

Universidad de las Ciencias Informáticas “Facultad 15”



Título: Fase de Inicio del desarrollo de la base de datos del módulo Gobierno y Administración del proyecto Sistema para los Tribunales Cubanos.

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

Autor(es): Yinet Garbey Gainza.

Alfredo Badillo Pérez.

Tutor(es): Ing. Alain Osorio Rodríguez.

Lic. Leonid Mendoza Robaina.

Ciudad de la Habana, Junio 2010

DECLARACIÓN DE AUTORÍA

Nosotros, Yinet Garbey Gainza y Alfredo Badillo Pérez declaramos que somos los únicos autores de este trabajo y autorizamos a la Facultad 15 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los días del mes del año 2010.

Firma del Autor

Firma del Autor

Firma del Tutor

Firma del Tutor



“Saber no es suficiente, debemos aplicar. Desear no es suficiente, debemos hacer.”

Johann W. Von Goethe



Agradecimientos

De Yinet:

Antes que todo agradecerle a nuestro comandante Fidel y a la Revolución por hacer realidad el sueño de crear esta universidad y ofrecerme la oportunidad de formarme en ella.

A mis adorados padres por ofrecerme todo su apoyo y confiar plenamente en mí, por las frases de aliento en los momentos más duros, por sus infinitos sacrificios, por el amor ofrecido, aún cuando una gran distancia nos separó durante estos años y nos golpeó día a día. Mil gracias.

A mis hermanas que siempre estuvieron ahí para apoyarme, que me alentaron y confiaron siempre en mí.

A mi familia en general que durante la ausencia de mis padres no desmayaron nunca sus atenciones conmigo haciéndome la vida un tanto menos dura.

A los tutores que fueron de gran ayuda.

A la profesora Daimí Lamorú, a ella le debo la realización de este trabajo y parte de su éxito.

A mi novio que no ha vacilado jamás en apoyarme ante una situación difícil en estos 5 años, gracias por no dejarme caer. Has sido novio, hermano y amigo, sin ti no habría sido posible.

A mis amigos, los viejos y los nuevos, en especial a las niñas que no me han abandonado nunca a pesar de todo, las llevo en mi corazón a donde quiera que vaya. Mil gracias.

A mi compañero de tesis que ha sido compañero y amigo.

A todos los que de una forma u otra han colaborado con la realización de este trabajo...

Muchísimas gracias.

De Alfredo:

A mi compañera de tesis Yinet por estar ahí a mi lado, por su amor y entrega en la realización de este trabajo.

A mis padres que me han brindado todo su apoyo.

A mis tutores que han estado siempre dispuestos a ayudar en todo momento.

A la profesora Daimí Lamorú, a ella le debo la realización de este trabajo y parte de su éxito.

A todas las personas que han colaborado con la realización de este trabajo.

Muchas Gracias.

Dedicatoria

Dedico este trabajo a mis padres, en especial a mi mamá que ha dado la vida por mí, a ella le debo todo lo que soy. A mi hermanita del alma para quien deseo ser ejemplo y de quien estoy infinitamente orgullosa.

Yinet Garbey Gainza.



Este trabajo va dedicado especialmente a mi mamá Emilia y a mi papá Alfredo, pues gracias a su Amor de estudiantes hoy tengo la enorme dicha de estar vivo.

A la memoria de mi abuelo Esteban del que recibí tanto Amor que me atrevería a decir que quiso más que sus hijos.

A la memoria de mi abuelo Alfredo que desde pequeño siempre conté con su apoyo incondicional y por ser la persona más maravillosa que he conocido.

A la memoria de mi tío Rudy por brindarme todo su cariño.

A mis abuelas Elsia y Onelia que nunca podré retribuirles todo el Amor que me han dado durante todos estos años.

A mi primo Rudito, ese pequeñín que quiero con todo mi corazón.

A mi hermano Dagoberto un ser que quiero muchísimo.

A mi hermana Yanet que aunque no la conozco no pierdo las esperanzas de abrazarla algún día.

A la persona que me cantaba cuando pequeño para dormirme, que por todo el Amor que me ha dado siempre lo he considerado como un padre y sé que me considera su hijo, a ti Rubén te agradezco por los valores que me enseñaste que me identifican como hombre.

A mi prima Iliana por ser tan cariñosa conmigo.

A mis amigos que los considero mis hermanos, a Fuoman, Manuel, Elie y Arnel. Gracias por existir y estar ahí cuando los necesité.

A toda mi familia.

Alfredo Badillo Pérez.

Resumen

En los Tribunales Populares Cubanos se procesa un alto volumen de información, la misma se encuentra dispersa en múltiples libros de registro y archivos por lo que la introducción de errores suele ser muy frecuente. Con el objetivo de dar solución a este problema en la facultad 15 de la Universidad de las Ciencias Informáticas se desarrolla un software que cuenta con un módulo denominado Gobierno y Administración, el mismo es un módulo administrativo cuya base de datos será propuesta en el presente trabajo. La misma permitirá centralizar y organizar la información tramitada en los tribunales.

Para el diseño de la base de datos fue necesario realizar una investigación previa sobre las tendencias actuales en cuanto a soluciones informáticas existentes en el mundo, relacionadas con sistemas jurídicos, de igual forma se recopiló información relacionada con las bases de datos y sistemas gestores de bases de datos. Una vez obtenida la información necesaria se procedió a la construcción de la base de datos requerida, la misma se basó en un modelo para el desarrollo de bases de datos, ofreciéndose organización tanto del equipo de desarrollo como del trabajo en sí. Se realizó el correspondiente modelado de los datos así como las configuraciones del servidor de base de datos y la implementación de algunas funcionalidades iniciales. Además fueron tomadas una serie de medidas que garantizaron seguridad e integridad de la información almacenada según el diseño realizado.

Índice

Agradecimientos	III
Dedicatoria.....	V
Resumen	VII
Índice de Figuras.	XI
Introducción	1
Capítulo #1: Fundamentación Teórica	4
1.1 Introducción	4
1.2 Clasificación de la Informática Jurídica.....	4
1.2.1 La informática jurídica de gestión	5
1.3 Sistemas informáticos para instituciones de justicia.	5
1.3.1 Sistemas informáticos jurídicos a nivel internacional.	5
1.3.2 Sistemas informáticos en el ámbito jurídico en Cuba.....	7
1.4 Introducción a las Bases de Datos	8
1.4.1 Componentes de una Base de Datos	9
1.4.2 Definición de Integridad.....	10
1.4.2.1 Integridad de los datos	10
1.5 Clasificaciones de las Bases de Datos	11
1.5.1 Según la forma en que cambia la información almacenada.....	11
1.5.2 Según la información que almacenan.....	11
1.6 Fases del diseño de base de datos.	12
1.6.1 Diseño Conceptual	12
1.6.2 Diseño Lógico.....	12
1.6.3 Diseño Físico.....	12
1.7 Modelos para el desarrollo de Bases de Datos.	13
1.7.1 Modelo de datos Jerárquico	13
1.7.2 Modelo de datos de red.....	14
1.7.3 Modelo de datos relacional.....	15
1.7.4 Modelo de datos orientado a objetos.....	16
1.7.5 Modelos a utilizar	17
1.9 Seguridad de las Bases de Datos	21

1.10 Análisis y descripción de las herramientas a utilizar.....	23
1.10.1 Sistemas Gestores de Bases de Datos (SGBD).....	23
1.10.1.1 MySQL.....	24
1.10.1.2 Oracle.....	25
1.10.1.3 PostgreSQL.....	26
1.10.2 Herramientas CASE.....	28
1.10.2.1 Rational Rose.....	29
1.10.2.2 Visual Paradigm.....	29
1.10.2.3 Embarcadero Erwin Studio.....	30
1.11 Tendencias actuales y futuras.....	31
1.12 Conclusiones.....	32
Capítulo #2: Descripción de la Solución.....	33
2.1 Introducción.....	33
2.2 Argumentación de los requisitos funcionales y no funcionales.....	33
2.2.1 Requisitos Funcionales del Módulo Gobierno y Administración.....	33
2.2.2 Requisitos no Funcionales.....	34
2.3 Arquitectura.....	35
2.3.1 Arquitectura de datos.....	36
2.4 Modelo Lógico - Conceptual del Módulo Gobierno y Administración.....	37
2.4.1 Tribunal.....	37
2.4.2 Roles.....	38
2.4.3 Gerencial.....	39
2.5 Modelo Físico del Módulo Gobierno y Administración.....	40
2.6 Descripción de las principales tablas del diseño de la base de datos.....	40
2.7 Configuraciones.....	43
2.8 Implementación.....	44
2.9 Conclusiones.....	48
Capítulo #3: Validación de la Solución.....	49
3.1 Introducción.....	49
3.2 Validación teórica del diseño.....	49
3.2.2 Restricciones de integridad de la BD.....	49
3.2.3 Análisis de redundancia de la información.....	52
3.2.4 Normalización de la Base de Datos.....	53
3.2.5 Seguridad.....	54
3.3 Modelo de Desarrollo de Bases de Datos.....	55
3.4 Resultados preliminares.....	58

3.5 Conclusiones	59
Conclusiones Generales	60
Recomendaciones	61
Bibliografía.....	62
Anexos.....	65
Anexo1: Modelo Lógico del Módulo Gobierno y Administración.....	65
Anexo2: Modelo Físico de la BD del Módulo Gobierno y Administración.....	66
Anexo 3: Entrevista	67
Anexo 4: Aval de calidad.....	68

Índice de Figuras.

<i>Figura 1: Modelo de Datos Jerárquico</i>	<i>14</i>
<i>Figura 2: Modelo de Desarrollo de BD</i>	<i>18</i>
<i>Figura 3: Estructura de servidores.</i>	<i>36</i>
<i>Figura 4: Submodelo Tribunal</i>	<i>37</i>
<i>Figura 5: Submodelo Roles</i>	<i>38</i>
<i>Figura 6: Estructuración de procesos</i>	<i>39</i>
<i>Figura 7: Nivel de Organización.</i>	<i>57</i>
<i>Figura 8: Cantidad de Artefactos.</i>	<i>57</i>

Introducción

La batalla que actualmente se desarrolla en el país por la informatización de la sociedad ha ido creciendo gradualmente, con el propósito de elevar la calidad de vida del pueblo cubano, alcanzar un mayor desarrollo tecnológico en el país así como convertir la industria del software en uno de los principales rubros de ingresos de recursos a la nación. La misma ha comenzado por los principales sectores de la sociedad, de forma que los Tribunales Populares Cubanos no han sido la excepción. Esta importante institución juega un papel decisivo en cuanto al orden público se refiere, razón por la que se impone lograr un ágil y efectivo procesamiento de la información.

Los tribunales están estructurados por instancias y materias. Existen tres instancias: municipal, provincial y supremo, cada uno de los cuales cuenta con diferentes procesos distribuidos en cinco materias que son tratadas a todos los niveles. Las materias anteriormente mencionadas son:

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

La Universidad de las Ciencias Informáticas como motor impulsor de la producción de software para la exportación y para el país ha asumido la responsabilidad de informatizar el trabajo en los Tribunales Populares Cubanos, de ahí que en la Facultad 15 se desarrolla el sistema informático para los mismos. Dicho sistema está compuesto por 6 módulos, donde cada materia representa un módulo y adicionalmente se concibió el módulo Gobierno y Administración para la gestión de los usuarios, tribunales, salas, días hábiles y la obtención de reportes estadísticos que ayuden a la toma de decisiones. Su objetivo fundamental es el de vincular y controlar la información generada por el resto de los módulos.

En los tribunales se tramita un alto volumen de información, proceso que se realiza manualmente, esto provoca que la introducción de errores sea frecuente. También existe duplicidad en el número de expediente que tienen asignados los casos o asuntos¹, incluso aunque se encuentren en materias diferentes puede ocurrir que se elaboren dos expedientes que posean el mismo número, de igual forma sucede con la información registrada en los diferentes libros que con el objetivo de lograr rapidez en el trabajo fueron creados. Además los documentos en formato físico con el tiempo se van deteriorando debido a factores como la humedad y los insectos.

Un aspecto importante es que no se reutiliza la información existente, fundamentalmente en la creación de los documentos de los expedientes, dígame resoluciones, diligencias², citaciones³ y notificaciones⁴, pues cada vez que se genera un documento hay que escribir nuevamente informaciones que con anterioridad fueron registradas. Otro problema que afecta estas instituciones es la inexistencia de sistemas automatizados que permitan obtener reportes estadísticos en tiempo real, lo que acarrea grandes demoras en el procesamiento de la información manejada por las diferentes instancias. Además existen sistemas informáticos aislados que no proporcionan visibilidad entre los procesos, por lo que se hace necesario integrar las funcionalidades de estos programas en aras de que el personal autorizado pueda trabajar con dichos sistemas de forma conjunta y accediendo a una única fuente de información.

Dada esta situación problemática surge el siguiente problema: ¿Cómo solucionar la descentralización de la información en los Tribunales Populares Cubanos de modo que contribuya a la gestión eficiente de sus procesos?

Constituye el objeto de estudio: sistemas informáticos para la gestión jurídica.

¹ **Caso o Asunto:** Conflicto o litigio que acontece en la vida social y se hace del conocimiento de un órgano imparcial (el tribunal) para su solución.

² **Diligencias:** Acto procesal que se firma por la persona con quien se entienda y por la persona autorizada que la practique, dejándose constancia de día y la hora en que efectúe.

³ **Citaciones:** Acto procesal en virtud del cual se cita a una persona que debe comparecer ante el tribunal o mantenerse en su domicilio para la realización de un acto judicial.

⁴ **Notificaciones:** Acto procesal que se efectúa entregando a la persona con quien debe entenderse copia literal autorizada de la resolución.

Constituye el campo de acción el desarrollo de bases de datos para los sistemas informáticos de los Tribunales Populares Cubanos.

El objetivo general de la investigación es: Contribuir a la solución integral del sistema mediante el desarrollo de las actividades de modelado de datos, configuración, implementación y ADTP⁵ para la fase de inicio del desarrollo de la base de datos del módulo Gobierno y Administración.

La hipótesis sobre la cual se sustentó la investigación es que si se desarrollan las actividades de modelado de datos, configuración, implementación y ADTP para la fase de inicio del módulo Gobierno y Administración del proyecto “Sistema para los Tribunales Cubanos”, entonces se contribuirá a solucionar la descentralización de la información existente en los mismos.

Entre las principales tareas de la investigación a realizar se encuentran:

- ✓ Realización del estudio del estado del arte de sistemas informáticos jurídicos en Cuba y el mundo.
- ✓ Realización del estudio del estado del arte de sistemas de bases de datos.
- ✓ Desarrollo del modelo de datos y su documentación.
- ✓ Desarrollo de las Configuraciones del Sistema Operativo – Sistema Gestor de Bases de Datos – Base de Datos y su automatización.
- ✓ Desarrollo de la codificación de las funcionalidades.
- ✓ Validación de la solución propuesta.

⁵ **ADTP:** Acceso a Datos, Tuning(Optimización), Prueba.

Capítulo #1: Fundamentación Teórica

1.1 Introducción

En el presente capítulo se da una breve reseña del tema de los sistemas informáticos para instituciones jurídicas en Cuba y el mundo, de igual forma se describen aspectos importantes de las Bases de Datos en la actualidad, las tendencias tecnológicas de las mismas y las metodologías de diseño con sus fases. Se abordan también temas como la seguridad, integridad de la información y normalización. Además se especificarán las herramientas empleadas para el diseño e implementación de la base de datos (BD) del módulo Gobierno y Administración del proyecto Sistema para los Tribunales Cubanos.

1.2 Clasificación de la Informática Jurídica.

Hace algunos años la aplicación de las Tecnologías de la Información y las Comunicaciones (TICs) en el área judicial no se había concebido con claridad, pero con las tendencias actuales de la informatización y el uso de las tecnologías, no puede quedarse al margen de su uso, pues sería ir en contra del desarrollo de la humanidad.

Teniendo en cuenta lo anteriormente expresado surge el concepto de informática jurídica como la tecnología aplicada a la sistematización que estudia el tratamiento automatizado de:

- ✓ *Las fuentes del conocimiento jurídico a través de los sistemas de documentación legislativa, jurisprudencial y doctrinal que sería lo mismo que la informática jurídica documental.*
- ✓ *Las fuentes de producción jurídica, a través de la elaboración informática de los factores lógico-formales que concurren en el proceso legislativo y en la decisión judicial, es decir, informática jurídica decisional.*
- ✓ *Los procesos de organización de la infraestructura o medios instrumentales con los que se gestiona el Derecho, que no es más que la informática jurídica de gestión, la cual se utiliza para automatizar sistemas de gestiones jurídicas que facilitan el trabajo de las diferentes entidades que se encargan de consultar y/o prestar servicios jurídicos. (Silva, 2009)*

1.2.1 La informática jurídica de gestión

La Informática Jurídica de Gestión es el método o el sistema empleado que hace de toda actividad jurídica algo estructurado, algo más simple y efectivo. Sirve para organizar y hacer más funcionales los despachos que puedan de alguna forma tratar temas jurídicos. (Solano, 2003).

