi-valores aislando semánticamente relaciones múltiples. Una tabla se dice que está en 5NF si y sólo si está en 4NF y cada dependencia de unión (JOIN) en ella es implicada por las claves candidatas. (Pérez, Valero, & Zavala, 2011)

Una base de datos normalizada hasta la tercera forma normal evita en gran medida problemas de actualización de los datos y la redundancia de los mismos y protege además la integridad de los datos. Por estas razones la base de datos del módulo Laboral se normalizará hasta la tercera forma normal y luego se desnormalizarán algunas tablas para lograr un menor tiempo de respuesta en algunas consultas.

Desnormalización.

Para los que se guían estrictamente por las reglas de las formas normales, la desnormalización no es una opción muy "elegante" pero esta puede reducir enormemente el esfuerzo y el tiempo de respuesta en términos de consultas a las bases de datos. (Marqués, enero, 2009)

La desnormalización es una de las tareas que se realiza en el diseño lógico, después de obtener un esquema lógico normalizado considera la introducción de redundancias controladas y otros cambios en el esquema. En ocasiones puede ser conveniente relajar las reglas de normalización introduciendo redundancias de forma controlada con objeto de mejorar las prestaciones del sistema. En la etapa del diseño lógico se recomienda llegar, al menos, hasta la tercera forma normal para obtener un esquema con una estructura consistente y sin redundancias. Pero a menudo sucede que las bases de datos así normalizadas no proporcionan la máxima eficiencia, con lo que es necesario volver atrás y desnormalizar algunas tablas, sacrificando los beneficios de la normalización para mejorar las prestaciones. Es importante hacer notar que la desnormalización sólo debe realizarse cuando se estime que el sistema no puede alcanzar las prestaciones deseadas. Y desde luego, el que en ocasiones sea necesario

Capítulo 1. Fundamentación Teórica

desnormalizar no implica eliminar la fase de normalización del diseño lógico ya que la normalización obliga al diseñador a entender completamente cada uno de los atributos que se han de representar en la base de datos. Además hay que tener en cuenta que la desnormalización hace que la implementación sea más compleja, sacrifica la flexibilidad y puede hacer que los accesos a datos sean más rápidos pero ralentiza las actualizaciones. (Marqués, enero, 2009)

Por regla general, la desnormalización puede ser una opción viable cuando las prestaciones que se obtienen no son las deseadas y las tablas involucradas se actualizan con poca frecuencia, pero se consultan muy a menudo. Las redundancias que se pueden incluir al desnormalizar son de varios tipos: se pueden introducir datos derivados (calculados a partir de otros datos), se pueden duplicar atributos o se puede hacer concatenaciones de unión (JOIN) de tablas. El incluir redundancias dependerá del coste adicional de almacenarlas y mantenerlas consistentes, frente al beneficio que se consigue al realizar consultas. (Marqués, enero, 2009)

No se puede establecer una serie de reglas que determinen cuándo desnormalizar tablas, pero hay algunas situaciones bastante comunes en donde puede considerarse esta posibilidad, por ejemplo: (Marqués, enero, 2009)

- Combinar relaciones de uno a uno: esto puede ser conveniente cuando hay tablas involucradas en relaciones de uno a uno, se accede a ellas de manera conjunta con frecuencia y casi no se accede a ellas por separado.
- Duplicar atributos no clave en relaciones de uno a muchos: para evitar operaciones de unión (JOIN) entre tablas, se puede incluir atributos de la tabla madre en la tabla hija de las relaciones de uno a muchos.
- Duplicar atributos en relaciones de muchos a muchos: durante el diseño lógico se crea una nueva tabla para almacenar las ocurrencias de una relación de muchos a muchos, de modo que si se quiere obtener la información de la relación de muchos a muchos, se tiene que realizar el JOIN de tres tablas. Para evitar algunos de estos JOIN se puede incluir algunos de los atributos de las tablas originales en la tabla intermedia.
- Partir tablas: las tablas se pueden partir horizontalmente (por datos) o verticalmente (por atributos) de modo que a partir de una tabla grande, que tiene datos que no se acceden con frecuencia, se obtengan tablas más pequeñas, algunas de las cuales contienen solamente datos a los que sí se

accede muy a menudo.

1.6 Integridad de los datos

La integridad de los datos consiste en garantizar la no contradicción entre los datos almacenados de modo que en cualquier momento los datos almacenados sean correctos, es decir, que no se detecte inconsistencia entre los datos. Está relacionada con la minimización de la redundancia, ya que es más fácil garantizar la integridad si se elimina la redundancia. (Mato García, 2005)

Los sistemas gestores de bases de datos ofrecen mecanismos para implantar restricciones de integridad en las bases de datos, estas restricciones protegen las bases de datos contra daños accidentales. Los valores de los datos que se almacenan deben satisfacer ciertos tipos de restricciones de consistencia y reglas de integridad y el sistema gestor de bases de datos puede determinar si se produce una violación de la restricción. El SGBD debe además asegurar el acceso concurrente y ofrecer mecanismos para conservar la consistencia de los datos en el caso de que varios usuarios actualicen la base de datos de forma concurrente.

1.7 Seguridad en las bases de datos

El contenido de la base de datos de cualquier organización judicial requiere un extremo cuidado, por lo que se debe proteger el valor de los datos, garantizar la privacidad y controlar el acceso. Los objetivos a considerar al diseñar una aplicación de bases de datos segura son: (Marqués, enero, 2009)

- Privacidad: la información no debe estar disponible para los usuarios no autorizados.
- Integridad: sólo los usuarios autorizados pueden modificar los datos.
- Disponibilidad: los usuarios autorizados no deben ver denegados sus accesos.

Para conseguir estos objetivos se debe establecer una política de seguridad (qué datos proteger y cuáles usuarios pueden acceder a estos) y utilizar los mecanismos de seguridad del sistema gestor de bases de datos para poder seguir esta política. Es de vital importancia para la seguridad de la base de datos impedir que personas no autorizadas accedan al sistema, bien para obtener información o bien para modificarla. Para restringir el acceso al sistema se realiza un control de accesos creando cuentas con claves para los usuarios. El administrador de la base de datos es quien se encarga de conceder privilegios a los usuarios,

Capítulo 1. Fundamentación Teórica

clasificando datos y usuarios según indique la política de seguridad a seguir. (Marqués, enero, 2009)

Además las técnicas de copia de seguridad y restauración compatibles con los sistemas de bases de datos deben garantizar la preservación de los datos ante cualquier imprevisto o falla del sistema, incluso en estos casos el administrador de base de datos debe definir procedimientos para recuperar la información perdida. (Marqués, enero, 2009)

Los SGBD deben proporcionar los siguientes mecanismos para garantizar la seguridad de los datos:

- Garantizar la protección de los datos contra accesos no autorizados, tanto intencionados como accidentales.
- Controlar que sólo los usuarios autorizados accedan a la base de datos.
- Proporcionar herramientas y mecanismos para la planificación y realización de copias de seguridad y restauración.
- Debe ser capaz de recuperar la base de datos, llevándola a un estado consistente en caso de ocurrir algún suceso que la dañe.

1.8 Lenguajes de modelado. UML

Los lenguajes de modelado son un conjunto estandarizado de símbolos y las distintas combinaciones de la disposición para modelar un diseño de software. Los lenguajes de modelado facilitan un buen diseño de los sistemas de software tanto a la hora de desarrollar como al hacer la integración y mantenimiento de sistemas de software.

UML.

El lenguaje de modelado unificado (por sus siglas en inglés Unified Modeling Language), permite visualizar, especificar, construir y documentar los artefactos de un sistema que involucra gran cantidad de software. Proporciona una forma estándar de representar los planos de un sistema, y comprende tanto elementos conceptuales, como los procesos de negocio y las funciones del sistema. Permite una comunicación sencilla y rápida entre desarrolladores y clientes del software que se desarrolla y simplifica el proceso complejo de análisis y diseño de software. (Pressman, 2005)

Capítulo 1. Fundamentación Teórica

Los diagramas UML permiten examinar un sistema desde distintos puntos de vista. Es necesario contar con diferentes perspectivas de las personas implicadas en el desarrollo de un sistema, las cuales tienen enfoques particulares de diferentes aspectos y UML tiene el objetivo de satisfacer a cada persona implicada en el sistema.

1.9 Herramientas CASE

Las herramientas CASE⁴ tienen diversas aplicaciones informáticas destinadas a aumentar los aspectos claves de todo el proceso de desarrollo de un software, desde el principio hasta el final.

“CASE proporciona al ingeniero la posibilidad de automatizar actividades manuales y de mejorar su visión general de la ingeniería [...] Las herramientas CASE ayudan a garantizar que la calidad se diseñe antes de llegar a construir el producto” (Pressman, 2005).

Las herramientas que se presentan a continuación (ER/Studio y PostgreSQL) ya estaban definidas por el equipo de arquitectura del proyecto para el trabajo en el Sistema de Informatización de Tribunales, se analizan y describen para lograr una mayor profundidad en su conocimiento.

1.9.1 ER/Studio

ER/Studio es una poderosa herramienta CASE utilizada para efectuar el modelado lógico y físico de bases de datos relacionales. Esta aplicación permite hacer una separación entre los modelos lógicos y físicos, pero manteniendo una total integración entre ellos, así como con la base de datos. Permite efectuar ingenierías de reversa de esas bases de datos para documentar las tablas, vistas, índices, objetos físicos y definiciones de seguridad. Algunas de sus funcionalidades son: (Embarcadero Technologies, 2009)

- Posibilita transformar el modelo lógicos en un modelo físico.
- Precisa la ingeniería inversa de bases de datos.
- Ofrece instalaciones basadas en HTML de documentación y presentación de informes.
- Posee un repositorio de datos de modelado para el trabajo cooperativo en equipo.

Además ER/Studio permite crear y modificar el modelo de datos y facilita una progresión natural de diseño que permite centrarse en las cuestiones importantes que se desarrollan durante el proceso de diseño.

⁴ **CASE:** Computer Aided Software Engineering, en español: Ingeniería de Software Asistida por Computadora.

Capítulo 1. Fundamentación Teórica

Durante la fase de diseño lógico, permite centrarse en la obtención de un modelo de datos bien documentado y normalizado. Esta herramienta también permite comunicar y validar los elementos del diseño, ayuda la ejecución de la integridad referencial y permite la aplicación de las claves lógicas a través de índices o restricciones de tabla. Una vez completado el diseño lógico y transformado en un modelo físico, proporciona un fuerte apoyo para formular y aplicar la desnormalización. (Embarcadero Technologies, 2009)

Repositorio de ER/Studio.

El repositorio de ER/Studio distribuye el trabajo a los diseñadores de bases de datos de una manera segura y controlada que facilita un ambiente de modelado en equipo. La solución da las utilidades y las características de ejecución, que permiten modelar concurrentemente el modelo y los objetos del modelo. El ambiente de trabajo que brinda este repositorio presenta las siguientes ventajas: (Embarcadero Technologies, 2009)

- Facilita el acceso en tiempo real y de manera concurrente a los diagramas entre los usuarios de ER/Studio.
- Proporciona el control comprensivo del historial, de la versión del diagrama y los componentes del diagrama.
- Ofrece seguridad en los documentos y los perfiles de usuario.

