

UNIVERSIDAD DE LAS CIENCIAS INFORMATICAS

FACULTAD 3



Título: Diseño e implementación de la base de datos del módulo Inspección a Locales de Detección y Centros Penitenciarios

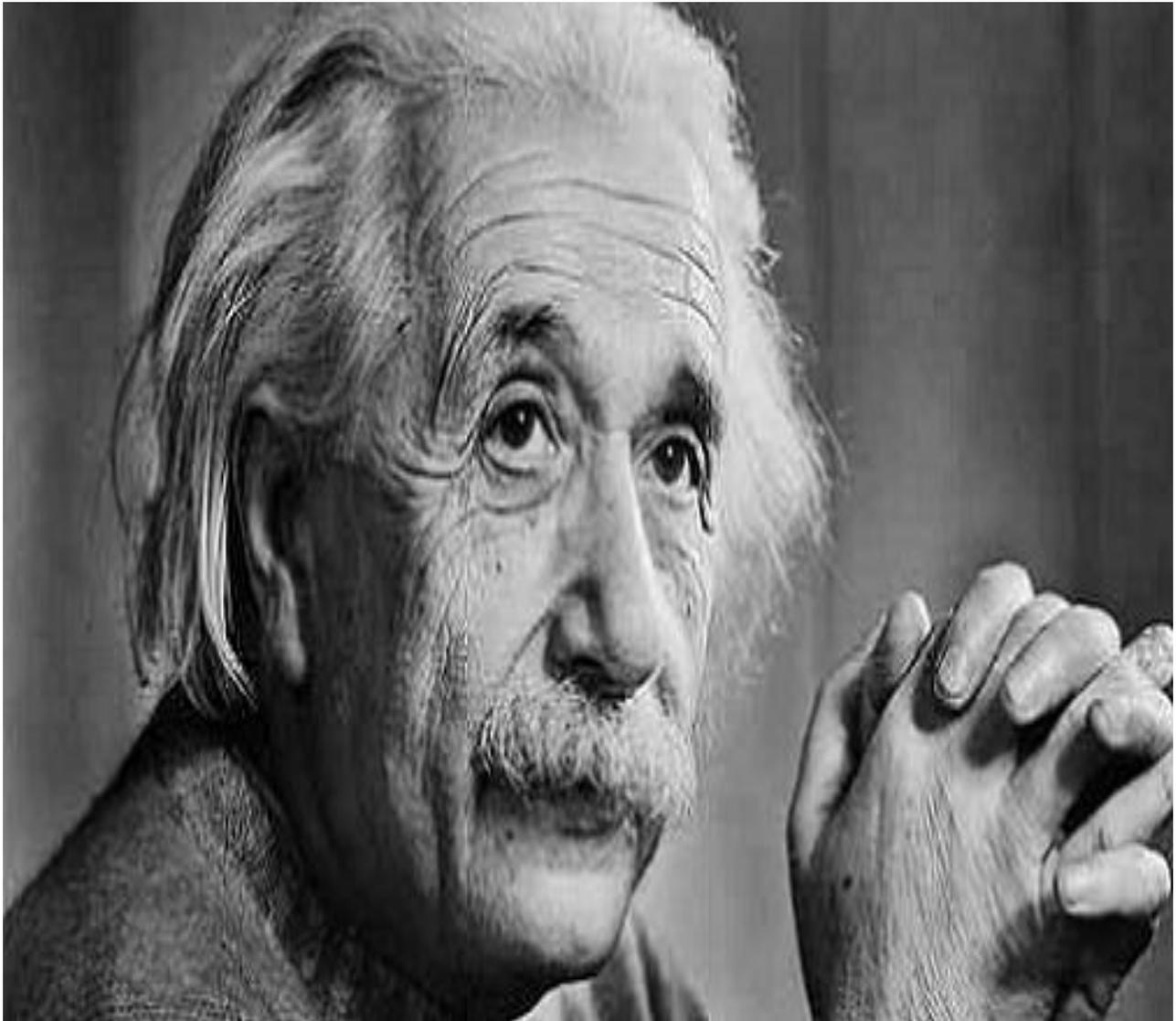
Trabajo de Diploma para optar por el título de Ingeniero en
Ciencias Informáticas

Autor: Jose Manuel García Alvarez

Tutor: Ing. Marieta Peña Abreu

La Habana, junio de 2012

“Año 54 de la Revolución”



*"Nunca consideres el estudio como una obligación sino como una oportunidad
para penetrar en el bello y maravilloso mundo del saber."*

Albert Einstein

Declaración de autoría

Declaro que soy el único autor de este trabajo: “Diseño e Implementación de la base de datos para el módulo de Inspección a Locales de Detección y Centros Penitenciarios” y apruebo a la facultad 3 y en particular al proyecto Sistema de Informatización de la Gestión en las Fiscalías (SIGF) de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su provecho.

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

Jose Manuel García Alvarez

Ing. Marieta Peña Abreu

Autor

Tutora

Agradecimientos

Agradezco a la UCI por fomentar valores en mi conducta y hacerme una mejor persona. A todos los profesores que contribuyeron con mi formación profesional.

A mis hermanos de todo este tiempo, Rafael, Dannier, Dayron y Rasiel por apoyarme cada vez que lo necesité, por compartir conmigo, tanto momentos malos como buenos.

A los amigos de ayer, de hoy y para siempre Ramón, Maye, Lino, Carlos, Manuel, Yosmany, Arli, Pedro Frank, Giorgy, Pedro Enrique, Felipe, Edgar y Raul, fanático mío, gracias por su existencia y por haber convertido estos años, en un libro de recuerdos.

A Vlamir, Marieta y Mangly por su preocupación, por su enorme e incuestionable apoyo durante el desarrollo del trabajo. Gracias de corazón por todo lo que hicieron por mí durante estos largos años.

A Celia, Bertha y Manuel por tratarme en todo momento como un hijo, por brindarme tanto cariño y preocupación. A ellos les estoy enormemente agradecido, y me siento orgulloso por saber que puedo contar con ellos. A toda mi familia en especial a mi abuela Hilda y mi abuelo Evacio que siempre han estado al tanto de mis pasos.

Por último quiero dar agradecimiento especial a mis padres y a mi hermano por ser la luz que me ha guiado a transitar por la vida, por fomentar valores éticos en mí, por su enorme preocupación, por los lazos de amor y cariño con mi persona, por estar al tanto de cada paso que doy en la vida, por convertirme en un verdadero profesional.

A todos desde lo más profundo de mi corazón les doy **GRACIAS**.

Dedicatoria

De manera especial, dedico el presente trabajo de diploma:

A mi tía Yolanda, por haber hecho tanto por mí en estos tiempos, a Celia, a la cual considero como mi madre, persona especial que siempre me supo ayudar sin importar nada.

A Bertha y Manuel, mi familia de Bayamo, que tanto cariño me ofrecieron y tanto hicieron por mí durante estos largos 5 años.

A mi hermano de estos 5 años en la escuela, Rasiel, por guiar mis pasos, y estar conmigo en todo momento tanto en los buenos como en los malos.

Por último se la dedico a las tres personas más importantes para mí de la vida, Jose Alberto (padre), Estrella (madre) y Carlos mi hermano, gracias a ellos he superado todo los obstáculos para llegar hasta donde estoy, gracias por su cariño, por su atención, gracias por todo.

Resumen

En el grupo de fiscalías cubanas existentes uno de los procesos que se personifica es el Control de las Legalidades en Establecimientos Penitenciarios, en el mismo la gestión de la información se lleva a cabo de modo manual; esto trae consigo que se pueda originar mediación de errores, así como existencias de duplicaciones de datos, además se corre el riesgo de extraviar determinada información ya que la misma se encuentra a expensas de sufrir daños debido a factores de riesgo como la humedad y las plagas que pueden actuar sobre los lugares que se estimaron para el almacenamiento. Para dar solución a estos problemas se desarrolló el módulo Inspección a Locales de Detección y Centros Penitenciarios, el mismo debe poseer la cualidad de gestionar un gran cúmulo de información y para ello se llevo a cabo el desarrollo de una base de datos.

El presente trabajo de diploma se ajusta a la idea de conformar el diseño e implementación de una base de datos para el módulo de Inspección a Locales de Detección y Centros Penitenciarios, para el desarrollo del mismo se evidenció el estudio de una serie de temáticas referentes al diseño de base de datos, además sentó su guía en la utilización de herramientas especificadas por el proyecto, como gestor de base de datos y herramientas administrativas para la gestión de las mismas. Se realizó el diseño del modelo de datos a través de la utilización de patrones de diseño, se aplicó el proceso de normalización a dicha base de datos y se tuvo en cuenta el tratamiento de los índices.

Además se ejecutaron un conjunto de pruebas para medir rendimiento propiciando que las validaciones otorgaran resultados satisfactorios. Finalmente se alcanzó efectuar una propuesta de modelo de datos que responde a los objetivos definidos.

Palabras claves

Base de datos, diseño, modelo de datos.

ÍNDICE

Introducción	1
Capítulo 1 Fundamentación teórica.....	6
1.1 Introducción.....	6
1.2 Fundamentos de las base de datos.	6
1.2.1 Características, ventajas y desventajas de las bases de datos.	7
1.3 Tipos de Bases de Datos.	9
1.4 Modelo lógicos de base de datos.	10
1.5 Diseño de base de datos.	12
1.6 Patrones de diseño de Base de Datos.....	15
1.7 Normalización de las bases de datos.....	19
1.8 Integridad de los datos	20
1.9 Lenguajes de modelado.	21
1.9.1 Herramienta CASE para modelado de datos. Visual Paradigm5.0.	21
1.10 SGBD. PostgreSQL 8.4.....	22
1.11 Herramientas Administrativas. PgAdmin III.....	23
1.12 Conclusiones Parciales.....	24
Capítulo 2 Propuesta de solución	26
2.1 Introducción.....	26
2.2 Requisitos.	26
2.2.1 Requisitos Funcionales.	27
2.2.2 Requisitos no Funcionales.	35
2.3 Arquitectura de datos.....	36
2.4 Características generales de la base de datos del módulo ILDCP.	38
2.5 Diseño de la BD.....	38
2.5.2 Patrones de diseño.....	39
2.5.3 Modelo de datos.	40
2.5.4 Descripción de las tablas de la base de datos.....	40
2.6 Uso de índices.....	46
2.6.1 Resultados de las pruebas de los índices.	48
2.7 Conclusiones parciales.....	50

Capítulo 3 Validación de la solución propuesta.....	51
3.1 Introducción.....	51
3.2 Validación teórica del diseño.....	51
3.2.1 Integridad de los datos.....	51
3.2.2 Normalización de la BD.....	53
3.2.3 Análisis de la seguridad de la base de datos.....	55
3.3 Validación Funcional.....	56
3.3.1 Prueba de rendimiento.....	56
3.3.2 Prueba de volumen.....	56
3.3.3 Prueba de Estrés.....	59
3.4 Conclusiones Parciales.....	60
Conclusiones Generales.....	61
Recomendaciones.....	62
Referencias Bibliográficas	63
Bibliográficas Consultadas.....	64

ÍNDICE DE ILUSTRACIONES

Ilustración 1. Diagrama de modelo de datos jerárquico.	10
Ilustración 2. Ejemplo de un Modelo de red.	11
Ilustración 3. Ejemplo de un modelo de datos orientado a objetos.	11
Ilustración 4. Ejemplo de un modelo de datos relacional.	12
Ilustración 5. Fases del diseño de una base de datos.	13
Ilustración 6. Ejemplo de patrón grafo simple.	16
Ilustración 7. Ejemplo de patrones para flujos de trabajo.	17
Ilustración 8. Ejemplo del patrón árboles simples.	18
Ilustración 9. Ejemplo de patrón árboles fuertemente codificados.	18
Ilustración 10. Arquitectura de la base de datos.	37
Ilustración 11. Descripción de los estándares de nomenclatura.	39
Ilustración 12. Ejemplo del patrón llave subrogada.	39
Ilustración 13. Modelo de datos.	40
Ilustración 14. Relaciones de la tabla tbpersonaclep.	41
Ilustración 15. Relaciones de la tabla nomencladora nviolacionclep.	44
Ilustración 16. Ejemplo 1.	54
Ilustración 17. Ejemplo 2.	54
Ilustración 18. Resultados de la generación de datos de tbdocumentoclep.	58
Ilustración 19. Resultados de la generación de datos de tbperosnaclep.	58

ÍNDICE DE TABLAS

Tabla 1. Descripción de la tabla tbpersonaclep.	41
Tabla 2. Descripción de la tabla tbviolacionresolucionpersonaclep.	41
Tabla 3. Descripción de la tabla tbpersonalresolucionclep.	42
Tabla 4. Descripción de la tabla tbviolacionactapersonaclep.	42
Tabla 5. Descripción de la tabla tbpersonaactaclep.	42
Tabla 6. Descripción de la tabla tbpersonadestinosolclep.	42
Tabla 7. Descripción de la tabla tbresolucionclep.	43
Tabla 8. Descripción de la tabla tbactainspeccionclep.	43
Tabla 9. Descripción de la tabla tbviolacionresolucionclep.	44
Tabla 10. Descripción de la tabla tbautolibertadclep.	45
Tabla 11. Descripción de la tabla tbviolacionactaclep.	45
Tabla 12. Descripción de la tabla nviolacionclep.	46
Tabla 13. Descripción de la tabla tbviolacioninspeccionclep.	46
Tabla 14. Definición de índices.	48
Tabla 15. Definición de los indicadores de prueba.	49
Tabla 16. Índice (planinspeccion_confecciona_index).	49
Tabla 17. Índice (planinspeccion_aprueba_index).	49
Tabla 18. Índice (registroinspeccion_fiscalconfecciona_index).	49
Tabla 19. Índice (regresoluciones_tipoinfractor_index).	49
Tabla 20. Datos generados en la base de datos clep.	57
Tabla 21. Pruebas.	59
Tabla 22. Resultados de las muestras.	60

Introducción

En el mundo actual, principalmente en países desarrollados, ha ido progresando con el tiempo la idea de conformar una sociedad informatizada, la cual propicie innumerables ventajas y flexibilidad en el trabajo en las diferentes esferas de la sociedad.

El estado cubano está consciente de que una sociedad para ser más eficaz, eficiente y competitiva debe aplicar la informatización en todas sus esferas, procesos y segura de que para países subdesarrollados resulta imprescindible este propósito. Nuestro país se ha percatado de la necesidad de introducir en la práctica social y cotidiana las Tecnologías de la Información y las Comunicaciones (TIC) y alcanzar una cultura digital como una de las características esenciales del hombre nuevo. Las TIC son de gran importancia en la actualidad ya que completan el conjunto de recursos necesarios para el manejo de la información que circula constantemente por el mundo, apoyado esto en las vías de comunicación que existen actualmente.

Debido a la necesidad imperante de Cuba de formar parte de la Era Digital manifestada en la actualidad, surge como una de las propuestas al llamado de informatizar la sociedad cubana, la Universidad de las Ciencias Informáticas (UCI), la misma encargada de formar profesionales en la materia y alcanzar un elevado nivel en la producción de software.

La UCI está compuesta por diferentes facultades, una de estas facultades es la 3 en la cual radica el Centro de Gobierno Electrónico (CEGEL), el mismo se ha hecho responsable de brindar potencialidades a determinadas entidades de orden jurídico, es decir desarrollar el proceso de informatización en las mismas, esto con el fin de colaborar con el proceso de auge del gobierno electrónico en el país.

Debido a los distintos convenios realizados entre la UCI, con determinadas entidades y organizaciones de la sociedad, viene al caso la creación del proyecto Sistema de Informatización de la Gestión en las Fiscalías (SIGF) el mismo constituyó un acuerdo trazado entre dicha universidad y la Fiscalía General de la República de Cuba (FGR). Dicho proyecto presenta como objetivo primario informatizar los procesos que se originan en la Fiscalía General de la República, Fiscalías Provinciales y Fiscalías Municipales, en

aras de engrandecer la perfección en la planificación, organización y ejecución de los mismos, lograr alcanzar una mejor eficiencia en el acceso a la información.

La FGR, según lo fijado en la Constitución es el órgano del estado que tiene trazado sus objetivos en el control y la preservación de la legalidad sobre la vigilancia del estricto cumplimiento de la Constitución, las leyes y demás disposiciones legales por los organismos del Estado, así como el proceder de la acción penal pública en representación del Estado.

