

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD REGIONAL DE ARTEMISA



**ADMINISTRACIÓN DE DATOS DEL MÓDULO DE LA DIRECCIÓN
DE TRANSPORTE Y SEGURO DEL MINISTERIO DEL COMERCIO
EXTERIOR Y LA INVERSIÓN EXTRANJERA (MINCEX)**

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO
EN CIENCIAS INFORMÁTICAS**

AUTOR: José Ramón Cabrera Alfonso

TUTORA: Ing. Marlies Quiala Torres

DECLARACIÓN DE AUTORÍA

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

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

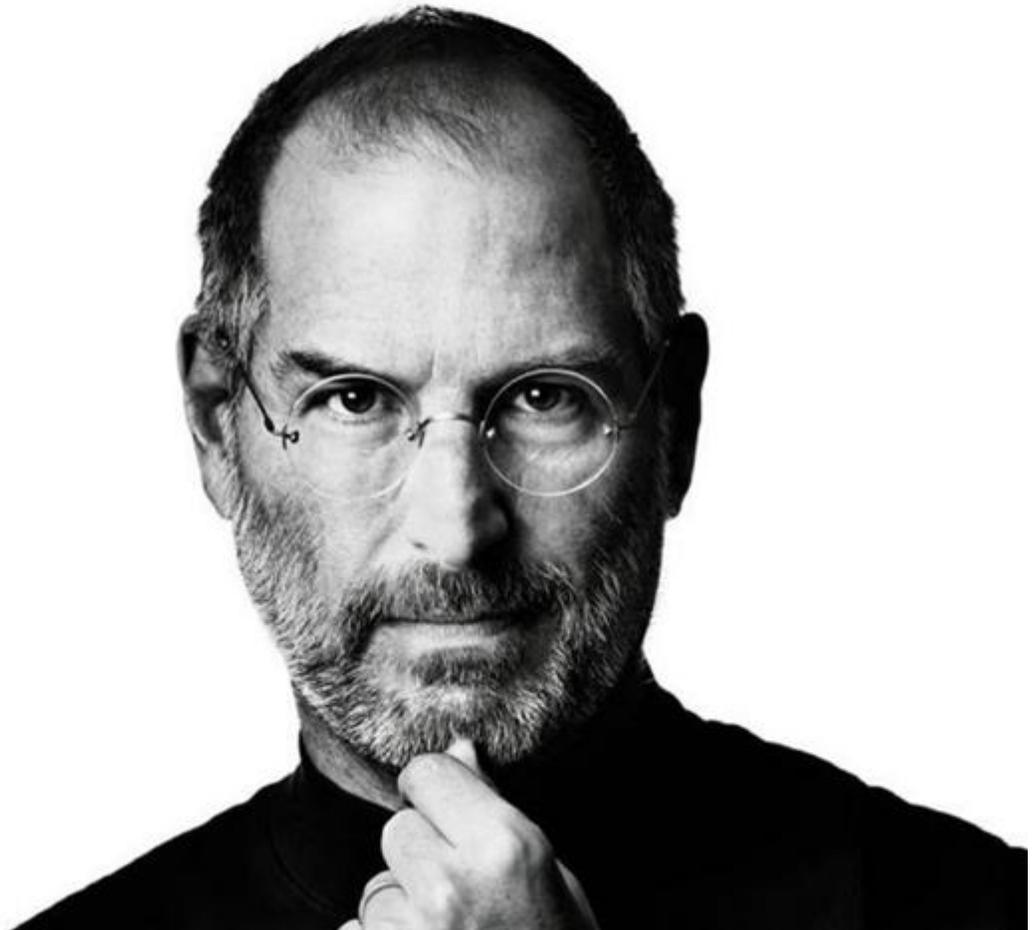
Firma del Autor
José Ramón Cabrera Alfonso

Firma del Tutor
Ing. Marlies Quiala Torres

FRASE

“El único modo de hacer un gran trabajo es amar lo que haces. Si no lo has encontrado todavía, sigue buscando. No te acomodes. Como con todo lo que es propio del corazón, lo sabrás cuando lo encuentres.”

Steve Jobs



AGRADECIMIENTOS

A toda, toda mi familia en especial a mis abuelitos Basilio y Lorelei, que han hecho hasta lo imposible para que yo pudiese llegar al lugar donde estoy.

A todos mis amigos de la facultad, gracias por su apoyo y su amistad, sepan que nunca olvidaré los momentos que compartimos.

A Luis Ernesto y Amed por su ayuda en la realización de la tesis.

A mi tutora Marlies, por su apoyo para el desarrollo de la tesis.

A la Revolución por darme esta oportunidad.

DEDICATORIA

A mi padre donde quiera que este, por sus consejos, su apoyo, por ser mi amigo, a mis abuelitos por su amor infinito, a mis hermanitas, a mi mamá, a toda mi familia.

José Ramón

RESUMEN

Una base de datos es una colección de registro, que permite guardar grandes cantidades de información de forma centralizada, posibilitando que se pueda consultar y utilizar fácilmente lo que se desea. En fin, permite a grades rasgos mejorar la recuperación, la persistencia y la centralidad de cada uno de los datos almacenados. Así como, la redundancia mínima, la integridad de los datos, las consultas complejas optimizadas, la seguridad de acceso y auditoría, y el acceso concurrente por parte de múltiples usuarios.

Para construir una base de datos, se debe conocer el negocio y la información con la que se trabaja, para la construcción de la misma en la investigación se realizó un análisis del proceso de gestión de la información y administración de los datos en la Dirección de Transporte y Seguro, la cual forma parte de la estructura organizativa del MINCEX. Para ello, se estudiaron varios conceptos y temas a tener en cuenta para realización de la misma. Entre los conceptos se encuentran: base de datos, gestión y sistemas gestores de base de datos, y entre los temas las principales herramientas, tecnologías, metodologías, los patrones de diseño, específicamente el patrón Data Access Object, y un estudio del arte relacionado con las base de datos, para ver si alguna satisfacía las necesidades detectadas en la entidad que se tiene como objeto físico. Por último, se le realizaron un conjunto de pruebas en busca de errores a lo implementado.

Palabras clave: bases de datos, gestión, herramientas, información, metodología, patrones, sistemas gestores de base de datos, tecnologías.