Por su enorme potencial para el modelado de los datos se selecciona ER/Studio en su versión 7.5.1 que cuenta con un repositorio, factor de gran importancia para el trabajo en equipo.

1.10 PostgreSQL

PostgreSQL es uno de los gestores de bases de datos de código abierto más avanzado hoy en día. Diseñado para administrar grandes volúmenes de datos ofrece control de concurrencia multi-versión, soporta la sintaxis SQL⁵ (incluyendo subconsultas, transacciones y funciones definidas por el usuario); también cuenta con un amplio conjunto de enlaces con lenguajes de programación (incluyendo C, C++, Java, PHP) (PostgreSQL Global Development Group, 2009)

⁵ **SQL:** Lenguaje de Consulta Estructurado (en inglés Structured Query Language).

Capítulo 1. Fundamentación Teórica

PostgreSQL presenta las siguientes características: (PostgreSQL Global Development Group, 2009)

- Permite la declaración de funciones propias, así como la definición de disparadores.
- Soporta el uso de índices, reglas y vistas.
- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.
- Soporta funciones y procedimientos en el servidor.
- Está disponible para los Sistemas Operativos Linux y Windows.
- Soporta totalmente el modelo relacional de bases de datos.
- Posee extensiones propias de SQL para realizar consultas sobre la base de datos.

El sistema gestor de bases de datos a utilizado en el Sistema de Informatización de Tribunales es PostgreSQL versión 8.4 que permite gestionar grandes volúmenes de datos y fortalecer la seguridad de la base de datos, aspectos claves a tener en cuenta en el manejo de una base de datos jurídica, el mismo contará con una cuenta de súper usuario que solo será del conocimiento del administrador de la base de datos.

1.11 Conclusiones parciales

Las bases de datos jurídicas tienen un gran impacto en el marco legal, entre sus principales características se encuentran que almacenan un enorme volumen de información, precisan de constante actualización y debido al carácter confidencial de su información se hace imprescindible que posean un alto nivel de seguridad.

Existen varios sistemas dedicados a la informática jurídica pero la mayoría son propietarios, tienen costos muy elevados y no se adaptan a las necesidades de los Tribunales Populares Cubanos que no cuentan con una infraestructura que los soporte.

Los modelos de datos determinan la organización, estructura y rendimiento de las bases de datos. El modelo lógico seleccionado es el modelo relacional pues proporciona una manera simple de representar los datos a través de un conjunto de tablas y este modelo puede ser entendido fácilmente por todo el equipo de desarrollo.

Los patrones permiten solucionar problemas que se dan con frecuencia en el diseño de bases de datos y

Capítulo 1. Fundamentación Teórica

su uso permite mejorar el diseño de las mismas.

La normalización de las tablas en una base de datos mejora la independencia y reduce la redundancia de los datos y en ocasiones es necesario desnormalizar algunas para reducir el esfuerzo y el tiempo de respuesta en términos de consultas a las bases de datos.

La herramienta de modelado ER/Studio ofrece la posibilidad de transformar el diseño lógico en un diseño físico y cuenta con un repositorio, muy útil para el trabajo en equipo.

El sistema gestor de bases de datos a utilizar será PostgreSQL soporta la sintaxis SQL, permite la gestión de varios usuarios y los permisos asignados a ellos y se puede enlazar con el lenguaje de programación PHP utilizado en el desarrollo del Sistema de Informatización de Tribunales.

Capítulo 2. Propuesta de solución

Capítulo 2. Propuesta de solución

2.1 Introducción

En este capítulo se realiza la descripción de la propuesta de solución, se describen los procesos principales que tienen lugar en la construcción de la base de datos, se hace referencia a la vista de datos del documento de arquitectura del proyecto para los Tribunales Populares Cubanos y se tratan las configuraciones relacionadas con la base de datos. Además se presenta el modelo de datos, se describen las principales tablas de este modelo y la optimización de la base de datos empleando la normalización de las tablas e índices. También se abordan los temas de seguridad e integridad y cómo se aplican estos en la base de datos.

2.2 Entorno de desarrollo

Las herramientas para desarrollar la base de datos del Sistema de Informatización de Tribunales se presentan en la tabla 1.

Herramienta	Propósito
ER/Studio 7.5.1	Modelo lógico de datos.
Embarcadero Repository 4.5.1	Repositorio para el modelo lógico de datos.
PgAdmin III	Cliente de bases de datos.
PostgreSQL 8.4	Sistema gestor de bases de datos.

Tabla 1. Herramientas para el entorno de desarrollo.

Configuración del entorno de desarrollo para el modelado de datos:

- En el servidor está instalado el Embarcadero Repository 4.5.1, el cual es el repositorio para el trabajo en equipo de manera concurrente con la herramienta ER/Studio 7.5.
- En las estaciones de trabajo se utiliza la herramienta de modelado ER/Studio 7.5.
- PostgreSQL 8.4 está instalado en el servidor para gestionar la base de datos.
- En las estaciones de trabajo se utiliza PgAdmin III como cliente de PostgreSQL para administrar la

Capítulo 2. Propuesta de solución

base de datos.

- Se configurará una salva automática del modelo y de la base de datos del repositorio, programada para realizarse diariamente a las 4 AM.

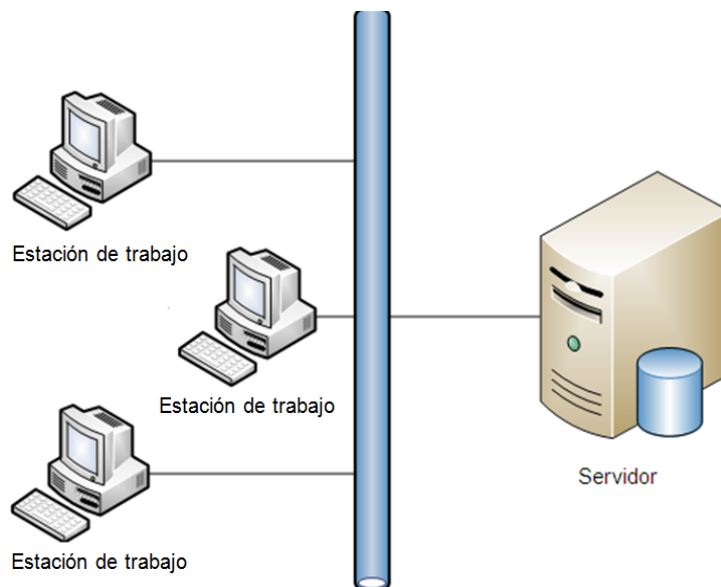


Figura 1. Entorno de desarrollo para el modelado de datos.

Usuarios y privilegios.

El repositorio tiene un usuario para la administración del mismo. Fueron creados además usuarios con permisos suficientes y necesarios para trabajar en el modelo de datos; como medida de seguridad estos usuarios no tienen acceso a elementos de administración del repositorio, así como a modificar el modelo de datos de otro módulo (uno de estos usuarios cuenta con los permisos necesarios y suficientes para trabajar en el modelo de datos del módulo Laboral) y los usuarios designados a trabajar en un modelo de datos son los responsables del mismo.

Para el trabajo con la base de datos del Sistema de Informatización de Tribunales en PostgreSQL se crearon diferentes roles. El rol administrador tiene permisos para administrar la base de datos en su totalidad y existen varios roles por cada módulo los cuales están estrechamente relacionados con la estructura de trabajo de los Tribunales Populares Cubanos, del cual no queda exento el módulo Laboral, con permisos para seleccionar, insertar, actualizar los datos de las tablas y realizar funciones en

Capítulo 2. Propuesta de solución

dependencia de los permisos que tenga cada rol.

Para el trabajo con PostgreSQL el rol administrador tiene permisos para administrar toda la base de datos y existen roles por módulo para los programadores, incluyendo el módulo Laboral, con permisos para seleccionar, insertar y actualizar los datos de las tablas.

Herramienta de modelado y vinculación con el repositorio.

La herramienta de modelado se conectará al repositorio para controlar los modelos de datos realizados. El repositorio estará organizado por módulos y cada módulo tendrá su respectivo modelo de datos. Tanto el módulo como el modelo de datos se nombrarán como el módulo al cual pertenecen.

Tipos de datos.

Se utilizará un diccionario de datos para definir los tipos de datos a utilizar como dominio, los cuales son expuestos en la tabla 2, de forma tal que se utilicen los mismos tipos de datos. Si en el futuro se realiza algún cambio en cuanto a un tipo de dato, solo se cambiaría este tipo de dato en el diccionario de datos y automáticamente todos los campos con ese dominio se actualizarán. (Osorio Rodríguez & Gómez Perdomo, febrero, 2012)

Dominio	Tipo de Dato	Campo
Binario	BINARY	Documentos, imágenes, etc.
Bool	BIT	Verdadero o Falso.
Contraseña	TEXT	Contraseña de usuario.
EMail	TEXT	Correo electrónico.
Entero	INTEGER	Números enteros positivos.
FechaHora	DATETIME	Fecha y hora.
Nombre	VARCHAR(30)	Nombre hasta 30 caracteres.
Numero	NUMERIC	Valores numéricos y decimales.

Capítulo 2. Propuesta de solución

NumeroCI	VARCHAR	Número de carné de identidad.
Texto	TEXT	Texto en general.

Tabla 2. Diccionario de datos.

2.3 Arquitectura de datos

La tesina del ingeniero Alain Osorio Rodríguez de noviembre del año 2010 relacionada con la base de datos jurídica del SIT indica que: “cada tribunal contará con una base de datos independiente para gestionar toda la información competente en el mismo. También existirá un centro de datos con toda la información de todos los tribunales, el cual estará sincronizado con cada uno de los tribunales de forma tal que en cada momento se pueda consultar información, con el acceso autorizado, de cualquier tribunal. Los tribunales no se comunicarán directamente entre sí sino a través del centro de datos. De esta forma se garantiza la independencia de la tramitación judicial en cada uno de los tribunales y un control centralizado de toda la información que se maneja en estos”.

El Sistema de Informatización de Tribunales cuenta con una estructura de servidores de bases de datos, ubicándose un servidor en cada tribunal independientemente de la instancia en la que se encuentre el mismo.

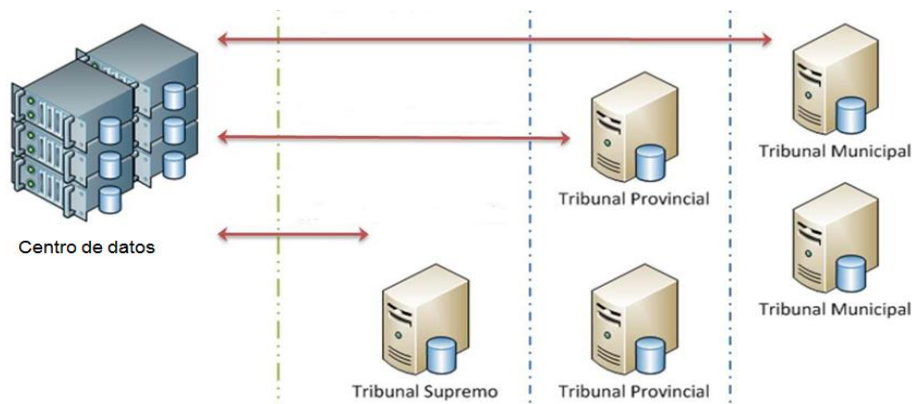


Figura 2. Distribución de la información.

