



Facultad Regional Mártires de Artemisa

*Trabajo de diploma para optar por el título de
Ingeniero en Ciencias Informáticas*

Título: *Diseño e Implementación de las Bases de datos del Módulo Dirección de Gestión Empresarial Externa para el Ministerio de Comercio Exterior y la Inversión Extranjera.*

Autor: *Leodan Diaz Novales*

Tutor: *Ing. Leidy Laura Sánchez González*

Co-Tutor: *Lic. Boris Ramón Núñez Fernández*

Artemisa, Junio 2012

Declaración de autoría

Declaro ser el autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Leodan Diaz Novales
Autor

Ing. Leidy Laura Sánchez González
Tutor

Lic. Boris Ramón Núñez Fernández
Co-Tutor



“Los que se enamoran de la práctica sin la teoría son como los pilotos sin timón ni brújula, que nunca podrán saber a dónde van.”

Leonardo da Vinci

Dedicatoria

A mi padre, que siempre me ha apoyado y que nunca ha dejado de confiar en mí. A él por ser mi maestro y mi ejemplo, por recordarme que no existe dificultad alguna cuando tu familia se encuentra a tu lado.

A mi hermana que me ha soportado siempre, y todos esos momentos vividos han formado parte de las páginas de mi vida.

A mi segunda madre Rupe que siempre ha defendido todas las decisiones que he tomado en mi vida.

*A mi amigos que siempre han estado cuando les he necesitado y nunca
me han dado la espalda.*

Agradecimientos

A mi padre por confiar en mí y darme siempre su ejemplo, por haberme ayudado a lograr la más grande meta de mi vida, sin él esto sería imposible.

A mi hermana por haberme apoyado en cada momento de mi vida y brindarme siempre su ayuda.

A Rupe por apoyarme en todo momento, brindarme su confianza y quererme como un hijo.

A mis amigos, pues aunque todos no estén presentes en este momento, me han acompañado en los buenos y malos momentos de la vida, sorteando dificultades y aprendiendo a ser mejor persona, especialmente a Yunior, Jose Carlos, Roilan, Doryan, Osniel, Yilian y Belkita los cuales considero mis hermanos.

A mis amigos de la universidad que durante estos cinco años han estado compartiendo conmigo a Henry, Livan, Reinier, Yasmany, Perdomo, Virgilio, Juan Manuel, Jennifer, Nagyara y a todos los que de una forma o de otra hemos tenido algún momento junto durante los estudios.

A mi tutora por haber dedicado parte de su preciado tiempo a orientarme y apoyarme en todo lo concerniente a la realización de este trabajo.

En fin a todos los que de una forma u otra han hecho posible que lo que un día fue un sueño en mi vida, hoy se convierta en realidad. A todos gracias.....

Resumen

En la actualidad el avance que se produce en temas de informática y debido a sus enormes beneficios hace que las instituciones busquen alternativas para que su desempeño sea más eficiente y cómodo. Esto provoca que se revolucionen las soluciones informáticas en cualquier ámbito entre ellas las relacionadas con los Sistemas de Gestión Empresarial, realizándose así en la Facultad Regional “Mártires de Artemisa” un proyecto con el MINCEX encargado del desarrollo de una variante en la cual se pretende consolidar una solución con la que se pueda estructurar, organizar y controlar la planificación en este centro, por lo cual se hace necesario contar con una base de datos que permita la recopilación, control y organización de la información deseada para que pueda ser encontrada y utilizada fácilmente.

El presente trabajo de diploma se centra en el diseño de dos bases de datos para las direcciones Sucursales y Agentes de Sociedades Mercantiles Extranjeras y Registro de Entidades Extranjeras con Intereses Cubanos ambas del MINCEX. El desarrollo del trabajo fue guiado por las actividades establecidas y por la utilización de las herramientas especificadas por la dirección del proyecto, tales como: instrumentos de diseño, gestor de base de datos y pruebas de concepto sobre las soluciones propuestas.

Se obtuvo como resultado un modelo de datos que responde a las necesidades planteadas, el cual ha sido validado teóricamente y funcionalmente teniendo en cuenta un conjunto de aspectos importantes para el área de las bases de datos relacionales como son la normalización, redundancia, integridad y seguridad.

Palabras claves: Base de datos, MINCEX, Organización.

Contenido

Introducción	7
CAPÍTULO 1: Fundamentación Teórica.....	13
Introducción	13
1 Base de Datos	13
1.1 Clasificaciones de las Bases de Datos.	14
1.2 Modelos de Base de Datos.....	14
1.2.1 Modelo Jerárquico	14
1.2.2 Modelo Red.....	15
1.2.3 Modelo Relacional.....	15
1.2.4 Modelo Orientado a Objeto	15
1.3 Sistemas de Gestión de Bases de Datos.....	16
1.3.1 Gestores de Bases de Datos.	16
1.3.1.1 Microsoft Access	16
1.3.1.2 Oracle	17
1.3.1.3 PostgreSQL	17
1.3.1.4 MySQL.....	18
1.4 Bases De Datos Aplicadas a la Gestión Empresarial.....	18
1.5 Metodologías de desarrollo.....	19
1.5.1 Metodologías Ágiles	20
1.5.1.1 Scrum	20
1.5.1.2 Extreme Programming (XP)	20
1.5.1.3 SXP	21
1.5.2 Metodologías Tradicionales.....	22
1.5.2.1 Rational Unified Process (RUP).....	22
1.6 Mapeado de Objetos Relacionales	24
1.7 Recuperación de Información.....	25
1.8 Tendencia de las Bases de Datos	25
1.8.1 Internacional.....	25
1.8.2 Nacional	26

1.9 Integración de la Capa de Persistencia de los Datos y el Servidor	26
Capítulo 2: Descripción y análisis de la solución propuesta.....	28
Introducción	28
2.1 Propuesta de solución	28
2.2 Herramientas y metodologías seleccionadas para el desarrollo del sistema	29
2.2.1 Metodología a Utilizar	29
2.2.2 Modelo Utilizado.....	30
2.2.3 Sistema Gestor de Base de Datos Utilizado.....	31
2.3 Patrones de diseño	31
2.3.1 Cuando no utilizar patrones de diseño.....	31
2.3.2 Patrones de diseño a utilizar	32
2.4 Diagrama de clases persistentes.....	32
2.5 Selección de los requisitos	34
2.5.1 Requisitos funcionales: Área Sucursales y agentes de sociedades mercantiles extranjeras.	35
2.5.2 Requisitos Funcionales: Área de Entidades extranjeras con intereses cubanos.	39
2.6 Requisitos no funcionales.....	40
2.7 Descripción de las clases persistentes.	41
Capítulo 3: Implementación y validación de la base de datos.....	59
Introducción	59
3.1 Estructura de los archivos mediante paquetes.	59
3.2 Creación y estructuración de las clases DAO.....	60
3.3 Descripción de los métodos implementados.....	60
3.4 Integridad de los datos	61
3.4.1 Integridad de entidad.....	62
3.4.2 Integridad de clave.....	62
3.4.3 Integridad de dominio.....	62
3.4.4 Integridad referencial.....	63
3.5 Normalización	63
Tercera forma normal (3FN)	64
3.6 Validación de la capa de acceso a datos.....	64

3.7 Pruebas de rendimiento	65
3.7.1 Pruebas Unitarias	65
3.7.2 Diseños de casos de pruebas	65
3.7.3 Prueba de Volumen	69
3.8 Aporte social	70
Conclusiones Generales	72
Recomendaciones	73
Referencias Bibliográficas	74
Anexos	¡Error! Marcador no definido.
Glosario de términos	76

Introducción

La capacidad de razonamiento del hombre en su afán de mejorar el nivel de vida se ha visto en la obligación de evolucionar sus métodos de trabajo, dando paso a nuevas ideas, herramientas y usando nuevas tecnologías. Los avances tecnológicos han sido un eslabón fundamental para el desarrollo de la humanidad y con este el desarrollo de la informática, ciencia definida como el tratamiento automático de la información.

Muchas son las personas que han invertido su tiempo en como buscar la manera de guardar los datos y en hacerla prosperar desde el siglo pasado hasta la actualidad, utilizando dispositivos electrónicos y sistemas computacionales, en sus inicios procesar los datos mediante el uso de la informática consistía originalmente en almacenar la información de manera centralizada, al darse cuenta que esta no era la manera más óptima se hizo necesario desarrollar técnicas más avanzadas para el tratamiento automático de estos datos, que permitiera almacenar y manejar grandes cantidades de información en poco tiempo, por lo cual surgen las bases de datos distribuidas dando respuesta así a esta gran problemática.

Hoy en día es difícil concebir un área que no use de alguna forma el apoyo de la informática y de las base de datos; existiendo así un enorme abanico que cubre desde las más simples cuestiones hogareñas hasta los más complejos cálculos científicos. Entre las utilidades más importantes está facilitar información en forma oportuna y veraz, lo cual, por ejemplo, puede facilitar la toma de decisiones a nivel gerencial como permitir el control de procesos críticos. El desarrollo de estas es una de las actividades más importantes en el campo de la informática ya que traen consigo una notable disminución de los costos y un importante incremento de la producción; tanto así que son muy pocas las empresas que no se encuentran dentro de la revolución de las computadoras.

Cuba no ha sido la excepción y ha experimentado un importante auge en este sector durante el transcurso de la última década, al punto que se quiere una

informatización masiva de toda la sociedad, también muchas de las empresas se han visto obligadas a informatizarse ya que en estas ha crecido la capacidad de generación y almacenamiento de la información por lo que dificulta ser analizada por los métodos tradicionales existentes ya que mientras mayor es la capacidad para almacenar datos, mayor es la incapacidad para extraer información realmente útil de estos, por lo que se aboga entonces por el uso de recursos en orden de lograr una mayor productividad y calidad.

La Universidad de las Ciencias Informáticas (UCI) está a la vanguardia de la producción de software y son múltiples los proyectos productivos que se realizan en ella, como el diseño y construcción de base de datos para la administración de la información en varias empresas del país. Nuestra Facultad Regional “Mártires de Artemisa” también ha jugado un papel fundamental ejecutando proyectos de gran envergadura como el que se desarrolla para el Ministerio de Comercio Exterior y la Inversión Extranjera (MINCEX) el cual será capaz de responder con calidad a las necesidades de información del país para enfrentar las metas del progreso del desarrollo, tanto social como económico de la isla.

El Ministerio del Comercio Exterior y la Inversión Extranjera (MINCEX) situado en la capital desde su creación en los años 70, cuenta con una estructura organizativa de 23 direcciones. Una de ellas es la Dirección de Gestión Empresarial Externa (DGEE) que está dividida en dos áreas. Sucursales y Agentes de Sociedades Mercantiles Extranjeras, que se encarga de evaluar y de realizar dictámenes de todos los asuntos relacionados con las sucursales que requieran autorización del MINCEX u otras instancias de Gobierno, y el Registro de Entidades Extranjeras con Intereses Cubanos donde toda información que se maneja es confidencial.

En ambas líneas de trabajo, la información que se recibe es por correo electrónico o en formato duro y el manejo de los datos se realiza de forma manual, imposibilitando el orden, la calidad y la claridad de la información. El almacenamiento de los datos se realiza en una carpeta ubicada en el ordenador de

la directora y los especialistas deben acceder a ella para realizar su trabajo, lo que provoca poca confidencialidad y seguridad en la información, estos datos pueden ser manipulados por los diferentes usuarios lo que puede afectar la disponibilidad de los mismos después de las diversas manipulaciones. El almacenamiento aglomerado de la información dificulta los reportes inmediatos, y puede ocasionar redundancia y duplicado de información, ya que no existe un proceso correcto para la salva y recuperación de estas informaciones en la dirección.

Por lo planteado anteriormente se define como **problema científico**:

¿Cómo contribuir a mejorar la calidad de los procesos de almacenamiento de datos en la Dirección de Gestión Empresarial Externa del MINCEX de forma que permita la organización, almacenamiento y recuperación de la información?

Por tanto, el **objeto de estudio**: Procesos de gestión de datos en los sistemas de Gestión Empresarial.

A partir del objeto de estudio se determina como **campo de acción**: Bases de datos relativas a la información procesada en los sistemas de Gestión Empresarial.

Para dar solución al problema planteado se trazó como **objetivo general**: Realizar el diseño y la implementación de las Bases de Datos que contribuya a mejorar la calidad de los procesos de almacenamiento de datos en el Módulo de la Dirección de Gestión Empresarial Externa.

La **idea a defender** será: Con el diseño y la implementación de las base de datos del Módulo Dirección de Gestión Empresarial Externa se logrará mejorar la calidad de los procesos de almacenamiento de datos en dicha dirección.

Como **objetivos específicos** se tienen:

- 1-Elaborar la fundamentación teórica de la investigación.
- 2- Implementar las bases de datos para el módulo de la DGEE del MINCEX.
- 3-Desarrollar las capas de acceso a datos para el módulo de la DGEE el MINCEX
- 4- Realizar la validación teórica y funcional del diseño.

