

Universidad de las Ciencias Informáticas

Facultad 3



*Diseño e implementación de la base de datos del módulo
Civil y familia.*

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

Autor:

Yelienny Barroso Mainegra

Tutor(es): Ing. Manuel Alvarez Alonso

Ing. José Carlos Pupo Acosta

Co-tutor: Ing. Yenier Figueroa Machado

La Habana, Curso 2013-2014



La fuerza de uno no vale nada sino está rodeada de la fuerza de todos.

Che



Declaración de autoría

Se declara que Yelienny Barroso Mainegra es el único autor del trabajo de diploma Diseño e implementación de la base de datos del módulo Civil y familia, y se le reconoce a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste se firma la presente a los ____ días del mes de _____ del año 2014.

Firma del autor

Yelienny Barroso Mainegra

Firma del tutor

Ing. Manuel Alvarez Alonso

Firma del tutor

Ing. José Carlos Pupo Acosta

Datos de contacto

Datos del Autor:

Nombre: Yelienny Barroso Mainegra

Correo electrónico: ymainegra@estudiantes.uci.cu

Datos de tutor:

Nombre: Ing. Manuel Alvarez Alonso

Ingeniero en Ciencias Informáticas.

Correo electrónico: malvareza@uci.cu

Datos de tutor:

Nombre: Ing. José Carlos Pupo Acosta

Ingeniero en Ciencias Informáticas.

Correo electrónico: jcpupo@uci.cu

Datos de Co-Tutor:

Nombre: Ing. Yenier Figueroa Machado

Ingeniero en Ciencias Informáticas.

Correo electrónico: yfigueroa@uci.cu

Agradecimientos

A mis padres y a mi hermano, por todo el sacrificio, las preocupaciones, la confianza, la dedicación, la comprensión y sobre todo muchas gracias por su amor incondicional.

A mis sobrinos, porque son lo más grande e importante que tengo en la vida.

A mi familia, por preocuparse y apoyarme siempre, esté donde esté, especialmente a Made por ser mi amiga, mi hermana y mi compañera de toda la vida.

A la gente del barrio, por estar siempre pendientes de mí y desearme lo mejor.

A mis amistades, en especial a Hernán porque sin tu ayuda no hubiese llegado hasta aquí, a Marian, a Grettel, a Yari y a Mari por estar siempre para mí, por soportarme y quererme como soy, y sobre todo por ser las mejores amigas del mundo.

A las muchachitas del 56 especialmente a Eime, Lorena, Jaca, Deborah y Roxana.

A la gente del grupo, los que ya no están aquí y los que como yo siguieron adelante, luchando por llegar hasta el final.

A mis tutores y amigos Manuel y Pupo, a Abel y a Fabra, por sufrir mi estrés y mis dolores de cabeza, por ayudarme a lograr mis metas y por estar siempre ahí para mí.

A los profes del proyecto, a los que me dieron clases y a los que en algún momento me ayudaron, incluyendo a mi tribunal, por apoyarme y por creer en mi sin siquiera conocerme.

A la UCI y a la Revolución por todos los momentos que viví aquí y por la oportunidad de crecer como persona y como profesional.

A cada una de las personas que conocí durante estos cinco años.

Dedicatoria

A mi mamá, por su apoyo incondicional, por ayudarme a levantar después de cada tropiezo y por los consejos que tanto me han ayudado a seguir adelante cuando las cosas se tronaban imposibles.

A mi papá, por cada gota de sudor que ha derramado para que yo pueda cumplir mis sueños, por la confianza, el apoyo y por dejarme seguir mi camino.

A mi hermano, por creer en mí, por ser mi compañero, mi ejemplo y mi orgullo, y porque después de todo no somos tan diferentes.

A mis sobrinos Anita y Alain Miguel, por ser mi sustento, mi razón, mi primer pensamiento al levantarme y los dueños de mi corazón.

Resumen

Las Bases de Datos juegan un papel primordial en cuanto a la recopilación y organización de información, para su uso y manejo por sistemas informáticos. El presente trabajo pretende realizar el diseño y la implementación de una base de datos para el módulo Civil y familia del Sistema de Informatización de la Gestión de las Fiscalías, la cual tendrá el objetivo de mejorar la calidad de los procesos que se desarrollan en las fiscalías de todo el país, permitiendo almacenar y procesar los datos de los procesos civiles y de familia. En el mismo se realizó un estudio de los principales conceptos asociados al desarrollo de bases de datos permitiendo identificar y definir elementos relevantes asociados al diseño e implementación de las mismas. Se caracterizaron las herramientas y tecnologías del entorno de desarrollo. También se definen elementos importantes relacionados con la arquitectura, las configuraciones y la optimización de la solución obtenida, la cual se valida teórica y funcionalmente con el objetivo de garantizar el nivel de calidad deseado.

Palabras claves: Información, base de datos, diseño, implementación y pruebas, rendimiento, integridad.

Índice

Introducción.....	1
Capítulo 1. Fundamentación teórica.....	5
1.1 Introducción.....	5
1.2 Base de datos y sistemas gestores de bases de datos.....	5
1.2.1 Base de datos (BD).....	5
1.2.2 Sistemas gestores de base de datos (SGBD).....	6
1.3 Modelos de datos.....	6
1.4 Diseño de base de datos.....	8
1.4.1 Patrones de diseño de bases de datos.....	9
1.5 Metodologías de desarrollo de Base de Datos.....	9
1.5.1 Dbplanning Framework.....	10
1.6 Índices.....	11
1.7 Normalización de base de datos.....	13
1.8 Integridad en la base de datos.....	14
1.9 Seguridad en la base de datos.....	15
1.9.1 Control de Acceso Basado en Roles.....	16
1.10 Herramientas y tecnologías del entorno de desarrollo.....	16
1.10.1 Lenguaje de modelado UML 2.0.....	17
1.10.3 Visual Paradigm 8.0.....	17
1.10.4 Subversion 1.5.....	17
1.10.5 RapidSVN 0.12.1.....	18
1.10.6 PGAdmin III 1.14.0.....	18
1.10.6 PostgreSQL 9.1.....	19

1.10.7 ORM Doctrine 2.3.2.....	19
1.10.8 IDE NetBeans 7.3.....	19
1.10.9 Datanamic Data Generator for PostgreSQL.....	20
1.10.10 Apache JMeter 2.9.	20
1.11 Conclusiones parciales.	21
Capítulo 2. Análisis de la solución propuesta.....	22
2.1 Introducción.	22
2.2 Arquitectura de datos.	22
2.2.1 Usuarios y privilegios.	22
2.3 Configuración del entorno de desarrollo.....	22
2.3.1 Descripción de las herramientas.	23
2.4 Nomenclatura y normas para el modelo de base de datos.....	23
2.5 Modelos de datos.....	25
2.5.1 Patrones utilizados en el diseño.....	31
2.6 Requisitos transaccionales.....	31
2.7 Acceso a datos.....	34
2.8 Reportes.	36
2.9 Optimización	37
2.9.1 Normalización y denormalización del modelo.....	38
2.9.2 Estrategia de indexado utilizada.....	38
2.10 Conclusiones parciales.	39
Capítulo 3. Validación y pruebas.	40
3.1 Introducción.	40
3.2 Validación teórica.....	40

3.2.1	Integridad de la base de datos.	40
3.2.2	Métricas.	41
3.3	Validación Funcional.	42
3.3.1	Pruebas de rendimiento.	42
3.3.2	Pruebas de volumen.	44
3.3.3	Pruebas de carga y estrés.	45
3.4	Conclusiones parciales.	47
	Conclusiones Generales.....	48
	Recomendaciones.....	49
	Bibliografía.....	50
	Anexos	¡Error! Marcador no definido.

Índice de figuras.

Fig. 6 Esfuerzo de ejecución de cada una de las actividades en cada una de las fases. 11

Fig. 7 Distribución de los datos 22

Fig. 8 Tablas principales del módulo Civil y familia. 26

Fig. 9 Entidad nnotaria del módulo Civil y familia. 31

Fig. 11 Mapeo objeto relacional de entidad RolloExpropiacionForzosa del módulo Civil y familia... 35

Fig. 12 Función listarRolloExpropiacionForzosaPaso 35

Fig. 13 Reporte estadístico provincial Recursos de Casación del módulo Civil y familia. 37

Fig. 14 Entidad drolloexpropiacionforzosa del módulo Civil y familia..... 38

Fig. 15 Índice sobre el atributo salario de la tabla ddatostutor del módulo Civil y familia..... 39

Fig. 16 Tabla drollojvotro del módulo Civil y familia..... 40

Fig. 17 Consulta a realizar como prueba 43

Fig. 18 Resultado de consulta sin indexado..... 43

Fig. 19 Resultado de consulta con indexado..... 44

Fig. 20 Prueba de volumen sobre la tabla drollojurisdiccionvoluntaria del módulo Civil y familia..... 45

Fig. 21 Reporte resumen de prueba de carga y estrés sobre la base de datos Civil y familia. 47

Fig. 22 Modelo de datos del módulo Civil y familia..... **¡Error! Marcador no definido.**

Índice de tablas

Tabla 1 Relación de herramientas y funciones en el entorno de desarrollo..... 23

Tabla 2 Entidad dproceso del módulo Base..... 27

Tabla 3 Entidad dexpedienteprejudicial del módulo Civil y familia..... 27

Tabla 4 Entidad drollojuridiccionvoluntaria del módulo Civil y familia 27

Tabla 5 Entidad drollojyotro del módulo Civil y familia.....	28
Tabla 6 Entidad drollorecursoocasacionfg del módulo Civil y familia.....	28
Tabla 7 Entidad drollorecursoocasacionfp del módulo Civil y familia.....	28
Tabla 8 Entidad drollorevisioncivil del módulo Civil y familia	29
Tabla 9 Entidad drollorecursoapelacionfm del módulo Civil y familia	29
Tabla 10 Entidad drollorecursoapelacionfp del módulo Civil y familia	29
Tabla 11 Entidad drolloprocesosucesorio del módulo Civil y familia.....	29
Tabla 12 Entidad drolloasuntosnotariales del módulo Civil y familia.....	30
Tabla 13 Entidad drollocontencioso del módulo Civil y familia	30
Tabla 14 Entidad drollopcsumariootro del módulo Civil y familia.....	30
Tabla 15 Entidad drollopcordinariootro del módulo Civil y familia.....	30
Tabla 16 Entidad drolloexpropiacionforzosa del módulo Civil y familia.....	31
Tabla 17 Requisitos transaccionales del módulo Civil y familia.....	34
Tabla 18 Descripción de integridad de dominio	41
Tabla 19 Descripción de los principales elementos de hardware de los entornos de prueba y despliegue	44
Tabla 20 Cantidad de trabajadores por tipo de fiscalía	46

Introducción.

El almacenamiento y manejo de información ha sido históricamente y sigue siendo en la actualidad una de las necesidades más importantes de la humanidad. Con el surgimiento y desarrollo de las nuevas Tecnologías de la Información y las Comunicaciones (TIC) se da un gran paso de avance en este ámbito, específicamente las Bases de Datos juegan un papel primordial en cuanto a la recopilación y organización de la información, para su uso y manejo por sistemas informáticos.

La Revolución Cubana tiene como uno de sus principales retos actuales la informatización de disímiles áreas priorizadas que abarcan tareas sociales, políticas y económicas. Uno de los instrumentos más importantes dentro de esta línea lo constituye la Universidad de las Ciencias Informáticas (UCI), que tiene como misión la de *“...formar profesionales comprometidos con su Patria y altamente calificados en la rama de la Informática, producir aplicaciones y servicios informáticos, a partir de la vinculación estudio-trabajo como modelo de formación, servir de soporte a la industria cubana de la informática”*. (UCI, 2011)

Dentro de la UCI el Centro de Gobierno Electrónico¹ (CEGEL) de la Facultad 3 se dedica a la investigación y desarrollo de sistemas informáticos para áreas como la informática jurídica, con el objetivo de mejorar el desempeño de entidades gubernamentales y jurídicas y así lograr un mejor desarrollo de sus funciones para con la sociedad y el estado cubano.

El Sistema de Informatización de la Gestión de las Fiscalías fase 2 (SIGEF II) constituye una solución informática para mejorar los procesos que se desarrollan en las fiscalías de todo el país, con el objetivo de disminuir tiempo y esfuerzo, procurando optimizar la calidad de la tramitación, supervisión y control en tiempo de los procesos fiscales, teniendo como premisa obtener una fuerza fiscal con mayor economía y seguridad en cada uno de sus procesos.

Entre las áreas o subsistemas por las cuales está compuesto el sistema SIGEF II se encuentra Protección de los Derechos Ciudadanos (PDC), que se encarga de controlar y supervisar el proceso de tramitación, supervisión y respuesta en disposición legal a todo ciudadano. A su vez PDC contiene varios módulos correspondientes a las tareas que se realizan en la institución, dentro de ellos se encuentra Civil y familia (CF) el cual tiene como objetivo velar por el cumplimiento de las

¹ Gobierno electrónico: incluye todas aquellas actividades basadas en las tecnologías informáticas 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.

normas jurídicas y principios que regulan las relaciones personales o patrimoniales entre personas privadas o públicas, tanto naturales como jurídicas, siempre que actúen desprovistas de imperio².

El sistema actual de trabajo en este subsistema no satisface las necesidades de la institución, debido fundamentalmente a que se genera un alto volumen de información, la cual debe ser procesada de forma manuscrita, lo que propicia errores en los archivos, unido a que estos tienden a acumularse y a deteriorarse con el tiempo lo que dificulta su utilización, además los reportes que se emiten sobre el estado de los mismos no ofrecen la información necesaria sobre los datos que se manejan.

Para resolver estos problemas respecto al tratamiento de la información del proceso Civil y familia del subsistema PDC, se ha creado un módulo de igual nombre el cual tendrá la capacidad de gestionar gran volumen de información, de forma estructurada y organizada para facilitar su acceso y manipulación. En estos momentos no se cuenta con una estructura que permita almacenar los datos del proceso Civil y familia de manera que este pueda brindar las funcionalidades para la que fue creado.

Dadas las circunstancias anteriores se identifica el siguiente **problema a resolver**: ¿Cómo garantizar el almacenamiento de la información gestionada para el módulo Civil y familia de forma que se contribuya a la integridad de los datos?