ÍNDICE

Introducción.....	10
Capítulo 1: Fundamentación teórica.	18
1.1 Introducción.	18
1.2 Origen de las bases de datos.	18
Definición de base de datos (DB)	21
1.2.1 Sistema Gestor de base de datos (SGBD).	22
Características de los Sistemas Gestores de Base de Datos (SGBD):	22
Arquitectura de los Sistemas Gestores de Base de Datos (SGBD)	24
Lenguajes del SGBD.	25
Diferencia entre procesamiento de archivos y procesamiento de base de datos.	26
1.3 Tendencias nacionales e internacionales.	26
1.4 Análisis de las posibles herramientas a utilizar.	27
Hibernate	29
Las características de esta tecnología son las siguientes:	29
Framework Spring	30
Módulos que posee Spring	30
1.5 Ventajas y Desventajas de PostgreSQL:.....	31
Ventajas:	31
Desventajas:	32
Elección del SGBD a utilizar en la solución propuesta (PostgreSQL):	32
1.6 Metodologías de desarrollo de software	32
Programación Extrema XP:	33
Características De La Metodología XP	33
Metodología Scrum:	35
Metodología a utilizar (Metodología SXP):.....	35
1.7 Conclusiones parciales:.....	35
Capítulo 2: Diseño y arquitectura de la Capa de Acceso a Datos.	36
2.1 Introducción	36
2.2 Descripción de los requisitos funcionales y no funcionales.....	36
Requisitos funcionales (RF)	37
Requisitos no funcionales (RNF)	39

2.3 Modelo de entidad relación (MER):	40
Tipos de Atributos	41
Diagrama entidad relación de la base de datos de la Dirección de Transporte y Seguro del MINCEX.	41
2.4 Normalización de las bases de datos	51
Existen varias Formas Normales (FN)	52
2.5 Conclusiones Parciales:	53
Capítulo 3: Implementación y validación de la Capa de Acceso a Datos.	54
3.1 Introducción:	54
3.2 Validación teórica	54
Integridad de los datos	55
Integridad de la clave	55
Integridad del dominio	55
Integridad referencial	55
Integridad semántica	56
Análisis de redundancia	56
Seguridad de los datos	56
3.2 Capa de acceso a datos	57
Patrones de diseño de software	57
Data Access Object (DAO):	58
3.3 Casos de pruebas	62
Tabla 1. Prueba al Método save.	63
Tabla 2. Prueba al Método saveOrUpdate.	63
Tabla 3. Prueba al Método find.	63
Tabla 4. Prueba al Método findByPK.	64
Tabla 5. Prueba al Método findByParamAndValue	64
Tabla 6. Prueba al Método findByParamsAndValues	64
Tabla 7. Prueba al Método update.	65
Tabla 8. Prueba al Método delete.	65
Tabla 9. Prueba al Método deleteAll.	65
3.4 Resultados y Funcionalidades obtenidos	65
Resultados	65
Funcionalidades	66

3.5 Conclusiones Parciales:	66
Conclusiones generales	67
Recomendaciones	68
Referencias Bibliográficas	69
Bibliografías.....	70
Glosario de términos.....	72
Anexos	75
Anexo 1: Modelo de Entidad relación de la base de datos del módulo de transporte y Seguro del MINCEX.	75
Anexo 2: Metodología de Desarrollo de Software a utilizar, SXP.....	76
Anexo 3: Fases de la Metodología de Desarrollo de Software XP.	77

INTRODUCCIÓN

Los conocimientos han tenido siempre un valor incalculable, desde los primeros tiempos el hombre se vio en la necesidad de almacenar este conocimiento de forma tal que pudiera gestionar el mismo de una forma favorable. Con el objetivo de usar esa información para diversos fines de la vida cotidiana así como conservar los documentos y con ello rescatar el patrimonio y el acervo cultural de esos tiempos, y además este debería ser usado por futuras generaciones. Pero sin los medios adecuados almacenar y realizar búsquedas de información resultaban tareas muy complejas. Desde ese entonces los formatos en que ha sido guardada la información han sufrido muchísimos cambios que han sido el producto de los cambios culturales, políticos y económicos por los cuales a transitado la humanidad, es así que de las tablillas de arcilla se fue pasando a la madera, el pergamino y papel entre otros.

Con la evolución de las tecnologías se ha transformado la vida del ser humano, el surgimiento y uso de las máquinas computadoras ha servido como medio para optimizar tareas de la vida diaria. Esto convirtió a las maquinas computadoras en potentes herramientas, casi indispensables para muchas actividades que se han logrado automatizar, ganando así en calidad y tiempo en los servicios en las que se utilizan.

En la actualidad la gestión de la información mediante la utilización de estos medios informáticos se le brinda especial atención. Los mismos son capaces de agilizar enormemente la fluidez de los procesos que se llevan a cabo en las diferentes instituciones, imprimiéndole calidad, sustituyendo los procesos de manipulación tradicionales y llevando el control de la información, a través de métodos automatizados de almacenamientos. Propiciando un ambiente de trabajo que facilita la toma de decisiones y el manejo de resultados estadísticos, además, es una forma de almacenar la información más eficiente, segura y rápida. Este sistema de almacenamiento digital trae muchísimas ventajas por ejemplo:

Disminución del riesgo de la pérdida de información y datos, que puede ser

ocasionada por incendios, robos, inundaciones. Además, los documentos se conservan intactos a lo largo del transcurso del tiempo. Los documentos no se sufren deterioro ya que se encuentran como archivos digitales. Permite un acceso inmediato a los documentos según los diferentes criterios de búsqueda a los que puede tener acceso el usuario, por ejemplo, por año, por tipo de documento, por cliente entre otros. Posibilita también el acceso desde diversos lugares, a través de Internet o el tipo de red con la cual se trabaje o se tenga acceso. Permite establecer criterios de acceso de forma flexible y fácil, el acceso simultáneo de varios usuarios al mismo archivo digital. Es posible también duplicar y enviar los documentos a través de medios digitales (sin sacarlo de su ubicación).

Las bases de datos son el método preferido para el almacenamiento estructurado de datos. Desde las grandes aplicaciones multiusuario, hasta los teléfonos móviles y las agendas electrónicas utilizan tecnología de bases de datos para asegurar la integridad de los datos y facilitar la labor tanto de usuarios como de los programadores que las desarrollaron. Teniendo en cuenta estos aspectos, las bases de datos han demostrado superioridad por lo cual se ha convertido en una necesidad su uso, sobre todo, en casos donde es necesario almacenar y trabajar con grandes volúmenes de información.

A pesar del impulso en las últimas décadas de las Tecnologías de la Informática y las Comunicaciones (TICs), el Ministerio del Comercio Exterior y la Inversión Extranjera de la República de Cuba (MINCEX) es un ministerio que cuenta con un incipiente nivel de. Por la eficiencia y la calidad en los procesos que brindan las TICs, se crea un Acuerdo de Colaboración entre el MINCEX y la Facultad Regional Mártires de Artemisa (FRA). Situada en la actual provincia de Artemisa y nacida bajo el fervor de la batalla de ideas que lleva a cabo el pueblo cubano, esta forma parte de la Universidad de las Ciencias Informáticas (UCI). Además, cuenta con los medios e infraestructura necesarios para el desarrollo de software, así como, con un personal especializado formado por alumnos y profesores. Por ello y el hecho de que en esta facultad se vincula el estudio con la producción e investigación, se crea este acuerdo. El objetivo del mismo es darle soluciones informáticas a los

problemas de cada dirección, centrándolas en un sólo Sistema Informativo para todo el Ministerio.

Es esencial aclarar que la Inversión Extranjera en Cuba, surge dada la imposibilidad de acceder a financiamientos externos, tecnologías de avanzada y mercados externos que presenta el país. Principalmente debido al Bloqueo Económico, Comercial y Financiero impuesto por el Gobierno de Estados Unidos de América, por más de 50 años, lo que produce en materia de Inversión Extranjera una afectación aproximada de 2 mil millones USD anualmente. La misma se comprende como un complemento al desarrollo de la economía nacional y se caracteriza por un alto grado de selectividad y promoción al desarrollo sostenible en correspondencia con el deber de cuidar el medio ambiente.

