



Universidad de las Ciencias Informáticas
Facultad Regional de Artemisa

**Trabajo de Diploma para optar por el Título de Ingeniero
en Ciencias Informáticas.**

*Base de datos para la gestión de la información
de Exportación y Sustitución de Importaciones
del Ministerio del Comercio Exterior y la
Inversión Extranjera.*

Autor: Yannel Orlando Rivera Fernández.
Tutor: Ing. Isandra Cruz Díaz.
Co-tutor: Msc. Héctor Estévez Noguera.

Junio del 2012
“Año 54 de la Revolución”



*"EL QUE NO TENGA EL VALOR DE SACRIFICARSE, POR LO MENOS DEBE
TENER EL PUDOR DE CALLARSE FRENTE A LOS QUE SE SACRIFICAN"*

JOSÉ MARTÍ

Declaración de autoría

Declaro que soy el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas para que hagan el uso que estimen pertinente con este.

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

Yanniel Orlando Rivera Fernández

Firma del Autor

Ing. Isandra Cruz Días

Firma del Tutor

A mí madre.

A mí familia, en especial a mí abuela.

A mi mamá, Silvia Beatriz Fernández Bazaíl por ser la estrella que ha guiado de mi vida, en las buenas y en las malas siempre ha estado ahí cuando más lo he necesitado. “ Mamí, te adoro”.

A mi abuela, Caridad Pérez Ramos por dedicar su vida a mí.

A mi padrastro, Pedro Jesús González Ortega, por enseñarme y educarme hasta llegar a convertirme en lo que hoy soy.

A mi novia, Daílyn Pérez Lores, por ser la persona más maravillosa del mundo, le doy gracias a Dios de habérmela puesto en mi camino.

A mi papá, Orlando Rafael Rivera Pérez, por escucharme y entenderme y nunca reprocharme nada.

A mis tíos, Sergio, Esperanza, Orfilio y Belkís, por siempre preocuparse por mí y por mis estudios, se los agradezco mucho.

A mis primos, Yoniel, Elíer, Yunet y Yusleídys, por confiar en mí y siempre pedirme más de lo que daba.

Al mejor amigo del mundo, Adolfo Salas Peña, por siempre estar ahí, y nunca de su boca salir un no puedo para mí.

A toda mi familia, materna y paterna, en especial a Maruca.

A mi suegra, Nirsia Lores Reyes, que siempre se mantuvo orando por mí.

A todas mis amistades de la universidad, por esos momentos que pasamos juntos y que jamás olvidaré, en especial al Ruso, Nene, Yisel, Mirna, y Manso.

A mi tutora y co-tutor, por guiarme y estar a mi lado para poder llegar hasta el final de esta carrera.

A todas las personas que de una forma u otra estuvieron pendientes de mis estudios y siempre me dieron una palabra de ánimo.

Como resultado del avance de las Tecnologías de la Información y las Comunicaciones (TICs) han surgido numerosas herramientas informáticas aplicables en disímiles ámbitos, siendo las bases de datos (BD) una de las más importantes para el avance de la sociedad. Es por ello que en la actualidad muchas empresas optan por este elemento para garantizar la integridad y disponibilidad en el acceso y distribución de los datos.

El presente trabajo expone la propuesta de diseño de la base de datos y el diseño e implementación de la capa de acceso a datos o capa de persistencia para la dirección de Exportación y Sustitución de Importaciones del Ministerio del Comercio Exterior y la Inversión Extranjera, teniendo en cuenta los problemas existentes en esta dirección con la seguridad y centralización de la información y las consecuencias que esto acarrea.

La propuesta contribuye al correcto almacenamiento y recuperación de la información garantizando su seguridad y centralización. Los componentes implementados en la capa de persistencia constituyen la base para la implementación de los componentes de las capas superiores o específicamente de la capa de negocio, posibilitándole a los desarrolladores de la misma, centrarse en la implementación de las reglas de negocio o requerimientos del sistema.

Una vez confeccionada la propuesta, se sometió a evaluación mediante la realización de pruebas de unidad permitiendo la validación o evaluación de los componentes desarrollados para la persistencia de objetos en la aplicación.

Palabras Claves: Bases de datos, Capa de acceso a datos, Seguridad.

Introducción.....	1
Capítulo1: Fundamentación Teórica.....	7
Introducción.....	7
1.1 Conceptos asociados al dominio del problema.....	7
1.1.1 Gestión	7
1.1.2 Información	7
1.1.3 Gestión de la Información	8
1.1.4 Exportación.....	8
1.1.5 Sustitución de Importaciones	9
1.1.6 Seguridad de la información.....	9
1.1.7 Centralización de la información.....	10
1.2 Bases de Datos	10
1.2.1 Elementos de una base de datos	10
1.2.2 Tipos de base de datos.....	11
1.2.3 Modelos de bases de datos	11
1.2.4 Capa de acceso a datos.....	13
1.3 Soluciones Existentes.....	14
1.4 Metodología para el diseño de bases de datos relacionales.....	16
1.5 Herramientas y tecnologías.....	16
Capítulo 2: Diseño y Arquitectura de la solución propuesta	20
2.1 Arquitectura de software para la DESI.....	20
2.2 Descripción de la arquitectura de la base de datos.....	21
2.3 Arquitectura de la Capa de Acceso a Datos.....	24
2.4 Patrones de diseño a utilizar.....	25
2.5 Requisitos del sistema.....	25
2.5.1 Requisitos funcionales.....	26
2.5.2 Requisitos no funcionales.....	27
2.6 Descripción de las entidades del Diagrama Entidad-Relación.....	28

Capítulo 3: Implementación y validación de la solución propuesta.....	33
3.1 Estructura de los archivos mediante paquetes.....	33
3.2 Proceso de mapeo.	34
3.3 Estándares de codificación.....	34
3.3.1 Estilos de codificación.....	35
3.4 Validación teórica.	36
3.4.1 Integridad de entidad.	36
3.4.2 Integridad de dominio.....	37
3.4.3 Normalización.	38
3.5 Creación de las clases AbstractEntity y GenericDao.....	41
3.5.1 Creación del DAO Auxiliar “GenerarReporte”	46
3.6 Tratamiento de errores.	48
3.7 Validación funcional.	48
3.7.1 Pruebas Unitarias.....	49
3.7.2 Diseño de los casos de pruebas a la clase GenericDao.....	49
3.7.3 Resultado de las pruebas a la clase GenericDao.	51
3.7.4 Diseño de los casos de pruebas a la clase GenerarReporte.	54
3.7.5 Resultado de las pruebas a la clase GenerarReporte.	56
3.7.6 Conclusiones de las pruebas unitarias.	58
3.8 Pruebas de volumen.	59
Conclusiones Generales.....	60
Recomendaciones.....	61
Referencias Bibliográficas	62
Bibliografías Consultadas	64

Introducción

En las últimas décadas, la economía mundial experimentó un rápido crecimiento, al que contribuyó entre otros factores la aceleración del comercio internacional, fruto del avance tecnológico y de un esfuerzo concertado para reducir las barreras comerciales. Algunos países en desarrollo han abierto sus economías para aprovechar al máximo las oportunidades de desarrollo económico que ofrece el comercio internacional.

Muchas organizaciones y empresas optan por la actividad para su negocio de exportar los recursos que disponen así como importar aquellos que necesitan para satisfacer las necesidades y objetivos de su industria en general. La exportación consiste en trasladar o transferir un bien o servicio de un lugar a otro mediante el mercado, siendo el lugar que recibe el que está en presencia de una importación.

El mercado es donde se une la oferta y la demanda, es decir, es el lugar donde se contratan vendedores y compradores clasificándose este en tres clases: mercado nacional o interno, mercado externo e internacional, por lo cual es claro que el primero no es el único y en consecuencia se entiende que para poder exportar se deben tener en cuenta los otros dos mercados mencionados anteriormente.

Las exportaciones mejoran la economía de las empresas y por supuesto la de su país, motivo por el cual los agentes económicos de dichas empresas deben tender a la exportación y diseñar políticas económicas en dicho sentido al igual que desincentivar las importaciones, puesto que un país en el cual la importación juegue un papel fundamental tiende o está orientado al fracaso, por lo cual es claro que amerita realizar los estudios correspondientes.

Las exportaciones tienen grandes ventajas como son el crecimiento empresarial, el dinamismo de las ventas e ingresos y la estabilidad en los negocios al no depender del negocio interno. Además mejora la reducción del costo al manejarse mayores volúmenes, provocando mayor rentabilidad al generarse ingresos superiores, de este modo aumenta la competitividad de la empresa y la calidad de

sus productos, contribuyendo al desarrollo económico y social del país a través de la generación de empleos y divisas.

El proceso de sustitución de importaciones (PSI) comenzó a desarrollarse de manera espontánea, es una estrategia o modelo económico adoptado en el territorio latinoamericano y en otros países en desarrollo con posterioridad a la Segunda Guerra Mundial. Tiene como objetivo suplir las importaciones, por lo que aquellos bienes que antes importaba el país, pasan a ser producidos internamente.

Una de las principales ventajas de la sustitución de importaciones se basa en la existencia de un mercado local definido, el cual debe protegerse contra competidores externos que amenazan la industria nacional, esta posición se interpreta como una ayuda al crecimiento industrial del país.

A nivel mundial, en el transcurso de los últimos años se ha tenido como principal objetivo fomentar el comercio electrónico a través de las Tecnologías de la Información y las Comunicaciones (TICs), especialmente a través de su Programa de Centros de Comercio, que abarca la creación de centros relacionados al comercio a fin de reducir los costes de transacción y utilizar la tecnología para mejorar el acceso de las pequeñas y medianas empresas a la información, los servicios y las redes mundiales relacionadas con el comercio.

Una de las instituciones encargadas de fomentar el desarrollo del comercio electrónico nacional es la Universidad de las Ciencias Informáticas (UCI), creada al calor de la Batalla de Ideas, por nuestro Comandante en Jefe, en el año 2002, es una universidad de nuevo tipo, donde se vincula el estudio con la producción y la investigación. Esta universidad tiene como principal meta, formar ingenieros altamente calificados y comprometidos con nuestra Revolución y la creación de productos informáticos de calidad internacional. La misma cuenta con varias facultades entre las cuales se encuentra: La Facultad Regional “Mártires de Artemisa” (FRA).

La FRA se encuentra ubicada en la provincia Artemisa en el municipio de igual nombre y abrió sus puertas en el año 2007. Esta institución cuenta con un centro

de desarrollo de software el cual tiene como objetivo desarrollar productos informáticos a diferentes entidades del país. En este centro se ha firmado un contrato con el Ministerio de Comercio Exterior y la Inversión Extranjera (MINCEX) para desarrollar una solución de software que le permita mejorar la gestión de la información.

El MINCEX cuenta con una estructura de 23 direcciones, dentro de las que se encuentra la Dirección de Exportaciones y Sustitución de Importaciones (DESI). Esta tiene la misión de orientar y controlar la actividades tales como: exportación de bienes y servicios que desarrolla el país; así como la de sustitución de importaciones que contribuya a reducir el plan de compras en el exterior.

En esta dirección se genera principalmente información referente a las entidades que atiende, al cumplimiento de los planes de exportación y sustitución de importaciones y al programa de fomento de productos. Esta información es digital la cual es enviada por los Organismos de la Administración Central del Estado (OACE) y las Entidades atendidas, a los correspondientes especialistas de la DESI.

Cada uno de estos la conserva en sus archivos personales lo que provoca que no se disponga de una información precisa en un determinado momento por el alto grado de descentralización de la misma.

Por otra parte resulta imprescindible que solo el personal autorizado tenga acceso a dicha información por lo que resulta necesario garantizar la seguridad de la misma.

Debido a lo descrito anteriormente se plantea el siguiente **problema de investigación**: ¿Cómo garantizar mejores niveles de centralización y seguridad en el acceso y distribución de los datos en la gestión de los planes de exportaciones y sustitución de importaciones del MINCEX?

El **objeto de estudio** del presente trabajo está enmarcado en los procesos de gestión de la información de exportaciones y sustitución de importaciones, delimitando como **campo de acción** el acceso y distribución de la información en la gestión de los planes de exportaciones y sustitución de importaciones del MINCEX.

Se define como **objetivo general** del presente trabajo: Desarrollar una Base de Datos que garantice mayores niveles de centralización y seguridad en el acceso y distribución en la gestión de los planes de exportaciones y sustitución de importaciones del MINCEX.

Para el cumplimiento del objetivo general se determinaron los siguientes **objetivos específicos**:

- Elaborar la Fundamentación Teórica de la investigación.
- Diseñar la base de datos para la DESI del MINCEX.
- Desarrollar la capa de acceso a datos para la DESI del MINCEX.
- Validar los resultados obtenidos.

Para desarrollar este trabajo se ha planteado la siguiente **idea a defender**: Con el desarrollo de una BD se garantizará la centralización y seguridad en el acceso y distribución de la información de exportación y sustitución de importaciones del MINCEX.

Para darle cumplimiento a los objetivos trazados se desarrollaron las siguientes **tareas de la Investigación**:

- Establecimiento de los fundamentos teóricos y metodológicos para el desarrollo de la BD.
- Diseño de la BD para la DESI del MINCEX.
- Desarrollo de la capa de acceso a datos para la DESI del MINCEX.

- Validar la base de datos y la capa de acceso encargadas del acceso y distribución de la información en la DESI del MINCEX.