El proyecto SIGF se encuentra estructurado por 5 subsistemas compuestos estos por 13 módulos ajustados a aquellos procesos que se personifican en las fiscalías, entre los que se hallan:

- **Control de la Legalidad en Establecimientos Penitenciarios (CLEP):** Inspección a Locales de Detección y Centros Penitenciarios.
- **Herramientas Comunes a todas las Áreas (HCTA):** Diccionario jurídico y Gestión de Entrada y Salida de Documentación.
- **Procesos Penales (PP):** Sumario, Ordinario e Índice de Peligrosidad Pre delictiva.
- **Gestión de Cuadros y Personal de Apoyo (GCPA):** Gestión de Cuadros, Gestión del Personal de Apoyo, Capacitación, Reportes y Búsquedas, Generalidades y Capacitación de Plantilla.
- **Verificación Fiscal (VF):** Procesos confiscatorios por Decreto Ley 149.

Dentro del subsistema CLEP, radica el módulo Inspección a Locales de Detección y Centros Penitenciarios, el mismo dedicado a la planificación y ejecución de las visitas de inspección a los establecimientos y unidades en que se cumpla la prisión provisional de acusados.

Actualmente en las fiscalías las actividades o procesos referentes a este módulo no se llevan a cabo de una manera eficiente, debido a que se le hace engorroso al personal responsable de dicha tarea mantener un buen tratamiento de los datos, debido al gran cúmulo de escritos que deben manejar. Las acciones de búsquedas conllevan grandes sumas de tiempo a raíz de la cantidad de información que hay registrada, y por encima de todo se corre un enorme riesgo de extraviar determinada información debido al desgaste al que son sometidos todos esos datos. Además de que toda la tramitación de los datos

se realiza de forma manual lo que puede provocar mediación de errores en la gran aglomeración de expedientes y documento que se originan.

Toda esta información es conservada en lugares poco seguros, en los cuales están a expensas de ser dañados ya sea por insectos, humedad, fuegos u otros factores. Por lo cual se origina la urgencia de garantizar un modo seguro de almacenamiento y gestión para toda la información manejada en las fiscalías referentes al módulo Inspección a Locales de Detección y Centros Penitenciarios.

De lo antes expuesto se deriva el siguiente **problema a resolver**: ¿Cómo garantizar el almacenamiento de la información gestionada por el Módulo de Inspección a Locales de Detección y Centros Penitenciarios?

Siendo el **objeto de estudio**: el diseño e implementación de base de datos relacionales.

Para dar solución al problema planteado se define como **objetivo general**, realizar el diseño y la implementación del modelo de datos para la gestión de la información del módulo Inspección a Locales de Detección y Centros Penitenciarios.

Se presenta como **campo de acción** el diseño e implementación del modelo de datos del módulo Inspección a Locales de Detección y Centros Penitenciarios.

A raíz del problema enunciado anteriormente, se define la siguiente **idea a defender**, con la construcción del diseño e implementación del modelo de datos para el módulo de Inspección a Locales de Detección y Centros Penitenciarios se contribuirá de una mejor manera al almacenamiento y gestión de la información generada en dicho módulo.

Como **objetivos específicos** se definen:

- ✓ Elaborar el marco teórico de la investigación.
- ✓ Diseñar la propuesta de la base de datos para el módulo de Inspección a Locales de Detección y Centros Penitenciarios.
- ✓ Implementar la propuesta de la base de datos para el módulo de Inspección a Locales de Detección y Centros Penitenciarios.
- ✓ Validar la solución propuesta a través de pruebas de carga y rendimiento.

Para propiciar el cumplimiento de estos objetivos se trazaron las siguientes **tareas investigativas**:

- ✓ Estudio del marco teórico que fundamenta el objeto de investigación.
- ✓ Selección de la herramienta de modelado.
- ✓ Selección del gestor de base de datos.
- ✓ Diseño del Modelo de datos.
- ✓ Implementación de la base de datos.
- ✓ Selección de las pruebas a realizar para validar solución.
- ✓ Validación de la solución propuesta.

Para dar cumplimiento a las tareas antes planteadas se contará con el empleo de los **Métodos Científicos de la Investigación**: teórico y empírico.

Los métodos teóricos permiten estudiar las características del objeto de investigación que no se observan directamente, facilitan la construcción de modelos e hipótesis de investigación y crean las condiciones para ir más allá de las características fenomenológicas y superficiales de la realidad, contribuyendo al desarrollo de las teorías científicas y para su ejecución se apoyan en el proceso de análisis y síntesis. Dentro de los métodos investigativos teóricos se emplearán:

- ✓ **Analítico – Sintético**: Se utilizó este método ya que permitió procesar la información y llegar a conclusiones referentes a la investigación, así como definir el conjunto de herramientas a utilizar para el diseño e implementación de dicha base de datos.
- ✓ **Histórico – Lógico**: Se empleó debido a que se llevó a cabo una profunda investigación para contemplar un mejor análisis de la trayectoria de los sistemas de base de datos referentes a las fiscalías.

Los métodos empíricos nos permiten extraer de los fenómenos analizados las informaciones que se necesitan sobre ellos a través de observaciones, del uso de técnicas ofimáticas y la propia experimentación. Dentro de los Métodos de Investigación empíricos se emplearán:

- ✓ **Entrevista**: Se desarrollaron entrevistas con conocedores y especialistas con alta experiencia en temas relacionados con el diseño e implementación de base de datos, con el propósito de profundizar en las diferentes temáticas a tratar.

La presente investigación está conformada por 3 capítulos.

Capítulo 1: “Fundamentos teóricos de la investigación”.

En el presente capítulo se fundamentan algunos aspectos de interés relacionados con el tema de las bases de datos, se hace alusión a conceptos, características, ventajas y desventajas, entre otros temas se hace mención a los distintos modelos de bases de datos que existen así como a las diferentes etapas de diseño de bases de datos. Además se realizó un estudio de las distintas herramientas que se utilizarán durante el desarrollo del trabajo.

Capítulo 2: “Propuesta de solución”.

En el presente capítulo se expondrán los procesos principales que toman lugar en la elaboración y construcción de la BD. Se expone una descripción de la propuesta de solución, se evidencia la arquitectura de datos, así como la utilidad de estándares de nomenclatura y patrones de diseño. Se exhibe el modelo de datos, así como la descripción de las tablas contenidas en la misma. Se argumenta el tratamiento de los índices.

Capítulo 3: “Validación de la solución”.

En el presente capítulo tendrá lugar el desarrollo del proceso de validación del diseño e implementación de la base de datos del módulo Inspección a Locales de Detección y Centros Penitenciarios, abordando elementos de gran importancia e interés como la integridad, redundancia, normalización y seguridad de la base de datos. Para contribuir al desarrollo del proceso de validación se realizan pruebas de carga y rendimiento centradas en la idea de comprobar el correcto funcionamiento de la base de datos, así como el rendimiento de la aplicación.

Capítulo 1 Fundamentación teórica.

1.1 Introducción.

En el presente capítulo se fundamentan algunos aspectos de interés relacionados con el tema de las bases de datos, se hace alusión a conceptos, características, ventajas y desventajas, entre otros temas se hace mención a los distintos modelos de bases de datos que existen así como a las diferentes etapas de diseño de bases de datos. Además se realizó un estudio de las distintas herramientas que se utilizarán durante el desarrollo del trabajo.

1.2 Fundamentos de las base de datos.

El término Base de Datos (BD) fue escuchado por primera vez en el año 1963. Una BD no es más que un contenedor en el cual se almacena un gran cúmulo de información estructurada con la cual el usuario interactúa siempre que le sea necesario. (Valdes, 2007)

Como lo dice C.J Date en su libro “Introducción a los sistemas de base de datos” una base de datos se puede definir como: “un conjunto de datos persistentes que es utilizado por los sistemas de aplicación de alguna empresa dada.” (Date, 2001)

Otro concepto propuesto es el plasmado en el libro: “Las Bases de Datos: Una herramienta moderna que contribuirá a la oportuna difusión de información científica” una base de datos es: “una acumulación ordenada de observaciones y anotaciones sobre un tema, hecha de tal manera que los datos tengan una relación lógica entre si y una estructura bien definida.” (Jiménez, 1985)

En términos informáticos, la BD es un sistema formado por un conjunto de datos almacenados en dispositivos (discos), los cuales posibilitan acceso directo a ellos y un grupo de programas o software que manipulen y propicien administración a ese conjunto de datos. (Valdes, 2007). Este concepto será tomado como referencia para el desarrollo del trabajo.

Un concepto más general de lo que es una BD es el siguiente. Una BD es una serie de elementos organizados y relacionados entre sí, los cuales son explotados por los sistemas de información de una empresa, entidad o negocio en particular. (Valdes, 2007)

¿Qué se entiende por un sistema gestor de Base de Datos? Un sistema gestor de Base de Datos no es más que una agrupación de programas o software que posibilitan la definición, construcción y manipulación de una base de datos, es conocido además como Database Management System por sus siglas en inglés (DMBS). (Cavsi, 2004)

Los mismos contienen una serie de servicios tales como:

- Creación de bases de datos.
- Gestión de la información por medio de consultas.
- Establecen un control de acceso regulado a los datos a través de normas de seguridad.
- Mantener una correcta integridad de la información.
- Regular la concurrencia a las bases de datos.

1.2.1 Características, ventajas y desventajas de las bases de datos.

Entre las principales características que poseen los sistemas de BD están: (Valdes, 2007)

- **Independencia lógica y física de los datos.**

Esto indica e infiere que los datos no tienen ninguna dependencia del programa y por tanto cualquier aplicación puede hacer uso de ellos. Esto se refiere a la capacidad de modificar una definición de esquema en un nivel de la arquitectura sin que esta modificación llegue a afectar al nivel superior.

- **Redundancia mínima.**

Lo que se entiende por redundancia es la existencia de duplicaciones de los datos, al llegar a reducir las mismas al máximo se obtiene un mayor aprovechamiento del espacio y se evitan además la existencia de inconsistencias entre los datos, estas inconsistencias se presentan cuando son encontrados datos contradictorios.

- **Acceso concurrente por parte de múltiples usuarios.**

Esto hace posible un control de concurrencia mediante técnicas de bloqueo o cerrado de datos accedidos.

- **Integridad de los datos.**

La misma hace referencia a las llamadas medidas de seguridad que impiden que se introduzcan datos erróneos. Esto puede posibilitarse tanto por motivos físicos (defectos de hardware, actualizaciones incompletas), como de operación (introducción de datos incoherentes).

- **Consultas complejas optimizadas.**

Lo que se deriva de esto es que la optimización de consultas permite la rápida ejecución de las mismas.

- **Seguridad de acceso y auditoría.**

La misma hace referencia al derecho de acceso a los datos existentes en la base de datos por parte de personas, entidades u organismos. El sistema de auditoría mantiene el control de acceso a la base de datos, con el propósito de conocer qué o quién realizó una determinada modificación y en qué momento.

- **Respaldo y recuperación.**

Esto hace referencia a la capacidad de un sistema de base de datos de llegar a recuperar su estado en un momento previo a la pérdida de los datos.

- **Acceso a través de lenguajes de programación estándar.**

Esto consiste en hacer posible el acceso a los datos de una base de datos mediante lenguajes de programación ajenos al sistema de base de datos propiamente dicho.

Los sistemas de Bases de Datos no solo contienen cosas positivas, también poseen desventajas como son el caso: (Valdes, 2007)

- **Complejidad:** los sistemas gestores de BD son programas que pueden llegar a ser complejos con una gran funcionalidad. Es importante llegar a comprender esta funcionalidad para poder brindar un buen uso de ellos.
- **Coste del equipamiento adicional:** las BD pueden hacer que sea necesario depender de más espacio de almacenamiento, para adquirir estas prestaciones, se hace necesario adquirir una máquina más grande o una que se dedique por completo a la tarea de administrar el sistema gestor de BD. Todo esto conlleva a que la implantación del sistema sea más costosa.
- **Vulnerable a los fallos:** el hecho de que todas las tareas estén centralizadas en el sistema gestor de BD hace que sea más vulnerable a los fallos que puedan propiciarse, es necesario por tal motivo tener copias de seguridad.

1.3 Tipos de Bases de Datos.

Las bases de datos pueden clasificarse de diferentes maneras, atendiendo al tema y utilidad que se necesite. (Valdes, 2007)

Según la variabilidad de los datos que se almacenan

- **Bases de datos estáticas:** son BD de solo lectura, utilizadas para almacenar datos históricos que puedan ser utilizados luego para estudiar el comportamiento de un cúmulo de datos a través del tiempo y tomar decisiones.
- **Bases de datos dinámicas:** en estas la información almacenada se modifica con el tiempo, permitiendo operaciones básicas como son, adicción, eliminación y actualización de datos., además de operaciones fundamentales de consulta.

Según el contenido

- **Bases de datos bibliográficas:** solo contienen un subrogante. Un registro de esta BD contiene información sobre el autor, fecha de publicación, editorial, título, todo esto de una determinada publicación.
- **Bases de datos de texto completo:** son las encargadas de almacenar todas las fuentes primarias de cualquier tema.

- **Directorios:** un ejemplo clásico de una base de datos de este tipo son las guías o directorios telefónicos.

1.4 Modelo lógicos de base de datos.

Se hace necesario conocer que es un modelo de base de datos. “Un modelo de base de datos es un conjunto de ideas lógicas utilizadas para representar la estructura de datos y las relaciones entre ellos dentro de la base de datos.” (Ortiz, 2000)

En el transcurso de los años se han ido creando diversos modelos para el diseño de base de datos. Cada uno de ellos centra sus objetivos en actividades matemáticas para la descripción de la base de datos.

Modelos jerárquicos: modelo de datos en el cual la información es estructurada y organizada en forma de un árbol. Dicha estructurada posibilita la utilización de padres e hijos sobre la información que maneja, donde cada padre puede tener varios hijos, pero cada hijo solo posee un padre. (Chavez, 2005)

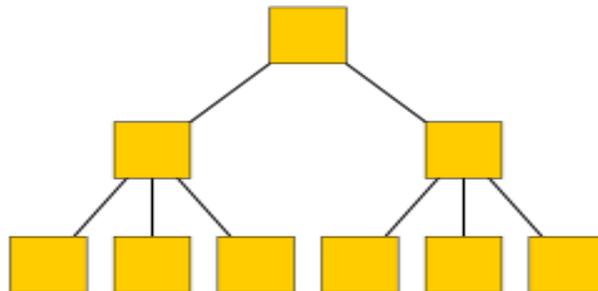


Ilustración 1. Diagrama de modelo de datos jerárquico.

Modelo de red: modelo de base de datos, previsto como un modo flexible de representar objetos y su relación. Las actividades matemáticas que utilizan para la descripción de la base de datos son herramientas semánticas. (Chavez, 2005)

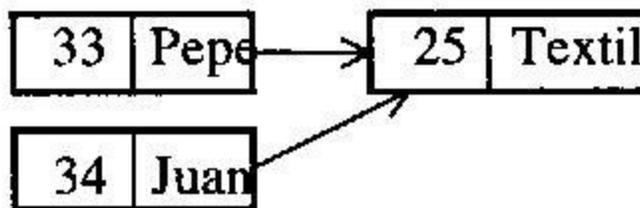


Ilustración 2. Ejemplo de un Modelo de red.

Modelo orientado a objetos: en este modelo la información es representada mediante objetos. Aquellos objetos que presentan la misma estructura forman parte de una misma clases, las cuales son estructuradas mediante grafos acíclicos. (Chavez, 2005)

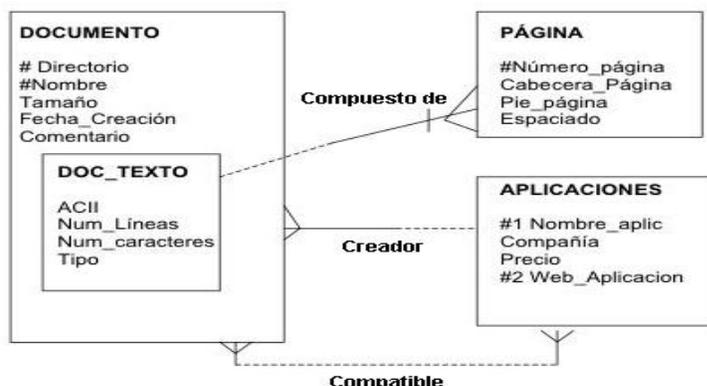


Ilustración 3. Ejemplo de un modelo de datos orientado a objetos.