El comercio exterior de Cuba ejerce un fuerte y determinante impacto en la economía del país. Por tal motivo entre sus objetivos se encuentra la diversificación de sus socios comerciales, el trabajar por lograr un reordenamiento de la política comercial en la búsqueda de mercados favorables para sus exportaciones, en la diversidad y competitividad de sus renglones exportables, así como, en el objetivo estratégico de la sustitución de exportaciones. Dentro de los principales socios comerciales de Cuba, se encuentran Venezuela, China, Rusia, España y Brasil. Cuba mantiene relaciones comerciales con más de 170 países. Desde los años 90, la exportación de servicios ha tenido un creciente papel en la economía cubana. Esto se manifiesta en su influencia cada vez más importante en los resultados del comercio exterior cubano.

La venta de servicios profesionales, especialmente en salud, ingeniería, informática y biotecnología, se consolidan como la mayor fuente de divisas del país. El ambiente sano y seguro que brinda Cuba para los negocios en general, es un punto a favor determinante para promover las exportaciones de bienes y servicios, la colaboración, la inversión extranjera y avanzar en la sustitución de importaciones. Lo anterior contribuye al desarrollo productivo y científico-técnico del país.

El MINCEX posee una estructura organizativa de 23 direcciones, una de estas

direcciones, es la dirección de Transporte y Seguro, que tiene como misión elaborar y controlar el cumplimiento de la política sobre las condiciones de contratación de las mercancías de exportación e importación en relación con la transportación internacional. Además el seguro de las cargas y las reclamaciones e igualmente con los planes de transportación internacional de las cargas por zonas geográficas del país. Esto se coordina previamente con los Organismos de la Administración Central del Estado. Para el desarrollo de sus actividades se apoyan de procedimientos que estarán estrechamente vinculado con las siguientes funciones : participación en los Comités de Contrataciones de las Empresas del MINCEX; atención a empresas comercializadoras de la rama en materia de transporte y seguros; relativos a embarques de cargas con navieras; atención y tramitación de consultas recibidas de las empresas de la rama sobre actividad de Transporte y Seguros; análisis diferenciados con empresas incumplidoras: esta función hace referencia a los análisis que se hacen con las empresas que presentaron contenedores con más de 3 días en puerto al cierre del mes anterior; trabajos conjuntos con el MITRANS sobre políticas y casos operativos; trabajo conjunto con ESICUBA para fomentar el uso del seguro cubano; emisión de Dictámenes o medidas de casos recibidos; relaciones sistemáticas con entidades relacionadas con transporte y seguros; y el trabajo en el marco de la OPTEI.

En la dirección de transporte y seguro actualmente existe variedad en cuanto a las responsabilidades que presentan los especialistas. Estas están relacionadas con el control que lleva el MINCEX, ya sea, diario, mensual, quinquenal o anual, sobre la tramitación de fletes, cargas y seguros, así como de las transitorias, teniendo en cuenta que para la confección de estas informaciones dependen de la información que les tributan las empresas exportadoras, las importadoras, y los Organismos Centrales del Estado.

Para el desarrollo de las actividades, se apoyan de las informaciones que les mandan las empresas y los organismos en Excel. Esta situación provoca que un dato mal o una llegada tardía tribute a que se vea afectado el trabajo que ellos realizan. En dependencia del documento o el informe que estén desarrollando, puede provocar

pérdidas económicas al país, que está relacionada específicamente por las mercancías, ya que estas tiene asociado un valor de flete y un valor de seguro, pago que se realiza en USD. Otro de los problemas que puede ocasionar el retraso de la información, es que una mala diligencia en cuanto a la gestión del transporte nacional, una vez que la mercancía este en la Aduana, es el recargo por los días de almacenamiento de las cargas agrupadas. La búsqueda de información constituye una labor obligatoria a la hora de hacer los planes, por las comparaciones que hacen en cuanto al plan anterior y el que se desarrolla, búsqueda que se hace engorrosa por el modo de almacenamiento que presentan. Esto trae consigo que, los informes que se encuentran en formato impreso sufran deterioro y provoquen falta de claridad en un dato determinado que se desea consultar. Además de que hace que el trabajo por parte del especialista sea lento y en ocasiones afectado. Después de analizar la sensibilidad y la importancia del tipo de información con la que se trabaja en la Dirección, se llega a la conclusión de que la misma presenta problemas en cuanto a el modo de almacenamiento , recuperación y manejo de los datos a la hora de efectuar alguna consulta.

Después de analizar cada una de las dificultades, se plantea el siguiente **problema de científico**: ¿Cómo contribuir a mejorar la persistencia y recuperación de los datos, así como la centralidad de la información en el proceso de gestión de la información que se lleva a cabo en la Dirección de Transporte y Seguro del MINCEX?

Analizando el problema anteriormente planteado y con el fin de darle solución al mismo se ha propuesto como **objeto de estudio**: proceso de gestión de la información y administración de datos en la Dirección de Transporte y Seguro del MINCEX.

Campo de Acción: diseño de la persistencia, recuperación y centralidad de la información en los sistemas de gestión de información.

Se propone como **objetivo general**: Desarrollar los componentes necesarios que contribuyan a mejorar la persistencia, recuperación y centralidad de la información en el proceso de gestión de la información que se lleva a cabo en la Dirección de

Transporte y Seguro del MINCEX.

De esta manera quedan planteadas en este contexto las siguientes **preguntas científicas**:

- ¿Cuáles son los fundamentos teóricos de la administración de datos de un sistema de gestión de información?
- ¿Cómo desarrollar la base de datos y la capa de acceso a datos para el módulo del sistema, perteneciente a la Dirección de Transporte y Seguro del MINCEX?
- ¿Cómo validar los resultados obtenidos?

Con el objetivo de dar respuesta a estas preguntas se plantean las siguientes **tareas de la investigación**:

- Establecer los fundamentos teórico-metodológicos para el desarrollo de sistemas informáticos para la gestión de la información.
- Caracterizar el proceso de gestión de información en la Dirección de Transporte y Seguro del MINCEX en lo relativo a la centralidad, persistencia y recuperación de la información.
- Diseño e implementación de la capa de acceso a datos para la Dirección de Transporte y Seguro del MINCEX.
- Realización de pruebas funcionales a la capa de acceso del sistema de gestión con el objetivo de validar la contribución lograda a través de la introducción del sistema informático en cuanto a la persistencia, recuperación y centralidad de la información en el proceso de gestión de la información en la Dirección de Transporte y Seguro del MINCEX.

Se espera obtener como **posible resultado**:

- La base de datos y la capa de acceso a datos que garantice el correcto almacenamiento y la recuperación de la información en el procesos de

gestión de información que se lleva a cabo en la Dirección de Transporte y Seguro del Ministerio del Comercio Exterior y la Inversión Extranjera (MINCEX).

Para facilitar la construcción de los modelos e hipótesis de investigación, se utilizaron **métodos científicos** tales como:

MÉTODOS TEÓRICOS:

- Análisis histórico – lógico: para realizar el análisis histórico de los procesos de gestión de información en la Dirección de Transporte y Seguro del MINCEX.
- Análisis y síntesis: para la confección del diseño teórico y metodológico de la investigación, análisis de las tendencias y tecnologías actuales de BD a nivel mundial.
- Modelación: para la representación del modelo lógico y físico de la BD.

MÉTODOS EMPÍRICOS:

- Observación: para identificar las principales actividades que desarrollan los procesos de la Dirección de Transporte y Seguro del MINCEX.