Como **objeto de estudio**: diseño e implementación de base de datos (BD) operacionales.

A partir del problema anteriormente descrito se infiere como **objetivo general**: realizar el diseño y la implementación de una BD para el módulo Civil y familia, que contribuya a la integridad de los datos.

Campo de Acción: diseño e implementación de base de datos para sistemas de gestión fiscal.

Idea a defender:

Con el desarrollo del diseño e implementación de la base de datos para el módulo Civil y familia, se contribuirá al almacenamiento y gestión de los datos que generan los procesos del área de Protección de los Derechos Ciudadanos.

Para dar cumplimiento al objetivo general se definen los siguientes **objetivos específicos**:

² Representación legal

- Elaborar el marco teórico de la investigación.
- Diseñar la propuesta de la base de datos para el módulo Civil y familia.
- Implementar la propuesta de la base de datos para el módulo Civil y familia.
- Validar la solución propuesta a través de pruebas de carga y rendimiento.

Para dar cumplimiento a los objetivos específicos se definen como **tareas de la investigación**:

- Estudio del marco teórico que fundamenta el objeto de investigación.
 - Caracterización de las herramientas de modelado.
 - Caracterización del gestor de base de datos.
 - Caracterización del marco de trabajo.
 - Definición de las estrategias de validación del diseño.
- Diseño del modelo de datos.
 - Creación del modelo lógico.
 - Creación del modelo físico.
- Implementación de la base de datos.
 - Definición de Índices.
 - Implementación de dominios y reglas de negocio.
 - Implementación de funciones y tipos de datos personalizados para la inclusión en el Generador Dinámico de Reportes.
 - Implementación de la capa de acceso a datos utilizando Doctrine 2.0.
- Validación de la solución propuesta.
 - Aplicación de la estrategia de prueba seleccionada.
 - Aplicación de la estrategia de validación del diseño seleccionada

Para la realización del presente trabajo se aplicaron los siguientes **métodos de investigación**:

- Métodos teóricos:
 - Histórico-lógico: se empleó para analizar la trayectoria y evolución de los sistemas de bases de datos, elaborando el estudio del estado del arte.
 - Analítico-sintético: permitió el procesamiento de la información y arribar a las conclusiones prácticas y teóricas de la investigación, así como determinar las herramientas utilizadas para el desarrollo 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 desarrollo de bases de datos, con el objetivo de profundizar en las diferentes temáticas a tratar.
 - Experimental: se empleó para crear las condiciones que permiten realizar las pruebas de validación a la solución propuesta.

El siguiente trabajo de diploma está estructurado en 3 capítulos, a continuación una descripción de cada uno de ellos:

Capítulo 1: Este capítulo abarca la fundamentación teórica que sustenta la investigación para el posterior desarrollo de la base de datos, en él se ubica al lector en las tendencias del uso de base de datos, abordando temáticas relacionadas con las características y el funcionamiento de las mismas, además se desarrollan temas relacionados con los modelos de datos y el diseño de bases de datos, así como un análisis de las herramientas que serán utilizadas para el desarrollo de la solución.

Capítulo 2: En este capítulo se describe la propuesta de solución y las características de la base de datos del módulo Civil y familia del sistema SIGEF II, para ello se obtiene el modelo de datos y se muestran las características generales del diseño, ejemplificando los patrones utilizados, además se definen los estándares de nomenclatura utilizados, y se describen los aspectos fundamentales relacionados con la optimización.

Capítulo 3: Este capítulo aborda el proceso de validación teórica y funcional de la base de datos de la solución, teniendo en cuenta la integridad y consistencia de los datos almacenados a partir del diseño de la misma, así como de las pruebas de rendimiento, volumen, carga y estrés realizadas a la aplicación.

Capítulo 1. Fundamentación teórica.

1.1 Introducción.

En el siguiente capítulo se aborda el estudio realizado sobre el desarrollo de bases de datos para sistemas de gestión fiscal, plataforma teórica que sirve de base al desarrollo de este trabajo. Se abarcan los principales conceptos a dominar sobre el desarrollo de base de datos y se incluyen las herramientas que se utilizan para realizar dicho desarrollo. Además se tratan los elementos fundamentales del diseño de base de datos, así como los pasos generales para la normalización y las características de la integridad de los datos.

1.2 Base de datos y sistemas gestores de bases de datos.

1.2.1 Base de datos (BD).

“Una Base de Datos (BD) no es más que un conjunto de datos persistentes que es utilizado por los sistemas de aplicación de alguna empresa dada”. (Date, 2003)

En términos generales, los sistemas de bases de datos tienen las siguientes características: (Riordan, 2000)

- Todos los datos se representan conceptualmente como una disposición ordenada de datos en filas y columnas, lo que se denomina una relación.
- Todos los valores son escalares. Es decir, en cada posición de la relación dada por una fila y una columna hay un único valor individual.
- Todas las operaciones se realizan sobre una relación completa dando como resultado toda una relación, un concepto que se conoce como cierre.

La función básica de las bases de datos es guardar datos de forma organizada, así como dar servicio a las necesidades de las aplicaciones que las utilizan, de modo que los datos deben estar disponibles instantáneamente para diversos usos.

Algunas ventajas de las Bases de Datos son: (Buyens, 2000)

La base de datos es mucho más fácil de mantener que todas las páginas web individuales.

- El empleo de una base de datos facilita la búsqueda de los elementos deseados.
- Una base de datos facilita la presentación de los mismos datos de distintas formas: por categoría, por descripción, por edad o por cualquier otro campo de la base de datos.

En teoría, una base de datos se puede crear desde cero, pero lo normal sería utilizar los servicios de un sistema de gestión de bases de datos.

1.2.2 Sistemas gestores de base de datos (SGBD).

“Un SGBD es una colección de archivos interrelacionados o un conjunto de programas que permiten a los usuarios acceder y modificar estos archivos”. (Rozic, 2004)

El propósito general de un SGBD es proporcionar a los usuarios una visión abstracta de los datos, para ello el sistema esconde ciertos detalles de almacenamiento y mantenimiento de los datos.

El SGBD provee acceso a la información a un alto nivel de abstracción: en lugar de manipular archivos, registros, índices, cilindros, el programa de aplicación se maneja enteramente en términos de clientes, cuentas, saldos, etc., que son traducidos por el SGBD a su implementación física. La base de datos no estará bajo el control de ninguno de los grupos encargados de construir sistemas de aplicación, sino que tal control estará centralizado en un grupo o persona, el administrador de base de datos, quien es responsable de definir, mantener y garantizar la integridad y seguridad de la base de datos. Finalmente, el SGBD proveerá generalmente una aplicación especial llamada lenguaje de consulta que permite a los usuarios finales interrogar la base de datos a discreción sin tener que pasar por el analista o programador como intermediarios. (Mendelzon - Ale, 2000)

Las funciones provistas por un SGBD se pueden agrupar en tres clases: (Mendelzon - Ale, 2000)

- Consulta y actualización de los datos
- Mantenimiento de esquemas
- Manejo de transacciones

Entre los gestores más utilizados en el mundo se encuentran Oracle, MySQL, PostgreSQL, Microsoft SQL Server, entre otros.

1.3 Modelos de datos.

“Cada SGBD provee a sus usuarios un vocabulario básico de conceptos en términos de los cuales estos pueden definir y manipular la información a ser almacenada. Este vocabulario se denomina el modelo de datos”. (Mendelzon - Ale, 2000)

El nivel de abstracción más alto del diseño de una base de datos se encuentra en el modelo de datos, la descripción conceptual del espacio del problema. Los modelos de datos se expresan en función de entidades, atributos, dominios y asociaciones. (Riordan, 2000)

Un modelo de datos se define por los siguientes componentes: (Mendelzon - Ale, 2000)

- Un conjunto de objetos y sus interrelaciones. Esto representa las características estáticas o invariantes e incluye las propiedades de los objetos.
- Un conjunto de operaciones, o lenguaje, que representan las características dinámicas.
- Restricciones sobre los objetos, sus interrelaciones y las operaciones definidas sobre ellos.

Existen diferentes modelos para interpretar y representar los datos y las relaciones entre ellos. Dichos modelos son una abstracción del mundo real y tratan de representar cómo se agrupan los datos y la relación que existe entre ellos en la realidad. A continuación se describen algunos de ellos:

Modelo Entidad-Relación:

El modelo de Entidad-Relación describe los datos en términos de entidades, atributos y relaciones. Los conceptos de este modelo son de alto nivel semántico, cercanos al universo de discurso del usuario y no al del sistema. Diversas herramientas de software conocidas como herramientas CASE³ realizan la traducción en forma automática al modelo relacional, generando las sentencias de creación de la base de datos con sus tablas y restricciones de integridad en SQL.

Modelo relacional:

El modelo relacional describe las bases de datos, tablas, registros, campos y operadores como select, project y join de un modo formal, matemático, también ofrece una base firme para enfocar y analizar formalmente muchos problemas relacionados con la gestión de BD, como el diseño de la BD, la redundancia, la distribución, etcétera. Este modelo es matemáticamente completo (no contiene incoherencias ni le faltan vínculos). Las relaciones representan las entidades que se consideran interesantes en la BD, cada instancia de la entidad encontrará sitio en una tupla de la relación, mientras que los atributos de la relación representan las propiedades de la entidad.

Modelo orientado a objetos:

³ Computer-Aided Software Engineering o Ingeniería de Software Asistida por Computadoras.

El modelo de datos orientado a objetos es una extensión del paradigma de programación orientada a objetos, este se basa en objetos, los cuales contienen valores y métodos, los métodos son fragmentos de código entendidos como órdenes que actúan sobre los valores, en niveles de anidamiento. Los objetos que contienen los mismos tipos de valores y los mismos métodos se agrupan en clases, relacionándose mediante el envío de mensajes.

Modelo semántico:

Un modelo semántico es un mecanismo formal para representar y manipular información de manera general y sistemática. Los modelos semánticos permiten captar el significado de los datos contenidos en la base de datos, es extendido por su simplicidad, rigurosidad y gran capacidad expresiva, además es independiente del almacenamiento y los métodos de acceso, que permiten reflejar fielmente las necesidades de información de una organización.

Modelo jerárquico:

En el modelo jerárquico los datos y relaciones entre los datos se representan mediante registros y enlaces, dichos datos se estructuran en árboles. El tipo de registro que aparece en el tope del árbol se denomina raíz. El modelo jerárquico es de difícil mantenimiento debido a la complejidad de almacenamiento. Además hay dependencia existencial obligatoria entre los registros de un nivel y los del nivel inmediatamente superior y las relaciones son unidireccionales, dichas relaciones son hijo-padre, pero no padre-hijo. Otro problema es que la integridad es costosa de mantener debido a que hay mucha información redundante. Tampoco existe el concepto de campos clave o campos únicos.

Modelo de red:

Los modelos de red son similares a los modelos jerárquicos ya que estos se representan por medio de colecciones de registros y las relaciones entre los datos se representan por medio de enlaces que se pueden ver como apuntadores, aunque poseen una estructura más homogénea. Las relaciones entre conjuntos de entidades se llevan a cabo por medio de unos registros especiales llamados conectores, cada ocurrencia del conector representa una asociación distinta. En este modelo se pueden insertar nuevas entidades en un conjunto de forma independiente.

1.4 Diseño de base de datos.

El diseño de una BD es un proceso complejo que requiere la toma de decisiones a distintos niveles.

Un buen diseño hace que resulte fácil trabajar con la base de datos y ofrece suficiente flexibilidad como para admitir los requisitos futuros. Un diseño pobre obstaculiza las operaciones select, project y join, haciendo que la base de datos resulte poco flexible y sea difícil trabajar con ella. (Buyens, 2000)

Cuando se hace correctamente, el diseño de bases de datos proporciona un almacenamiento y recuperación eficiente de los datos.

1.4.1 Patrones de diseño de bases de datos.

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes.

“Un patrón no es más que un fragmento de un modelo que es recurrente, que deriva como una solución a un problema específico que se ha mantenido a pesar del tiempo y este ha sido documentado en un formato coherente”. (Erl, 2009)

Los patrones brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares. (Tedeschi, 2014)

La utilización de patrones implica identificar aspectos claves de la estructura de un diseño que puede ser aplicado en una gran cantidad de situaciones lo que reduce los esfuerzos de desarrollo y mantenimiento, mejora la seguridad, eficiencia y consistencia del diseño, además de la flexibilidad, modularidad y extensibilidad, factores internos e íntimamente relacionados con la calidad percibida por el usuario. (Cray, 1999)

1.5 Metodologías de desarrollo de Base de Datos

La concepción de una Base de Datos es una tarea larga y costosa por lo que existe la necesidad de contar con procedimientos ordenados que faciliten su desarrollo. Para alcanzar la mejor solución a problemas prácticos, haciendo uso de bases de datos como medio de almacenamiento de la información; se requiere de una metodología que describa un conjunto de pasos.

Las metodologías ágiles, por la propia naturaleza de las mismas, no tienen distinción entre desarrollo de aplicaciones y desarrollo de base de datos dentro de dicho desarrollo de aplicaciones, ya que es usual que los propios desarrolladores realicen todas las funciones de base de datos, además no ofrecen detalles generales aplicables al desarrollo de la Base de Datos. Específicamente de Agile Data Method es un buen enfoque, ya que define un grupo de buenas

prácticas desde el punto de vista ágil, dígame un conjunto de técnicas y filosofías que permitirán trabajar de manera efectiva, pero no define una continuidad durante todo el ciclo de desarrollo.

Las metodologías tradicionales se focalizan en documentación, planificación y procesos. En el caso de RUP⁴ se plantea el desarrollo de bases de datos como una tarea de diseño durante todo el ciclo de desarrollo y no como una disciplina dentro de su propia metodología con actividades específicas. Oracle Unified Process es una metodología desarrollada por Oracle, basada en RUP, para estructurar el desarrollo solo de aplicaciones desarrolladas por esta compañía, incluyendo bases de datos.