Además está dirigida a actuar dentro de un proceso fundamentalmente administrativo que está dominado por el procesamiento de datos.

Se subdivide en:

- ✓ Informática Registral: Se ocupa de todos los tipos de registros (públicos o privados).
- ✓ Informática Operacional: Permite el control de los pasos que en una clase de proceso específico se vayan cumpliendo.
- ✓ Informática Decisional: Emplea el ordenador en la ayuda de toma de decisiones a través de la solución de casos (Inteligencia Artificial).

1.3 Sistemas informáticos para instituciones de justicia.

1.3.1 Sistemas informáticos jurídicos a nivel internacional.

Actualmente la creación de software para el campo jurídico está comenzando a crecer, debido a la utilización de las TICs con fines de agilizar y modernizar los procesos llevados a cabo por las organizaciones y aprovechar todas las ventajas que las tecnologías aportan en cualquier área de la vida. Ejemplo de ello son los sistemas informáticos que se han comenzado a usar en países como Argentina (GIAJ - TRAMIX) y España (LEX NET, MINERVA, SEINSIR), con el objetivo de facilitar la tramitación de las causas, de los expedientes y acortar los plazos de duración de los juicios. Estos sistemas actúan sobre la base de la tecnología de los certificados digitales permitiendo el uso de la firma digital y garantizando la autenticidad de los documentos que se envían.

A continuación se ofrece una breve descripción de algunos de los sistemas mencionados.

GIAJ-TRAMIX (Argentina)

El sistema está planteado con arquitectura Cliente/Servidor para los usuarios internos del Poder Judicial, y arquitectura webenable con motor Oracle para los usuarios externos (abogados y otros justiciables). Las funcionalidades previstas incluyen la posibilidad para los profesionales de ingresar a sus expedientes y efectuar el control de las actuaciones, notificarse y también incorporar escritos en forma remota, es decir a través de Internet, todo mediante el uso de cualquier dispositivo que permita esta navegación.

Este sistema genera importantes ventajas, que se traducen en avances relacionados con la agilización de los procesos, incremento de la tarea de oficio favorecida por la incorporación de procesos automatizados y el descongestionamiento de los circuitos administrativos, factores que permitirían proporcionar una operatoria más eficiente y eficaz, disminuyendo también los costos.

Lex Net (España)

Es un sistema de comunicaciones electrónicas que permite la comunicación bidireccional de las oficinas judiciales con los distintos operadores jurídicos, esto es, con los abogados y procuradores, facilitando los flujos y procesos internos de actos de comunicación procesal por los órganos judiciales como la presentación de escritos y documentos por los profesionales del derecho.

Minerva (España)

Es un sistema que automatiza e integra el seguimiento de los recursos y sentencias de la consejería de educación de la junta de Andalucía incluyendo el seguimiento de la vertiente económica de estos expedientes. El mismo es un tramitador de expedientes que ha sido implantado por varias Consejerías de la Junta de Andalucía, el cual ha gestionado alrededor de más de 5000 expedientes hasta el momento y se encuentra en una fase de desarrollo para incluirle mejoras que contribuyan a una mayor eficiencia y eficacia de sus funcionalidades.

SEINSIR (España)

SEINSIR es un Sistema Informático de Gestión Judicial que permite integrar nuevas tecnologías (tratamiento de imágenes, bases de datos documentales, conexiones telemáticas, arquitectura cliente-servidor a tres niveles, etc.) en los productos desarrollados para las oficinas judiciales de cualquier instancia y los servicios comunes relacionados, incrementando su nivel tecnológico y productivo y mejorando el servicio al ciudadano.

SEINSIR permite generar aplicaciones para dar solución a las necesidades de registro, control y seguimiento de los asuntos judiciales que se resuelvan en un órgano judicial de cualquier tipo. Esto es

posible por el sofisticado sistema de parametrización que incorpora y que permite la definición externa de la colección de datos y su representación y los listados, estadísticas y consultas necesarias para el funcionamiento informatizado del órgano judicial.

Este sistema ha sido concebido básicamente a partir del conocimiento de las necesidades del ámbito judicial de la comunidad Latinoamericana y de España.

1.3.2 Sistemas informáticos en el ámbito jurídico en Cuba.

En el campo de la Informática Jurídica de Gestión el país ha comenzado a dar sus primeros pasos, con el objetivo de la automatizar los procesos de las diferentes áreas en cada una de las instancias existentes para ganar en rapidez al tramitar la información. Como parte fundamental de los sistemas de gestión jurídica que están surgiendo se encuentran las bases de datos, que contienen toda la información persistente que utilizan estos procesos de justicia y sin la cual no podrían funcionar dichos sistemas.

“Es de vital importancia señalar que aún cuando el Sistema Jurídico Cubano, se origina del Sistema Jurídico Romano -Francés, del cual provienen también los de los países que usan hoy esos sistemas de gestión, como es el caso de España y Argentina, sus realidades actuales en el plano jurídico difieren totalmente de la cubana, razón por la cual resulta de alta complejidad la adaptación de estos sistemas. Todos ellos se rigen por las leyes judiciales de su país por lo que sus módulos no se corresponden con las leyes socialistas establecidas en Cuba”. (Silva, 2009)

Los elevados costos de estos sistemas, además del mantenimiento de sus licencias de uso, hacen que Cuba no los pueda asumir desde el punto de vista económico, cuestión que por las realidades financieras actuales es inevitable que se tome en consideración como punto de partida para cualquier análisis a la hora de proponer un sistema.

Tomando en cuenta todo lo anterior, se considera, que Cuba debe apostar por la realización de un sistema que cumpla con las características requeridas para satisfacer lo concerniente a la gestión de servicios legales que pueda ser utilizable en cualquier despacho legal. Un sistema que se ajuste a la realidad jurídica cubana, desarrollado por especialistas con herramientas de código abierto que permitan su mantenimiento, adaptaciones y ulteriores versiones.

Varios intentos de informatización de los Tribunales Populares Cubanos se han llevado a cabo en el país, entre ellos:

- ✓ SISPRO (Procesos penales, Villa Clara, Cienfuegos).

- ✓ SISECO (Procesos Económicos).
- ✓ SI de la Sala Penal del Tribunal Supremo, (No se concluyó pero funciona actualmente)
- ✓ SI de la Sala Civil del Tribunal Supremo, (No se concluyó pero funciona actualmente).

De forma general estos programas presentan las siguientes deficiencias:

- ✓ No ofrecen reportes estadísticos.
- ✓ Nula comunicación entre instancias.
- ✓ No superan las barreras del papel.
- ✓ Arraigados a los registros tradicionales (libros).
- ✓ Utilización de software propietario.
- ✓ Se puede hacer cualquier acción sobre el expediente sin tener en cuenta el estado del mismo.

Otro software que es utilizado en el país con fines jurídicos es el “Software para el Control de Contratos” del que se ofrecen algunos detalles a continuación.

Software para el Control de Contratos.

La Empresa Cubana Nacional de Software más conocida por Desoft S.A, ha trabajado por ir informatizando todos sus procesos internos. En Ciego de Ávila existe una aplicación para el control de los contratos, a la cual se le hizo una primera mejora. Se tomó como base el procedimiento de calidad para el control de los contratos aprobado por la Oficina Central. Los abogados son los encargados de registrar los clientes e insertar los contratos. El número del contrato es invariable, pero el software permite incorporar detalles como son: pagos, plazos de entrega y detalles sobre la venta.

Los especialistas de ventas y los subgerentes tienen su sesión para hacer las consultas que necesiten. También existe la posibilidad de hacer reportes y exportar a una tabla en Excel. Cantidad de contratos, por servicios, vendedores y período de tiempo.

Sus requerimientos técnicos son: Java 2 EE (iBatis como framework de persistencia y Structs como de presentación), servidor de aplicaciones TomCat y base de datos SQL Server.

1.4 Introducción a las Bases de Datos

El término de bases de datos fue escuchado por primera vez en 1963, en un simposio celebrado en California, Estados Unidos. Inicialmente una base de datos podía ser una de las antiguas bibliotecas o

cualquier tipo de información coherente que fue persistida en papel y seguidamente almacenada en un estante o similar. En la actualidad, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital lo que ofrece un amplio rango de soluciones al problema de almacenar datos. Existen varios criterios o definiciones de las bases de datos enunciadas por diferentes personalidades en el mundo, entre ellas se encuentran:

“Una Base de Datos es una colección de datos interrelacionados almacenados conjuntamente en uno o más ficheros de computadora” (IEEE, 1990).

“Un sistema de bases de datos es básicamente un sistema computarizado cuya finalidad general es almacenar información y permitir a los usuarios recuperar y actualizar esa información con base en peticiones”. (Date, 2003)

“Colección o depósito de datos, donde los datos están lógicamente relacionados entre sí, tienen una definición y descripción comunes y están estructurados de una forma particular. Una base de datos es también un modelo del mundo real y, como tal, debe poder servir para toda una gama de usos y aplicaciones”. (Kybele,2006)

“Una Base de datos es una colección de datos interrelacionados almacenados conjuntamente en uno o más ficheros de computadora de acuerdo a una organización y estructura sistemática determinada por un modelo de bases de datos”. (Miguel-Ángel Sicilia, 2008)

“Conjuntos de ficheros interrelacionados, con estructuras complejas y compartidos por varios procesos de forma simultánea (unos on-line y otros por lotes), recibieron al principio el nombre de Data Banks, y después, a inicios de los años setenta, el de Data Bases.” (Rafael Camps Paré, 2005)

“Una base de datos es una colección de datos organizados de una manera particular.” (Chari, 2009)

En resumen una base de datos no es más que un conjunto de datos organizados y coherentemente relacionados entre sí los cuales son recolectados para su posterior explotación por los sistemas de información.

1.4.1 Componentes de una Base de Datos

- ✓ **Hardware:** constituido por dispositivos de almacenamiento como discos, tambores, cintas, etc.
- ✓ **Software:** el Sistema Gestor de Bases de Datos (SGBD).
- ✓ **Datos:** están almacenados de acuerdo a la estructura externa y van a ser procesados para convertirse en información.

1.4.2 Definición de Integridad

Son restricciones que definen los estados de consistencia de la Base de Datos, esto significa que la BD o los programas que generaron su contenido, incorporen métodos que aseguren que el contenido de los datos del sistema no se rompa, así como las reglas del negocio. La integridad de los datos puede verse afectada añadiendo datos no válidos, modificando datos existentes, que se les asigne un valor incorrecto o eliminando datos que violen alguna regla. La integridad de datos presenta varias restricciones que posibilitan la declaración y comprobación de condiciones para expresar la consistencia, corrección y exactitud de datos almacenados, estas son: Integridad de dominio, de clave, de entidad, referencial, entre otras.

1.4.2.1 Integridad de los datos

La integridad de los datos no es más que la corrección y completitud de la información en una base de datos. Cuando los contenidos se modifican con sentencias INSERT, DELETE o UPDATE, la integridad de los datos almacenados puede perderse de muchas maneras diferentes. Pueden añadirse datos no válidos a la base de datos, tales como un pedido que especifica un producto inexistente. Los cambios en la base de datos pueden perderse debido a un error del sistema o a un fallo en el suministro de energía. Una de las funciones importantes de un sistema gestor de base de datos relacional es preservar la integridad de sus datos almacenados en la mayor medida posible.

Existen cuatro restricciones fundamentales que sustentan la integridad de los datos, las mismas se mencionan a continuación:

- ✓ Datos Requeridos: establece que una columna tenga un valor no NULL. Se define efectuando que la declaración de una columna es NOT NULL cuando la tabla que contiene las columnas se crea por primera vez, como parte de la sentencia CREATE TABLE.
- ✓ Chequeo de Validez: Cuando se crea una tabla donde en cada columna tiene un tipo de datos y el sistema gestor de base de datos asegura que solamente los datos del tipo especificado sean ingresados en la tabla.
- ✓ Integridad de entidad: Establece que la clave primaria de una tabla debe tener un valor único para cada fila de la tabla; sino, la base de datos perderá su integridad. Se la sentencia CREATE TABLE. El sistema

gestor de base de datos comprueba automáticamente la unicidad del valor de la clave primaria con cada sentencia INSERT Y UPDATE. Un intento de insertar o actualizar una fila con un valor de la clave primaria ya existente fallará.

- ✓ Integridad Referencial: La integridad referencial es un sistema de reglas que utilizan la mayoría de las bases de datos relacionales para asegurarse que los registros de tablas relacionadas son válidos y que no se borren o cambien datos relacionados de forma accidental produciendo errores de integridad.

1.5 Clasificaciones de las Bases de Datos

Las bases de datos pueden ser clasificadas de múltiples formas teniendo en cuenta algunos criterios existentes.

1.5.1 Según la forma en que cambia la información almacenada

Bases de datos estáticas: Son aquellas bases de datos en las que no se puede modificar la información que está guardada, es decir, solo se puede consultar, son utilizadas preferentemente para guardar datos históricos con los que luego se podrán realizar estudios sobre su comportamiento en el transcurso del tiempo, lo cual ayudará a tomar decisiones.

Bases de datos dinámicas: Son aquellas bases de datos en las que la información almacenada está en constante cambio, ya que sobre ella se pueden realizar operaciones tales como: adicionar, actualizar, eliminar, etc., así como las operaciones de consultas.

1.5.2 Según la información que almacenan

Bases de datos bibliográficas: Almacenan la información necesaria para poder localizar la fuente primaria. Un registro típico de una base de datos bibliográfica contiene información sobre el título, fecha de publicación, autor, editorial de una determinada publicación, etc. Puede contener un fragmento de la publicación original, pero nunca el texto completo.

Bases de datos de texto completo: Almacenan las fuentes primarias, es decir, guardan toda la información íntegra de las fuentes primarias, no fragmentos, ni referencias.

Directorios: Este es el caso de directorios digitales como los que edita la Empresa de Telecomunicaciones de Cuba (ETECSA) con los números telefónicos de sus usuarios, ya sean privados o estatales, donde se puede encontrar el nombre al que pertenece dicho número y la dirección dónde se encuentra.

Banco: es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

1.6 Fases del diseño de base de datos.

El proceso de diseño de una base de datos cuenta con tres fases fundamentales: Diseño Conceptual, Diseño Lógico y Diseño Físico.

1.6.1 Diseño Conceptual

El objetivo del diseño conceptual, también conocido como modelo conceptual, y que constituye la primera fase de diseño, es obtener una buena representación de los recursos de información, con independencia de usuarios y aplicaciones en particular y sin considerar aspectos como eficiencia de la computadora. Esta primera fase consta de dos momentos: análisis de requisitos, donde se centra el trabajo en definir qué es lo que se va a representar, y la conceptualización, donde se piensa en cómo se va a proceder para representar lo antes definido.

1.6.2 Diseño Lógico.

El diseño lógico parte del esquema conceptual. Un esquema lógico es la descripción de la estructura de la base de datos que puede procesarse por el Sistema Gestor de Base de Datos. El objetivo principal del diseño lógico es obtener un esquema lógico eficiente en cuanto a operaciones de consulta y actualización.

1.6.3 Diseño Físico.

En este paso se especifican las estructuras de almacenamiento internas y la organización de los archivos de la base de datos. El principal objetivo del diseño físico es conseguir una instrumentación lo más eficiente posible del esquema lógico. Para lograrlo se analizan aspectos como las características del Sistema Operativo, el Sistema Gestor de Base de Datos, la herramienta para realizar el diseño, aspectos relacionados con el rendimiento y los requisitos de procesos así como las características del hardware, en

fin, cualquier factor cercano con la computadora, para con ello lograr optimizar el consumo de recursos, minimizar el espacio de almacenamiento, proporcionar la seguridad máxima, disminuir los tiempos de respuesta y evitar las reorganizaciones.

1.7 Modelos para el desarrollo de Bases de Datos.

1.7.1 Modelo de datos Jerárquico

Los modelos de datos jerárquicos almacenan la información de forma jerárquica. Los datos se organizan en forma similar a un árbol donde un padre tiene muchos hijos. Este árbol está compuesto de unos elementos llamados nodos. El nivel más alto del árbol se denomina raíz. Cada nodo representa un registro con sus correspondientes campos.

Este modelo es especialmente útil en el caso de aplicaciones que manejan un gran volumen de información y datos muy compartidos, permitiendo crear estructuras estables y de gran rendimiento.