El documento está estructurado de la siguiente manera: tres capítulos que abarca todo lo referente a dicha investigación, así como, con la propuesta final que ha dado como resultado. Además de, las conclusiones generales, recomendaciones, referencias bibliográficas, bibliografías, anexos y glosario de términos.

A continuación se brinda una panorámica del contenido de cada uno de estos capítulos.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA: En este capítulo se realiza una fundamentación teórica acerca del estado del arte de la investigación, teniendo en cuenta la tecnología y herramientas más usadas y las posibles a utilizar, así como la justificación las mismas. Se realiza un análisis acerca de las

diferencias de los sistemas de archivo y las bases de datos.

CAPÍTULO 2: DISEÑO Y ARQUITECTURA DE LA CAPA DE ACCESO A DATOS: Se describen los patrones de diseño a utilizar y se definen los artefactos correspondientes al rol diseñador de base de datos, dígame artefactos, modelo entidad relación, modelo relacional. Además, se verán las reglas de transformación del modelo entidad relación y al modelo relacional.

CAPÍTULO 3: IMPLEMENTACIÓN Y VALIDACIÓN DE LA CAPA DE ACCESO A DATOS: Implementación de los artefactos del diseño de la base de datos e integración con la aplicación en la capa de persistencia.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

1.1 INTRODUCCIÓN.

Desde finales del pasado siglo el mundo se ha visto en la necesidad de mejorar los sistemas de almacenamiento de datos, el hecho de tratar con texto, gráficos y otros tipos de datos así lo exigen. En la actualidad las bases de datos son capaces de ofrecer una variedad de funcionalidades muy grande, esto tiene como objetivo hacer más cómodo el trabajo de los usuarios con aplicaciones de este tipo lo que contribuye en gran medida con el desarrollo y competitividad de las empresas y organizaciones que hacen uso de estas. Es por ello que en este capítulo se brinda una breve reseña histórica acerca del origen de las bases de datos, la diferencia entre procesamiento de archivos y procesamiento de base de datos, así como de las ventajas de utilizar una base de datos, las tendencias nacionales e internacionales, se aborda acerca de las características de los sistemas gestores de base de datos (SGBD). Además, se realizó un estudio sobre las diferentes herramientas que se utilizan para gestionar y modelar las bases de datos (BD), haciendo énfasis en las que se usan durante el desarrollo de la base de datos de la Dirección de Transporte y Seguro del MINCEX.

1.2 ORIGEN DE LAS BASES DE DATOS.

El diseño de una base de datos es fundamental para el éxito de todo tipo de producto. Estas han existido desde los comienzos de las civilizaciones y de echo define a las mismas. En sus comienzos el hombre necesitaba guardar los conocimientos y para ello lo escribía y lo cataloga usando índices de papel, esto llevo a que el libro fuese la primera base de datos. Estos no era bases de datos electrónicas sin embargo servían para el mismo propósito. Hoy cuando se trabaja con o se crean base de datos electrónicas es difícil pensar en estos objetos los cuales definieron a la civilización hace varios miles de años.

Las base de datos sirven para solucionar algunos problemas, por ejemplo, compensan el hecho de una memoria terrible, si se olvida algo simplemente se busca, compensan también el hecho de que los humanos sólo puede analizar pequeñas piezas de información, las BD agrupan los datos introducido por el usuario, además, permiten compartir hechos y eventualmente conocimiento.

El nacimiento de las Bases de Datos relacionales:

Cuando se usa este termino, se refiere a bases de datos electrónicas más estructuradas tales como Relacionales, Objetos, OLAP o espaciales. Estas tienen su origen hacia el año de 1970 cuando E.F. Codd de la compañía IBM introdujo la idea de un modelo relacional de Bases de datos en un documento titulado “A Relational Model of data for Large Shared Banks”, antes de eso la mayoría de bases de datos estaban basadas en un modelo de red o una simple estructura de archivo plano.

Este modelo relacional se basaba en una teoría de conjuntos matemáticos que servía para distintos propósitos: estos fueron; abstraer la representación de datos de su almacenaje físico y manipularlos, minimizar la redundancia de datos, dividiéndolos en distintos grupos no duplicados que pueden ser relacionados en un infinito número de maneras para producir un infinito número de representaciones, incrementar la consistencia de datos, por ejemplo si se cambia el nombre de un cliente, este cambiará en todos los reportes que se hagan acerca de ese cliente, porque esa parte es guardada en una sola parte pero genera varias vistas o representaciones del dato.

Más tarde IBM desarrolla un lenguaje llamado SQL (Lenguaje estructurado de consultas; en español), este tenía como objetivo generar reportes y actualizar datos en este nuevo modelo relacional. El Sistema R de IBM nació de este trabajo, pero fue ignorado por IBM, y más tarde Oracle lanza una versión comercial de BD basada en la teoría relacional de Codd, y el Berkely Ingres. El proyecto de investigación Berkely Ingres fue también comenzado por este tiempo; el mismo

consistía en extender el modelo relacional para que trabajara con modelos más complejos de datos, muchos modelos de Objetos y Objetos relacionales tienen sus principios en Ingres. La aparición de estos modelos impulsó la creación de otros por ejemplo, Informix, Sybase y el proyecto Ingres dieron nacimiento al Postgres el cual agrega más características Orientadas a Objetos al modelo relacional, después se transformó en PostgreSQL.

La unión del modelo relacional y SQL fueron la base para otros modelos entre los cuales se puede mencionar: Bases de datos de Objetos, Objetos Relacionales, Espaciales, OLAP y XML entre otras.

El modelo de objetos fue desarrollado para compensar las deficiencias del Modelo Relacional en cuanto a la construcción de consultas complejas, y estructuras de datos sin tener que dividirlos en una estructura relacional de 2 dimensiones. Este modelo usa un lenguaje muy similar al SQL llamado OSQL (object SQL).

Las bases de datos espaciales fueron desarrolladas para relacionar datos en el espacio, proveen respuestas a preguntas tales como ¿Cuanto dinero se tiene que gastar en un radio de 15 kilómetros de un lugar específico? o ¿A cuántos kilómetros de distancia está el hospital más cerca de mi casa?

La mayoría de las bases de datos espaciales son sólo extensiones de las bases de datos relacionales usan un dialecto del SQL llamado Spatial Features Structured Query Language (SFSQL) el cual simplemente agrega funciones espaciales al SQL.

Las bases de datos OLAP (On-Line Analytical Processing Databases) fueron diseñadas para analizar datos en lugar de actualizarlos. Son usadas para manejar procesos de negocios basados en análisis estadísticos de datos. La principal característica de las bases de datos OLAP es la velocidad de las consultas y la multidimensionalidad. Tienen su propio dialecto de SQL específicamente diseñado para tratar con la multidimensionalidad de datos en OLAP.