Dbplanning Framework no es una metodología, sino un marco de trabajo, el cual posibilita al equipo de desarrollo establecer una línea base de desarrollo de base de datos. (Osorio, et al., 2012) Mediante su aplicación es posible una coordinación justificada e intencionada de los principales elementos de base de datos dentro del proceso de construcción de software. (Osorio, 2012)

El estudio realizado constató que las metodologías de desarrollo de software referidas no abordan de manera específica el desarrollo de base de datos, por lo que no son las más adecuadas para la solución de base de datos para el módulo Civil y familia del Sistema de Informatización de la Gestión de las Fiscalías. En cambio Dbplanning Framework ofrece una alternativa viable, ya que define un conjunto de elementos que permitirán dar seguimiento a todas las fases del proceso de construcción de base de datos, por lo que satisfase las necesidades de desarrollo de la solución.

1.5.1 Dbplanning Framework.

Dbplanning transita por las fases de Inicio, Desarrollo y Despliegue. Plantea 4 actividades las cuales se desarrollan durante las 3 fases definidas: (Osorio, 2012)

1. Modelado de Datos: estructurar evolutivamente el Modelo de Datos incluyendo su documentación.
2. Configuraciones: definir y aplicar las configuraciones del Sistema Operativo – SGBD – BD para diferentes escenarios mediante automatizaciones. Así como otras configuraciones de terceras aplicaciones que interactúen con la base de datos.

⁴ Rational Unified Process o Proceso Unificado de Desarrollo de Software

3. Implementación: implementar las funcionalidades de almacenamiento y recuperación de la información de la base de datos incluyendo su documentación embebida en el código fuente.
4. ADTP⁵: estructurar el acceso a datos garantizando seguridad y optimización en los tiempos de respuesta.

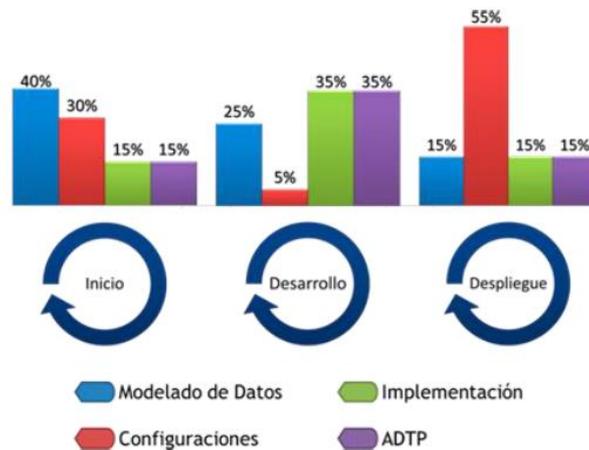


Fig. 1 Esfuerzo de ejecución de cada una de las actividades en cada una de las fases.

Dbplanning asume los roles de Arquitecto de Datos, Administrador de Base de Datos y Desarrollador de Base de Datos para la ejecución de las actividades y la elaboración de los 6 artefactos que define: (Osorio, 2012)

1. Arquitectura de Datos
2. Modelo de Datos
3. Configuraciones
4. Código Fuente
5. Scripts de datos
6. Acceso a Datos

Dbplanning no plantea ningún elemento innovador de forma individual, sino un seguimiento del desarrollo de base de datos dentro del ciclo de desarrollo de software con el objetivo de ganar en productividad. (Osorio, 2012)

1.6 Índices

⁵ Acceso a Datos, Tuning (Optimización) y Prueba.

“Un índice es una estructura auxiliar que permite mejorar el rendimiento de las consultas realizadas mediante la disminución de las entradas y salidas realizadas para recuperar la información de dicha consulta”. (Rozic, 2004)

Los índices se pueden almacenar junto a los datos o separados de ellos. Si la tabla sobre la que se realiza la consulta no posee aplicado ningún índice, el motor de la base de datos se encontrará obligado, para cada consulta que se realice sobre ella a recorrer todos los datos de manera secuencial, en cambio si la tabla posee un índice aplicado se minimizará la cantidad de operaciones de lectura sobre la tabla. Cuanto mayor es el volumen de datos albergados por una tabla más se aprecia la utilización de los índices para mejorar el tiempo de acceso a la información. (Rozic, 2004)

Existen diferentes estrategias de indexado algunas de las cuales se relacionan a continuación:

- La estrategia árbol-B es una implementación de los b-trees de alta concurrencia de Lehman-Yao, proporciona un rápido acceso a los datos por medio de la búsqueda de un valor clave del índice y agrupan registros con claves similares, además dichos árboles se gestionan de tal manera que se insertan ramas según sea necesario, de modo que al recorrerlo hacia abajo para buscar un dato y encontrar un registro específico sea necesario unos pocos accesos sus diferentes páginas.
- Los árboles-R se implementan utilizando el algoritmo de partición cuadrática de Guttman, son estructuras de datos de tipo árbol similares a los árboles-B, con la diferencia de que se utilizan para indexar información multidimensional; por ejemplo, las coordenadas (x, y) de un lugar geográfico.
- El Árbol de Búsqueda Generalizada (GiST del inglés Generalized Search Tree) es una estructura en árbol balanceado. Este no es un solo tipo de índice, sino más bien una infraestructura dentro de la cual muchas estrategias diferentes de indexación se pueden aplicar. Los GiST son una generalización de las formas árbol-R.
- Índices GIN⁶, no son más que 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.
- Índices Hash, son una implementación de las dispersiones lineales de Litwin, estos índices representan una tabla o mapa hash, es una estructura de datos que asocia llaves o claves

⁶ Generalized Inverted Index o Índice Invertido Generalizado.

con valores, la operación principal que soporta de manera eficiente es la búsqueda pero un índice Hash no puede manejar búsquedas por rangos.

Estos métodos de acceso son completamente dinámicos, y no necesitan ser optimizados periódicamente. Actualmente, solo el método de acceso árbol-B soporta índices multi-columna y se pueden especificar hasta 7 claves. (PostgreSQL, 2013)

1.7 Normalización de base de datos

“En una tecnología relacional, el proceso de organizar los campos en tablas se denomina normalización”. (Mendelzon - Ale, 2000)

El proceso de obtención de un modelo de datos totalmente normalizado involucra la eliminación de la redundancia mediante la división de las relaciones, de tal forma que las relaciones resultantes se puedan combinar sin que se pierda ninguna información. (Riordan, 2000)

La teoría de la normalización se ha desarrollado para obtener estructuras de datos eficientes que eviten las anomalías de actualización. La normalización es la expresión formal del modo de realizar un buen diseño y provee los medios necesarios para describir la estructura lógica de los datos en un sistema de información.

Las formas normales (FN) proveen a los diseñadores de BD de lo siguiente:

- Un marco formal para analizar los esquemas de relación de la BD con sus claves y las dependencias funcionales entre sus atributos.
- Una serie de pruebas que pueden efectuarse sobre los esquemas de relación individualmente, de modo que la BD relacional pueda normalizarse hasta el grado deseado. Cuando una prueba falla, la relación que provocó el fallo debe descomponerse en relaciones que individualmente satisfagan las pruebas de normalización. Las FN apartadas de otros factores no garantizan un buen diseño de la Base de Datos.

Primera forma normal:

Una relación está en 1ra FN si cumple la propiedad de que sus dominios no tienen elementos que, a su vez, sean conjuntos. La 1ra FN prohíbe los atributos multivaluados, los atributos compuestos y sus combinaciones (relaciones anidadas), además no permite tener un conjunto de valores, una tupla de valores o una combinación de ambas como valor de un atributo para una tupla individual, tampoco permite listados como valores de atributos, ni las “relaciones dentro de relaciones” o “relaciones como atributos de tuplas”, solo se permiten valores atómicos.

Segunda forma normal:

Cada campo no clave debe depender de todos los campos de la clave principal. Esta forma también especifica que dos o más tablas no pueden tener la misma clave principal. Si su diseño tiene dos o más tablas con la misma clave principal, debe combinar dichas tablas en una.

Tercera forma normal:

Ningún campo no clave depende de otro campo no clave. Igual que ocurre con la segunda forma normal, para conseguir la tercera forma normal hace falta llevar los campos que no la cumplan a una tabla más adecuada.

Es importante señalar que para que una relación esté en una determinada forma normal antes debe estar en todas las formas normales que la preceden.

Usualmente es suficiente realizar el proceso hasta la 3FN para asegurar, la no ocurrencia de problemas de redundancia en la actualización.

La denormalización es el proceso de procurar optimizar el desempeño de una base de datos permitiendo agregar datos redundantes con el objetivo de aumentar la eficiencia con respecto al tiempo de respuesta de las consultas realizadas sobre dicha información.

La denormalización 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, 2009).

Un modelo de datos denormalizado no es lo mismo que un modelo de datos que no ha sido normalizado, y la denormalización debe tomar lugar solamente después de que haya ocurrido un nivel satisfactorio de normalización y de que hayan sido creadas las restricciones y/o reglas requeridas para ocuparse de las anomalías inherentes en el diseño.

1.8 Integridad en la base de datos.

Una de las funciones más importantes de un sistema gestor de base de datos relacional es preservar la integridad de los datos almacenados en la mayor medida posible.

“La integridad se encarga de asegurar que las operaciones ejecutadas por los usuarios sean correctas y mantengan la consistencia de la base de datos”. (Hansen, et al., 1997)

La propiedad de integridad es un aspecto clave que debe prevalecer sobre la información que se maneja garantizando la conservación de la misma libre de modificaciones no autorizadas. (Machado, 2012)

Para que exista integridad en un modelo relacional deben cumplirse cuatro condiciones fundamentales (UCV, 2013):

Datos requeridos:

Establece que una columna tenga un valor no nulo. Se define efectuando que la declaración de una columna es NOT NULL cuando la tabla que contiene las columnas se crea por primera vez, como parte de la sentencia CREATE TABLE.

Chequeo de validez:

Cuando se crea una tabla donde en cada columna tiene un tipo de datos y el sistema gestor de base de datos asegura que solamente los datos del tipo especificado sean ingresados en la tabla.

Integridad de entidad:

Establece que la clave primaria de una tabla debe tener un valor único para cada fila de la tabla; sino, la base de datos perderá su integridad. Se especifica en la sentencia CREATE TABLE. El sistema gestor de base de datos comprueba automáticamente la unicidad del valor de la clave primaria con cada sentencia INSERT Y UPDATE. Un intento de insertar o actualizar una fila con un valor de la clave primaria ya existente fallará.

Integridad Referencial:

La integridad referencial es un sistema de reglas que utilizan la mayoría de las bases de datos relacionales para asegurarse que los registros de tablas relacionadas son válidos y que no se borren o cambien datos relacionados de forma accidental produciendo errores de integridad, esta implica que no existan en el modelo relacional valores para las claves foráneas sin coincidencia con los valores de las claves primarias asociadas a ella.

1.9 Seguridad en la base de datos.

La seguridad es una de las temáticas de mayor importancia a tomar en cuenta al momento de crear una base de datos ya que el resguardo de la información es de vital importancia, (Rozic, 2004) esta es responsabilidad del Administrador de Base de Datos.

Existen varios elementos importantes que se debe tener en cuenta a la hora de gestionar la seguridad en una Base de Datos:

- Control de acceso: consiste en evitar manipulaciones no deseadas, ya que no todos los usuarios de un sistema de bases de datos deberían poder acceder a todos los datos, los SGBD gestionan la seguridad con el uso de claves para identificar al personal autorizado a acceder y manipular la base de datos.
- Copia de seguridad: es un mecanismo de resguardo que posee la información de toda la base de datos en cuestión, no solamente se resguardan los datos que posee la base de datos, sino que también se guarda o se puede guardar información sobre los usuarios, el esquema de la base de datos, los distintos elementos que la componen y las operaciones que se realizaron sobre ella por los diferentes usuarios de la base. Existen diferentes tipos de copia de seguridad como son: completa, diferencial e incremental, el tipo de copia o copias a realizar y la frecuencia de las mismas es decisión del administrador de la BD, en dependencia de las características propias de cada sistema.
- Restaurar la base de datos: el tipo de restauración a realizar queda totalmente condicionado al tipo de copia de seguridad que se genere previamente, para realizar la restauración de una base de datos, esta no debe encontrarse en uso por ningún usuario.

1.9.1 Control de Acceso Basado en Roles

El Control de Acceso Basado en Roles (RBAC) es un enfoque para restringir acceso al sistema solo a usuarios autorizados, este proporciona control granular sobre qué usuarios pueden realizar acciones específicas en objetos determinados de la base de datos.

El control de acceso basado en roles ofrece una alternativa más segura al modelo de superusuario, ya que puede aplicar una política de seguridad en un nivel más específico. RBAC utiliza el principio de seguridad del privilegio mínimo, que significa que un usuario dispone exactamente de la cantidad de privilegios necesaria para realizar un trabajo, además recopila las capacidades de superusuario en perfiles de derechos que se asignan a cuentas de usuario especiales denominadas roles. Luego, un usuario puede asumir un rol para realizar un trabajo que requiere algunas de las capacidades de superusuario. (Oracle, 2014)

1.10 Herramientas y tecnologías del entorno de desarrollo.

La arquitectura base del proyecto SIGEF II para el desarrollo de la aplicación tiene definido un conjunto de herramientas específicas, estas en su mayoría son software libre y en muchos casos multiplataforma, de manera que se tribute a la política de migración hacia software libre que tiene la Fiscalía General de la República, enmarcada en la estrategia del país que tiene su base en la

soberanía tecnológica (Machado, 2012), además la potencialidad de la multiplataforma lo que permite ejecutarse en varios Sistemas Operativos.