Con el objetivo de resolver el problema, dar cumplimiento a los objetivos y argumentar la idea a defender se hace uso de los siguientes métodos científicos, tanto teóricos como empíricos:

Métodos teóricos

- **Histórico-lógico:** Utilizado para investigar sobre los antecedentes de los procesos de gestión de la información, para una mejor comprensión de los mismos.
- **Analítico-sintético:** Se analizaron los aspectos más importantes relacionados con la administración de una base de datos y el desarrollo de capas de acceso a datos para posteriormente sintetizar y exponer los resultados obtenidos.
- **Modelación:** Para confeccionar el modelo de datos de la dirección de Exportación y Sustitución de Importaciones del MINCEX con el objetivo de lograr un mejor entendimiento de la posible solución a implementar.

Métodos empíricos

- **Análisis Documental:** Para revisar y hacer estudios de los documentos previos que contaba el proyecto, ejemplo el documento Levantamiento de la Información del MINCEX.

Se identificaron como **variable independiente** la BD para la DESI del MINCEX y como **variables dependientes** la centralización y seguridad en el acceso y distribución de la información.

Por lo descrito anteriormente se espera como **aporte práctico:**

- Base de datos para la DESI del MINCEX que garantice la centralización y seguridad en el acceso y distribución de la información.

El presente trabajo está estructurado en tres capítulos los cuales contienen la siguiente información:

Capítulo 1: Fundamentación teórica: En este capítulo se realiza una fundamentación teórica acerca del estado del arte de la investigación, teniendo en cuenta las bases de datos utilizadas a nivel mundial para las actividades de exportación y sustitución de importaciones, las tecnologías y herramientas a utilizar, así como la justificación de las mismas.

Capítulo 2: Diseño y Arquitectura 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. Además, se abordará sobre todo lo relacionado con el desarrollo de la capa de persistencia.

Capítulo 3: Implementación y validación de la solución propuesta: Validación teórica y funcional del diseño de las bases de datos e integración con la aplicación en la capa de persistencia.

Capítulo1: Fundamentación Teórica.

Introducción

En este capítulo se abordarán los conceptos asociados al dominio del problema así como las tecnologías y herramientas utilizadas con el objetivo de desarrollar la solución propuesta. Dichos conceptos, producto de los estudios realizados, contribuirán a un mejor entendimiento de la investigación y de los temas relacionados con el objeto de estudio planteado.

1.1 Conceptos asociados al dominio del problema.

1.1.1 Gestión

Este término proviene del latín gestio. La Real Academia Española define gestión como acción y efecto de gestionar. (Española, 2009)

Pablo Murray expresa que la gestión no es más que la herramienta que engloba fases de la administración siendo la gestión misma un elemento con desarrollo de la administración. Por otra parte es la actividad dirigida a obtener y asignar los recursos necesarios para el cumplimiento de los objetivos de una organización. (Murray, 2003)

En general, se puede decir que la gestión implica al conjunto de trámites que se llevan a cabo para resolver un asunto o concretar un proyecto.

1.1.2 Información

En un estudio realizado por Pablo Murray este dice que la información es la agrupación de datos organizados y presentados en un contexto. (Murray, 2003)

Para Sergio D´ Ambrosio la información es un conjunto de datos significativos y pertinentes que describen sucesos o entidades. (Heber, 2009)

El término información, según Verónica Sorberamurina, alude a un conjunto de datos organizados de manera tal que portan o arrojan un significado. (Heber, 2009)

Por tanto la información es un conjunto de datos asociados a un contexto, además de ser un elemento fundamental en el proceso de comunicación ya que va a suministrar o arrojar algún significado.

1.1.3 Gestión de la Información

La gestión de la información es el proceso de analizar y utilizar la información que se ha recabado y registrado para permitir a los administradores (de todos los niveles) tomar decisiones documentadas. (Bartle, 2009)

Gloria Donjuán Dante define gestión de la información al proceso mediante el cual se obtienen, despliegan o utilizan recursos básicos (económico, físicos, humanos, materiales) para manejar información dentro y para la sociedad a la que sirve. (Dante, 2010)

Gilberto Sotolongo expresa que la gestión de la información se ocupa de los resultados finales, no sólo de citas y localizaciones. Fernández-Molina lo corrobora cuando afirma que los profesionales de la información y sus instituciones son un factor indispensable para la permanencia de la organización en el mercado. (Dante, 2010)

Por lo anteriormente planteado se puede llegar a la conclusión de que la gestión de la información se puede definir como el conjunto de actividades realizadas con el fin de controlar, almacenar y recuperar adecuadamente la información producida, recibida o retenida por cualquier organización en el desarrollo de sus actividades.

1.1.4 Exportación

Una exportación es cualquier bien o servicio enviado a otra parte del mundo, con propósitos comerciales. La exportación es el tráfico legítimo de bienes y/o servicios nacionales de un país pretendidos para su uso o consumo en el extranjero. Las exportaciones pueden ser cualquier producto enviado fuera de la frontera de un Estado. Estas son generalmente llevadas a cabo bajo condiciones específicas. (Daniels 2004)

La exportación consiste en la salida de mercancías, capitales y servicios con destino al mercado exterior. La exportación, como la importación, se halla condicionada para el desarrollo de la producción mercantil y por la división internacional del trabajo. (Boríssov 2004).

1.1.5 Sustitución de Importaciones

La Sustitución de Importaciones es el proceso mediante el cual aquellos bienes que antes importaba el país, pasan a ser producidos internamente.

En un sentido más amplio, Sustitución de Importaciones también puede involucrar aquellas medidas de Política Económica que favorecen el proceso de Sustitución de Importaciones, tales como establecer altas tarifas a la importación de Bienes, depósitos previos de importación, y en general cualquier barrera artificial a la internación de bienes extranjeros al país. No obstante, la sustitución de Importaciones puede ser consecuencia de la evolución natural del mercado, el que responde a las disminuciones de los costos de la producción nacional, o a las mejoras de calidad de los productos nacionales. (Boríssov 2004).

1.1.6 Seguridad de la información.

Se entiende por seguridad de la información a todas aquellas medidas preventivas y reactivas del hombre, de las organizaciones y de los sistemas tecnológicos que permitan resguardar y proteger la información buscando mantener la confidencialidad, la disponibilidad e Integridad de la misma. El concepto de seguridad de la información no debe ser confundido con el de seguridad informática, ya que este último sólo se encarga de la seguridad en el medio informático, pudiendo encontrar información en diferentes medios o formas. (Gómez 2007)

1.1.7 Centralización de la información.

La centralización de la información consiste en la técnica utilizada para recopilar toda la información necesaria en un centro común. El principal objetivo de la centralización está dado por la necesidad de acceder a una información precisa en un determinado momento.

1.2 Bases de Datos

Muchos autores, han planteado diferentes definiciones, sin embargo, todos siguen la misma idea y ejemplo de estos son los siguientes conceptos:

Una base de datos es un conjunto de datos persistentes que es utilizado por los sistemas de aplicación de alguna empresa dada. (Date, 2003).

Conjunto de datos interrelacionados entre sí, almacenados con carácter más o menos permanente en la computadora. O sea, que puede considerarse una colección de datos variables en el tiempo. (Mato, 2005).

Una base de datos es una entidad en la cual se pueden almacenar datos de manera estructurada, con la menor redundancia posible, es la recopilación de datos en forma ordenada y concreta, donde diferentes programas y diferentes usuarios deben poder utilizar estos datos. (Cali, 2006)

Es un conjunto de datos almacenados entre los que existen relaciones lógicas y que han sido diseñadas para satisfacer los requerimientos de información de una empresa u organización. (Codd, 2011)

Por lo tanto, una base de datos o un almacén de datos no son más que un conjunto organizado de información estructurada sobre la que haya existido un análisis documental y que disponga de un sistema de búsqueda específico.

1.2.1 Elementos de una base de datos

- Datos: Es la parte esencial de la información, es decir, la información que llega a la base de datos.

- Atributos: Son los diferentes campos que conforman la estructura de una base de datos.
- Campos: Es la unidad más pequeña de datos.
- Registro: Es un conjunto de campos o atributos relacionados entre sí.
- Archivo: Es un conjunto de registros relacionados.

1.2.2 Tipos de base de datos

- Las BD se clasifican según la variabilidad de los datos almacenados:
- Bases de datos estáticas: Son bases de datos 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. (Barajas, 2009)
- Bases de datos dinámicas: Estas son bases de datos donde la información almacenada se modifica con el tiempo, permitiendo operaciones como actualización, borrado y adición de datos, además de las operaciones fundamentales de consulta. (Barajas, 2009)

1.2.3 Modelos de bases de datos

Además de la clasificación por la función de las bases de datos, estas también se pueden clasificar de acuerdo a su modelo de administración de datos.

Algunos modelos que con frecuencia son utilizados en el diseño de las bases de datos son:

- **Base de datos jerárquica:** Es aquella donde los datos se presentan en nivel múltiples que se representan con raíz y sus ramificaciones. Las bases de datos jerárquicas 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. Una de las principales limitaciones de este modelo es su incapacidad de representar eficientemente la redundancia de datos.

- **Bases de datos de red:** Es aquella que permite la conexión de los nodos en forma multidireccional, por lo que cada nodo puede tener varios dueños a la vez. Fue una gran mejora con respecto al modelo jerárquico, ya que ofrece una solución eficiente al problema de redundancia de datos; pero, aun así, la dificultad que significa administrar la información en una base de datos de red ha significado que sea un modelo utilizado en su mayoría por programadores más que por usuarios finales. (García, 2005)
- **Bases de datos relacionales:** Este es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. Cumple con el modelo relacional, el cual es adecuado para implementar bases de datos ya planificadas. Permiten establecer interconexiones entre los datos que están guardados en tablas, y a través de dichas conexiones relacionar los datos de ambas tablas. En una base de datos relacional, las filas representan registros y las columnas representan campos.

Las características fundamentales de estas BD es que se componen de varias tablas o relaciones, la forma que se almacena los datos no tiene relevancia, no pueden existir dos tablas con el mismo nombre ni registro, la relación entre una tabla padre y un hijo se lleva a cabo por medio de las claves primarias y foráneas, las claves primarias son la clave principal de un registro dentro de una tabla y las claves foráneas que se colocan en la tabla hija, contienen el mismo valor que la clave primaria del registro padre; por medio de estas se hacen las relaciones.

Las ventajas principales de las BD relacionales es que proveen herramientas que garantizan evitar la duplicidad de registros, garantizan la integridad referencial, así, al eliminar un registro elimina todos los registros relacionados dependientes y favorece la normalización por ser más comprensible y aplicable.

- **Bases de datos orientadas a objetos:** Este modelo, bastante reciente, y propio de los modelos informáticos orientados a objetos, trata de almacenar en la BD los objetos completos estado y comportamiento.
- **Bases de Datos documentales**
Permiten la indexación a texto completo, y en líneas generales realizar búsquedas más potentes, permiten el acceso a los documentos en formato electrónico, se instalan en una red local y se consultan desde ordenadores conectados a la red. (Rodríguez, 2003)
- **Bases de Datos multidimensionales**
Son bases de datos ideadas para desarrollar aplicaciones muy concretas, como creación de Cubos OLAP. Básicamente no se diferencian demasiado de las bases de datos relacionales (una tabla en una base de datos relacional podría serlo también en una base de datos multidimensional), la diferencia está más bien a nivel conceptual; en las bases de datos multidimensionales los campos o atributos de una tabla pueden ser de dos tipos, o bien representan dimensiones de la tabla, o bien representan métricas que se desean estudiar. (Riccuti, 1994).

Selección del tipo de Base de Datos

Después de realizar un estudio sobre los tipos de base de datos existentes se selecciona como modelo de base de datos a utilizar el relacional dadas las disímiles ventajas que esta brinda, además de una fácil integración con la herramienta de programación a utilizar en la solución propuesta.

1.2.4 Capa de acceso a datos

La capa de acceso a datos es la encargada de interactuar directamente con la BD, proporcionándole, a la capa de negocio, las funcionalidades necesarias de almacenamiento y recuperación de datos, aunque para esto es necesario, que tengan procedimientos almacenados y funciones dentro de la capa. Normalmente,

contiene clases que implementan los métodos para tener acceso a los datos de la base de datos subyacente. (Luque, 2005)

1.3 Soluciones Existentes.

- **Base de datos en línea para la UNCTAD** (Conferencia de las Naciones Unidas para el Comercio y el Desarrollo)

- **LEGISCOMEX**

Base de datos que incluye el registro de exportaciones e importaciones de Latinoamérica a su máximo nivel de detalle desde el año 2001.

- **Global Trade Atlas®**

El Global Trade Atlas®, de Global Trade Information Services, es una base de datos en línea que le ofrece una perspectiva única para ver las estadísticas del comercio mundial. Les permite a los usuarios ver el flujo de comercio en el mundo para los productos de su interés, utilizando información actualizada de importaciones/exportaciones de las fuentes oficiales de más de 70 países que la reportan.

Los usuarios pueden ver las importaciones o exportaciones de múltiples países en una sola pantalla, o pueden enfocarse en el comercio de un país en específico. El Global Trade Atlas® reporta el comercio mundial utilizando los códigos arancelarios a dos, cuatro, seis y ocho dígitos. El usuario podrá ver las importaciones y exportaciones usando valor, volumen y el precio unitario; así como también porcentaje de participación en el mercado y porcentaje de cambio por producto o país.

- **Base de datos Bacedel**

BADECEL es una base de datos estadísticos sobre comercio exterior, generado y mantenido por más de dos décadas por la División de Estadística y Proyecciones Económicas de la CEPAL. Su propósito es apoyar los análisis y estudios económicos que se efectúan, en particular, en el ámbito de la CEPAL y en las

esferas públicas y privadas en general, y facilitar el intercambio oportuno de información. Nace de un proyecto conjunto entre la CEPAL y el Banco Interamericano de Desarrollo (BID), con la colaboración de la Asociación Latinoamericana de Libre Comercio (ALALC), a fin de armonizar las clasificaciones nacionales de los países latinoamericanos en relación a las clasificaciones estadísticas internacionales utilizadas en esta área.