Capítulo 2. Propuesta de solución

La arquitectura anteriormente explicada, es la abstracción del funcionamiento de un sistema de base de datos distribuida, que no es más que una colección de datos distribuidos en diferentes instancias a través de una red de computadoras.

Los sistemas de bases de datos distribuidas aumentan la disponibilidad de los datos, reducen el tráfico de comunicación y disminuyen enormemente los costos en el equipamiento y la infraestructura de comunicaciones de las redes de computadoras. El diseño de las bases de datos distribuidas posee la fase de diseño centralizado y cuenta, además, con dos nuevos problemas que caracterizan el proceso de distribución de datos: cómo obtener los datos que se encuentran en diferentes instancias y cómo y qué cantidad de datos debe ser replicados. (Marcotte, 2005)

Réplica.

Para lograr la sincronización⁶ de los datos de cada una de las instancias del sistema distribuido se utilizará la réplica de datos (solamente se replicarán las tablas de la base de datos que no sean nomencladoras), que no es más que la operación de transportar idénticamente la información de las tablas deseadas de una instancia a otra a través de la red, posibilitando la corrección, disponibilidad, coordinación y efectividad de los datos en un momento y lugar determinado. (Marcotte, 2005)

Básicamente existen dos tipos o entornos de réplicas: el de solo lectura o maestro-esclavo (master-slave en inglés) que permite al nodo maestro realizar consultas de lectura/escritura, mientras que los nodos esclavos solo de lectura; y el otro entorno es el multimaestro donde muchos nodos maestros interactúan entre sí facilitando la sincronización de los cambios. El Sistema de Informatización de Tribunales contará con un entorno de réplica maestro-esclavo.

La arquitectura definida y el establecimiento de sistemas distribuidos proporcionan un valor agregado al resguardar la información pues si el centro de datos perdiera sus datos por algún motivo, a través de la réplica de los datos de cada uno de los nodos restauraría la información contenida y si esto le ocurriera a alguno de los tribunales en alguna instancia, este obtendría la información del centro de datos.

⁶ **Sincronización:** Realización de forma simultánea de varios procesos.

Capítulo 2. Propuesta de solución

2.4 Configuraciones

Las configuraciones e instalaciones de los servidores de bases de datos son parte de una estrategia de la arquitectura de datos.

Como sistema operativo se instalará Debian en su versión 5.0.4, el mismo es un sistema operativo libre, de ahí que su utilización esté en correspondencia con las políticas de la universidad con respecto al desarrollo de software para sistemas de código abierto. El sistema gestor de bases de datos utilizado será PostgreSQL versión 8.4, este contiene una cuenta de súper usuario que sólo será del conocimiento del administrador de la base de datos y fueron creados los siguientes esquemas con sus abreviaturas, cada uno de los cuales posee una cuenta de usuario propia: (Gómez Perdomo, 2012)

- Administrativo: adm.
- Civil: civ.
- Económico: ecm.
- Gobierno y Administración: gov.
- Laboral: lab.
- Penal: pnl.

2.5 Caracterización de la base de datos del módulo Laboral

La base de datos del módulo Laboral pertenece a un esquema único compuesto por esquemas de bases de datos pertenecientes a los módulos del SIT; es una base de datos relacional y distribuida pues una vez desplegado el sistema estará distribuido geográficamente en varios servidores de bases de datos y en todas sus instancias: Tribunal Municipal, Tribunal Provincial y Tribunal Supremo. El esquema del módulo Laboral utiliza datos del esquema perteneciente al módulo Administración y Gobierno del Sistema de Informatización de Tribunales, para intervenir en la informatización y gestión de los procesos fundamentales en los que interviene la Materia Laboral. El esquema del módulo Laboral tiene 90 tablas de las cuales 35 son nomencladores.

2.6 Nomenclatura

Para el desarrollo de la base de datos se tuvieron en cuenta aspectos relacionados con la nomenclatura de sus tablas, columnas e índices y se muestran a continuación:

Capítulo 2. Propuesta de solución

Consideraciones generales de nomenclatura: (Osorio Rodríguez & Gómez Perdomo, febrero, 2012)

- Los nombres se limitan a 35 caracteres y se deben evitar números en ellos.
- Se deben utilizar guiones bajos lo menos posible y no se pueden usar espacios.
- Se utiliza una letra como primer caracter del nombre y no se comienza con guiones bajos.
- Se evitan abreviaciones pues puede malinterpretarse el nombre.
- Se evitan acrónimos pues algunos tienen más de un significado.
- No utilizar tildes ni diéresis y se utiliza “nn” en sustitución de la ñ.

Nomenclatura para las tablas:

- Todas las tablas comienzan con el prefijo lab_, por ejemplo la tabla: lab_dTramite. Las tablas nomencladoras comienzan de esta forma: lab_n, por ejemplo: lab_nDiligencia. El resto de las tablas comienzan de esta forma: lab_d. Ejemplo: lab_dEmplazamiento.
- Las especializaciones se nombrarán: nombre de la tabla padre + nombre de la tabla hija. Ejemplo:
 - ✓ Tabla padre: lab_dExpediente.
 - ✓ Tabla hija: lab_dExpedienteUltimoTramite.

Nomenclatura para las columnas:

- No hay necesidad de mencionar el nombre de la tabla en el nombre de la columna. El campo llave primaria es la única excepción a esta regla donde incluir el nombre está justificado para tener un campo más descriptivo. Por ejemplo IdExpediente en vez de Id como nombre del campo llave de la tabla lab_dExpediente.
- La restricción de llave primaria se nombra con el prefijo pk_ seguido del nombre de la tabla.
- Las llaves foráneas deben tener el mismo nombre que aparece en la tabla a la que hace referencia donde son llave primaria, excepto cuando se especifique un rol, y la relación correspondiente entre tablas, debe nombrarse con los nombres de ambas tablas, primero con el nombre de la tabla padre y luego con el nombre de la tabla hija, con el prefijo fk_.
- Las llaves alternativas se nombran con el prefijo ak_ seguido del nombre de la tabla.

Nomenclatura para los índices:

- Los índices se nombran con el prefijo idx_ seguido del nombre de la tabla + el nombre del campo, o seguido del nombre de la tabla + el nombre de los campos según sea la característica del índice.

Capítulo 2. Propuesta de solución

2.7 Diseño de la base de datos

2.7.1 Modelo de datos

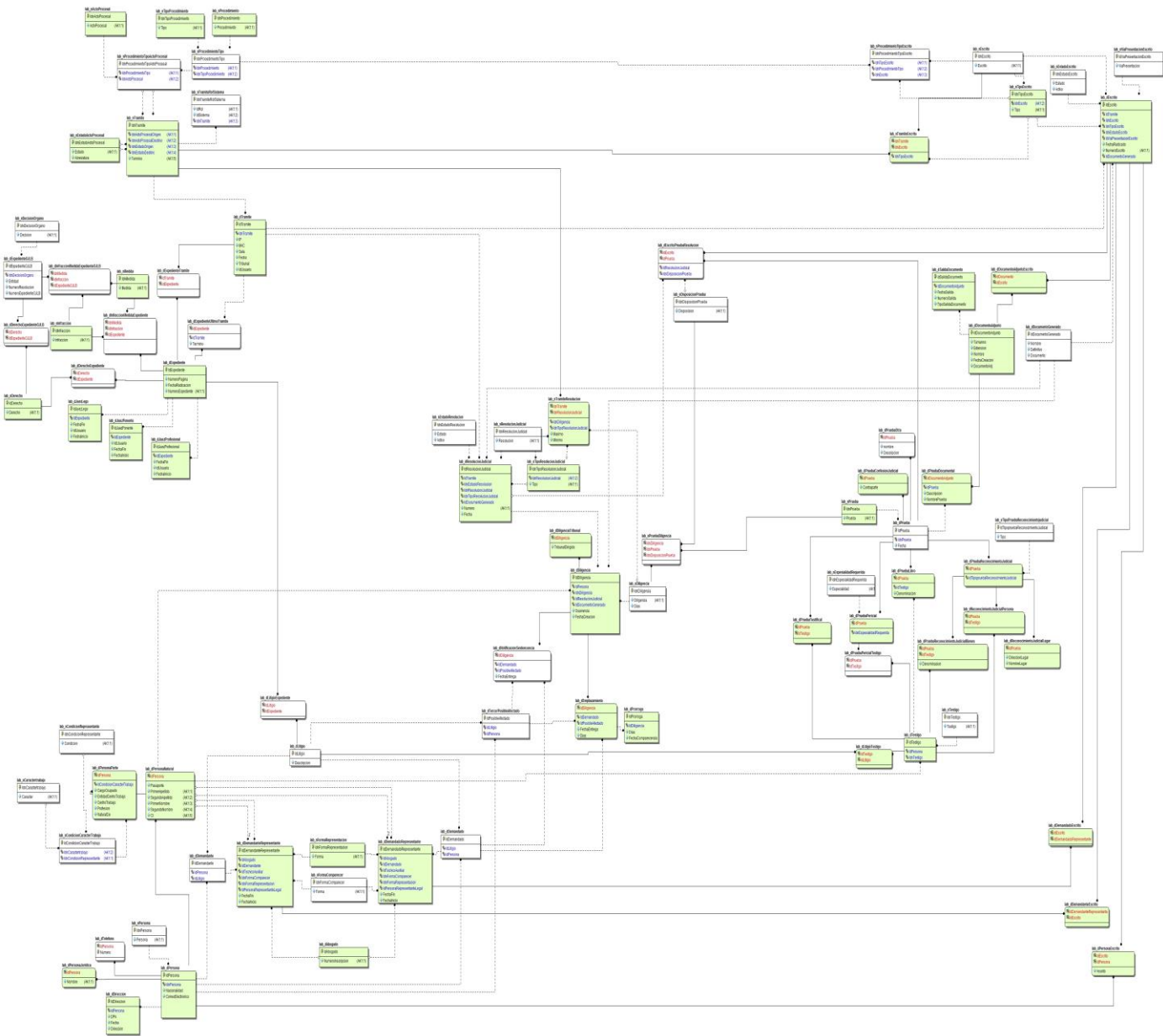


Figura 3. Modelo de datos del módulo Laboral del SIT.

Capítulo 2. Propuesta de solución

2.7.2 Patrones de diseño de bases de datos utilizados

Los patrones de diseño de bases de datos son la base para la búsqueda de soluciones a problemas comunes en el modelado y diseño de las bases de datos. Durante la realización del modelo de datos fueron utilizados varios patrones de diseño dentro de los que se encuentran:

Patrón de llaves subrogadas.