Para el cumplimiento de los objetivos de esta investigación se plantean las **tareas de la investigación**

- 1- Estudio bibliográfico para profundizar en el estudio de la metodología y herramientas a utilizar.
- 2- Diseño de las bases de datos para el módulo de la DGEE del MINCEX
- 3- Implementación de la capas de acceso a datos para el módulo de la DGEE.
- 4- Realizar los métodos de prueba para las Bases de Datos.

Variables de la investigación

Variable Independiente: Capa de acceso a datos para las Direcciones de Gestión Empresarial Externa y Esquema Externo.

Variable Dependiente: Correcto almacenamiento y recuperación de la información.

Como **métodos teóricos** se utilizarán:

Analítico-Sintético: Se realizará un estudio profundo de las metodologías y herramientas a utilizar en el desarrollo de la investigación basándose en la revisión de documentos, artículos e informes.

Modelación: Es considerado de gran importancia para el diseño de la base de datos, por lo que se utilizará para la modelación de los requerimientos el cual

demostrará con claridad la forma que se va a presentar la solución.

Histórico-Lógico: Este método se utiliza para el estudio de los temas relacionados con el desarrollo de las bases de datos, su evolución y para la metodología a utilizar en el desarrollo.

Como **método Empírico** se utilizará:

Análisis-documental: Este método ayuda a analizar la situación existente, documentando las variables de la investigación identificadas y los procesos en la dirección, permitiendo comprender los aspectos de interés e identificar las contradicciones existentes entre los mismos, creando una mejor comprensión de la realidad. (Ver Anexo 1).

Al concluir la investigación se esperan los siguientes **Aportes prácticos**:

- 1- Informe detallado con toda la base teórico-práctica sobre la cual se sustenta la solución propuesta.
- 2- Capa de acceso a datos para el área de Sucursales y Agentes de Sociedades Mercantiles Extranjeras del Módulo de la DGEE del MINCEX que garantice el correcto almacenamiento y la recuperación de la información.
- 3- Capa de acceso a datos para el área de Registro de Entidades Extranjeras con Intereses Cubanos del Módulo de la DGEE del MINCEX que garantice el correcto almacenamiento y la recuperación de la información.

Este trabajo está compuesto por tres capítulos organizados de la siguiente forma:

Capítulo 1: Fundamentación teórica

Precisa la actualidad y el tratamiento del tema desde una perspectiva histórica, sustentado en una revisión bibliográfica amplia, en el que sobresale la consulta de diversos textos sobre sistemas informáticos vinculados al campo de acción, así como el uso de las tecnologías para el desarrollo de sistemas de base de datos.

Capítulo 2: Descripción y análisis de la solución propuesta

Se describen los patrones de diseño a utilizar y se definen los artefactos correspondientes al rol diseñador de base de datos, como artefactos, modelo entidad relación, modelo relacional. Además, se verán las reglas de transformación del modelo entidad relación y al modelo relacional.

Capítulo 3: Implementación y validación de la base de datos

Implementación de los artefactos del diseño de las bases de datos e integración con la aplicación en la capa de persistencia, así como las métricas y pruebas utilizadas para la validación de las mismas.

CAPÍTULO 1: Fundamentación Teórica

Introducción

A lo largo de este capítulo se exponen aspectos importantes de las Bases de Datos y los Sistemas Gestores, además de una pequeña historia de la evolución de estos tanto en nuestro país como a nivel mundial, también se argumentará todo lo relacionado con las tecnologías y herramientas definidas por el proyecto del MINCEX para el desarrollo de la aplicación, con el objetivo que se tenga una visión general de lo que se usará para la obtención del producto.

1 Base de Datos Relacionales.

Sobre las bases datos de datos existen varios conceptos brindados por grandes profesionales del tema, ejemplo de estos son:

Damián Pérez con experiencia en aplicaciones de base datos y reconocido mundialmente define “una base de datos es un almacén que nos permite guardar grandes cantidades de información de forma organizada para que luego podamos encontrar y utilizar fácilmente”. (Pérez, 2007)

De acuerdo con C. J. Date, escritor del libro “Introducción a las bases de datos”: “es una colección de datos integrados, con redundancia controlada y con una estructura que refleje las interrelaciones y restricciones existentes en el mundo real”. (Date, 2006).

Según Raúl Uranga otro gran especialista define “una base de datos como una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información de una empresa o negocio en particular”. (Uranga, 2009)

No entrando en contradicción con las definiciones anteriores y considerando que recoge de forma concisa las exigencias que se pretenden alcanzar, el autor se

adscribe a la definición de base datos expresada por Damián Pérez:

Una base de datos es un almacén que nos permite guardar grandes cantidades de información de forma organizada para que luego podamos encontrar y utilizar fácilmente”. (Pérez, 2007)

1.1 Clasificaciones de las Bases de Datos Relacionales.

Bases de datos estáticas: Son BD de sólo lectura, utilizadas primordialmente para almacenar datos históricos que posteriormente se pueden utilizar para estudiar el comportamiento de un conjunto de datos a través del tiempo, realizar proyecciones y tomar decisiones. Son aquellas BD en las que no se puede modificar la información que está guardada, es decir, solo se puede consultar. (Silvente, 2008)

Bases de datos dinámicas: Son BD donde la información almacenada se modifica con el tiempo, permitiendo operaciones como actualización y adición de datos, además de las operaciones fundamentales de consulta. Son aquellas BD en las que la información que está almacenada está en constante cambio, ya que sobre ella se pueden realizar operaciones de inserción, actualización y eliminación así como las operaciones de consultas. (Silvente, 2008)

1.2 Modelos de Base de Datos

Las Bases de Datos a través de su evolución han atravesado diferentes estructuras, desde los modelos de red hasta los modelos actuales orientados a objetos, cada uno de estos modelos tiene características peculiares que los identifican, estos se pueden agrupar en dos grandes grupos:

Los modelos tradicionales (modelo jerárquico, modelo de red, modelo relacional) y los modelos avanzados (de los modelos orientados a objetos en adelante).

1.2.1 Modelo Jerárquico

Una base de datos jerárquica consiste en una colección de registros que se conectan entre sí por medio de ligas. Los registros y las ligas son similares a los del

modelo de red, pero en el modelo jerárquico se organiza en forma de árbol con raíz (donde la raíz es nodo ficticio); de tal manera que una base de datos jerárquica es una colección de árboles de este tipo, formando un bosque. A cada árbol con raíz se le denomina árbol de base de datos donde son especialmente útiles en el caso de aplicaciones que manejan un gran volumen de información y datos muy compartidos permitiendo crear estructuras estables y de gran rendimiento. (Leyva, 2008)

1.2.2 Modelo Red

Este fue creado para representar relaciones de datos complejas más eficientes de lo que el modelo anterior permitía, para mejorar el desempeño de las bases de datos y para imponer un estándar. Este modelo es similar al jerárquico en muchos aspectos, sin embargo la diferencia radica, en que el modelo red, permite que un registro tenga más de un padre, por consiguiente, las relaciones pueden manejarse fácilmente por este modelo. (Fábregas, 2007)

1.2.3 Modelo Relacional

Una base de datos relacional es una base de datos que cumple con el modelo relacional, el cual es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. Permiten establecer relaciones entre los datos (que están guardados en tablas), y trabajar con ellos conjuntamente. A diferencia de otros modelos como el jerárquico y el de red en este modelo no tiene importancia el lugar y la forma en que se almacenan los datos. Permitiendo una considerable ventaja al hacerlo más fácil de entender y de utilizar para usuarios ocasionales de la BD. (W. Hansen, 2006)

1.2.4 Modelo Orientado a Objeto

Trata los problemas desde un punto de vista realista, y modelando cada uno de ellos como si se tratase de un conjunto de elementos u objetos que interrelacionan entre sí para solucionar el problema. Cada uno de estos objetos es un elemento.

De esta manera, una modificación del estado de un objeto por parte de un usuario, desencadena una serie de acciones cuyo objetivo final es solucionar un problema al usuario. (Benítez, 2002)

1.3 Sistemas de Gestión de Bases de Datos.

Los gestores de Base de Datos, son programas para la creación y manipulación de las bases de datos los mismos deben permitir, definir una base de datos(especificar tipos, estructuras y restricciones de datos), construir la base de datos(guardar los datos en algún medio controlado por el mismo SGBD) y manipular la base de datos(realizar consultas, actualizarla, generar informes) hoy en día estos programa se desarrollan constantemente, y especializándose en los diferentes modelos de Bases de Datos.(Álvarez, 2007)

1.3.1 Gestores de Bases de Datos.

En la actualidad existen un gran número de SGBD, con el auge de la programación orientada a objetos, los gestores orientados hacia este paradigma han tomado un camino revolucionario y han aparecido muchos nuevos, al igual los SGBD que funcionan sobre el modelo relacional, tampoco se han quedado atrás y más aún cuando este es el modelo más usado actualmente, por esto es difícil decidir cuál es el mejor gestor de bases de datos, dentro de estos existen 2 grandes grupos los de software propietario y los de software libre, a continuación ejemplos de algunos de ellos :

1.3.1.1 Microsoft Access

Es un programa Sistema de gestión de base de datos relacional creado y modificado por Microsoft para uso personal de pequeñas organizaciones. Es un componente de la suite Microsoft Office aunque no se incluye en el paquete "básico". Una posibilidad adicional es la de crear ficheros con bases de datos que pueden ser consultados por otros programas. (Acevedo, 2009)

Desventajas

Este SGBD no es recomendable para bases de datos de gran calibre (en cuanto a volumen de datos o de usuarios). - Entre sus mayores inconvenientes figuran que no es multiplataforma, pues está disponible para sistemas operativos de Microsoft nada más, Su uso es inadecuado para grandes proyectos de software que requieren tiempos de respuesta críticos o muchos accesos simultáneos a la base de datos.

1.3.1.2 Oracle

Es un sistema de gestión de base de datos relacional fabricado por Oracle Corporation. Se considera a Oracle como uno de los sistemas de bases de datos más completos, destacando su: Soporte de transacciones, estabilidad, escalabilidad, soporte multiplataforma. (Leyva, 2008)

Desventajas

El mayor inconveniente de Oracle es quizás su precio. Incluso las licencias de Personal Oracle son excesivamente caras.

Otro problema es la necesidad de ajustes. Un error frecuente consiste en pensar que basta instalar el Oracle en un servidor y enchufar directamente las aplicaciones clientes. Un Oracle mal configurado puede ser lento.

También es elevado el coste de la formación, y últimamente es que han comenzado a aparecer buenos libros sobre asuntos técnicos distintos de la simple instalación y administración.

1.3.1.3 PostgreSQL

Es un sistema de gestión de bases de datos objeto-relacional (ORDBMS) basado en el proyecto POSTGRES, de la universidad de Berkeley. Fue el pionero en muchos de los conceptos existentes en el sistema objeto-relacional actual, incluido, más tarde en otros sistemas de gestión comerciales. PostgreSQL es un sistema objeto-relacional, ya que incluye características de la orientación a objetos, como

puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. A pesar de esto, PostgreSQL no es un sistema de gestión de bases de datos puramente orientado a objetos. (Alma, 2006)

Desventajas

Consume bastantes recursos y carga el sistema.

Límite del tamaño de cada fila de las tablas a 8k (se puede ampliar a 32k recompilando, pero con un coste añadido en el rendimiento).

Menos funciones en PHP.

1.3.1.4 MySQL

Es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. MySQL AB desde enero de 2008 una subsidiaria de Sun Microsystems desarrolla MySQL como software libre en un esquema de licenciamiento dual. (Alma, 2006)

Desventajas

Un gran porcentaje de las utilidades de MySQL no están documentadas.

No es intuitivo, como otros programas.

1.4 Bases De Datos Aplicadas a la Gestión Empresarial

Hoy en día el desarrollo de las tecnologías de las telecomunicaciones y de la informática nos han traído a lo que muchos denominan “la sociedad de la información”, por tener acceso información ilimitada. Las organizaciones operan con información, de hecho es la pieza clave que permite precisamente la organización de los individuos o partes para su buen funcionamiento, por lo tanto dichos avances tecnológicos tienen gran impacto en ellas. Para que sea un impacto positivo, los gerentes deben de tener la capacidad de utilizarlas y dirigirlas.

Capítulo I: Fundamentación Teórica

La toma de decisiones en una organización es una actividad de la dirección que debe hacerse rápida, oportuna y fundamentalmente, que permita tomar decisiones efectivas, eficientes y de bajo costo para la empresa; para esto se necesita conocer algunos datos del comportamiento de esta, por lo cual se hace indispensable una base de datos. Dicha base de datos son un conjunto de elementos orientados al tratamiento y administración de dichos datos e información, organizándolos y dejándolos listos para su posterior uso, que son recopilados mediante sistemas informativos. (Cardoza, 2011)

1.5 Metodologías de desarrollo

El avance de la industria del software ha determinado el uso de metodologías capaces de estructurar, planear y controlar el proceso de desarrollo de las aplicaciones informáticas, con el fin de evitar clientes y desarrolladores insatisfechos con los resultados obtenidos. Su objetivo es dar solución a los problemas existentes en la producción de software, englobando procedimientos, técnicas, documentación y herramientas que se utilizan en la creación de un producto de software.

Existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Por una parte están las propuestas más tradicionales que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, las herramientas y notaciones que se usarán.

Por otra parte están las que se centran en otras dimensiones, como lo son el factor humano o el producto software. Esta última es la filosofía de las metodologías ágiles, las cuales dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas. A continuación se pondrán ejemplos y se explicarán ambos tipos de metodologías:

1.5.1 Metodologías Ágiles

El término “ágil” enfocado al desarrollo del software, nace con el objetivo fue esbozar los valores y principios que deberán permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto. A continuación se describen algunas de estas metodologías.

1.5.1.1 Scrum

Es un modelo de referencia que define un conjunto de prácticas y roles, y que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante un proyecto. Sus roles principales son: el director de proyecto, encargado de mantener los procesos, los clientes externos o internos y los desarrolladores. (Vegazo , 2012)

1.5.1.2 Extreme Programming (XP)

Esta metodología de desarrollo de software es una de las más exitosas utilizada en la actualidad para proyectos de corto plazo, atrasados y con escaso personal. Consiste en una programación rápida o extrema con mucha reutilización de código que ha sido desarrollado previamente. Una particularidad de XP es satisfacer al completo las necesidades del cliente, por eso lo integra como una parte más del equipo de desarrollo.

Fue diseñada inicialmente para el desarrollo de aplicaciones donde el cliente no sabe lo que en verdad quiere, lo que provoca un cambio constante en los requisitos que debe cumplir la aplicación.

Las características principales de XP son las siguientes:

Pruebas Unitarias: Se basa en las pruebas realizadas a los principales procesos, de tal manera que se hagan pruebas de las fallas que pudieran ocurrir. Es como adelantarse a obtener los posibles errores.

Refabricación: Radica en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexibles al cambio.

Programación en pares: Una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento. (Como el chófer y el copiloto: mientras uno conduce, el otro consulta el mapa).

1.5.1.3 SXP

SXP es una metodología de desarrollo de software compuesta por las metodologías SCRUM y XP que ofrece una estrategia tecnológica, a partir de la introducción de procedimientos ágiles que permitan actualizar los procesos de software para el mejoramiento de la actividad productiva fomentando el desarrollo de la creatividad, aumentando el nivel de preocupación y responsabilidad de los miembros del equipo, ayudando al líder del proyecto a tener un mejor control del mismo. SCRUM es una forma de gestionar un equipo para que trabaje eficientemente y tenga siempre medidos los progresos. XP más bien es una metodología encaminada para el desarrollo; consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

Consta de 4 fases principales:

Planificación-Definición: Donde se establece la visión, se fijan las expectativas y se realiza el aseguramiento del financiamiento del proyecto.

Desarrollo: Es donde se realiza la implementación del sistema hasta que esté listo para ser entregado.

Entrega: Es la puesta en marcha.

Mantenimiento: Donde se realiza el soporte para el cliente.

De cada una de estas fases se realizan numerosas actividades tales como el levantamiento de requisitos, la priorización de la Lista de Reserva del Producto, definición de las Historias de Usuario, diseño, implementación, pruebas, entre otras; de donde se generan artefactos para documentar todo el proceso. Las entregas son frecuentes, lo que permite mejorar el diseño cada vez que se le añada una nueva funcionalidad.

SXP está especialmente indicada para proyectos de pequeños equipos de trabajo, rápido cambio de requisitos o requisitos imprecisos, muy cambiantes, donde existe un alto riesgo técnico y se orienta a una entrega rápida de resultados y una alta flexibilidad. Ayuda a que trabajen todos juntos, en la misma dirección, con un objetivo claro, permitiendo además seguir de forma clara el avance de las tareas a realizar, de forma que los jefes pueden ver día a día cómo progresa el trabajo.(Peñalver., 2008.)

1.5.2 Metodologías Tradicionales

Las metodologías tradicionales ofrecen un modelo de referencia sobre buenas prácticas, maduras, consolidadas, y probadas para el desarrollo y mantenimiento de productos y servicios, cubriendo todo el ciclo de vida, desde la concepción a la entrega y mantenimiento. (Luis Infante, 2009)

1.5.2.1 Rational Unified Process (RUP)

Es la metodología más utilizada en la actualidad para proyectos a medio y largo plazo. El nivel de detalle con el que la metodología RUP propone trabajar ayuda a lograr el producto esperado. Divide en cuatro fases el desarrollo del software, en las cuales se realizan actividades que van dando lugar a los artefactos necesarios

Capítulo I: Fundamentación Teórica

para el avance del proyecto. Tales actividades son conocidas como flujos de trabajo, los cuales tienen mayor o menor peso según la fase en que se encuentre el proyecto. Existen nueve flujos, de ellos los seis primeros son conocidos como flujos de ingeniería y los tres restantes como flujos de apoyo. A continuación se relacionan las fases y los flujos de trabajo.

Fases

- **Inicio:** En esta etapa se describe el negocio y se determinan los límites del proyecto.
- **Elaboración:** Se define la línea base de la arquitectura.
- **Construcción:** Se obtiene un producto documentado, listo para su uso, se obtienen uno o varias versiones que han pasado por pruebas.
- **Transición:** Se obtiene un producto final listo para su uso. Puede implicar reparación de errores.

Características de RUP

Dirigido por Casos de Uso: Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. Los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen como resultado de los diferentes flujos de trabajo, representan la realización de estos.

Centrado en la arquitectura: La arquitectura en un sistema software se describe mediante diferentes vistas del sistema en construcción. Este concepto incluye los aspectos estáticos y dinámicos más significativos del sistema y se refleja en los casos de uso, pues cada producto tiene tanto una función como una forma, ninguna es suficiente por sí sola.

Iterativo e incremental: Resulta práctico dividir el trabajo en partes más pequeñas o miniproyectos, los cuales no son más que iteraciones que resultan en un incremento. (Pressman, 2005)

1.6 Mapeado de Objetos Relacionales

Un Mapeador de Objetos Relacionales (ORM) es una técnica de programación que permite convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional, es decir, las tablas de la base de datos pasan a ser clases y los registros objetos que se pueden manejar con facilidad

Utilizar un ORM tiene una serie de ventajas que facilitan enormemente tareas comunes y de mantenimiento:

Reutilización: La principal ventaja que aporta un ORM es la reutilización permitiendo llamar a los métodos de un objeto de datos desde distintas partes de la aplicación e incluso desde diferentes aplicaciones.

Encapsulación: La capa ORM encapsula la lógica de los datos pudiendo hacer cambios que afectan a toda la aplicación únicamente modificando una función.

Portabilidad: Utiliza una capa de abstracción que permite cambiar en mitad de un proyecto de una base de datos PostgreSQL a una Oracle sin ningún tipo de complicación. Esto es debido a que no utiliza una sintaxis MySQL, Oracle o PostgreSQL para acceder al modelo, sino una sintaxis propia del ORM utilizado que es capaz de traducir a diferentes tipos de bases de datos.

Seguridad: Los ORM suelen implementar mecanismos de seguridad que protegen las aplicaciones de los ataques más comunes como inyecciones SQL.

Mantenimiento del código: Gracias a la correcta ordenación de la capa de datos, modificar y mantener el código es una tarea sencilla. (Unaula, 2010)

1.7 Recuperación de Información

El proceso de recuperación se lleva a cabo mediante consultas a las bases de datos donde se almacena la información estructurada, mediante un lenguaje de interrogación adecuado. Es necesario tener en cuenta los elementos claves que permiten hacer la búsqueda, determinando un mayor grado de pertinencia y precisión, como son: los índices y las palabras claves. Uno de los problemas que surgen en la búsqueda de información es si lo que recuperamos es "mucho o poco" es decir, dependiendo del tipo de búsqueda se pueden recuperar multitud de documentos o simplemente un número muy reducido. A este fenómeno se denomina Silencio o Ruido documental.

- **Silencio documental:** Son aquellos documentos almacenados en la base de datos pero que no han sido recuperados, debido a que la estrategia de búsqueda ha sido demasiado específica o que las palabras clave utilizadas no son las adecuadas para definir la búsqueda.
- **Ruido documental:** Son aquellos documentos recuperados por el sistema pero que no son relevantes. Esto suele ocurrir cuando la estrategia de búsqueda se ha definido demasiado genérica. (Pinto, 2011)

1.8 Tendencia de las Bases de Datos

1.8.1 Internacional

La evolución de las bases de datos en la actualidad a nivel mundial está en incremento, muchos expertos en la materia coinciden que estas van en aumento día a día en todas las áreas profesionales, como la investigación, tecnología, arte, educación, entre otros. Estos datos son importante para las empresas, debido a que su mayor utilización es para la toma de decisiones, en sí los simples datos no dicen nada, sino la información ya estructurada por lo que se ha logrado actualmente que en las bases de datos se pueden realizar consultas en tiempo

real, y visualizarlo en reportes que en realidad es lo que los directivos de una empresa necesitan ver.

1.8.2 Nacional

El desarrollo y uso de las bases de datos en Cuba ha ido avanzando considerablemente desde la última década hasta la actualidad, a partir de la utilización masiva de las tecnologías de la información y las comunicaciones en todas las esferas de la sociedad cubana. Sin embargo como consecuencia de los efectos del bloqueo y por las posibilidades que brinda al país el uso del software libre se ha visto una progresiva migración hacia los SGBD de código abierto como son el Postgre y MySQL, los cuales presentan licencia de código abierto y están a disposición del público de forma gratuita y un ejemplo de esto es la Facultad "Mártires de Artemisa" que dio un paso de avance con respecto a este tema para así apoyar a la economía cubana.

1.9 Integración de la Capa de Persistencia de los Datos y el Servidor

Para integrar la capa de persistencia y acceso de los datos con el cliente – servidor se utilizará la herramienta hibernate, el cual es una herramienta de mapeo objeto/relacional para entornos Java. El término de mapeo objeto/relacional (ORM) se refiere a la técnica de mapear una representación de datos desde un modelo de objeto a un modelo de datos relacionales con un esquema basado en SQL.

Hibernate no solamente se ocupa del mapeo desde las clases Java a las tablas de las bases de datos (y desde los tipos de datos de Java a los tipos de datos de SQL), sino que también facilita la consulta y recuperación de datos. Esto puede reducir de manera importante el tiempo de desarrollo que se tomaría con el manejo de datos de forma manual en SQL.

Es posible que Hibernate no sea la mejor solución para aquellas aplicaciones centralizadas en datos que solamente utilizan los procedimientos almacenados para implementar la lógica empresarial en la base de datos, este es mucho más útil con modelos de dominio orientados a objetos. (King, 2010)

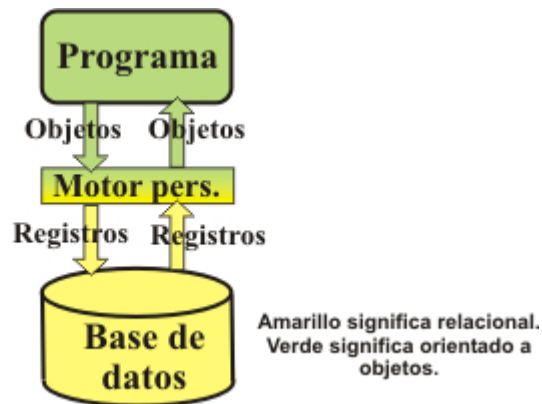


Fig. 1. Motor de Persistencia

Conclusiones Parciales

Se abordaron en el capítulo elementos importantes relacionados con los aspectos fundamentales de los sistemas que han presentado soluciones similares en función de demostrar la necesidad de una nueva alternativa, se tocaron aspectos vinculados a BD así como Gestores de Base de Datos y las herramientas utilizadas para alcanzar el resultado esperado, tratando de abarcar los elementos teóricos que sustentan la solución del problema permitiendo de esta forma sentar las bases teóricas para la correcta realización del trabajo que se pretende llevar a cabo.

Capítulo 2: Descripción y análisis de la solución propuesta.

Introducción

En este capítulo se describen los procesos principales que tiene lugar para la construcción de la BD, entre los que se encuentran: la selección y argumentación de los requisitos, el desarrollo del modelo lógico y físico presentando el diagrama de clases persistentes, la descripción de las clases y tablas principales obtenidas en ambos modelos, se describirán las metodologías y herramientas específicas que se van a utilizar en la tesis y se expondrán los conceptos de patrones de diseño existentes enfatizando en el Data Acces Object (DAO).