Modelo Relacional: fue presentado por Edgar Codd en una publicación en 1970, refiriéndose a un modelo de datos en específico, que se caracterizaba por contar con relaciones como único objeto de tratamiento en el modelo, así también definió un álgebra como lenguaje de consulta a la que llamó Álgebra Relacional (AR), pero no contaba con ninguna manera de expresar actualizaciones, restricciones y/o cálculos sobre el modelo. (Vieyra, 2005). Entre sus principales características se destaca:

- Cada tabla es a su vez un conjunto de registros almacenados en filas o tuplas.
- Cada registro representa un objeto del mundo real.
- Los valores almacenados en una columna deben ser del mismo tipo de dato.

- Todas las filas de una misma tabla poseen el mismo número de columnas.
- No se considera el orden en que se almacenan los registros en las tablas.
- No se considera el orden en que se almacenan las tablas en la base de datos.

El modelo relacional es el modelo de datos más ampliamente usado, la gran mayoría de los sistemas de bases de datos actuales están basados en él. (Vieyra, 2005)

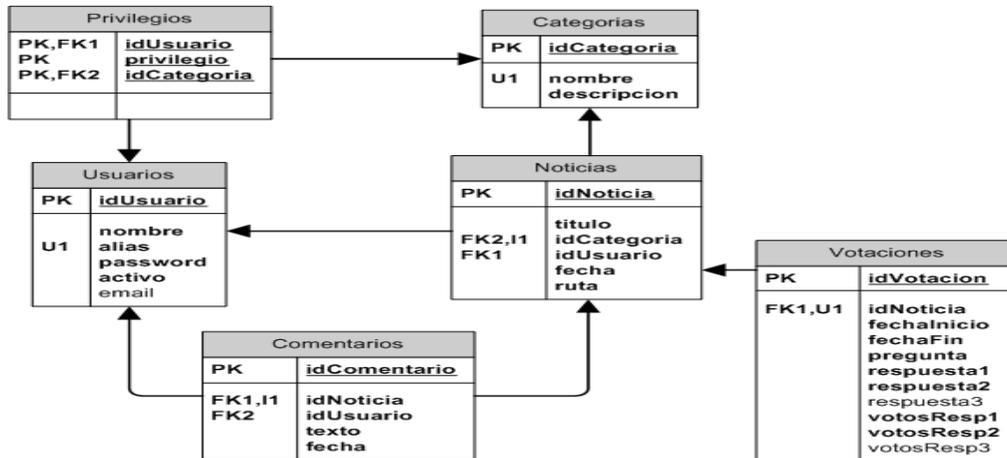


Ilustración 4. Ejemplo de un modelo de datos relacional.

En este modelo, basado en la teoría matemática de las relaciones, los datos se estructuran lógicamente en forma de relaciones. Esta formalización matemática convirtió rápidamente al modelo en una fuente fundamental de investigación. (Vieyra, 2005). El desarrollo de este trabajo tiene sentadas sus bases sobre este modelo, ya que el mismo asegura el tratamiento de los datos por medio de relaciones, es de esta forma que se continua con la estrategia desplegada por el proyecto Sistema de Informatización de la Gestión en las Fiscalías.

1.5 Diseño de base de datos.

El diseño de bases de datos es aquel proceso encargado de facturar la organización de la base de datos. Dicho proceso se guía por una serie de principios. Uno de ellos es que se debe evitar la redundancia de datos, ya que malgastan el espacio y aumentan la probabilidad de que se produzcan errores e incoherencias. El segundo principio es que es importante que la información sea correcta y completa. (Chavez, 2005)

Entre los objetivos que persigue este proceso, se encuentran los siguientes: (Chavez, 2005)

- Acceso eficiente a la información con redundancia mínima.
- Propiciar un modelo de datos que tolere las acciones que se vayan a realizar al cúmulo de información.
- Especificaciones de limitantes, así como definir un esquema que responda a las peticiones pretendidas.

Un correcto diseño de base de datos propicia obtener acceso a información exacta y actualizada. Puesto que un diseño correcto es esencial para lograr los objetivos fijados para la base de datos. (Chavez, 2005)

El proceso de diseño de bases de datos está conformado por una serie de pasos o fases a seguir, estas son:

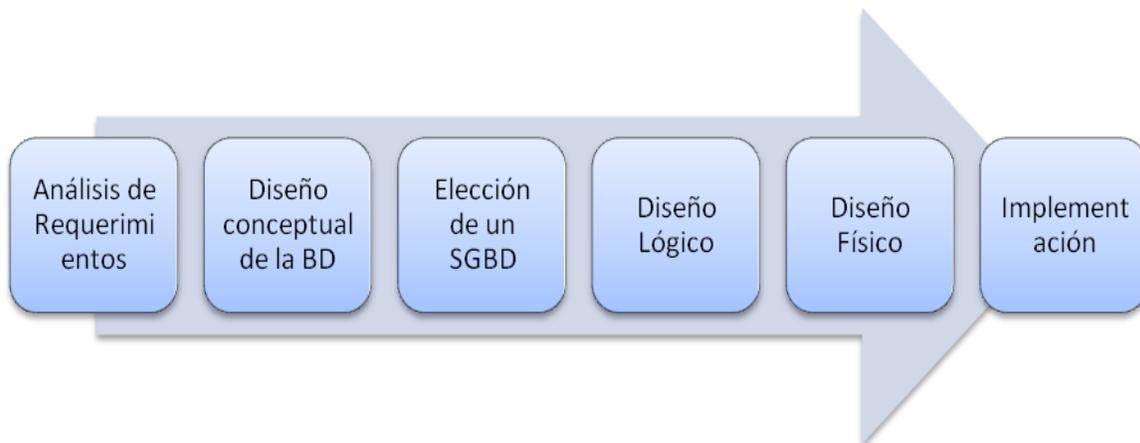


Ilustración 5. Fases del diseño de una base de datos.

Fase 1: Análisis de Requerimientos.

En dicha fase se realizan entrevistas a los usuarios por parte de los diseñadores para conocer las ideas de cada cual, y documentar las demandas de información. Además se hace importante definir los requerimientos funcionales que son operaciones que se harán a la base de datos. Se identifican los posibles usuarios, junto con las áreas de aplicación de dicha base de datos.

Fase 2: Diseño Conceptual de la Base de Datos.

En esta etapa se obtiene una estructura de la información de la futura base de datos independientemente de la tecnología que se emplee. En dicha etapa no se tiene en cuenta aún el tipo de base de datos que se va a utilizar, así como tampoco el sistema gestor de base de datos, ni con el lenguaje que se va a llevar a cabo la implementación de la base de datos. Esta etapa permite concentrarse únicamente en la problemática de la estructuración de la información. El resultado es expresado mediante algún modelo de datos de alto nivel, ejemplo de esto el modelo entidad-relación.

Fase 3: Elección de un SGBD.

Con el objetivo de realizar la elección de un SGBD para dar cumplimiento a esta fase se tienen en cuenta un grupo de factores técnicos, económicos y de beneficio. Resulta un tanto complejo la forma de considerar los factores.

Fase 4: Diseño Lógico.

En dicha etapa se comienza a raíz del resultado del diseño conceptual, que es transformado de forma que se adapte a la tecnología a emplear. Aquí es preciso que se ajuste al modelo del SGBD con el que se desea implementar la base de datos.

Esta etapa parte del hecho de que ya haya sido resuelta la problemática de la estructuración de la información en el ámbito conceptual, y posibilita la concentración en las cuestiones tecnológicas relacionadas con el modelo de bases de datos.

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

Fase 5: Diseño Físico.

El objetivo primario del diseño físico no es otro que conseguir una instrumentación bastante eficiente del esquema lógico, para esto son analizados algunos aspectos como son los casos específicos de las características del sistema operativo, el sistema gestor de base de datos, la herramienta para lograr la realización del diseño, además aspectos

relacionados con el rendimiento y requisitos de procesos así como las características del hardware, en resumen, cualquier factor relacionado con el ordenador, logrando la optimización del consumo de recursos, minimizar el espacio de almacenamiento, proporcionar la seguridad máxima, disminuir los tiempos de respuesta y evitar las reorganizaciones.

Fase 6: Implementación.

En esta fase se hace realidad la base de datos, mediante la creación y la compilación de uno o varios esquemas y ficheros de bases de datos, así como de las transacciones a través de las aplicaciones.

1.6 Patrones de diseño de Base de Datos.

En el diseño y el modelado de bases de datos es frecuente la presencia de elementos repetitivos en modelos, durante el tratamiento y resolución de estos últimos se pueden manifestar excelentes prácticas que pueden llegar a constituir patrones de diseño. Por lo tanto, un patrón de diseño es una abstracción de una solución en un nivel alto. Los patrones solucionan problemas que existen en muchos niveles de abstracción y abarcan las distintas etapas del desarrollo. Concluyendo, un patrón de diseño es un conjunto de reglas que describen como afrontar tareas y solucionar problemas que surgen durante el desarrollo de software. (Blaha, 2010)

Los Patrones de Diseños ayudan a conseguir diseños optimizados de manera ágil y con pocos esfuerzos al proponer buenas prácticas, soluciones pensadas y testeadas. Estos aportan ideas reutilizables y adaptables a un amplio campo de problemas. (Blaha, 2010)

Patrón Entidad-Atributo-Valor

El patrón de modelo de datos, entidad-atributo-valor cuenta con gran utilidad, a la hora de describir las entidades donde el número de atributos a utilizar es en gran medida enorme, pero que aplicados a una entidad concreta es pequeño. Dicho patrón cuenta con beneficios tales como: (Blaha, 2010)

- Flexibilidad. El número de parámetros puede crecer como la base de datos se desarrolla.
- Espacio de almacenamiento eficiente de datos muy escasos. No se hace necesario conservar espacio para los atributos de valores nulos.

Grafo

Los grafos dirigidos representan un conjunto de nodos, así como un conjunto de caminos dirigidos entre los nodos. Los caminos pueden significar cambios de fases o conexiones entre nodos donde cada uno representa roles diferentes. Desde un mismo nodo se puede originar una serie de caminos. Los patrones que se derivan de estos son: (Blaha, 2010)

- **Grafo dirigido simple:** estos patrones se asemejan al de los árboles, solo que las relaciones de uno a mucho son sustituidas por relaciones de mucho a mucho, originando de esta manera una nueva entidad a raíz de la relación. (Blaha, 2010)
- **Grafo dirigido estructurado:** dicho patrón es utilizado cuando los nodos contienen el mismo tipo de dato. Es similar al árbol estructurado, la diferencia que en dicho caso la relación que se establece sobre los nodos tiene cardinalidad de muchos a muchos. (Blaha, 2010)

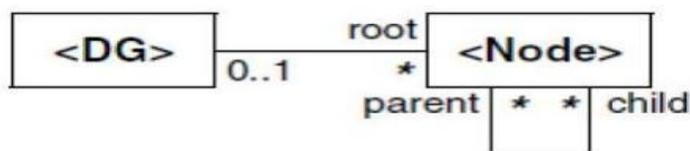


Ilustración 6. Ejemplo de patrón grafo simple.

Patrones para flujos de trabajo

Se debe prescindir primero de lo que se entiende por flujo de trabajo, no es más que el estudio de los aspectos funcionales u operacionales de una actividad de trabajo. En los últimos años se ha trabajado en la idea de llevar estos patrones al campo de las bases de

datos orientadas a objetos. En la actualidad existen dos patrones definidos para diseñar un flujo de trabajo, estos son:

- **Máquina de estado para un tipo de entidad:** se emplea con el fin de mostrar los posibles cambios de estado por lo que puede cursar un tipo de entidad.
- **Máquina de estado para escenario:** se emplea para representar el cambio de estado en un escenario de una entidad dada.

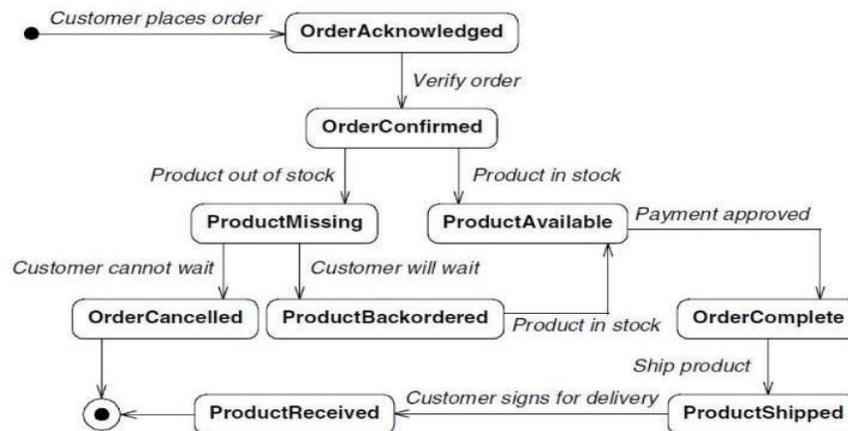


Ilustración 7. Ejemplo de patrones para flujos de trabajo.

Patrones para árboles

Antes de dar paso al tema se hace necesario prescindir de lo que se entiende por árbol en términos informáticos, es una estructura de datos ampliamente usada que imita la forma de un árbol, o se plantea también como un conjunto de nodos conectados. Actualmente en el diseño de BD se prescinde mucho de los árboles, y existen patrones que se derivan de ellos, ejemplo de esto:

- **Árboles Simples:** Este patrón es muy usado cuando el árbol es la representación de una estructura de datos.

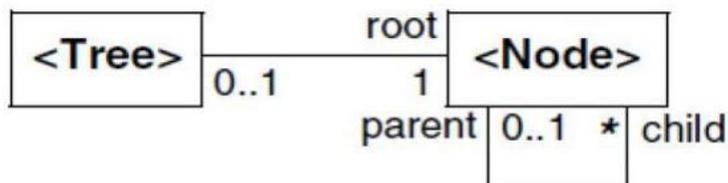


Ilustración 8. Ejemplo del patrón árboles simples.

- **Árboles Fuertemente Codificados:** Este patrón es utilizado en gran medida en las relaciones de uno a muchos para representar jerarquías donde la estructura esté bien definida.

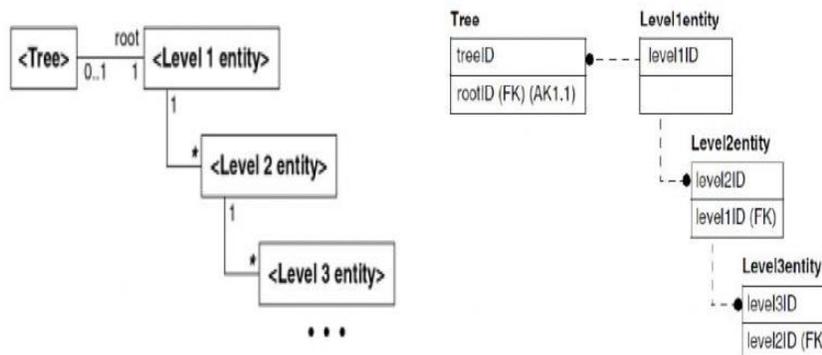


Ilustración 9. Ejemplo de patrón árboles fuertemente codificados.

Patrones de llaves subrogadas

Se conoce como llave subrogada a un identificador único que se asigna a cada registro de una tabla. Por lo general es de tipo numérico. Como dato curioso en la actualidad el empleo de este patrón goza de una amplia reputación por parte de los diseñadores de BD. Algo relevante de este patrón es el hecho de las ventajas que propicia, como son:

- Uniformidad. Se pueden programar tareas de mantenimiento sobre tablas que asumen un esquema de llave primaria común.
- La lógica del negocio no está en las claves.
- Los mecanismos de generación secuencial son rapidísimos y los provee el sistema.

- Dichas claves son pequeñas, enteros que ocupan muy poco. Cuenta con Joins más rápidos.

Gracias al estudio realizado con relación a los patrones de diseño de base de datos existentes y vinculándolo al desarrollo del trabajo en curso, se apoya la idea de utilizar por las posibilidades que aportan, el patrón de llaves subrogadas ya que el mismo genera un identificador único para cada entidad contenida en la base de datos en vez de utilizar un atributo identificador, esto hace más flexible el trabajo, ya que los identificadores serán enteros auto-incrementales.