“El modelo de datos jerárquico presenta importantes inconvenientes, que provienen en gran medida de su rigidez, por la falta de capacidad de las organizaciones jerárquicas para representar sin redundancias ciertas estructuras muy difundidas en la actualidad. La poca flexibilidad de este modelo puede obligar a la introducción de redundancias cuando es preciso instrumentar, mediante el modelo jerárquico, situaciones del mundo real que no responden a una jerarquía”. (Moraga, 2001)

La representación gráfica de este modelo se realiza mediante la creación de un árbol invertido, los diferentes niveles quedan unidos mediante relaciones. Ejemplo:

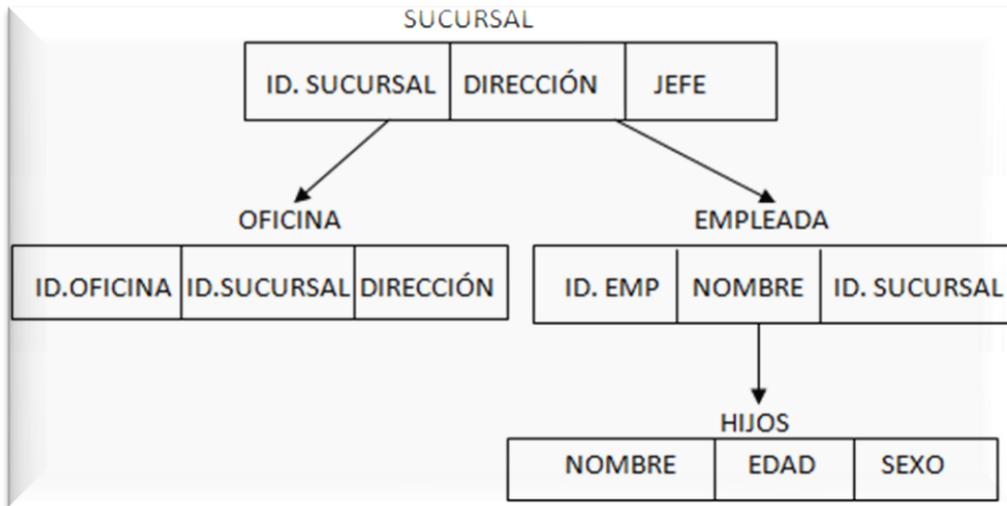


Figura 1: Modelo de Datos Jerárquico

Además en este modelo solo se pueden representar relaciones 1: M, lo que provoca que ocurran problemas como son:

- ✓ No se admiten relaciones N:M
- ✓ Un segmento hijo no puede tener más de un padre.
- ✓ No se permiten más de una relación entre dos segmentos.
- ✓ Para acceder a cualquier segmento es necesario comenzar por el segmento raíz.
- ✓ El árbol se debe de recorrer en el orden designado.

1.7.2 Modelo de datos de red

Una base de datos en red consiste en un conjunto de registros conectados entre sí mediante punteros. Los registros son en muchos aspectos parecidos a las entidades del modelo entidad-relación (E-R). Cada registro es un conjunto de campos (atributos), cada uno de los cuales sólo contiene un valor de datos. Los punteros son asociaciones entre exactamente dos registros. Por tanto, los punteros pueden considerarse una forma restringida (binaria) de relación en el sentido del modelo E-R.

A diferencia del modelo jerárquico, en este modelo, un hijo puede tener varios padres. Esto ofrece una solución eficiente al problema de la redundancia de datos. Este modelo de datos permite representar relaciones N: M sin embargo es poco utilizado.

Los conceptos básicos en el modelo en red son:

- ✓ El tipo de registro, que representa un nodo.
- ✓ Elemento, que es un campo de datos.
- ✓ Agregado de datos, que define un conjunto de datos con nombre.

Los modelos anteriores presentan deficiencias como son:

- ✓ Difíciles de administrar.
- ✓ Ejecución compleja.
- ✓ Carencia de independencia estructural.

1.7.3 Modelo de datos relacional

En 1970 E. F. Codd introdujo el modelo relacional, todos los datos están estructurados a nivel lógico como tablas formadas por filas y columnas, aunque a nivel físico pueden tener una estructura completamente distinta. Un punto fuerte del modelo relacional es la sencillez de su estructura lógica. Algunas de las ventajas del modelo relacional son:

- ✓ Define un álgebra, llamada álgebra relacional a partir de la cual se realizan todas las manipulaciones posibles sobre las relaciones, que se obtienen mediante el uso de operadores.
- ✓ Garantía de independencia de los datos.
- ✓ Conectividad garantizada con los lenguajes de programación estándar.
- ✓ Compatibilidad y estandarización.
- ✓ Favorece la normalización por ser más comprensible y aplicable.
- ✓ Garantiza la integridad referencial, así al eliminar un registro elimina todos los registros relacionados dependientes.
- ✓ Garantiza herramientas para evitar la duplicidad de registros, a través de campos claves o llaves.

Este modelo presenta desventajas cuando se compara con modelos más actuales, algunas de ellas son:

- ✓ Imposibilidad de representar conocimiento en forma de reglas.
- ✓ Inexistencia de mecanismos de herencia de propiedades.
- ✓ Falta de poder expresivo (por ejemplo, para representar jerarquías).

- ✓ Dificultad para gestionar datos no atómicos.

1.7.4 Modelo de datos orientado a objetos.

Las bases de datos orientadas a objetos se crearon para tratar de satisfacer las necesidades de las nuevas aplicaciones. La orientación a objetos ofrece flexibilidad para manejar algunos de estos requisitos y no está limitada por los tipos de datos y los lenguajes de consulta de los sistemas de bases de datos tradicionales. Una característica clave de las Bases de Datos Orientadas a Objetos (BDOO) es la potencia que proporcionan al diseñador al permitirle especificar tanto la estructura de objetos complejos, como las operaciones que se pueden aplicar sobre dichos objetos. Otro motivo de la creación de estos modelos es el creciente uso de los lenguajes orientados a objetos. Algunas de sus ventajas son:

- ✓ Combinación de los procedimientos de una entidad con sus datos, permite modelar tipos de datos complejos y definir operaciones entre ellos.
- ✓ La cantidad de información que puede modelarse en una base datos orientada a objetos se incrementa, y es más fácil modelar esta información.
- ✓ Los Sistemas de Bases de Datos Orientados a Objetos son capaces de tener mayores capacidades de modelado por medio de la extensibilidad.
- ✓ En una BDOO, el manejo de versiones está disponible para ayudar a modelar cambios diversos a los sistemas.
- ✓ La reutilización de clases juega un rol vital en el desarrollo y mantenimiento más rápido de aplicaciones. Las clases genéricas son potentes, pero más importante es que ellas pueden ser usadas nuevamente.

Entre sus principales desventajas se pueden mencionar:

- ✓ Cuentan con cierto nivel de inmadurez en el mercado de BDOO.
- ✓ Inexistencia de estándares en la industria orientadas a objetos.
- ✓ Estos sistemas apenas permiten flexibilidad para modificaciones, y el sistema debe desactivarse cuando se requiere modificar estructuras de objetos y métodos.

- ✓ Su arquitectura no es flexible, lo cual motiva, por ejemplo, el uso del modelo relacional.

1.7.5 Modelos a utilizar

Para el diseño de la base de datos se utilizará el modelo relacional pues ofrece un grupo de ventajas importantes en comparación con otros modelos existentes, de ahí que la dirección del proyecto apostara por su puesta en práctica.

Con el fin de lograr mayor organización en el trabajo se utilizará como guía el Modelo de Desarrollo de Bases de Datos para Procesos de Desarrollo de Software propuesto en el trabajo de diploma del Ingeniero Alain Osorio Rodríguez. En el mismo se contemplan 3 fases fundamentales:

1. Inicio o Definición.
2. Desarrollo o Construcción.
3. Transición o Despliegue.

En la fase de inicio el mayor esfuerzo está en desarrollar la primera estructura de la base de datos y a su vez las configuraciones iniciales, lo cual tributa a los primeros pasos del desarrollo. En la fase de desarrollo suele ocupar el mayor tiempo las tareas de implementación y acceso a datos, con determinados cambios en el modelo de datos, y en la fase de transición suele ocupar mayor importancia las configuraciones para la puesta en marcha del software a su entorno real de ejecución.

El proceso está dividido en cuatro actividades fundamentales las cuales incluyen tareas generales.

1. Modelado de Datos.
2. Configuraciones.
3. Implementación.
4. Acceso a Datos, Tuning, Prueba(ADTP).

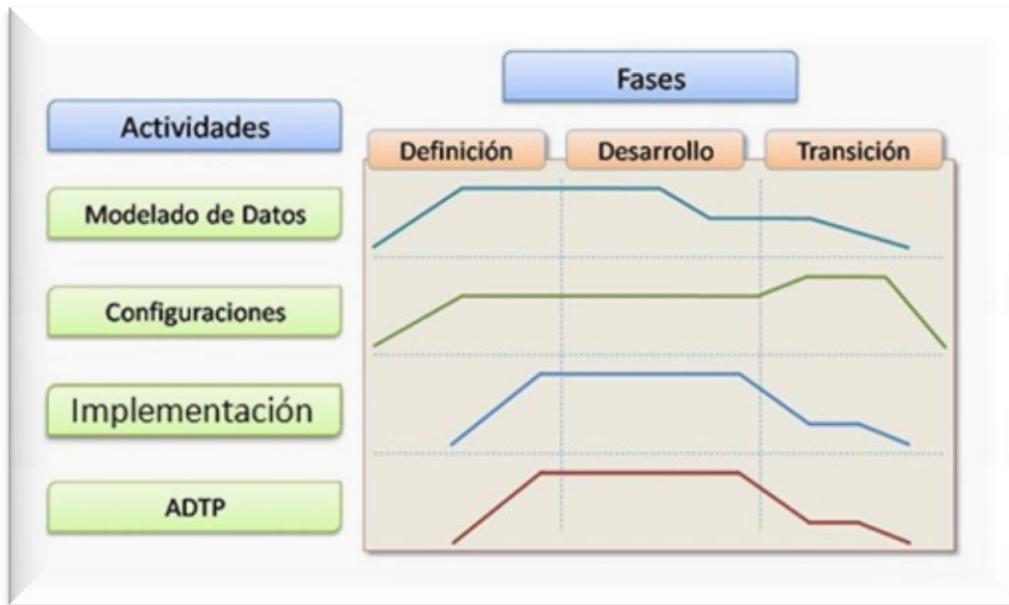


Figura 2: Modelo de Desarrollo de BD.

Modelado de Datos

La actividad tiene como tarea general la estructuración del Modelo de Datos y su respectiva documentación.

Los objetivos de esta actividad son:

1. Estructurar iterativamente el Modelo de Datos.
2. Documentar el Modelo de Datos a la par del desarrollo del mismo.

Se debe ir refinando la documentación asociada a la estructura de la base de datos a medida que evoluciona el Modelo de Datos.

El artefacto que se genera es el Modelo de Datos.

Configuraciones

La actividad tiene como tareas generales el establecimiento de las configuraciones del servidor de base de datos, así como la estructuración de una estrategia de instalación y configuración de la arquitectura de la BD para diferentes entornos de desarrollo, como son el propio desarrollo, prueba y despliegue.

El objetivo de esta actividad es:

1. Definir y aplicar las configuraciones del sistema operativo – sistema gestor de bases de datos – base de datos para diferentes entornos de desarrollo mediante automatizaciones.

Los artefactos que se generan en esta actividad son:

1. Documento de configuraciones del sistema operativo – sistema gestor de bases de datos – base de datos.
2. Estructura automatizada de aplicación de las configuraciones.

Ejemplo de automatización de las configuraciones puede ser crear una imagen del sistema operativo y sus configuraciones para los servidores de base de datos.

Implementación

Esta actividad se centra en la implementación de todas las funcionalidades requeridas en la BD.

Los objetivos de esta actividad son:

1. Implementar las funcionalidades de la base de datos.
2. Documentar las funcionalidades.

Se debe ir documentando las funcionalidades a medida que se van implementando.

El artefacto que se genera es la Codificación de las Funcionalidades. La documentación debe formar parte de este artefacto.

ADTP

El nombre de esta actividad viene dado por las tareas generales que se realizan:

1. Acceso a Datos.
2. Tuning (Optimización).
3. Pruebas Generales.

La tarea general Acceso a Datos se refiere a la creación y modificación de la interfaz de comunicación entre la aplicación de negocio y la base de datos. La tarea general Tuning se refiere a las optimizaciones

de E/S vinculadas con las funcionalidades implementadas en la base de datos. Por último la tarea general de Pruebas Generales se refiere a las pruebas en el marco de las tareas de Acceso a Datos y Tuning, y puede definirse como una tarea intermedia de comprobación y terminación.

En esta actividad, las tareas generales suelen estar muy relacionadas y solo en determinadas ocasiones se realizan aisladamente.

Los objetivos de esta actividad son:

1. Estructurar el acceso a datos.
2. Optimizar las operaciones de E/S vinculadas a la codificación de las funcionalidades de la base de datos.
3. Probar las funcionalidades hasta obtener un resultado eficiente en la medida de los requerimientos del sistema.
4. Garantizar un acceso a datos óptimo y seguro.

El artefacto que se genera es el Acceso a Datos.

1.8 Normalización de las Bases de Datos

La normalización es una técnica para diseñar la estructura lógica de los datos de un sistema de información en el modelo relacional, desarrollada por E. F. Codd en 1972. En este proceso se definen varias formas normales (FN), 1raFN, 2daFN, 3raFN y la Forma Normal de Boyce – Codd (FNBC), donde se somete el esquema de relación a una serie de pruebas para certificar si pertenecen a una cierta forma normal.

Cuando el esquema relacional esté en FNBC, es que ya el esquema se llevó a 3FN, y para estar en 3FN el esquema se tuvo que haber llevado a 2FN y así sucesivamente, por lo que se puede decir que la FNBC es una de las formas normales más deseadas para un buen diseño de BD. No obstante, no siempre que el diseño de BD esté en la FNBC podemos garantizar que sea un diseño idóneo de los datos.

Las bases de datos relacionales se normalizan para:

- ✓ Evitar la redundancia de los datos.
- ✓ Evitar problemas de actualización de los datos en las tablas.

- ✓ Proteger la integridad de los datos.
- ✓ Mejorar la independencia de datos.
- ✓ No establecer restricciones artificiales en la estructura de los datos.

1.8.1 Denormalización de Bases de Datos

La denormalización es el proceso al cual se someten los diseños de bases de datos con el fin de ganar en rendimiento. Una base de datos con la medida justa de denormalización reduce el número de tablas que deben combinarse sin dificultar en exceso el proceso de actualización. Entonces, tendrá consecuencias de redundancia de datos, pero mejorará el rendimiento ya que se evitarían uniones y subconsultas, pero es importante tener en cuenta que sólo se debería implementar si presenta notables ventajas de performance. El punto de vista teórico, marca la premisa en general, de evitar la redundancia de datos para obtener un diseño normalizado y optimizado. Pero si se parte desde la realidad cotidiana en la cual se desenvuelve el negocio, en la práctica, dependiendo del contexto en que se esté trabajando a veces se debe desestimar parte de la normalización, para mejorar el comportamiento de un sistema y lograr un mejor performance en el funcionamiento del mismo. Cabe destacar que a decisión de incluir redundancia en los datos (por más mínima que esta sea) debe ser perfectamente estudiada y analizada, y por otro lado los beneficios deberían ser claramente notorios para que justifique la acción.

1.9 Seguridad de las Bases de Datos

Consiste en las acciones que toma el diseñador de base de datos al momento de crearla, tomando en cuenta el volumen de las transacciones y las restricciones que tiene que especificar en el acceso a los datos; esto permitirá que el usuario adecuado sea quién visualice la información adecuada.

La seguridad en las bases de datos abarca varios temas:

- ✓ Cuestiones éticas y legales relativas al derecho a tener acceso a cierta información.
- ✓ Cuestiones de política en el nivel gubernamental, institucional o corporativo relacionadas con la información que no debe estar disponible para el público.
- ✓ Cuestiones relacionadas con el sistema.

Las bases de datos hoy en día son de gran importancia en cada una de las aplicaciones que la requieran, es por ello que la seguridad para su acceso y manejo de la información se restringe en una jerarquía de usuarios. Los sistemas gestores de bases de datos (SGBD) permiten mantener la seguridad mediante el establecimiento de claves para identificar al personal autorizado a utilizar la base de datos. Las autorizaciones se pueden realizar a nivel de operaciones, de modo que un usuario puede estar autorizado a consultar ciertos datos pero no a actualizarlos.

La protección de los datos deberá llevarse a cabo contra fallos físicos, fallos lógicos y fallos humanos (intencionados o no). Estos fallos alteran indebidamente los datos, los corrompen; por lo que la base de datos ya no puede servir a los fines para los que fue creada.