- Base de Datos OMC (Organización Mundial del Comercio)

Esta base de datos contiene amplia información sobre los aranceles de la nación más favorecida (NMF) aplicados y consolidados a nivel de las subpartidas normalizadas del Sistema Armonizado (SA) para todos los Miembros de la OMC. Cuando estén disponibles, también se proporcionan los datos sobre los regímenes arancelarios no NMF que un país concede a sus socios. La información procede de las comunicaciones destinadas a la Base Integrada de Datos (BID) de la OMC, en el caso de los aranceles aplicados y las importaciones, y de la Base de Datos de Listas Arancelarias Refundidas (Base de Datos LAR) en el caso de los derechos consolidados de todos los Miembros de la OMC.

Las bases de datos mencionadas anteriormente cuentan con numerosas ventajas en cuanto a procesos de gestión de información se refiere aunque las mismas cuentan con una desventaja particular la cual consiste en que fueron diseñadas con herramientas privadas lo que deviene un gran gasto su uso además de ir en contra de la política del centro la cual define el uso de software y herramientas libres para el desarrollo de aplicaciones informáticas. Por las razones explicadas anteriormente se decide el diseño de la base de datos para la gestión de la información de la Dirección de Exportación y Sustitución de Importaciones del MINCEX.

1.4 Metodología para el diseño de bases de datos relacionales

Una metodología es "un conjunto de modelos, lenguajes y otras herramientas que facilitan la representación de los datos en cada fase del proceso de diseño de una BD, junto con las reglas que permiten el paso de una fase a la siguiente", el análisis de todos estos elementos es fundamental para poder comprender y aplicar correctamente una metodología de diseño. (Mato, 2005)

El propósito de la metodología es ayudar al diseñador de la BD en la planificación, gestión, control y evaluación de los proyectos, mediante una serie de pasos:

1. Determinación de entidades y atributos.
2. Normalización de entidades.
3. Determinación de relaciones.
4. Obtención del modelo lógico global de los datos.
5. Diseño físico de la BD.

Cuando se realiza el diseño de la BD para un sistema determinado, es necesario establecer los datos a tomar en cuenta y las dependencias funcionales existentes entre ellos. Esto se obtiene luego de realizada la etapa de análisis del sistema y partiendo de lo obtenido en esta.

1.5 Herramientas y tecnologías

En Cuba en los últimos años se ha llevado a cabo el proceso de migración hacia el software libre. Esto trae consigo innumerables ventajas en el campo de la producción de productos informáticos ya que reduce considerablemente el costo que constituyen las aplicaciones privativas, entre otras.

- Herramientas CASE a utilizar

Se decide utilizar Power Architect y Visual Paradigm UML ya que estas herramientas brindan la posibilidad de realizar un excelente modelado lo cual permite lograr un mejor entendimiento del problema en cuestión, además, aumentan la calidad del proyecto reduciendo el tiempo y costo del mismo y

automatizan la documentación del proyecto ayudando a la reutilización del software, portabilidad y estandarización de la información.

- Sistema Gestor de Base de Datos a utilizar

Luego de realizar un estudio de los diferentes SGBD, se evidencia que PostgreSQL es el indicado para el trabajo con la BD del Sistema Informativo del MINCEX. Esta elección se basa en que SQL Server no es multiplataforma y al igual que Oracle, para su uso es necesario pagar importes por su licencia, lo cual no cumple con las políticas de la Universidad de migrar al software libre. Además PostgreSQL es capaz de ajustarse al número de computadoras y a la cantidad de memoria que posee el sistema de forma óptima, por lo que está apto para soportar mayor cantidad de peticiones simultáneas de manera correcta. PostgreSQL implementa el uso de subconsultas y transacciones, lo que hace su funcionamiento mucho más eficaz y ofrece soluciones en campos. El uso de PostgreSQL es cada vez mayor en las empresas que buscan un servidor de base de datos altamente sofisticado, con alto rendimiento, estable y capacitado para lidiar con grandes volúmenes de datos. El hecho de ser un producto de código abierto, sin costos de licencia, convierte al PostgreSQL en una alternativa extremadamente adecuada para las empresas que buscan un ahorro significativo.

- Herramienta de programación a utilizar

La herramienta de programación seleccionada fue el NetBeans. Esta elección se basa en que es un entorno integrado para el desarrollo de aplicaciones web. La plataforma NetBeans es una base modular y extensible usada como una estructura de integración para crear aplicaciones. Proporciona extensiones adicionales que se integran fácilmente en la plataforma y que pueden también utilizarse para desarrollar sus propias herramientas y soluciones. La plataforma ofrece servicios comunes a las aplicaciones de escritorio, permitiéndole al desarrollador enfocarse en la lógica específica de su aplicación.

- Aplicación Gráfica utilizada para el SGBD.

PgAdmin III es una aplicación gráfica para gestionar el gestor de bases de datos PostgreSQL, siendo la más completa y popular con licencia Open Source. Está escrita en C++ usando la librería gráfica multiplataforma wxWidgets, lo que permite que se pueda usar en Linux, FreeBSD, Solaris, Mac OS X y Windows. Es capaz de gestionar versiones a partir de la PostgreSQL 7.3 ejecutándose en cualquier plataforma, así como versiones comerciales de PostgreSQL como Pervasive Postgres, EnterpriseDB, Mammoth Replicator y SRA PowerGres.

PgAdmin III está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas. La interfaz gráfica soporta todas las características de PostgreSQL y facilita enormemente la administración. La aplicación también incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor, un agente para lanzar scripts programados, soporte para el motor de replicación Slony-I y mucho más. La conexión al servidor puede hacerse mediante conexión TCP/IP o Unix Domain Sockets (en plataformas *nix), y puede encriptarse mediante SSL para mayor seguridad. (PgAdminIII, 2008)

- Framework a utilizar

En la solución propuesta se utilizara Hibernate + Spring por la ventajas que estos proporcionan ya que pueden emplearse en cualquier aplicación hecha en Java. Spring Framework provee amplio soporte para Hibernate. En particular, brinda implementaciones de DAOs (Objetos de Acceso a Datos o Data Access Objects) que ofrecen diversas utilidades para acceder a la sesión de Hibernate, esto simplifica mucho el trabajo con Hibernate.

Conclusiones Parciales.

A raíz de la anterior investigación, donde se aborda conceptualmente los principales aspectos del problema en cuestión, se obtuvo un mayor conocimiento del sector sobre el cual gira el problema a resolver el cual consiste en gestionar la información relacionada con la exportación y la sustitución de importaciones en el país. Se realizó un estudio sobre las soluciones existentes en el mundo, sus ventajas y las desventajas que conllevaron a no utilizarlas para resolver el problema en cuestión. Se expusieron las herramientas y tecnologías a utilizar para el desarrollo de la solución propuesta.

Capítulo 2: Diseño y Arquitectura de la solución propuesta

Introducción

En el presente capítulo, se abordará todo lo relacionado al diseño de la base de datos y al desarrollo de la capa de acceso a datos. Se hará énfasis en las técnicas, objetivos y pasos a seguir para el desarrollo de la capa de acceso.

2.1 Arquitectura de software para la DESI.

Con la informatización de los procesos en la DESI, se pretende lograr la centralización y seguridad de la información a través del módulo a desarrollar. Este módulo estará soportado por una arquitectura denominada n-capas, la estructura de las capas de dicha arquitectura se muestra en la figura1.

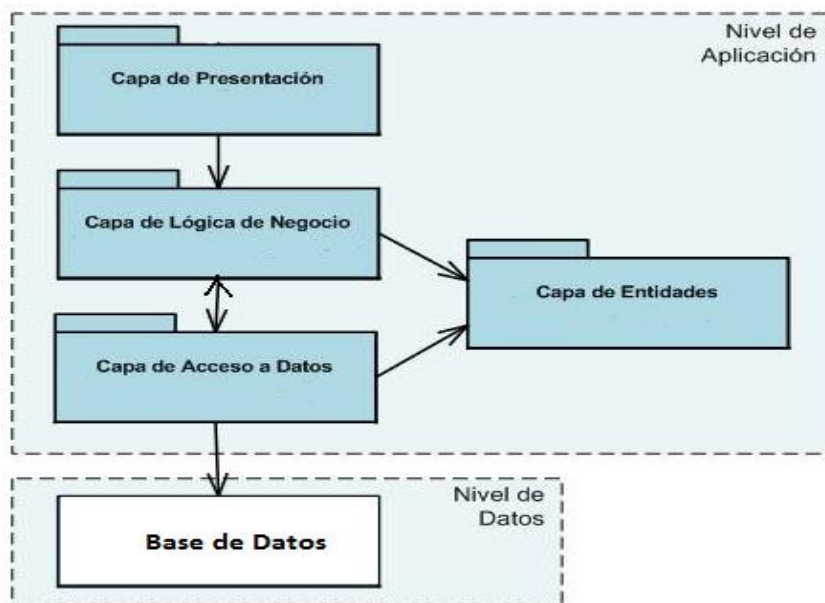


Figura 1. Arquitectura de software para la DESI.

- Capa de presentación: Es la que ve el usuario (también se la denomina "capa de usuario"), presenta el sistema al usuario, le comunica la información y captura la

información del usuario en un mínimo de proceso (realiza un filtrado previo para comprobar que no hay errores de formato). Esta capa se comunica únicamente con la capa de negocio.

- Capa de negocio: Es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de acceso a datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él.

- Capa de acceso a datos: Es donde residen los datos y es la encargada de acceder a los mismos. A través del sistema gestores de bases de datos se realiza toda la gestión de los datos, en esta capa se reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio. La capa de presentación no trabaja directamente con datos. En su lugar, invoca clases y métodos en la capa de acceso a datos para todas las solicitudes de datos.

La solución propuesta estará centrada en la capa de acceso a datos, la cual contendrá las clases en las que se implementarán los métodos necesarios para la gestión de la información y algunas consultas específicas del módulo cliente-servidor, en la capa de acceso a datos no estará implementado ningún método complejo en ese caso los mismos pasarían a ser implementados en la capa lógica del servidor.

2.2 Descripción de la arquitectura de la base de datos.

La arquitectura de la base de datos estará estructurada en tres niveles: lógico, conceptual y físico.

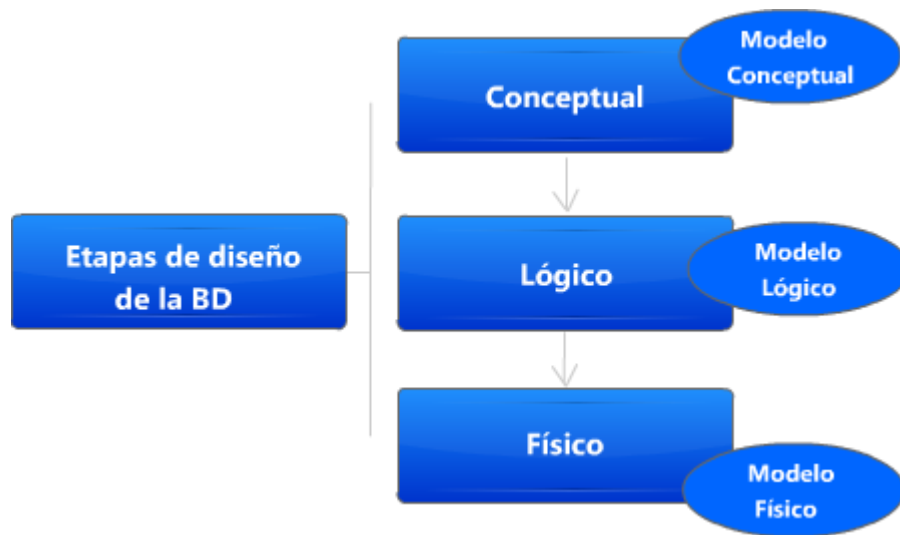


Figura 2. Arquitectura de la Base de Datos.

Diseño Conceptual: El objetivo principal es construir un modelo conceptual de los datos de acuerdo con los requisitos de datos de la organización, enteramente independiente de los detalles de implementación. Y que sirvan de base para las demás etapas de diseño.

El diseño conceptual quedó definido durante el levantamiento de información en la DESI donde se generó el documento Guía del Levantamiento de Información en el cual se recogió un listado previo de las necesidades del cliente. En este documento quedó definido conceptualmente el diseño a realizar ya que se identificaron las entidades que iban a conformar el diseño lógico con sus correspondientes atributos.

Diseño Lógico: Describe varios esquemas externos o vistas de usuario. Cada uno describe la parte de la base de datos que le interesa y lo oculta al resto de grupo. En este nivel se recibe un modelo conceptual o un modelo lógico para especificar los esquemas.

Capítulo2: Diseño y Arquitectura de la Solución Propuesta

Para realizar el diseño lógico de la base de datos se realizó el análisis del modelo de dominio establecido. En este se definieron las entidades persistentes a almacenar en la base de datos las cuales daban respuesta a los requisitos establecidos en el documento Guía del Levantamiento de la Información. A continuación se muestra el diseño lógico de la base de datos a desarrollar.