1.10.1 Lenguaje de modelado UML 2.0.

El Lenguaje de Modelado Unificado UML es la especificación más utilizada de OMG (consorcio de estándares de tecnología sin fines de lucro), y la forma en que el mundo modela no solo la estructura de la aplicación, el comportamiento y la arquitectura, sino también los procesos de negocio y la estructura de datos. (UML, 2013)

1.10.2 UML 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 los desarrolladores y clientes del software, y simplifica el proceso de análisis y diseño del mismo. (Pressman, 2005)

1.10.3 Visual Paradigm 8.0.

Visual Paradigm es una herramienta de diseño de software, diseñado para proyectos de software ágiles. Soporta estándares de modelado como UML, SysML, ERD, DFD, BPMN, ArchiMate, etc. Facilita la construcción de software y sistemas que se destacan en la experiencia del usuario mediante el apoyo a la identificación, el uso eficaz de los casos, la recopilación de requisitos, el flujo de los acontecimientos, la generación de especificación de requisitos, etc. (VisualParadigm, 2014)

Se utilizará la herramienta CASE Visual Paradigm teniendo en cuenta sus ventajas para modelar los diferentes artefactos que se obtienen en los flujos de trabajo y sus diferentes fases. Las restricciones propias del diseño radican en las pautas que se establecerán, así como las diferentes relaciones que se formen durante el modelado. (Machado, 2012)

Visual Paradigm también ofrece: (VisualParadigm, 2014)

- Navegación intuitiva entre la escritura del código y su visualización
- Documentación automática Ad-hoc
- Ambiente visualmente superior de modelado
- Sofisticado diagramador automáticamente de layout
- Sincronización de código fuente en tiempo real

1.10.4 Subversion 1.5.

Subversion es un sistema de control de versiones de código abierto desarrollado como un proyecto de la Fundación de Software Apache, ha gozado y sigue gozando de la adopción generalizada tanto en el campo de código abierto como en el mundo corporativo. (Apache Software Foundation, 2014)

Subversion puede acceder al repositorio a través de redes, lo que le permite ser usado por personas que se encuentran en distintas computadoras. A cierto nivel, la posibilidad de que varias personas puedan modificar y administrar el mismo conjunto de datos desde sus respectivas ubicaciones fomenta la colaboración. Se puede progresar más rápidamente sin un único conducto por el cual deban pasar todas las modificaciones. Y puesto que el trabajo se encuentra bajo el control de versiones, no hay razón para temer porque la calidad del mismo vaya a verse afectada, si se ha hecho un cambio incorrecto a los datos, simplemente deshace ese cambio. (Postgresql Cuba, 2012).

1.10.5 RapidSVN 0.12.1

RapidSVN es una plataforma de interfaz gráfica de usuario para el sistema de revisión de Subversion escrito en C + + utilizando el marco de wxWidgets. RapidSVN está licenciado bajo la licencia GNU GENERAL PUBLIC v3. (RapidSVN, 2014)

Características: (RapidSVN, 2014)

- Simple - proporciona una interfaz fácil de usar para las funciones de Subversion
- Eficiente - sencillo para los principiantes, pero lo suficientemente flexible como para aumentar la productividad de los usuarios de Subversion experimentados
- Portable - funciona en cualquier plataforma en la que Subversion y wxWidgets pueden correr: Linux, Windows, Mac OS / X, Solaris, etc.
- Rápido - completamente escrito en C + +
- Multilingüe - ha sido traducido a muchos idiomas.
- Soporte completo para Unicode.

1.10.6 PGAdmin III 1.14.0

Es el más popular y rico en características de administración de código abierto y una plataforma de desarrollo para PostgreSQL, la base de datos más avanzada de código abierto en el mundo. La aplicación puede utilizarse en Linux, FreeBSD, Solaris, Mac OSX y las plataformas de Windows para administrar PostgreSQL 7.3 y superiores que se ejecuta en cualquier plataforma, así como las

versiones comerciales y derivados de PostgreSQL. Es desarrollado por una comunidad de expertos de PostgreSQL en todo el mundo y está disponible en más de una docena de idiomas. Es un software libre publicado bajo la Licencia PostgreSQL. Además cuenta con una amplia documentación. (pgAdmin, 2013)

1.10.6 PostgreSQL 9.1.

PostgreSQL es un sistema gestor de base de datos objeto-relacional. Tiene más de 15 años de desarrollo activo y una arquitectura probada que le ha ganado una reputación fuerte en cuanto a fiabilidad, integridad de los datos y exactitud. Se ejecuta en casi todos los principales sistemas operativos: Linux, Unix, BSDs, Mac OS, Beos, Windows, etc. Tiene soporte para llaves foráneas, joins, vistas, triggers y procedimientos almacenados (en múltiples lenguajes). Tiene interfaces nativas de programación para C/C ++, Java, .Net, Perl, Python, Ruby, Tcl, ODBC, entre los otros, y una documentación excepcional. El código fuente de PostgreSQL está disponible bajo una licencia de código abierto. (PostgreSQL, 2013)

1.10.7 ORM Doctrine 2.3.2.

Es un marco de trabajo de mapeo de objeto relacional (ORM) para *PHP* 5.3.0 o versiones superiores, el cual proporciona persistencia transparente de objetos PHP, situándose en la parte superior de una poderosa capa de abstracción de base de datos (DBAL por DataBaseAbstractionLayer). La principal tarea de los ORM para PHP es la traducción transparente entre objetos (PHP) y las filas relacionales de la base de datos. (Blanco, 2012)

La librería Doctrine proporciona herramientas para simplificar el acceso y manejo de la información de la base de datos. (Fabien Potencier, 2013)

1.10.8 IDE NetBeans 7.3.

NetBeans IDE es un entorno de desarrollo, una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas, está escrito en Java pero puede servir para otros lenguajes de programación, también existen un número importante de módulos para extender el NetBeans IDE, además es un producto libre y gratuito sin restricciones de uso. (NetBeans Community, 2014)

Netbeans (7.2) es un IDE que tiene fuerte integración con php y demás herramientas a utilizar en el proyecto, es libre y bien ajustado a las condiciones de desarrollo del centro. Existe gran

conocimiento por parte del equipo de desarrollo, lo que garantiza mayor productividad. (Blanco, 2012)

Características fundamentales: (NetBeans Community, 2014)

- El mejor soporte para las últimas tecnologías de Java (apoyo integral para las tecnologías Java más recientes y las últimas mejoras de la especificación de Java antes de que otros IDEs).
- Rápida y elegante edición de código (guiones líneas, combina palabras y soportes, y destaca código fuente sintáctica y semánticamente, proporciona plantillas de código, consejos de codificación, y las herramientas de refactorización).
- Soporta muchos lenguajes desde Java, C / C + +, XML y HTML, PHP, Groovy, Javadoc, JavaScript y JSP. Debido a que el editor es extensible, se puede conectar en el apoyo a muchos otros idiomas.
- Gestión de Proyectos Fácil y Eficiente (proporciona diferentes vistas de los datos, herramientas útiles para la creación y gestión eficiente de aplicaciones).

1.10.9 Datanamic Data Generator for PostgreSQL.

Datanamic Data Generator for PostgreSQL es un generador de datos que incluye características útiles para la ejecución de pruebas a bases de datos PostgreSQL entre las que se pueden mencionar: (Datanamic, 2014)

- Acceso directo a bases de datos PostgreSQL (con controlador nativo).
- Rellenar la base de datos directamente o generar scripts de inserción.
- Soporte para PostgreSQL versión 7,8 y 9.
- Incorpora en los generadores más de 40 idiomas.
- Carga datos aleatorios de fuentes tanto externas como internas.
- Previsualización en tiempo real de los datos que se generan.
- Alto rendimiento con miles de inserciones por segundo.
- Configuración para definir el porcentaje de valores nulos.

1.10.10 Apache JMeter 2.9.

Apache JMeter es un software de código abierto, una aplicación Java puro 100% diseñado para cargar el comportamiento funcional de prueba y medir el rendimiento. Puede ser utilizado para probar el rendimiento tanto en recursos estáticos como dinámicos (archivos, lenguajes dinámicos

Web como PHP, Java, ASP, .NET, etc., Objetos Java, bases de datos y consultas, servidores FTP y más). Se puede utilizar para simular una carga pesada en el servidor, grupo de servidores, la red o el objeto a probar su resistencia o para analizar el rendimiento general bajo diferentes tipos de carga. Se puede utilizar para hacer un análisis gráfico de rendimiento o para probar el comportamiento de su servidor. (Apache Software Foundation, 2014)

1.11 Conclusiones parciales.

Con el desarrollo de este capítulo se evidencia un estudio detallado de los principales conceptos sobre el desarrollo de base de datos, lo que permite concluir que:

- Los modelos de datos deberán determinar la organización, estructura y rendimiento de las bases de datos, para ello se realizarán el modelo lógico y físico de la Base de Datos.
- La metodología a utilizar será Dbplanning framework ya que facilita el desarrollo de la solución permitiendo establecer una línea que guiará dicho proceso.
- La normalización se llevará hasta la 3ra FN, con el objetivo de mejorar la independencia y reducir la redundancia de los datos.
- Se deben identificar las tablas en las que sea necesario aplicar la denormalización, para reducir el esfuerzo y el tiempo de respuesta en términos de consultas a las bases de datos.
- Se deben aplicar medidas para garantizar el aseguramiento de la integridad de los datos.
- Las herramientas que se describen en el desarrollo de este capítulo cumplen con los requisitos necesarios para desarrollar la solución ya se adaptan a las necesidades del sistema y responden a la política de migración del país hacia el software libre.

Capítulo 2. Análisis de la solución propuesta.

2.1 Introducción.

El siguiente capítulo aborda especificaciones del desarrollo de la base datos de la solución propuesta, incluyéndose las configuraciones del entorno de desarrollo, arquitectura de datos, nomenclatura y normas que se deben aplicar al modelo para hacer más entendible el resultado final. Además se exponen las descripciones sobre los datos, los patrones utilizados, las formas de acceso y las estrategias de optimización realizadas para mejorar el desempeño de la base de datos de la solución obtenida.

2.2 Arquitectura de datos.

Como parte del desarrollo y despliegue de la aplicación SIGEF II, la Fiscalía General de la República ha concebido la existencia de un flujo información a lo largo de todas las fiscalías del país, por lo que los servidores de bases de datos estarán situados en cada una de dichas fiscalías. Se partirá de un servidor central en la Fiscalía General de la República, el cual se comunicará con las fiscalías provinciales y estas a su vez con las fiscalías municipales en el nivel más bajo.



Fig. 2 Distribución de los datos

2.2.1 Usuarios y privilegios.

Para el control de usuarios sobre el servidor de BD se utilizó el patrón Control de Acceso Basado en Roles (RBAC) que implementa el gestor de base de datos. Se crearon diferentes roles, el rol administrador tiene permisos para administrar la base de datos en su totalidad, y el rol usuario que contiene privilegios de actualización, inserción y eliminación de datos.

2.3 Configuración del entorno de desarrollo.

Teniendo en cuenta el entorno real de despliegue de la aplicación se concibe un entorno de desarrollo con características similares, el mismo cuenta con un repositorio para el control de

versiones y un servidor compartido en el cual se alojan el servidor de base de datos y el servidor de aplicaciones los cuales cuentan con 2 GB de memoria RAM y 160 GB de espacio en disco duro, cada uno con Ubuntu LTS 12.04 como sistema operativo, además los servidores y estaciones de trabajo disponen de las mismas aplicaciones citadas en Capítulo 1 y que fueron definidas en la Arquitectura de Datos del sistema.

2.3.1 Descripción de las herramientas.

Para desarrollar el modelo de datos del módulo Civil y familia de SIGEF II se utilizó la siguiente configuración.

Herramienta	Función
Subversion 1.5	Repositorio para el versionado del modelo
RapidSVN 0.12.1	Gestión de versiones del repositorio
Visual Paradigm 8.0	Diseño del modelo de datos
PGAdmin III 1.14.0	Cliente de base de datos
PostgreSQL 9.1	Gestor de base de datos
IDE NetBeans 7.3	Desarrollo del acceso a datos

Tabla 1 Relación de herramientas y funciones en el entorno de desarrollo

Las herramientas de modelado y desarrollo, NetBeans 7.3 y Visual Paradigm 8.0, incluyen opciones de conexión al repositorio de versiones, posibilitando actualizar diariamente los cambios desde las mismas, la herramienta para guiar este proceso será el RapidSVN.

2.4 Nomenclatura y normas para el modelo de base de datos.

Se define una nomenclatura para todas las entidades y sus atributos en el modelo con el objetivo de lograr un mejor entendimiento de la base de datos para todos los involucrados y proporcionar un mejor manejo y gestión de los datos.

Generalidades:

Capítulo 2. Análisis de la solución propuesta

- Todos los caracteres para nombrar atributos y entidades estarán escritos con minúscula.
- No están permitidos caracteres como tildes, puntos, ñ.

Nomenclatura para entidades:

- Los nombres serán concisos y claros con el negocio, no excederán los 35 caracteres, además de adaptarse a los nombres definidos en el diagrama de clases.
- Los nomencladores estarán descritos inicialmente por una n y a continuación el nombre identificativo de la entidad especificada en el diagrama de clases.
- Las tablas de datos variados tendrán de prefijo una d y a continuación el nombre identificativo de la entidad, especificada en el diagrama de clases.
- Las relaciones de mucho a mucho donde siempre se genera una nueva tabla estará compuesto el nombre por el de las dos tablas separados por una guion bajo.
- Los nombres compuestos se escribirán juntos.

Nomenclatura para atributos:

- Los atributos tendrán nombres identificativos y claros de lo que representan y no excederán los 35 caracteres.
- Las llaves primarias tendrán como prefijo id seguido de un guion bajo y el nombre de la tabla.
- Los nombres de los atributos compuestos estarán separados por un guion bajo.
- Para atributos con nombres compuestos la nomenclatura será (prefijo id, para las llaves primarias) separar las palabras con guion bajo, tratando siempre de no superar los 35 caracteres.

Nomenclatura para funciones:

- Las funciones se nombrarán según el tipo de funcionalidad a la que responden, utilizando un verbo que describa el tipo de operación que realiza.

Nomenclatura para disparadores (triggers):

- Los disparadores se nombrarán utilizando el prefijo “tr” seguido del nombre de la tabla. De forma similar las funciones asociadas a los triggers se nombran utilizando el prefijo “ftr” seguido del nombre de la tabla.

Capítulo 2. Análisis de la solución propuesta

Nomenclatura para los índices:

- Los índices se nombran utilizando el prefijo “idx” seguido de guion bajo y nombre del campo.

Nomenclatura para las secuencias:

- Las secuencias se nombran siguiendo la siguiente estructura:
nombredeentidad_nombredecampo_seq.

2.5 Modelos de datos.

El modelo resultante ofreció un total de 102 tablas para el módulo de Civil y familia, de ellas 81 son tablas de datos y 21 son nomencladores, además se incluyen un conjunto de 10 tablas pertenecientes al módulo Base que se consideran relevantes para la comprensión de los procesos de Civil y familia, lo que da un total de 112 tablas.