1.7 Normalización de las bases de datos

El proceso de normalización de la base de datos consiste en aplicar una serie de reglas a las relaciones obtenidas una vez hecho el traspaso del modelo entidad-relación al modelo relacional. (WikiProd, 2009)

Entre los objetivos que persigue dicho proceso está el evitar la redundancia de datos, evitar problemas de actualización de los datos en las tablas, así como proteger la integridad de los datos. (WikiProd, 2009)

Primera Forma Normal (1FN)

Un esquema de relación se encuentra en primera forma normal si los dominios de todos los atributos de la relación son atómicos, un dominio es atómico cuando se considera que los elementos del dominio son unidades indivisibles. Esto se definió para evitar los atributos multivalorados o compuestos. (WikiProd, 2009)

Segunda Forma Normal (2FN)

Un esquema de relación está en segunda forma normal si está en primera forma normal y, además cada atributo del esquema de relación que no está en la clave primaria depende funcionalmente de la clave primaria completa y no sólo de una parte de esta. Dicha regla solo se aplica a los esquemas de relación que tienen claves primarias compuestas por dos o más atributos. (WikiProd, 2009)

Tercera Forma Normal (3FN)

Un esquema de relación está en tercera forma normal si está en segunda forma normal y, además cada atributo del esquema de relación que no está en la clave primaria solo depende funcionalmente de la clave primaria, y no de ningún otro atributo. (WikiProd, 2009)

1.8 Integridad de los datos

La exigencia de integridad de los datos garantiza la calidad de la información de una base de datos, la misma hace alusión a la corrección y completitud de toda la información contenida. Existen condiciones que hacen posible la integridad de los datos, las cuales pueden ser de dos modos. (García, 2005)

- Las restricciones de integridad de usuario: son las que se corresponde cumplir en una base de datos articular con usuarios estipulados, pero que no son prominentes en otra base de datos.
- Las reglas de integridad de modelo: son circunstancias propias de un modelo de datos, y se hace estricto cumplir en toda base de datos regida por dicho modelo.

Existen una serie de reglas de integridad las cuales se fundamentan a continuación:

Regla de integridad de unicidad de la clave primaria

Dicha regla establece que toda llave primaria elegida para una relación no debe tener valores repetidos. (García, 2005)

Regla de integridad de entidad de la clave primaria

Esta regla establece que todos los atributos de la llave primaria de una relación no pueden tener valores nulos. Dicha regla se hace necesaria para el hecho de que puedan identificar las tuplas individuales de las relaciones. (García, 2005)

Regla de integridad referencial

Dicha regla determina que todos los valores que toma una llave foránea deben ser valores nulos o valores que existen en la llave primaria que referencia. La necesidad de esta regla es debido a que las llaves foráneas tienen por objetivo establecer una conexión con la llave primaria que referencian. (García, 2005)

Regla de integridad de dominio

Dicha regla está relacionada con la noción del dominio. Consta de dos condiciones: (García, 2005)

- Todos los valores no nulos que contiene la base de datos para un determinado atributo deben ser del dominio declarado para dicho atributo.
- Establece que los operadores que pueden aplicarse sobre los valores dependen de los dominios de estos valores.

1.9 Lenguajes de modelado.

Los lenguajes de modelado constituyen un conjunto estandarizado de símbolos, así como las distintas maneras de disponer de estos para modelar un diseño de software. (Pressman, 2005). Uno de estos lenguajes existentes es el caso del Lenguaje de Modelado Unificado (UML por sus siglas en inglés).

El mismo constituye un lenguaje gráfico que permite visualizar, especificar, construir y documentar un sistema de software. Dicho lenguaje se emplea para definir un sistema de software, para detallar los artefactos en el sistema y para documentar, así como construir. Dicho lenguaje brinda un modo estándar de representar los planos de un sistema, incluyendo aspectos conceptuales entre los que están los procesos de negocio, así como las funciones del sistema. (Pressman, 2005)

1.9.1 Herramienta CASE para modelado de datos. Visual Paradigm 5.0.

Las herramientas CASE son un grupo de aplicaciones destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en cuanto a tiempo y dinero. Las mismas intentan posibilitar ayuda automatizada a actividades contenidas en el proceso del software. (Pressman, 2005)

Visual Paradigm en su versión 5.0, es una herramienta CASE que brinda un grupo de ayudas para el desarrollo de sistemas informáticos, profundizando aspectos tales como la planificación, análisis y el diseño. La misma ha sido concebida para soportar el proceso completo de desarrollo del software a través de la representación de todo tipo de diagramas.

El uso de esta herramienta se sustenta en el hecho de que el estado cubano cuenta con licencia para la utilidad de la misma, además de que ayuda a la construcción de aplicaciones de calidad de manera más rápida y con un menor coste. Permite el diseño de todos los tipos de diagramas de clases, código inverso, además de generar código desde diagramas y generar documentación. Entre las principales características que ofrece están:

- Soporta aplicaciones web.
- Genera código para Java y exportación como HTML.
- Fácil de instalar y actualizar.
- Compatibilidad entre ediciones.
- Modelado de Base de Datos: Generación de bases de datos, conversión de diagramas entidad-relación a tablas de base de datos, mapeos de objetos y relaciones, ingeniería inversa desde gestores de bases de datos.

1.10 SGBD. PostgreSQL 8.4.

Existen una gran variedad de gestores de bases de datos entre los que destacan por las ofertas que demandan: Oracle, MySQL y PostgreSQL. PostgreSQL 8.4 constituye la selección para el desarrollo del trabajo en curso esto con el objetivo de estar en sintonía con el gestor de bases de datos definido por el proyecto SIGF y colaborar con la idea nacional de migración hacia software libre.

Es un potente motor de bases de datos, el mismo contiene prestaciones y funcionalidades equivalentes a muchos otros gestores de bases de datos. Este gestor de bases de datos posibilita métodos almacenados, restricciones de integridad y vistas. PostgreSQL es considerado actualmente como uno de los gestores de bases de datos más avanzados en todo el mundo dentro de software libre o código abierto. La versión de PostgreSQL 8.4 contiene una serie de mejoras para propiciar la administración, consulta y programación

de los datos de manera más fácil que las versiones anteriores. Entre las ofertas más populares están los siguientes casos: (PostgreSQL, 2009)

- Restauración de bases de datos en procesos paralelos, que acelera recuperación de un respaldo hasta 8 veces.
- Privilegios por columna, que permiten un control más granular de datos confidenciales.
- Configuración de ordenamiento configurable por base de datos, lo cual hace a PostgreSQL más útil en entornos con múltiples idiomas.
- Nuevas herramientas de monitoreo de consultas que le otorgan a los administradores mayor información sobre la actividad del sistema.

Existen factores negativos los cuales se consideran los mayores inconvenientes de este SGBD entre los cuales se encuentra:

- Consume gran cantidad de recursos.
- Tiene un límite de 8k por fila, aunque se puede aumentar a 32k, con una disminución considerable del rendimiento.

El sistema gestor de bases de datos PostgreSQL trabaja continuamente en el perfeccionamiento de todas sus áreas, incluyendo el área de principales mejoras entre las que destacan:

- Posee una gran escalabilidad. Es capaz de ajustarse al número de máquinas y a la cantidad de memoria que posee el sistema de forma óptima, haciéndole capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta.
- Implementa el uso de transacciones, haciendo su funcionamiento mucho más eficaz, y ofreciendo de esta manera soluciones en campos en los que otros gestores de bases de datos no podrían.
- Posee la capacidad de comprobar la integridad referencial, así como de almacenar procedimientos en la propia base de datos, equiparándolos con los gestores de bases de datos de alto nivel, como está el caso de Oracle.

1.11 Herramientas Administrativas. PgAdmin III 1.12.

PgAdmin III es una herramienta o aplicación gráfica para gestionar el gestor de base de datos PostgreSQL. Esta interfaz soporta todas las características del gestor de bases de datos PostgreSQL y facilita su administración. Esta aplicación está diseñada para

responder a las necesidades de todos los usuarios, desde escribir consultas simples hasta la construcción de bases de datos complejas. La herramienta además incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor, un agente para lanzar scripts programados. La conexión al servidor es realizada mediante conexión TCP/IP y puede llegar a encriptarse mediante SSL para mayor seguridad. Presenta una serie de características como:

- Entradas SQL aleatorias.
- Pantallas de información y ayudas para bases de datos, tablas, índices, secuencias, vistas, programas de arranque, funciones y lenguajes.
- Informes predefinidos en bases de datos, tablas, índices, secuencias, lenguajes y vistas.
- Ayuda a migrar Bases de datos.
- Ayudas para importar y exportar datos.
- Control de revisión con mejoras de la generación de script.
- Preguntas y respuestas para configurar usuarios, grupos y privilegios.

1.12 Conclusiones Parciales.

Luego de haber realizado un estudio de las bases de datos, así como las variadas definiciones y términos que existen con relación a estas, los modelos existentes, las herramientas y tecnologías necesarias para el desarrollo del trabajo en curso, se concluye con la necesidad de diseñar e implementar una base de datos para el módulo de Inspección a Locales de Detección y Centros Penitenciarios en la cual:

Se ha seleccionado el modelo de base de datos relacional puesto que el mismo posibilita la estructuración de datos en forma de relaciones, el mismo es bastante flexible y entendible para el personal de desarrollo.

Debido al papel que brindan los patrones de diseño se decidió utilizar para perfeccionar el diseño de la BD el patrón de llaves subrogadas y el patrón de entidad atributo valor.

El sistema gestor de base de datos a utilizar es PostgreSQL 8.4 utilizado el mismo para la gestión de la base de datos, con el objetivo de seguir la idea planteada por el proyecto SIGF, además de hacer posible la idea de migrar hacia software libre.

La herramienta administrativa a utilizar es PgAdmin III, ya que la misma soporta todas las características del SGBD designado y facilita en gran medida el trabajo de los administradores.

La herramienta para el diseño de la base de datos es Visual Paradigm5.0, ya que la misma ayuda a diseñar aplicaciones de calidad de manera más rápida y con un menor costo.

Capítulo 2 Propuesta de solución

2.1 Introducción.

En el presente capítulo se expondrán los procesos principales que toman lugar en la elaboración y construcción de la BD. Se expone una descripción de la propuesta de solución, se evidencia la arquitectura de datos, así como la utilidad de estándares de nomenclatura y patrones de diseño. Se exhibe el modelo de datos, así como la descripción de las tablas contenidas en la misma. Se argumenta el tratamiento de los índices.

2.2 Requisitos.

En la actualidad los requisitos de software constituyen un punto clave en el desarrollo y producción de aplicaciones informáticas. El hecho de factores perjudiciales como mala gestión de los cambios de los requisitos o requisitos incompletos conlleva a que proyectos de desarrollo ya culminados naden en el fracaso. El Glosario de Terminología Estándar de Ingeniería de Software define a los requisitos como:

- Condición o capacidad que necesita un usuario para dar solución a un problema o alcanzar un objetivo.
- Condición o capacidad que debe ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar, u otro documento.(IEEE, 1990)

Pressman expresó que para alcanzar un buen desarrollo de software, es necesario antes comprender con claridad los requisitos del software. La calidad con que se realiza el proceso de captura de requisitos repercute en gran medida en todo el proceso de desarrollo del software. Una claridad en la definición de los requisitos propicia un mejor soporte en la gestión de cambios y se obtiene una mayor eficiencia en las pruebas, mejorando la calidad y permitiendo la automatización. Contribuye además a tomar grandes y mejores decisiones de diseño y arquitectura. (Pressman, 2005)

A continuación se exponen tanto los requisitos funcionales, como los no funcionales pero solo con el objetivo de brindar entrada al diseño de la base de datos, los mismos están

referenciados en el Trabajo de Diploma titulado “Implementación y prueba del módulo Inspección a Locales de Detección y Centros Penitenciarios”.

2.2.1 Requisitos Funcionales.

Los requisitos funcionales no son más que declaraciones de los servicios que debe proporcionar el sistema, o sea ellos rigen el comportamiento del sistema.

RF1_Crear Plan Trimestral de Inspección a Locales de Detección. El sistema permitirá entrar los datos a la aplicación para la creación de un nuevo Plan Trimestral de Inspección a Locales de Detención.

RF2_Guardar Plan Trimestral de Inspección a Locales de Detección. El sistema permitirá almacenar el Plan Trimestral de Inspección para registrar que locales de detención serán visitados durante el trimestre correspondiente, la fecha en que se realizará dicha inspección, y si se hará o no en conjunto.

RF3_Modificar Plan Trimestral de Inspección a Locales de Detección. El sistema debe permitir modificar el Plan de Inspección a Locales de Detección, para actualizar los campos a los que se deseen cambiar su valor.

RF4_Buscar Plan Trimestral de Inspección a Locales de Detección. El sistema permitirá al usuario buscar un Plan Trimestral de Inspección a Local de Detención determinado de acuerdo a criterios.

RF5_Crear Plan Trimestral de Inspección a Centros Penitenciarios. El sistema permitirá entrar los datos a la aplicación para la creación de un nuevo Plan Trimestral de Inspección a Centros penitenciarios, ya sea Municipal o Provincial.

RF6_Guardar Plan Trimestral de Inspección a Centros Penitenciarios. El sistema permitirá almacenar el Plan Trimestral de Inspección a Centros Penitenciarios.

RF7_Modificar Plan Trimestral de Inspección a Centros Penitenciarios. El sistema debe permitir modificar el Plan de Inspección a Centros Penitenciarios, para actualizar los campos a los que se deseen cambiar su valor.

RF8_Buscar Plan Trimestral de Inspección a Centros Penitenciarios. El sistema debe permitir buscar un Plan de Inspección específico.

RF9_Crear Plan de Objetivos. El sistema permitirá entrar los datos a la aplicación para la creación de un nuevo Plan de Objetivos.

RF10_Guardar Plan de Objetivos. El sistema permitirá el almacenamiento en la aplicación de los objetivos de la inspección a un Centro Penitenciario determinado, las acciones, y los plazos para su realización.

RF11_Modificar Plan de Objetivos. El sistema debe permitir modificar el Plan de Objetivos, para actualizar los campos a los que se deseen cambiar su valor.

RF12_Buscar Plan de Objetivos. El sistema permitirá al usuario la búsqueda de un Plan de Objetivos determinado.

RF13_Crear Auto de Libertad. El sistema permitirá la introducción en la aplicación de los datos siguientes para la creación de un nuevo Auto de Libertad por encontrarse detenidos en locales de detención fuera de los casos previstos de la ley o reclusos que hayan extinguido la sanción.

RF14_Guardar Auto de Libertad. El sistema permitirá guardar el Auto de Libertad para almacenar el documento de forma digital.

RF15_Modificar Auto de Libertad. El sistema permitirá actualizar los campos que requieran modificación en la información almacenada.

RF16_Buscar Auto de Libertad. El sistema debe dar la posibilidad al usuario de buscar un Auto de Libertad según determinados criterios.

RF17_Crear Diligencia de Notificación. El sistema permitirá crear la Diligencia de Notificación del Auto de libertad de la Resolución como constancia de la decisión tomada por el Fiscal.

RF18_Guardar Diligencia de Notificación. El sistema propiciará guardar o almacenar los datos entrados relacionados a Diligencias de Notificación.

RF19_Buscar Diligencia de Notificación. El sistema brindará la posibilidad de realizar búsquedas de diligencias por determinados campos.

RF20_Modificar Diligencia de Notificación. El sistema debe permitir cambiar los datos que ya han sido guardados, actualizando de esta manera la Diligencia de Notificación con la nueva información insertada.

RF21_Crear Documento de Comunicación para la Fiscalía Militar, Autoridad o Tribunal Competente. El sistema debe brindar la posibilidad al cliente de entrar datos referentes a un Documento de Comunicación para su creación.

RF22_Guardar Documento de Comunicación. El sistema permitirá almacenar en la aplicación los datos correspondientes a un Documento de Comunicación.

RF23_Modificar Documento de Comunicación. El sistema debe permitirle al usuario actualizar los datos de un Documento de Comunicación que ya ha sido guardado y de esta manera modificarlo con la nueva información insertada.

RF24_Buscar Documento de Comunicación. El sistema permitirá al usuario buscar un Documento de Comunicación determinado, en un período de tiempo determinado basándose en determinados criterios de búsquedas.

RF25_Crear Documentos de otras Modalidades de Reacción Fiscal. El sistema debe dar la posibilidad al usuario de entrar todos los datos para crear un nuevo Documento de otra Modalidad Fiscal.

RF26_Guardar Documentos de otras Modalidades de Reacción Fiscal. El sistema permitirá almacenar en la aplicación los datos correspondientes a un Documento de otra Modalidad Fiscal.

RF27_Modificar Documentos de otras Modalidades de Reacción Fiscal. El sistema permitirá al usuario modificar los datos de un Documento de otra Modalidad de Reacción Fiscal, actualizando de esta manera el documento con la nueva información insertada.

RF28_Buscar Documentos de otras Modalidades de Reacción Fiscal. El sistema dará la posibilidad de buscar un Documento de otra Modalidad de Reacción Fiscal determinado, utilizando para esto varios criterios de búsqueda.

RF29_Buscar Respuesta al Documento de otra Modalidad de Reacción Fiscal. El sistema dará la posibilidad de buscar una Respuesta al Documento de otra Modalidad de Reacción Fiscal determinado, utilizando para esto varios criterios de búsqueda.