2.1 Propuesta de solución

En la Dirección de Gestión Empresarial Externa (DGEE) del Ministerio del Comercio Exterior y la Inversión Extranjera (MINCEX), la información no se gestiona correctamente, debido a esto se está desarrollando un sistema de gestión de información capaz de mejorar la integridad y disponibilidad de la misma. Actualmente en la DGEE que está dividida en dos áreas, Sucursales y Agentes de Sociedades Mercantiles Extranjeras, que se encarga de evaluar y de realizar dictámenes de todos los asuntos relacionados con las sucursales que requieran autorización del MINCEX u otras instancias de Gobierno, y el Registro de Entidades Extranjeras con Intereses Cubanos donde toda información que se maneja es confidencial.

El Módulo de la (DGEE) del MINCEX tiene como objetivo garantizar la correcta gestión de la información de todos los datos que se manejan en el área de Sucursales y Agentes de Sociedades Mercantiles Extranjeras, permitiendo un mejor y rápido trabajo en el área y un mayor acceso a los datos que se manejan dentro de la organización. El sistema a desarrollar debe permitir al usuario realizar un

Capítulo II: Descripción y análisis de la solución propuesta

adecuado seguimiento y control de los procesos desarrollados en la dirección. Los usuarios podrán acceder y trabajar con la información que tengan permiso de autorización, el sistema contará con una página principal de libre acceso para cualquier usuario. Toda la información se encontrará centralizada brindando mayor confiabilidad y seguridad de toda la documentación permitiendo que la misma esté a tiempo y disponible para consultar algún dato o la solicitud de cualquier reporte. El sistema contará con una base de datos que se encargará de guardar toda la información y permitirá que dada cualquier solicitud la misma cargue de forma fácil y rápida.

Para la otra línea de trabajo conocida como Esquema Externo se realizará una aplicación propia para esa área que no esté vinculada al sistema informativo por el nivel de confidencialidad de la información con la que se trabaja. Ambas aplicaciones deben garantizar la correcta gestión de la información y permitir mejorar la confiabilidad, portabilidad y confidencialidad de la información generada en la DGEE.

Ambos módulos contarán con funcionalidades propias que darán respuesta a la problemática de la Dirección, específicamente la generación de modelos y reportes con el objetivo de agilizar el trabajo que se realiza en la Dirección y dar solución a los diferentes procesos que se desarrollan en la misma.

2.2 Herramientas y metodologías seleccionadas para el desarrollo del sistema

2.2.1 Metodología a Utilizar

Uno de los factores que influyen en la ejecución exitosa de un proyecto es la metodología con la que se desarrolla. Por ello la metodología que se elija debe recoger el aspecto filosófico de la aproximación deseada, es decir que los objetivos generales del desarrollo deben estar implementados en la metodología de desarrollo. La metodología de desarrollo que será empleada para la realización del proyecto será SXP, la cual está constituida por las metodologías SCRUM y XP que

Capítulo II: Descripción y análisis de la solución propuesta

ofrece una estrategia tecnológica, a partir de la introducción de procedimientos ágiles que permitan actualizar los procesos de software para el mejoramiento de la actividad productiva fomentando el desarrollo de la creatividad, aumentando el nivel de preocupación y responsabilidad de los miembros del equipo, ayudando al líder del proyecto a tener un mejor control del mismo.

Consta de 4 fases principales: Planificación-Definición donde se establece la visión, se fijan las expectativas y se realiza el aseguramiento del financiamiento del proyecto.

- Desarrollo, es donde se realiza la Implementación del sistema hasta que esté listo para ser entregado.
- Entrega, puesta en marcha.
- Mantenimiento, donde se realiza el soporte para el cliente.

De cada una de estas fases se realizan numerosas actividades tales como el levantamiento de requisitos, la priorización de la Lista de Reserva del Producto, definición de las Historias de Usuario, diseño, implementación, pruebas, entre otras; de donde se generan artefactos para documentar todo el proceso. Las entregas son frecuentes, y existe una refactorización continua, lo que nos permite mejorar el diseño cada vez que se le añada una nueva funcionalidad. (Peñalver, 2008)

2.2.2 Modelo Utilizado

Para el desarrollo de este trabajo se usó el modelo de Base de Dato Relacional porque permite una absoluta libertad en las relaciones entre las tablas, es el más usado en la actualidad para representar problemas reales y para realizar la administración de datos de forma dinámica. Además es de fácil utilización por usuarios esporádicos, lo cual facilita su uso y mantenimiento.

Capítulo II: Descripción y análisis de la solución propuesta

2.2.3 Sistema Gestor de Base de Datos Utilizado

Aunque a la hora de la elección de una herramienta de trabajo los criterios a favor de una o de otra son variables, diversos y confusos ya que varias de ellas tienen comportamientos similares y no debe decidirse a la suerte o al azar: tiene que haber un análisis previo de las opciones existentes, y partiendo de las necesidades del problema, por tal motivo se escogió el PostgreSQL ya que está considerado como uno de los gestores de bases de datos de código abierto más avanzados del mundo. Es básicamente una herramienta Cliente/Servidor para la gestión de bases de datos. Comparándolo con otros SGBD's libres como MySQL, Ingres o Firebird, dada la amplitud de posibilidades que PostgreSQL ofrece, es sin duda el SGBD que más se asemeja en versatilidad a los SGBD propietarios como Oracle, Sybase, DB2 o SQLServer.

2.3 Patrones de diseño

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser reutilizable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias. (Guzmán, 2011)

2.3.1 Cuando no utilizar patrones de diseño

Del mismo modo que no es aconsejable optimizar prematuramente, no se deben utilizar patrones de diseño antes de tiempo. Seguramente sea mejor implementar algo primero y asegurarse de que funciona, para luego utilizar el patrón de diseño para mejorar las flaquezas; esto es cierto, sobre todo, cuando aún no ha identificado todos los detalles del proyecto. Estos patrones de diseño pueden

Capítulo II: Descripción y análisis de la solución propuesta

incrementar o disminuir la capacidad de comprensión de un diseño o de una implementación, disminuirla al añadir accesos indirectos o aumentar la cantidad de código, disminuirla al regular la modularidad, separar mejor los conceptos y simplificar la descripción. (Tedeschi, 2001)

2.3.2 Patrones de diseño a utilizar

La mayoría de las aplicaciones, tienen que persistir datos en algún momento, ya sea guardándolos en una base de datos relacional, o una base de datos orientada a objetos, etc. Para hacer esto, la aplicación interactúa con la base de datos. El "como interactúa" no debe ser asunto de la capa de lógica de negocio de la aplicación, ya que para eso está la capa de persistencia, que es la encargada de interactuar con la base de datos. Sabiendo esto, podemos decir que DAO es el patrón de diseño que se va a utilizar para crear esta capa de persistencia. Este consiste básicamente en una clase que es la que interactúa con la base de datos. Los métodos de esta clase dependen de la aplicación y de lo que queramos hacer. Pero generalmente se implementan los métodos CRUD para realizar las "4 operaciones básicas" de una base de datos.

2.4 Diagrama de clases persistentes

El diagrama de clases del diseño da lugar a la confección del diagrama de clases persistentes. La persistencia de una clase está definida por la propiedad de los objetos de trascender su estado en el tiempo y el espacio: una clase persistente existirá durante la ejecución de un programa (incluyendo sus versiones), y deberá sobrevivir incluso a la eliminación o colapso del mismo.

Capítulo II: Descripción y análisis de la solución propuesta

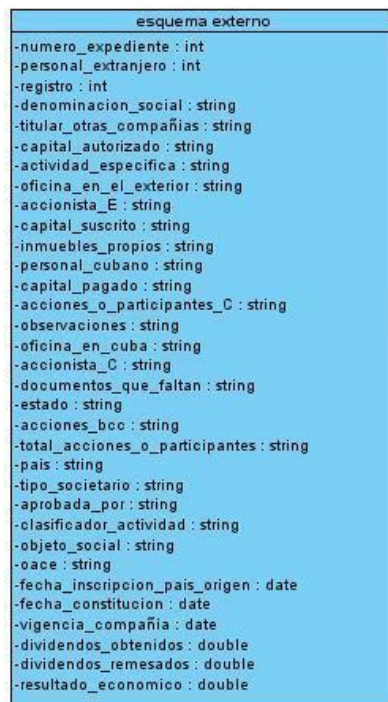


Fig. 2. Diagrama de clases persistentes de Esquema Externo

Capítulo II: Descripción y análisis de la solución propuesta

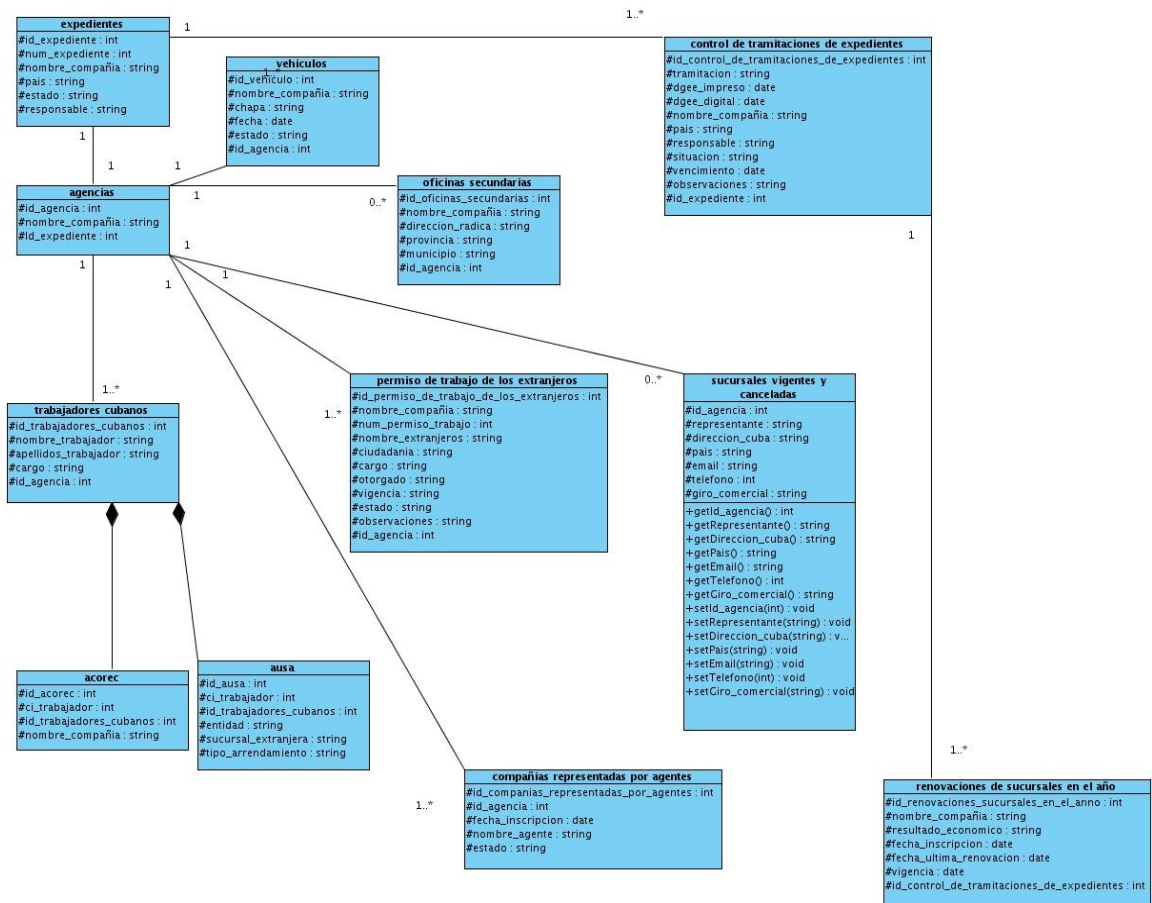


Fig. 3. Diagrama de clases persistentes de Sucursales

2.5 Selección de los requisitos

Para que un sistema de software funcione correctamente se debe lograr una comunicación efectiva entre los usuarios y el equipo de proyecto con el objetivo de llegar a acorec un entendimiento de lo que hay que hacer, lo que constituye la clave del éxito en la producción de un software.

Aquí radica la importancia de identificar los requerimientos como parte del proceso de desarrollo del software. Los requisitos se clasifican en funcionales y no funcionales.

Capítulo II: Descripción y análisis de la solución propuesta

2.5.1 Requisitos funcionales: Área Sucursales y agentes de sociedades mercantiles extranjeras.

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir, los cuales constituyen un paso esencial para el diseño de las entidades de la BD propuesta.

RF1: Obtener reporte de Sucursales y agentes de sociedades mercantiles extranjeras.

RF2: Gestionar listado de expedientes.

- Insertar listado de expedientes.
- Actualizar listado de expedientes.
- Eliminar listado de expedientes.
- Buscar listado de expedientes.
- **RF3:** Mostrar Listado de expedientes.

RF4: Exportar Listado de expedientes.

RF5: Cargar Listado de expedientes.

RF6: Obtener reporte del Listado de expedientes.

RF7: Gestionar Tabla de control de tramitaciones de expedientes.