Pudieran tomarse en cuenta algunos aspectos de importancia para lograr mayor seguridad en las bases de datos, algunos de ellos se enuncian a continuación:

- ✓ Protección en profundidad: entre más acciones se tomen para incrementar la protección de la base de datos, menor será la probabilidad de que un atacante tenga éxito, y exponga o abuse de cualquier información secreta que estuviera almacenada.
- ✓ Protección de los ficheros de la base de datos, ya que todos los ficheros almacenados en la base de datos están protegidos contra escritura por cualquier cuenta que no disponga de los permisos necesarios.
- ✓ Las conexiones de los clientes se deben restringir por dirección IP y/o por nombre de usuario.
- ✓ Las conexiones de los clientes pueden ser autenticadas mediante otros paquetes externos.
- ✓ A cada usuario se le asigna un nombre de usuario y (opcionalmente) una contraseña. Por defecto, los usuarios no tienen permiso de escritura a bases de datos que no hayan creado.
- ✓ Los usuarios pueden ser incluidos en grupos, y el acceso a las tablas puede restringirse en base a esos grupos.
- ✓ Evitar que los puertos de la base de datos se encuentren publicados hacia el exterior.
- ✓ Usar las herramientas de seguridad que proporcione el SGBD. Perfiles de usuario, vistas, restricciones de uso de vistas, etc.

1.10 Análisis y descripción de las herramientas a utilizar.

Con el objetivo de lograr seguridad, integridad y consistencia de la información se utilizarán varias herramientas para el diseño y desarrollo de la base de datos del módulo Gobierno y Administración del Sistema para los Tribunales Cubanos.

1.10.1 Sistemas Gestores de Bases de Datos (SGBD)

Ante la notable demanda de soluciones informáticas para la progresiva informatización de todas las organizaciones con la necesidad de optimizar servicios y productos, han surgido diferentes gestores de bases de datos; estos son programas que permiten manejar la información de modo sencillo y que prestan servicios para el desarrollo y el manejo de bases de datos. Los mismos sirven de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan.

Los SGBD ofrecen un control centralizado de la información teniendo como objetivos evitar la redundancia de los datos, mejorar los mecanismos de seguridad de los mismos y la privacidad, mantener la integridad de los datos realizando las validaciones necesarias y mejorar la eficacia del acceso a los datos. En los textos que tratan este tema, o temas relacionados, se mencionan los términos SGBD y DBMS, siendo ambos equivalentes, y acrónimos, respectivamente, de Sistema Gestor de Bases de Datos y DataBase Management System, su expresión inglesa.

Los SGBD se desglosan en:

- ✓ SGBD Libres.

- ✓ SGBD Proprietarios.

Dentro de estos dos grupos se encuentran:

SGBD Proprietarios:

- ✓ Oracle (Oracle Corp.).

- ✓ MySQL (SUN Microsystems).

- ✓ DB2 (IBM).

- ✓ InterBase (CodeGear).
- ✓ MaxDB (SAP).
- ✓ JavaDB (SUN Microsystems).
- ✓ SQL Server (Microsoft Corp.).
- ✓ Infomix (IBM).

SGBD Libres:

- ✓ PostgreSQL.
- ✓ FireBird.
- ✓ SQLite.

Hay que destacar que hasta el año 2008 MySQL formaba parte de los Sistemas Gestores de Base de Datos Libres pero actualmente forma parte de la lista de Gestores de Base de Datos Privativos, aunque el mismo tiene doble licenciamiento.

1.10.1.1 MySQL

Es uno de los más populares sistemas de gestión de bases de datos relacional, utiliza como lenguaje de programación el Structured Query Language (SQL). Es capaz de almacenar una enorme cantidad de datos de gran variedad y de distribuirlos para cubrir las necesidades de cualquier tipo de organización.

MySQL es un software de código abierto, licenciado bajo la GPL de la GNU, aunque MySQL AB distribuye una versión comercial, en lo único que se diferencia de la versión libre, es en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario, ya que de otra manera, se vulneraría la licencia GPL.

Ventajas:

- ✓ Velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento.
- ✓ Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema.
- ✓ Facilidad de configuración e instalación.

- ✓ Soporta gran variedad de Sistemas Operativos.
- ✓ Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.

Inconvenientes:

- ✓ Un gran porcentaje de las utilidades de MySQL no están documentadas.
- ✓ No es intuitivo, como otros programas.
- ✓ No soporta subconsultas.
- ✓ No soporta transacciones.
- ✓ No soporta vistas.
- ✓ Se hace inestable cuando contiene gran cantidad de datos.
- ✓ No soporta transacciones, "roll-backs" ni subselects.
- ✓ No considera las claves ajenas. Ignora la integridad referencial, dejándola en manos del programador de la aplicación.

1.10.1.2 Oracle

Oracle es un sistema de administración de base de datos (o RDBMS Relational Data Base Management System por las siglas en inglés), fabricado por Oracle Corporation, básicamente una herramienta cliente/servidor para la gestión de bases de datos. Es un producto vendido a nivel mundial, aunque la gran potencia que tiene y su elevado precio hacen que sólo se vea en empresas muy grandes y multinacionales, por norma general.

Oracle es sin duda una de las mejores bases de datos que existen en el mercado, es un sistema gestor de base de datos robusto, tiene muchas características que garantizan la seguridad e integridad de los datos; que las transacciones se ejecuten de forma correcta, sin causar inconsistencias.

Ventajas:

- ✓ Oracle ofrece soporte mundial a través de sus centros de soporte y sus sitios Web donde pueden encontrarse desde scripts hasta documentos de instalación.
- ✓ Ayuda a administrar y almacenar grandes volúmenes de datos; estabilidad, escalabilidad y es multiplataforma.
- ✓ Es el motor de base de datos relacional más usado a nivel mundial.

- ✓ Tiene productos para interactuar con otras BD, tales como los Transparent Gateways para: DB2, SQL Server, Informix, Mysql y muchas otras.
- ✓ Ofrece mayor seguridad de la BD con Oracle Virtual Database y Oracle Portal.
- ✓ Puede ejecutarse en todas las plataformas, desde una computadora personal hasta un supercomputador.
- ✓ Permite el uso de particiones para la mejora de la eficiencia, de replicación e incluso ciertas versiones admiten la administración de bases de datos distribuidas.

Inconvenientes:

- ✓ El mayor inconveniente de Oracle es su precio. Incluso las licencias de la Edición Personal de Oracle son excesivamente caras.
- ✓ Necesidad de ajustes. Un error frecuente consiste en pensar que basta instalar el Oracle en un servidor y conectar directamente las aplicaciones clientes. Un Oracle mal configurado puede ser desesperantemente lento.
- ✓ El elevado el coste de la formación. Sólo últimamente han comenzado a aparecer buenos libros sobre asuntos técnicos distintos de la simple instalación y administración.

1.10.1.3 PostgreSQL

PostgreSQL es un potente Sistema Gestor de Base de Datos Relacional libre (Open Source, su código fuente está disponible) liberado bajo licencia Berkeley software Distribución (BSD). Desarrollado en la Universidad de California, en el departamento de ciencias de la computación de Berkeley. Posee más de 15 años de activo desarrollo y arquitectura probada que se ha ganado una muy buena reputación por su confiabilidad e integridad de datos. Funciona en todos los sistemas operativos importantes, incluyendo Linux, UNIX (AIX, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), y Windows.

Principales características:

- ✓ *“Soporta buena parte de la sintaxis SQL, tiene soporte total para llaves extranjeras, uniones, vistas, triggers, y procedimientos almacenados (en múltiples lenguajes).”*

- ✓ *Cliente/Servidor: PostgreSQL usa una arquitectura proceso-por-usuario cliente/servidor. Hay un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intente conectar a PostgreSQL.*
- ✓ *Lenguajes Procedurales: PostgreSQL tiene soporte para lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL/pgSQL. Este lenguaje es comparable al lenguaje procedural de Oracle, PL/SQL. Otra ventaja de PostgreSQL es su habilidad para usar Perl, Python, o TCL como lenguaje procedural embebido, además de en C, C++ y, Java.*
- ✓ *Interfaces con lenguajes de programación: La flexibilidad del API de PostgreSQL ha permitido a los vendedores proporcionar soporte al desarrollo fácilmente para el RDBMS PostgreSQL. Dichas Interfaces incluyen Object Pascal, Python, Perl, PHP, ODBC, Java/JDBC, Ruby, TCL, C/C++, Pike, etc.*
- ✓ *Herencia de tablas.*
- ✓ *Incluye la mayoría de los tipos de datos SQL92 y SQL99 (INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL y TIMESTAMP), soporta almacenamiento de objetos grandes binarios, además de tipos de datos y operaciones geométricas.*
- ✓ *Puntos de recuperación a un momento dado, tablespaces, replicación asincrónica, transacciones jerarquizadas (savepoints), copia de seguridad en línea.*
- ✓ *Un sofisticado analizador/optimizador de consultas.*
- ✓ *Soporta juegos de caracteres internacionales, codificación de caracteres multibyte.” (Ochoa, 2009)*

Inconvenientes:

- ✓ *La velocidad de respuestas que ofrece este gestor es un tanto deficiente.*
- ✓ *Es más lento que los otros gestores de base de datos.*
- ✓ *Consume gran cantidad de recursos cuando se realizan muchas transacciones de datos.*

Ventajas:

- ✓ *“Máximo tamaño de base de datos: ilimitado.*

- ✓ *Máximo tamaño de tabla: 32 TB.*
- ✓ *Máximo tamaño de tupla: 1.6 TB.*
- ✓ *Máximo tamaño de campo: 1 GB.*
- ✓ *Máximo tuplas por tabla: ilimitado.*
- ✓ *Máximo columnas por tabla: 250 - 1600 dependiendo de los tipos de columnas.*
- ✓ *Máximo de índices por tabla: ilimitado". (Ochoa, 2009)*
- ✓ Soporta la configuración automática y la auto-optimización.
- ✓ Administración multiservidor para un gran número de servidores.
- ✓ Gran variedad de opciones de duplicación de cualquier base de datos.
- ✓ Acceso universal a los datos (Universal Data Access).
- ✓ Fácil de usar.

Dadas las ventajas antes mencionadas el proyecto decidió utilizar el PostgreSQL para gestionar la información procesada en el Sistema para los Tribunales Cubanos. Además se tuvo en cuenta la política del país de utilizar tecnologías de código abierto para el desarrollo de software.

1.10.2 Herramientas CASE.

Desarrollar sistemas o herramientas que faciliten encontrar técnicas que permitan incrementar la productividad y el control de la calidad en cualquier proceso de elaboración de software, ha sido el principal área de desarrollo durante las últimas décadas. Las herramientas CASE ⁶ (Computer Aided Software Engineering) reemplazan al papel y al lápiz por el ordenador para transformar la actividad de desarrollar software en un proceso automatizado.

⁶ **Herramientas CASE:** conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un software.

1.10.2.1 Rational Rose

Es una herramienta para el modelado visual de sistemas de software mediante el Lenguaje Unificado de Modelado (UML) manteniendo la consistencia de dichos modelos. Permite el chequeo de la sintaxis UML, genera documentación automáticamente así como código a partir de los modelos y viceversa. Otra ventaja de Rose es que los diseñadores pueden modelar sus componentes e interfaces en forma individual y luego unirlos con otros componentes del proyecto. Además soporta la construcción de componentes en lenguajes como C++, VisualBasic, Java, Ada, genera IDL's para aplicaciones CORBA. Esta herramienta propone la utilización de cuatro tipos de modelos para realizar el diseño del sistema, utilizando una vista estática y otra dinámica de los modelos del sistema, uno lógico y otro físico. Permite crear y refinar estas vistas creando de esta forma un modelo completo que representa el dominio del problema y el sistema de software. Rational Rose utiliza un proceso de desarrollo iterativo controlado, donde el desarrollo se lleva a cabo en una secuencia de iteraciones. Sin embargo esta herramienta no es propiamente para diseño de bases de datos, tiene una instalación costosa y es poco amigable.

1.10.2.2 Visual Paradigm

Visual Paradigm es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software. La herramienta de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales, demostraciones interactivas y proyectos UML. Permite la integración con diversos IDE's como

- ✓ NetBeans (de Sun).
- ✓ JDeveloper (de Oracle).
- ✓ Eclipse (de IBM).
- ✓ JBuilder (de Borland).

Visual Paradigm tiene licencia tanto libre como comercial, es fácil de instalar y utilizar, puede encontrarse en varios idiomas y existe compatibilidad entre ediciones, sin embargo no permite su uso en proyectos comerciales e incluye marca de agua recordando este hecho. Muestra muchas otras funcionalidades no disponibles como gancho para las versiones de pago y las imágenes y reportes generados, no disponen

de buena calidad. Además no es una herramienta destinada específicamente al diseño de bases de datos, no permite el indexado de tablas ni el direccionamiento de datos.

1.10.2.3 Embarcadero Erwin Studio

El ER/Studio es una de las herramientas para la arquitectura y modelación de base de datos más utilizadas en la actualidad, ofrece a los desarrolladores de software y profesionales de la base de datos herramientas premiadas para diseñar, construir, optimizar y ejecutar aplicaciones de software y bases de datos en múltiples plataformas y lenguajes de programación. Provee de mejores reportes y mejor soporte para la implementación de prácticas y estándares en modelos complejos que contienen metadatos críticos. Mejora la calidad del modelo de datos y reduce los riesgos asociados a la seguridad de la información. Es una herramienta multinivel destinada para la construcción desde un modelo lógico de los requerimientos de información, hasta el modelo físico perfeccionado para las características específicas de la base de datos diseñada. Está equipado para crear y manejar diseños de base de datos grandes y complejos. Unido a ello facilita la construcción automática de base de datos, permite realizar ingeniería inversa partiendo del modelo físico al modelo lógico. Brinda mayor soporte en la integración de almacenamiento de datos diseñados para visualizar, documentar y compartir el conocimiento.

Además brinda productividad en diseño, generación, y mantenimiento de aplicaciones. ERwin permite visualizar la estructura, los elementos importantes, y optimizar el diseño de la base de datos. Genera automáticamente las tablas, vistas, reglas de integridad referencial, llaves foráneas, valores por defecto y restricciones de campos y dominios así como miles de líneas de procedimientos almacenados y triggers para los principales tipos de base de datos.

ERwin habilita la creación de un diccionario de atributos reusables, asegurando la consistencia de nombres y definiciones para su base de datos. Se mantienen las vistas de la base de datos como componentes integrados al modelo, permitiendo que los cambios en las tablas sean reflejados automáticamente en las vistas definidas. Por último la migración automática garantiza la integridad referencial de la base de datos.

Todas estas fueron razones que motivaron al equipo de desarrollo del proyecto a seleccionar la herramienta en cuestión para el modelado de la base de datos, teniendo en cuenta además que es ideal para el trabajo en equipo, factor de gran importancia.

1.11 Tendencias actuales y futuras

La utilización de Bases de Datos como plataforma para el desarrollo de Sistemas de Aplicación en las organizaciones se ha incrementado notablemente en nuestros días. Actualmente las BD cuentan con numerosas funcionalidades de gran importancia para el almacenamiento y la manipulación de la información en las instituciones. Esto trae consigo el desarrollo de nuevas técnicas y herramientas que brindan una solución eficiente a las solicitudes planteadas por los usuarios de las mismas. Algunas de estas tendencias actuales y futuras de bases de datos se citan a continuación.

- ✓ La explotación efectiva de la información dará ventaja competitiva a las organizaciones.
- ✓ Se piensa que en el futuro la mayoría de las organizaciones cambiarán la forma convencional de manejo de la información al uso de base de datos debido a las ventajas que ofrecen.
- ✓ El uso de las bases de datos facilita y soporta en gran medida a los sistemas de información para la toma de decisiones.
- ✓ Los lenguajes de consulta permitirán el uso del lenguaje natural para solicitar información de la base de datos, haciendo más rápido y fácil su manejo.
- ✓ El uso de las bases de datos distribuidas se incrementará de manera considerable en la medida en que la tecnología de comunicación de datos brinde más facilidades para ello.

En cuanto a experiencia y resultados tangibles, Microsoft y Oracle Corporation son empresas de avanzada respecto a este tema. Sin embargo desde el punto de vista económico no constituye una opción factible la aplicación de sus herramientas en entidades cubanas, dado los inaccesibles precios de sus licencias. La tendencia hoy en el mercado del software indica una renovadora revolución e interesante propuesta, que convida a inclinarse por el uso de herramientas de software libre, que independientemente de los resultados que se han obtenido resulta innecesario disipar cuantiosas sumas de capital y brinda la posibilidad de desarrollar herramientas propias.

1.12 Conclusiones

El presente capítulo ofreció una panorámica de la situación actual de los sistemas informáticos para instituciones jurídicas en Cuba y el mundo evidenciándose que en el país cuenta con un escaso desarrollo en esta materia.