RF30_Crear Resolución. El sistema debe dar la posibilidad al cliente de entrar todos los datos en una plantilla para crear una nueva Resolución.

RF31_Guardar Resolución. El sistema permitirá almacenar en la aplicación los datos correspondientes a una Resolución.

RF32_Modificar Resolución. El sistema debe permitirle al usuario cambiar los datos de una Resolución que ya ha sido guardada en la aplicación, modificando de esta manera dicho documento con los nuevos datos insertados. Se debe tener en cuenta que todos los usuarios no pueden realizar esta funcionalidad.

RF33_Buscar Resolución. El sistema encuentra de existir, el registro deseado, según el criterio de búsqueda insertado.

RF34_Crear Registro de Control Primario de Resoluciones. El sistema debe dar la posibilidad al usuario de entrar todos los datos en una plantilla para crear un nuevo Registro de Control Primario de Resoluciones.

RF35_Guardar Registro de Control Primario de Resoluciones. El sistema permitirá almacenar en la aplicación los datos correspondientes al Registro de Control Primario de Resoluciones.

RF36_Modificar Registro de Control Primario de Resoluciones. El sistema debe permitirle al usuario cambiar los datos del Registro de Control Primario de Resoluciones que ya ha sido guardado en la aplicación, modificando de esta manera dicho documento con los nuevos datos insertados. Se debe tener en cuenta que todos los usuarios no pueden realizar esta funcionalidad.

RF37_Eliminar Registro de Control Primario de Resoluciones. El sistema debe dar la posibilidad al usuario de borrar un registro completo del Registro de Control Primario de Resoluciones especificado por el usuario.

RF38_Buscar Registro de Control Primario de Resoluciones. El sistema encuentra de existir el registro deseado, según los criterios de búsqueda insertados.

RF39_Crear Acta Resultado de la Inspección. El sistema permitirá entrar los datos a la aplicación para la creación de una nueva Acta de Inspección.

RF40_Guardar Acta Resultado de la Inspección. El sistema permitirá almacenar Acta Resultado de la Inspección en la base de datos para su creación.

RF41_Modificar Acta Resultado de la Inspección. El sistema debe permitir modificar un Acta Resultado de la Inspección, para actualizar los campos a los que se deseen cambiar su valor.

RF42_Buscar Acta Resultado de la Inspección. El sistema permitirá al usuario buscar un Acta Resultado de la Inspección, según determinados criterios de búsqueda.

RF43_Crear Registro de Control de Inspecciones a Celdas. El sistema permitirá entrar los datos a la aplicación para la creación de un nuevo Control Primario de Inspecciones a Celdas.

RF44_Guardar Registro Control Primario de Inspecciones a Celdas. El sistema permitirá almacenar Registro Control Primario de Inspecciones a Celdas en la base de datos para su creación.

RF45_Modificar Registro Control Primario de Inspecciones a Celdas. El sistema debe permitir modificar el Registro de Control Primario de Inspecciones a Celdas, para actualizar los campos a los que se deseen cambiar su valor.

RF46_Buscar Registro Control Primario de Inspecciones a Celdas. El sistema permitirá al usuario buscar el Registro de Control Primario de Inspecciones a Celdas basado en determinados criterios de búsquedas.

RF47_Crear Registro de Control de Visitas a Prisión o Campamentos. El sistema deberá permitirle al usuario entrar los datos necesarios para crear un nuevo Registro de Control de Visitas a Prisión o Campamentos.

RF48_Guardar Registro de Control de Visitas a Prisión o Campamentos. El sistema permitirá almacenar el Registro de Control de Visitas a Prisión o Campamentos en la base de datos para su creación.

RF49_Modificar Registro de Control de Visitas a Prisión o Campamentos Primarios. El sistema debe permitir modificar el Registro de Control de las Visitas a Prisión o Campamentos Primarios, para actualizar los campos en los que se desee cambiar su valor.

RF50_Buscar Registros de Control de Visitas a Prisión o Campamentos Primarios. El sistema permitirá al usuario buscar el Registro de Control de Visitas a Prisión o Campamentos Primarios, basado en determinados criterios de búsqueda.

RF51_Crear Resolución por tipo de decisión. El sistema permitirá entrar los siguientes datos a la aplicación para la creación de una nueva Resolución

RF52_Guardar una Resolución por tipo de decisión. El sistema permitirá almacenar los datos de una Resolución por tipo de decisión en la aplicación.

RF53_Modificar una Resolución por tipo de decisión. El sistema debe darle la posibilidad al usuario de modificar los campos que necesiten ser cambiados porque haya algún error en su almacenamiento.

RF54_Buscar Resolución por tipo de decisión. El sistema debe darle la posibilidad al usuario de buscar una Resolución por tipo de decisión, basado en determinados criterios de búsqueda.

RF55_Insertar datos del Plan de Medidas. El sistema permitirá al usuario entrar los datos referentes al Plan de Medidas para su creación.

RF56_Modificar datos insertados del Plan de Medidas. El sistema debe permitirle al usuario cambiar los datos insertados de un Plan de Medidas que ya ha sido guardado en la aplicación, modificando de esta manera dichos datos.

RF57_Guardar datos insertados del Plan de Medidas. El sistema permitirá almacenar en la aplicación los datos correspondientes al Plan de Medidas.

RF58_Buscar Plan de Medidas. El sistema dará la posibilidad de buscar un Plan de Medidas determinado, utilizando para esto determinados criterios de búsqueda.

RF59_Notificar sobre la confección y envío del Plan Trimestral de Inspección a Centros Penitenciarios. El sistema muestra automáticamente una alerta el día 10 del último mes de cada trimestre, notificándole al Fiscal Jefe de Municipio que debe confeccionar y enviar su Plan Trimestral de Inspección al Departamento Provincial CLEP antes del día 15.

RF60_Notificar sobre la confección del Plan Trimestral de Inspección a Centros Penitenciarios. El sistema muestra automáticamente una alerta, notificándole al Fiscal Jefe de Departamento que debe confeccionar el Plan Trimestral de Inspección antes del día 15 del último mes de cada trimestre.

RF61_Notificar sobre el término de la violación del envío del Plan Trimestral de inspección a Centros Penitenciarios. El sistema muestra automáticamente una alerta el día 16 del último mes de cada trimestre, notificándole al Fiscal Jefe de Departamento del incumplimiento del envío por los municipios del Plan de Inspección a Centros Penitenciarios.

RF62_Notificar sobre el término de confección del Acta resultado de la inspección. El sistema muestra automáticamente una alerta, después de 6 días hábiles de concluida la inspección, notificándole al Fiscal CLEP del término de 10 días hábiles para la confección y envío del Acta. (A partir de la conclusión de la inspección).

RF63_Notificar sobre el término violación de envío del Acta resultado de la inspección. El sistema muestra automáticamente una alerta, después de los 11 días hábiles de concluida la inspección, notificándole al Fiscal Jefe de Departamento, al Fiscal

Jefe de Dirección y al Fiscal Jefe Municipal del incumplimiento de la confección y envío del Acta resultado de la Inspección.

RF64_Notificar sobre el término de tramitación de la impugnación. El sistema muestra automáticamente una alerta, notificándole al Fiscal Jefe sobre término de diez días naturales para tramitar la impugnación. (A partir de que se recibe el escrito de impugnación).

RF65_Notificar sobre el término de tramitación de la impugnación. El sistema muestra automáticamente una alerta, después de 6 días hábiles de recibir el escrito de impugnación, notificándole al Fiscal Jefe encargado de la tramitación de la impugnación sobre la culminación del término de diez días naturales para hacer el pronunciamiento correspondiente.

RF66_Notificar sobre el término de confección de Resolución resultado de la inspección. El sistema muestra automáticamente una alerta, después de 6 días hábiles de concluida la inspección, notificándole al Fiscal CLEP del término de 10 días hábiles para la confección y envío de la Resolución. (A partir de la conclusión de la inspección).

RF67_Notificar sobre el término de violación de envío de Resolución o Acta resultado de la inspección. El sistema muestra automáticamente una alerta, después de los 11 días hábiles de concluida la inspección, notificándole al Fiscal Jefe de Departamento, al Fiscal Jefe de Dirección y al Fiscal Jefe Municipal del incumplimiento de la confección y envío de la Resolución resultado de la Inspección.

RF68_Notificar sobre el término de violación del envío del Plan de Medidas por el infractor. El sistema muestra automáticamente al Fiscal Jefe de Departamento, al Fiscal Jefe de Dirección y al Fiscal Jefe Municipal, después de 14 días de notificada la Resolución, una alerta indicando las resoluciones de las que no se ha recibido el Plan de Medidas.

RF69_Generar Plan Trimestral Provincial de Inspección a centros Penitenciarios. El sistema debe permitir, una vez creado al menos un Plan de Inspección a CP, generar un plan provincial con todos los datos de todos los planes municipales de inspección a CP que han sido creados con anterioridad.

RF70_Mostrar reportes. El sistema permite mostrar los resultados de las búsquedas para los diversos reportes.

2.2.2 Requisitos no Funcionales.

Los requisitos no funcionales son criterios o propiedades que deben ser cumplidas por el sistema, estos aseguran disponer de un sistema manejable que brinde funcionalidades requeridas de forma fiable e ininterrumpida.

A continuación se argumentan los requisitos no funcionales vinculados a la BD definidos por el personal de trabajo referente al módulo o subsistema.

RNF de Software

- En cuanto a los requisitos no funcionales de software el Sistema Gestor de BD a utilizar debe ser PostgreSQL 8.4
- La herramienta encargada de gestionar el Sistema Gestor de BD debe ser PgAdmin III.

RNF de Hardware

- El servidor de base de datos debe contener una tarjeta de red, con un microprocesador a 3.00 GHz o superior, en cuanto a Memoria de Acceso Aleatorio (RAM) mínimo 1Gb, capacidad de disco duro superior a los 120Gb, y debe contener además puertos USB.

RNF de Seguridad

- La información contenida en la BD deberá estar resguardada, por medio de una protección de acceso no deseado o autorizado, impidiendo de esta manera alteraciones de los datos.
- En caso de que se originen fallos, debe darse alta prioridad a la recuperación de los datos, esto a través de copias de respaldos.

RNF de Usabilidad

- Siempre se solicitarán los datos por medio de servidores locales. Los tiempos de respuestas del servidor están estimados en 3 segundos.
- Desde cada punto en que se encuentre un servidor local se establecerá una conexión con servidores centrales, esto con el fin de propiciar una buena actualización de los datos.

RNF de Soporte

- Inicialmente la aplicación será instalada en las fiscalías que radican en la capital, es decir en la Habana, donde en la misma se harán estudios previos, para luego esparcir dicha aplicación y hacerla llegar a las demás fiscalías del país.
- Serán los administradores de la base de datos los encargados de asegurar el soporte, creando copias de seguridad en distantes períodos de tiempo, para propiciar una mejor gestión de la información.

2.3 Arquitectura de datos.

Con respecto al tema de la arquitectura de los datos, cada una de las fiscalías existentes (fiscalías generales, provinciales y municipales), presentará un servidor de datos independiente con el propósito de gestionar la información pertinente al mismo. Existirá además un servidor central ubicado en la fiscalía general, el cual se mantendrá en conexión con el resto de los servidores locales, es dicha fiscalía la única autorizada para acceder a la información registrada en cualquier otra fiscalía, tanto las fiscalías provinciales como las municipales solo pueden acceder a revisar su propia información.

Estos servidores se encargarán de proporcionar servicios a las fiscalías a las que pertenecen respectivamente, pero entre ellos establecen conexiones con el objetivo de realizar transferencias de datos. Dicha transferencia se realizará a través de técnicas de replicación de datos, esto consiste en transpolar la información deseada de un lugar a otro por medio de conexiones de red.

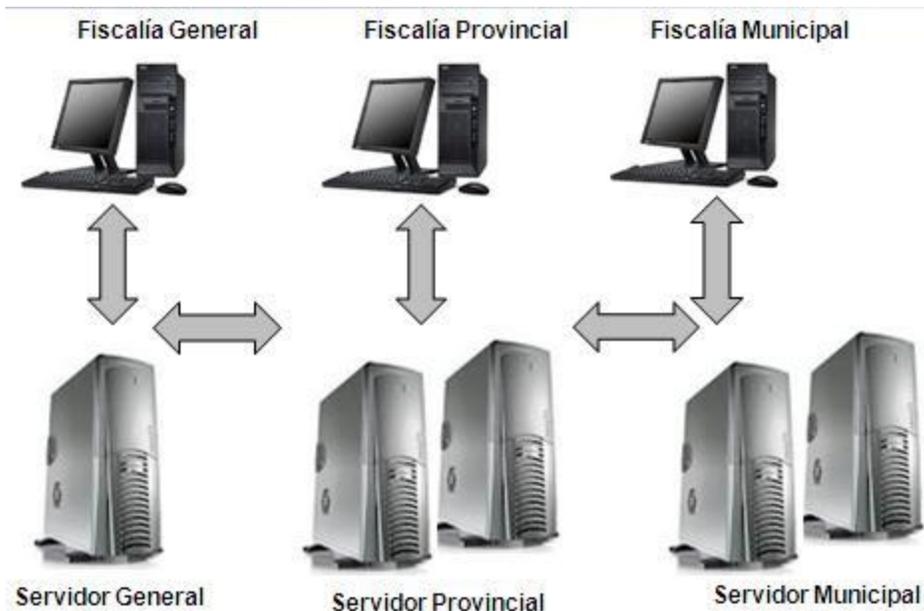


Ilustración 10. Arquitectura de la base de datos.

Existen dos entornos de réplica de datos: uno de ellos es el maestro-esclavo, en el cual existe un maestro, que es el encargado de propagar los cambios y los esclavos, que son los que reciben estos cambios a fin de tener una copia exacta de lo que tiene el maestro. El otro entorno es el multi-maestro, en el cual son habilitados los datos para ser distribuidos en un grupo, y estos datos pueden ser actualizados por cualquier miembro del grupo. El Sistema de Informatización de la Gestión en las Fiscalías utilizará el entorno de réplica multi-maestro.

La replicación de dichos datos será realizada en ambos sentidos es decir que las fiscalías municipales replican datos hacia las provinciales y estas hacia la general, y el proceso se podrá realizar en el orden inverso. Dicho proceso asegura un mejor resguardo de los datos, ya que en caso de sufrir alteraciones o pérdidas en una de las fiscalías, se pudiera obtener la información previamente guardada en otro nivel de fiscalías.

La herramienta encargada de propiciar el proceso de replicación de datos es Reko, el cual es un software de réplica de datos que cubre una serie de necesidades de gestores de BD como PostgreSQL, como la protección, recuperación, sincronización de datos, transferencia de datos entre diversas localizaciones. La misma fue diseñada para posibilitar el proceso de replicación de datos tanto con conexión de red y sin ella.

2.4 Características generales de la base de datos del módulo ILDCP.

La base de datos del módulo Inspección a Locales de Detección y Centros Penitenciarios formará parte de un esquema único compuesto por 8 esquemas referentes a los procesos originados en el proyecto SIGF. Constituye una base de datos relacional y distribuida ya que formará parte de varios servidores de BD como: fiscalías generales, provinciales y municipales. Dicha BD está vinculada al esquema CLEP, que es el encargado de atender el módulo ILDCP. Dicho esquema presenta un total de 37 tablas de las cuales 5 son nomencladores.

2.5 Diseño de la BD.

2.5.1 Estándares de nomenclatura.

Hay ciertos puntos a tener en cuenta con relación a la nomenclatura de la BD. Por ejemplo:

- No es permitido el uso de tildes, ni espacios entre palabras.
- Se modifica la letra “ñ” por “nn”.
- Los nombres serán escritos en minúsculas, con un límite máximo de 30 caracteres, comenzando siempre con una letra, nunca con números ni caracteres especiales.
- Se debe limitar el uso de números, guiones bajos, y abreviaciones a solo el uso indispensable.

	Nomenclatura
Tablas de la BD	“<nombre del esquema único de la BD + ‘.’ + ‘tb’ + nombre de la tabla + ‘clep’>”
Nomencladores	“<nombre del esquema único de la BD + ‘.’ + ‘n’ + nombre de la tabla + ‘clep’>”
Llaves primarias	“<id + nombre de la tabla>”
Llaves foráneas	“<id + nombre de la tabla referenciada>”
Restricción de llaves primarias	“<nombre de la tabla + ‘_’ + pkey>”
Restricción de	“<nombre de la tabla referenciada + ‘_’ + fk>”

llaves foráneas	
Índices	“<nombre de la tabla + atributo indexado + ‘index’>”

Ilustración 11. Descripción de los estándares de nomenclatura.