Este es uno de los patrones más utilizados. Utilizándolo se genera una llave primaria, que no tiene relación con el negocio, de tipo entero grande para cada entidad. Se utiliza en la tabla lab_nProcedimientoTipo, formada de la relación de muchos a muchos entre las tablas lab_nProcedimiento y lab_nTipoProcedimiento, donde el atributo idnProcedimientoTipo es una llave subrogada.

Patrón árbol simple.

Es la representación de la estructura de un árbol en programación, con su nodo raíz y el resto de sus nodos que pueden ser hojas. En la tabla nomencladora lab_nDPA que almacena datos correspondientes a la División Político Administrativa (DPA) se utiliza el patrón de árbol simple, el atributo “raiz” (sin tilde para respetar la nomenclatura definida) de tipo booleano toma valor “true” (verdadero) para las DPAs que son países, por ejemplo Cuba, y el atributo hoja toma valor “true” para los municipios de Cuba que tiene provincias y municipios, en el caso de países como Canadá que tiene estados y localidades estas últimas serían las hojas.

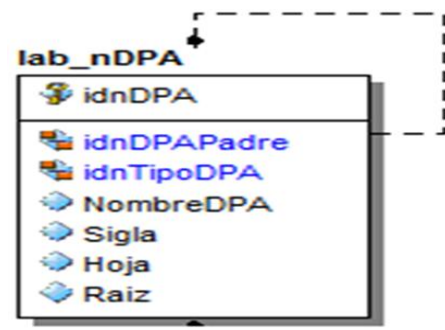


Figura 4. Tabla lab_nDPA.

Capítulo 2. Propuesta de solución

2.7.3 Descripción de las principales tablas de la base de datos del módulo Laboral

Principales tablas relacionadas con la tabla lab_dPersona.

En la siguiente figura se muestran las principales tablas relacionadas con la tabla lab_dPersona, la cual almacena datos de las personas y se relaciona con las tablas: lab_dPersonaNatural que almacena los datos principales de una persona común y lab_dPersonaJurídica que son las entidades en general (UCI, Tribunal Supremo Popular, Banco Metropolitano, entre otras). Las tablas lab_dDireccion y lab_dTelefono contienen las direcciones y los números telefónicos de las personas.

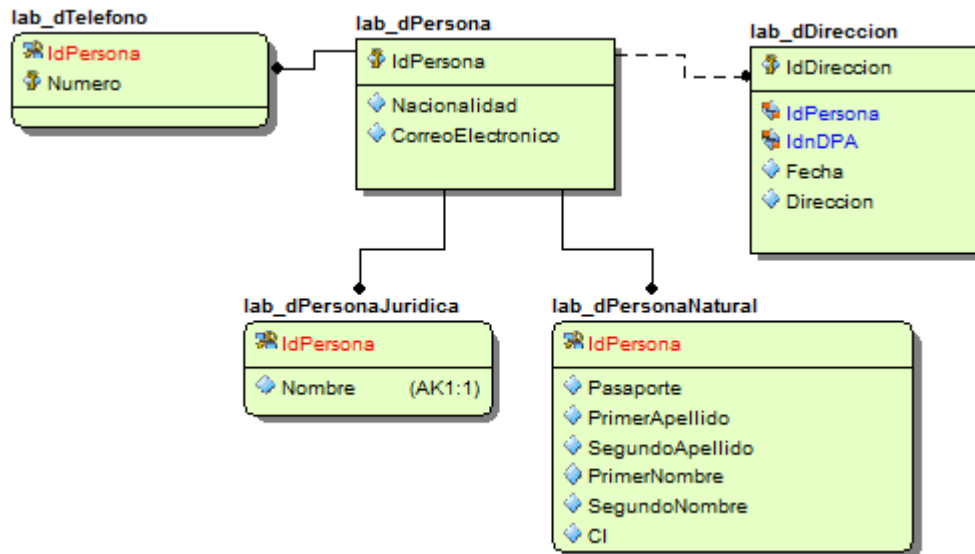


Figura 5. Tablas relacionadas con la tabla lab_dPersona.

A continuación se muestran las descripciones de las principales tablas de esta relación:

Entidad	lab_dpersona	
Descripción	Personas.	
Atributos	Tipo	Descripción de los atributos

Capítulo 2. Propuesta de solución

idpersona	bigint	Identificador de la persona.
nacionalidad	text	Nacionalidad de la persona.
correoelectronico	text	Correo electrónico de la persona.
idnpersona	bigint	Identificador del tipo de persona.

Tabla 3. Tabla del modelo de datos: lab_dpessoa.

Entidad	lab_dpessoa	
Descripción	Datos de las personas naturales.	
Atributos	Tipo	Descripción de los atributos
idpersona	bigint	Identificador de la persona.
primerapellido	text	Primer apellido de la persona.
segundoapellido	text	Segundo apellido de la persona.
primernombre	text	Nombre de la persona.
segundonombre	text	Segundo nombre de la persona.
ci	varchar(11)	Carné de identidad de la persona.
pasaporte	text	Pasaporte de la persona.

Tabla 4. Tabla del modelo de datos: lab_dpessoajuridica.

Entidad	lab_dpessoajuridica	
Descripción	Personas jurídicas.	
Atributos	Tipo	Descripción de los atributos

Capítulo 2. Propuesta de solución

idpersona	bigint	Identificador de la persona.
nombre	text	Nombre de la persona jurídica.

Tabla 5. Tabla del modelo de datos: lab_dpersonajurídica.

Entidad	lab_ddireccion	
Descripción	Dirección de las personas.	
Atributos	Tipo	Descripción de los atributos
iddireccion	bigint	Identificador de la dirección
direccion	text	Dirección conocida.
idndpa	bigint	Identificador de la División Político Administrativa.
idpersona	bigint	Identificador de la persona.
fecha	timestamp	Fecha de la última dirección conocida.

Tabla 6. Tabla del modelo de datos: lab_ddireccion.

Tablas lab_dTramite, lab_dExpediente y lab_dExpedienteTramite.

En la siguiente figura se muestran las tablas lab_dTramite, la cual registra los actos procesales realizados y la tabla lab_dExpediente que representa una carátula de un archivo que contiene los documentos con toda la tramitación judicial en un caso judicial; entre estas dos tablas existe una relación de muchos a muchos por lo que se crea en el modelo físico la tabla lab_dExpedienteTramite.

Capítulo 2. Propuesta de solución

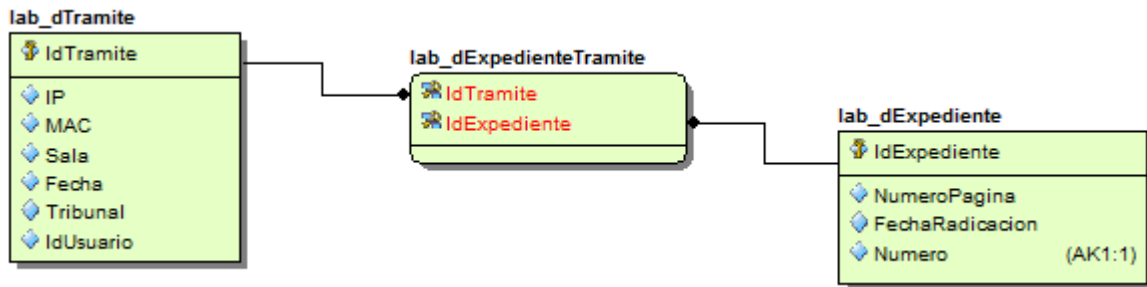


Figura 6. Tablas lab_dTramite, lab_dExpediente y lab_dExpedienteTramite.

A continuación se muestran las descripciones de las principales tablas de esta relación:

Entidad	lab_dtramite	
Descripción	Tabla donde se registran los actos procesales realizados.	
Atributos	Tipo	Descripción de los atributos
idtramite	bigint	Identificador del trámite.
mac	MACADDR	Número MAC.
ip	CIDR	Dirección IP.
fecha	timestamp	Fecha de realización del trámite.
idntramite	bigint	Identificador del tipo de trámite.
idusuario	bigint	Identificador del usuario que realiza el trámite.
tribunal	NUMERIC(19,0)	Tribunal donde se realiza el trámite.
sala	NUMERIC(19,0)	Sala donde se realiza el trámite.

Tabla 7. Tabla del modelo de datos: lab_dtramite.

Entidad	lab_dexpediente
---------	-----------------

Capítulo 2. Propuesta de solución

Descripción	Legajo que contiene toda la tramitación judicial en un caso concreto.	
Atributos	Tipo	Descripción de los atributos
idexpediente	bigint	Identificador del expediente.
numero	text	Número del expediente.
fecharadicacion	timestamp	Fecha en que se crea el expediente.
numeropagina	bigint	Número de páginas que tiene el expediente.

Tabla 8. Tabla del modelo de datos: lab_dexpediente.

Principales tablas relacionadas con la tabla lab_dDiligencia.

En la siguiente figura se muestran las principales tablas relacionadas con la tabla lab_dDiligencia, la cual guarda los datos de las diligencias y se relaciona con las tablas: lab_dDiligenciaTribunal que almacena los datos del tribunal al cual va dirigido la diligencia y lab_dEmplazamiento que almacena citaciones que poseen una cantidad de días para presentarse.

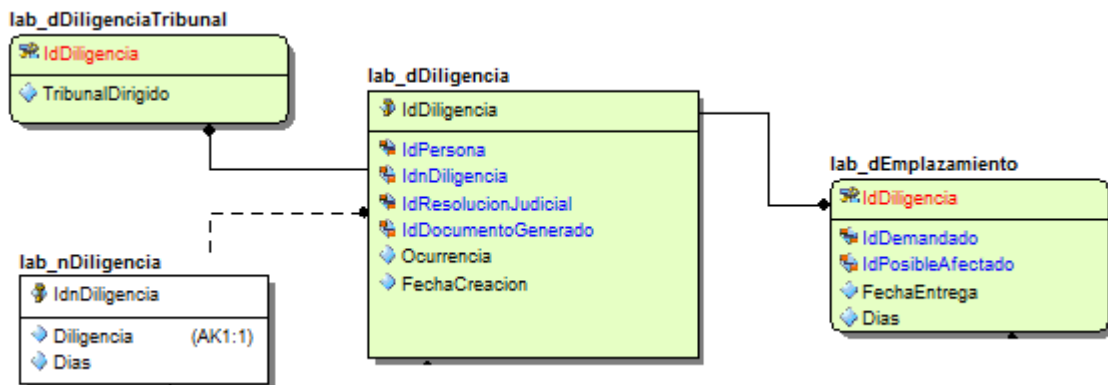


Figura 7. Principales tablas relacionadas con la tabla lab_dDiligencia.

A continuación se muestran las descripciones de las principales tablas de esta relación:

Capítulo 2. Propuesta de solución

Entidad	lab_ddiligencia	
Descripción	Diligencias realizadas.	
Atributos	Tipo	Descripción de los atributos
iddiligencia	bigint	Identificador de la diligencia.
fechacreacion	timestamp	Fecha en que fue creada la diligencia.
ocurrencia	text	Ocurrencia en caso de existir algún problema con la entrega de la diligencia.
idresolucionjudicial	bigint	Identificador de la resolución judicial.
iddiligencia	bigint	Identificador del tipo de diligencia.
idpersona	bigint	Identificador de la persona.
iddocumentogenerado	bigint	Identificador del documento digital.