También se ha realizado un estudio de las bases de datos, enunciándose algunos conceptos importantes al respecto y analizándose temas de importancia como son la seguridad e integridad de la información que las mismas contienen. Se evidenció además la importancia de utilizar como SGBD el PostgreSQL dadas las ventajas que ofrece incluyendo que es una tecnología de código abierto.

Capítulo #2: Descripción de la Solución.

2.1 Introducción

En el presente capítulo se definen las actividades más importantes que posibilitaron la realización del diseño de la base de datos del módulo Gobierno y Administración del proyecto “Sistema para los Tribunales Cubanos” a partir del Modelo de Desarrollo de Bases de Datos, exponiéndose los artefactos generados en la fase de Inicio de dicho modelo.

Se exponen además los requisitos funcionales de mayor relevancia que fueron utilizados para la creación del modelo lógico-conceptual del módulo. Se hará una descripción y fundamentación de la arquitectura de datos y se ofrecerá una descripción detallada de las entidades más importantes en cuestión de funcionalidad para el esquema Gobierno y Administración.

2.2 Argumentación de los requisitos funcionales y no funcionales.

Es de vital importancia para la creación de un sistema informático llevar a cabo una elicitación de requisitos adecuada para llegar a acuerdos positivos entre los usuarios finales y los desarrolladores del sistema, es por ello que la captura de requisitos es tan importante para asegurar la calidad del producto final, que será evaluada por los clientes.

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir. Los no funcionales son propiedades o cualidades que el producto debe tener, permitiendo que el mismo sea atractivo, usable, rápido y confiable.

2.2.1 Requisitos Funcionales del Módulo Gobierno y Administración.

Para el diseño e implementación de la base de datos del módulo Gobierno y Administración se tomaron en cuenta una serie de requisitos funcionales que a continuación se describen brevemente.

- ✓ RF.01 Crear tribunal.

- ✓ RF.02 Modificar tribunal.
- ✓ RF.03 Buscar tribunal.
- ✓ RF.04 Crear sala.
- ✓ RF.05 Modificar sala.
- ✓ RF.06 Crear usuario.
- ✓ RF.07 Modificar usuario.
- ✓ RF.08 Buscar usuario
- ✓ RF.09 Visualizar historial de un usuario.
- ✓ RF.10 Visualizar historial de los usuarios de una o varias instancia.
- ✓ RF.11 Visualizar reporte de usuarios por instancia.
- ✓ RF.12 Visualizar reporte de acciones por usuarios.
- ✓ RF.13 Registrar acción
- ✓ RF.14 Autenticar usuarios.
- ✓ RF.15 Cambiar contraseña de un usuario.

2.2.2 Requisitos no Funcionales.

- ✓ Requerimientos de software:
 - Servidor:
 - Sistema Operativo: Debian 5.0.4.
 - Sistema Gestor de Base de Datos: PostgreSQL 8.4.
 - PC Cliente: Aún están por definir.
- ✓ Requerimientos de hardware:
 - Servidor:
 - Memoria RAM: 2Gb.
 - Discos Duros: 80Gb (RAID 1⁷).
 - PC Cliente:
 - Memoria RAM: 512Mb.

⁷ **RAID 1:** Matriz redundante de discos independientes, es también conocido como "disco espejo" y consiste en que el Sistema Operativo graba los mismos datos simultáneamente en ambos discos, lo que ofrece mayor tolerancia a fallos.

- Discos Duros: 80Gb

- ✓ Requerimientos de seguridad:
 - ❖ Confiabilidad: La información manejada por el sistema está protegida de acceso no autorizado y divulgación.
 - ❖ Integridad: La información manejada por el sistema será objeto de cuidadosa protección contra la corrupción y estados inconsistentes.
 - ❖ Disponibilidad: A los usuarios autorizados se les garantizará el acceso a la información, los dispositivos o mecanismos utilizados para lograr la seguridad no ocultarán o retrasarán a los usuarios para obtener los datos deseados en un momento dado. De acuerdo al tipo de usuario entonces se tendrá disponibilidad sobre determinada información.
- ✓ Requerimientos de usabilidad: El sistema podrá ser usado por las personas autorizadas, debe ser sencillo, fácil de utilizar, manejable, además de contener todos los requisitos necesarios para su funcionamiento y que cumpla con lo pedido por el usuario.
- ✓ Requerimientos de soporte: Se le debe dar mantenimiento periódicamente al servidor de bases de datos controlando la integridad de la información. Se deben realizar pruebas para chequear la calidad del producto. Una vez finalizado el producto deberá continuarse dándole mantenimiento al mismo.

2.3 Arquitectura.

El sistema para los tribunales cubanos cuenta con una estructura de servidores, ubicándose un servidor en cada tribunal independientemente de la instancia en la que se encuentre el mismo. Además existirá un clúster que contendrá unido a las funciones que se realizan en cada tribunal, una estructura de datos general que permita el control en tiempo real de la información procesada en el país proporcionando además integración y una administración única y estandarizada.

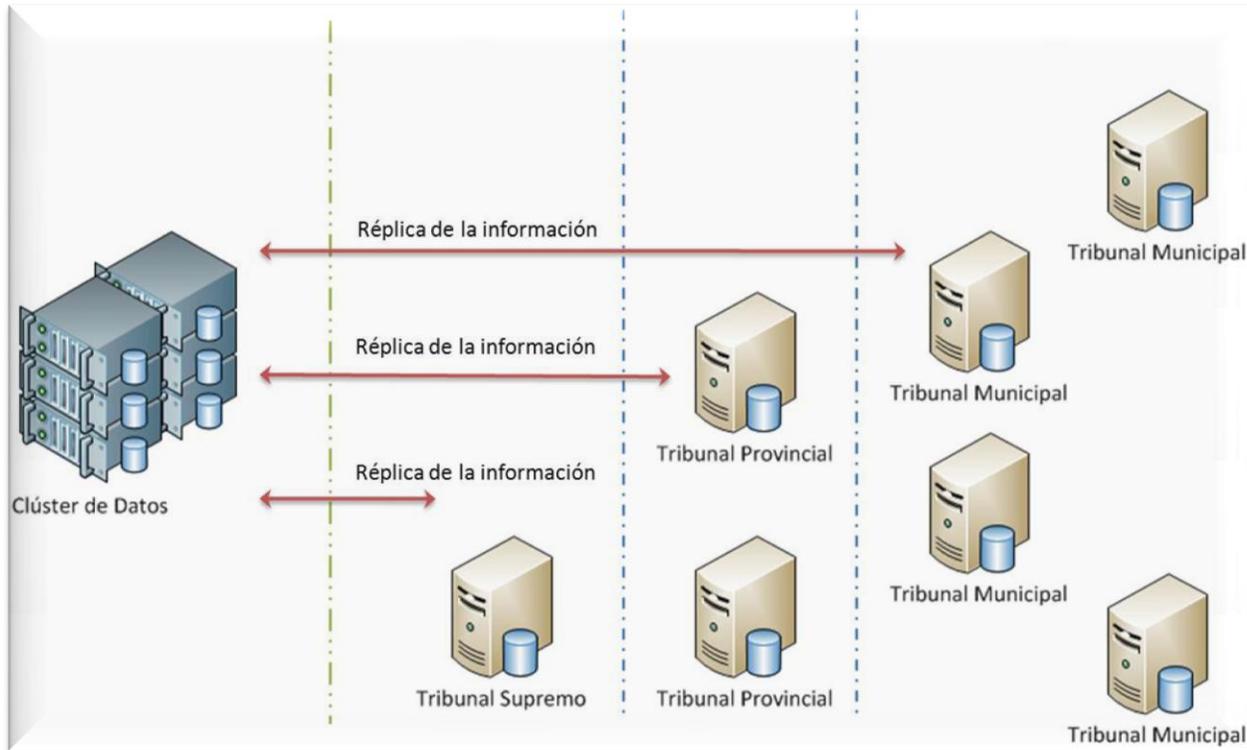


Figura 3: Estructura de servidores.

Como artefacto general del proceso de desarrollo de software se define el documento “Entorno de Desarrollo de Base de Datos”, el cual forma parte del documento de arquitectura del sistema, con el objetivo de organizar el equipo de desarrollo definiendo los responsables por cada módulo. Además contiene la descripción de las herramientas para desarrollar, la propia configuración del entorno de desarrollo, nomenclatura y estándares definidos. Para mayor información consultar el artefacto “Entorno de Desarrollo de Base de Datos”.

2.3.1 Arquitectura de datos.

La base de datos propuesta para el módulo Gobierno y Administración contiene treinta y cinco tablas en modelo físico, las principales tablas son: Tribunal, Persona Natural, Usuario, Dirección, Esquema-Funcionalidad, Acto Procesal y Acto Procesal-Resolución; las cuales recogen la información de mayor peso en el módulo. Dentro de las treinta y cinco tablas existen 25 nomencladores.

2.4 Modelo Lógico - Conceptual del Módulo Gobierno y Administración.

Luego de analizar los requisitos funcionales del módulo Gobierno y Administración, se elaboró el modelo lógico-conceptual de los datos como segundo artefacto que se genera siguiendo el Modelo de Desarrollo de Bases de Datos. El mismo puede estar sujeto a cambios en dependencia de la variabilidad de los requisitos. Ver Anexo 1

Con el objetivo de lograr mayor claridad y organización en el diseño, el esquema "grc" (Gerencial) se subdividió en sub-modelos que serán próximamente descritos.

2.4.1 Tribunal

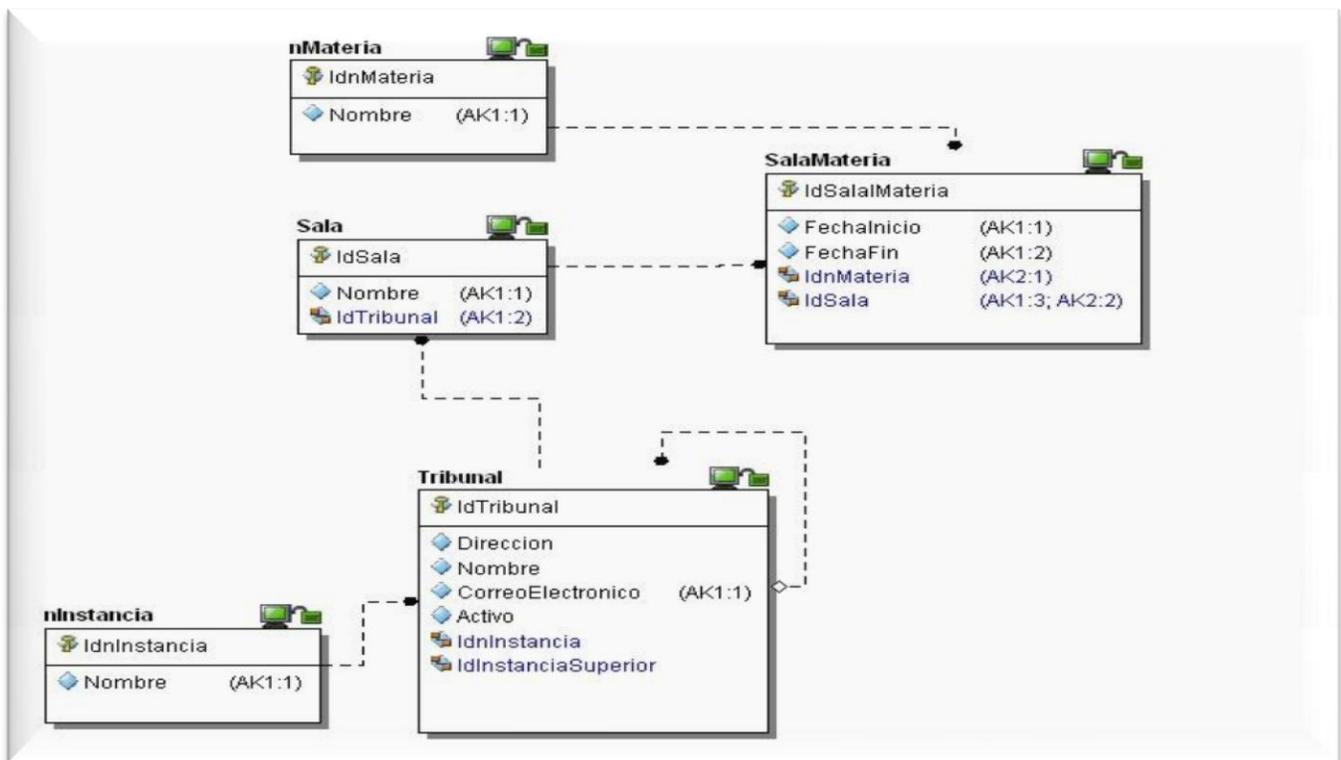


Figura 4: Submodelo Tribunal

Un instancia tiene asociados uno o varios tribunales cada uno de los cuales cuenta con al menos una sala y esta debe tener asociada al menos una materia. Pudiera darse el caso de que una materia se encuentre

ubicada en varias salas o de forma contraria que una sala contenga varias materias, de ahí que surge la tabla SalaMateria, para representar dicha relación.

2.4.2 Roles

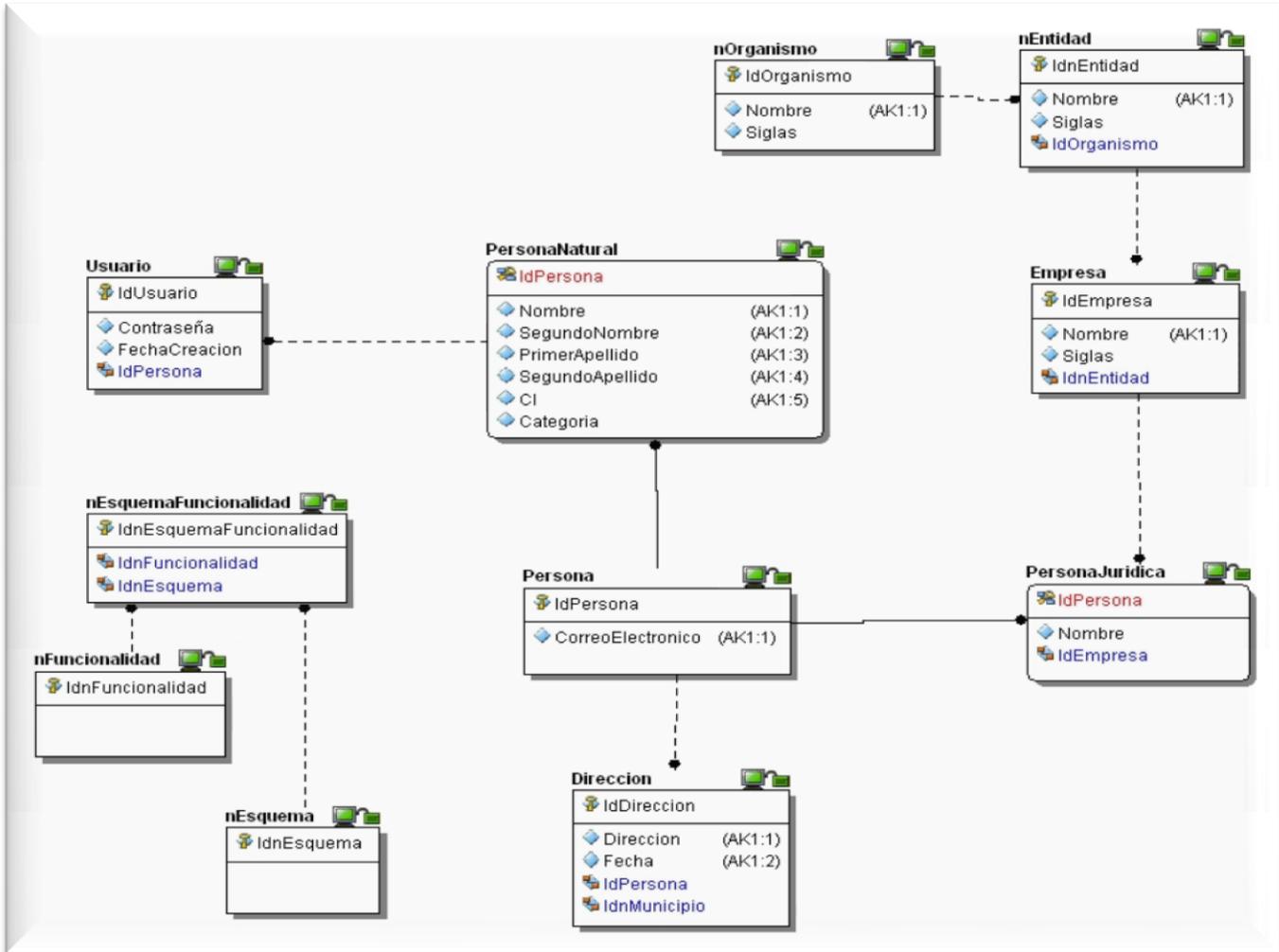


Figura 5: Submodelo Roles.