2.5.2 Patrones de diseño.

Los patrones de diseño de BD constituyen una guía para el desarrollo de problemas durante la fase de diseño de BD. Los mismos permiten a los usuarios crear una BD más fortalecida. (Blaha, 2010). Durante la realización del modelo de datos se recurrió al uso de varios patrones de diseño que se evidencian a continuación.

En la base de datos del subsistema CLEP están contenidas 5 tablas nomencladoras, estas son: nviolacion, nmodalidad, nobjetivo, nvisitasconjuntas, ninfractores.

El hecho de ser tablas nomencladoras indica que las mismas contienen todos los estados o posibles valores en los que puede quedar expresado un determinado atributo o elemento. Ejemplo claro de esto es el caso específico de la tabla nomencladora:

tbdocumento. La misma posee un identificador llamado “iddocumento”, el cual es un identificador auto-incremental (patrón de llave o clave subrogada), esto se realiza con el objetivo de generar una llave primaria única para dicha entidad en vez de usar un atributo identificador en el contexto dado.



Ilustración 12. Ejemplo del patrón llave subrogada.

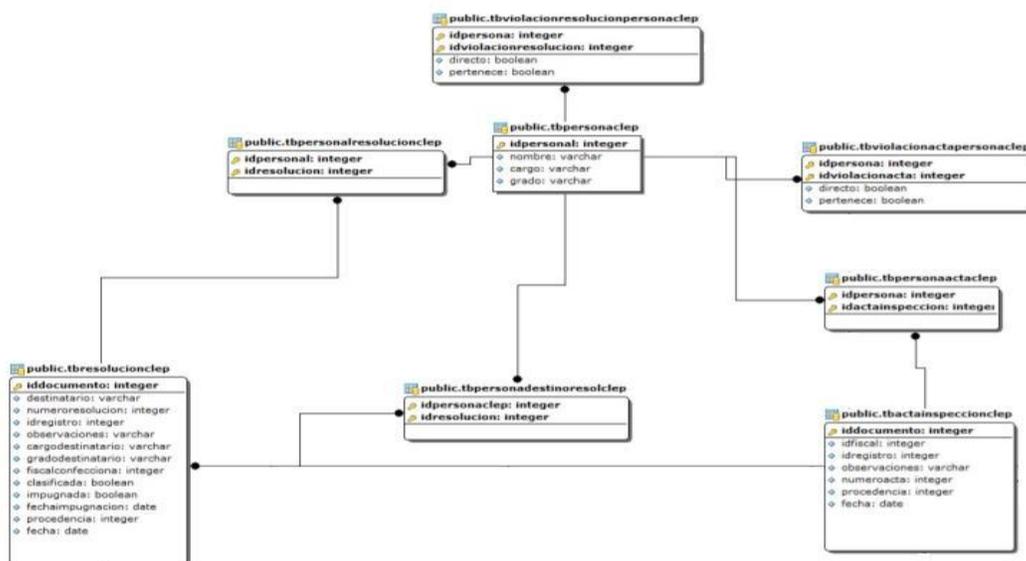


Ilustración 14. Relaciones de la tabla tbpersonaclep.

Nombre: tbpersonaclep		
Descripción: En dicha tabla se almacenan los datos que guardan relación con el personal.		
Atributos	Tipo	Descripción
idpersonal	Integer	Atributo identificador de la persona.
Nombre	Varchar	Nombre de la persona.
Cargo	Varchar	Cargo de la persona.
Grado	Varchar	Grado de la persona.

Tabla 1. Descripción de la tabla tbpersonaclep.

Nombre: tbviolacionresolucionpersonaclep		
Descripción: En dicha tabla se almacenan los datos que guardan relación con el personal y la violación de resolución.		
Atributos	Tipo	Descripción
idpersona	Integer	Atributo identificador de la persona.
Idviolaciónresolucion	Integer	Atributo identificador de la violación de resolución.
Directo	Boolean	Determina si lo atiende el personal directo o no.
pertenece	Boolean	Determina si pertenece o no a la fiscalía.

Tabla 2. Descripción de la tabla tbviolacionresolucionpersonaclep.

Nombre: tbpersonalresolucionclep		
----------------------------------	--	--

Descripción: En dicha tabla se almacenan los datos que guardan relación con el personal y la resolución.

Atributos	Tipo	Descripción
idpersona	Integer	Atributo identificador de la persona.
idresolucion	Integer	Atributo identificador de la resolución.

Tabla 3. Descripción de la tabla tbpersonalreolucionclep.

Nombre: tbviolacionactapersonaclep

Descripción: En dicha tabla se almacenan los datos que guardan relación con el personal y el acta de violación.

Atributos	Tipo	Descripción
idpersona	Integer	Atributo identificador de la persona.
idviolacionacta	Integer	Atributo identificador del acta de violación.
Directo	Boolean	Determina si lo atiende el personal directo o no.
Pertenece	Boolean	Determina si pertenece o no a la fiscalía.

Tabla 4. Descripción de la tabla tbviolacionactapersonaclep.

Nombre: tbpersonaactaclep

Descripción: En dicha tabla se almacenan los datos que guardan relación con el personal y el acta de inspección.

Atributos	Tipo	Descripción
idpersona	Integer	Atributo identificador de la persona.
idactainspeccion	Integer	Atributo identificador del acta de inspección.

Tabla 5. Descripción de la tabla tbpersonaactaclep.

Nombre: tbpersonadestinosolclep

Descripción: En dicha tabla se almacenan los datos que guardan relación con el personal y la resolución.

Atributos	Tipo	Descripción
idpersona	Integer	Atributo identificador de la persona.
idresolucion	Integer	Atributo identificador de la resolución.

Tabla 6. Descripción de la tabla tbpersonadestinosolclep.

Nombre: tbresolucionclep

Descripción: En dicha tabla se almacenan los datos que guardan relación con la resolución.

Atributos	Tipo	Descripción
iddocumento	integer	Atributo identificador de la persona.
destinatario	varchar	Destinatario de la resolución.
numeroresolucion	integer	Número de la resolución.
idregistro	integer	Atributo identificador del registro.
observaciones	varchar	Observaciones elaboradas.
cargodestinatario	varchar	Cargo del personal a quien va destinado.
fiscalconfecciona	varchar	Fiscal que confecciona la resolución.
clasificada	boolean	Plantea si la información es o no clasificada.
impugnada	boolean	Plantea si es o no impugnada la resolución.
fechaimpugnacion	date	Fecha de impugnación de la resolución.
procedencia	integer	Procedencia de la resolución.
fecha	date	Fecha de la resolución.

Tabla 7. Descripción de la tabla tbresolucionclep.

Nombre: tbactainspeccionclep		
Descripción: En dicha tabla se almacenan los datos que guardan relación con las actas de inspección.		
Atributos	Tipo	Descripción
iddocumento	integer	Atributo identificador de la persona.
Idfiscal	Integer	Atributo identificador del fiscal.
Idregistro	integer	Atributo identificador del registro.
Observaciones	varchar	Observaciones elaboradas.
Numeroacta	integer	Número del acta.
Procedencia	integer	Procedencia de la resolución.
fecha	date	Fecha de la resolución.

Tabla 8. Descripción de la tabla tbactainspeccionclep.

Principales relaciones de la tabla nomencladora nviolacionclep.

En la figura que se expone a continuación se presencia las relaciones que posee la tabla nomencladora nviolacionclep, la misma encargada del control de las violaciones que transcurren en los procesos.

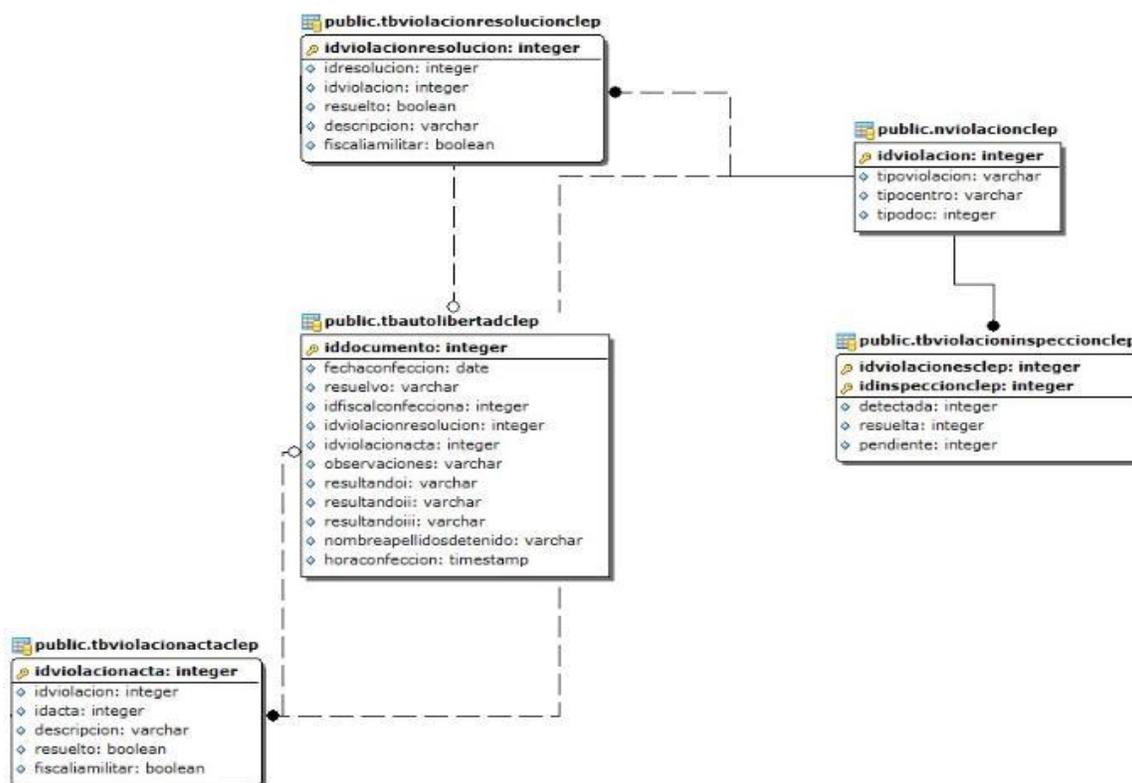


Ilustración 15. Relaciones de la tabla nomencladora nviolacionclep.

Nombre: tbviolacionresolucionclep		
Descripción: En dicha tabla se almacenan los datos que guardan relación con las resoluciones de violación.		
Atributos	Tipo	Descripción
idviolacionresolucion	Integer	Atributo identificador de las resoluciones de violación.
idresolucion	Integer	Atributo identificador de la resolución.
idviolacion	Integer	Atributo identificador de la violación.
Resuelto	boolean	Plantea si se está resuelta o no la resolución de violación.
descripcion	Varchar	Descripción de la violación.
fiscaliamilitar	boolean	Si está o no en la fiscalía militar.

Tabla 9. Descripción de la tabla tbviolacionresolucionclep.

Nombre: tbautolibertadclep		
----------------------------	--	--

Descripción: En dicha tabla se almacenan los datos que guardan relación con los autos de libertad.

Atributos	Tipo	Descripción
iddocumento	Integer	Atributo identificador de los autos de libertad.
fechaconfeccion	Date	Fecha de confección del documento.
resuelvo	Varchar	Atributo que plantea lo que se resuelve.
idfiscalconfecciona	Integer	Identificador del fiscal que confecciona el documento.
idviolacionresolucion	Integer	Identificador de la violación de la resolución.
idviolacionacta	Integer	Identificador del acta de violación.
observaciones	Varchar	Observaciones que se toman.
resultadoi	Varchar	Primer resultado obtenido.
resultadoii	Varchar	Segundo resultado obtenido.
resultadoiii	Varchar	Tercer resultado obtenido.
nombreakellidosdetenido	Varchar	Nombre y apellidos del detenido.
horaconfeccion	timestamp	Hora de confección del documento.

Tabla 10. Descripción de la tabla tbautolibertadclep.

Nombre: tbviolacionactaclep

Descripción: En dicha tabla se almacenan los datos que guardan relación con las actas de violación.

Atributos	Tipo	Descripción
idviolacionacta	integer	Atributo identificador del acta de violación.
idviolacion	integer	Atributo identificador de la violación.
idacta	integer	Atributo identificador del acta.
descripcion	varchar	Descripción del acta de violación.
resuelto	boolean	Plantea si está o no resuelta el acta.
fiscaliamilitar	boolean	Si está o no en la fiscalía militar.

Tabla 11. Descripción de la tabla tbviolacionactaclep.

Nombre: nviolacionclep

Descripción: En dicha tabla se almacenan los datos que guardan relación con las violaciones.

Atributos	Tipo	Descripción
-----------	------	-------------

idviolacion	integer	Atributo identificador de la violación.
tipoviolicion	varchar	Atributo que indica el tipo de violación.
tipocentro	varchar	Atributo que indica el tipo de centro.
tipodoc	integer	Atributo que indica el tipo de documento.

Tabla 12. Descripción de la tabla nviolacionclep.

Nombre: tbviolacioninspeccionclep		
Descripción: En dicha tabla se almacenan los datos que guardan relación con las violaciones presenciadas en las inspecciones.		
Atributos	Tipo	Descripción
idviolacion	integer	Atributo identificador de la violación.
idinspeccion	integer	Atributo identificador de la inspección.
detectada	integer	Cantidad de violaciones detectadas.
resuelta	integer	Cantidad de violaciones resueltas.
pendiente	integer	Cantidad de violaciones pendientes.

Tabla 13. Descripción de la tabla tbviolacioninspeccionclep.

2.6 Uso de índices.

Los índices empleados en una base de datos no son más que estructuras de datos que mejoran drásticamente la velocidad de las operaciones, dando un acelerado paso de acceso a determinados registros de una tabla de base de datos. Los mismos juegan presencia sobre aquellos elementos a los cuales se suele frecuentar durante las variadas búsquedas.

Los mismos pueden ser diseñados utilizando columnas de la base de datos, posibilitando el acceso rápido a los registros. Dichos índices son bastante similares en funcionamiento al índice de un libro, guardando y conservando parejas de elementos, uno es el elemento que se desea indexar y el otro la posición que ocupa en la base de datos. El espacio requerido para el proceso de almacenamiento de estos índices es menor que el espacio estimado de almacenamiento de la tabla, ya que los mismos contienen solamente los campos claves de acuerdo a lo que se desee, excluyen el resto de la información de la tabla, es decir que los mismos son solamente una porción de una tabla determinada.

Con relación al tratamiento de los índices se debe tener presente una serie de factores como: (González, 2009)

- Los índices deben crearse sobre campos con valores únicos, pues esto posibilita su mejor funcionamiento.
- En las consultas en las cuales se enmarcan cláusulas “where” o “join”, se debe indexar sobre los atributos que generan mayor frecuencia en los procesos de búsquedas, esto para mejorar dichas consultas con operadores “select”.
- Se debe realizar solo el uso indispensable de índices pues los mismos se actualizan con cada cambio que sufre la tabla vinculada y esto puede propiciar que el proceso de modificación se ralentice.

Entre los distintos tipos de índices que se pueden aplicar están los siguientes casos:

- B-tree: se utiliza para localizar un único valor o con el objetivo de investigar en un área de distribución las búsquedas de los valores claves con el empleo de operadores como: <, <=, =, >=, >. EL mismo cobra funcionamiento sobre datos numéricos, así como textos. Los mismos brindan utilidad para evitar las grandes operaciones de ordenación. (González, 2009)
- Hash: La operación principal que soporta de manera eficiente es la búsqueda. Las tablas hash almacenan la información en posiciones pseudo-aleatorias, así que el acceso ordenado a su contenido es bastante lento. Este tipo de índice solo soporta o puede ser usado para consultas de equivalencia con el operador “=”. (González, 2009)
- Gist: Generalized Search Tree (árbol de búsqueda generalizada), constituyen una infraestructura dentro de la cual muchas estrategias diferentes de indexación se pueden aplicar. Los operadores que utiliza la distribución PostgreSQL son: <<, &<, &>, >>, <<|, &<|, |&>, |>>, @>, <@, ~=, &&, los mismos solo actúan sobre datos geométricos. Se utilizan para aumentar la velocidad de búsqueda en todos los tipos de estructuras de datos irregulares. (González, 2009)
- Gin: Generalized Inverted Index (Índice Invertido Generalizado). Los mismos brindan la posibilidad de tratar los valores que poseen más de una clave. Soporta varias estrategias de indexación definidas por los usuarios. La distribución

PostgreSQL admite consultas utilizando operadores como: <@, @>, =, &&, los mismos constituyen operadores para arreglos de datos. (González, 2009)

Los índices utilizados en la base de datos del subsistema o esquema CLEP son los índices de tipo B-tree, ya que el mismo presenta determinadas ventajas en relación a la estructura en que se basa, esto indica que en los procesos de búsquedas no se hace necesario recorrer toda la tabla, sino que se basa en estructuras de árboles, comenzando en la raíz y recorriendo las ramas hasta encontrar el elemento deseado. Un ejemplo claro de imposición o creación de índice de este tipo son los siguientes:

Subsistema	Tabla	Atributo	Índice
CLEP	planinspeccion	confecciona	planinspeccion_confecciona_index
CLEP	planinspeccion	aprueba	planinspeccion_aprueba_index
CLEP	registroinspeccion	fiscalconfecciona	registroinspeccion_fiscalconfecciona_index
CLEP	regresoluciones	tipoinfractor	regresoluciones_tipoinfractor_index

Tabla 14. Definición de índices.