Actualmente hay una tendencia creciente para proveer lógica de negocios con bases de datos. Inicialmente las bases de datos solo controlaban que tipo de datos se podía capturar en un campo, después se hicieron más sofisticadas y se hicieron mejoras como los triggers, actualización y borrado en cascada para manejar la inconsistencia entre las tablas. Ahora las BD están empezando a manejar lógica más compleja, en algunos manejadores se pueden escribir procedimientos de bases de datos en Java, VB, Perl y Python para manejar funciones de las BD. (Anzaldo, 2005)

DEFINICIÓN DE BASE DE DATOS (DB)

Una BD es un conjunto de datos que tiene las siguientes propiedades implícitas:

- Representa algún aspecto del mundo real, llamado mini mundo o universo de discurso.
- Las modificaciones del mini mundo se reflejan en la BD.
- Es un conjunto de datos lógicamente coherentes, con un cierto significado inherente. Una colección aleatoria de datos no puede considerarse propiamente una BD.
- Una BD se diseña, construye y puebla con datos para propósito específico. Está dirigida a un grupo de usuarios y tiene ciertas aplicaciones preconcebidas que interesan a distintos usuarios.

O sea, una BD tiene:

- Una fuente de la cual se derivan los datos.
- Cierta grado de interacción con los hechos del mundo real.
- Un público activamente interesado en el contenido de la BD.
- Su tamaño es variado.
- Debe ser posible buscar, obtener y actualizar los datos siempre que sea

necesario.

En resumen se puede decir que una **DB** es un conjunto de datos interrelacionados entre sí, almacenados de forma más o menos permanente en una computadora. O sea, una base de datos se puede definir como una colección de datos variable en el tiempo, un sistema de archivos electrónicos. (García, 2005)

1.2.1 SISTEMA GESTOR DE BASE DE DATOS (SGBD).

Es un conjunto de programas que permiten la creación, la actualización, el procesamiento y la consulta de los datos almacenados en una o varias bases de datos por uno o varios usuarios desde diferentes puntos de vista y a la vez, de forma segura y eficiente, por lo tanto se puede decir que un SGBD es un software de propósito general que facilita el proceso de definir, construir y manipular la base de datos para diversas aplicaciones. Estos pueden ser de propósito general o específico. (García, 2005)

CARACTERÍSTICAS DE LOS SISTEMAS GESTORES DE BASE DE DATOS (SGBD):

Las razones para la utilización de las bases de datos son varias, por ejemplo, estas no necesitan archivos de papel, los cuales ocupan mucho espacio, las máquinas pueden obtener y modificar datos con mucha mayor velocidad que un ser humano, de esta manera es posible satisfacer con rapidez consultas de casos particulares en un momento sin necesidad de búsquedas visuales o manuales que requieren mucho tiempo, se dispone de un momento de información preciso y al día. Por lo tanto, los SGBD han sido creados para cumplir con las siguientes características:

- La disminución de la redundancia: se define redundancia como la duplicación de datos. La duplicación de datos genera a su vez una duplicación del trabajo a la hora de mantenerlos y actualizarlos. Por tanto, las Bases de Datos al reducir la duplicación de datos, disminuyen el trabajo.

Es fundamental hacer copia de seguridad de la base de datos cada vez que esta quede actualizada. Si compensa duplicar datos para aumentar la velocidad de la base de datos en cuestión, se obtendrá como circunstancia la compensación de la redundancia de dichos datos. También puede darse que una duplicación de datos sea obligatoria por las circunstancias. Pero al ser posible siempre es mejor intentar evitar la redundancia.

- La compartición de datos: se habla de datos actuales, ya que al ser centralizados, se puede tener acceso a los datos con la última actualización en prácticamente tiempo real.
- La posibilidad de aplicar restricciones de seguridad: para mantener la seguridad a cerca del mantenimiento de los datos, los administradores de la Base de Datos, crean una jerarquía de acceso, que permitirá o prohibirá a los usuarios hacer una u otra acción sobre dicha base de datos.
- Cumplimiento de las restricciones de integridad: el SGBD ofrece recursos para definir y garantizar el cumplimiento de las restricciones de integridad.
- Integridad: en una base de datos se puede mantener una coherencia, no dejar que se introduzcan caracteres en un campo numérico, por ejemplo, esto se lleva a cabo mediante reglas de validación.
- Independencia de los datos: las aplicaciones que interactúan con la base de datos no se verán afectadas por los cambios en la estructura de almacenamiento y en la estrategia de acceso.
- Consistencia: en caso de que la redundancia no halla sido posible de eliminar, la información repetida debe actualizarse de forma coherente, lo que significa que todos los datos duplicados o repetidos en la base de datos se actualicen de forma simultánea.
- Copias de respaldo: todo SGBD debe poseer una manera de realizar copias de respaldo de la información que se encuentra almacenada en ellos.

- Facilidad de manipulación de la información: un SGBD debe tener la capacidad de buscar los datos de una forma rápida atendiendo a varios criterios, además de, permitir que todo usuario pueda hacer uso del mismo con facilidad.
- Control centralizado de la información: uno de los objetivos principales de los SGBD es que el mismo sea capaz de garantizar el control de la información de forma centralizada, permitiendo controlar los datos almacenados de manera sistemática y única, así como también el acceso a esta. Lo anterior implica que debe existir una persona o conjunto de personas que se responsabilicen por los datos operacionales, en este caso el administrador de base de datos, el mismo puede considerarse parte del SGBD y tiene como tareas fundamentales: administrar la estructura de la Base de Datos, administrar la actividad de los datos, administrar el Sistema Manejador de Base de Datos, establecer el Diccionario de Datos, asegurar la confiabilidad de la Base de Datos, confirmar la seguridad de la Base de Datos .

ARQUITECTURA DE LOS SISTEMAS GESTORES DE BASE DE DATOS (SGBD)

La Arquitectura de tres niveles propuesta por el grupo ANSI/SPARC está presente en gran número de SGBD, esta tiene como objetivo la separación entre las aplicaciones del usuario y la forma de los datos física.

- Nivel Interno: tiene un esquema interno que describe la estructura física de almacenamiento de la BD. El esquema interno emplea un modelo físico de los datos y describe todos los detalles para su almacenamiento, así como los caminos de acceso a la BD.
- Nivel Conceptual: tiene un esquema conceptual que describe la estructura de toda la BD para una comunidad de usuarios. Oculta detalles de la

estructura física de almacenamiento y se concentra en describir entidades, tipos de datos, vínculos, operaciones de los usuarios y restricciones. Se puede usar modelos de datos de alto nivel o uno de implementación.

- Nivel Externo o Vistas: incluye varios esquemas externos o vistas de usuarios. Cada uno describe la parte de la BD que interesa a un grupo de usuarios. Se puede usar el modelo de datos de alto nivel o uno de implementación.

LENGUAJES DEL SGBD.

Dentro de las características de los SGBD se puede hacer mención también de los lenguajes que utilizan, estos son los siguientes:

- Lenguaje de Definición de Datos (DDL): permite definir ambos esquemas. El SGBD contará con un compilador de DDL cuya función será procesar enunciados escritos en el DDL para implementar las descripciones de los elementos de los esquemas y almacenar la descripción del esquema en el catálogo del SGBD.
- Lenguaje de Definición de Almacenamiento (SDL): permite especificar el esquema interno. Las correspondencias entre los dos esquemas se pueden especificar en cualquiera de los lenguajes.
- Lenguaje de definición de Vistas (VDL): permite especificar las vistas del usuario y su correspondencia con el esquema conceptual.
- Lenguaje de Manipulación de Datos (DML): una vez compilados los esquemas de la BD y que en esta se han introducido datos, los usuarios requerirán algún mecanismo para manipularla. Las operaciones más comunes son: obtención, inserción, eliminación y modificación de los datos.