Tabla 9. Tabla del modelo de datos: lab_ddiligencia.

Entidad	lab_ddiligenciatribunal	
Descripción	Tribunal al cual va dirigido la diligencia.	
Atributos	Tipo	Descripción de los atributos
iddiligencia	bigint	Identificador de la diligencia.
tribunaldirigido	numeric	Identificador del tribunal al cual va dirigida la diligencia.

Tabla 10. Tabla del modelo de datos: lab_ddiligenciatribunal.

Entidad	lab_demplazamiento
---------	--------------------

Capítulo 2. Propuesta de solución

Descripción	Citaciones con un término de días.	
Atributos	Tipo	Descripción de los atributos
iddiligencia	bigint	Identificador de la diligencia
iddemandado	bigint	Identificador del demandado
fechaentrega	timestamp	Fecha en que se entregó la cédula de emplazamiento
idposibleafectado	bigint	Identificador del tercero posible afectado
días	bigint	Días hábiles a presentarse.

Tabla 11. Tabla del modelo de datos: lab_demplazamiento.

Tabla lab_dEscrito y sus nomencladores.

La tabla lab_dEscrito es importante para el módulo Laboral pues los escritos son documentos judiciales que se almacenan.

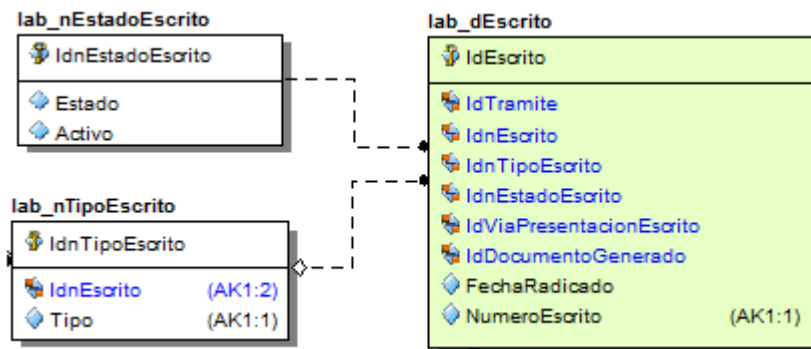


Figura 8. Tabla lab_dEscrito y sus nomencladores.

Descripción de la tabla lab_descrito:

Entidad	lab_descrito
---------	--------------

Capítulo 2. Propuesta de solución

Descripción	Cualquier documento sin formalidad ni requisitos.	
Atributos	Tipo	Descripción de los atributos
idescrito	bigint	Identificador del escrito.
numeroescrito	text	Número único que identifica al escrito.
fecharadicado	timestamp	Fecha de ingreso del escrito.
idviapresentacionescrito	text	Identificador de la vía de presentación.
idtramite	bigint	Identificador del trámite.
idnescrito	bigint	Identificador del tipo de escrito.
idntipoescrito	bigint	Identificador del subtipo de escrito.
idnestadoescrito	bigint	Identificador del estado del escrito.
iddocumentogenerado	bigint	Identificador del documento digital.

Tabla 12. Tabla del modelo de datos: lab_descrito.

Tabla lab_dResolucionJudicial y sus nomencladores.

La tabla lab_dResolucionJudicial contiene las resoluciones judiciales, las cuales son decisiones adoptadas por los jueces en los tribunales en el curso de una causa.

Capítulo 2. Propuesta de solución

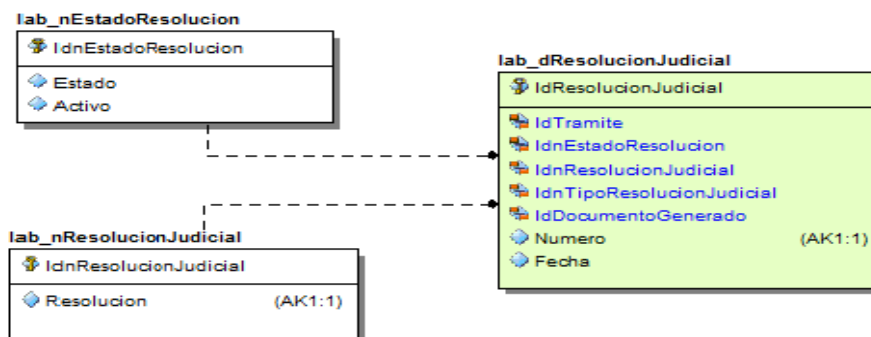


Figura 9. Tabla lab_dResoluciónJudicial y sus nomencladores.

Descripción de la tabla lab_dresolucionjudicial:

Entidad	lab_dresolucionjudicial	
Descripción	Resoluciones judiciales.	
Atributos	Tipo	Descripción de los atributos
idresolucionjudicial	bigint	Identificador de la resolución judicial.
numerosolucion	text	Número único que identifica a la resolución.
fecha	timestamp	Fecha de inicio de la resolución judicial.
idtramite	bigint	Identificador del trámite.
idnresolucionjudicial	bigint	Identificador del tipo de resolución.
idntiporesolucionjudicial	bigint	Identificador del subtipo de resolución.
idnestadoresolucion	bigint	Identificador del estado de la resolución.
iddocumentogenerado	bigint	Identificador del documento digital.

Tabla 13. Tabla del modelo de datos: lab_dresolucionjudicial.

Es necesario acotar que en algunos nombres y atributos de las tablas de este acápite se utiliza la notación

Capítulo 2. Propuesta de solución

utilizada en el ER/Studio para nombrarlas, donde la primera letra de las palabras es mayúscula.

Las descripciones de las tablas y atributos expuestas anteriormente fueron extraídas del documento “Módulo de Bases de Datos (Laboral). Versión 1.1”, donde se puede consultar la descripción de todas las tablas y los atributos de la base de datos del módulo Laboral.

2.8 Optimización de la base de datos

Es posible lograr la optimización⁷ ya sea por el tipo específico de gestor con que se manejen los datos y sus configuraciones puntuales, por el modelo de datos seleccionado, por configuraciones que se realicen sobre la base de datos, como por las optimizaciones en las consultas. Las demoras del sistema ante operaciones que involucren un gran volumen de datos pueden ser reducidas. Las técnicas de optimización pueden enfocarse entonces tanto en el nivel físico, por ejemplo, distribuyendo la información en distintos ficheros, discos, o incluso servidores, realizando un mayor número de operaciones en paralelo; como a la hora de proponer un diseño conceptual, seleccionando un modelo con el que se prevea realizar menos operaciones costosas. Muchas veces en la práctica se deciden aplicar varias de estas alternativas juntas. Algunas de estas técnicas fueron utilizadas en el diseño e implementación de la base de datos del módulo Laboral y a continuación se describen.

2.8.1 Normalización

Al realizar el diseño de una base de datos relacional, se debe considerar que sus tablas cumplan con las reglas de normalización, la cual permite disminuir la redundancia de los datos.

La base de datos relacional del módulo Laboral se encuentra en primera forma normal pues se puede asegurar que cada tupla contiene exactamente un valor para cada atributo de las tablas de la base de datos, o sea, no existen campos multivaluados. La base de datos también está en segunda forma normal, pues se encuentran en primera forma normal y todos los atributos que no son claves en las tablas, dependen totalmente de la clave primaria. Además está en tercera forma normal, al encontrarse en segunda forma normal y no tener dependencias transitivas los atributos no primos.

Algunas tablas se desnormalizaron para lograr flexibilidad en el diseño, la optimización de consultas y

⁷ **Optimización:** es buscar la mejor manera de realizar una actividad.

Capítulo 2. Propuesta de solución

mejorar el acceso a los datos. Por ejemplo la tabla lab_dtramite contiene los atributos sala y tribunal que pertenecen al módulo Administración y Gobierno pero es necesario tenerlos en los trámites para obtener sus respectivos datos, se pudo haber creado dos tablas nuevas con esos valores pero era poco recomendable pues se repetirían valores de entidades creadas ya en otros módulos.

2.8.2 Índices

Los índices utilizados en las bases de datos son como los índices que se usan en los libros en los cuales están presentes los capítulos y la página donde empieza cada uno. En una base de datos los índices son estructuras de datos que mejoran la velocidad de las operaciones en las tablas asociadas, permiten que las búsquedas sean más eficientes pues ofrecen un indicador de búsquedas para las consultas y evitan al sistema gestor de bases de datos que revise todos los datos disponibles para devolver el resultado. Son un grupo de datos asociados con una o varias columnas de la tabla en los que aparece la relación entre el contenido y el número de fila donde está ubicado.

Al insertar índices en las tablas se deben tener en cuenta los siguientes factores:

- Se pueden indexar los campos que son más utilizados en las búsquedas (los que aparecen en las cláusulas WHERE o JOIN) para mejorar una consulta con el operador (SELECT).
- Los índices se deben crear sobre campos con valores únicos pues funcionan de una mejor manera si el campo no tiene valores duplicados.
- Se deben indexar campos con valores de la menor longitud posible, preferiblemente enteros y si se indexa un campo de texto, se debe evitar hacerlo sobre campos de longitud variable.
- Por último se recomienda evitar crear índices innecesariamente pues estos se actualizan con cada cambio en la tabla asociada y pueden ralentizar las modificaciones de la misma.

PostgreSQL tiene los siguientes tipos de índices:

Btree (árbol-B): el árbol-B es un tipo de índice estándar, donde B es sinónimo de equilibrio. Un árbol equilibrado es aquel en el cual la cantidad de datos en el lado izquierdo y derecho de cada división es la misma. El árbol-B se puede utilizar para encontrar un único valor o para explorar en un área de distribución la búsqueda de los valores claves mediante el empleo de los operadores: <, <=, =, >, >=. Funciona tanto en datos numéricos como de texto. Los índices Btree se utilizan para evitar las grandes

Capítulo 2. Propuesta de solución

operaciones de ordenación. Se obtiene un mejor resultado cuando se aplican sobre columnas con una alta cardinalidad, es decir, sobre columnas que tengan muchos valores diferentes. (PostgreSQL Global Development Group, 2009)

Hash: una tabla hash o mapa hash es una estructura de datos que asocia llaves o claves con valores. La operación principal que soporta de manera eficiente es la búsqueda. Las tablas hash son más útiles cuando se almacenan grandes cantidades de información. Estas almacenan la información en posiciones pseudo-aleatorias, por lo que el acceso ordenado a su contenido es bastante lento. El hash sólo puede ser usado para consultas de equivalencia, o sea, para consultas con el operador “=”. No soporta el comando IS NULL, a diferencia del B-tree que sí lo permite. (PostgreSQL Global Development Group, 2009)