En el modelo obtenido se identificaron un total de 15 tablas fundamentales, las cuales abarcan los principales procesos que se desarrollan en el módulo, de ellas dexpedienteprejudicial, drollojuridiccionvoluntaria, drollorecursoocasacionfg, drollorecursoocasacionfp, drollorevisioncivil, drollorecursoapelacionfm, drollorecursoapelacionfp, drolloprocesosucesorio, drolloasuntosnotariales, drollocontencioso, y drolloexpropacionforzosa heredan de la tabla dproceso perteneciente al módulo Base y los procesos drollojvotro, drollopordinariootro y drollopcsumariootro heredan a su vez del proceso dexpedienteprejudicial.

Capítulo 2. Análisis de la solución propuesta

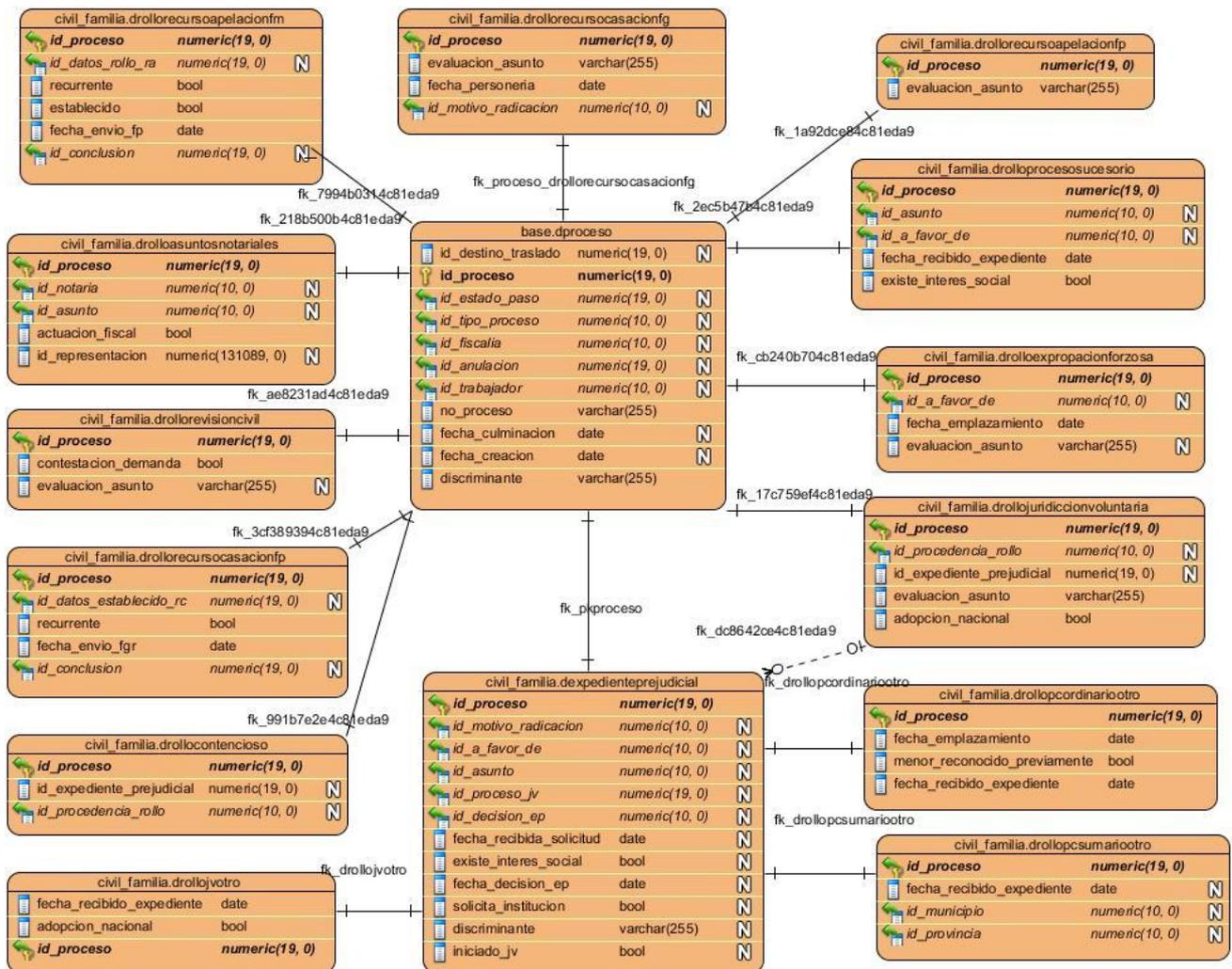


Fig. 3 Tablas principales del módulo Civil y familia.

Entidad	dproceso	
Descripción	Datos generales del proceso	
Atributos	Tipo	Descripción de los atributos
id_proceso	numeric (19)	Identificador del proceso.
id_estado_paso	numeric (19)	Identificador que asocia el estado y paso de un proceso.
id_tipo_proceso	numeric (10)	Identificador asociado al tipo de proceso.
id_fiscalia	numeric (10)	Identificador de la fiscalía asociada al proceso.
id_anulacion	numeric (19)	Identificador para asociar la anulación al proceso.
id_trabajador	numeric (10)	Identificador del trabajador asociado al proceso
no_proceso	varchar (255)	Número que tiene el proceso en la fiscalía.
fecha_culminacion	date	Fecha de culminación del proceso

Capítulo 2. Análisis de la solución propuesta

fecha_creacion	date	Fecha de creación del proceso
discriminante	varchar (255)	Para uso del ORM Doctrine 2, en el mapeo de la herencia.

Tabla 2 Entidad dproceso del módulo Base

Entidad		Dexpedienteprejudicial
Descripción		Datos generales del expediente prejudicial
Atributos	Tipo	Descripción de los atributos
id_proceso	numeric (19)	Identificador del proceso.
id_motivo_radicacion	numeric (10)	Identificador del origen del expediente prejudicial.
id_a_favor_de	numeric (10)	Identificador de la persona que el fiscal representa.
id_asunto	numeric (10)	Identificador del asunto asociado al proceso que es iniciado por un expediente prejudicial.
id_proceso_jv	numeric (19)	Identificador del proceso de jurisdicción voluntaria asociado al expediente prejudicial.
id_decision_ep	numeric (10)	Identificador de la decisión del expediente prejudicial.
fecha_recibida_solicitud	date	Fecha de recibo de la solicitud.
existe_interes_social	bool	Si existe o no interés social.
fecha_decision_ep	date	Fecha de decisión del proceso.
iniciado_jv	bool	Si el proceso contencioso surge a causa de una jurisdicción voluntaria o no.
solicita_institucion	bool	Si es o no solicitado por la institución.
discriminante	varchar(255)	Discriminante para uso del ORM Doctrine 2, en el mapeo de la herencia.

Tabla 3 Entidad dexpedienteprejudicial del módulo Civil y familia

Entidad		Drollojuridiccionvoluntaria
Descripción		Datos generales de la jurisdicción voluntaria
Atributos	Tipo	Descripción de los atributos
id_proceso	numeric (19)	Identificador del proceso.
id_procedencia_rollo	numeric (10)	Identificador de la procedencia del rollo.
id_expediente_prejudicial	numeric (19)	Identificador del expediente prejudicial asociado a una jurisdicción voluntaria.
evaluacion_asunto	varchar (255)	Comentarios del proceso cuando el tribunal envía su resolución judicial.
adopcion_nacional	bool	Si es no adoptado nacionalmente.

Tabla 4 Entidad drollojuridiccionvoluntaria del módulo Civil y familia

Entidad	Drollojvotro
----------------	--------------

Capítulo 2. Análisis de la solución propuesta

Descripción	Datos generales de la jurisdicción voluntaria cuando no inicia a través de un expediente prejudicial.	
Atributos	Tipo	Descripción de los atributos
id_expediente_prejudicial	numeric (19)	Identificador del expediente prejudicial asociado a una jurisdicción voluntaria.
fecha_recibido_expediente	date	Fecha en la que fue recibido el expediente.
adopcion_nacional	Bool	Si es o no adoptado nacionalmente.

Tabla 5 Entidad drollojvotro del módulo Civil y familia

Entidad	drollorecurso casacionfg	
Descripción	Datos generales del recurso de casación (fiscalía general)	
Atributos	Tipo	Descripción de los atributos
id_proceso	numeric (19)	Identificador del proceso.
evaluacion_asunto	varchar (255)	Comentarios del proceso cuando el tribunal envía su resolución judicial.
fecha_personeria	Date	Fecha de presentado el documento ante el tribunal.
id_motivo_radicaion	numeric (10)	Identificador del motivo de radicación del recurso.

Tabla 6 Entidad drollorecurso casacionfg del módulo Civil y familia

Entidad	drollorecurso casacionfp	
Descripción	Datos generales del recurso de casación (fiscalía provincial)	
Atributos	Tipo	Descripción de los atributos
id_proceso	numeric (19)	Identificador del proceso.
id_datos_establecido_rc	numeric (19)	Identificador de los datos adicionales en caso de que el fiscal actúe como no recurrente.
recurrente	Bool	Si el fiscal actúa o no como recurrente.
fecha_envio_fgr	Date	Fecha de envío del recurso de casación a la fiscalía general.
establecer_recurso_tpp	Bool	Si se establece no el recurso de casación ante el tribunal provincial popular.

Tabla 7 Entidad drollorecurso casacionfp del módulo Civil y familia

Entidad	Drollorevisioncivil	
Descripción	Datos generales de la revisión civil	
Atributos	Tipo	Descripción de los atributos
id_proceso	numeric (19)	Identificador del proceso.
contestacion_demanda	Bool	Si se contesta o no la demanda.
evaluacion_asunto	varchar (255)	Comentarios del proceso cuando

Capítulo 2. Análisis de la solución propuesta

		el tribunal envía su resolución judicial.
--	--	---

Tabla 8 Entidad drollorevisioncivil del módulo Civil y familia

Entidad		Drollorecursoapelacionfm
Descripción		Datos generales del recurso de apelación (fiscalía municipal)
Atributos	Tipo	Descripción de los atributos
id_proceso	numeric (19)	Identificador del proceso.
id_datos_rollo_ra	numeric (19)	Identificador de los datos adicionales cuando el fiscal actúa como no recurrente.
recurrente	Bool	Si el fiscal actúa o no como recurrente.
establecido	bool	Si el recurso de apelación está o no establecido.
fecha_envio_fp	date	Fecha de envío del recurso de apelación a la fiscalía provincial.
fecha_admision_emplazamiento	date	Fecha de admisión o emplazamiento del recurso de apelación.

Tabla 9 Entidad drollorecursoapelacionfm del módulo Civil y familia

Entidad		Drollorecursoapelacionfp
Descripción		Datos generales del recurso de apelación (fiscalía provincial)
Atributos	Tipo	Descripción de los atributos
id_proceso	numeric (19)	Identificador del proceso.
evaluacion_asunto	varchar (255)	Comentarios del proceso cuando el tribunal envía su resolución judicial.
fecha_personeria	date	Fecha de presentado el documento ante el tribunal.

Tabla 10 Entidad drollorecursoapelacionfp del módulo Civil y familia

Entidad		Drolloprocesosucesorio
Descripción		Datos generales del proceso sucesorio
Atributos	Tipo	Descripción de los atributos
id_proceso	numeric (19)	Identificador del proceso.
id_asunto	numeric (10)	Identificador del asunto asociado al proceso sucesorio.
fecha_recibido_expediente	date	Fecha de recibo del expediente.
id_a_favor_de	numeric (10)	Identificador de la persona que el fiscal representa.
existe_interes_social	bool	Si existe o no interés social.

Tabla 11 Entidad drolloprocesosucesorio del módulo Civil y familia

Entidad	drolloasuntosnotariales
----------------	-------------------------

Capítulo 2. Análisis de la solución propuesta

Descripción		Datos generales del asunto notarial.
Atributos	Tipo	Descripción de los atributos
id_proceso	numeric (19)	Identificador del proceso.
id_notaria	numeric (10)	Identificador de la notaria asociada al asunto notarial.
id_asunto	numeric (10)	Identificador del asunto.
actuacion_fiscal	bool	Si el fiscal actúa en representación de una persona o si dictamina un asunto.

Tabla 12 Entidad drolloasuntosnotariales del módulo Civil y familia

Entidad		drollocontencioso
Descripción		Datos generales del proceso contencioso.
Atributos	Tipo	Descripción de los atributos
id_proceso	numeric (19)	Identificador del proceso.
id_expediente_prejudicial	numeric (19)	Identificador del expediente prejudicial asociado proceso contencioso.
id_procedencia_rollo	numeric (10)	Identificador de la procedencia del rollo.

Tabla 13 Entidad drollocontencioso del módulo Civil y familia

Entidad		drollopctsumariootro
Descripción		Datos generales del proceso contencioso sumario.
Atributos	Tipo	Descripción de los atributos
id_proceso	numeric (19)	Identificador del proceso.
fecha_recibido_expediente	date	Fecha de recibo del expediente.

Tabla 14 Entidad drollopctsumariootro del módulo Civil y familia

Entidad		drollopccordinariootro
Descripción		Datos generales del proceso contencioso ordinario.
Atributos	Tipo	Descripción de los atributos
id_expediente_prejudicial	numeric (19)	Identificador del expediente prejudicial asociado proceso contencioso ordinario.
fecha_emplazamiento	date	Fecha de emplazamiento
menor_reconocido_previamente	bool	Si el menor fue o no reconocido previamente.
fecha_recibido_expediente	date	Fecha de recibo del expediente.

Tabla 15 Entidad drollopccordinariootro del módulo Civil y familia

Entidad		drolloexpropiacionforzosa
Descripción		Datos generales del asunto notarial.
Atributos	Tipo	Descripción de los atributos
id_proceso	numeric (19)	Identificador del proceso.

id_a_favor_de	numeric (10)	Identificador de la persona que el fiscal representa.
fecha_emplazamiento	Date	Fecha del emplazamiento.
evaluación_asunto	varchar (255)	Comentarios del proceso cuando el tribunal envía su resolución judicial.