DIFERENCIA ENTRE PROCESAMIENTO DE ARCHIVOS Y PROCESAMIENTO DE BASE DE DATOS.

La diferencia principal entre un sistemas de procesamiento de archivos y un sistemas de procesamiento de base de datos radica en la manera de almacenar, recuperar y actualizar los datos; lo anterior se debe a que con los sistemas de procesamiento de archivos, como su nombre lo indica, los datos se guardan en diversos archivos y muchas veces dichos archivos son de diferente formato, ocasionando problemas como la redundancia que trae consigo almacenamiento y dificulta el acceso a los datos. Dicha redundancia también implica inconsistencias en lo que se tiene almacenado, en cambio un sistema de procesamiento de bases de datos mantiene los datos centralizados, todos en un mismo formato y la gestión de los mismos se desarrolla de una forma rápida.

1.3 TENDENCIAS NACIONALES E INTERNACIONALES.

Nacional: el desarrollo y uso de las bases de datos en Cuba ha ido en aumento, a partir de la utilización ordenada y masiva de las Tics en gran parte de las esferas de la sociedad cubana. La mayoría de los SGBD utilizan con mayor frecuencia el modelo relacional, el de red y el jerárquico, aunque en algunos más modernos se utilizan los orientados a objetos. Los SGBD más utilizados en Cuba son MySQL y Access, PostgreSQL y Oracle en algunos casos. A pesar de que Cuba no tiene acceso a los principales avances de la ciencia y la técnica a nivel mundial por estar sometida a un férreo bloqueo económico, el desarrollo de la informática, y con ella las BD, en el país se ha realizado de una manera constante. Es por ello que para el desarrollo de bases de datos se utilizó el gestor de bases de datos Access que viene con el paquete de Office de Windows como herramienta para la elaboración de las bases de datos de las instituciones cubanas, por su facilidad de uso y la poca complejidad que presenta, ejemplo de ello es:

Instituto Nacional de Recursos Hidráulicos (INRH) que cuenta con una base de

datos elaborada en Microsoft Access para recopilar los datos hidrológicos y de la calidad del agua.

Sin embargo ante las posibilidades que brinda al país el uso del software libre, se ha venido trabajando en una progresiva migración hacia los SGBD de código abierto como son el Postgres y MySQL, la principal razón es que estos presentan licencia de código abierto y están a disposición del público de forma gratuita.

Internacional: Desde la década de 1970 el uso de los ficheros para almacenar la información fue disminuyendo, puesto que surgieron las bases de datos (BD) y los sistemas de gestión de bases de datos (SGBD) y su uso se generalizó en todo el ámbito empresarial al reconocer su importancia y efectividad en el manejo de los datos. Debido al auge de la utilización de las BD han surgido nuevas aplicaciones para estas como son:

Bases de datos de multimedia: con dibujos, videos, sonidos. Bases de datos orientadas a objetos, bases de datos distribuida, sistemas de información geográfica: con mapas, datos del tiempo, imágenes de satélite, almacenes de datos (o Data Warehouses): analizan y extraen información útil para la toma de decisiones. Base de datos activas y de tiempo real: útiles en procesos de control industrial y fabricación.

1.4 ANÁLISIS DE LAS POSIBLES HERRAMIENTAS A UTILIZAR.

En el mundo actualmente existe una gran variedad de sistemas gestores de bases de datos (SGBD) los cuales poseen distintos tipos de licencias para su uso y distribución, así pues, teniendo en cuenta la política de la facultad, la cual está orientada al uso del software libre, se realizará un estudio solo de los posibles SGBD que cumplan con esta característica.

SGBD con licencia de software libre:

- SQLite: es un SGBD relacional, que está contenida en una relativamente pequeña biblioteca en C. Posee código fuente de dominio público libre de:

copiar, modificar, vender, publicar, usar, o distribuir el código original. Este sistema difiere de los SGBD cliente-servidor: la biblioteca SQLite se enlaza con el programa pasando a ser parte integral del mismo a través del cual realiza llamadas simples a subrutinas y funciones. El conjunto de la base de datos (definiciones, tablas, índices, y los propios datos), son guardados como un sólo fichero estándar en la máquina host.

- Apache Derby: un sub-proyecto de Apache PP, es una base de datos relacional de código abierto aplicado en su totalidad en Java y disponible bajo la licencia Apache, Versión 2.0. Está basado en los estándares de Java, JDBC, y SQL. Derby es fácil de instalar, implementar y utilizar. (Derby, 2007)
- Firebird: es una base de datos relacional que ofrece muchas características de SQL ANSI estándar y que funciona en Linux, Windows, MacOSX, y una variedad de plataformas UNIX. Ofrece una concurrencia excelente, alto rendimiento y un poderoso lenguaje de procedimientos almacenados y disparadores. Firebird es un proyecto independiente de programadores de C y C++, asesores técnicos y colaboradores que lo desarrollan. Basada en código fuente liberado por Inprise Corp (ahora conocida como Borland Software Corp) en el 2000. (Firebird., 2008)
- PostgreSQL: Es un SGBD relacional que incluye características como la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. Posee código abierto y licencia BSD.
- MySQL (versiones anteriores): SGBD relacional multiusuario, con más de 100 millones de copias y descargas en todo el mundo, lo cual la convierte en el SGBD más usado hasta el momento. En sus primeras versiones posee licencia GNU GPL a partir de la versión 3.23.19 para cualquier uso compatible con esta licencia, pero la versión actual ha sido privatizada. Su principal objetivo de diseño fue la velocidad, por lo que en este aspecto es superior a los demás sistemas de su tipo en el cual se incluye, PostgreSQL.

Consume pocos recursos tanto de CPU como de memoria, y posee muy buenas utilidades de administración (backup, recuperación de errores), y aunque se bloquee no suele perder información ni corromper los datos. (MySQL, 2008)

HIBERNATE

Otra de las herramientas a utilizar es Hibernate, cuyo objetivo principal es facilitar la persistencia de los datos Java en bases de datos relacionales y al mismo tiempo permite realizar consultas de estas bases de datos para obtener objetos. Este software se distribuye bajo la licencia GNU LGPL, por lo que cumple con la política de la facultad del uso de software libre para el desarrollo de aplicaciones informáticas.

LAS CARACTERÍSTICAS DE ESTA TECNOLOGÍA SON LAS SIGUIENTES:

- Soluciona el problema de la diferencia entre los dos modelos de datos coexistentes en una aplicación: el usado en la memoria de la computadora (orientación a objetos) y el usado en las bases de datos (modelo relacional).
- Es flexible en cuanto al esquema de tablas utilizado, para poder adaptarse a su uso sobre una base de datos ya existente.
- Tiene la funcionalidad de crear la base de datos a partir de la información disponible.
- Ofrece un lenguaje de consulta de datos llamado HQL (Hibernate Query Language) el mismo permite utilizar un lenguaje intermedio dependiendo de la base de datos que se utilice, y el dialecto será traducido al SQL dependiente de la base de datos de manera transparente y automática, y simplificando el código, además, posee una API para construir las consultas programáticamente, llamada de forma común "criteria".
- Hibernate para Java puede ser utilizado en aplicaciones Java

independientes o en aplicaciones Java EE(Java Enterprise Edition), mediante el componente Hibernate Annotations que implementa el estándar JPA, que es parte de esta plataforma.(Hibernate,2008)