En la base de datos se modelan personas que pueden ser “Naturales” o “Jurídicas” cada una de las cuales tiene asociada una dirección.

Las personas naturales poseen un usuario. Por su parte las empresas son un tipo de persona jurídica que pertenecen a una determinada entidad y ella a un organismo.

En la base de datos fue modelado un esquema por cada módulo, así es que estos serán recogidos en una tabla nomencladora que se relacionará con otra de su mismo tipo, en la que se guardarán las funcionalidades que se realizan en cada uno de los esquemas.

2.4.3 Gerencial

En este apartado se describe de forma íntegra las entidades que conforman el módulo, incluyéndose las representadas en “Roles” y “Tribunal”.

La siguiente estructura de tablas modela la estructuración de los procesos en un Tribunal Popular Cubano.



Figura 6: Estructuración de procesos.

Dentro de un proceso ya sea ordinario o sumario se realizan una serie de procedimientos que pueden ser de varios tipos, los mismos se relacionan con recursos como son las Apelaciones y las Casaciones⁸.

Por su parte los actos procesales guardan estrecha relación tanto con los procedimientos como con los recursos, pues describen los flujos básico y alterno de ambos. En dependencia del acto procesal que se esté ejecutando se emitirá una resolución judicial que puede ser una Providencia, una Sentencia o un Auto, de esta forma será posible seguir la traza de las acciones que se suceden en un caso.

2.5 Modelo Físico del Módulo Gobierno y Administración.

Una vez diseñado el modelo lógico se obtuvo el modelo físico con el objetivo de convertir la información de forma que pueda ser procesada por el gestor de base de datos. Ver Anexo 2

2.6 Descripción de las principales tablas del diseño de la base de datos.

En esta sección se describen los datos que se almacenan en las tablas más importantes del diagrama entidad relación de la base de datos propuesta, así como una breve descripción de los atributos de cada una de ellas. La nomenclatura de los atributos y tablas se basó en un estándar definido en el proyecto con la finalidad de lograr uniformidad en el modelo. Para mayor información consultar el artefacto “Entorno de desarrollo de bases de datos.”

Tabla 1: Tribunal

Atributos	Dominio	Tipos de Datos	NULL	Descripción
IdTribunal 	Secuencia	SERIAL/INTEGER	NO	Identificador del tribunal.
Nombre	Texto	TEXT	NO	Nombre del tribunal.
CorreoElectronico	Email	TEXT	NO	Correo del tribunal.
Activo	Bool	BIT	NO	Tribunal Activo o no.
IdnInstancia	Entero	INTEGER	NO	Identificador de la

⁸ **Apelaciones y Casaciones:** son recursos que sirven como medios de impugnación. Son una forma de que el acusado manifieste su inconformidad con la sentencia dictada, resolviéndose esta en otro tribunal.

				instancia a la que pertenece el tribunal.
Direccion	Texto	TEXT	NO	Dirección del tribunal.
IdInstanciaSuperior	Entero	INTEGER	YES	Identificador de la instancia superior a la que pertenece el tribunal.

Tabla 2: PersonaNatural

Atributos	Dominio	Tipo de Datos	NULL	Descripción
IdPersona 	Entero	INTEGER	NO	Identificador de la persona.
Nombre	Texto	TEXT	NO	Nombre de la persona.
SegundoNombre	Texto	TEXT	YES	Segundo nombre de la persona.
PrimerApellido	Texto	TEXT	NO	Primer apellido de la persona.
SegundoApellido	Texto	TEXT	NO	Segundo apellido de la persona.
CI	NumeroCI	VARCHAR(11)	NO	Carnet de Identidad de la persona.
Categoria	Texto	TEXT	YES	Categoría de la persona.

Tabla 3: Usuario

Atributos	Dominio	Tipo de Datos	NULL	Descripción
IdUsuario 	Texto	TEXT	NO	Identificador del usuario.
Contraseña	Contraseña	TEXT	NO	Contraseña del usuario.
FechaCreacion	FechaHora	DATETIME	NO	Fecha de creación del usuario.
IdPersona	Entero	INTEGER	NO	Identificador de la persona a la que pertenece el usuario.

Tabla 4: Direccion

Atributos	Dominio	Tipo de Datos	NULL	Descripción
-----------	---------	---------------	------	-------------

IdDireccion 		SERIAL/INTEGER	NO	Identificador de la dirección.
Direccion	Texto	TEXT	NO	Última dirección conocida.
Fecha	FechaHora	DATETIME	NO	Fecha de la última dirección conocida.
IdPersona	Entero	INTEGER	NO	Identificador de la persona que tiene asociada la dirección.
IdnMunicipio	Entero	INTEGER	YES	Identificador del municipio al que pertenece la dirección.

Tabla 5: nEsquemaFuncionalidad

Atributos	Dominio	Tipo de Datos	NULL	Descripción
IdnEsquemaFuncionalidad 	Entero	INTEGER	NO	Identificador.
IdnFuncionalidad	Texto	TEXT	NO	Identificador de la funcionalidad.
IdnEsquema	Texto	TEXT	NO	Identificador del Esquema al que pertenece la funcionalidad.

Tabla 6: nActoProcesal

Atributos	Dominio	Tipo de Datos	NULL	Descripción
IdnActoProcesal 	Entero	INTEGER	NO	Identificador del acto procesal.
Nombre	Texto	TEXT	NO	Nombre del acto procesal.
Orden	Entero	INTEGER	NO	Orden del acto procesal.
IdnTipoProcedimiento	Entero	INTEGER	YES	Identificador del tipo de procedimiento al que corresponde el acto procesal.

IdnRecurso	Entero	INTEGER	YES	Identificador del recurso al que corresponde el acto procesal.
-------------------	--------	---------	-----	--

Tabla 7: nActoProcesalResolucion

Atributos	Dominio	Tipo de Datos	NULL	Descripción
IdnActoProcesalResolucion 	Entero	INTEGER	NO	Identificador.
IdnActoProcesal	Entero	INTEGER	NO	Identificador del acto procesal correspondiente a la resolución.
IdnTipoResolucionJudicial	Entero	INTEGER	NO	Identificador de la resolución correspondiente al acto procesal.

2.7 Configuraciones

Como parte de las actividades que se realizan en la presente fase se encuentra el establecimiento de las configuraciones iniciales del servidor de base de datos, así como la estructuración de una estrategia de instalación y configuración de la arquitectura de la base de datos para diferentes entornos de desarrollo. El artefacto que se genera es el Documento de configuraciones del sistema operativo – sistema gestor de bases de datos – base de datos. Dicho artefacto contiene los principales elementos a tener en cuenta en el momento de la instalación de los programas necesarios y una vez concluida la misma.

En el servidor se instalará el sistema operativo Debian en su versión 5.0.4, el mismo es un sistema operativo libre, de ahí que su utilización esté en correspondencia con las políticas de la universidad y el país con respecto al desarrollo de software a través de tecnologías de código abierto. Una vez finalizada la instalación deberán configurarse debidamente la hora y la dirección IP del servidor, lo que garantizará un mejor funcionamiento. De igual forma debe conectarse al repositorio de actualizaciones del sistema

operativo de donde podrán obtenerse, en su mayoría, los programas y servicios indispensables para el servidor. Un aspecto que no debe quedar al margen es la instalación del servicio Openssh-sever, el mismo facilita la administración remota por consola del servidor.

Otro software que debe quedar instalado para el trabajo con la base de datos es el SGBD PostgreSQL 8.4. En el mismo como medida de seguridad se eliminará el esquema public que por defecto trae asociado, creándose en su lugar un esquema para cada materia, lo que contribuirá a lograr una mayor organización de la base de datos. Cada uno de los esquemas contará con una cuenta de usuario propia. Para mayor información remitirse al artefacto “Documento de configuraciones del sistema operativo – sistema gestor de bases de datos – base de datos”.

2.8 Implementación

Otra actividad desarrollada es la implementación de todas las funcionalidades requeridas en la base de datos. Estas quedaron documentadas en el artefacto “Codificación de las Funcionalidades”. Además quedaron registradas en el backup de la base de datos.

Las funciones implementadas responden a los requisitos funcionales del sistema. Se ha utilizado como lenguaje de codificación plpgsql⁹. Hasta el momento algunas de las funcionalidades a las que se les ha dado cumplimiento son:

1. Crear Tribunal: Será almacenada información referente a un tribunal independientemente de la instancia a la que pertenezca el mismo.

```
CREATE OR REPLACE FUNCTION grc."CrearTribunal"("pNombre" text, "pCorreoElectronico" text, "pActivo" boolean, "pNombreInstancia" text, "pldInstanciaSuperior" integer, "pDireccion" text)
```

```
RETURNS void AS
```

```
$BODY$begin
```

⁹ **Plpgsql**: Lenguaje procedural cargable para el sistema de bases de datos PostgreSQL.

```
INSERT INTO grc."TRIBUNAL"("NOMBRE", "CORREOELECTRONICO", "ACTIVO", "IDNINSTANCIA", "IDINSTANCIASUPERIOR", "DIRECCION")
```

```
VALUES ($1, $2, $3,(SELECT "IDNINSTANCIA" FROM grc."NINSTANCIA" WHERE grc."NINSTANCIA"."NOMBRE"=$4),$5,$6);
```

```
end;$BODY$
```

```
LANGUAGE 'plpgsql' VOLATILE
```

```
COST 100;
```

```
ALTER FUNCTION grc."CrearTribunal"(text, text, boolean, text, integer, text) OWNER TO postgres;
```

2. Crear Usuario: En esta función se recogerán todos los datos necesarios para la creación de un usuario actualizando las tablas correspondientes según el modelo realizado.

```
CREATE OR REPLACE FUNCTION grc."CrearUsuario"("pCorreoElectronico" text, "pNombre" text, "pSegundoNombre" text, "pPrimerApellido" text, "pSegundoApellido" text, "pCI" text, "pCategoria" text, "pIdUsuario" text, "pContraseña" text, "pFechaCreacion" timestamp with time zone, "pDireccion" text, "pFecha" timestamp with time zone, "pldnMunicipio" integer)
```

```
RETURNS void AS
```

```
$BODY$declare
```

```
id integer;
```

```
begin
```

```
INSERT INTO grc."PERSONA"("CORREOELECTRONICO") VALUES ($1);
```

```
select into id curval('grc."PERSONA_IDPERSONA_seq"');
```

```
INSERT INTO grc."PERSONANATURAL"("IDPERSONA", "NOMBRE", "SEGUNDONOMBRE", "PRIMERAPELLIDO", "SEGUNDOAPELLIDO", "CI", "CATEGORIA") VALUES (id, $2, $3,$4,$5,$6,$7);
```

```
INSERT INTO grc."USUARIO"("IDUSUARIO","CONTRASEÑA", "FECHACREACION", "IDPERSONA")
```

```
VALUES ($8,$9,$10,id);
```

```
INSERT INTO grc."DIRECCION"("DIRECCION", "FECHA", "IDPERSONA", "IDNMUNICIPIO")
```

```
VALUES ($11, $12,id,$13);
```

```
end;$BODY$
```

```
LANGUAGE 'plpgsql' VOLATILE
```

```
COST 100;
```

```
ALTER FUNCTION grc."CrearUsuario"(text, text, text, text, text, text, text, text, timestamp with time zone, text, timestamp with time zone, integer) OWNER TO postgres;
```

3. Autenticarse: La presente función permite la autenticación de los usuarios en el sistema. Podrán obtenerse 3 posibles valores: 0 si los datos introducidos son correctos, 1 si el usuario es incorrecto y 2 si la contraseña es incorrecta.

```
CREATE OR REPLACE FUNCTION grc."Autenticar"("pUsuario" text, "pContrasenna" text) RETURNS integer AS
```

```
$BODY$declare
```

```
usuario text;
```

```
contrasenna text;
```

```
begin
```

```
SELECT INTO usuario,contrasenna "IDUSUARIO","CONTRASEÑA" FROM grc."USUARIO" WHERE  
grc."USUARIO"."IDUSUARIO"=$1 ;
```

```
IF usuario IS NULL THEN
```

```
RETURN 1;
```

```
ELSE
```

```
IF contrasenna <> $2 THEN
```

```
RETURN 2;
```

```
        ELSE

        RETURN 0;

        END IF;

    END IF;

end;$BODY$

LANGUAGE 'plpgsql' VOLATILE

COST 100;

ALTER FUNCTION grc."Autenticar"(text, text) OWNER TO postgres;
```

4. Buscar Usuario: Debe ofrecerse la posibilidad de buscar un usuario, la búsqueda se podrá realizar atendiendo a 2 criterios, el primero de ellos es con el ID del usuario en cuestión, la segunda introduciendo el nombre y el primer apellido del mismo. Se obtendrá como resultado todos los datos del usuario si este es encontrado.

```
CREATE OR REPLACE FUNCTION grc."BuscarUsuario"("pNombre" text, "pApellido" text, "pIdUsuario" text)

    RETURNS refcursor AS

$BODY$declare

cur refcursor;

begin

open cur for select * from grc."USUARIO" t1,grc."PERSONANATURAL" t2

    where (t1."IDPERSONA" = t2."IDPERSONA")

    AND ($1 is NULL or t2."NOMBRE" = $1 )

    AND ($2 is NULL or t2."PRIMERPELLIDO" = $2)

    AND ($3 is NULL or t1."IDUSUARIO" = $3);
```

```
return cur;

end; $BODY$

LANGUAGE 'plpgsql' VOLATILE

COST 100;

ALTER FUNCTION grc."BuscarUsuario"(text, text, text) OWNER TO postgres;

COMMENT ON FUNCTION grc."BuscarUsuario"(text, text, text) IS 'Busca por nombre y apellidos';
```

Para mayor información consultar el artefacto “Codificación de las Funcionalidades”.

2.9 Conclusiones

Se ha presentado la descripción de la solución desarrollada para la base de datos del módulo Gobierno y Administración. El diseño del modelo propuesto se basó en los requisitos funcionales. Las tablas se diseñaron de forma tal que el tiempo de duración de las consultas fuera el menor posible y fueron construidas de manera que no exista redundancia en los datos, aumentando así la integridad de la información, lo cual es uno de los objetivos principales de la base de datos. Además fueron realizadas las configuraciones pertinentes del servidor de base de datos así como la implementación de algunas funcionalidades de importancia.

Capítulo #3: Validación de la Solución.

3.1 Introducción

Durante el proceso de desarrollo de un software, los errores pueden comenzar a aparecer incluso desde el mismo momento en que fueron definidos los objetivos y estos a su vez especificados de forma errónea e imperfecta, de la misma forma en los posteriores pasos del diseño y desarrollo.

En el capítulo anterior se planteó una propuesta de solución para el diseño e implementación de la Base de Datos del módulo Gobierno y Administración del proyecto “Sistema para los Tribunales Cubanos”, en el presente capítulo se realiza la validación de la misma, abordando aspectos como la integridad, seguridad y normalización de la BD. Es objetivo de este capítulo validar el diseño realizado, así como el impacto que ha tenido la utilización del modelo de desarrollo de la base de datos al que se hace referencia.

3.2 Validación teórica del diseño

Se hace imprescindible lograr un correcto diseño de la base de datos. Debe tenerse en cuenta que un buen diseño debe garantizar sobre todo la consistencia y seguridad de los datos, la no redundancia de los mismos, y que no ocurran, en general, las llamadas “anomalías de actualización”. Para ello deben considerarse elementos como las restricciones de integridad, la normalización de la base de datos y la seguridad de la misma.

3.2.2 Restricciones de integridad de la BD.

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

La integridad de datos pertenece a una de las siguientes categorías:

- ✓ Integridad de entidad: la clave primaria de una entidad no puede tener valores nulos y siempre deberá ser única, por ejemplo el CI en la tabla PersonaNatural.
- ✓ Integridad de dominio: restringe los valores que puede tomar un atributo respecto a su dominio, por ejemplo, la tabla nMateria sólo puede tomar los valores de las 5 materias que pueden existir en un tribunal (Económico, Administrativo, Laboral, Civil, Penal).
- ✓ Integridad referencial: La base de datos no debe contener valores de llaves ajenas sin concordancia.
- ✓ Datos Requeridos: establece que una columna tenga un valor no NULL.

Integridad de Entidad

Esta regla se aplica a las claves primarias de las relaciones base: ningún atributo que forme parte de una llave primaria puede aceptar valores nulos. La unicidad de la clave primaria se refiere a que dos tuplas no pueden ofrecer el mismo valor para la clave primaria. Para expresar estas restricciones pueden utilizarse determinados elementos del lenguaje SQL como PRIMARY KEY, y UNIQUE. Este último para expresar la unicidad los atributos que puedan constituir claves alternativas.

Es por esto que en muchas de las tablas de cada uno de los esquemas de la base de datos del sistema para los Tribunales Populares Cubanos generan las llaves de forma automática usando secuencias, garantizando así que no se cometan errores, uniformidad en la creación de la llaves, además el principal objetivo de la existencia de las mismas está dado por el tema de las réplicas entre los servidores.