- Insertar Tabla de control de tramitaciones de expedientes.
- Actualizar Tabla de control de tramitaciones de expedientes.
- Eliminar Tabla de control de tramitaciones de expedientes.
- Buscar Tabla de control de tramitaciones de expedientes.

Capítulo II: Descripción y análisis de la solución propuesta

RF8: Mostrar Tabla de control de tramitaciones de expedientes.

RF9: Exportar Tabla de control de tramitaciones de expedientes.

RF10: Cargar Tabla de control de tramitaciones de expedientes.

RF11: Obtener reporte de la Tabla de control de tramitaciones de expedientes.

RF12: Gestionar las Renovaciones de las sucursales en el año.

- Insertar Renovaciones de las sucursales en el año.
- Actualizar Renovaciones de las sucursales en el año.
- Eliminar Renovaciones de las sucursales en el año.
- Buscar Renovaciones de las sucursales en el año.

RF13: Mostrar las Renovaciones de las sucursales en el año.

RF14: Exportar las Renovaciones de las sucursales en el año.

RF15: Cargar las Renovaciones de las sucursales en el año.

RF16: Obtener reporte de Renovaciones de las sucursales en el año.

RF17: Gestionar Sucursales.

- Insertar Sucursales.
- Actualizar Sucursales.
- Eliminar Sucursales.
- Buscar Sucursales.

RF18: Mostrar Sucursales.

RF19: Exportar Sucursales.

RF20: Cargar Sucursales.

Capítulo II: Descripción y análisis de la solución propuesta

RF21: Obtener reportes de Sucursales.

RF22: Gestionar Compañías representadas por agentes.

- Insertar Compañías representadas por agentes.
- Actualizar Compañías representadas por agentes.
- Eliminar Compañías representadas por agentes.
- Buscar Compañías representadas por agentes.

RF23: Mostrar Compañías representadas por agentes.

RF24: Exportar Compañías representadas por agentes.

RF25: Cargar Compañías representadas por agentes.

RF26: Obtener reportes de Compañías representadas por agentes.

RF27: Gestionar Oficinas Secundarias.

- Insertar Oficinas Secundarias.
- Actualizar Oficinas Secundarias.
- Eliminar Oficinas Secundarias.
- Buscar Oficinas Secundarias.

RF28: Mostrar Oficinas Secundarias.

RF29: Exportar Oficinas Secundarias.

RF30: Cargar Oficinas Secundarias.

RF31: Obtener reportes de Oficinas Secundarias.

RF32: Gestionar Permiso de trabajo de los extranjeros.

- Insertar Permiso de trabajo de los extranjeros.

Capítulo II: Descripción y análisis de la solución propuesta

- Actualizar Permiso de trabajo de los extranjeros.
- Eliminar Permiso de trabajo de los extranjeros.
- Buscar Permiso de trabajo de los extranjeros.

RF33: Mostrar Permiso de trabajo de los extranjeros.

RF34: Exportar Permiso de trabajo de los extranjeros.

RF35: Cargar Permiso de trabajo de los extranjeros.

RF36: Obtener reportes de Permiso de trabajo de los extranjeros.

RF37: Gestionar Trabajadores cubanos por ACOREC.

- Insertar Trabajadores cubanos por ACOREC.
- Actualizar Trabajadores cubanos por ACOREC.
- Eliminar Trabajadores cubanos por ACOREC.
- Buscar Trabajadores cubanos por ACOREC.

RF38: Mostrar Trabajadores cubanos por ACOREC.

RF39: Exportar Trabajadores cubanos por ACOREC.

RF40: Cargar Trabajadores cubanos por ACOREC.

RF41: Obtener reportes de Trabajadores cubanos por ACOREC.

RF42: Gestionar Trabajadores cubanos por AUSA.

- Insertar Trabajadores cubanos por AUSA.

Capítulo II: Descripción y análisis de la solución propuesta

- Actualizar Trabajadores cubanos por AUSA.
- Eliminar Trabajadores cubanos por AUSA.
- Buscar Trabajadores cubanos por AUSA.

RF43: Mostrar Trabajadores cubanos por AUSA.

RF44: Exportar Trabajadores cubanos por AUSA.

RF45: Cargar Trabajadores cubanos por AUSA.

RF46: Obtener reportes de Trabajadores cubanos por AUSA.

RF47: Gestionar Vehículos.

- Insertar Vehículos.
- Actualizar Vehículos.
- Eliminar Vehículos.
- Buscar Vehículos.

RF48: Mostrar Vehículos.

RF49: Exportar Vehículos.

RF50: Cargar Vehículos.

RF51: Obtener reportes de Vehículos.

2.5.2 Requisitos Funcionales: Área de Entidades extranjeras con intereses cubanos.

RF1: Gestionar Tabla de Control de entidades en el exterior.

- Insertar Tabla de Control de entidades en el exterior.

Capítulo II: Descripción y análisis de la solución propuesta

- Actualizar Tabla de Control de entidades en el exterior.
- Eliminar Tabla de Control de entidades en el exterior.
- Buscar Tabla de Control de entidades en el exterior.

RF2: Mostrar Tabla de Control de entidades en el exterior.

RF3: Realizar reportes de entidades.

RF4: Exportar Tabla de Control de entidades en el exterior.

RF5: Cargar Tabla de Control de entidades en el exterior.

RF6: Obtener reportes de la Tabla de Control de entidades en el exterior.

2.6 Requisitos no funcionales

Los requerimientos no funcionales hacen relación a las características del sistema que aplican de manera general como un todo, más que a rasgos particulares del mismo. Estos requerimientos son adicionales a los requerimientos funcionales que debe cumplir el sistema, y corresponden a aspectos tales como la disponibilidad, flexibilidad, seguridad, facilidad de uso, etc.

Usabilidad

1- El sistema podrá ser empleado por cualquier persona que tenga conocimientos básicos de computación y de un ambiente web

2- El servidor donde se encuentre instalada la aplicación debe estar situado en un local protegido, donde no esté expuesto a desastres naturales o robo.

Fiabilidad

Capítulo II: Descripción y análisis de la solución propuesta

1- Deben crearse diferentes cuentas de usuario y asignarle a cada una solo los permisos que le corresponden.

2- El sistema debe ofrecer mensajes de verificación antes de ejecutar acciones irreversibles.

Eficiencia

1- El sistema debe responder en un tiempo razonable ante todos los eventos. Por lo que el tiempo máximo de respuesta será el de interacción con la base de datos.

Restricciones de diseño

1- Las variables serán declaradas en minúscula, si están compuesta por más de una palabra la siguiente comenzará con letra inicial mayúscula, las funciones serán comentariadas al comienzo de su declaración

Interfaz de Hardware

1- La interfaz debe ser amigable y fácil de manejar por el usuario

2- *El sistema debe mostrar a cada usuario sólo las funcionalidades de la aplicación sobre las cuales tiene permiso de acceso.*

3- Al tratarse de datos personales, las comunicaciones deben producirse bajo una conexión segura y cifrada.

4- El sistema va a estar disponible las 24 horas.

5- Las PCs clientes, deberán estar equipadas como mínimo con un 1Gb de memoria.

2.7 Descripción de las clases persistentes.

A continuación se ofrece la descripción de las clases persistentes modeladas en el diagrama anterior. En cada descripción se especifica el nombre, atributos, tipo de cada uno de estos, y descripción tanto de la clase como de los atributos. Para la definición de estas clases se evitó el uso de caracteres extraños en sus

Capítulo II: Descripción y análisis de la solución propuesta

nombres como vocales con tildes y la letra “ñ”, por tal motivo, en el presente documento aparecerán palabras sin tilde o con una “n” sustituyendo la “ñ” y un caso específico el de la palabra “Año” que en este caso se sustituyó la “ñ” por doble “n” , para respetar así la forma original en que aparecen en el diseño.

Área Sucursales y agentes de sociedades mercantiles extranjeras

Tabla 2.1: Descripción de la clase dgee_expediente

Entidad:	dgee_expediente		
Descripción:	Esta entidad contiene toda la información que se va a procesar de las sucursales y agentes de sociedades mercantiles extranjeras		
Relaciones:	dgee_direccion_gestion_empresarial_externa, dgee_agencias		
Campos	Tipo de Dato	Tamaño	Descripción
num_expediente	Entero (Identificador)	100	Llave primaria
pais	Cadena caracteres	100	Nombre del país
responsable	Cadena caracteres	100	Nombre del especialista que lleva la tramitación de la sucursal
estado	Cadena caracteres	100	Puede ser activo , pasivo o tramitación

Capítulo II: Descripción y análisis de la solución propuesta

nombre_compania	Cadena caracteres	100	Nombre de la compañía
id_direccion	Entero	100	Presenta el id de la dgee que es atributo único

Tabla 2.2: Descripción de la clase dgee_agencias

Entidad:	dgee_agencias		
Descripción:	Esta entidad contiene datos generales de todas las compañías existentes en el país		
Relaciones:	dgee_expediente,dgee_sucursales_vigentes_y_canceladas(herencia),dgee_compañías_representadas_por_agentes (herencia),dgee_oficinas_secundarias,dgee_vehículos,dgee_trabajadores_cubanos,dgee_permiso_de_trabajo_de_los_extranjeros		
Campos	Tipo de Dato	Tamaño	Descripción
id_agencias	Entero (Identificador)	100	Llave primaria
nombre_compania	Cadena caracteres	100	Nombre de la compañía
num_expediente	Entero	100	Presenta el número del expediente

Capítulo II: Descripción y análisis de la solución propuesta

Tabla 2.3: Descripción de la clase dgee_sucursales_vigentes_y_canceladas

Entidad:	dgee_sucursales_vigentes_y_canceladas		
Descripción:	Esta entidad contiene toda la información de las sucursales que operan actualmente en el país y las que han sido canceladas sus licencias		
Relaciones:	dgee_agencias(herencia)		
Campos	Tipo de Dato	Tamaño	Descripción
id_agencias	Entero (Identificador)	100	Llave foránea primaria
pais	Cadena caracteres	100	Nombre del país
giro_comercial	Cadena caracteres	100	Presenta el producto que se vende en la compañía
telefono	Entero	100	Número de teléfono
email	Cadena caracteres	100	Dirección de correo
direccion_cuba	Cadena caracteres	100	Presenta la dirección que tiene en cuba

Capítulo II: Descripción y análisis de la solución propuesta

representante	Cadena caracteres	100	Nombre del representante
---------------	-------------------	-----	--------------------------

Tabla 2.4: Descripción de la clase

dgee_companias_representadas_por_agentes

Entidad:	dgee_companias_representadas_por_agentes		
Descripción:	Esta entidad contiene los datos de la compañía y del agente el cual los representa		
Relaciones:	dgee_agencias(herencia)		
Campos	Tipo de Dato	Tamaño	Descripción
id_agencias	Entero (Identificador)	100	Llave foránea primaria
nombre_agente	Cadena caracteres	100	Nombre del agente
estado	Cadena caracteres	100	Este puede ser activa o canceladas
fecha_inscripcion	Date	100	Presenta la fecha que se va a inscribir

Tabla 2.5: Descripción de la clase dgee_oficinas_secundarias

Entidad:	dgee_oficinas_secundarias
-----------------	---------------------------

Capítulo II: Descripción y análisis de la solución propuesta

Descripción:	Esta entidad contiene los datos de las oficinas secundarias que pertenece a una determinada compañía		
Relaciones:	dgee_oficinas_secundarias		
Campos	Tipo de Dato	Tamaño	Descripción
id_oficina_secundaria	Entero (Identificador)	100	Llave primaria
nombre_compania_secundaria	Cadena caracteres	100	Nombre de la compañía secundaria
provincia	Cadena caracteres	100	Presenta el nombre de la provincia
municipio	Cadena caracteres	100	Presenta el nombre del municipio
direccion_radica	Cadena caracteres	100	Dirección donde radica
id_agencias	Cadena caracteres	100	Presenta el id de agencia que es atributo único

Tabla 2.6: Descripción de la clase dgee_vehiculos

Entidad:	dgee_vehículos
-----------------	----------------

Capítulo II: Descripción y análisis de la solución propuesta

Descripción:	Esta entidad contiene los datos de los vehículos asignados a cada compañía		
Relaciones:	dgee_agencias		
Campos	Tipo de Dato	Tamaño	Descripción
chapa	Entero (Identificador)	100	Llave primaria
nombre_compania	Cadena caracteres	100	Nombre de la compañía
estado	Cadena caracteres	100	Este puede ser apto, denegado o baja
fecha	Date	100	Presenta la fecha
id_agencias	Cadena caracteres	100	Presenta el id de agencia que es atributo único

**Tabla 2.7: Descripción de la clase
dgee_permiso_de_trabajo_de_los_extranjeros**

Entidad:	dgee_permiso_de_trabajo_de_los_extranjeros
Descripción:	Esta entidad contiene los datos de la persona a la que se le asigna el permiso y el estado en el cual