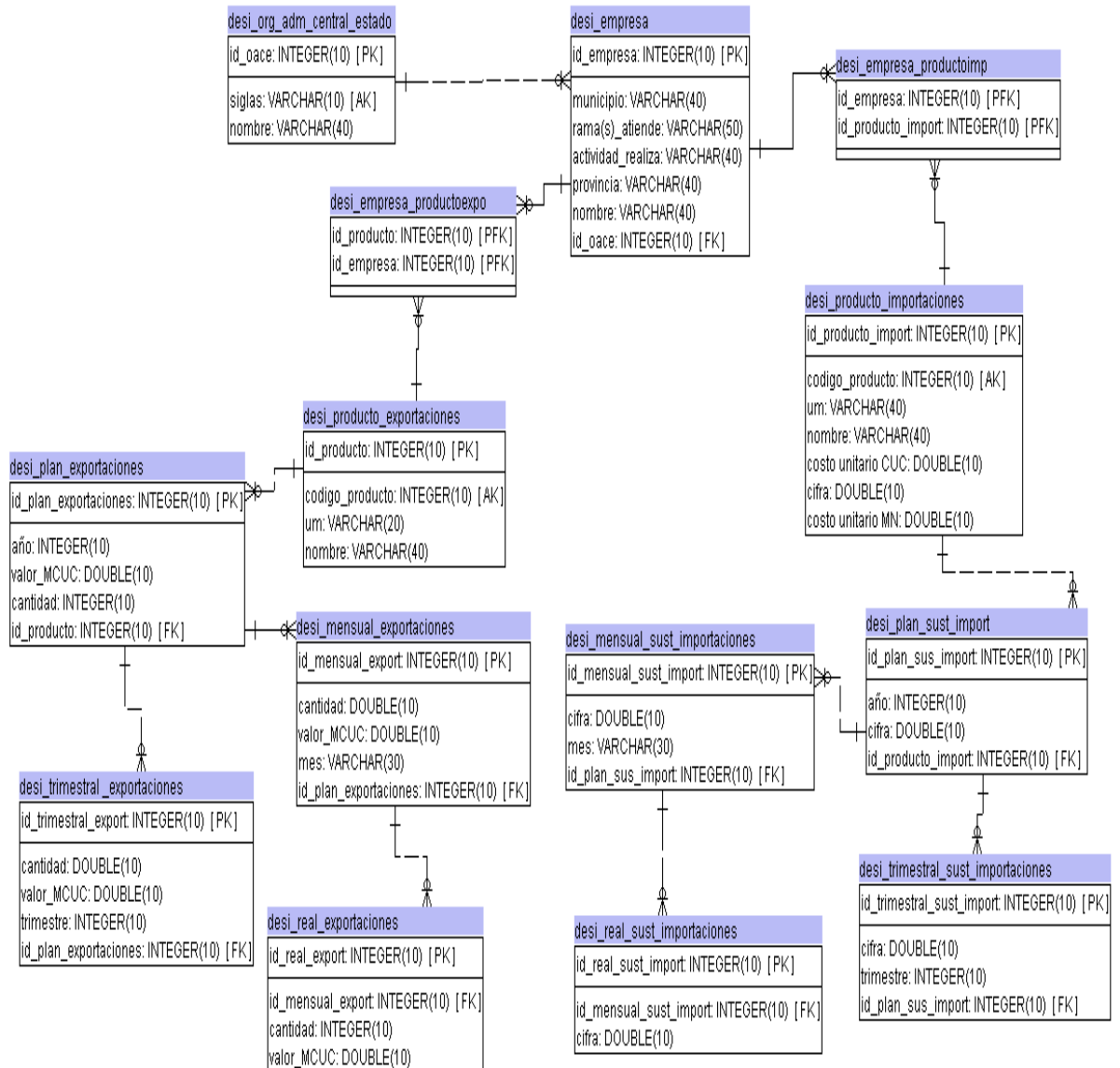


Figura 3. Modelo lógico de la Base de Datos.

Diseño Físico: Es la representación del nivel más bajo de abstracción, en éste se describe en detalle la forma en cómo se almacenan los datos en los dispositivos de almacenamiento (por ejemplo, mediante señaladores o índices para el acceso aleatorio a los datos).

A partir del diseño lógico realizado quedó conformado el diseño físico de la base de datos quedando convertidas las entidades del diseño lógico en las correspondientes tablas de la base de datos.

2.3 Arquitectura de la Capa de Acceso a Datos.

La capa de acceso a dato (CAD) estará estructurada por paquetes en los cuales se encontrarán las clases mapeadas a utilizar para la correcta gestión de la información. El objetivo del uso de los paquetes será lograr una mayor organización de las clases mapeadas.



Figura 4. Arquitectura de la Capa de Acceso a Datos.

En el paquete MODEL (Modelo) se encontrarán las entidades persistentes las cuales representan las clases java correspondientes a las entidades de la base de datos. Por su parte en el paquete DAO se encontrarán las clases GenericDao y

GenerarReporte las cuales contienen las funcionalidades que se encargarán de la gestión de la información en la base de datos.

2.4 Patrones de diseño a utilizar.

DAO: Un DAO es un componente de software capaz de suministrar una interfaz que actúa como puente de comunicación entre la aplicación y uno o más dispositivos de almacenamiento de datos, tales como una Base de datos o un archivo. El término se aplica frecuentemente al Patrón de diseño Object.

El objetivo de este patrón es abstraer y encapsular todos los accesos a la fuente de datos, ocultando completamente los detalles de la implementación de la fuente de datos a sus clientes y además la interfaz expuesta por DAO no cambia cuando cambia la implementación de la fuente de datos subyacente.

Este patrón se utilizó en la solución propuesta para la crear la clase GenericDao, en esta clase se encuentran implementadas las funcionalidades necesarias para la gestión de los datos en la base de datos.

Encapsulamiento:

Se trata de que el software cliente se centre en los datos que necesita y se olvide de cómo se realiza el acceso a los datos o de cuál es la fuente de almacenamiento. Este patrón surge de la necesidad de gestionar una diversidad de fuentes de datos, aunque su uso se extiende al problema de encapsular no solo la fuente de datos, sino además ocultar la forma de acceder a los datos.

2.5 Requisitos del sistema.

La definición de las necesidades del software es un proceso complicado, ya que hay que identificar todos los requisitos que el sistema debe cumplir para lograr satisfacer las necesidades de los usuarios finales. Además estos deben ser bien claros puesto que servirán de guía para el desarrollo del sistema.

2.5.1 Requisitos funcionales.

Los requisitos funcionales (RF) son capacidades o condiciones que el sistema debe cumplir, son peticiones a las que el sistema debe ser capaz de dar respuesta. A continuación se muestran los requisitos más importantes, para el estudio del resto de los requisitos identificados remitirse al **Anexo # 1: Lista de Reserva del Producto**.

RF 1: Insertar Organismo de la Administración Central del Estado: El sistema debe permitir que se inserte un nuevo Organismo de la Administración Central del Estado.

RF 2: Buscar Organismo de la Administración Central del Estado: El sistema debe permitir buscar el Organismo de la Administración Central del Estado que se necesite.

RF 3: Modificar Organismo de la Administración Central del Estado: El sistema debe permitir que se le realicen cambios al Organismo de la Administración Central del Estado seleccionado.

RF 4: Eliminar Organismo de la Administración Central del Estado: El sistema debe permitir que se eliminen los Organismos de la Administración Central del Estado.

RF 5: Insertar Empresa: El sistema debe permitir que se inserte una nueva Empresa.

RF 6: Buscar Empresa: El sistema debe permitir buscar la Empresa que se necesite.

RF 7: Modificar Empresa: El sistema debe permitir que se le realicen cambios a la Empresa seleccionada.

RF 8: Eliminar Empresa: El sistema debe permitir que se eliminen las Empresa.

RF 9: Insertar Producto de Exportación: El sistema debe permitir que se inserte un nuevo Producto de Exportación.

RF 10: Buscar Producto de Exportación: El sistema debe permitir buscar el Producto de Exportación que se necesite.

RF 11: Modificar Producto de Exportación: El sistema debe permitir que se le realicen cambios al Producto de Exportación seleccionado.

RF 12: Eliminar Producto de Exportación: El sistema debe permitir que se eliminen los Productos de Exportación.

2.5.2 Requisitos no funcionales.

Los requisitos no funcionales son restricciones de los servicios o funciones ofrecidas por el sistema. Entre estas funciones deben encontrarse que el sistema sea agradable a la vista del usuario, debe ser confiable y rápido.

RNF 1:

- El tiempo de respuesta máximo a las peticiones por parte de los usuarios en las consultas más complejas que actualmente se utilizan, no debe excederse de 10 segundos.
- El sistema no debe tener más de 1 hora de no disponibilidad en el mes.

RNF 2:

- La información almacenada o manejada desde la base de datos debe estar protegida de acceso no autorizado y divulgación.
- La base de datos será independiente de la aplicación.
- A los usuarios autorizados se les garantizarán el acceso a la información, los dispositivos o mecanismos utilizados para lograr la seguridad, no ocultarán o retrasarán a los usuarios para obtener los datos deseados en un momento dado.

RNF 3:

- El soporte y/o mantenimiento de la base de datos no debe necesariamente detener el servicio.

RNF 4:

- La base de datos garantizará su correcto funcionamiento en los sistemas operativos GNU/Linux y Windows, por las características que presenta el SGBD utilizado.

RNF 5:

- El servidor de bases de datos debe ser PostgreSQL versión 9.0.
- La aplicación gráfica de desarrollo de PostgreSQL deberá ser pgAdmin III, versión 1.12 o superior.

RNF 6:

- El servidor de base de datos debe contener un procesador Pentium 3 (o superior).
- Debe tener como mínimo 1GB de Memoria RAM.
- Debe contener al menos 1 disco duro de 160 GB.

2.6 Descripción de las entidades del Diagrama Entidad-Relación.

A continuación se muestra la descripción de las tablas:

- Organismo
- Producto de Exportaciones
- Producto de Sustitución de Importaciones.

Para el estudio de la descripción de las restantes tablas de la base de datos remitirse al **Anexo # 2** Descripción de las tablas de la base de datos.

Entidad:	Organismo		
Descripción:	Representa a cada Organismo de la Administración Central del Estado que atiende la Dirección de Exportaciones y Sustitución de Importaciones del MINCEX.		
Relaciones:	Empresa		
Campos	Tipo de Dato	Tamaño	Descripción
Id_oace	Entero	10	Identifica la Entidad
Nombre	Cadena de Caracteres	40	Nombre del Organismo
Siglas	Cadena de Caracteres	10	Siglas que identifican al organismo

Tabla 1. Descripción de la tabla: Organismo.

Entidad:	Producto de Exportaciones		
Descripción:	Representa a cada producto de exportaciones, perteneciente a una o más empresas.		
Relaciones:	Empresa, Plan de exportación Anual, Plan de exportación Trimestral, Plan de exportación Mensual.		
Campos	Tipo de Dato	Tamaño	Descripción
Id	Entero	10	Identifica la Entidad
Nombre	Cadena de Caracteres	10	Nombre del Producto
Unidad de Medida	Cadena de Caracteres	40	Unidad de Medida del Producto
Código del producto	Entero	10	Código único para cada producto.

Tabla 2. Descripción de la tabla: Producto de Exportaciones.

Entidad:	Producto de Sustitución de Importaciones.		
Descripción:	Representa a cada producto de sustitución de importaciones, perteneciente a una o más empresas.		
Relaciones:	Producto, Planes.		
Campos	Tipo de Dato	Tamaño	Descripción
Id	Entero	10	Identifica la Entidad
Costo Unitario CUC	Decimal	10	Costo unitario en CUC del Producto
Cifra	Decimal	10	Cifra del Producto
Costo Unitario MN	Decimal	10	Costo unitario en MN del Producto
Nombre	Cadena de Caracteres	10	Nombre del Producto
Unidad de Medida	Cadena de Caracteres	40	Unidad de Medida del Producto
Código del producto	Entero	10	Código único para cada producto.

Tabla 4. Descripción de la tabla: Producto de Sustitución de Importaciones.

Conclusiones Parciales.

En el presente capítulo tomando como punto de partida la situación problemática existente en la Dirección de Exportación y Sustitución de Importaciones y las

Capítulo2: Diseño y Arquitectura de la Solución Propuesta

necesidades presentes en dicha dirección se logró realizar el diseño y la arquitectura de la BD cumpliéndose con los requerimientos del cliente, así como con las restricciones y condiciones presentes dentro del alcance definido. La arquitectura propuesta garantiza la seguridad y la centralización de la información.

Capítulo 3: Implementación y validación de la solución propuesta

Introducción

En este capítulo se realiza la validación funcional de la capa de acceso a datos diseñada. Primeramente se analizan varios aspectos tales como: la integridad, el análisis de la redundancia de la información y la seguridad de la base de datos.

3.1 Estructura de los archivos mediante paquetes.

Mantener los archivos almacenados en una estructura de paquetes es un factor clave en cuanto a organización se refiere, puesto que en la capa servidora los archivos por parte del servidor y el desarrollador de la base de datos se encuentran integrados, por tanto 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.

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 diseñada, el desarrollador de la base de datos cuenta con dos paquetes que constituyen sus diferentes líneas de implementación, donde almacenará todo el código que cree.

En el primer paquete **cu/uci/hab/mincex/common/models/entity/** almacena las entidades la cuales representan a cada una de las tablas creadas en la base de datos. Por ultimo **cu/uci/hab/mincex/common/models/dao/** es el paquete que contiene las entidades GenericDao y GenerarReporte en las cuales estarán implementadas las funcionalidades encargadas de la gestión de la información.

3.2 Proceso de mapeo.

Para el desarrollo de la base de datos se utilizó el modelo relacional. Esto conllevó el surgimiento de un grave problema el cual venía dado por contar con dos modelos diferentes, modelo relacional para la base de datos y modelo orientado a objetos para la parte del servidor. Para poder resolver estas diferencias se hizo necesario utilizar el patrón DAO, el cual a través del ORM (Object Relational Mapping) Hibernate se mapearon las entidades de la base de datos con el objetivo de convertirlas en clases java orientadas a objetos. A continuación se muestran las clases java después de realizado el proceso de mapeo.