Tabla 16 Entidad drolloexpriacionforzosa del módulo Civil y familia

2.5.1 Patrones utilizados en el diseño.

Llave subrogada: Es uno de los patrones más usados en el diseño de base datos pues facilita la interacción con la misma. Está presente en el modelo resultante para generar una llave primaria única en cada una de las tablas, la misma se crea de tipo numérico (numeric), de tamaño 19 para las tablas de datos y 10 para los nomencladores. Esto permite que las tablas sean más fáciles de consultar a partir del identificador, pues todos tienen el mismo tipo en cada una de las tablas. Un ejemplo se observa en la tabla nnotaria del módulo Civil y familia.

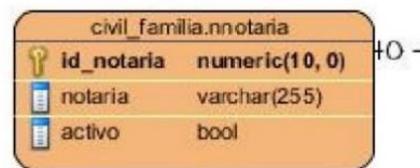


Fig. 4 Entidad nnotaria del módulo Civil y familia.

2.6 Requisitos transaccionales.

Como parte del desarrollo del módulo Civil y familia es necesario realizar una serie de consultas a la Base de Datos que permitan el acceso a la información contenida. A continuación se listan los requisitos transaccionales de dicho módulo:

Requisitos	Descripción
Listar expedientes prejudiciales	Mostrar una lista con los expedientes prejudiciales adicionados.
Buscar expediente prejudicial	Hacer una búsqueda de los expedientes prejudiciales que se inician que se han creado.
Buscar rollo de Jurisdicción Voluntaria	Buscar los rollos adicionados al sistema a partir de varios criterios de búsqueda.
Listar rollos de Jurisdicción Voluntaria	Mostrar una lista con los rollos de Jurisdicción Voluntaria adicionados.
Buscar rollo de Asuntos	Buscar uno o varios rollos de Asuntos Notariales

Capítulo 2. Análisis de la solución propuesta

Notariales	adicionados al sistema a partir de varios criterios de búsqueda.
Listar rollos de Asuntos Notariales	Mostrar una lista con los rollos de Asuntos Notariales adicionados.
Buscar rollo de Expropiación Forzosa	Hacer una búsqueda de los rollos adicionados al sistema de Expropiación Forzosa a partir de varios criterios.
Listar rollos de Expropiación Forzosa	Mostrar una lista con los rollos de Expropiación Forzosa adicionados.
Buscar rollo de Recurso de Apelación en la Fiscalía Municipal	Realizar una búsqueda de los rollos de apelación que se han insertado en Fiscalía Municipal.
Buscar rollo de Recurso de Apelación en la Fiscalía Provincial	Realizar una búsqueda de los rollos de Apelación que se han insertado en la Fiscalía Provincial.
Listar rollos de Recurso de Apelación y Recurso Casación	Mostrar una lista con los rollos de los recursos adicionados.
Listar rollos en tiempo de Recurso de Apelación	Mostrar una lista con los rollos que salieron de los procesos ordinarios y están en tiempo para que se les realice un Recurso de Apelación.
Listar información para personería en el Recurso de Apelación	Mostrar una lista con la información para personería que llega a la Fiscalía Provincial de los recursos establecidos en la Fiscalía Municipal.
Buscar rollo del Recurso de Casación en la Fiscalía Provincial	Realizar una búsqueda de los rollos de Recurso Casación que se han insertado en Fiscalía Provincial.
Buscar rollo del Recurso de Casación en la Fiscalía General de la República	Realizar una búsqueda de los rollos de Recurso de Casación que se han insertado en la Fiscalía General de la República. Permite al fiscal buscar los rollos del recurso concluidos en la instancia inferior.
Listar rollos de procesos en tiempo para casar sentencia	Mostrar una lista con los rollos de los procesos ordinarios y de apelación que ya finalizaron y están en tiempo de casar sentencia.

Capítulo 2. Análisis de la solución propuesta

Listar información para personería en Recurso Casación	Mostrar una lista con la información para personería que se envía desde la Fiscalía Provincial hasta la Fiscalía General de la Republica.
Listar motivos de Recurso de Casación	Mostrar una lista con los motivos del Recurso de Casación adicionados.
Buscar sentencia origen	Buscar la sentencia que da origen al proceso de Proceso de Revisión Civil.
Buscar rollo de Proceso de Revisión Civil	Permite buscar un rollo de Proceso de Revisión Civil.
Listar rollos de Proceso de Revisión Civil	Mostrar una lista con los rollos de Proceso de Revisión Civil adicionados.
Listar actas de entrevistas.	Listar las actas de entrevistas adicionadas a un rollo determinado.
Generar reporte.	Generar un reporte a partir de los parámetros seleccionados.
Generar reportes de procesos contenciosos y otros procedimientos en los que interviene el fiscal en la Fiscalía Municipal	Generar reportes de Procesos Contenciosos y los procedimientos de la instrucción 216 en los que interviene el Fiscal en la Fiscalía Municipal.
Generar reportes de procesos contenciosos y otros procedimientos en los que interviene el fiscal en la Fiscalía Provincial	Generar reportes de Procesos Contenciosos y los procedimientos de la instrucción 216 en los que interviene el Fiscal en la Fiscalía Provincial.
Generar reportes de asuntos de Jurisdicción Voluntaria con participación del fiscal	Generar reportes de Asuntos de Jurisdicción Voluntaria con participación del Fiscal, tanto promoviendo como dictaminando un asunto.
Generar reportes de tutela	Generar reportes de tutela.
Generar reporte de Asuntos Notariales	Generar reportes de Asuntos Notariales, tanto representando, en la firma de documentos notariales, como dictaminando Asuntos Notariales.
Generar reporte de expedientes	Generar reportes de expedientes de patria

de patria potestad	potestad.
Generar reportes de Recursos de Apelación en los que participó el fiscal	Generar reportes de RA en los que participó el fiscal, tanto en su actuación recurrente como no recurrente.
Generar reporte resumen de procesos contenciosos y otros procedimientos en los que interviene el fiscal	Generar un reporte resumen de los procesos contenciosos y otros procedimientos en los que interviene el fiscal.
Generar reporte resumen de Jurisdicción Voluntaria y Alimentos	Generar un resumen de los procesos de Jurisdicción Voluntaria y Alimentos que se han llevado a cabo en cada Fiscalía Municipal y Fiscalía Provincial.
Generar reporte resumen de tutela	Generar un reporte resumen del asunto tutela que se ha llevado a cabo en cada Fiscalía Municipal.
Generar reporte resumen de Asuntos Notariales	Generar un resumen de los Asuntos Notariales que se han llevado a cabo en las Fiscalías Municipales.
Generar reporte resumen de expedientes de patria potestad	Generar un reporte resumen de los expedientes de patria potestad que se han creado en las Fiscalía Municipal y Fiscalía Provincial.
Generar reporte resumen del total de Recursos de Apelación	Generar un reporte resumen de los Recursos de Apelación en los que ha participado el fiscal.
Generar reporte resumen del total de Recursos de Casación en los que participó el fiscal	Generar un reporte resumen de los Recursos de Casación en los que ha participado el fiscal.
Generar reporte del total de Procesos de Revisión Civil	Generar un reporte de los Procesos de Revisión Civil en los que ha participado el fiscal.

Tabla 17 Requisitos transaccionales del módulo Civil y familia.

2.7 Acceso a datos.

Para el acceso a datos se utiliza el ORM Doctrine 2, el cual proporciona herramientas para simplificar el acceso y manejo de la información, lo que facilita a los desarrolladores una mejor gestión y entendimiento. Para realizar dicho acceso Doctrine utiliza su propio lenguaje de consulta denominado Lenguaje de Consulta de Doctrine (DQL del inglés Doctrine Query Language). DQL permite convertir datos entre el sistema de tipos utilizado en un lenguaje de programación

Capítulo 2. Análisis de la solución propuesta

orientado a objetos y el utilizado en una base de datos relacional, es decir, las tablas de la base de datos pasan a ser clases y los registros objetos que podemos manejar con facilidad.

```
class RolloExpropiacionForzosa extends Proceso {  
  
    /**  
     * @ORM\Column(name="fecha_emplazamiento", type="date", nullable = false)  
     * @Assert\NotNull()  
     * @Assert\Date()  
     */  
    private $fecha_emplazamiento;  
  
    /**  
     *  
     */  
    private $evaluacion_asunto;  
  
    /**  
     *  
     */  
    private $acta_entrevistas;  
  
    /**  
     *  
     */  
    private $a_favor_de;  
  
    /**  
     *  
     */  
    protected $notificaciones_tribunal;  
}
```

Fig. 5 Mapeo objeto relacional de entidad RolloExpropiacionForzosa del módulo Civil y familia.

DQL también posibilita escribir las consultas de base de datos en un lenguaje SQL propio orientado a objetos como las que se muestran a continuación.

```
public function listarRolloExpropiacionForzosaPaso($id_paso_actual, $lista_paso_anteriores, $trabajador_fiscalia) {  
    $qb = $this->getEntityManager()->createQueryBuilder();  
    $qb->select('pq', 'pp', 'e', 'pr', 'tp', 'ep', 'ee', 'ps')  
        ->from('BaseBundle:CivilFamilia\RolloExpropiacionForzosa', 'pq')  
        ->leftJoin('pq.personas_proceso', 'pp')  
        ->leftJoin('pq.expedientes', 'e')  
        ->leftJoin('pp.persona', 'pr')  
        ->leftJoin('pp.tipo_persona', 'tp')  
        ->leftJoin('pq.trabajador_fiscalia', 'tf')  
        ->innerJoin('pq.estado', 'ep')  
        ->innerJoin('ep.estado', 'ee')  
        ->innerJoin('ep.paso', 'ps')  
        ->andWhere($qb->expr()->eq('ps.id_paso', ':prmPaso'))  
        ->andWhere($qb->expr()->orX('ee.id_estado =:prmEstadoPendiente', 'ee.id_estado =:prmEstadoEdicion'))  
        ->andWhere($qb->expr()->eq('tf.id_trabajador', ':prmTrabajadorFiscalia'))  
        ->setParameter('prmEstadoPendiente', ConfigUtilCivilFamilia::estado_pendiente)  
        ->setParameter('prmEstadoEdicion', ConfigUtilCivilFamilia::estado_edicion)  
        ->setParameter('prmPaso', $id_paso_actual)  
        ->setParameter('prmTrabajadorFiscalia', $trabajador_fiscalia->getIdTrabajador());  
    if (!is_null($lista_paso_anteriores)) {  
        $qb->orWhere($qb->expr()->andX('ee.id_estado =:prmEstadoFinalizado', $qb->expr()->in('ps.id_paso', $lista_paso_anteriores))  
            ->setParameter('prmEstadoFinalizado', ConfigUtilCivilFamilia::estado_finalizado);  
    }  
    $result = $qb->getQuery()->getResult();  
    return $result;  
}
```

Fig. 6 Función listarRolloExpropiacionForzosaPaso

2.8 Reportes.

Dada la complejidad en la obtención de informaciones estadísticas, se contempla dentro de los objetivos que persigue el desarrollo del sistema SIGEF II el de: “*ser una fuente de estadísticas para las más altas instancias del país facilitando la toma de decisiones a los diferentes niveles del Estado Cubano*”. (Figueroa Machado, et al., 2013)

Para lograr dicho objetivo se realiza la implementación de reportes como mecanismos que permitan proveer a los usuarios del sistema de una interfaz adecuada para la obtención de dichas informaciones.

Los reportes son una manera excelente de organizar y presentar los datos de una base de datos, estos recopilan los datos de las tablas o consultas para permitir su impresión y análisis, dando lugar a un diseño atractivo e informativo que les da mayor utilidad para los usuarios.

A continuación se presenta un ejemplo de uno de los reportes del módulo Civil y familia.



Reporte 5: CIVIL Y FAMILIA

Órgano Provincial Emisor: _____

Periodo que se informa: _____

Tabla 5.2: Recursos de Casación Civil y Familia

No.	INDICADOR	PERIODO	DIF AÑO ANT	ACUM	DIF AÑO ANT
1	Total de recursos de Casación en los que participó el Fiscal				
2	Cantidad de recursos de Casación en los que participó el Fiscal como recurrente				
3	De ellos cantidad declarados con lugar				
4	De ellos cantidad declarados sin lugar				
5	De ellos cantidad declarados con lugar en parte				
6	De ellos inadmisibles				
7	Cantidad de recursos de Casación en los que participó el Fiscal como no recurrente				
8	De ellos cantidad declarados con lugar				
9	De ellos cantidad declarados sin lugar				
10	De ellos cantidad declarados con lugar en parte				
11	De ellos inadmisibles				

Fig. 7 Reporte estadístico provincial Recursos de Casación del módulo Civil y familia.

Para la obtención de dichos reportes se utilizan funciones almacenadas, realizándose una por cada indicador, estas reciben un rango de fecha y devuelven los siguientes parámetros: **PERIODO** en el que se calculan la cantidad de procesos que ocurren en el rango de fecha que recibe la función, **DIF AÑO ANT** asociado al periodo que compara la cantidad de procesos del periodo en el año anterior, **ACUM** que son los procesos que ocurren desde el 1ro de enero hasta la fecha fin y **DIF AÑO ANT** asociado al acumulado que compara con el acumulado del año anterior.

2.9 Optimización

Es necesario realizar la optimización con el objetivo de reducir las demoras ante operaciones que involucren un gran volumen de datos. Existen varias estrategias para la optimización de una base de datos ya sea por el tipo específico de gestor y sus configuraciones, por el modelo de datos seleccionado, por configuraciones sobre la base de datos, o por las optimizaciones en las consultas. Muchas veces es aconsejable aplicar varias alternativas juntas. Algunas de estas

técnicas fueron utilizadas en el diseño e implementación de la base de datos desarrollada y a continuación se describen.