Estas columnas frecuentan ser utilizadas en determinadas consultas, por lo que se realizaron un grupo de pruebas con el propósito de ver los tiempos de respuestas de de las mismas y a raíz de los resultados arrojados se definieron dichos índices.

2.6.1 Resultados de las pruebas de los índices.

Características generales:

Número de tuplas por tablas: 1000.

Cantidad de test realizados: 5.

A continuación se exponen los indicadores a tener en cuenta en la realización de las pruebas.

Cost	Intervalo de tiempo en el que recoge (A..B): A: Costo inicial estimado (tiempo que se invierte antes del inicio del escaneo). B: Costo total estimado.
Rows	Número estimado de filas de salida
Actual time	Intervalo real de tiempo (milisegundos) empleado en la consulta.

Rows	Cantidad de filas devueltas.
Loops	Número de ciclos realizados en la consulta.
Total runtime (X)	Tiempo de ejecución de la consulta.

Tabla 15. Definición de los indicadores de prueba.

	(Cost; Rows)	(Actual time; Rows; Loops)	Total runtime ()
Resultados previos	(0.00..32.20; 12)	(0.021..0.720; 12; 1)	0.912 ms
Resultados posteriores	(0.00..3.52; 12)	(0.033..0.033; 12; 1)	0.210 ms

Tabla 16. Índice (planinspeccion_confecciona_index).

	(Cost; Rows)	(Actual time; Rows; Loops)	Total runtime ()
Resultados previos	(0.00..34.30; 10)	(0.026..0.782; 10; 1)	0.903 ms
Resultados posteriores	(0.00..3.12; 10)	(0.027..0.027; 10; 1)	0.201 ms

Tabla 17. Índice (planinspeccion_aprueba_index).

	(Cost; Rows)	(Actual time; Rows; Loops)	Total runtime ()
Resultados previos	(0.00..42.50; 34)	(0.030..0.882; 34; 1)	0.946 ms
Resultados posteriores	(0.00..4.27; 34)	(0.041...0.041; 34; 1)	0.315 ms

Tabla 18. Índice (registroinspeccion_fiscalconfecciona_index).

	(Cost; Rows)	(Actual time; Rows; Loops)	Total runtime ()
Resultadosprevios	(0.00..36.70; 20)	(0.045..0.793; 20; 1)	0.928 ms
Resultados posteriores	(0.00..4.12; 20)	(0.037..0.037; 20; 1)	0.304 ms

Tabla 19. Índice (regresoluciones_tipoinfractor_index).

La serie de índices propuestos acelera el proceso de búsquedas en determinadas tablas que por su importancia se hace necesario conocer o localizar un determinado elemento. Ejemplo de esto se evidencia en las tablas planinspeccion y registroinspeccion, en la cual se indexa en ambos casos por el atributo “confecciona”, el cual indica el fiscal responsable de confeccionar dichos documentos, el mismo posee un grupo cerrado de fiscales, lo que posibilita que las búsquedas sean más flexibles y con mejores resultados, parecido ocurre con el otro atributo por el cual se indexa en la tabla planinspeccion que es “aprueba”, que indica el fiscal que aprueba dicho plan.

2.7 Conclusiones parciales.

Con el desarrollo de este capítulo se resumieron determinados aspectos con relación al diseño de la BD, se ejemplificó el modo en que interactúan los datos, así como aspectos esenciales que presenta la propuesta que se expone.

Con el objetivo de elaborar el modelado de datos se utiliza la herramienta Visual Paradigm en su versión 5.0, así como el SGBD PostgreSQL en su versión 8.4 para garantizar la gestión de los datos, esto apoyado en la utilidad de la herramienta PgAdmin III.

Durante el proceso de diseño quedaron definidos determinados estándares de nomenclatura, así como se evidenció la utilidad de patrones de llaves subrogadas en cual está presente en cada una de las tablas del modelo de datos, y el patrón modelo entidad-atributo-valor, para representar determinadas entidades con sus atributos, es decir desglosando estas tablas para un mejor entendimiento.

Se describen determinadas tablas del modelo de datos, sobre todo aquellas que cobran mayor peso en el funcionamiento del módulo que atienden. Se fundamenta el tratamiento de los índices quedando evidenciada la razón de sus usos a la hora de realizar consultas a la BD, mostrando determinados ejemplos de los mismos, así como pruebas realizadas a estos.

Capítulo 3 Validación de la solución propuesta

3.1 Introducción.

En el presente capítulo tendrá lugar el desarrollo del proceso de validación del diseño e implementación de la base de datos del módulo Inspección a Locales de Detección y Centros Penitenciarios, abordando elementos de gran importancia e interés como la integridad, redundancia, normalización y seguridad de la base de datos. Para contribuir al desarrollo del proceso de validación se realizan pruebas de carga y rendimiento centradas en la idea de comprobar el correcto funcionamiento de la base de datos, así como el rendimiento de la aplicación.

3.2 Validación teórica del diseño.

El acertado diseño de una base de datos constituye un elemento crucial en el desarrollo de un sistema, debido a esto se hace imprescindible tener en cuenta determinados aspectos que propicien la eficiencia y un buen diseño de la base de datos.

Los aspectos que se dan lugar en el desarrollo de este epígrafe guardan relación con la integridad, privacidad y seguridad de los datos. Los mismos certifican el acceso a los datos por el personal autorizado y que la información no se tergiverse como resultado de acciones no controladas.

3.2.1 Integridad de los datos.

La integridad de los datos se fundamenta en determinadas normas que son utilizadas por las bases de datos relacionales. Estas reglas adoptadas en el trabajo se ejemplifican a continuación.

- Integridad de entidad

Este tipo de regla guarda dependencia con las llaves primarias de las tablas, y como cada entidad debe contemplar una llave primaria queda evidenciada la objetividad de dicha

regla, además establece que ningún elemento que sea pieza de la llave primaria puede admitir valores nulos.

En la propuesta de solución que se imparte, todas las llaves primarias han quedado expresadas en cada tabla y ninguna contenida admite valores nulos o repetidos. Esto queda verificado a través del uso de restricciones de llaves primarias posibilitando que no se obtenga información repetida para un valor de clave. Ejemplo de la misma queda evidenciado en la tabla `tbdocumentoclep`, el atributo identificador es `iddocumento`.

- Integridad referencial

Debido a la integridad referencial es que se asegura la integridad entre las llaves foráneas y las llaves primarias, es decir que la clave externa o foránea de una tabla de referencia siempre debe aludir a una fila válida de la tabla a la que haga referencia. Dicha integridad promueve que la relación entre dos tablas permanezca sincronizada durante la puesta en marcha de diferentes operaciones como la actualización y eliminación de datos, esto implica que se evidencien datos correctos, sin repeticiones indebidas, sin información perdida ni relaciones incorrectamente resueltas.

En la propuesta de solución que se imparte, dicha regla se hace visible, debido a que para cada valor de llave foránea existe un valor de llave primaria coherente. Un ejemplo de esto se manifiesta en la tabla `tbviolacionresolucionclep`, la misma contiene como llave foránea `idresolucion`, dicho atributo es llave primaria en la tabla referenciada `tbresolucionclep`.

- Integridad clave

Dicha regla plantea que la clave o llave primaria no puede contener valores iguales en tuplas desiguales. En la propuesta de solución que se imparte se avala que las llaves primarias no pueden quedar repetidas, debido a que las mismas son numéricas, procreándose auto incrementalmente.

3.2.2 Normalización de la BD.

En la propuesta de solución planteada un argumento relevante que se tuvo presente durante el estudio y desarrollo del trabajo, fue la normalización de la base de datos. La normalización es aquel proceso que consiste en organizar los datos de una BD. En la misma se encuentran incluidas la creación de tablas así como el establecimiento de relaciones entra dichas tablas propiciando protección a los datos, y haciendo que dicha BD sea más flexible al ser eliminadas las redundancias, así como las dependencias incoherentes.

En el proceso de normalización existen reglas denominadas estas como formas normales. La propuesta de base de datos diseñada para dar solución al trabajo en curso se encuentra normalizada en tercera Forma Normal, para esto tuvo que cumplir con determinados parámetros, tales como:

Primera forma normal:

- Eliminando los grupos repetidos de las tablas individuales.
- Creando una tabla independiente para cada conjunto de datos relacionados.
- Identificando cada conjunto de datos relacionados con una clave principal.

Esta regla la cumple la base de datos ya que todos los elementos contenidos en cada una de las tablas son atómicos.

Segunda forma normal:

- Creando tablas independientes para conjuntos de valores que se apliquen a varios registros.
- Relacionando estas tablas con una clave externa.

Esta regla es cumplida por la base de datos ya que además de cumplir la primera regla o primera forma normal, los elementos de las tablas con llaves compuestas no presentan dependencias funcionales parciales.

Tercera forma normal:

- Eliminando los campos que no dependan de la clave o llave primaria.

Esta regla es cumplida por la base de datos ya que además de cumplir con las reglas anteriores, todos los elementos contenidos en las tablas dependen directamente de la llave o clave primaria.

Se exponen a continuación algunos fragmentos de la base de datos donde se pone en evidencia que la misma se encuentra normalizada en tercera forma normal.

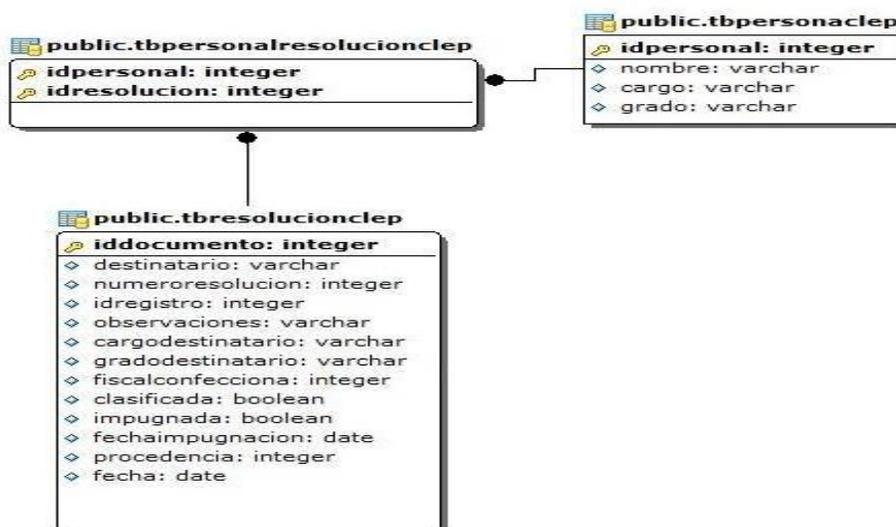


Ilustración 16. Ejemplo 1.

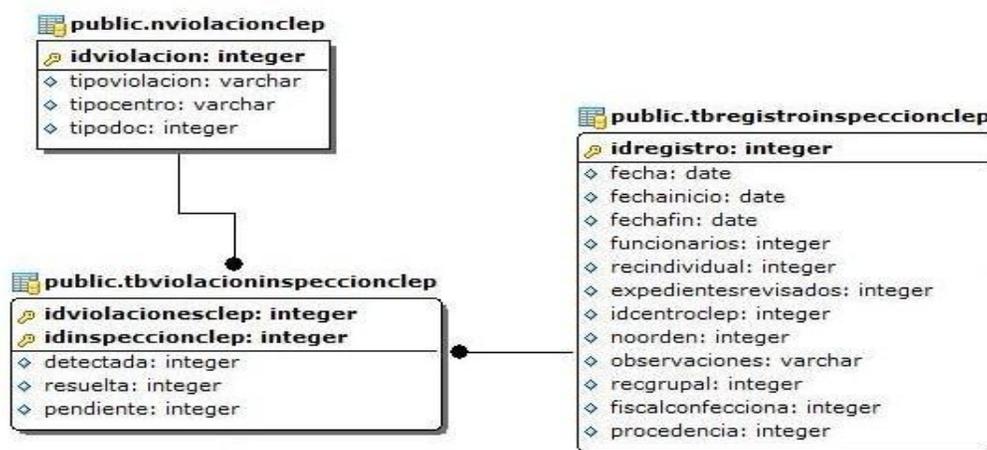


Ilustración 17. Ejemplo 2.

3.2.3 Análisis de la seguridad de la base de datos.

Las bases de datos actuales sirven de almacén a todo tipo de información, debido a que las mismas pueden ser tildadas de sensibles en relación a los datos que guarda, se hace necesario meditar la forma de protegerlas.

La seguridad en las bases de datos significa la capacidad de los usuarios para acceder y cambiar los datos de acuerdo a las políticas del negocio, es decir que son las acciones a tener en cuenta para propiciar que las consultas realizadas a la información que se almacena se haga solo por los usuarios autorizados.

En la propuesta de solución que se brinda están contenidos determinados niveles de seguridad, o más bien dos niveles. El primero de estos roles es el administrador el cual es el propietario de la BD, el mismo posee permiso sin ningún tipo de restricción para gestionar la BD, el otro rol es el de los usuarios, los cuales solo poseen permisos de lectura y escritura sobre las tablas de la BD. A continuación se enumeran una serie de normas que hacen posible una eficiente seguridad de la información como son:

- Se le realiza copias de respaldo diariamente a la base de datos.
- Se encuentra instalado y configurado los servidores de seguridad.
- Las contraseñas solo pueden ser alterados por el administrador de la base de datos.
- Todo intento de conexión con la base de datos queda registrado.
- Las contraseñas deberían al menos contener un carácter no alfabético.

La estrategia de copias de respaldo se realiza siguiendo una serie de pasos. Las mismas se realizan de forma diaria. A nivel general, es decir en la fiscalía general se realiza una salva semanal total de la BD, y una mensual para sistemas en cinta. Para el desarrollo de las mismas se ejecuta un script, el cual genera un back-up de la BD, dicho script es ejecutado desde la función “crontab”. Dicha función es un administrador regular de procesos que se encarga de ejecutar procesos o guiones en determinados intervalos de tiempo. En el mismo se define el tiempo exacto (minuto, hora, día del mes, mes, día de la semana) en que se desea ejecutar un determinado fichero o script.

3.3 Validación Funcional.

Para validar el rendimiento de la base de datos que se propone, así como su correcto funcionamiento se hace necesario efectuar determinadas pruebas. Dichas pruebas conforman el objetivo de simular cargas de producción real y prestar atención a como procede la base de datos ante esta carga, verificando así que el sistema satisface las necesidades existentes, esto sin violar la integridad de los datos.

3.3.1 Prueba de rendimiento.

Las pruebas de rendimiento son aspectos esenciales en el desarrollo de una aplicación, pues a raíz de ellas se llega a conocer la manera en la que se manifiesta el sistema frente a determinados ambientes con la intención de revelar algún tipo de grietas que infrinjan contra la calidad. Estas pruebas se emplean para determinados fines:

- Pueden demostrar que el sistema cumple con las normas de rendimiento.
- Pueden comparar dos sistemas para percibir cuál de ellos posee un mejor funcionamiento.
- Pueden medir qué partes del sistema inducen a un mal funcionamiento del conjunto.

Se hace imprescindible conocer que los resultados de las pruebas y como tal el rendimiento del sistema está vinculado en gran medida al hardware del cómputo que ejecutara el oficio de servidor. Los servidores definidos para dar soporte a dicha BD poseen una tarjeta de red, con un microprocesador a 3.00 GHz o superior, con un mínimo de memoria RAM de 1Gb, los discos duros poseen una capacidad superior a 120Gb.

Quedan plasmadas en el documento algunas de las pruebas de rendimiento realizadas al sistema.