Capítulo II: Descripción y análisis de la solución propuesta

	está el permiso		
Relaciones:	dgee_agencias		
Campos	Tipo de Dato	Tamaño	Descripción
numero_permiso_trabajo	Entero (Identificador)	100	Llave primaria
nombre_compania	Cadena caracteres	100	Nombre de la compañía
cargo	Cadena caracteres	100	Presenta el nombre del cargo
estado	Cadena caracteres	100	Este puede ser vigente o cancelado
otorgado	Cadena caracteres	100	Presenta si es otorgado o no
observaciones	Cadena caracteres	100	Presenta las observaciones que se realizan
vigencia	Cadena caracteres	100	Presenta la vigencia
nombre_extranjeros	Cadena caracteres	100	Presenta el nombre del extranjero
ciudadania	Cadena caracteres	100	Presenta la ciudadanía del extranjero
id_agencias	Cadena caracteres	100	Presenta el id de

Capítulo II: Descripción y análisis de la solución propuesta

			agencia que es atributo único
--	--	--	-------------------------------

Tabla 2.8: Descripción de la clase dgee_trabajadores_cubanos

Entidad:	dgee_trabajadores_cubanos		
Descripción:	Esta entidad contiene los datos de todos los trabajadores cubanos contratados por alguna compañía		
Relaciones:	dgee_agencias,dgee_acorec(herencia), dgee_ausa(herencia)		
Campos	Tipo de Dato	Tamaño	Descripción
ci_trabajador	Entero (Identificador)	11	Llave primaria
apellidos	Cadena caracteres	100	Presenta el apellido del trabajador
cargo	Cadena caracteres	100	Presenta el cargo del trabajador
nombre	Cadena caracteres	100	Presenta el nombre del trabajador
id_agencias	Cadena caracteres	100	Presenta el id de

Capítulo II: Descripción y análisis de la solución propuesta

			agencia que es atributo único
--	--	--	-------------------------------

Tabla 2.9: Descripción de la clase dgee_acorec

Entidad:	dgee_acorec		
Descripción:	Esta entidad contiene los datos de los trabajadores cubanos contratados por Acorec		
Relaciones:	dgee_trabajadores_cubanos(herencia)		
Campos	Tipo de Dato	Tamaño	Descripción
ci_trabajador	Entero (Identificador)	11	Llave foránea primaria
nombre_compania	Cadena caracteres	100	Nombre de la compañía

Tabla 2.10: Descripción de la clase dgee_ausa

Entidad:	dgee_ausa
-----------------	-----------

Capítulo II: Descripción y análisis de la solución propuesta

Descripción:	Esta entidad contiene los datos de los trabajadores cubanos contratados por AUSA		
Relaciones:	dgee_trabajadores_cubanos(herencia)		
Campos	Tipo de Dato	Tamaño	Descripción
ci_trabajador	Entero (Identificador)	11	Llave foránea primaria
tipo_arrendamiento	Cadena caracteres	100	Presenta el tipo de arrendamiento
sucursal_extranjera	Cadena caracteres	100	Presenta la sucursal extranjera a la que representa
entidad	Cadena caracteres	100	Presenta el nombre de la entidad

Tabla 2.11: Descripción de la clase

dgee_control_de_tramitaciones_de_expedientes

Entidad:	dgee_control_de_tramitaciones_de_expedientes
Descripción:	Esta entidad contiene todos los datos que se tiene en cuenta para realizar una tramitación de un expediente

Capítulo II: Descripción y análisis de la solución propuesta

Relaciones:	dgee_dirección_gestión_empresarial_externa, dgee_renovaciones_sucursales_en_el_año		
Campos	Tipo de Dato	Tamaño	Descripción
id_tramitacion	Entero (Identificador)	100	Llave primaria
dgee_impreso	Date	100	Presenta la fecha de entrada documentos
responsable	Cadena caracteres	100	Presenta el nombre del especialista
nombre_compania	Cadena caracteres	100	Nombre de la compañía
pais	Cadena caracteres	100	Presenta en nombre del país
vencimiento	Date	100	Presenta fecha en que se vence la licencia
situacion	Cadena caracteres	100	Puede ser Aprobada, Denegada, Cancelada
observaciones	Cadena caracteres	100	Presenta las observaciones que se realizan
dgee_digital	Date	100	Presenta la fecha de entrada

Capítulo II: Descripción y análisis de la solución propuesta

			documentos
tramitacion	Cadena caracteres	100	Puede ser inscripción, renovación y otros
id_direccion	Entero	100	Presenta el id de la dgee que es atributo único

Tabla 2.12: Descripción de la clase

dgee_renovaciones_sucursales_en_el_año

Entidad:	dgee_renovaciones_sucursales_en_el_año		
Descripción:	Esta entidad contiene los datos necesarios que debe presentar una empresa para realizar una renovación		
Relaciones:	dgee_control_de_tramitaciones_de_expedientes		
Campos	Tipo de Dato	Tam año	Descripción
id_renovacion	Entero (Identificador)	100	Llave primaria
vigencia	Date	100	Presenta la fecha hasta cuando está vigente

Capítulo II: Descripción y análisis de la solución propuesta

fecha_inscripcion	Date	100	Presenta la fecha en que se inscribió
fecha_ultima_renovacion	Date	100	Presenta la fecha de la última renovación
nombre_compania	Cadena caracteres	100	Nombre de la compañía
id_tramitacion	Entero	100	Presenta el id de control de tramitaciones de expedientes que es atributo único

Área de Entidades extranjeras con intereses cubanos

Tabla 2.13: Descripción de la clase esquema externo

Entidad:	esquema externo		
Descripción:	Esta entidad contiene toda la información que se va a procesar en esta dirección		
Relaciones:	plan		
Campos	Tipo de Dato	Tam año	Descripción
numero_expediente	Entero (Identificador)	100	Llave primaria
denominacion_social	Cadena	100	Nombre de la

Capítulo II: Descripción y análisis de la solución propuesta

	caracteres		denominación social
actividad_especifica	Cadena caracteres	100	Presenta el nombre de la actividad específica
fecha_inscripcion_paises_origen	Date	100	Presenta la fecha de inscripción del país de origen
fecha_constitucion	Date	100	Presenta la fecha de constitución
titular_otras_companias	Cadena caracteres	100	Se utiliza para titular otras compañías
capital_autorizado	Cadena caracteres	100	Presenta el nombre del capital autorizado
vigencia_compania	Date	100	Presenta la fecha de la vigencia de la compañía
accionista_E	Cadena caracteres	100	Presenta el nombre del accionista_E
inmuebles_propios	Cadena caracteres	100	Presenta si hay o no inmuebles propios y su dirección
capital_suscrito	Cadena caracteres	100	Presenta el capital_suscrito
dividendos_obtenidos	Double	100	Presenta el resultado de los dividendos obtenidos
dividendos_remesados	Double	100	Presenta el resultado de los dividendos remesados

Capítulo II: Descripción y análisis de la solución propuesta

oficina_en_el_exterior	Cadena caracteres	100	Presenta si se encuentra o no en el exterior y su dirección
personal_cubano	Cadena caracteres	100	Presenta el número y nombre ejecutivo principal
capital_pagado	Cadena caracteres	100	Presenta la cantidad y el tipo de moneda
acciones_o_participantes_C	Cadena caracteres	100	Presenta el nombre de las acciones_o_participantes_C
personal_extranjero	Entero	100	Presenta la cantidad de personal extranjero
resultado_economico	Double	100	Presenta el resultado económico
observaciones	Cadena caracteres	500	Presenta todas la observaciones realizadas
oficina_en_cuba	Cadena caracteres	100	Presenta si se encuentra o no en cuba y su dirección
accionista_C	Cadena caracteres	100	Presenta el nombre del accionista_C
documentos_que_faltan	Cadena caracteres	100	Presenta los documentos que faltan

Capítulo II: Descripción y análisis de la solución propuesta

estado	Cadena caracteres	100	Puede ser activa, inactiva, proceso disolución y disuelta
acciones_bcc	Cadena caracteres	100	Presenta el nombre del acciones_bcc
total_acciones_o_part icipantes	Cadena caracteres	100	Presenta la cantidad de acciones o los participantes
pais	Cadena caracteres	100	Presenta el nombre del país
tipo_societario	Cadena caracteres	100	Puede ser sociedad anónima, sociedad responsabilidad limitada u otro
aprobada_por	Cadena caracteres	100	Presenta el nombre de la persona que lo aprobó
clasificador_actividad	Cadena caracteres	100	Presenta el clasificador y su actividad
objeto_social	Cadena caracteres	100	Presenta el objeto social
oace	Cadena caracteres	100	nta el Organismo de la stración Central del) y posibilidad de otros
registro	Entero	100	Presenta el número del registro

Capítulo II: Descripción y análisis de la solución propuesta

Conclusiones Parciales

En el presente capítulo se ha planteado y descrito el diseño de las bases de datos.

Se ha realizado la descripción de las tablas en detalles, así como sus respectivos atributos y el tipo de dato a almacenar en cada caso. El diseño elaborado cubre todos los requisitos funcionales planteados para las bases de datos por parte de cada uno de los módulos del sistema. Además se analizaron varios aspectos fundamentales que deberán cumplir las bases de datos para garantizar la seguridad e integridad de los datos.

Capítulo 3: Implementación y validación de la base de datos.

Introducción

En el presente capítulo se realiza la validación teórica del diseño de la base de datos y se describirá detalladamente el transcurso de la implementación. Así como también se explicará cómo se lleva a cabo el proceso de integración de la aplicación cliente y servidora con la capa de persistencia de los datos. En fin este capítulo es el complemento del trabajo realizado en el diseño de la base de datos e implementación de las capas de acceso a datos.

3.1 Estructura de los archivos mediante paquetes.

La organización es un factor clave en cuanto a estandarización de código se refiere, puesto que en la capa servidora se implementan grandes cantidades de este no solo por parte del creador del servidor sino también por el desarrollador de la base de datos, es necesario establecer una estructura para poder identificar que archivos son creados y por quien. De esta forma cada cual trabaja en la parte que le corresponde y no tiene por qué tener en cuenta lo que implementen los demás.

Partiendo de analizar de quienes son los que trabajan directamente en el desarrollo del código del servidor, así como los archivos que debían ser creados y que correspondían a líneas de implementación diferentes; se elaboró una estructura general del proyecto para la capa servidora regida por paquetes. Un Paquete en Java es un contenedor de clases que permite agrupar las distintas partes de un programa cuya funcionalidad tienen elementos comunes. El uso de los mismos proporciona una serie de ventajas como son: la reutilización de código, mayor seguridad al existir niveles de acceso, entre otras.

En la estructura que fue diseñada, el desarrollador de la base de datos tiene 2

Capítulo III: Implementación y validación de la base de datos

paquetes que constituyen sus diferentes líneas de implementación, donde almacenará todo el código que se cree.

El primero **model.entity**: es el que contiene el modelaje de las clases abstractas generales. Por último **model.dao** es el paquete que contiene la programación relacionada con el lenguaje HQL de hibernate, dígame la creación de consultas para la gestión y recuperación de los datos etc.

3.2 Creación y estructuración de las clases DAO.

La aplicación cuenta con 2 modelos de objetos de acceso a datos (clases DAO) que son usados para aislar la aplicación de la tecnología de persistencia subyacente hibernate. Proporcionando que dicha tecnología pueda ser actualizada o cambiada sin modificar otras partes de la aplicación, dándole total libertad al desarrollador del servidor del sistema informativo del MINCEX para que implemente su parte sabiendo que esta no será afectada por cambios realizados dentro de los paquetes correspondientes al desarrollador de base de datos. Estas clases DAO contienen también métodos que constituyen el único modo de acceder o gestionar los datos desde la aplicación servidora hacia la base de datos.

La clase **GenericDao** es la más importante de las 2 creadas, porque posee una serie de métodos genéricos que son los que permiten realizar las operaciones relacionadas con un CRUD de datos; que no es más que el acrónimo de Crear, Obtener, Actualizar y Borrar (del original en inglés: Create, Read, Update and Delete). Es usado para referirse a las funciones básicas en bases de datos o la capa de persistencia en un sistema de software.

3.3 Descripción de los métodos implementados.

Los métodos implementados en la clase **GenericDao** son los siguientes:

saveOrUpdate: recibe como parámetro un objeto del tipo AbstractEntity que constituye la clase principal de la que heredan todas las demás clases resultantes del mapeo de objeto relacional; el mismo inserta en la base de datos este objeto o
o modifica si ya existe

Capítulo III: Implementación y validación de la base de datos

Save: recibe como parámetro un objeto AbstractEntity también, pero a diferencia del saveOrUpdate solo lo inserta en la base de datos si no está.

Update: este también recibe como parámetro un AbstractEntity lo busca en la base de datos por la clave primaria y si está lo modifica.

Find: este método recibe como parámetro un tipo de clase correspondiente a las entidades mapeadas devolviendo una lista con los objetos de esa clase que están en la base de datos.

FindByPK: recibe por parámetros un tipo de clase y una llave primaria devolviendo el objeto de esa clase correspondiente a esa llave que se encuentra en la base de datos.