GIST: es el acrónimo de Generalized Search Tree (árbol de búsqueda generalizada). Los índices GIST no son un solo tipo de índice, sino más bien una infraestructura dentro de la cual muchas estrategias diferentes de indexación se pueden aplicar. En consecuencia, los operadores concretos con los que un índice GIST pueden ser utilizados, varían dependiendo de la estrategia de indexación (la clase del operador). Por ejemplo, la distribución estándar de PostgreSQL incluye clases de operador de GIST para varios tipos de datos geométricos bidimensionales, que admiten consultas con índices utilizando estos operadores: <<, &<, &>, >>, <<|, &<|, |&>, |>>, @>, <@, ~=, &&, estos son operadores para datos geométricos. Se utilizan para aumentar la velocidad de búsqueda en todos los tipos de estructuras de datos irregulares (arreglos enteros, datos espaciales. etc) que no son posibles de ordenar con un índice de árbol B. (PostgreSQL Global Development Group, 2009)

GIN: es el acrónimo de Generalized Inverted Index (Índice Invertido Generalizado). Los GIN son índices invertidos que pueden controlar los valores que contienen más de una clave, por ejemplo las matrices. Al igual que con GIST, GIN puede soportar muchas estrategias de indexación diferentes definidas por el usuario, los operadores particulares con los que se puede utilizar un índice GIN varían dependiendo de la estrategia de indexación. Por ejemplo, la distribución estándar de PostgreSQL incluye clases de operador GIN para matrices unidimensionales, que admiten consultas utilizando estos operadores: <@, @>, =, &&, estos son operadores para arreglos de datos. (PostgreSQL Global Development Group, 2009)

Capítulo 2. Propuesta de solución

Se podría pensar entonces que la mejor opción es crear un índice por cada combinación de columnas, sin embargo no deben crearse estas estructuras innecesariamente. Además, la creación de demasiados índices puede traer consecuencias no deseadas, por ejemplo:

- Si se modifican valores en la tabla asociados a columnas sobre las que se hayan creado índices, o se insertan o eliminan filas, la estructura del índice se actualiza; esto va a influir en el comportamiento del gestor, pudiendo reducir la velocidad de procesamiento a la hora de realizar dichas operaciones. Aunque las operaciones que mayormente serán realizadas en una base de datos son de lectura, esto se debe tener en cuenta a la hora de realizar las cargas hacia el sistema, donde las operaciones de inserción y modificación son abundantes. Una alternativa que se podría analizar es eliminar los índices antes de comenzar la carga y volverlos a crear después.
- Los índices ocupan espacio en memoria: como los índices se almacenan en ficheros al igual que los datos de una tabla, van a ocupar espacio de almacenamiento físico. Mientras más grande sea una tabla, mayores serán los índices asociados a ella. Por lo tanto, se debe analizar la capacidad de almacenamiento de que se dispone.

En el desarrollo de la base de datos del módulo Laboral se utilizaron índices de tipo B-Tree. Las ventajas que tiene el uso de este tipo de índice están dadas, precisamente, por su estructura. Por ejemplo, las búsquedas de filas en las que un valor en particular aparezca no implican recorrer toda la tabla, sino que se utiliza la estructura arbórea del índice definido. Bajando desde la raíz del árbol sólo es necesario desprenderse por una de las ramas hasta encontrar en las hojas las referencias a las filas en el fichero. Con esto se consume menos tiempo en hallar el resultado y es menor la cantidad de veces que se accede al disco para leer.

La base de datos del módulo Laboral posee implementado el indexado que trae por defecto el gestor PostgreSQL para la búsqueda de datos utilizando las llaves primarias, todas las llaves primarias poseen índices, lo que implica que cualquier búsqueda que se realice utilizándolas se optimizará mediante este método. Además se indexaron atributos de tablas que tendrán grandes volúmenes de datos donde las búsquedas utilizando estos atributos serán muy frecuentes y se muestran a continuación:

Nombre de la tabla	Atributo o atributos indexados
--------------------	--------------------------------

lab_descrito	numeroescrito
lab_dexpediente	numero
lab_dpersona	nacionalidad
lab_dpersonajurídica	nombre
lab_dpersonanatural	primernombre, segundonombre, primerapellido, segundoapellido, ci
lab_dresolucionjudicial	numero
lab_dtelefono	numero

Tabla 14. Tablas y sus atributos indexados.

2.9 Integridad y seguridad de la base de datos

Un buen diseño de la base de datos debe garantizar la consistencia y seguridad de los datos y que la redundancia sea mínima, para lograrlo deben considerarse elementos como las restricciones de integridad y la seguridad de la base de datos.

Restricciones de integridad.

Establecer las reglas correspondientes a la consistencia de los datos requeridos en las tablas, chequeo de la unicidad de determinados atributos, así como las restricciones correspondientes a las llaves primarias y foráneas son aspectos a los que hay que prestar especial atención para garantizar la integridad de los datos.

La integridad de datos presenta las siguientes categorías:

- Integridad de entidad: la llave primaria de una entidad no puede tener valores nulos y siempre deberá ser única, por ejemplo en la tabla lab_dtramite el atributo llave primaria idtramite.
- Integridad de dominio: restringe los valores que puede tomar un atributo respecto a su dominio, por ejemplo en la tabla lab_nPersona el atributo persona solamente puede tomar los siguientes valores: natural o jurídica, estos valores se refieren a los dos únicos tipos de personas que existen en la Materia Laboral.

Capítulo 2. Propuesta de solución

- Integridad referencial: la base de datos no debe contener valores de llaves ajenas sin concordancia.

Análisis de la seguridad.

La seguridad de la base de datos se basa, entre otros aspectos, en acciones para garantizar que la información almacenada sea consultada por los usuarios autorizados. Los datos que guarda una base de datos tienen el riesgo de sufrir ataques que puedan provocar la pérdida o modificación no deseada de la información, por lo cual es sumamente importante establecer los mecanismos necesarios para asegurar la integridad de los mismos, dejando claro los privilegios de acceso de los usuarios. Es importante además garantizar la recuperación de la información en caso de que se produzca algún fallo en la base de datos y se pierda información.

Las bases de datos de cada una de las instancias en los tribunales contemplarán un plan de copias de seguridad de sus datos. En cada uno de los tribunales en las diferentes instancias y en el centro de datos se realizará una copia de seguridad normal o total (con una frecuencia mensual) la cual copia todos los archivos, una copia incremental (con una frecuencia diaria) la cual copia los archivos creados o modificados en el día desde la última copia de seguridad normal o incremental y una copia diferencial (con una frecuencia semanal) que copia los archivos creados o modificados desde la última copia de seguridad normal. Estas copias se realizarán de forma automática a las once de la noche. Como una medida más para preservar la información y aumentar la seguridad, una de las copias normales del centro de datos se almacenará en un local seguro en el Tribunal Supremo y otra en un Banco. (González Flores, Gómez Perdomo, & Fuentes Aguila, 2012)

Para garantizar la seguridad se define un administrador encargado de conceder privilegios a los usuarios y con un acceso total a la base de datos.

2.10 Conclusiones parciales

En este capítulo se describen las principales tablas del módulo Laboral, estas tienen un gran peso en el flujo central de los procesos en la Materia Laboral.

Para el modelado de los datos se utiliza la herramienta ER/Studio en su versión 7.5 con su repositorio

Capítulo 2. Propuesta de solución

para el trabajo en equipo y se utiliza un diccionario de datos para definir los tipos de datos. PostgreSQL en su versión 8.4 se utiliza para la gestión de los datos en la base de datos.

En el diseño de la base de datos relacional del módulo Laboral del SIT se utilizan los patrones de diseño de bases de datos de árboles simples y el de llaves subrogadas, este último está presente en todas las tablas del modelo de datos.

La base de datos está normalizada hasta la tercera forma normal permitiendo establecer un balance entre las redundancias, algunas tablas se desnormalizaron para disminuir el tiempo de respuesta en términos de consultas a la base de datos. La indexación de algunos atributos permite aumentar la eficiencia al realizar consultas a la base de datos.

La seguridad y la integridad de los datos son factores claves para realizar un buen diseño de la base de datos y permiten asegurar la calidad de la información y que el acceso a los datos solo sea posible por usuarios autorizados.

Capítulo 3. Validación de la solución propuesta

3.1 Introducción

Para validar la solución propuesta se realizaron pruebas de volumen para analizar el comportamiento de la base de datos ante volúmenes de datos almacenados, pruebas de rendimiento para observar el tiempo de respuesta de la base de datos ante las consultas y se aplicaron pruebas de estrés simulando varios usuarios conectados a la vez.

3.2 Pruebas de volumen

Las pruebas de volumen son pruebas típicas de entornos que utilicen bases de datos. Las mismas se realizan para analizar el comportamiento de la base de datos con volúmenes de datos almacenados para verificar si alcanza un límite de almacenamiento que pueda causar fallas.

Para tener una idea de la capacidad de almacenamiento de la base de datos es necesario conocer que se digitalizarán archivos judiciales en cada una de las instancias que tienen los 166 Tribunales Municipales Populares, en los 15 Tribunales Provinciales y en el Tribunal Supremo. En los archivos judiciales de los tribunales municipales y provinciales existen expedientes que oscilan entre 10 y 100 folios y en el Tribunal Supremo los expedientes pueden tener entre 10 y 50 folios.

Para el desarrollo de la prueba de volumen la base de datos se pobló con datos introducidos utilizando la herramienta Data Generator (Generador de Datos) 2005 para PostgreSQL. La generación de datos se configuró teniendo en cuenta datos reales de la Materia Laboral extraídos del Informe de Rendición de Cuentas del Tribunal Supremo Popular a la Asamblea Nacional del Poder Popular, redactado en La Habana en diciembre de 2011.

La prueba fue realizada en una máquina con un microprocesador Intel Core 2 Duo de 2.20 gigahertz y una Memoria de Acceso Aleatorio (RAM por sus siglas en inglés) de 1 gigabyte, estas propiedades serán superadas cuando la base de datos esté en explotación en los tribunales por tanto los resultados serán superiores si se repite esta prueba en el sistema desplegado.

Capítulo 3. Validación de la solución propuesta

Todas las tablas de la base de datos se llenaron con los datos estimados que tendrá esta en un período de 5 años. Por ejemplo, a partir del dato que los expedientes radicados en el país en la materia Laboral en el año 2011 fueron 7 430 y conociendo que este es aproximadamente el valor que tomarán anualmente en los próximos 5 años, se redondeó por exceso a 8 000 y se multiplicó por 5 que representa la cantidad de años que se tiene en cuenta para realizar la prueba, obteniendo las 40 000 tuplas que fueron insertadas en la tabla lab_dexpediente.

Utilizando la herramienta Data Generator se acumularon datos en todas las tablas de la base de datos, siendo las que mayores volúmenes de datos acumularon las tablas relacionadas con los trámites, documentos, escritos y resoluciones judiciales. En la tabla lab_dexpediente se insertaron las 40 000 tuplas en un minuto y 19 segundos y la tabla lab_tramite fue la que mayor cantidad de tuplas acumuló con 4 millones demorando 4 horas, 18 minutos y 5 segundos en insertar todos los datos como se muestra en la figura 10.