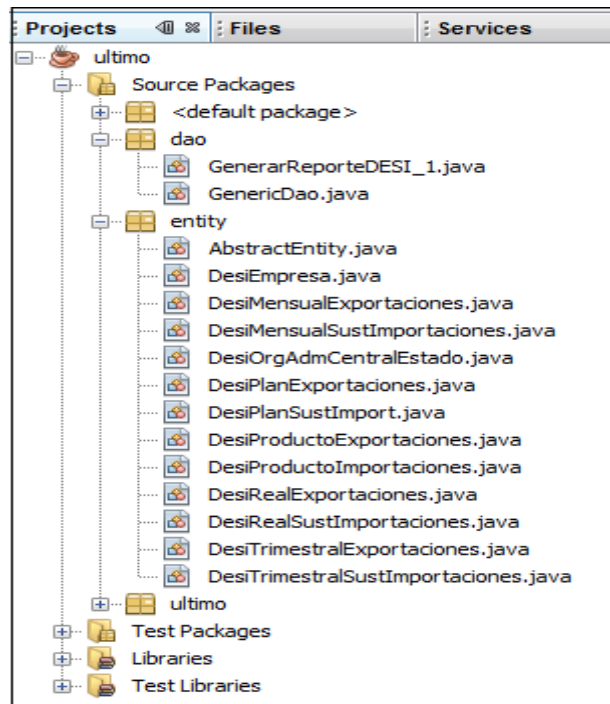


Figura 5. Clases java mapeadas de la base de datos.

3.3 Estándares de codificación.

En general, un estándar de codificación son reglas que se siguen para la escritura del código fuente. De tal manera que otros programadores se les facilite entender

la implementación realizada (como identificar las variables, las funciones o métodos).

Seguir un estándar de código permite que este pueda modificarse para añadirle nuevas características, modificar las ya existentes, depurar errores o mejorar el rendimiento por cualquier desarrollador. La legibilidad del código fuente repercute directamente en lo bien que un desarrollador comprende un sistema. Aunque la legibilidad y el mantenimiento son el resultado de muchos factores, un aspecto del desarrollo de software en la que todos los programadores influyen especialmente es en la técnica de codificación.

3.3.1 Estilos de codificación.

Nombres: Para la definición de los nombres de los identificadores haremos uso de los convenios establecidos por CamelCase excepto para los casos de las constantes, sus identificadores serán definidos completamente en mayúsculas.

Variables, Métodos: Para estos identificadores se hace uso de la variante lowerCamelCase, empiezan con minúsculas y si estos identificadores están compuestos por varias palabras las siguientes empezaran con mayúscula.

Clases, enumeradores, Interfaces: Para estos identificadores se hace uso de la variante UpperCamelCase Todas las palabras que componen a dichos identificadores empezaran con mayúscula.

Comentarios de Bloque: Los comentarios de bloque se usan para proporcionar descripciones de ficheros, métodos, estructuras de datos y algoritmos. Deben ser utilizados al comienzo de cada archivo y antes de cada método.

Comentario de una sola línea: Este tipo de comentario puede aparecer al mismo nivel del código al que se desea comentar. Un comentario de una sola línea debe ser precedido por una línea en blanco.

Comentario detrás del código: Los comentarios cortos pueden aparecer en la misma línea que el código que describen, pero deben estar suficientemente

separadas de las declaraciones. Si más de un comentario corto aparece en un fragmento de código, todos ellos deben tener la misma sangría.

Comentarios de las clases: Al inicio de cada clase se debe describir con un comentario de bloque el propósito de la clase e instrucciones de uso. También se pueden incluir recordatorios o avisos acerca de las mejoras necesarias o convenientes. Es necesario especificar el autor y la versión de la clase, utilizando anotaciones de Java.

Comentarios de los métodos: Utilizar las convenciones java para describir la finalidad, los parámetros, lo que devuelve y las excepciones a tratar.

3.4 Validación teórica.

El concepto de integridad de la base de datos (BD) hace referencia a la validez, corrección, consistencia, completitud y coherencia de los datos almacenados en la BD. Su principal función se basa en proteger la BD contra operaciones que introduzcan inconsistencia en los datos. Esta integridad se expresa normalmente a través de restricciones o reglas, las cuales la base de datos no puede quebrantar. El sistema gestor de base de datos es el encargado de mantener dichas reglas.

Cuando se realizan acciones tales como: insertar datos incorrectos o no válidos, modificar la información existente tomando un valor equivocado o eliminando datos que produzcan la violación de alguna restricción en la base de datos, se está ante una situación en la que la integridad de los datos se encuentra afectada.

3.4.1 Integridad de entidad.

La integridad de entidad define una fila como entidad única para una tabla determinada. La integridad de entidad exige la integridad de las columnas de los identificadores o la clave principal de una tabla, mediante índice y restricciones PRIMARY KEY.

Todas las tablas de la base de datos tienen definido una fila como identificador usando la restricción PRIMARY KEY. En la tabla `desi_producto_importaciones` se evidencia un ejemplo de lo anteriormente planteado.

desi_producto_importaciones		
+id_producto_import	int4	Nullable = false
#id_empresa	int4	Nullable = false
um	varchar(40)	Nullable = false
nombre	varchar(40)	Nullable = false
costo_unitario_cuc	float8	Nullable = false
cifra	float8	Nullable = false
costo_unitario_mn	float8	Nullable = false

Figura 6. Restricción PRIMARY KEY.

3.4.2 Integridad de dominio.

La integridad de dominio viene dada por la validez de las entradas para una columna determinada. Puede exigir la integridad de dominio para restringir el tipo mediante tipos de datos o el intervalo de valores posibles mediante restricciones FOREIGN KEY.

En las tablas de la base de datos diseñada se definieron los tipos de datos de los atributos correspondiendo al tipo de información a almacenar. Los tipos de datos definidos fueron int, varchar, float. Si el valor del atributo no puede ser vacío se establece como definición NOT NULL. Cuando existe una relación de uno a muchos entre dos entidades el atributo definido como PRIMARY KEY en la entidad con multiplicidad uno pasa a formar parte ese atributo como FOREIGN KEY en la entidad con multiplicidad mucho. La restricción FOREIGN KEY se evidencia también cuando existe multiplicidad de uno a uno donde la PRIMARY KEY de una de las tablas pasaría a formar FOREIGN KEY de la otra tabla.

En las tablas `desi_empresa`, `desi_producto_exportaciones`, `desi_producto_importaciones` se evidencia los tipos de datos definidos además de la restricción FOREIGN KEY.

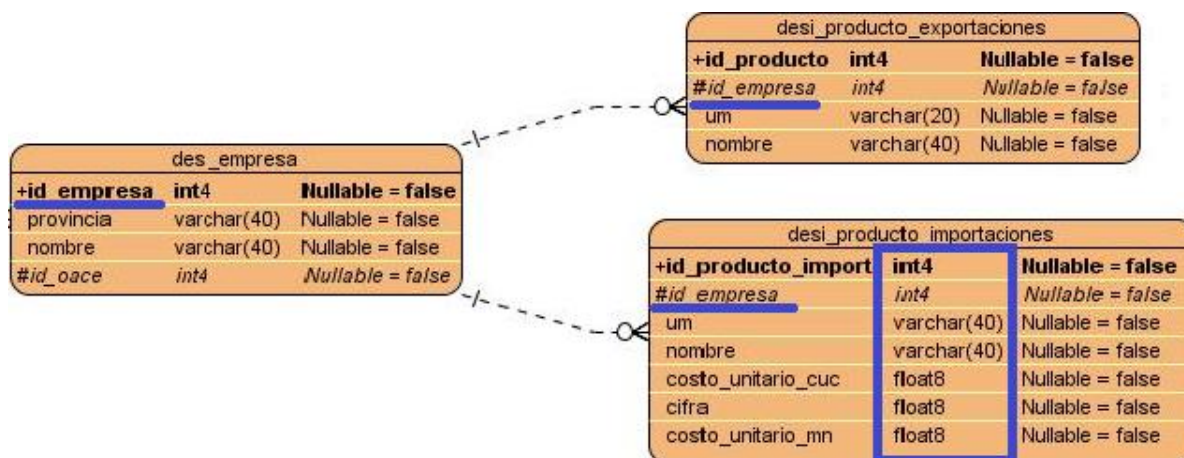


Figura 6. Restricción FOREIGN KEY.

3.4.3 Normalización.

La normalización consiste en aplicar un conjunto de reglas a las relaciones obtenidas tras el paso del modelo entidad-relación al modelo relacional, este proceso se realiza con el objetivo de evitar la redundancia de los datos, evitar problemas de actualización de los datos en las tablas y proteger la integridad de los datos.

A continuación se muestran ejemplos de transformaciones al realizar el análisis correspondiente a cada una de las formas normales por las que iba transitando la base de datos.

Primera Forma Normal: (1FN).

- Todos los atributos son atómicos. Un atributo es atómico si los elementos del dominio son indivisibles, mínimos.
- La tabla contiene una llave primaria única.
- La llave primaria no contiene atributos nulos.

Capítulo3: Implementación y Validación de la Solución Propuesta

- No debe existir variación en el número de columnas.
- Los campos no llave deben identificarse por la llave (Dependencia Funcional)
- Debe existir una independencia del orden tanto de las filas como de las columnas, es decir, si los datos cambian de orden no deben cambiar sus significados. (Mato 2005).

En la relación Empresa el atributo producto es un atributo multievaluado y esto viola los requerimientos para alcanzar la primera forma normal. Para solucionar esto debemos crear dos nuevas relaciones Empresa y Producto y establecer entre estas dos relaciones una multiplicidad de mucho a mucho generando una relación Empresa_Producto con el identificador de Empresa y de Producto como llave primaria de esta relación:

Empresa(id_empresa, rama_atiende, actividad_realiza, producto)

Empresa(id_empresa, rama_atiende, actividad_realiza

Producto_Exportacion(id_producto, nombre, código_producto, um).

Empresa_Producto(id_empresa, id_producto).

Segunda Forma Normal: (2FN).

Para que una base de datos se encuentre en 2FN debe encontrarse en 1FN, además debe cumplir con el concepto de dependencia funcional total.

Dependencia funcional total: Sea $X \rightarrow Y$. Si la eliminación de cualquier atributo A de X hace que la dependencia deje de ser válida se dice que esta dependencia es total o que Y depende totalmente de X. O sea $A \rightarrow X$, $(X - \{A\}) \rightarrow Y$, entonces la dependencia es total (los atributos no llaves dependen de todos los atributos que componen la llave y no de solo una parte de ellos). (Mato 2005)

En la relación Empresa se encuentra el identificador de Organismo de la Administración Central del Estado (OACE) que posibilita que existan dependencias

Capítulo3: Implementación y Validación de la Solución Propuesta

parciales por lo que una posible solución es crear dos nuevas relaciones Empresa (llave primaria identificador de empresa(id_empresa)) y OACE(llave primaria identificador de OACE(id_oace)).

Empresa(id_empresa, nombre, rama_atiende, actividad_realiza, municipio, provincia, id_oace, nombre, siglas).

Luego de aplicarse la regla de la 2FN las relaciones quedarían así.

Empresa(id_empresa, nombre, rama_atiende, actividad_realiza, municipio, provincia).

OACE(id_oace, nombre, siglas).

Tercera Forma Normal: (3FN).

Para que una base de datos se encuentre en 3FN esta debe encontrarse en 2FN y eliminar las dependencias transitivas.

Dependencia Transitiva: sea $X \rightarrow Y$ una dependencia funcional en un esquema de relación R, se dice que es transitiva si existe un conjunto de atributos Z que no sea un subconjunto de cualquier llave de R y se cumple tanto $X \rightarrow Z$ como $Z \rightarrow Y$ ($X \rightarrow Y$ donde Y no primo, X ni subconjunto ni superconjunto de alguna llave).

(Mato 2005).

En la relación Mensual_Sustitucion_Importaciones al igual que en la relación Trimestral_Sustitucion_Importaciones el dato año es común para ambas lo que evidencia la existencia de dependencias funcionales transitivas, por lo que la solución es eliminar el dato de las relaciones anteriormente mencionadas y crear la relación Plan_Sustitucion_Importaciones la cual representara al plan anual agregándole el dato año.

Capítulo3: Implementación y Validación de la Solución Propuesta

Trimestral_Sustitucion_Importaciones (id trimestral sust import, año, cifra, trimestre).

Mensuall_Sustitucion_Importaciones(id mensual sust import, año, cifra, mes).

Luego de aplicarse la regla de la 3FN las relaciones quedarían así.

Trimestral_Sustitucion_Importaciones (id trimestral sust import, cifra, trimestre, id_plan_sus_import).

Mensuall_Sustitucion_Importaciones(id mensual sust import, cifra, mes, id_plan_sus_import).

Plan_Sustitucion_Importaciones(id plan sus import, año, cifra, id_producto_import).

Una vez realizado todo el proceso de normalización la base de datos resultante para la realización del Módulo de la DESI se encuentra en 3FN.