Para eliminar los datos existen dos métodos, uno que elimina un solo objeto de la base de datos pasado por parámetros, denominado **delete** y otro que elimina toda una colección llamado **deleteAll**. También existe dos métodos relacionados con la búsqueda por parámetros, a los cuales se les pasa el nombre de un campo y el valor de este devolviendo una lista de los objetos que cumplen estos requisitos estos el **findByParamAndValue** y **findByParamsAndValues** que realiza la misma operación pero con varios campos y valores.

La otra clase EspecificDao posee un solo método llamado requisitoGeneral el cual su función es devolver una cierta cantidad de información de tablas diferentes mediante un número de expediente específico entrado por el usuario.

3.4 Integridad de los datos

La integridad en una BD se refiere a la corrección y exactitud de la información contenida. Cualquier BD puede estar sujeta a una serie de restricciones de integridad de diversa complejidad. La integridad de los datos puede verse afectada de diferentes maneras al realizar modificaciones al contenido con sentencias INSERT, UPDATE o DELETE, a través de adiciones de datos inválidos y de

Capítulo III: Implementación y validación de la base de datos

modificaciones de datos existentes haciendo que tomen valores incorrectos o eliminándolos.

La integridad de los datos agrupa en varias categorías como:

3.4.1 Integridad de entidad

Relacionada con las llaves primarias de las tablas, estableciendo que ningún atributo que forme parte de la llave primaria puede aceptar valores nulos. Por definición, una llave primaria es un identificador irreducible que se utiliza para identificar de modo único las tuplas. Que es irreducible significa que ningún subconjunto de la llave primaria sirve para identificar las tuplas de modo único. En la BD propuesta se define una llave primaria para cada tabla, la cual no puede ser nula.

3.4.2 Integridad de clave

Establece que el atributo clave o llave primaria no puede tomar valores iguales en tuplas diferentes. En la BD propuesta se garantiza que las llaves primarias no pueden repetirse, las mismas son de tipo integer, generándose incrementalmente.

3.4.3 Integridad de dominio

Una restricción de integridad de dominio es una regla que define los valores válidos de cada uno de los atributos de las tablas en la BD. En ocasiones es necesario definir más de una restricción de dominio para describir por completo un dominio.

Existe integridad de dominio básica, como no poder introducir letras en campos donde se va a almacenar números. Estas normas o reglas de integridad de dominio pueden indicar qué campos son necesarios tener obligatoriamente con valores, para que la base de datos no tenga datos sin conectar en el caso de tener relaciones o dependencias entre tablas. Se hace uso de los tipos de datos predefinidos por el gestor utilizado, específicamente: integer, varchar, date, tratando de no poner campos nulos para evitar resultados inconsistentes, a pesar que en algunos casos es necesario.

3.4.4 Integridad referencial

Garantiza que las relaciones que se establezcan entre las tablas de la BD sean coherentes, manteniendo la consistencia entre las tuplas de las dos relaciones, aplicándose específicamente a las llaves foráneas. Si se tiene en una tabla una columna declarada como llave foránea, sus valores deben coincidir con los valores presentes en la llave primaria de la tabla a la cual se hace referencia, de esta forma no se puede agregar un registro en la tabla que contiene la llave foránea si no existe uno que lo respalde, o sea, que exista en la tabla vinculada. Permite además realizar actualizaciones y eliminaciones en cascada, asegurando que las modificaciones que se realicen en una tabla se reflejen en la otra.

3.5 Normalización

El proceso de normalización se efectúa en el momento de realizar el modelo lógico de BD, con el objetivo de evitar anomalías de actualización en las estructuras de datos, asegurando así que las relaciones obtenidas no tengan datos redundantes.

La normalización es un proceso mediante el cual se dividen relaciones no convenientes, distribuyendo los atributos en relaciones más pequeñas que cumplan con una serie de requisitos o características deseables, siguiendo la premisa de eliminar redundancias y anomalías de inserción, eliminación y actualización.

El proceso de normalización es un estándar que consiste, básicamente, en un proceso de conversión de las relaciones entre entidades, evitando

- La redundancia de los datos: repetición de los datos en un sistema.
- Anomalías de actualización: inconsistencias de los datos como resultado de datos redundantes y actualizaciones parciales.
- Anomalías de borrado: pérdida no intencionada de datos debido a que se han borrado otros datos.
- Anomalías de inserción: imposibilidad de adicionar datos en la BD debido a la ausencia de otros datos.

Existen 9 formas normales, de las cuales las tres primeras son las más usadas:

Capítulo III: Implementación y validación de la base de datos

Primera forma normal (1FN)

Se dice que una tabla está en primera forma normal si y sólo si cada uno de los campos contienen un único valor para un registro determinado.

En esta forma normal están prohibidos los atributos multivaluados, compuestos y sus combinaciones, o sea que los atributos deben ser atómicos, que no puedan dividirse. En otras palabras, en esta fase no se permiten relaciones dentro de relaciones.

La BD propuesta cumple con la primera forma normal puesto que los atributos de cada una de las tablas son atómicos, no siendo ni multivaluados ni compuestos.

Segunda forma normal (2FN)

La primera condición que debe cumplir es que se encuentre en 1FN, luego que los atributos que no forman parte de la llave primaria, deben depender de manera total de los atributos que forman la llave primaria, lo cual establece que todas las dependencias parciales deben ser eliminadas. Esta forma normal se aplica a tablas que tienen una llave compuesta.

La segunda forma normal compara todos y cada uno de los campos de la tabla con la llave definida. Si todos los campos dependen directamente de la llave se dice que la tabla está en segunda forma normal (2FN).

La BD propuesta cumple con la segunda forma normal ya que además de estar en primera forma normal, los atributos de las tablas con llaves compuestas no presentan dependencias funcionales parciales.

Tercera forma normal (3FN)

Debe empezar cumpliendo que se encuentre en 2FN, luego que cada atributo de la relación que no está contenido dentro de la llave primaria dependa solo de la llave primaria y no de ningún otro atributo, o sea tiene como objetivo eliminar las dependencias transitivas. Entiéndase por dependencia transitiva aquella en la cual las columnas que no son llave son dependientes de otras columnas que tampoco son llave.

Se dice que una tabla está en tercera forma normal si y solo si los campos de la tabla dependen únicamente de la llave, dicho en otras palabras los campos de las tablas no dependen unos de otros.

La BD propuesta cumple con la tercera forma normal, pues todos los atributos de las tablas dependen de manera directa de la llave primaria.

3.6 Validación de la capa de acceso a datos.

La validación de la capa de acceso a datos se implementó a través de un tratamiento de excepciones (try catch). Excepciones que fueron identificadas

Capítulo III: Implementación y validación de la base de datos

mediante la realización de pruebas de estabilidad a dicha capa. A continuación se expondrán una serie de estas excepciones que fueron identificadas y tratadas.

org.postgresql.util.PSQLException: Excepción que es lanzada cuando no se puede tener acceso al servidor de base de datos, ya sea porque está apagado o bien porque el host o el puerto especificados no son los correctos.

org.springframework.dao.DataIntegrityViolationException: Excepción que es lanzada cuando se trata de acceder a un valor de clave primaria que no existe en la base de datos.

3.7 Pruebas de rendimiento

Durante todas las etapas del desarrollo de software las pruebas son fundamentales, a partir de ellas es posible controlar que los productos cumplan con los requisitos funcionales, además de verificar y revelar la calidad de un producto de software. Las pruebas constituyen una actividad en la cual un sistema o componente es ejecutado bajo condiciones específicas, se observan o almacenan los resultados y se realiza una evaluación de algún aspecto del sistema o componente.

3.7.1 Pruebas Unitarias

Las pruebas unitarias representan una forma de probar un fragmento de código por lo que tiene gran importancia para los desarrolladores y el equipo de pruebas. Aunque el tamaño de la "unidad" que se prueba puede variar, la herramienta JUnit en su versión 4.10 para pruebas del NetBeans permite generar pruebas unitarias específicas para métodos y funcionalidades.

3.7.2 Diseños de casos de pruebas

En la ingeniería del software, los casos de pruebas son un conjunto de condiciones o variables bajo las cuáles el analista determinará si el requisito de una aplicación es parcial o completamente satisfactorio. Con el propósito de comprobar que todos los requisitos de una aplicación son revisados, debe haber al menos un caso de prueba para cada requisito. Lo que caracteriza un diseño de caso de prueba es que hay una entrada conocida y una salida esperada, los cuales son formulados antes de que se ejecute la prueba.

A continuación se muestra el diseño de caso de prueba para todos los métodos de

Capítulo III: Implementación y validación de la base de datos

la clase “*GenericDao*”

Escenario	Respuesta	Parámetros	Descripción	Flujo
save	No retorna nada	<i>AbstractEntity</i> entity	Inserta en la base de datos una instancia de cualquiera de las entidades mapeadas	Chequear que el objeto haya sido insertado en la base de datos

Escenario	Respuesta	Parámetros	Descripción	Flujo
saveOrUpdate	No retorna nada	<i>AbstractEntity</i> entity	Inserta en la base de datos una instancia de cualquiera de las entidades mapeadas, o si existe la modifica	Chequear que el objeto haya sido insertado en la base de datos o modificado

Escenario	Respuesta	Parámetros	Descripción	Flujo
update	No retorna nada	<i>AbstractEntity</i> entity	Modifica en la base de datos una instancia de cualquiera de las entidades mapeadas insertada previamente	Chequear que el objeto haya sido modificado en la base de datos

Escenario	Respuesta	Parámetros	Respuesta	Flujo
-----------	-----------	------------	-----------	-------

Capítulo III: Implementación y validación de la base de datos

find	Retorna una lista de objetos de una entidad específica	<i>Class<T></i> <i>entityClass</i>	Busca en la base de datos todos los objetos de una entidad especificada	Chequear que los objetos retornados coincidan con los que están en la base de datos
------	--	---	---	---

Escenario	Respuesta	Parámetros	Descripción	Flujo
findByPK	Retorna un objeto de una entidad específica <T> AbstractEntity	1. <i>Class<T></i> <i>entityName</i> 2. <i>long id</i>	Busca en la base de datos un objeto de una entidad especificada por la clave primaria	Chequear que el objeto retornado coincida con el que está en la base de datos

Escenario	Respuesta	Parámetros	Descripción	Flujo
delete	No retorna nada	<i>AbstractEntity</i> <i>entity</i>	Elimina en la base de datos un objeto de una entidad especificada pasado como parámetro	Chequear que el objeto sea eliminado de la base de datos

Escenario	Respuesta	Parámetros	Descripción	Flujo
-----------	-----------	------------	-------------	-------

Capítulo III: Implementación y validación de la base de datos

findByParamAndValue	<code><T> List<T></code>	1. Class<T> entityClass 2. String param 3. Object value	Retorna una lista de objetos de una entidad, restringidos por parámetros y sus valores específicos	Chequear que los objetos retornados coincidan con los que están en la base de datos
---------------------	--------------------------------------	--	--	---

Para la implementación de los requisitos se tuvo en cuenta 3 iteraciones. En ellas se realizaron pruebas unitarias de cada uno de los requisitos correspondientes. Se aplicaron las pruebas a todos los métodos de los daos creados, a cada funcionalidad de estos le fueron generados los resultados esperados.

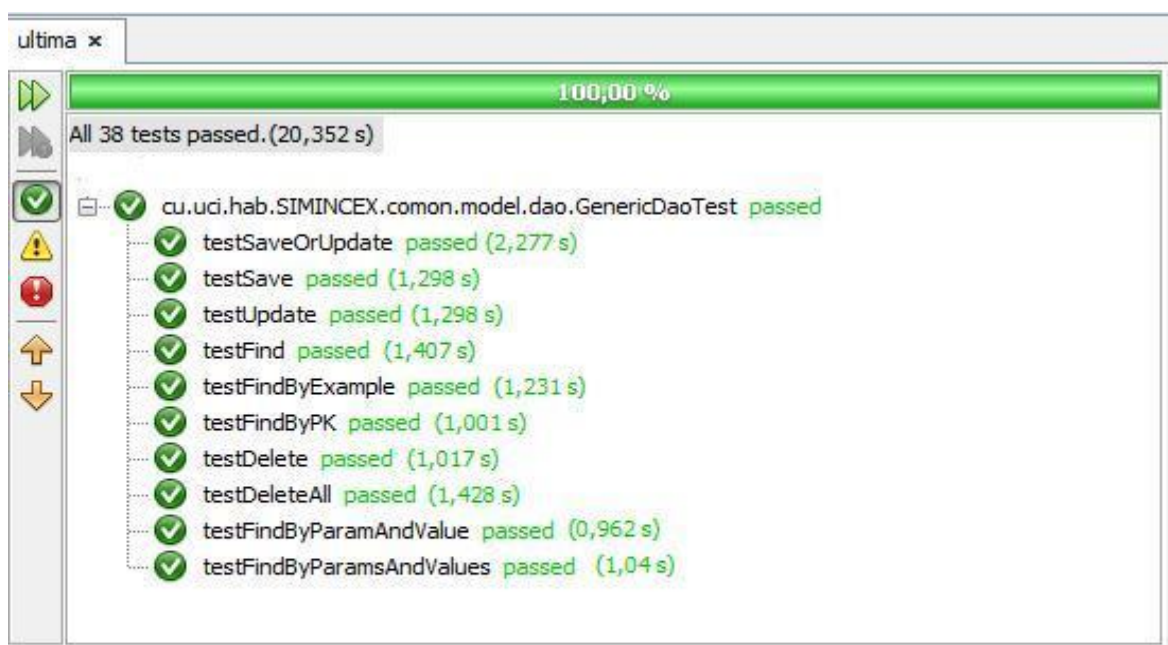


Fig. 4. Resultado final de las pruebas a los métodos de la clase "GenericDao".