3.3.2 Prueba de volumen.

Las pruebas de volumen son una de las pruebas realizadas con el fin de medir el rendimiento, está catalogada entre las más sencillas. Esta prueba consiste en estudiar la conducta de la base de datos verificando si la misma obtiene su límite de

almacenamiento y pueda provocar fallas. La puesta en marcha de esta prueba de rendimiento en la propuesta de solución, se manejó con el uso de la herramienta Data Generator 2005 para PostgreSQL con el fin de alcanzar el llenado voluminoso de la base de datos.

El desarrollo de esta prueba se llevo a cabo en una máquina con un microprocesador Intel Core 2 Duo a 2.20 GHz y 1Gb de memoria RAM, estas propiedades son inferiores a las que presentan los servidores que habitan en cada una de las fiscalías, por tanto el desarrollo de estas pruebas en dichos servidores ofrecerá mejores resultados.

En la siguiente tabla se arrojan los resultados de la prueba de volumen aplicada a determinadas tablas escogidas del esquema clep de la base de datos actual de SIGF.

Tabla	Número de tuplas	Tiempo	Errores
public.tbdocumentoclep	200000	6 min 27 seg	0
public.tbpersonaclep	200000	6 min 5 seg	0

Tabla 20. Datos generados en la base de datos clep.

En estos ejemplos de tablas contenidas en la BD se generaron datos con un rango mayor al que tendrá la base de datos en un período inferior al año, más específicamente entre 1 y 2 años. Como se muestra en la tabla 19, las tablas que mayor cantidad de tuplas aceptaron fueron tbdocumentoclep, y tbpersonaclep con 200 mil registros cada una, en un tiempo la primera de 6 minutos, 27 segundos, y la segunda en un tiempo de 6 minutos, 5 segundos. Una vez realizada la inserción de esta serie de datos no se evidenciaron fallas con respecto al límite de capacidad, así como desbordamientos de atributos. En ambos casos esta cantidad de datos solo ocupa una pequeña porción de los 120Gb estimados para el servidor, por tanto no representa preocupaciones el generar esa cantidad de información. Los resultados ofrecidos evidencian que tanto el diseño de la BD, como el gestor PostgreSQL toleran el conjunto de información que será fijada cuando la BD sea exportada a cada una de las fiscalías del estado cubano. Los resultados ofrecidos se muestran en las ilustraciones 18 y 19.

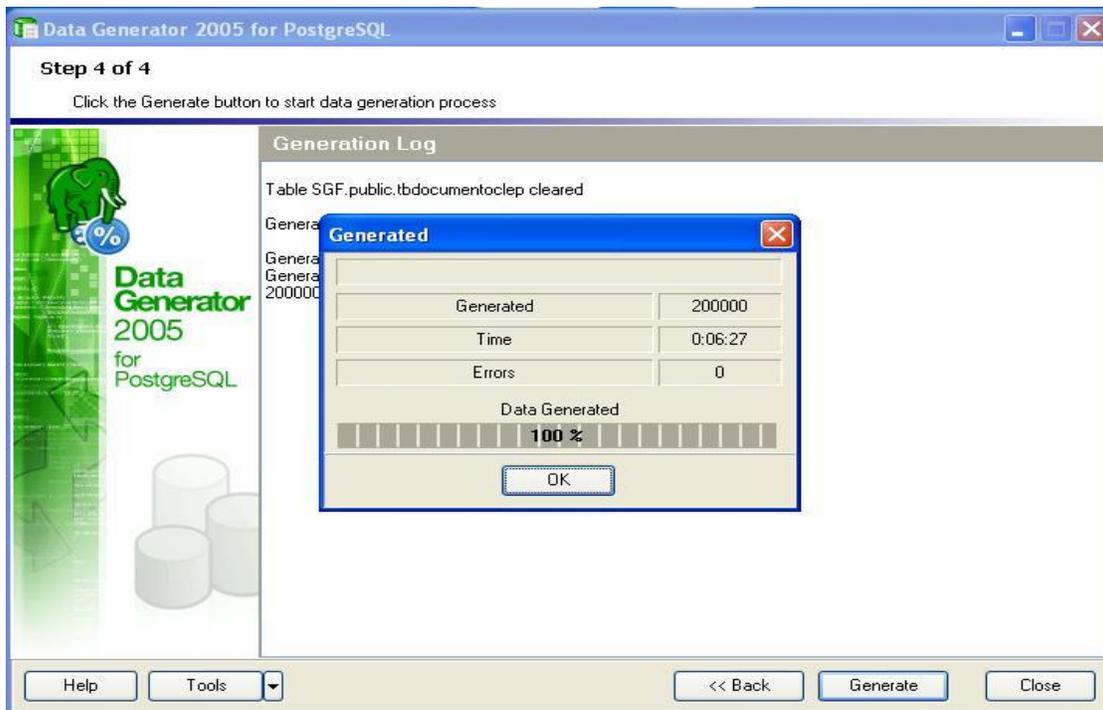


Ilustración 18. Resultados de la generación de datos de tbdocumentoclep.

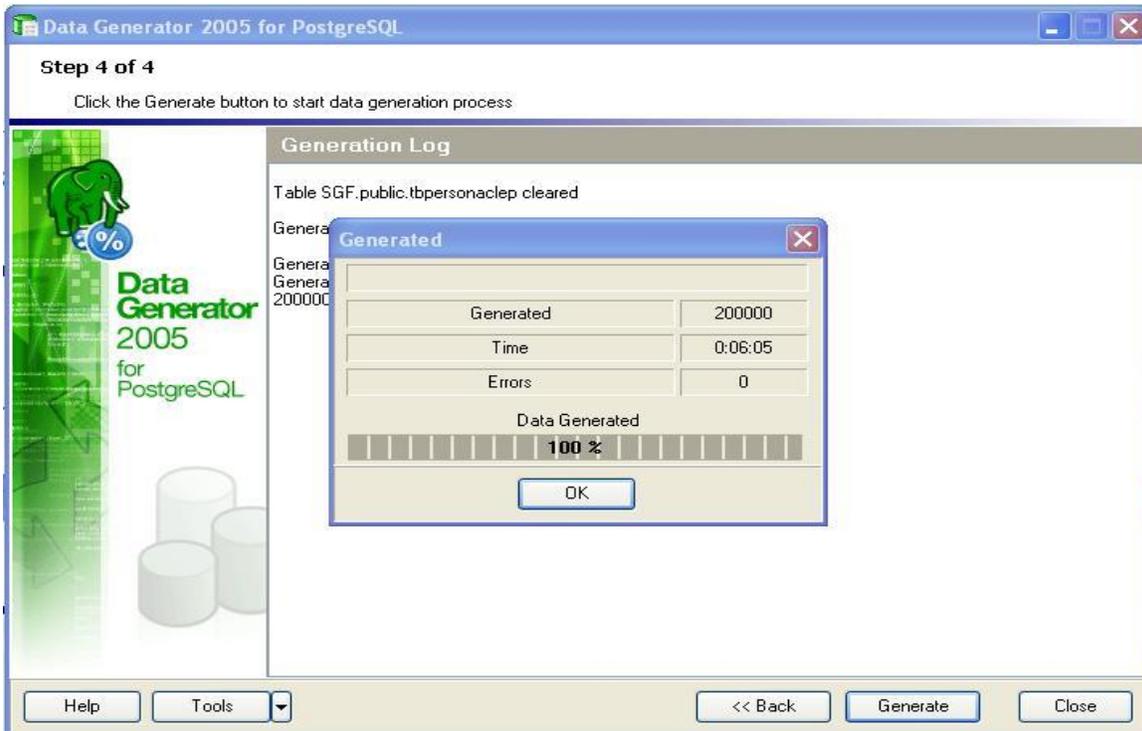


Ilustración 19. Resultados de la generación de datos de tbpersonaclep.

3.3.3 Prueba de Estrés.

Las pruebas de stress se aplican con el fin de medir la veracidad de la aplicación en las escenas de cargas al límite, además de brindar ayuda a los administradores para comprobar si el software fructificará en caso de que la carga real supere a la carga esperada. Para el desarrollo de dicha prueba se utilizará una herramienta Java llamada JMeter, la misma de distribución gratuita y contenida dentro del proyecto Apache Jakarta. Dicha herramienta rinde utilidad de prueba de carga para analizar y medir el desempeño de una variedad de servicios, con énfasis en aplicaciones web.

Se realizaron varias iteraciones de dicha prueba y a continuación quedan expuestas las propiedades de los hilos de la prueba, además del significado y el valor definido para cada uno.

- **Número de hilos:** números de usuarios a simular.
- **Período de subida:** tiempo que se toma el JMeter en lanzar todos los hilos.
- **Contador del bucle:** número de veces a realizar la prueba.

Propiedades de los hilos	Prueba 1	Prueba 2
Número de hilos	50	100
Período de subida (segundos)	1	1
Contador del bucle	5	5

Tabla 21. Pruebas.

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

- **# Muestras:** número de muestras de peticiones JDBC (Java Database Connectivity por sus siglas en inglés).
- **Media:** tiempo transcurrido para un conjunto de resultados.
- **Mediana:** mediana aritmética (elemento de una serie ordenada de valores crecientes de manera que divide en dos partes iguales).
- **Min:** tiempo mínimo transcurrido para las muestras de peticiones JDBC.
- **Max:** tiempo máximo transcurrido para las muestras de peticiones JDBC.
- **% Error:** porcentaje de las peticiones con errores.

Pruebas	# Muestras	Media	Mediana	Mín	Máx	% Error
Prueba 1	250	489	287	12	4381	0
Prueba 2	500	820	341	17	8413	0

Tabla 22. Resultados de las muestras.

Dichas pruebas fueron simuladas partiendo de la idea de que sean casos críticos, por lo general en las fiscalías no deben laborar en el funcionamiento de dicho módulo más de 20 personas. La primera prueba se desarrolló simulando 50 usuarios con un total de 5 iteraciones por cada uno, realizando los mismos una consulta de BD lo que generó un total de 250 peticiones demorando 489 milisegundos de tiempo en atender la solicitud, esto dato lo ofrece el indicador media. La segunda prueba se desarrolló simulando 100 usuarios, siendo este el caso más crítico, con 5 iteraciones por cada uno generando un total de 500 peticiones, la misma efectuó un tiempo de 820 milisegundos en atender dicha solicitud. En ambos casos se puso de manifiesto que la BD resiste una amplia cantidad de consultas efectuadas de manera simultánea, donde el tiempo de respuesta en atender dichas peticiones será minúsculo en relación al tiempo establecido en los requisitos definidos, que son 3 segundos.

Un hecho a acentuar es el referente a que en las pruebas desarrolladas no se mostraron incidencias de errores en algunas de sus peticiones.

3.4 Conclusiones Parciales.

En dicho capítulo se sostuvo un análisis con relación a determinados criterios que se tuvieron presentes para validar la propuesta de solución de base de datos impartida. Se desarrollaron pruebas con el fin de demostrar los acotamientos en los tiempos de respuestas y en la perfección del rendimiento.

Las pruebas de volumen posibilitaron conocer que no se genera problemas con el límite de capacidad de la BD, y las pruebas de estrés demostraron que dicha BD tolera un gran número de conexiones, así como peticiones de modo concurrente.

Conclusiones Generales

Dando finalidad al presente trabajo investigativo se puede alegar que se le ha dado cumplimiento al objetivo del mismo a raíz del diseño e implementación de una base de datos que ofreció el poder gestionar la información del módulo de Inspección a Locales de Detección y Centros Penitenciarios del subsistema CLEP. Consumado el siguiente trabajo investigativo se ofrecen las siguientes conclusiones:

- El desarrollo del diseño y la implementación de bases de datos relacionales posibilitó gestionar la información que se maneja en el módulo Inspección a Locales de Detección y Centros Penitenciarios.
- A raíz del diseño trazado se adquirió un modelo de datos como artefacto esencial para el desarrollo de la BD del módulo Inspección a Locales de Detección y Centros Penitenciarios.
- Las pruebas aplicadas evidenciaron que la BD proveerá respuestas satisfactorias ante gran cantidad de datos, así como brindará sostén ante una amplia suma de conexiones de modo concurrente.

Se elaboró una base de datos, con el propósito de adquirir el almacenamiento de toda la información con la que se cuenta, posibilitando una mejor gestión de la misma.

Recomendaciones

Luego de la investigación empeñada en el desarrollo del trabajo y teniendo presente las ideas que se originaron durante el progreso del mismo se recomienda:

- Perfeccionar los estándares de nomenclatura utilizados en dicho módulo, ejemplo de esto sería emplear la notación Camello con el objetivo de diferenciar los nombres de las tablas mediante la utilización de mayúsculas y minúsculas.
- Brindar un mantenimiento estable a la base de datos dirigido a obtener un mejor funcionamiento.

Referencias Bibliográficas

Blaha, M. (2010). *Los patrones de modelado de datos*. Estados Unidos: CRC Press.

Date, J.C. *Introducción a los Sistemas de Base de Datos*. 2003.

García Chávez, Carlos Alberto, 2005. <http://www.mailxmail.com/curso-diseno-base-datos-relacionales/planificacion-diseno-administracion-bases-datos>.

García, Rosa María Mato. (1999). *Diseño de Base de Datos*.

González, D. (2009). Programanía.net. Desde Programanía.net: <http://www.programania.net/disenode-software/indices-en-bases-de-datos>.

IEEE 610-1990. Instituto de Ingenieros Eléctricos y Electrónicos. IEEE Computer Dictionary Software Engineering Terms. 1990.

Jiménez, M.G. (1985). *Las bases de datos: Una herramienta moderna que contribuirá a la oportuna difusión de la Información Científica y Técnica*. Costa Rica: Bib. Orton.

Márquez, María Mercedes. *Historia de los Sistemas de Base de Datos*. 2001.

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

Olivares Rojas, Juan Carlos. *Patrones de Diseño*. 2010.

PostgreSQL, E. d. *Manual del usuario de PostgreSQL*. 1996.

Pressman, Roger S. *Ingeniería de Software, un enfoque práctico*. Quinta edición. S.I : McGraw-Hill Companies, 2002. ISBN: 8448132149.

Vieyra, Guadalupe, 2005. *Bases de datos relacionales*. Fismat. 2005.

Bibliográficas Consultadas

Blaha, M. (2010). *Los patrones de modelado de datos*. Estados Unidos: CRC Press.

Date, C. (02 de 01 de 2007). DataPrix. DataPrix: <http://www.dataprix.com/11-etapas-diseno-bases-datos>.

Date, J.C. *Introducción a los Sistemas de Base de Datos*. 2003.

García Chávez, Carlos Alberto, 2005. <http://www.mailxmail.com/curso-diseno-base-datos-relacionales/planificacion-diseno-administracion-bases-datos>.

García, Rosa María Mato. *Diseño de Base de Datos*. 1999.

González, D. (2009). *Programanía.net*. Desde Programanía.net: <http://www.programania.net/diseño-de-software/indices-en-bases-de-datos>.

Hansen, Gary W. *Diseño y Administración de base de datos*. 1997.

Harrington, Jan (2000). *Diseño Orientado a Objetos de base de datos explicado claramente*.

Hernández, M. C. (20 de 01 de 2002). EIRIS. Recuperado el 20 de 02 de 2012, de EIRIS: <http://elies.rediris.es/elies18/index.html>

IEEE 610-1990. *Instituto de Ingenieros Eléctricos y Electrónicos*. IEEE Computer Dictionary Software Engineering Terms. 1990.

Jiménez, M.G. (1985). *Las bases de datos: Una herramienta moderna que contribuirá a la oportuna difusión de la Información Científica y Técnica*. Costa Rica: Bib. Orton.

Márquez, María Mercedes. *Historia de los Sistemas de Base de Datos*. 2001.

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

Moreno Ortiz, A. (2000). *Diseño e implementación de un lexicón computacional para lexicografía y traducción automática.* Universidad de Málaga.

Olivares Rojas, Juan Carlos. *Patrones de Diseño.* 2010.

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

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

pgAdmin. 2009. pgAdmin. [En línea] 2009. <http://www.pgadmin.org>.

PostgreSQL, E. d. *Manual del usuario de PostgreSQL.* Thomas Lockhart. 1996.

Pressman, Roger S. *Ingeniería de Software, un enfoque práctico.* Quinta edición. S.I : McGraw-Hill Companies, 2002. ISBN: 8448132149.

Pressman, Roger S. *Ingeniería de Software.* McGraw Hill. 1995.

Vieyra, Guadalupe, 2005. *Bases de datos relacionales.* Fismat. 2005.

WikiProd. http://www.ecured.cu/index.php/Normalización_de_una_base_de_datos. 2009.

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