2.9.1 Normalización y denormalización del modelo.

Para el desarrollo del proyecto SIGEF2 se han definido un conjunto de acciones a realizar con el objetivo de mantener un nivel de adecuado de consistencia en la información evitando redundancia y protegiendo la consistencia de los datos, además que la base de datos funcione eficientemente propiciando respuestas rápidas ante los requerimientos del cliente. Estas acciones consisten en llevar el diseño de la base de datos a tercera forma normal, identificar las consultas más frecuentes y de ellas aquellas cuyo plan de solución sean engorrosos con demasiados inner join etc, aplicar patrón de denormalización al diseño obtenido en la acción 1 de forma adecuada para garantizar que solo las consultas identificadas en la acción 2 sean adecuadamente tratadas y aumente la eficiencia en su ejecución. (Pupo Acosta, 2013)

La Base de Datos de la solución propuesta se encuentra en 1ra FN ya que todos los atributos son atómicos, o sea no contiene atributos multivaluados. Además se puede asegurar que está en 2da FN ya que se encuentra en 1ra FN, todos los campos no clave de cada una de las tablas dependen de la clave primaria de dicha tabla y cada clave primaria es única. También se puede decir que está en 3ra FN ya que además de estar en 2da FN, podemos afirmar que los atributos no primos no tienen dependencias transitivas.

Algunas tablas se denormalizaron con el objetivo de optimizar las consultas sobre las mismas y mejorar el acceso a sus datos. Un ejemplo de denormalización se puede observar en la tabla drolloexpropiacionforzosa.



civil_familia.drolloexpropiacionforzosa	
id_proceso	numeric(19, 0)
id_a_favor_de	numeric(10, 0)
fecha_emplazamiento	date
evaluacion_asunto	varchar(255) N

Fig. 8 Entidad drolloexpropiacionforzosa del módulo Civil y familia

2.9.2 Estrategia de indexado utilizada.

Es importante construir los índices necesarios para garantizar elevados niveles de eficiencia en los resultados de las consultas, para ello se definió utilizar la estrategia de indexado estándar del gestor, árbol-B; ya que es

una estructura eficiente y flexible que se va reorganizando poco a poco y por lo tanto no requiere periódicas reorganizaciones masivas, además ofrece gran variedad de operadores, y puede ser utilizada sobre los campos más recorridos en las consultas, 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, con esto se consume menos tiempo en hallar el resultado y es menor la cantidad de veces que se accede al disco para leer.

Todas las llaves primarias poseen índices de tipo árbol-B lo que implica que cualquier búsqueda que se realice utilizando las llaves se optimizará mediante este método. Además se identificaron un conjunto de atributos recurrentes en las búsquedas los cuales fueron indexados para lograr optimizar dichas operaciones sobre la base de datos.

A continuación se presenta un ejemplo de dichos índices:

```
CREATE INDEX idx_salario
ON civil_familia.ddatostutor
USING btree
(salario );
```

Fig. 9 Índice sobre el atributo salario de la tabla ddatostutor del módulo Civil y familia.

2.10 Conclusiones parciales.

En este capítulo se documentó el desarrollo de la solución propuesta para el módulo Civil y familia de la aplicación SIGEF II. Además se obtuvo el modelo de datos como artefacto fundamental del diseño, a partir de la aplicación de la nomenclatura y la arquitectura de datos. También se explican las técnicas utilizadas para la optimización del resultado obtenido, las cuales contribuyeron al mejoramiento de la solución y la eliminación de problemas de diseño.

Capítulo 3. Validación y pruebas.

3.1 Introducción.

En el siguiente capítulo se aborda el proceso de validación y pruebas realizado a la base de datos obtenida, haciendo uso de técnicas de validación teórica para determinar aspectos importantes como la consistencia y la integridad de los datos, además de la realización de pruebas de carga, rendimiento, estrés y volumen con las cuales se valida funcionalmente la solución desarrollada.

3.2 Validación teórica.

Para garantizar el correcto funcionamiento de la base de datos es necesario velar por aspectos importantes como la integridad y la consistencia de los datos, para ello se hace uso de técnicas que regulan dichos parámetros con el objetivo de alcanzar el nivel de calidad deseado.

3.2.1 Integridad de la base de datos.

Con el objetivo de garantizar la integridad y consistencia de los datos se aplicaron las siguientes reglas:

Datos requeridos: Se definieron un conjunto de atributos no nulos (notnull), dada su importancia a la hora de insertar datos. Un ejemplo se observa en los campos `fecha_recibido_expediente` y `adopcion_nacional` de la tabla `drollojvotro`.

civil_familia.drollojvotro	
fecha_recibido_expediente	date
adopcion_nacional	bool
id_proceso	numeric(19, 0)

Fig. 10 Tabla `drollojvotro` del módulo Civil y familia.

Chequeo de validez: Se definieron restricciones sobre los tipos de datos utilizados en las columnas, con el objetivo de chequear que los valores agregados cumplan con el dominio definido previamente para cada una.

Dominio	Tipo de Datos	Campos donde puede aparecer
id_tabla	numeric(19)	Para identificadores de tablas de datos.
id_nomenclador	numeric(10)	Para identificadores de tablas nomencladoras.
descripción	varchar(255)	descripcion

Fecha	date	fecha_presentacion, fecha_notificacion, fecha_emplazamiento
Bool	boolean	recurrente, activo
comentario	texto	evaluacion_asunto

Tabla 18 Descripción de integridad de dominio

Integridad de entidad: las llaves primarias (Primary Key) de cada entidad son atributos no nulos y únicos.

Integridad referencial: las llaves foráneas (Foreign Key) agregadas en una tabla tomaran valores referentes y en concordancia de donde derivan.

3.2.2 Métricas.

Las métricas son un buen medio para entender, monitorizar, controlar, predecir y probar el desarrollo de software, aplicarlas al entorno de las bases de datos es importante debido a la relevancia que tanto el tamaño de los datos como su naturaleza tienen en aspectos como el esfuerzo del desarrollo. (Calero, y otros, 2000)

Uno de los indicadores más importantes para medir la calidad de una base de datos relacional es la teoría de la normalización, a partir de la cual el científico norteamericano Jim Gray propone en 1991 un ratio de normalidad (NR⁷) que se define como el número de tablas en tercera forma normal o superior dividido entre el número de tablas en el esquema:

$$NR = \frac{NT3FN}{NTS}$$

Dónde: NT3NF es el número de tablas en 3ra forma normal (o superior) y NTS es el número de tablas en el esquema.

En el caso específico de la base de datos desarrollada para el módulo Civil y familia del Sistema de Gestión de las Fiscalías el cálculo de dicha métrica queda del siguiente modo:

⁷ Del inglés Normality Ratio

$NR = \frac{NT3FN}{NTS} = \frac{89}{102} = 0.87$, lo que refleja un elevado nivel de normalización.

3.3 Validación Funcional.

Es necesario medir el rendimiento funcional del sistema teniendo en cuenta el modelo obtenido, el gestor utilizado y las características de hardware, para ello se realizan un conjunto de pruebas que permiten verificar el funcionamiento del sistema ante determinados escenarios.

3.3.1 Pruebas de rendimiento.

Las pruebas de rendimiento se centran en determinar la velocidad con la que el sistema realiza una tarea bajo determinadas condiciones, con el objetivo anticipar los problemas que puedan ocurrir una vez la aplicación esté en producción. Este tipo de pruebas son ejecutadas por medio de scripts automatizados, los cuales se encargan de emular las acciones que realizaría un usuario final sobre la aplicación. Durante la ejecución de dichas pruebas es necesario monitorizar el sistema por medio de indicadores de rendimiento.

Para las pruebas realizadas a la base de datos del sistema se define como tiempo de respuesta aceptable, menor a 3 segundos; un tiempo mayor será considerado como no aceptable.

La prueba consistió en realizar una consulta, bajo dos tipos de situaciones, y medir su tiempo de ejecución mediante el comando EXPLAIN ANALYZE, que devuelve los resultados en milisegundos (ms). La primera situación es realizar la consulta sobre campos sin indexar, y la segunda con campos indexados.

Consulta:

```

EXPLAIN ANALYZE
SELECT
  *
FROM
  base.dproceso,
  civil_familia.dexpedienteprejudicial,
  civil_familia.drollojvotro
WHERE
  dexpedienteprejudicial.id_proceso = dproceso.id_proceso AND
  dexpedienteprejudicial.id_proceso = drollojvotro.id_proceso AND
  drollojvotro.adopcion_nacional = TRUE AND
  drollojvotro.fecha_recibido_expediente BETWEEN '2010-01-01' And '2010-12-31'
ORDER BY
  drollojvotro.fecha_recibido_expediente ASC;

```

Fig. 11 Consulta a realizar como prueba

Resultados:

Ambiente sin indexado. Tiempo: 0.932 ms.

Data Output	Explain	Messages	History
	QUERY PLAN		
	text		
1	Sort (cost=11.27..11.28 rows=4 width=393) (actual time=0.665..0.667 rows=9 loops=1)		
2	Sort Key: drollojvotro.fecha_recibido_expediente		
3	Sort Method: quicksort Memory: 24kB		
4	-> Nested Loop (cost=2.59..11.23 rows=4 width=393) (actual time=0.221..0.559 rows=9 loops=1)		
5	-> Hash Join (cost=2.59..7.03 rows=7 width=190) (actual time=0.198..0.387 rows=9 loops=1)		
6	Hash Cond: (dexpedienteprejudicial.id_proceso = drollojvotro.id_proceso)		
7	-> Seq Scan on dexpedienteprejudicial (cost=0.00..4.00 rows=100 width=180) (actual time=0.014..0.054 rows=100 loops=1)		
8	-> Hash (cost=2.50..2.50 rows=7 width=10) (actual time=0.138..0.138 rows=9 loops=1)		
9	Buckets: 1024 Batches: 1 Memory Usage: 1kB		
10	-> Seq Scan on drollojvotro (cost=0.00..2.50 rows=7 width=10) (actual time=0.020..0.114 rows=9 loops=1)		
11	Filter: (adopcion_nacional AND (fecha_recibido_expediente >= '2010-01-01'::date) AND (fecha_recibido_expediente <= '2010-12-31'::date))		
12	Rows Removed by Filter: 91		
13	-> Index Scan using dproceso_pkey on dproceso (cost=0.00..0.59 rows=1 width=203) (actual time=0.010..0.012 rows=1 loops=1)		
14	Index Cond: (id_proceso = dexpedienteprejudicial.id_proceso)		
15	Total runtime: 0.932 ms		

Fig. 12 Resultado de consulta sin indexado

Ambiente con campos de llaves primarias y llaves foráneas indexadas. Tiempo: 0.590 ms.

Output pane	
Data Output	Explain Messages History
	QUERY PLAN text
1	Sort (cost=11.33..11.35 rows=7 width=393) (actual time=0.366..0.367 rows=9 loops=1)
2	Sort Key: drollojvotro.fecha recibido expediente
3	Sort Method: quicksort Memory: 24kB
4	-> Nested Loop (cost=2.59..11.23 rows=7 width=393) (actual time=0.111..0.293 rows=9 loops=1)
5	-> Hash Join (cost=2.59..7.03 rows=7 width=190) (actual time=0.091..0.177 rows=9 loops=1)
6	Hash Cond: (dexpedienteprejudicial.id proceso = drollojvotro.id proceso)
7	-> Seq Scan on dexpedienteprejudicial (cost=0.00..4.00 rows=100 width=180) (actual time=0.011..0.039 rows=100)
8	-> Hash (cost=2.50..2.50 rows=7 width=10) (actual time=0.053..0.053 rows=9 loops=1)
9	Buckets: 1024 Batches: 1 Memory Usage: 1kB
10	-> Seq Scan on drollojvotro (cost=0.00..2.50 rows=7 width=10) (actual time=0.013..0.042 rows=9 loops=1)
11	Filter: (adopcion nacional AND (fecha recibido expediente >= '2010-01-01'::date) AND (fecha recibido expediente <= '2010-01-01'::date))
12	Rows Removed by Filter: 91
13	-> Index Scan using dproceso pkey on dproceso (cost=0.00..0.59 rows=1 width=203) (actual time=0.007..0.008 rows=1 loops=1)
14	Index Cond: (id proceso = dexpedienteprejudicial.id proceso)
15	Total runtime: 0.590 ms

Fig. 13 Resultado de consulta con indexado

Las pruebas realizadas demuestran que la base de datos responde satisfactoriamente, obteniéndose tiempos de respuestas menores a los 3 segundos para ambos ambientes.

3.3.2 Pruebas de volumen.

Las pruebas de volumen se realizan con el objetivo de verificar que la aplicación funciona adecuadamente en escenarios críticos, definidos como el máximo de datos que tendría que soportar la base de datos en un escenario real bajo condiciones de hardware mínimas.

Los elementos de hardware del entorno de pruebas se encuentran por debajo del entorno real de despliegue:

Indicadores	Entorno de despliegue	Entorno de pruebas
Microprocesador	Quad-Core Intel(R) a 2.6 GHz	Pentium (R) Dual-Core a 1.6 GHz
RAM	2 GB	1 GB
HDD	1TB	160 GB

Tabla 19 Descripción de los principales elementos de hardware de los entornos de prueba y despliegue

En el caso específico del módulo Civil y familia dichas pruebas se realizaron sobre las tablas dexpedienteprejudicial, drollojurisdiccionvoluntaria y drollojvotro ya que se encuentran entre los procesos más robustos en cuanto a inserción de datos, para ellas estiman un máximo de 1000 procesos anuales y se define un período de explotación de 10 años, por lo que se decide insertar un total de 10000 tuplas para cada una.