3.5 Creación de las clases AbstractEntity y GenericDao.

Una vez mapeadas las clases correspondientes a las entidades de la base de datos se creó la clase AbstractEntity de la cual heredan las clases que fueron mapeadas. Esta clase implementa el método abstracto getId() con el objetivo de auto incrementar el Id (identificador) en cada una de las clases hijas. Para la validez de este proceso fue necesario redefinir el método getId() en cada una de las clases mapeadas.

```
package entity;

import java.io.Serializable;

@author postgres

public abstract class AbstractEntity implements Serializable {

    private static final long serialVersionUID = 5143168877262662951L;

    protected Long id;

    // Este método es abstracto porque las anotaciones no son heredables

    public abstract Long getId();

    // Este método es protegido para evitar que un programador pueda poner un
    // Identificador en la instancia, ya que los identificadores deben ser
    // gestionados por la capa de persistencia

    protected void setId(final Long id) {

        this.id=id;    }

}
```

Fig 7. Clase AbstractEntity

A continuación se muestra una clase la cual hereda de la clase AbstractEntity en la cual se redefinió el método getId().

```
public class DesiOrgAdmCentralEstado extends AbstractEntity implements java.io.Serializable {

    private String nombre;

    private String siglas;

    public DesiOrgAdmCentralEstado () {

    }

    public DesiOrgAdmCentralEstado(String nombre, String siglas) {

        this.nombre = nombre;

        this.siglas = siglas;  }

    @Id

    @SequenceGenerator(name = "id_oace",
sequenceName="dir_esi.desi_org_adm_central_estado_id_oace_seq")

    @GeneratedValue(generator = "id_oace")

    @Column (name="id_oace", unique=true, nullable=false)

    @Override

    public Long getId() {

        return this.id;  }

}
```

Fig 8. Clase Organismo de la Administración Central del Estado.

Una vez se haya redefinido el método `getId()` en las clases mapeadas y estas se encuentren heredando de la clase `AbstractEntity` se procederá a la creación de la clase `GenericDao` en la cual se implementarán las funciones necesarias para la persistencia de los datos.

```
public class GenericDao extends HibernateDaoSupport {  
  
    public GenericDao() {}  
  
    public void saveOrUpdate(AbstractEntity entity) {}  
  
    public void save(AbstractEntity entity) {}  
  
    public void update(AbstractEntity entity) {}  
  
    public <T> List<T> find(Class<T> entityClass) {}  
  
    public AbstractEntity findByExample(AbstractEntity exampleEntity) {}  
  
    public <T> AbstractEntity findByPK(Class<T> entityName, long id) {}  
  
    public void delete(AbstractEntity entity) {}  
  
    public <T> void deleteAll(Collection<T> collection) {}  
  
    public <T> List<T> findByParamAndValue(Class<T> entityClass, String param, Object value) {}  
  
    public <T> List<T> findByParamsAndValues(Class<T> entityClass, String[] params, Object[] values) {}  
  
}
```

Fig 9. Clase GenericDao

Funcionalidades definidas en la clase GenericDao.

1. void saveOrUpdate(AbstractEntity entity)

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 y lo modifica si ya existe.

2. void save(AbstractEntity entity)

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

3. void update(AbstractEntity entity)

Recibe como parámetro un AbstractEntity lo busca en la base de datos por la

clave primaria y si está lo modifica.

4. `<T> List<T> find(Class<T> entityClass)`

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.

5. `<T> AbstractEntity findByPK(Class<T> entityName, long id)`

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.

6. `AbstractEntity findByExample(AbstractEntity exampleEntity)`

Devuelve una entidad semejante a la pasada por parámetros.

7. `void delete(AbstractEntity entity)`

Elimina un solo objeto de la base de datos pasado por parámetros

8. `<T> void deleteAll(Collection<T> collection)`

Elimina la lista de entidades pasada por parámetros.

9. `<T> List<T> findByParamAndValue(Class<T> entityClass, String param, Object value)`

Devuelve una lista de la entidad pasada por parámetros donde el valor de value coincida con el campo en la base de datos establecido en la variable param.

10. `<T> List<T> findByParamsAndValues(Class<T> entityClass, String[] params, Object[] values)`

Devuelve una lista de la entidad pasada por parámetros donde todos los valores pasados en values coincidan con los campos en la base de datos establecidos en el arreglo de valores params.

3.5.1 Creación del DAO Auxiliar “GenerarReporte”

Con el objetivo de realizar una serie de consultas específicas a la base de datos, las cuales no pudieron ser ejecutadas con los métodos definidos en la clase GenericDao, se procede a la creación de una clase DAO auxiliar “GenerarReporte” el cual contará con funcionalidades específicas para la realización de dichas consultas.

```
public class GenerarReporteDESI extends HibernateDaoSupport {  
    public GenerarReporteDESI () {}  
    public <T> List<T> FindByHQL(String hql) {}  
    public List<DesiPlanSustImport> SeeksImportPlanByYear(int year) {}  
    public double ThirdValueOfTheReport(int currentYear, int previousYear) {}  
    public List<DesiMensualSustImportaciones> SeeksImportPlanByYearAndMonth(int year, String month) {}  
    public List<DesiRealSustImportaciones> SeeksRealImportPlanByMonth(String month) {}  
    public double SeventhValueOfTheReport(int currentYear, int previousYear, String month) {}  
    public double EighthValueOfTheReport(int currentYear, int previousYear, String month) {}  
    public double NinthValueOfTheReport(int currentYear, int previousYear, String month) {}  
    public List<DesiRealExportaciones> SeeksRealExportPlanByYearAndMonth(int year, String month) {}  
    public List<DesiPlanExportaciones> SeeksExportPlanByYear(int currentYear) {}  
    public double PercentValueOfTheFirstTableExport(int currentYear, int previousYear, String month) {}  
    public double PercentValueOfTheSecondTableExport(int year, String month) {}  
    public double PercentValueOfTheThirdTableExport(int year, String month) {}  
}
```

Fig 10. Clase GenerarReporte

Funcionalidades definidas en la clase GenerarReporte.

1- <T> List<T> findByHQL(String hql)

Realiza una búsqueda utilizando Hibernate Query Language (HQL).

2- List<DesiPlanSustImport> SeeksImportPlanByYear(int year)

Devuelve una lista de DesiPlanSustImport dado un año.

3- double ThirdValueOfTheReport(int currentYear, int previousYear)

Devuelve el tercer valor del reporte de sustitución de importaciones el cual se obtiene de restar la cifra del plan del año actual con la del plan del año anterior.

4-List<DesiMensualSustImportaciones> SeeksImportPlanByYearAndMonth(int year, String month)

Devuelve una lista de DesiMensualSustImport dado el año y el mes.

5- List<DesiRealSustImportaciones> SeeksReallImportPlanByMonth(String month)

Devuelve una lista de DesiRealSustImport dado el mes.

6- double SeventhValueOfTheReport(int currentYear, int previousYear, String month)

Devuelve el séptimo valor del reporte de sustitución de importaciones el cual se obtiene de restar la cifra del plan del año actual con el real obtenido el año anterior.

7- double EighthValueOfTheReport(int currentYear, int previousYear, String month)

Devuelve el octavo valor del reporte de sustitución de importaciones el cual se obtiene de restar la cifra del real del año actual con el real obtenido el año anterior.

8- double NinthValueOfTheReport(int currentYear, int previousYear, String month)

Devuelve el noveno valor del reporte de sustitución de importaciones el cual se obtiene calculando el promedio con el séptimo y el octavo valor del reporte de sustitución de importaciones.

9- List<DesiRealExportaciones> SeeksRealExportPlanByYearAndMonth(int year, String month)

Devuelve una lista de DesiRealExportaciones dado el año y el mes.

10- List<DesiPlanExportaciones> SeeksExportPlanByYear(int currentYear)

Devuelve una lista de DesiPlanExportaciones dado el año.

11- double PercentValueOfTheFirstTableExport(int currentYear, int previousYear, String month)

Devuelve el valor del primer porcentaje del reporte de exportaciones el cual se obtiene dividiendo el real acumulado del año actual entre el real acumulado del año anterior.

12- double PercentValueOfTheSecondTableExport(int year, String month)

Devuelve el valor del segundo porcentaje del reporte de exportaciones el cual se obtiene dividiendo el real acumulado del año actual entre el plan del año actual.

13- double PercentValueOfTheThirdTableExport(int year, String month)

Devuelve el valor del tercer porcentaje del reporte de exportaciones el cual se obtiene dividiendo el real acumulado del año actual entre el plan acumulado del año actual.

3.6 Tratamiento de errores.

En la BD para lograr una respuesta rápida ante cualquier error se realizó un estudio de errores, para lanzar excepciones en un lenguaje de fácil comprensión para los usuarios. Se generó un paquete de excepciones, para informarle al desarrollador los diversos errores que cometa. En el *GenericDao* se evitan, minimizan y tratan los posibles errores. El *GenericDao* utiliza un componente para el seguimiento de errores que permite la escritura de cualquier excepción en un archivo el cual puede ser controlado para mandar alertas indicando la corrección de los errores existentes.

3.7 Validación funcional.

Para comprobar que la base de datos para el módulo de la Dirección de Exportación y Sustitución de Importaciones cumple con los requisitos funcionales definidos, es necesario realizarle pruebas antes de dar por terminado el proceso de diseño. El propósito de estas pruebas es saber a la hora de integrarla con la

aplicación si es capaz de devolver correctamente los registros determinados por el usuario y de conocer si a partir de dicha aplicación se puede realizar operaciones tales como insertar, eliminar, buscar y actualizar sobre las tuplas de la base de datos.

3.7.1 Pruebas Unitarias.

Son pruebas dirigidas a probar clases java aisladamente y están relacionadas con el código, la responsabilidad de cada clase y sus fragmentos de código más críticos. (Rodríguez 2006)

La realización de las pruebas unitarias permite comprobar la correcta ejecución de un segmento de código de la aplicación, examinando el estado del programa en varios puntos, determinando si el estado real coincide con el esperado.

3.7.2 Diseño de los casos de pruebas a la clase GenericDao.

En la ingeniería del software, los casos de pruebas son un conjunto de condiciones o variables bajo las cuales 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.

Capítulo3: Implementación y Validación de la Solución Propuesta

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
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.
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.
find	Retorna una lista de objetos de una entidad específica	<i>Class<T></i> entityClass	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.
FindByPK	Retorna un objeto de una	1. <i>Class<T></i> entityName	Busca en la base de	Chequear que el

	entidad específica <T> AbstractEntity	2. long id	datos un objeto de una entidad especificada por la clave primaria	objeto retornado coincide con el que está en la base de datos
delete	No retorna nada	AbstractEntity entity	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
findByParamAndValue	<T> List<T>	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

Tabla 4 Diseño de casos de prueba para la clase GenericDao.

3.7.3 Resultado de las pruebas a la clase GenericDao.

Para la implementación de los requisitos se tuvieron en cuenta 4 iteraciones. En ellas se realizaron pruebas unitarias de cada uno de los requisitos necesarios por la aplicación. A continuación se muestra el código correspondiente al método save (insertar) al que se le realizó la prueba unitaria para verificar su correcto funcionamiento. Se aplicaron las pruebas a todos los métodos de la clase GenericDao, a cada funcionalidad de estos le fueron generados los resultados esperados y posteriormente comparados con los arrojados de la invocación de los mismos.

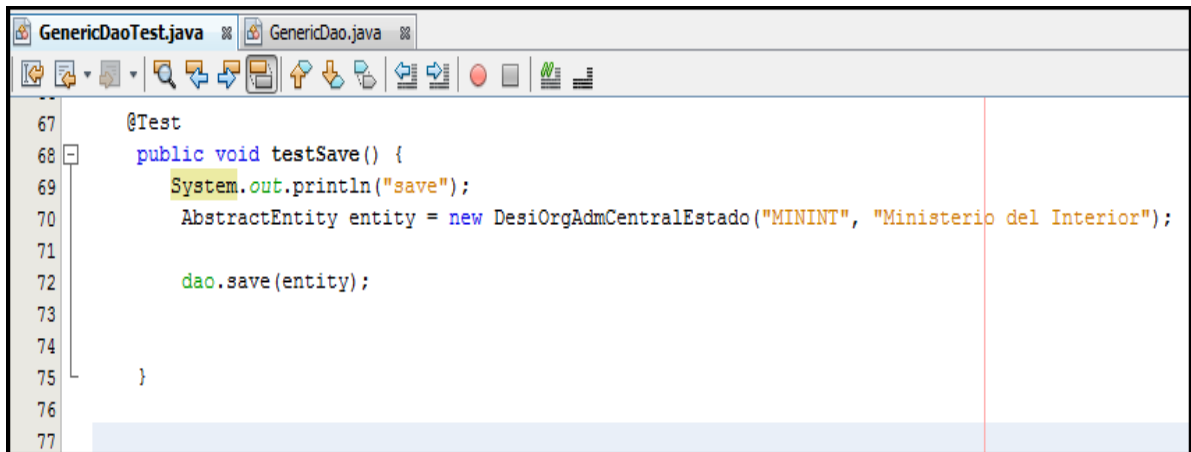


Fig. 11. Prueba al método save de la clase GenericDao con junit.

En la imagen que se muestra a continuación se puede apreciar el resultado satisfactorio de las pruebas realizadas con junit a los métodos implementados en la clase GenericDao.

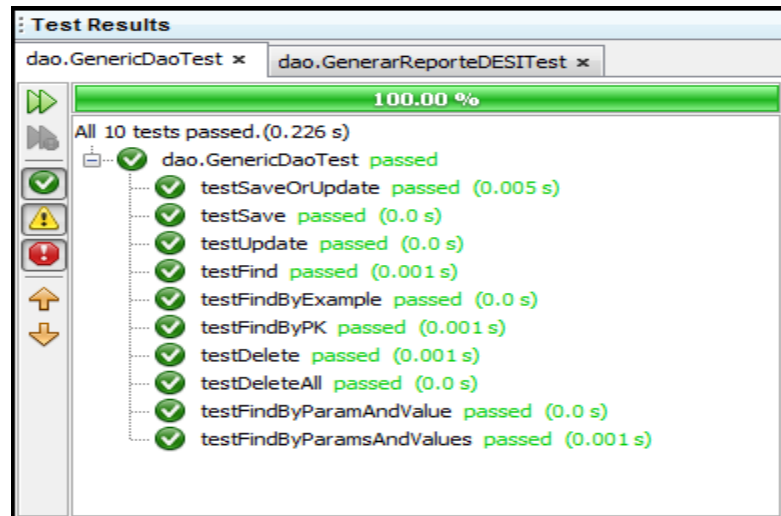


Fig. 12. Resultados de las pruebas con junit a la clase GenericDao.