Integridad de Dominio

La integridad de dominio viene dada por el conjunto de valores posibles para un atributo determinado. Puede exigir la integridad de dominio restringiendo los valores de un atributo mediante tipos de datos, reglas y restricciones CHECK, definiciones DEFAULT, definiciones NOT NULL. Un dominio de valores puede estar asociado a cada atributo. Los límites de dominio son la forma más elemental de restricciones de integridad. Son fáciles de probar en el sistema siempre que se introduce un nuevo dato.

Un ejemplo ilustrativo de algunas de las restricciones de dominio implementadas se expone a continuación, indicando el dominio, tipo de dato correspondiente al mismo, campos a los que será aplicado así como la restricción perteneciente a cada uno de ellos. Para mayor información consultar el artefacto “Entorno de Desarrollo de Bases de Datos”.

Dominio	Tipo de Dato	Campo	Restricción
Binario	BINARY	Documentos, imágenes, etc.	
Bool	BIT	Verdadero o Falso	@var = TRUE or @var = FALSE
Contraseña	TEXT	Contraseña de usuario	@var like '%_____%'
EMail	TEXT	Correo electrónico	@var like '%_@__%'
Entero	INTEGER	Números enteros positivos	@var > 0
FechaHora	DATETIME	Fecha y hora	A nivel de tabla comparando las fechas: FechaInicio <= FechaFin
Nombre	VARCHAR(30)	Nombre hasta 30 caracteres	
Numero	NUMERIC	Valores numéricos y decimales	@var > 0
NumeroCI	VARCHAR	Número de carnet de identidad	@var like '_____'
Texto	TEXT	Texto en general	

Tabla 8: Restricciones de dominio.

Integridad Referencial

La integridad referencial significa que la clave externa de una tabla de referencia siempre debe aludir a una fila válida de la tabla a la que se haga referencia, garantizando que los valores de clave sean coherentes en las distintas tablas. Para conseguir esa coherencia, es preciso que no haya referencias a valores inexistentes y que, si cambia el valor de una clave, todas las referencias a ella se cambien en consecuencia en toda la base de datos.

La integridad referencial permite que la relación entre dos tablas permanezca sincronizada durante las operaciones de actualización y eliminación, para lograrlo se utilizan las sentencias SQL ON DELETE

CASCADE o RESTRICT y ON UPDATE CASCADE o RESTRICT en dependencia de la relación que exista entre las tablas implicadas.

La integridad referencial debe proporcionar Durabilidad, Aislamiento, Consistencia e Indivisibilidad a la base de datos.

- ✓ Indivisibilidad: es la propiedad que asegura que la operación se ha realizado o no, y por lo tanto ante un fallo del sistema no puede quedar a medias.
- ✓ Consistencia: es la propiedad que asegura que sólo se empieza aquello que se puede acabar. Por lo tanto se ejecutan aquellas operaciones que no van a romper la reglas y directrices de integridad de la base de datos.
- ✓ Aislamiento: es la propiedad que asegura que una operación no puede afectar a otras. Esto asegura que dos transacciones sobre la misma información nunca generarán ningún tipo de error.
- ✓ Durabilidad: es la propiedad que asegura que una vez realizada la operación, ésta persistirá y no se podrá deshacer aunque falle el sistema.

Un ejemplo de lo antes expuesto es que se tiene una tabla nInstancia quien tiene como llave primaria IdnInstancia, además se encuentra la tabla Tribunal quien tiene como llave primaria IdTribunal. La relación que se establece entre estas tablas es de uno a muchos (ya que una instancia puede tener asociados varios tribunales pero estos sólo podrán pertenecer a una instancia) razón por la cual IdnInstancia pasa para la tabla Tribunal como llave foránea. De esta forma se conocería a que instancia pertenece un determinado tribunal.

Datos Requeridos

Se define efectuando que la declaración de una columna es NOT NULL cuando la tabla que contiene las columnas se crea por primera vez, como parte de la sentencia CREATE TABLE.

En la mayoría de los casos resultaría inútil declarar que un campo acepte valores nulos si se quiere guardar información importante en el mismo, sin embargo en otros casos resulta necesario, por ejemplo, la tabla Tribunal contiene el atributo IdInstanciaSuperior, el mismo acepta valores nulos pues en el caso del Tribunal Supremo, no existe una instancia superior.

3.2.3 Análisis de redundancia de la información.

El diseño de una BD debe ser realizado cuidadosamente, procurando permitir un fácil acceso a la información, facilitando un alto rendimiento, una buena velocidad de respuesta y una gran consistencia de los datos. Un factor que atenta contra esto es la redundancia de los datos que se refiere a la información

duplicada en la base de datos, esta genera problemas como: incremento del trabajo y desperdicio de espacio de almacenamiento.

Con el propósito de reducir en la mayor medida posible la redundancia de la información almacenada, la base de datos propuesta fue sometida a un proceso de normalización, lográndose que cada tabla disponga solo de los atributos que necesite.

3.2.4 Normalización de la Base de Datos.

Cuando se va a realizar el diseño de una base de datos relacional, se debe considerar que las tablas que la componen cumplan con las reglas de normalización, que no es más que el proceso que permite eliminar la redundancia de los datos y evitar los problemas al insertar, eliminar y actualizar los datos, es decir, optimizar el trabajo con la información almacenada. El grado de normalización no es el único criterio para calificar lo bueno que es un esquema relacional. En este proceso se debe ir comprobando que cada relación (tabla) cumple una serie de reglas que se basan en la clave primaria y las dependencias funcionales. Cada regla que se cumple aumenta el grado de normalización. Si una regla no se cumple, la relación se debe descomponer en varias relaciones que sí la cumplan.

En el proceso de normalización existen varios niveles, pero los tres primeros son los más usados. Las reglas de normalización se relacionan entre ellas de forma dependiente, lo que indica que para que una base de datos se encuentre en una determinada forma normal, primeramente tiene que cumplir con las reglas de los niveles inferiores de normalización. Esto quiere decir que una base de datos está en 2da Forma Normal si previamente cumple con las reglas de normalización del 1er nivel y además cumple con las restricciones correspondientes a dicha forma normal.

El esquema de relación del módulo está en 1ra Forma Normal puesto que se puede asegurar que cada tupla contiene exactamente un valor para cada atributo de las tablas de la base de datos, o sea, no existen campos multivaluados. También se cumple que el esquema de relación está en 2da Forma Normal, pues primeramente se encuentran en 1ra Forma Normal, y además todos los atributos que no son claves en las tablas, dependen totalmente de la clave primaria. Finalmente puede plantearse que el esquema de relación se encuentran en 3ra Forma Normal, pues se encuentra en 2da Forma Normal, y además no existen dependencias transitivas atributos no primos. Se puede afirmar entonces que el diseño del módulo de la base de datos propuesto se encuentra

normalizado hasta Tercera Forma Normal, lográndose de esta forma mayor independencia entre los datos y evitándose anomalías al trabajar con los datos.

3.2.5 Seguridad

Las bases de datos están bajo constante amenaza de sufrir ataques de personas o sistemas no autorizados que ponen en riesgo su integridad y confidencialidad; es por ello que deben tomarse una serie de medidas que impidan estos sucesos y permitan conservar salvadas de la BD para restaurar los datos si se pierden o corrompen por alguna razón pues ellos constituyen un recurso valioso que debe ser estrictamente controlado y gestionado al igual que cualquier otro recurso corporativo.

Los sistemas gestores de bases de datos permiten definir autorizaciones o derechos de acceso teniendo en cuenta los usuarios, ubicaciones desde donde se puede acceder, así como asignar privilegios que tendrán los usuarios una vez autenticados. Lo primero que debe garantizarse es que sólo puedan acceder a los datos los usuarios autorizados.

PostgreSQL, permite configurar los permisos de los usuarios utilizando tres niveles de acceso. En el primer nivel se configuran los permisos de conexión para los host y los usuarios a la o las BD en el archivo `pg_hba.conf`, se define qué dirección o direcciones IP tendrán acceso a cuál o cuáles BD, y en qué modo podrán conectarse: conexión sin contraseña, validando el usuario y la contraseña o que rechace cualquier conexión desde el IP o rangos IP y usuarios seleccionados. Para mayor información consultar artefacto “Documento de configuraciones del sistema operativo – sistema gestor de bases de datos – base de datos”.

En el segundo nivel, se configuran a qué BD pueden acceder determinados usuarios, utilizando las opciones del archivo `pg_ident.conf`. En el tercer nivel permite configurar los accesos de los usuarios a las tablas de la BD, utilizando las sentencias `GRANT` y `REVOKE`, se pueden asignar o denegar respectivamente los permisos para insertar, eliminar y actualizar datos, entre otros. En la BD propuesta se establecen controles de seguridad para los datos almacenados, garantizando que sólo los usuarios autorizados puedan efectuar operaciones correctas sobre algunas tablas o sobre la BD. Las contraseñas de los usuarios, tanto los de la aplicación como el administrador de la BD estarán encriptadas, para evitar que intrusos se apoderen de ellas con malos fines.

Además, este SGBD puede ejecutarse en dos modos: `fsync` y `no fsync`. Si se ejecuta en el primer modo el gestor sería más lento, pero garantizaría que si el sistema operativo se bloquea o se produce una pérdida

de energía, todos sus datos estarían almacenados y sin daños en el disco. El modo de ejecución de PostgreSQL en el servidor de BD de la aplicación será fsync, para evitar pérdida de la información en caso de que ocurra algún error inevitable en el sistema.

3.3 Modelo de Desarrollo de Bases de Datos.

La desorganización en el desarrollo de las bases de datos es un factor que hoy en día golpea el desarrollo ágil y eficiente de las mismas pues no existe una metodología o teoría escrita y probada de desarrollo de bases de datos que encamine cómo llevar el desarrollo de una forma estándar para mejorar el mismo de principio a fin.

Con el objetivo de validar el desarrollo de la base de datos en la fase de inicio según el modelo seguido, y mediante la identificación de la estructura y organización del desarrollo de bases de datos en varios proyectos de la facultad, así como artefactos identificados y el avance según las fases o iteraciones dentro del ciclo de desarrollo, se realizó una entrevista a las personas responsables de las bases de datos en los mismos. Ver anexo 3

De un total de 9 proyectos de la facultad 15 que fueron visitados, se tomaron en consideración 5 por tener más de un año de trabajo que es el tiempo que lleva laborando el proyecto “Sistema para los Tribunales Cubanos”.

La entrevista realizada arrojó los siguientes resultados:

Aspectos \ Proyectos	Aduana	Banco	SGF (Sistema de Gestión Fiscal)	RN (Registros y Notarías)	ERP (Planificación de Recursos Empresariales)
Modelo para el desarrollo.	No	No	No	No	Sí
Organización del equipo de desarrollo	Sí	No	Sí	Sí	Sí
Artefactos	Sí	Sí	Sí	Sí	Sí
Fases de desarrollo	No	No	No	No	Sí

Tabla 9: Resultados de la entrevista realizada.

Mediante la comparación establecida se pudo arribar a las siguientes conclusiones:

1. La mayoría de los proyectos carecen de un modelo que guíe el desarrollo de la BD, solo utilizan un estándar de nomenclatura.
2. Por lo general el equipo de desarrollo se encuentra organizado en cada proyecto.
3. En todos los proyectos se genera al menos un artefacto que generalmente es el modelo de datos, sin embargo no existe documentación suficiente respecto al avance del trabajo que se realiza. Tampoco quedan documentados aspectos de importancia que no se encuentren precisamente relacionados con el diseño como son las configuraciones necesarias para el correcto funcionamiento del servidor de base de datos, detalles de implementación etc.
4. El trabajo con la base de datos no se encuentra organizado por fases de desarrollo que ofrezcan una medida del avance alcanzado; que permitan al equipo conocer en qué punto del desarrollo se encuentra el proyecto respecto a este tema así como que establezcan una organización lógica de las tareas a realizar. Solamente el proyecto ERP contempla este aspecto.

Hasta el momento ha quedado demostrado que la mayoría de los proyectos de la facultad 15 no utilizan ningún modelo que guíe y estructure el desarrollo de la base de datos. En el caso del proyecto “Sistema para los Tribunales Cubanos”, se emplea el “Modelo de Desarrollo de Bases de Datos para Procesos de Desarrollo de Software” lo que permitió que se ganara en cuanto a organización y rapidez del trabajo, factores imprescindibles para el desarrollo de software.

Un conjunto de 3 indicadores han sido evaluados para medir el impacto que produce la utilización del modelo en cuestión: nivel de organización, cantidad de artefactos identificados y porcentaje de avance en el desarrollo. El estudio realizado arrojó los siguientes resultados:

Nivel de Organización.



Figura 7: Nivel de Organización.

En cuanto a este indicador el modelo asegura la organización del trabajo con la base de datos en el proyecto mediante fases y una serie de actividades a realizar en cada una de ellas, lo que ha propiciado que los resultados obtenidos hasta el momento sean satisfactorios, pues el equipo de desarrollo tiene claridad respecto a las tareas que debe realizar y en qué momento debe llevarlas a cabo; además se encuentra organizado de manera que cada rol conozca cuáles son sus responsabilidades.

Artefactos Identificados.

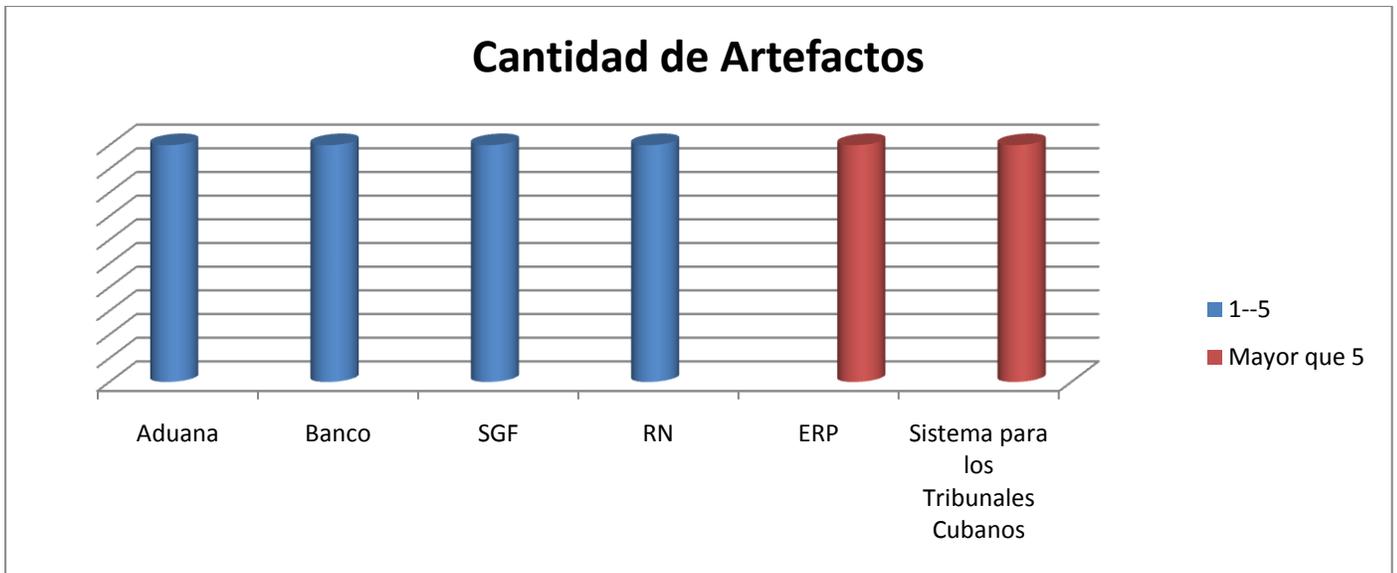


Figura 8: Cantidad de Artefactos.

En el proyecto mediante la aplicación del modelo se han identificado 6 artefactos que se encuentran correctamente documentados. El resto de los proyectos entrevistados, exceptuando el ERP, cuentan

únicamente con el “Modelo de Datos” y en ocasiones no se dispone de una detallada documentación del mismo, lo que influye negativamente en el trabajo pues el equipo de desarrollo puede necesitar un material de consulta para entender y trabajar con el modelo de datos. Esta carencia de información conduce a la pérdida de tiempo durante el ciclo de desarrollo de la base de datos. El proyecto Sistema para los Tribunales Cubanos de un total de 6 artefactos identificados tiene desarrollados más del 75 % de los mismos en correspondencia con las actividades que se encuentran en ejecución.

Resulta de importancia señalar que aún cuando la cantidad de artefactos generados no define la calidad del modelo utilizado, expresan el grado de avance alcanzado en cuanto a aspectos básicos y necesarios para el desarrollo de la base de datos.

Porcentaje de avance en el desarrollo.