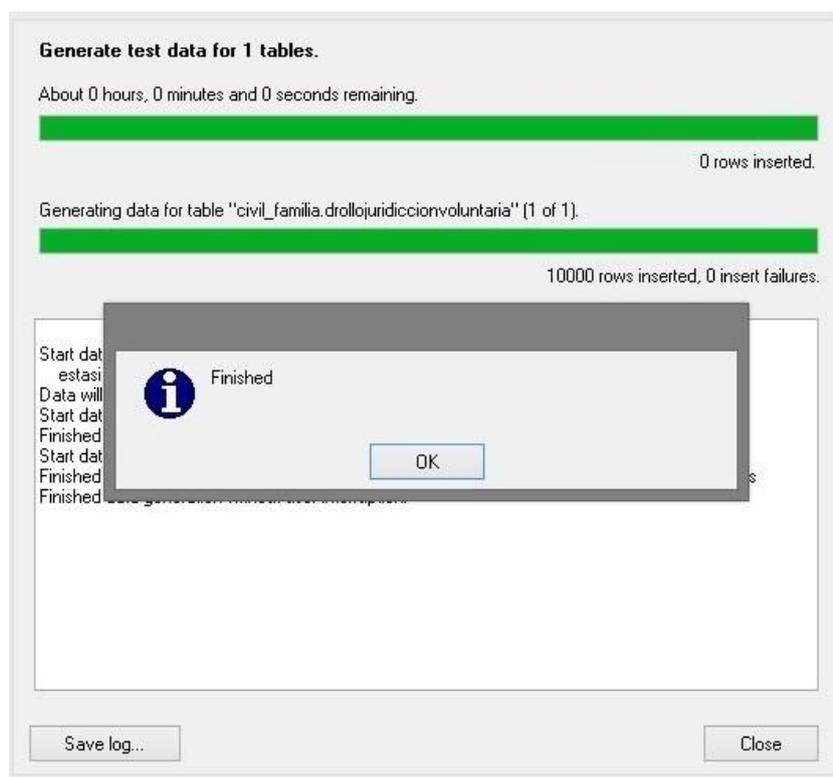


Fig. 14 Prueba de volumen sobre la tabla drollojurisdiccionvoluntaria del módulo Civil y familia.

El resultado de la prueba arroja, que la base datos puede soportar la carga de datos que se genera en las fiscalías del país, sin presentar límites de capacidad, desbordamiento de columnas, atributos o tipos de datos.

3.3.3 Pruebas de carga y estrés.

Las pruebas de carga se hacen generalmente para ver el comportamiento de la aplicación bajo una carga esperada, estas miden tiempos de respuesta, índices de procesamiento y otros requisitos sensibles al tiempo y dependen de elementos tales como el aumento de carga esperado, la carga continuada por un periodo prolongado de tiempo, la resistencia a las caídas o las pruebas de

estrés. Las pruebas de estrés se hacen para observar el punto de ruptura del sistema, escalando la cantidad de carga hasta que se encuentran los límites del mismo.

En las fiscalías de todo el país la cantidad de trabajadores es diferenciada dependiendo su tipo (Machado, 2012).

Tipo de Fiscalía	Cantidad de trabajadores
Municipal	3-15
Provincial	15-25
Fiscalía General de la República	100

Tabla 20 Cantidad de trabajadores por tipo de fiscalía

Las pruebas estarán centradas en cubrir el máximo de trabajadores que concurrentemente puedan estar realizando peticiones a la base de datos desde la aplicación. Tomándose entonces la cantidad posible en la Fiscalía General de la República como caso más crítico, por el tipo de entidad donde puede acumularse mayor carga de trabajo.

A continuación se muestran las propiedades de los hilos de la prueba realizada, así como el significado y valor definido para cada uno de ellos.

- **Número de hilos:** Número de usuarios a simular.
- **Periodo de subida** (en segundos): Tiempo que debiera llevarle a JMeter lanzar todos los hilos (si se seleccionan 10 hilos y el periodo 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 el test.

Se define para la base de datos del módulo Civil y familia:

- Numero de hilos: 100
- Periodo de subida: 0
- Contador de bucle: 100

Resultado:

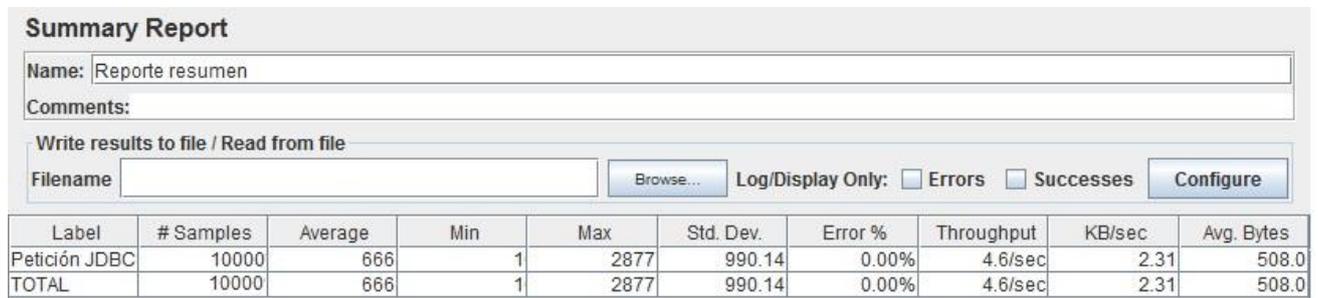


Fig. 15 Reporte resumen de prueba de carga y estrés sobre la base de datos Civil y familia.

Los resultados demuestran que la base de datos puede atender cada una de las peticiones realizadas en 0.666 segundos, y en caso crítico no superaría los 2.877 segundos, tiempos que a su vez son considerados aceptables.

3.4 Conclusiones parciales.

Las pruebas realizadas demostraron que la base de datos responderá de manera satisfactoria ante su explotación en las Fiscalías de todo el país:

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

Conclusiones Generales

A partir del desarrollo de este trabajo se cumplió el objetivo de realizar el diseño y la implementación de una BD para el módulo Civil y familia, contribuyendo a la integridad de los datos, al concluir lo siguiente:

El estudio de los elementos teóricos relacionados con el objeto de estudio sirvió de plataforma para llegar a un correcto diseño de base de datos.

Con el diseño de base de datos y la aplicación de patrones se obtuvo el modelo de datos como artefacto fundamental.

Las estrategias de optimización aplicadas, contribuyeron al mejoramiento de la solución y la eliminación de problemas de diseño.

Las pruebas realizadas, corroboraron la factibilidad del resultado obtenido, demostrando que la base datos responderá de manera satisfactoria ante el entorno real de explotación.

Recomendaciones

Proporcionar mantenimiento y soporte regular a la base de datos con el objetivo de garantizar su funcionamiento óptimo.

Bibliografía

García Chávez , Carlos Alberto. [En línea] [Citado el: 13 de 05 de 2014.] <http://www.mailxmail.com/curso-diseno-base-datos-relacionales/diseno-fisico-bases-datos>.

Acosta, Jose Carlos Pupo. 2013. *0120_4 Arquitectura Vista de Datos*. s.l. : Universidad de las Ciencias Informáticas, 2013.

2014. Agile Data. [En línea] Ambyssoft Inc, 2014. [Citado el: 24 de febrero de 2014.] <http://www.agiledata.org/>.

Apache Software Foundation. 2014. Apache JMeter. [En línea] Apache Software Foundation, 2014. [Citado el: 24 de febrero de 2014.] <http://jmeter.apache.org/>.

Baizán, María Covadonga Fernández. 1987. *El modelo racional de datos: De los fundamentos a los modelos deductivos*. s.l. : Ediciones Díaz de Santos, 1987.

Batini, Carlos. 2004. *Diseño conceptual de bases de datos*. 2004.

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

Blanco, Ing. Héctor Fuentes. 2012. *Documento Arquitectura de Software SIGEF II*. 2012.

Buyens, Jim. 2000. *Aprenda Desarrollo de bases de datos web*. 2000. ISBN: 84-48 1-2903-2.

Calero, Coral, y otros. 2000. *Métricas para la Evaluación de la Complejidad de Bases de Datos Relacionales*. España : s.n., 2000.

Cray, Larman. 1999. *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. 1999. 970-17-0261-1.

Datanamic. 2014. Datanamic. *Datanamic*. [En línea] 2014. [Citado el: 02 de 05 de 2014.] <http://www.datanamic.com/>.

Date, Christopher J. 2003. *Sistemas de Bases de Datos*. La Habana : Felix Varela, 2003.

Doctrine. 2013. Doctrine Project. [En línea] 2013. [Citado el: 9 de 12 de 2013.] <http://docs.doctrine-project.org/projects/doctrine-orm/en/latest/reference/transactions-and-concurrency.html>.

Dunning , David. ehow. [En línea] http://www.ehowenespanol.com/tres-fases-del-diseno-tradicional-bases-datos-info_197446/.

Elsuari R., Navathe. 1993. *Sistemas de Bases de Datos, Conceptos fundamentales*. s.l. : Addison-Wesley, 1993.

Erl, Thomas. 2009. Revista Universidad EAFIT. [En línea] 2009. [Citado el: 27 de enero de 2014.] <http://publicaciones.eafit.edu.co/index.php/revista-universidad-eafit/article/view/8>.

Fabien Potencier, Ryan Weaver. *Symfony 2.3, el libro oficial.*

Figuroa Machado, Yenier, del Rosario Lalcebo, Yanelys y Fuentes Blanco, Hector. 2013. *IMPACTO DE LA INFORMATIZACIÓN DE LA FISCALÍA GENERAL DE LA REPÚBLICA EN LA SOCIEDAD CUBANA.* La Habana : s.n., 2013.

Group, The PostgreSQL Global Development. 2014. PostgreSQL 9.1.13 Documentación. [En línea] 2014. [Citado el: 02 de 04 de 2014.] <http://www.postgresql.org/docs/9.1/static/routine-vacuuming.html>.

Hansen, Gary W y Hansen, James V. 1997. *Diseño y Administración de Bases de Datos.* s.l. : Prentice Hall, 1997. ISBN 84-8322-002-4.

Hernández Suarez, Eivys, y otros. *REPLICADOR DE DATOS REKO: HERRAMIENTA DE CÓDIGO ABIERTO PARA SISTEMAS DISTRIBUIDOS.* La Habana : s.n.

Kroenke, David M. 2003. *Procesamiento de bases de datos: fundamentos, diseño e implementación Octava Edición.* 2003.

Machado, Ing. Yenier Figuroa. 2012. *Proyecto Tecnico SIGEF II, CEGEL.* 2012.

Marqués, M. 2009. *Bases de Datos.* Valencia, España : s.n., 2009.

Marqués, Mercedes. 2001. *Apuntes de Ficheros y Bases de Datos.* 2001.

Mato García, Rosa María. 2005. *Sistemas de Base de Datos.* 2005.

Mendelzon - Ale. 2000. *Introducción a las Bases de Datos Relacionales.* Buenos Aires : s.n., 2000. ISBN: 987-97892-2-9.

NetBeans Community. 2014. NetBeans. [En línea] 2014. [Citado el: 24 de febrero de 2014.] https://netbeans.org/index_es.html.

Oracle. 2014. Oracle. [En línea] 2014. [Citado el: 28 de febrero de 2014.] http://docs.oracle.com/cd/E24842_01/html/E23286/rbac-1.html.

Osorio Rodriguez, Alain. *MODELO DE DESARROLLO DE BASE DE DATOS v2.0.* La Habana : s.n.

Osorio, Alain and Martínez, Mairelys. 2012. *Guía Práctica para Arquitecturas de Datos Empresariales.* 2012.

Osorio, Alain. 2012. *Dbplanning framework.* 2012.

pgAdmin. 2013. pgAdmin PostgreSQL tools. *pgAdmin PostgreSQL tools.* [En línea] 2013. [Citado el: 14 de 12 de 2013.] <http://www.pgadmin.org/>.

PostgreSQL. 2013. PostgreSQL. *PostgreSQL.* [En línea] 2013. [Citado el: 14 de 12 de 2013.] http://www.postgresql.org.es/sobre_postgresql.

Pressman. 2005. *Aprendiendo UML en 24 horas.* 2005.

Pupo Acosta, Jose Carlos. 2013. *0120_4 Arquitectura Vista de Datos.* s.l. : Universidad de las Ciencias Informáticas, 2013.

Quiñones Rosales, Diosnel y Santana García, Tatiana . 2010. Diseño del Motor de Procesamiento de Sentencias para el Sistema Chronos. [En línea] 2010. [Citado el: 01 de marzo de 2014.] http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_03337_10.

RapidSVN. 2014. RapidSVN. *RapidSVN.* [En línea] 2014. [Citado el: 24 de febrero de 2014.] <http://www.rapidsvn.org/>.

Riordan, Rebecca M. 2000. *Diseño de bases de datos relacionales con Access y SQL Server.* España : s.n., 2000. ISBN: 84-481-2770-6.

Rodríguez, Esteban. 2014. Administración de Base de Datos. Rplica de Base de Datos. [En línea] 2014. [Citado el: 24 de febrero de 2014.] https://www.academia.edu/4964988/Replicacion_BD.

Rozic, Sergio Ezequiel. 2004. *Bases de Datos y su aplicación con SQL.* Buenos Aires : MP Ediciones, 2004. ISBN 987-526-213-7.

SQL Manager. 2014. SQL Manager .net. [En línea] 2014. [Citado el: 24 de febrero de 2014.] <http://www.sqlmanager.net/products/postgresql/datagenerator>.

Tedeschi, Nicolás. 2014. Microsoft Developer Network. *Microsoft Developer Network.* [En línea] 2014. [Citado el: 13 de 05 de 2014.] <http://msdn.microsoft.com/es-es/library/bb972240.aspx>.

UCI. 2011. www.uci.cu. *Portal de la Universidad de las Ciencias Informáticas.* [En línea] 2011. [Citado el: 12 de noviembre de 2012.] www.uci.cu.

UCV. 2013. Universidad Central de Venezuela. [En línea] 2013. [Citado el: 5 de diciembre de 2013.] [www.ciens.ucv.ve:8080/genasig/Taller%203%20\(Integridad\).doc](http://www.ciens.ucv.ve:8080/genasig/Taller%203%20(Integridad).doc).

UML. 2013. UML. *UML.* [En línea] 2013. [Citado el: 14 de 12 de 2013.] <http://www.uml.org/>.

Usero Ruiz, Alejandro y Valero Vergara, Francisco José. 2014. Ejemplo de desarrollo software utilizando la metodología XP. [En línea] Universidad Politecnica de Valencia (DESIIC), enero de 2014. [Citado el: 24 de febrero de 2014.] <http://users.dsic.upv.es/asignaturas/facultad/lisi/ejemploxp/>.

UTM. 2013. Universidad Tecnologica de la Mixteca. [En línea] 2013. [Citado el: 8 de diciembre de 2013.] <http://www.utm.mx/~caff/doc/El%20Proceso%20Unificado%20Rational.pdf>.

2014. VisualParadigm. *VisualParadigm*. [En línea] 2014. [Citado el: 15 de 01 de 2014.] <http://www.visual-paradigm.com/>.