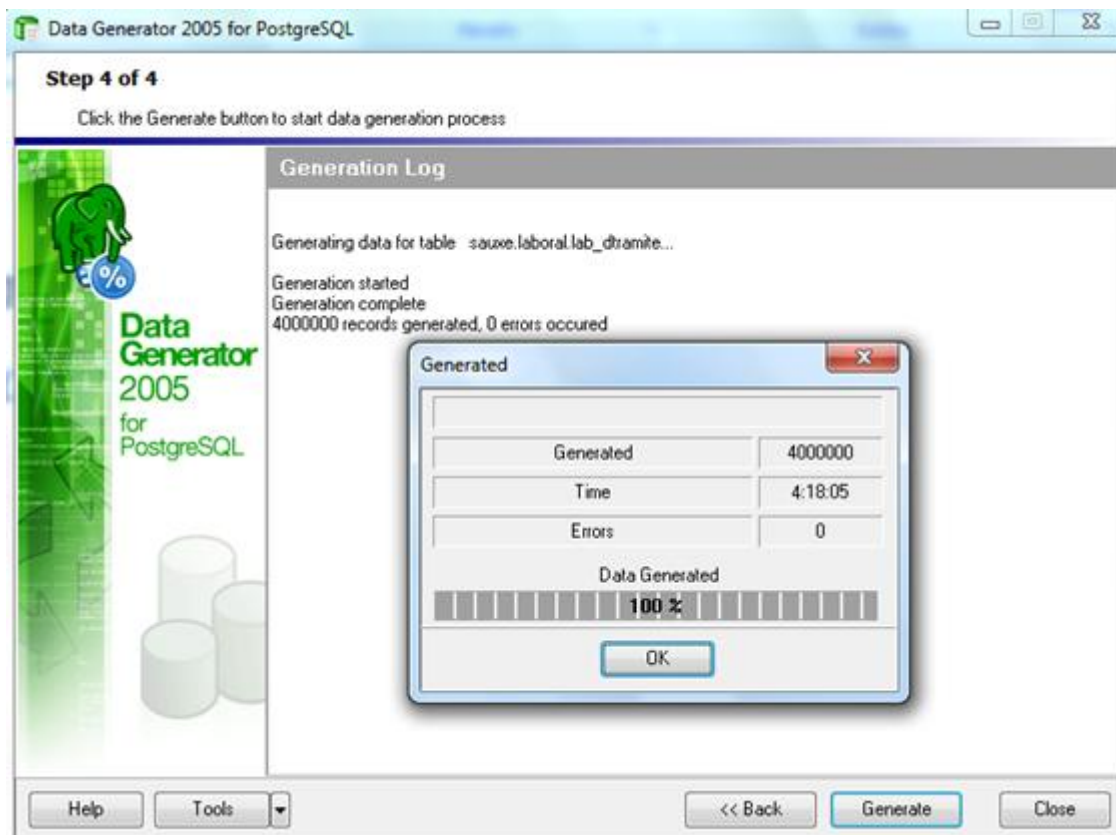


Figura 10. Prueba de volumen para la tabla lab_dtramite.

Al introducir los datos en las tablas no se presentaron problemas de límites de capacidad, desbordamiento de columnas, atributos o tipos de datos. Además el sistema gestor de bases de datos PostgreSQL soporta cualquier cantidad de tuplas en una tabla mientras que esta no exceda los 1.2 terabytes. Los resultados de esta prueba permiten asegurar que el diseño de la base de datos y el gestor utilizado soportan el cúmulo de información que será insertada cuando la base de datos esté en explotación en los Tribunales Populares Cubanos.

3.3 Pruebas de rendimiento

Con estas pruebas se puede obtener como resultado el tiempo de ejecución de las consultas con el objetivo de comprobar el comportamiento de la base de datos y el tiempo de respuesta ante las consultas. Las consultas fueron ejecutadas anteponiendo el comando "EXPLAIN ANALYZE" que permite observar el

Capítulo 3. Validación de la solución propuesta

plan de ejecución.

Las pruebas se realizaron para obtener datos de los trámites, los expedientes y de las personas; por ejemplo para obtener los datos de una persona natural se realizaron las acciones que a continuación se explican.

Primero se realizó un llenado volumétrico a la base de datos en las siguientes tablas y con la cantidad de tuplas que se muestra a continuación:

Entidad	Cantidad de tuplas
lab_dpersona	10 000
lab_dpersonanatural	3 000
lab_ddireccion	10 000
lab_dpersonaparte	2 000
lab_dtelefono	5 000

Tabla 15. Entidades y sus tuplas en la base de datos.

Posteriormente se realizó la siguiente consulta SQL que obtiene los datos de una persona natural, utilizando la herramienta PGAdmin III:

```
EXPLAIN ANALYZE SELECT
```

```
laboral.lab_dpersonanatural.primernombre, laboral.lab_dpersonanatural.primerapellido,
```

```
laboral.lab_dpersonanatural.segundoapellido, laboral.lab_dpersonaparte.profesion,
```

```
laboral.lab_dtelefono.numero, laboral.lab_dpersona.nacionalidad, laboral.lab_dpersona.correoelectronico,
```

```
laboral.lab_ddireccion.direccion, laboral.lab_ddireccion.dpa,
```

```
laboral.lab_dpersonaparte.naturalde, laboral.lab_dpersonanatural.segundonombre,
```

Capítulo 3. Validación de la solución propuesta

laboral.lab_dpersonanatural.ci

FROM

laboral.lab_dpersona, laboral.lab_dpersonanatural, laboral.lab_dtelefono,

laboral.lab_dpersonaparte, laboral.lab_ddireccion

WHERE

laboral.lab_dpersona.idpersona = laboral.lab_ddireccion.idpersona AND

laboral.lab_dpersonanatural.idpersona = laboral.lab_dpersona.idpersona AND

laboral.lab_dtelefono.idpersona = laboral.lab_dpersona.idpersona AND

laboral.lab_dpersonaparte.idpersona = laboral.lab_dpersonanatural.idpersona;

Los resultados que se muestran en la siguiente figura demuestran que la consulta se realizó en un tiempo de 0.997 milisegundos, menor a los 3 segundos que es el tiempo de respuesta establecido en los requisitos funcionales del sistema.

Panel de Salida	
Salida de datos	
Comentar Mensajes Historial	
	QUERY PLAN text
1	Merge Join (cost=18.00..19.26 rows=50 width=570) (actual time=0.255..0.825 rows=20 loops=1)
2	Merge Cond: (lab_dpersona.idpersona = lab_dpersonanatural.idpersona)
3	-> Merge Join (cost=0.00..107.01 rows=100 width=331) (actual time=0.053..0.574 rows=20 loops=1)
4	Merge Cond: (lab_dpersona.idpersona = lab_ddireccion.idpersona)
5	-> Nested Loop (cost=0.00..90.76 rows=100 width=227) (actual time=0.042..0.503 rows=20 loops=1)
6	-> Index Scan using pk_dpersona on lab_dpersona (cost=0.00..14.75 rows=100 width=120) (actual time=0.016..0.074 rows=100 loops=1)
7	-> Index Scan using fk_dpersonanaturaldpersonapart on lab_dpersonaparte (cost=0.00..0.75 rows=1 width=107) (actual time=0.002..0.002 rows=0 loops=100)
8	Index Cond: (lab_dpersonaparte.idpersona = lab_dpersona.idpersona)
9	-> Index Scan using fk_dpersonadireccion on lab_ddireccion (cost=0.00..14.75 rows=100 width=104) (actual time=0.006..0.019 rows=21 loops=1)
10	-> Sort (cost=17.75..17.87 rows=50 width=279) (actual time=0.197..0.199 rows=21 loops=1)
11	Sort Key: lab_dpersonanatural.idpersona
12	Sort Method: quicksort Memory: 30kB
13	-> Hash Join (cost=2.13..16.34 rows=50 width=279) (actual time=0.084..0.130 rows=30 loops=1)
14	Hash Cond: (lab_dpersonanatural.idpersona = lab_dtelefono.idpersona)
15	-> Seq Scan on lab_dpersonanatural (cost=0.00..12.70 rows=270 width=217) (actual time=0.005..0.014 rows=30 loops=1)
16	-> Hash (cost=1.50..1.50 rows=50 width=62) (actual time=0.062..0.062 rows=50 loops=1)
17	-> Seq Scan on lab_dtelefono (cost=0.00..1.50 rows=50 width=62) (actual time=0.005..0.026 rows=50 loops=1)
18	Total runtime: 0.997 ms

Figura 11. Resultados de la prueba de carga utilizando la herramienta PGAdmin III.

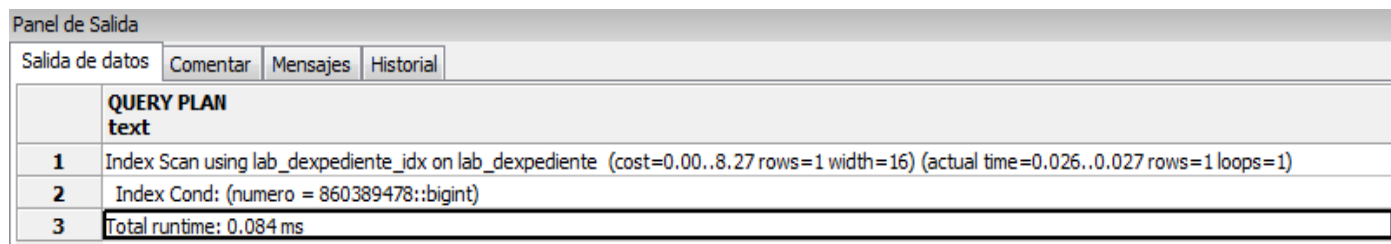
Capítulo 3. Validación de la solución propuesta

Pruebas de rendimiento para los índices.

Además se probaron todos los índices de la base de datos realizando las mismas consultas con los atributos indexados y sin indexar, obteniendo en todos los casos que el tiempo de respuesta utilizando índices era inferior a los casos donde no eran utilizados. Por ejemplo en la tabla lab_dexpediente se utilizó la consulta:

```
EXPLAIN ANALYZE SELECT idexpediente FROM laboral.lab_dexpediente WHERE numero = '860389478';
```

Después de realizar esta prueba se obtuvo como resultado que el tiempo de respuesta de la consulta con el atributo “numero” indexado (0.084 milisegundos) fue menor que cuando se realizó sin indexarlo (16.573 milisegundos) como se muestra en las figuras 12 y 13.

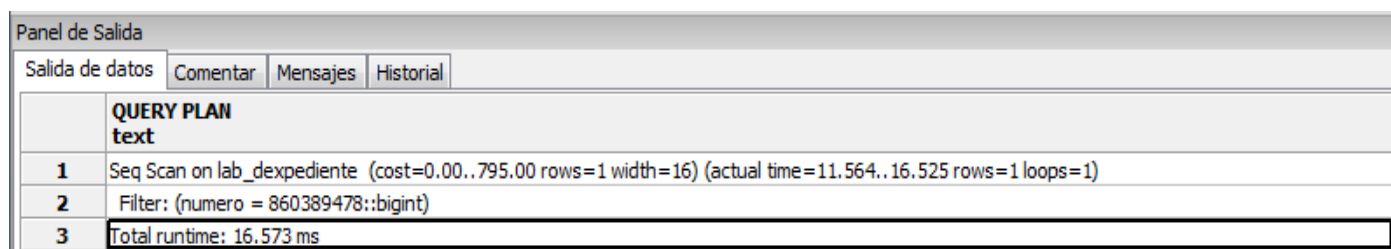


Panel de Salida

Salida de datos Comentar Mensajes Historial

	QUERY PLAN text
1	Index Scan using lab_dexpediente_idx on lab_dexpediente (cost=0.00..8.27 rows=1 width=16) (actual time=0.026..0.027 rows=1 loops=1)
2	Index Cond: (numero = 860389478::bigint)
3	Total runtime: 0.084 ms

Figura 12. Resultados de la prueba con el atributo “numero” indexado.