Como resultado relevante de la entrevista realizada se obtuvo que la mayoría de los proyectos comienzan el modelado de los datos una vez concluida la elicitación de requisitos, no siendo así en el caso del proyecto Sistema para los Tribunales Cubanos en el que se comenzó desde el momento mismo en que se realizó la descripción del negocio, lo que le permitió ganar en tiempo en el cumplimiento de esta actividad. Fue este uno de los factores que contribuyó a que aún encontrándose en la fase inicial del desarrollo de la base de datos se cuente con alrededor del 80% del modelo de datos completado para el módulo Gobierno y Administración a pesar de que el mismo se encuentra sujeto a cambios.

Además se encuentra listo el artefacto “Documento de configuraciones del sistema operativo – sistema gestor de bases de datos – base de datos” generado en la actividad Configuraciones, la misma, a pesar de que se realiza en la totalidad de los proyectos no se tiene concebida su documentación.

Los artefactos generados fueron liberados por el grupo de calidad de la facultad. Ver anexo 4

3.4 Resultados preliminares.

Una vez concluido el diseño y la implementación de la base de datos del sistema para los tribunales cubanos se obtuvo los resultados siguientes:

- ✓ La base de datos cumple con los requisitos funcionales.
- ✓ Las relaciones entre las tablas fueron construidas adecuadamente.
- ✓ Fue alcanzado el tercer nivel de normalización.
- ✓ Las restricciones de integridad garantizan que los datos introducidos o modificados sean los correctos.

- ✓ La redundancia en la información se ha reducido en gran medida.
- ✓ Fueron obtenidos los artefactos de las actividades de modelado de datos, configuración e implementación para la fase de inicio.
- ✓ El modelo de desarrollo de base de datos utilizado ofreció además de gran rapidez, organización tanto del equipo de desarrollo como del trabajo realizado.

3.5 Conclusiones

La validación del diseño culminó con la valoración de aspectos importantes como la integridad de la información almacenada, definiéndose varias restricciones que permitieran garantizar la validez de la misma. Además fueron tratadas la seguridad y la normalización de la Base de Datos. La validación se efectuó poniendo en práctica las experiencias adquiridas como diseñadores de BD, sobre la base de los requerimientos del negocio.

También fue corroborado el impacto del Modelo de Desarrollo de Bases de Datos para procesos de desarrollo de software utilizado para el trabajo con la base de datos propuesta.

Conclusiones Generales

La organización del equipo de desarrollo, en proyectos de software, es un factor decisivo para el éxito del mismo. Actualmente, en el caso del desarrollo de bases de datos es una responsabilidad del propio equipo pues las metodologías de desarrollo de software no plantean estos elementos organizativos respecto este tema. Es por ello que resulta de gran utilidad contar con un modelo que guíe al desarrollador durante todo el proceso, que indique los elementos, técnicas y procedimientos a seguir.

El modelo utilizado estructura el desarrollo de bases de datos de forma tal que puede ser aplicado en cualquier entorno de desarrollo sin que sea un impedimento determinada metodología, sistema gestor de base de datos o herramientas utilizadas, ya que organiza los elementos que deben estar presentes a la hora de desarrollo de bases de datos de forma general. Por otro lado permite la organización tanto de las actividades como de los recursos humanos de los cuales se necesita, permitiendo desarrollar más rápidamente y con calidad.

Apoyados en el modelo antes mencionado y con la ayuda de la herramienta Embarcadero Erwin Studio para el modelado de la base de datos, así como el sistema gestor de bases de datos PostgreSQL se obtuvo un diseño consistente, seguro y con un nivel de normalización hasta tercera forma normal, además responde a los requisitos funcionales del sistema.

Por último, con el diseño realizado se garantiza la centralización de la información procesada en los Tribunales Populares Cubanos contribuyendo a la gestión eficiente de los procesos que en los mismos se desarrollan, proporcionando fiabilidad y rapidez a los mismos.

Recomendaciones

Las metas planteadas con este trabajo se han cumplido y en base a los resultados obtenidos se recomienda:

- ✓ Extender el modelo utilizado al resto de los proyectos de la universidad.
- ✓ Utilizar la propuesta de diseño de la base de datos presentada en este trabajo en el proyecto Sistema para los Tribunales Cubanos.
- ✓ Continuar con el desarrollo de la base de datos acorde al resto de las fases propuestas por el Modelo de Desarrollo de Bases de Datos.
- ✓ Proporcionar mantenimiento y soporte regular a la base de datos enfocados a su mejor funcionamiento.

Bibliografía

2010. *msdn*. [Online] 2010. [Cited: mayo 12, 2010.] <http://msdn.microsoft.com/es-es/library/aa292166%28VS.71%29.aspx>.

© Copyright 2009 IEEE – All Rights Reserved. *IEEE*. [Online] <http://www.ieee.org/portal/site>.

Alarcón, José Manuel H. 2006. *Administración de SGBD PostgreSQL*. 2006.

Albuquerque, Amado. 2007. *Sistema Integrador de Gestión Estadística (SIGE)*. Ciudad de la Habana : Universidad de las Ciencias Informáticas, 2007.

Alfaro, Félix Murillo. 1999. *Herramientas CASE*. 1999.

Alvarez, Sara. 2007. *Desarrolloweb*. [Online] agosto 14, 2007. [Cited: febrero 9, 2010.] <http://www.desarrolloweb.com/articulos/modelos-base-datos.html>.

Ambysoft Copyright © 1997-2009 Scott W. Ambler. Scott W. Ambler's Home Page. [Online] <http://www.ambysoft.com/scottAmbler.html>.

AmbySoft Copyright © 2002-2009 Scott W. Ambler. Agile Data Method. [Online] <http://www.agiledata.org/essays/vision.html>.

—. Evolutionary/Agile Database Best Practices. [Online] <http://www.agiledata.org/essays/bestPractices.html>.

—. Techniques for Successful Evolutionary/Agile Database Development. *Agile Data Home Page*. [Online] <http://www.agiledata.org/>.

Ambysoft. 2005 - 2006. Home Page. *The Agile Unified Process*. [Online] Ambysoft Inc. Copyright ©, 2005 - 2006. <http://www.ambysoft.com/unifiedprocess/agileUP.html>.

B, Orlando Solano. 1997. *Manual de informática Jurídica*. Bogotá, Colombia : Ediciones Jurídicas Gustavo Ibáñez, 1997.

Benítez, Jorge Parra. 1994. *Informática Jurídica y Derecho Informático*. Medellín, Colombia : Señal Editora, 1994.

2007. Bujarra. [Online] Diciembre 1, 2007. [Cited: marzo 22, 2010.] <http://www.bujarra.com/black/pdfs/ProcedimientoRAID1.pdf>.

Chari, Ranjan. 2009. AtlasTechnologies. [Online] 2009. [Cited: mayo 19, 2010.] <http://www.atlasindia.com/sql.htm>.

- 2009.** Construcciones cm. [Online] agosto 25, 2009. [Cited: febrero 8, 2010.] <http://ingecivil1308.blogspot.es>.
- 2006.** Csaе. [Online] junio 2, 2006. [Cited: febrero 10, 2010.] http://www.csaе.map.es/csi/tecniap/tecniap_2006/01T_PDF/sistema%20para%20el%20seguimiento.pdf.
- Date, CJ. 2003.** *Sistemas de Bases de Datos*. La Habana : Félix Varela, 2003.
- 2010.** Embarcadero. [Online] 2010. [Cited: mayo 19, 2010.] <http://www.embarcadero.com/>.
- Erich Mario Gómez Pérez, Ariel Torres Gálvez. 2008.** *Administración y optimización de un Sistema de Base de Datos Descentralizado, en PostgreSQL*. Ciudad de la Habana : Universidad de las Ciencias Informáticas, 2008.
- Fábregas, Yuniesky and Fernández, Daniel. 2007.** *Administración, configuración y optimización de un Sistema de Base de Datos Descentralizado en Oracle Database 10g release 2*. Ciudad de la Habana : Universidad de las Ciencias Informáticas, 2007.
- Florez, Fernando Jordan. 1987.** *La Informática Jurídica (teoría y práctica)*. Bogotá, Colombia : Editora Guadalupe LTDA, 1987.
- Hassan, Ahmed M. and Elssamadisy, Amr.** *Extreme Programming and Database Administration: Problems, Solutions, and Issues*.
- Judicial, Centro de Información. 2008.** CIJ . *Centro de Información Judicial*. [Online] noviembre 30, 2008. [Cited: febrero 10, 2010.] <http://www.cij.gov.ar/nota-257-Avanza-la-informatizacion-de-la-gestion-judicial-en-el-pais.html>.
- Marqués, Merche. 2002.** *Base de datos Orientado a Objetos. Diseño de Sistemas de Bases de Datos*. España : s.n., 2002.
- Martínez, M^aIsabel Sánchez. 2003.** Guía de prácticas. Diagramas de UML en ARGO-UML. Valencia : s.n., 2003.
- Matos, Daylen Benitez and Garcia, Lidysbett Horta. 2010.** *Especificación de Requisitos de Software*. La Habana : s.n., 2010.
- Moraga, M. Ángeles. 2001.** *Modelo de Datos Jerárquico*. . La Mancha : Universidad de Castilla, 2001.
- Ochoa, Darián González. 2009.** *Diseño e Implementación de un Almacén de Datos Operacionales para la Corporación CIMEX*. Ciudad de la Habana : s.n., 2009.
- Ortiz, Antonio Moreno. 2000.** Elies. [Online] 2000. [Cited: febrero 9, 2010.] <http://elies.rediris.es/elies9/4-2-3.htm>.
- Osorio Rodríguez, Alain. 2008.** Proceso general de desarrollo – administración de bases de datos: propuesta de modelo de desarrollo de bases de datos para procesos de desarrollo de software. Caracas : s.n., junio 14, 2008.

PostgreSQL. [Online] Copyright © 1996 – 2010. [Cited: abril 19, 2010.]
<http://www.postgresql.org/docs/8.0/interactive/plpgsql.html>.

PROAÑO, ING. DIEGO JAVIER BURBANO. 2006. *ANALISIS COMPARATIVO DE BASES DE DATOS DE CODIGO ABIERTO*. Quito,Ecuador : s.n., 2006.

Rafael Benitez, Ivan Roa,Diego Benítez. 2002. Dei. [Online] 2002. [Cited: febrero 9, 2010.]
<http://www.dei.uc.edu.py/tai2002/BDOO/ventoo.htm>.

Rafael Camps Paré, Luis Alberto Casillas Santillán, Dolors Costal Costa, Marc Gibert Ginestà,, 2005. *Bases de Datos*. Barcelona : s.n., 2005.

Rodríguez, Alexei Domínguez. 2009. *Diseño e implementación de la Base de Datos del Sistema de planificación del Entrenamiento deportivo de Judo*. Ciudad de la Habana : s.n., 2009.

Silva, Luis Raciél Rodríguez. 2009. *Los sistemas de gestión de servicios legales. Propuesta de modelación de un sistema para una oficina de asistencia legal en Cuba*. Ciudad de la Habana : s.n., 2009.

2010. Smdata. [Online] enero 21, 2010. [Cited: marzo 22, 2010.] <http://www.smdata.com/NivelesRAID.htm>.

Solano, Orlando. 2003. Slideshare. [Online] 2003. [Cited: mayo 14, 2010.]
<http://www.slideshare.net/guest9ca8c4/informatica-juridica-de-gestion-97-2003>.

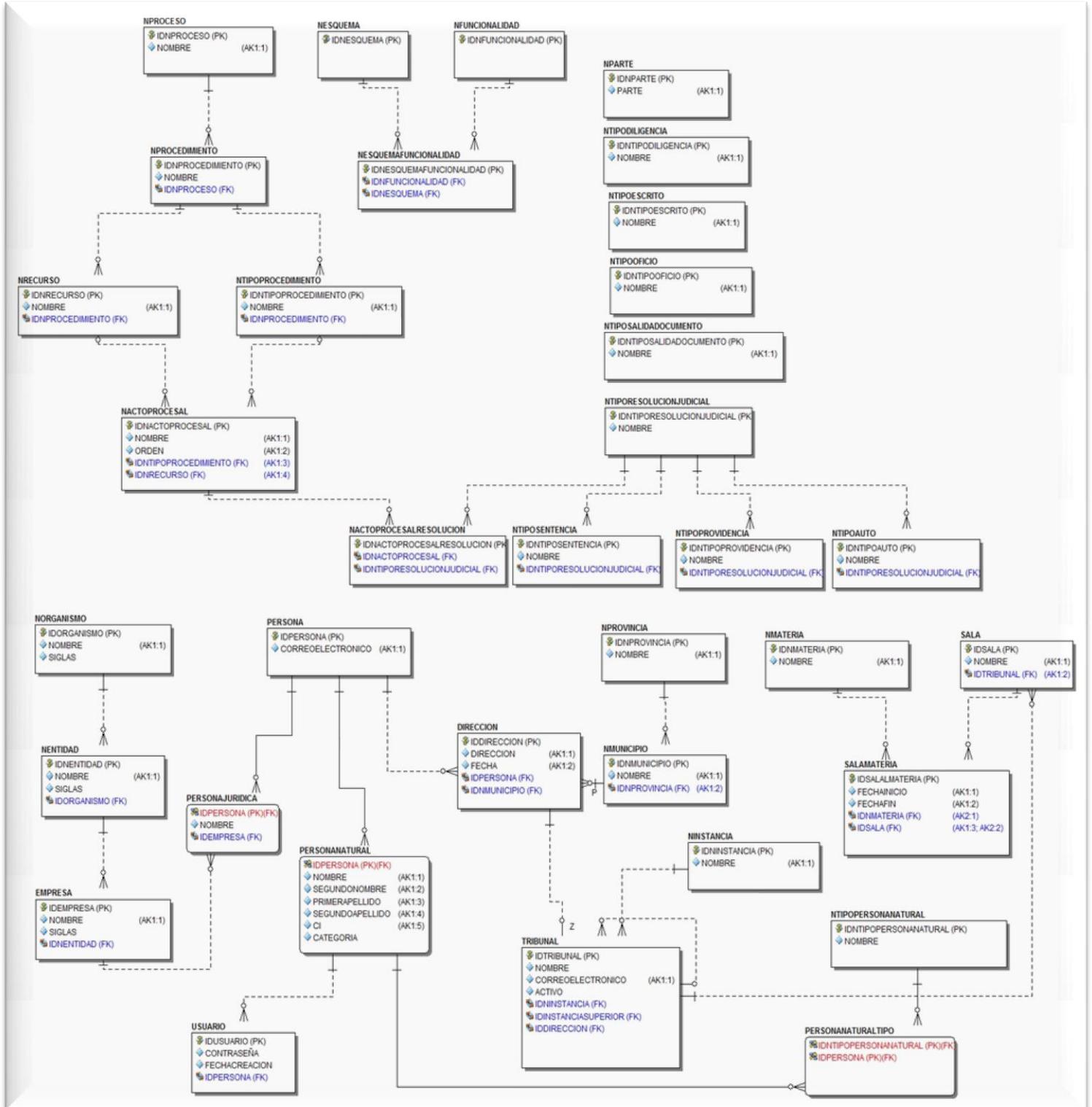
Thoughtworks. *Thoughtworks*. [Online] <http://www.thoughtworks.com/>.

Tobalina, Rubén González Blanco y Sergio Pérez. *Funcionalidad General. LESE-2 Introducción a Rational Rose*. Barcelona : s.n.

2010. Webmaster. [Online] febrero 12, 2010. [Cited: abril 9, 2010.] <http://www.guia-ubuntu.org/index.php?title=PostgreSQL>.

Zapata, Luis Giraldo y uliana. 2005. *HERRAMIENTAS DE DESARROLLO DE INGENIERIA DE SW PARA LINUX*. 2005.

Anexo2: Modelo Físico de la BD del Módulo Gobierno y Administración.



Anexo 4: Aval de calidad.



Acta de Liberación de Artefactos, Grupo de la Calidad Centro CEGEL de la Facultad 15 de la Universidad de las Ciencias Informáticas.

Lunes, 31 de mayo de 2010.

Luego de haber efectuado 2 iteraciones de revisiones a los artefactos: Documento de configuraciones del sistema operativo – Sistema gestor de bases de datos – Base de datos, Codificación de las Funcionalidades, Entorno de desarrollo de Bases de Datos y Modelo de Datos del módulo Gerencial del proyecto Tribunales Populares Cubanos del Centro CEGEL de la Facultad 15 y haberse detectado un promedio de 7 No Conformidades, se puede afirmar que se han corregido los defectos encontrados, por lo que quedan liberados todos los artefactos.

A handwritten signature in black ink, appearing to be 'Raúl', is written over a horizontal line.

Firma del Asesor y Jefe del Grupo de la Calidad Centro CEGEL

Ing. Raúl Velázquez Álvarez