Capítulo3: Implementación y Validación de la Solución Propuesta

En la siguiente tabla se muestran los resultados de cada una de las pruebas que se hicieron a los métodos de la clase GenericDao para la entidad DesiOrgAdmCentralEstado.

GenericDao				
Requisitos	1	2	3	4
Insertar entidad	Pasó	Pasó	Pasó	Pasó
Modificar entidad	Pasó	Pasó	Pasó	Pasó
Buscar entidad por PK(Llave primaria)	No Pasó	Paso	Pasó	Pasó
Eliminar entidad	No Pasó	Pasó	Pasó	Pasó
Buscar Entidad por parámetros y valores específicos	No Pasó	No Pasó	No Pasó	Pasó
Eliminar entidad	No Pasó	No Pasó	Pasó	Pasó

Tabla 5. Comportamiento de los métodos por iteraciones.

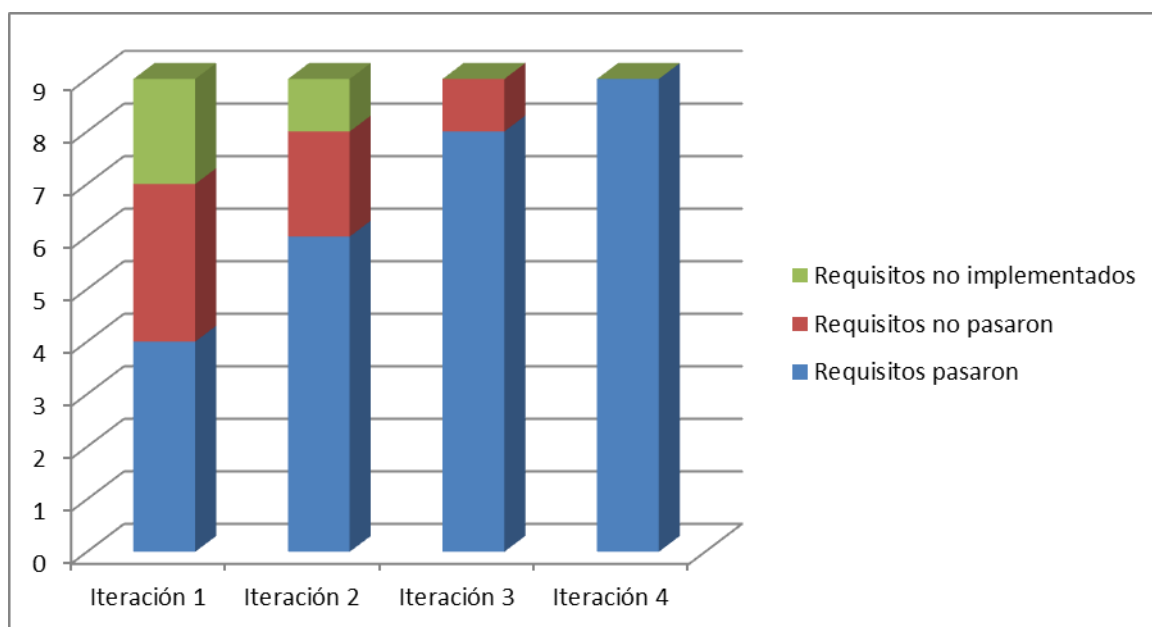


Gráfico 1. Resultados de las pruebas a la clase GenericDao por iteraciones.

3.7.4 Diseño de los casos de pruebas a la clase GenerarReporte.

Escenario	Respuesta	Parámetros	Descripción	Flujo
FindByHQL	Retorna una lista de coincidencias	<i>Consulta en lenguaje HQL</i>	Busca en la base de datos todas las instancias que coincidan con los criterios de búsqueda.	Chequear que se obtenga la lista de entidades.
SeeksImportPlanBy Year	Retorna una lista de planes de sustitución de importaciones	<i>Int</i> año	Selecciona las entidades que coincidan con el año entrado	Chequear que se obtenga la lista de entidades.
ThirdValueOfTheReportImport	Retorna el valor que	<i>Int</i> añoActual,	Realiza el cálculo a	Chequear que se

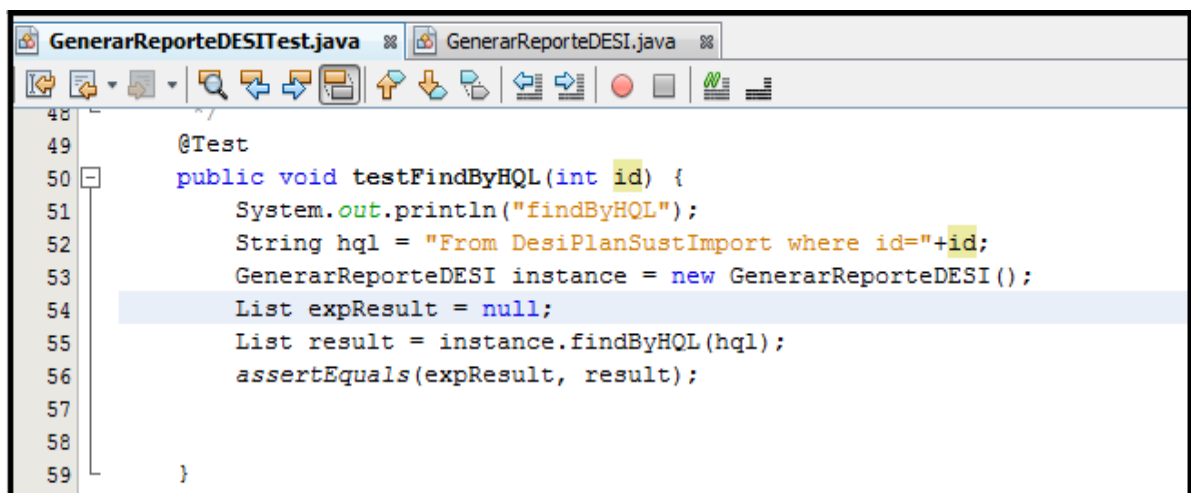
	coincida con los criterios de búsqueda.	<i>Int</i> añoAnterior.	insertar en el reporte	obtenga el valor a insertar.
SeeksImportPlanByYearAndMonth	Retorna una lista de planes de sustitución de importaciones.	<i>Int</i> año, <i>String</i> mes	Selecciona las entidades que coincidan con el año y el mes entrado	Chequear que se obtenga la lista de entidades.
SeeksRealImportPlanByMonth	Retorna una lista de reales de sustitución de importaciones.	<i>String</i> mes	Selecciona las entidades que coincidan con el mes entrado	Chequear que se obtenga la lista de entidades.
SeventhValueOfTheReportImport	Retorna el valor que coincida con los criterios de búsqueda.	<i>Int</i> añoActual, <i>Int</i> añoAnterior, <i>String</i> mes.	Realiza el cálculo a insertar en el reporte	Chequear que se obtenga el valor a insertar.
EighthValueOfTheReportImport	Retorna el valor que coincida con los criterios de búsqueda.	<i>Int</i> añoActual, <i>Int</i> añoAnterior, <i>String</i> mes.	Realiza el cálculo a insertar en el reporte	Chequear que se obtenga el valor a insertar.
NinthValueOfTheReport	Retorna el valor que coincida con los criterios de búsqueda	<i>Int</i> añoActual, <i>Int</i> añoAnterior, <i>String</i> mes.	Realiza el cálculo a insertar en el reporte	Chequear que se obtenga el valor a insertar.
SeeksRealExportPlanByYearAndMonth	Retorna una lista de reales de exportación.	<i>Int</i> año, <i>String</i> mes	Selecciona las entidades que coincidan con el año y el mes entrado	Chequear que se obtenga la lista de entidades.
SeeksExportPlanByYear	Retorna una lista de planes de exportación.	<i>Int</i> año.	Selecciona las entidades que coincidan con el año entrado	Chequear que se obtenga la lista de entidades.
PercentValueOfThe	Retorna el	<i>Int</i>	Realiza el	Chequear

FirstTableExport	valor que coincida con los criterios de búsqueda.	añoActual, <i>Int</i> añoAnterior, String mes.	cálculo a insertar en el reporte	que se obtenga el valor a insertar.
PercentValueOfTheSecondTableExport	Retorna el valor que coincida con los criterios de búsqueda.	<i>Int</i> año, String mes	Realiza el cálculo a insertar en el reporte	Chequear que se obtenga el valor a insertar.
PercentValueOfTheThirdTableExport	Retorna el valor que coincida con los criterios de búsqueda.	<i>Int</i> año, String mes	Realiza el cálculo a insertar en el reporte	Chequear que se obtenga el valor a insertar.

Tabla 6 Diseño de casos de prueba para la clase GenerarReporte.

3.7.5 Resultado de las pruebas a la clase GenerarReporte.

Para la implementación de los requisitos se tuvieron en cuenta 4 iteraciones. Al igual que a la clase GenericDao se le realizaron las pruebas a todos los métodos implementados en esta clase. A continuación se muestra el código correspondiente al método FindByHQL al que se le realizó la prueba unitaria para verificar su correcto funcionamiento.



```

48
49     @Test
50     public void testFindByHQL(int id) {
51         System.out.println("findByHQL");
52         String hql = "From DesiPlanSustImport where id="+id;
53         GenerarReporteDESI instance = new GenerarReporteDESI();
54         List expResult = null;
55         List result = instance.findByHQL(hql);
56         assertEquals(expResult, result);
57
58
59     }

```

Fig. 13. Prueba al método FindByHQL de la clase GenerarReporte con junit.

En la imagen que se muestra a continuación se puede apreciar el resultado satisfactorio de las pruebas realizadas con junit a los métodos implementados en la clase GenerarReporte.

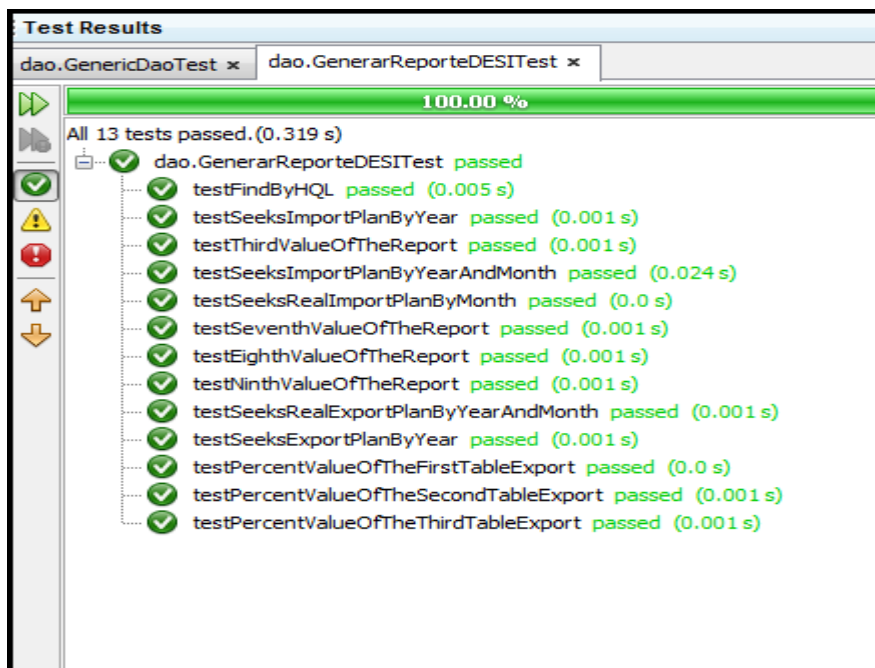


Fig. 14 Resultados de las pruebas con junit a la clase GenerarReporte.

En la siguiente tabla se muestran los resultados de las pruebas que se hicieron a los métodos de la clase GenerarReporte para la entidad DesiPlanSustImport.

GenerarReporte				
Requisitos	1	2	3	4
Obtener Plan anual de sustitución de importaciones dado el año	Pasó	Pasó	Pasó	Pasó
Obtener Real anual de sustitución de importaciones dado el año y el mes	No Pasó	No Pasó	Pasó	Pasó

Obtener Real anual de exportaciones dado el año y el mes	Pasó	Paso	No Pasó	Pasó
Obtener Plan anual de exportaciones dado el año	No Pasó	Pasó	Pasó	Pasó

Tabla 7. Comportamiento de los métodos por iteraciones.

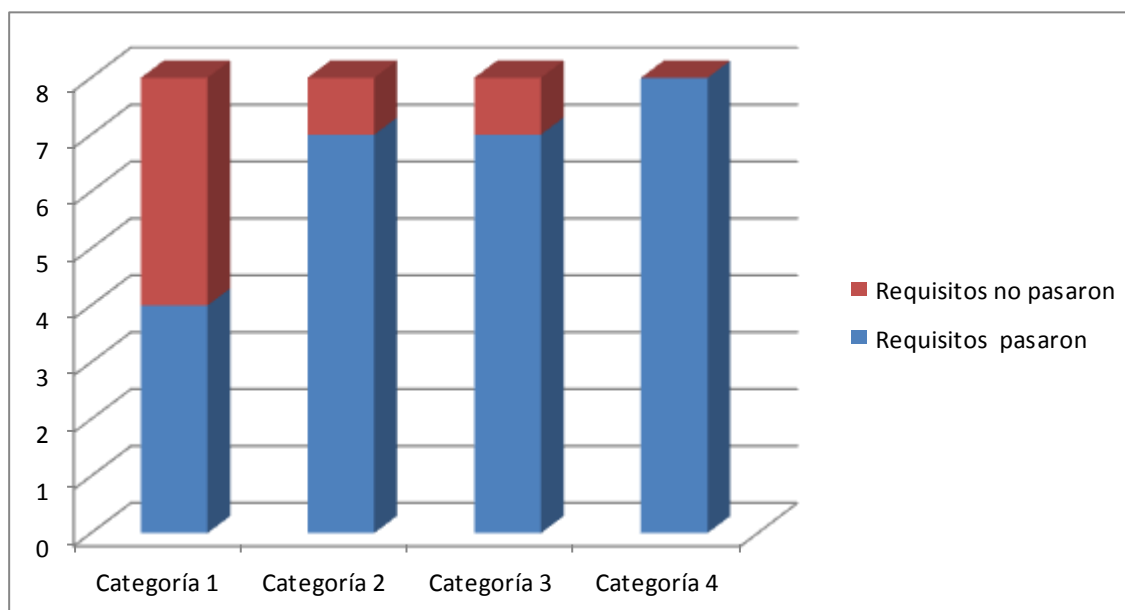


Gráfico 2. Resultados de las pruebas por iteraciones.