Panel de Salida

Salida de datos Comentar Mensajes Historial

	QUERY PLAN text
1	Seq Scan on lab_dexpediente (cost=0.00..795.00 rows=1 width=16) (actual time=11.564..16.525 rows=1 loops=1)
2	Filter: (numero = 860389478::bigint)
3	Total runtime: 16.573 ms

Figura 13. Resultados de la prueba con el atributo “numero” sin indexar.

Luego de realizadas las pruebas y valorados los resultados se puede afirmar que las respuestas a las solicitudes realizadas a la base de datos son relativamente rápidas y cumplen con los requisitos funcionales establecidos para el desarrollo del Sistema de Informatización de Tribunales. Otro resultado

Capítulo 3. Validación de la solución propuesta

obtenido fue que los índices optimizan el tiempo de respuesta de las consultas a la base de datos.

3.4 Pruebas de carga y estrés

Las pruebas de carga y estrés someten a la base de datos a un régimen de carga de trabajo por simulación de concurrencia similar al esperado en la explotación real del sistema. El objetivo de estas pruebas es conocer el tiempo de respuesta a las consultas, detectar problemas de control de concurrencia, entre otros inconvenientes.

Para realizar estas pruebas se utilizó la herramienta de distribución gratuita JMeter, desarrollada con el lenguaje de programación Java en el proyecto Jakarta, la cual permite realizar pruebas sobre aplicaciones web. JMeter se destaca por su facilidad de su uso, versatilidad, estabilidad y la variedad de funcionalidades que brinda.

A continuación se muestra una serie de aspectos a tener en cuenta para realizar las pruebas:

- Número de hilos: es el número de usuarios a simular.
- Período de subida (en segundos): tiempo que debiera llevarle a JMeter lanzar todos los hilos (si se seleccionan 10 hilos y el período de subida es de 1 segundo, entonces cada hilo comenzará 0,1 segundo después de que el hilo anterior haya sido lanzado).
- Contador del bucle: número de veces a realizar la prueba.

Propiedades de los hilos	Prueba 1	Prueba 2
Número de hilos	50	200
Período de subida (en segundos)	1	1
Contador del bucle	10	10

Tabla 16. Número de hilos para las pruebas de estrés 1 y 2.

En los informes de cada una de las pruebas se muestran los siguientes indicadores:

- # Muestras: número de muestras de peticiones JDBC (Java Database Connectivity por sus siglas en inglés).
- Media: tiempo medio transcurrido para un conjunto de resultados.

Capítulo 3. Validación de la solución propuesta

- Mediana: mediana aritmética (elemento de una serie ordenada de valores crecientes de forma que la divide en dos partes iguales, superiores e inferiores a él).
- Desviación: desviación estándar (diferencia entre la medida de una magnitud y el valor de referencia).
- Mín: mínimo tiempo transcurrido para las muestras de peticiones JDBC⁸.
- Máx: máximo tiempo transcurrido para las muestras de peticiones JDBC.
- % Error: porcentaje de las peticiones con errores.
- Rendimiento: rendimiento medido en base a peticiones por segundo.

Pruebas	# Muestras	Media	Mediana	Mín	Máx	% Error	Rendimiento
Prueba 1	500	690	375	15	4406	0,00	87,7
Prueba 2	2000	1445	265	15	10630	1,05	81,8

Tabla 17. Resultados de las pruebas 1 y 2.

La primera prueba se realizó simulando 50 usuarios con 10 iteraciones por cada uno, realizando una consulta a la base de datos para un total de 500 peticiones y demoró 0.690 segundos de tiempo de respuesta (dado por la media) en atender la solicitud. Para la segunda prueba se simularon 200 usuarios con 10 iteraciones por cada uno para un total de 2000 peticiones, retornando como tiempo de respuesta 1.445 segundos en atender la solicitud. Por tanto la base de datos soporta una gran cantidad de consultas realizadas simultáneamente y el tiempo de respuesta en atender las peticiones será menor que los 3 segundos establecidos en los requisitos funcionales.

3.5 Conclusiones parciales

Las pruebas de volumen permitieron conocer que la base de datos no presenta problemas con el límite de capacidad ni de desbordamiento de búfer, las pruebas de rendimiento mostraron que las respuestas a las solicitudes son relativamente rápidas y las pruebas de estrés manifestaron que soporta un gran número de conexiones y peticiones de manera concurrente.

⁸ **JDBC**: es una especificación de un conjunto de clases y métodos que permiten a programas desarrollados en Java acceder a sistemas de bases de datos.

Capítulo 3. Validación de la solución propuesta

Las pruebas realizadas demostraron que la base de datos responderá de manera satisfactoria cuando esté en explotación en los Tribunales Populares Cubanos.

Conclusiones generales

Con el desarrollo del presente trabajo se arribó a las siguientes conclusiones generales:

- Las bases de datos jurídicas tienen un gran impacto en el marco legal, entre sus principales características se encuentran que almacenan un enorme volumen de información, precisan de constante actualización y deben ser muy seguras.
- El diseño y la implementación de una base de datos relacional permite gestionar la información que se procesa en el módulo Laboral del Sistema de Informatización de Tribunales.
- A partir del diseño se obtuvo el modelo de datos como artefacto fundamental para el desarrollo de la base de datos del módulo Laboral.
- Las pruebas de volumen, rendimiento y estrés realizadas, demostraron que la base de datos responderá de manera satisfactoria ante gran cantidad de información, las respuestas a las solicitudes son relativamente rápidas y soporta un gran número de conexiones y peticiones de manera concurrente.
- De manera general se obtuvo una base de datos organizada y estructurada que permite el almacenamiento y la gestión de la información jurídica del módulo Laboral.

Recomendaciones

Recomendaciones

- Actualizar el diseño de la base de datos en las próximas fases de desarrollo del proyecto.
- Proporcionar mantenimiento y soporte regular a la base de datos enfocados a su mejor funcionamiento.

Referencias bibliográficas

Alvite Díez, M. L. (abril,2001). *La recuperación de la información en Bases de Datos Jurídicas*. Universidad de León, Madrid, España.

Betancourt Vázquez, A., & Montané Iza guirre, J. (2010). *Documento Visión Tribunales Populares Cubanos versión 1.0*. Universidad de las Ciencias informáticas.

Blahe, M. (2010). *Patterns of Data Modeling*. Estados Unidos: CRC Press.

Departamento de Documentación de Jurisoft. (marzo, 2009). *Requisitos del Sistema Infolex 7. Versión 2.0. España*.

Embarcadero Technologies. 2009. *ER/Studio Enterprise Design & Modeling Suite*. California, Estados Unidos.

Garbey Gainza, Y., & Badillo Pérez., A. (2010). *Documento de configuraciones del sistema operativo y la base de datos del Módulo Penal de los Tribunales Populares Cubanos*. La Habana, Cuba.

García Chávez, C. A. (2005, abril 13). *mailxmail.com*. Retrieved febrero 2012, from mailxmail.com: <http://www.mailxmail.com/curso-diseno-base-datos-relacionales/sistemas-bases-datos>

Gómez Perdomo, Y. (2012). *Documento de Configuraciones para el Sistema de Informatización de Tribunales. La Habana, Cuba*.

González Flores, I., Gómez Perdomo, Y., & Fuentes Aguila, M. (2012). *EL EXPEDIENTE JUDICIAL DIGITAL EN EL SISTEMA DE INFORMATIZACIÓN DE TRIBUNALES*. Universidad de las Ciencias Informáticas, La Habana, Cuba.

Guerra Cabrera, M. N. (2010, febrero). *mailxmail.com*. Retrieved febrero 2012, from mailxmail.com: <http://www.mailxmail.com/curso-diseno-creacion-bases-datos/sistemas-gestion-bases-datos-concepto-diseno-mer-normalizacion-3>

Marcotte, L. (2005, abril). *Linux Journal*. Retrieved mayo 2012, from Linux Journal:

Bibliografía

<http://www.linuxjournal.com/article/7834>

Marqués, M. (enero, 2009). *Bases de Datos*. Valencia, España.

Mato García, R. M. (2005). *Sistemas de Bases de Datos*. CEIS.

Nogales Flores, J. T., Martín Galán, B., & Arellano Pardo, M. d. (2003). *Informática, Derecho y Documentación. Experiencias y posibilidades de aplicación de los lenguajes de marcado de texto (SGML, HTML y XML) a los documentos jurídicos*. Universidad Pontificia de Comillas, Instituto de Informática Jurídica, Madrid, España.

Osorio Rodríguez, A., & Gómez Perdomo, Y. (febrero, 2012). *Entorno de desarrollo de Base de Datos*. Universidad de las Ciencias Informáticas, Tribunales Populares Cubanos, La Habana. Cuba.

Pelaez, C. (2002). *Alfa-Redi*. Retrieved 2012, from Alfa-Redi: <http://www.alfa-redi.org/rdi-articulo.shtml?x=458>

Pérez, M., Valero, E., & Zavala, M. (2011). *Base de Datos Ingeniería de Sistemas*. Universidad politécnica de la fuerza Armada Bolivariana. Universidad politécnica de la fuerza Armada Bolivariana, Zulia, Venezuela.

Pérez Rojas, A., & Montañó Arango, O. (junio, 2007). *Base de reglas activas en un sistema bancario*. Universidad Autónoma del Estado de Hidalgo, México.

PostgreSQL Global Development Group. (2009) *PostgreSQL 8.4.4 Documentation*. California, Estados Unidos.

Pressman, R. (2005). *Ingeniería de Software Un Enfoque Práctico*. La Habana: Félix Varela.

Technologies, E. (agosto, 2009). *ER/Studio Enterprise Design & Modeling Suite*. California, Estados Unidos.

Tomás Diago. (2004). *softonic.com*. Retrieved febrero 2012, from softonic.com: <http://soft-class-para-abogados-2001e.softonic.com>

Zambrano Ramírez, R. (2008). *Sistemas Gestores de Bases de Datos*. Cordoba.

Bibliografía

Bibliografía

Blaha, M. (2010). *Patterns of Data Modeling*. Estados Unidos: CRC Press.

Mato García, R. M. (2005). *Sistemas de Bases de Datos*. CEIS.

PostgreSQL Global Development Group. (2009) *PostgreSQL 8.4.4 Documentation*. California, Estados Unidos.

Pressman, R. (2005). *Ingeniería de Software Un Enfoque Práctico*. La Habana: Félix Varela.