FRAMEWORK: SPRING

Spring es un framework (marco de trabajo) distribuido de forma libre, su código es abierto, el consumo de recursos es mínimo lo cual indica que el costo de utilización de este por parte de la aplicación que se realiza y para el sistema y hardware que lo soporta es despreciable. Otra de las características de este framework es que las clases de una aplicación basada en Spring generalmente no dependen de las clases específicas de este, lo que lo califica como no intrusivo. Este framework está basado en la Programación Orientada a Aspectos y en la Inversión de control, lo primero se define como el paradigma de programación que permite una adecuada modularización de las aplicaciones y posibilitar una mejor separación de conceptos, lo segundo se puede decir que son un conjunto de técnicas y patrones de diseño de software que invierte el control del flujo de un sistema; el control es invertido en comparación con el modelo tradicional de interpretación expresado en una serie de llamadas consecutivas a procedimientos. Todas estas características indican que se puede clasificar como un framework ligero. (Jhonson, 2002)

MÓDULOS QUE POSEE SPRING

Este framework está compuesto por varios módulos los cuales brindan puntos de extensión con varios frameworks y bibliotecas de la plataforma J2EE tales como Hibernate, JMS, Jax-RPC entre otros. Estos módulos tienen como base el Core el cual contiene las principales funcionalidades de Spring, los restantes módulos se desarrollan sobre este y a su vez pueden servir de base a otros por ejemplo ORM y Web.

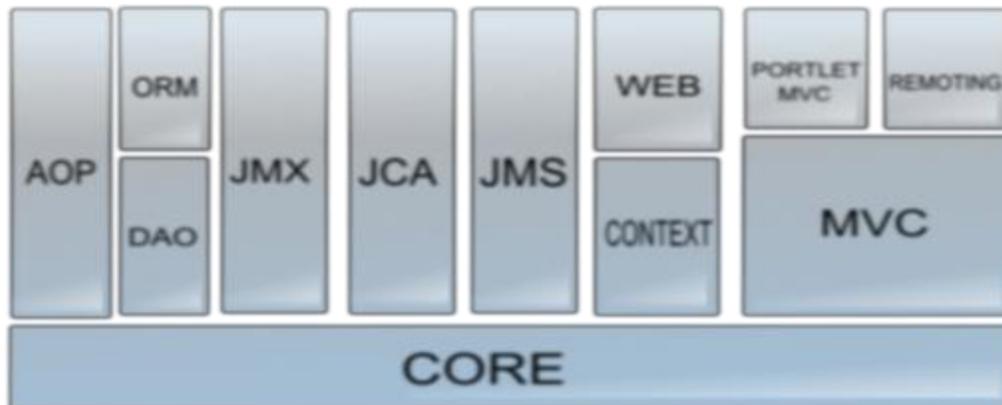


Figura No.1 Módulos del Framework Spring

1.5 VENTAJAS Y DESVENTAJAS DE POSTGRESQL:

VENTAJAS:

- Modelos de negocios más rentables con instalaciones a gran escala.
- No existe la posibilidad de ser auditado para verificar cumplimiento de licencia en ningún momento.
- Flexibilidad para hacer investigación y desarrollo sin necesidad de incurrir en costos adicionales de licenciamiento.
- El código fuente está disponible de forma gratuita, para que quien necesite extender o personalizar el programa pueda hacerlo sin costes.
- Posee un gran soporte brindado por la gran comunidad de usuarios que existe en el mundo que aportan experiencias y resultados obtenidos del uso del mismo.
- Buen sistema de seguridad mediante la gestión de usuarios, grupos de usuarios y contraseñas.
- Gran capacidad de almacenamiento.
- Buena escalabilidad ya que es capaz de ajustarse al número de CPU y a la cantidad de memoria disponible de forma óptima, soportando una mayor

cantidad de peticiones simultáneas a la base de datos de forma correcta.

- Soporta los tipos de datos, cláusulas, funciones y comandos de tipo estándar SQL92/SQL99 y extendidos propios de PostgreSQL.
- Diseñado para ambientes de alto volumen: Utilizando una estrategia de almacenamiento de filas llamada MVCC (Acceso concurrente multiversión, por sus siglas en inglés), consigue mejor respuesta en grandes volúmenes pues permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos.

DESVENTAJAS:

- En comparación con MySQL es más lento en inserciones y actualizaciones, ya que cuenta con cabeceras de intersección que no tiene MySQL.
- Soporte en línea: Hay foros oficiales, pero no hay una ayuda obligatoria.
- Consume más recursos que MySQL.
- La sintaxis de algunos de sus comandos o sentencias no es nada intuitiva.

ELECCIÓN DEL SGBD A UTILIZAR EN LA SOLUCIÓN PROPUESTA (POSTGRESQL):

En la actualidad es muy frecuente que las bases de datos comerciales sean instaladas en más servidores de lo que es permitido por la licencia. Es por ello que muchos proveedores comerciales ven esto como la principal fuente de incumplimiento de licencia. Con PostgreSQL, no ocurre este problema, puesto que no hay costo asociado a la licencia del software y es multiplataforma (en general funciona en todas las modernas basadas en Unix)

1.6 METODOLOGÍAS DE DESARROLLO DE SOFTWARE

Las Metodologías de Desarrollo de Software surgen dada la necesidad de utilizar un conjunto de técnicas, procedimientos, herramientas y soporte documental a la

hora de desarrollar un producto software. Estas tienen como objetivo guiar a los desarrolladores, pero la diferencia entre los requisitos de un software y otro pueden variar, esto a dado lugar a existencia de una amplia variedad de metodologías para la creación del software separadas en dos grupos:

- Las metodologías orientadas al control de los procesos, estableciendo rigurosamente las actividades a desarrollar, herramientas a utilizar y notaciones que se usarán. Estas metodologías son llamadas metodologías pesadas.
- Las metodologías orientadas a la interacción con el cliente y el desarrollo incremental del software, mostrando versiones parcialmente funcionales del software al cliente en intervalos cortos de tiempo, para que pueda evaluar y sugerir cambios en el producto según se va desarrollando. Estas son llamadas metodologías ligeras/ágiles.

PROGRAMACIÓN EXTREMA XP:

La Programación Extrema es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software. Promueve el trabajo en equipo, preocupándose en todo momento del aprendizaje de los desarrolladores y estableciendo un buen clima de trabajo. Este tipo de programación es la adecuada para los proyectos con requisitos imprecisos, muy cambiantes y con un riesgo técnico excesivo (ver Anexo 3, Fases de la metodología XP). (Henrik, 2009)

CARACTERÍSTICAS DE LA METODOLOGÍA XP

- Desarrollo iterativo e incremental: pequeñas mejoras, unas tras otras.
- Pruebas unitarias continuas, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión. Se aconseja escribir el código de la prueba antes de la codificación.