Fig. 5. Gráfico de los resultados generales de las pruebas unitarias para todos los requisitos.

3.7.3 Prueba de Volumen

Esta prueba se centra, a través del cúmulo de información en la BD, en analizar el comportamiento de está verificando si la misma alcanza su límite de almacenamiento y pueda causar fallas. Se utilizó la herramienta Data Generator para PostgreSQL para el llenado voluminoso de la BD.

Capítulo III: Implementación y validación de la base de datos

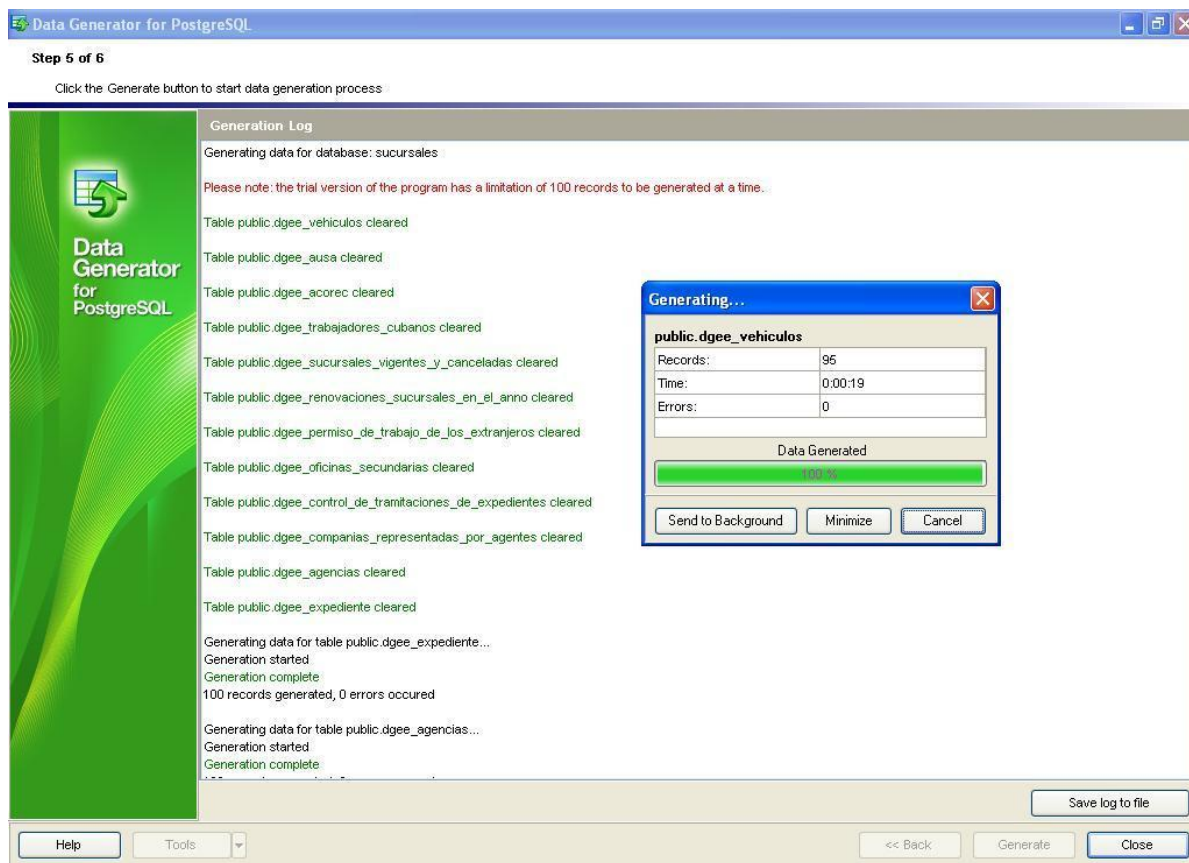


Fig. 6. Resultados de la generación de datos

Al introducir los datos en las tablas no se presentaron problemas de límites de capacidad, desbordamiento de columnas, atributos o tipos de datos. La utilización de esta herramienta permitió verificar la integridad de los datos, además de garantizar que el diseño de las estructuras de la BD y el gestor utilizado para el desarrollo soportan el cúmulo de información requerida para el funcionamiento de la BD.

3.8 Aporte social

En una investigación realizada al Ministerio del Comercio Exterior y la Inversión Extranjera se llegó a la conclusión de que sus trabajadores, por la gran cantidad de información que almacenaban sin un formato definido se demoraban en la realización del trabajo que realizan en este Ministerio, por lo que se dio a la tarea

Capítulo III: Implementación y validación de la base de datos

de informatizar cada una de las direcciones con las que cuenta para agilizar todo lo que se lleva a cabo en cuanto al almacenamiento de la gran cantidad de modelos que presenta el ministerio en general. Es de estos problemas que se presentan, que se da a la tarea de informatizarla, para un rápido y ágil proceso de la información que presenta.

Al medir los resultados que se obtendrá, se pueden establecer pronósticos que ayuden almacenar y recuperar la información que se maneja en cada una de las direcciones existentes en el MINCEX así como establecer líneas de trabajo en función de mejorar la calidad de todo lo que se maneje. Esta información además es de mucha utilidad para todos los trabajadores de la Dirección de Gestión Empresarial Externa y para el ministerio en general. Todo esto, en conjunto, contribuye a mejorar de forma significativa, la calidad del trabajo que aquí se realiza y en general la satisfacción del usuario como cliente final.

Conclusiones Parciales

En este capítulo se realizó un análisis de varios aspectos de gran importancia para poder demostrar que el modelado de las bases de datos así como su implementación y validación de la capa de acceso a datos fueron las más óptimas.

Ya que mediante ellas fue que se transformó en un software de prototipo funcional y cumpliendo con todos los requisitos que debe de tener un producto de alta calidad.

Conclusiones Generales

Concluido el presente trabajo se puede afirmar que se ha cumplido con el objetivo del mismo a través del desarrollo de las capas de acceso a datos que permitió el almacenamiento y recuperación de la información del Módulo Dirección de Gestión Empresarial Externa. Del cual se tienen las siguientes conclusiones:

- Durante la investigación se realizó un estudio acerca de los elementos teóricos principales relacionados con el diseño de base de datos que favoreció el desarrollo del trabajo permitiendo sentar las bases para un correcto diseño de BD.
- Se realizó el diseño de la BD, el cual cumple con los requerimientos especificados, aplicando diversas soluciones que contribuyeron a mejorar el diseño.
- Se implementó la capa de acceso a datos, lo cual nos permitió utilizar los métodos más comunes que se denominan Crud (crear, buscar, actualizar o eliminar)
- Se analizó el comportamiento de la BD ante situaciones determinadas a través de diferentes pruebas, a las cuales respondió satisfactoriamente.

De manera general se obtuvo una BD que permite el almacenamiento de la información requerida, permitiendo mayor organización y control de la misma.

Recomendaciones

Debido a los resultados de todo el proceso de investigación realizado y basado en la experiencia adquirida se proponen las siguientes recomendaciones:

- Que se tenga presente el diseño propuesto para futuras versiones
- Dar continuidad al trabajo realizado en pos de robustecer el diseño.
- Mantener sobre la BD un estricto cumplimiento del proceso de mantenimiento y copias de seguridad periódicas, logrando así que se mantenga la fiabilidad y funcionamiento óptimo de la BD.

Referencias Bibliográficas

Alma, Enríquez Toledo. 2006. MySQL. [Online] 2006. [Cited: diciembre 03, 2011.] <http://www.uaem.mx/posgrado/mcruz/cursos/miic/MySQL.pdf>.

Alvarez, Sara. 2007. Equipo DesarrolloWeb.com. [Online] julio 31, 2007. [Cited: noviembre 30, 2011.] <http://www.desarrolloweb.com>.

Benitez, R and Roa, I. 2002. Bases de Datos de Orientadas a Objetos. 2002.

desarrolloweb.com. 2002. Modelos de bases de datos. [Online] 2002. [Cited: noviembre 26, 2011.] <http://www.desarrolloweb.com>.

PostgreeSQL. 2008. PostgreeSQL. [Online] 2008. [Cited: diciembre 2, 2011] <http://grupove.org.ve/?q=ventajas>

Sánchez, Jorge. 2004. Principios sobre Bases de Datos Relacionales. [Online] 2004. [Cited: noviembre 25, 2011.] <http://creativecommons.org/licenses/by-nc-sa/2.0/>.

W.Hansen, Gary and Hansen, James V. 2006. Diseño y Administración de Base de Datos. 2da Edición. 2006.

Acevedo. 2009. [En línea] 2009. http://docs.jboss.org/gestores1_de_bd_single.html.

Date. 2006. [En línea] 2006. http://www.mariapinto.es/e-coms/base_de_datos_infor.htm.

Leyva. 2008. [En línea] 2008. <http://gladisutc.blogspot.com/2008/05/modelo-de-datos.html>.

Perez. 2007. [En línea] 2007. <http://www.buenastareas.com/ensayos/Conceptos-De-Las-Bases-De-Datos/2663772.html>.

Uranga. 2009. [En línea] 2009. <http://www.srbyte.com/2009/09/que-es-base-de-datos.html>.

Referencias Bibliográficas

Gonzalez, Henry Raul. Knol. [En línea] 2009.

<http://knol.google.com/k/henryraul/erp-cuba/3hor39g24xzh5/5#>.

Definición.de. Definición.de. [En línea] 2008. <http://definicion.de/modelo-de-datos/>.

García Chávez, Carlos Alberto . [En línea] 2005. <http://www.mailxmail.com/curso-disenio-base-datos-relacionales/planificacion-disenio-administracion-bases-datos>.

RODAS, C. [En línea] 2009. <http://www.slideshare.net/ChristianR/diseo-de-base-de-datos>.

CASELLES,J. [En línea] 2008. <http://www.coninteres.es/sql/Normalizacion.pdf>.

PostgreeSQL. 2008. PostgreeSQL. [En línea] 2008. [Citado el: 10 de diciembre de 2008.] <http://grupove.org.ve/?q=ventajas>.

MySQL. 2008. About MySQL. MySQL. [En línea] 2008. [Citado el: 18 de enero de 2009.] <http://www.mysql.com/about/>.

Sanchez, María A. Mendoza. 06. *Informatizate*. [En línea] 7 de 7 de 06. [Citado el: 21 de 11 de 09.]

http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.htm

Microsoft; Integridad de los datos, [citado: marzo de 2009]; Disponible en: <http://msdn.microsoft.com/es-es/library/ms184276.aspx>

Dr. Conejo Muñoz, Ricardo Tipos de Bases de Datos.[citado: noviembre de 2008];Disponible en: <http://www.lcc.uma.es/~galvez/ftp/bdst/Tema2.pdf>

Glosario de términos

1. Atributos: es la unidad menor de información sobre un objeto almacenado en la BD. Son propiedades que poseen las entidades, por ejemplo: el color, el nombre, identificador, o cualquier otro elemento que describa a la entidad que se necesite conocer.

2. Llave: atributo (o conjunto de atributos) que identifica a una entidad como un objeto único e irreplicable dentro de la base de datos. Ejemplo: la llave de la entidad PERSONA puede ser el NoCI (número de carnet de identidad); en la entidad PIEZA la llave puede ser el número de serie.

3. Claves foráneas: utilizadas para relacionar tablas.

4. Tupla: es el nombre que recibe cada una de las filas de la tabla.

5. Normalización: es un proceso que permite obtener relaciones de la forma plana bidimensional, logrando que las tablas de datos tengan la forma adecuada, para evitar tanto la ocurrencia de anomalías de actualización como la falta de consistencia de los datos.

6. PostgreSQL: es un sistema de gestión de base de datos relacional orientada a objetos y libre.

8. ORM: mapeador de Objetos Relacionales.

9. Requisitos funcionales (del sistema): capacidades o condiciones que el producto debe cumplir.

10. Requisitos no funcionales (del sistema): propiedades o cualidades que el producto debe tener.

11. Redundancia: repetición de una información previamente existente.

12. Mincex: El Ministerio del Comercio Exterior y la Inversión Extranjera.

13. DGEE: Dirección de Gestión Empresarial Externa