3.7.6 Conclusiones de las pruebas unitarias.

En el desarrollo de las pruebas se realizaron un total de 4 iteraciones en las cuales se encontraron y corrigieron errores llegando a un resultado positivo en todas las funcionalidades. El resultado general de las pruebas se muestra en la tabla a continuación.

Iteraciones	1	2	3	4
% Requisitos No Implementados	50	25	10	0
% Requisitos que No Pasaron	20	10	2	0
% Requisitos que Pasaron	10	40	50	54

Tabla 8 Resultados de las pruebas por iteraciones a la base de datos.

3.8 Pruebas de volumen.

Las pruebas de volumen consisten en examinar el funcionamiento del sistema cuando está trabajando con grandes volúmenes de datos, simulando las cargas de trabajo esperadas. A continuación se muestra el resultado satisfactorio de la inserción de 100 tuplas en la base de datos simultáneamente a todas las entidades.

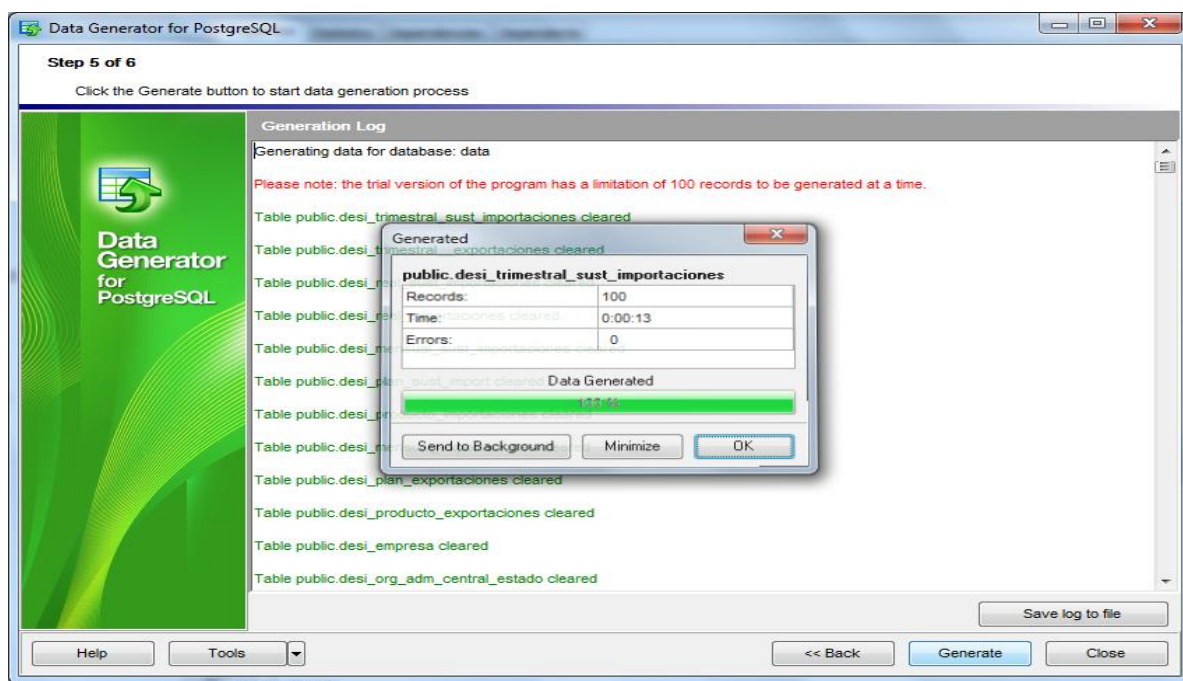


Fig. 15 Resultados satisfactorio de las pruebas de volumen a la base de datos.

Conclusiones Parciales

En este capítulo se expuso la validación de la solución propuesta en el capítulo anterior, probando los métodos implementados y su ejecución satisfactoria, que evidencian el correcto funcionamiento de los componentes para la persistencia de objetos.

Conclusiones Generales

- Se realizó un análisis del estado del arte sobre bases de datos existentes en el mundo relacionadas con la exportación y la sustitución de importaciones además de estudio detallado de las alternativas para la persistencia de objetos en Java. A partir de este análisis y estudio fueron cumplidos el objetivo general y tareas especificadas y se arriba a las siguientes conclusiones:
- A partir del modelo de datos propuesto se diseñó una base de datos la cual tiene como objetivo la seguridad y la centralización de la información de exportación y sustitución de importaciones del MINCEX.
- Se implementó la capa de acceso a datos la cual cuenta con las funcionalidades que optimizan la persistencia de la información.
- Se probaron todos los métodos implementados para la persistencia de objetos.

Recomendaciones

- ✓ Refinar o incluir nuevas funcionalidades en la BD para la persistencia de objetos.
- ✓ Aplicar las concepciones adoptadas para el desarrollo de la capa de acceso a datos en otros proyectos que utilicen las mismas tecnologías e incluso en otras plataformas de desarrollo.
- ✓ Implementar un sistema de réplicas que permita el intercambio de información entre la BD central y las distintas sedes que necesiten implantarlas.
- ✓ Utilizar el contenido de esta investigación como referencia para el desarrollo de próximas BD en los sistemas gubernamentales.

Referencias Bibliográficas

1. **Barajas**, Matilde Moreno. EL SERVICIO DE CONSULTA DINÁMICA. 2009. GestioPolis. [En línea] 12 de noviembre 2010.
2. **Bartle, Phil. 2009.** INFORMACIÓN PARA LA GESTIÓN y gestión de la información. [En línea] 2009. [Citado el: 5 de abril de 2012.] <http://www.scn.org/mpfc/modules/mon-miss.htm>.
3. **Boríssov**, Boiko. Diccionario de Economía Política.
4. **Cali**, Santiago. (2006) Base de Datos: Conceptualización y sistemas de administración. [En línea] 25 noviembre de 2011. <http://www.buenastareas.com/ensayos/Bases-De-Datos-Conceptos-Basico/1536159.html>.
5. **Codd, E. F.** (2011) The Relational Model for Database Management: Version 2. Reading, Mass. Los Altos, CA: Addison-Wesley Publishing.
6. **Daniels**, John D. L. H. (2004). Negocios internacionales. Prentice Hall.
7. **Dante, Gloria. 2010.** Gestión de la Información. [En línea] 2010. [Citado el: 3 de abril de 2012.] <http://www.fahce.unlp.edu.ar/academica/Areas/bibliotecologia/Catedras/gestindeinformacin/>.
8. **Date**, Christopher. 2003. Introducción a los Sistemas de Bases de Datos. Primera Parte. La Habana: Felix Varela, 2003.
9. **Española, Real Academia. 2009.** Real Academia Española. [En línea] Real Academia Española, 2009. [Citado el: 5 de abril de 2012.] <http://buscon.rae.es/drael/>.
10. **García, 2005**, Ignacio García (2005) título: Base de datos Modelo red general.
11. **Gómez Vieites**, Álvaro (2007). Enciclopedia de la Seguridad Informática.

12. **Heber, Romero. 2009.** Gestión del conocimiento y la información en el Polo de Gestión Universitaria. Universidad de las Ciencias Informáticas. Ciudad de la Habana: s.n., 2009.
13. **Luque**, Fernando Sánchez - DCE 4 Estrellas. Aplicaciones en N-Capas (12/Oct/2005).
14. **Mato**, Rosa María García. Sistemas de Bases de Datos. La Habana: Pueblo y Educación, 2005.
15. **Murray, Pablo. 2003.** GESTIÓN – INFORMACIÓN - CONOCIMIENTO. Escuela de Bibliotecarios de la Biblioteca Nacional de Buenos Aires. Buenos Aires, Argentina: s.n., 2003.
16. **PgAdminIII.** 2008. PgAdmin III. [Online] 03 10, 2008. http://www.guia-ubuntu.org/index.php?title=PgAdmin_III.
17. **Riccuti**, 1994. Riccuti, Mike (1994). "Multidimensional Analysis: Winning the Competitive Game".
18. **Rodríguez**, Jorge. Pruebas Unitarias. 2006.
19. **Rodríguez**, Luis. Luis Rodríguez (2003). Bases de datos documentales.

Bibliografías Consultadas

- ✓ Akif, M., S. Brodhead, et al. (2001). Java y XML. Referencia para programadores., Wrox Press Ltd. 817 p.
- ✓ Ambler, 2002. Ambler, S. (2002). "Mapping Objects to Relational Databases". Extraído el 26 de abril, 2007. white paper. www.amblysoft.com/essays/mappingObjects.html.
- ✓ Bauer, C. y G. King. (2005). Hibernate in Action. A guide to the concepts and of object/relational mapping. Manning Publications Co.
- ✓ Hoeller, J. (2007). "Class HibernateDaoSupport" Extraído el 27 de Febrero, 2007 de <http://www.springframework.org/docs/api/org/springframework/orm/hibernate/support/HibernateDaoSupport.html>.
- ✓ Hommel, S. (1999). "Conversiones de Código para el lenguaje de programación JAVA." Extraído el 2 de Febrero, 2007 de <http://usuarios.lycos.es/manualesjava/manuales/convencionescodigojava/convencionescodigojava.pdf>.
- ✓ King, G. (2006). "Hibernate Reference Documentation 3.2.2". Extraído el 5 de Enero, 2007 de http://www.hibernate.org/hib_docs/v3/reference/en/pdf/hibernate_reference.pdf.
- ✓ Microsystem, S. (2004). "Interface Serializable.", JavaTM 2 Platform Standard Ed. 5.0 Extraído el 15 de Abril, 2007 de <http://java.sun.com/j2se/1.5.0/docs/api/java/io/Serializable.html>.
- ✓ Panduranga, S. N. (2005). Pro Hibernate: From Professional to Expert, Apress L. 325 p.

- ✓ Richardson, C. (2006). *Pojos in Action: Developing Enterprise Applications with Lightweight Frameworks*, Manning Publications Company. 560 p.
- ✓ Tutorial: Crear las capas de acceso a datos y de lógica empresarial en ASP.NET Disponible en: <http://msdn.microsoft.com/es-es/library/bb470374.aspx>.
- ✓ Yailin Fundora Carrasco. Diseño e implementación de la base de datos del Sistema de Gestión Académica de Postgrado, 2011.
- ✓ Yordan Gallardo Avilés. Base de Datos del Sistema para el Manejo Integral de la Perforación de Pozos de Petróleo y Gas (SIPP). Versión 2.0. 2011.
- ✓ Elián Luis Fernández de la Pera, Danis López Naranjo. Desarrollo de la capa de acceso a datos para los módulos de administración y nomencladores del sistema de gestión de inventario de almacenes SIGIA, 2007.
- ✓ Glenda María Santos Hernández. Diseño de la base de datos para la plataforma de televisión informativa PRIMICIA versión 2.0 , 2011.
- ✓ Herramientas CASE para el proceso de desarrollo de Software (página 2) - Monografias.com. [cited 14 December 2011]. Available from world wide web: <http://www.monografias.com/trabajos73/herramientas-case-proceso-desarrollo-software/herramientas-case-proceso-desarrollo-software2.shtml>.
- ✓ Israel Pérez González / Madelaine Sosa Díaz. Propuesta de optimización de la base de datos para el proyecto Sistema de Informatización de la Gestión de las Fiscalías, 2011.
- ✓ Jorge Rodriguez. Pruebas Unitarias. 2006.
- ✓ Lianet Camejo Dominguez. Diseño de una Base de Datos para el Sistema de Captura y Catalogación de Medidas.
- ✓ M. Piattini, E. Marcos, C. Calero y B. Vela.(2006) Tecnología y Diseño de Bases de Datos.

- ✓ Manuales : Acceso a datos .ed. Última actualización:. (). Disponible en:<http://www.canalvisualbasic.net/manual/acceso-datos-dao/acceso-datos/>.
- ✓ Metodología de diseño de base de datos .ed. Disponible en:<http://www.slideshare.net/emnero/metodologia-de-diseo-de-base-de-datos-360978>.
- ✓ PgAdmin III - Guía Ubuntu. [cited 14 December 2011]. Available from World Wide Web: <http://www.guia-ubuntu.org/index.php?title=PgAdmin_III>.
- ✓ PostgreSQL: Welcome .ed. Última actualización:. (). Disponible en:<http://www.postgresql.org/>.
- ✓ <http://www.un.org/spanish/News/fullstorynews.asp?NewsID=19494> base de datos unctad
- ✓ http://www.gtis.com/Spanish/GTIS_GTA_sp.html
- ✓ http://www.gtis.com/Spanish/GTIS_About_sp.html
- ✓ http://websie.eclac.cl/badecel/badecel_new/index.html
- ✓ <http://corporacion.remington.edu.co/biblioteca/recursos-lectronicos/basesde-datos/base-de-datos-libres>