- Programación en parejas: se recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto. Se supone que la mayor calidad del código escrito de esta manera (el código es revisado y discutido mientras se escribe) es más importante que la posible pérdida de productividad inmediata.
- Frecuente integración del equipo de programación con el cliente o usuario.
- Corrección de todos los errores antes de añadir nueva funcionalidad.
- Refactorización del código, es decir, reescribir ciertas partes del código para aumentar su legibilidad y mantenibilidad pero sin modificar su comportamiento. Las pruebas han de garantizar que en la refactorización no se ha introducido ningún fallo.
- Propiedad del código compartida: en vez de dividir la responsabilidad en el desarrollo de cada módulo en grupos de trabajo distintos, este método promueve el que todo el personal pueda corregir y extender cualquier parte del proyecto. Las frecuentes pruebas de regresión garantizan que los posibles errores serán detectados.
- Simplicidad en el código: es la mejor manera de que las cosas funcionen. Cuando todo funcione se podrá añadir funcionalidad si es necesario. La programación extrema apuesta que es más sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado y quizás nunca utilizarlo. La simplicidad y la comunicación son extraordinariamente complementarias. Con más comunicación resulta más fácil identificar qué se debe y qué no se debe hacer. Mientras más simple es el sistema, menos tendrá que comunicar sobre este, lo que lleva a una comunicación más completa, especialmente si se puede reducir el equipo de programadores.

METODOLOGÍA SCRUM:

La metodología SCRUM es una forma de gestionar un equipo de manera que trabaje de forma eficiente y de tener siempre medidos los progresos, de forma que se sepa por dónde está el desarrollo del proyecto.

METODOLOGÍA A UTILIZAR (METODOLOGÍA SXP):

Esta metodología esta compuesta por las dos metodologías anteriormente descritas, la misma ofrece una estrategia tecnológica a partir de procedimientos ágiles los cuales permiten actualizar los procesos de software para el mejoramiento de la actividad productiva favoreciendo y fomentando el desarrollo de la creatividad, mediante el aumento del nivel de preocupación y responsabilidad de los miembros del equipo de desarrollo.

SXP está esencialmente creada para proyectos donde los equipos de trabajo son pequeños, los requisitos tiende a ser imprecisos, muy cambiantes, existe un alto riesgo técnico y se orienta a una entrega rápida de resultados y una alta flexibilidad. Todos los miembros trabajan juntos, siguen la misma dirección, tienen un objetivo claro, y permite seguir de forma clara el avance de las tareas a realizar, de esta forma los jefes de proyecto pueden ver diariamente cómo progresa el trabajo (ver Anexo 2). (EcuRed, 2012)

1.7 CONCLUSIONES PARCIALES:

En el capítulo se abordaron elementos importantes acerca de las herramientas que se utilizarán para dar solución al problema que presenta hoy la Dirección de Transporte y Seguro del MINCEX en cuanto a la persistencia y recuperación de los datos, se han introducido además, conceptos fundamentales para la comprensión del proceso de diseño y desarrollo de los componentes necesarios. Se hace mención de las tendencias nacionales e internacionales, así como de las características de la metodología de desarrollo de software que se utilizará, en este caso será SXP.

CAPÍTULO 2: DISEÑO Y ARQUITECTURA DE LA CAPA DE ACCESO A DATOS.

2.1 INTRODUCCIÓN

En este capítulo se describen los patrones de diseño a utilizar y se definen los artefactos: modelo entidad relación, modelo relacional. Además, se verán las reglas de transformación del modelo entidad relación y el modelo relacional, correspondiente a la base de datos en cuestión teniendo en cuenta los requisitos funcionales y no funcionales propuestos desde el negocio y la fase de requerimientos. Se realizará una breve descripción del concepto de normalización haciendo mención a cada una de las formas normales derivan del mismo.

2.2 DESCRIPCIÓN DE LOS REQUISITOS FUNCIONALES Y NO FUNCIONALES

De manera general los requisitos son funcionalidades o condiciones que los sistemas informáticos deben alcanzar o poseer. Los mismos son clasificados en dos tipos: requisitos funcionales (capacidades o condiciones que el sistema debe cumplir) y no funcionales (son propiedades que el producto debe tener, éstos ni describen información a guardar, ni funciones a realizar, a través de los mismos se debe obtener un producto atractivo, usable, rápido y confiable).

REQUISITOS FUNCIONALES (RF)

Nota: el gestor recoge las funcionalidades insertar, modificar, eliminar y mostrar.

- **R1** Gestionar proforma de control de empresas de la rama.
- **R2** Gestionar priorización de mercancías de importación.
- **R3** Gestionar información mensual de los niveles de flete marítimo.
- **R4** Gestionar resumen de certificado de actividad de contenedores del

MINCEX.

- **R5** Gestionar contenedores propios extraídos en el puerto.
- **R6** Gestionar plan de arribo de contenedores.
- **R7** Gestionar contenedores y situación de cargas viejas en puerto.
- **R8** Gestionar causas de contenedores entregados con más de 3 días.
- **R9** Gestionar Plan diario de extracción de contenedores.
- **R10** Gestionar Resultados de la operación de los buques.
- **R11** Gestionar Plan mensual y semanal de las operaciones de carga.
- **R12** Realizar reporte de proforma de control de empresas de la rama.
- **R13** Exportar reporte de proforma de control de empresas de la rama a formato pdf.
- **R14** Realizar reporte de priorización de mercancías de importación.
- **R15** Exportar reporte de priorización de mercancías de importación en formato pdf.
- **R16** Exporta información mensual de los niveles de flete marítimos a formato pdf.
- **R17** Realizar reporte de información mensual de los niveles de flete marítimos.
- **R18** Realizar reporte de resumen de certificado de actividad de contenedores por el MINCEX.
- **R19** Exportar reporte de resumen de certificado de actividad de contenedores por el MINCEX en formato pdf.
- **R20** Realizar reporte de contenedores propios extraídos en el puerto.
- **R21** Exportar reporte de contenedores propios extraídos en el puerto en

formato pdf.

- **R22** Realizar reporte de Plan de arribo de contenedores.
- **R23** Exportar reporte de Plan de arribo de contenedores en formato pdf.
- **R24** Realizar reporte de contenedores y situación de cargas viejas en puerto.
- **R25** Exportar reporte de contenedores y situación de cargas viejas en puerto en formato pdf.
- **R26** Realizar reporte de las causas de los contenedores entregados con más e 3 días.
- **R27** Exportar reporte de las causas de los contenedores entregados con más e 3 días en formato pdf.
- **R28** Realizar reporte de Plan diario de extracción de contenedores.
- **R29** Exportar reporte de Plan diario de extracción de contenedores en formato pdf.
- **R30** Realizar reporte de los Resultados de las operaciones con los buques.
- **R31** Exportar reporte de los Resultados de las operaciones con los buques en formato pdf.
- **R32** Realizar reporte de Plan mensual y semanal de las operaciones de carga.
- **R33** Exportar reporte de Plan mensual y semanal de las operaciones de carga en formato pdf.

REQUISITOS NO FUNCIONALES (RNF)

- El almacenamiento de los datos dependerá de los servidores de PostgreSQL 8.0 o superior y Apache; por lo que estos deben estar equipados con tarjeta de red, el servidor de base de datos debe tener 1Gb de RAM y 60 Gb de

disco duro (HD) como mínimo.

- Se podrán probar las funcionalidades de la BD desde el IDE desarrollo, mostrándose los resultados a nivel de consola.
- La BD independiente de la aplicación correrá sobre todo tipo de sistema operativo, aunque se recomienda tener instalado Linux.
- Fácil para el mantenimiento, de configuración sencilla y factible para los clientes.

2.3 MODELO DE ENTIDAD RELACIÓN (MER):

Este modelo se conoce además como modelo de datos, es un tipo de diagrama utilizado para el modelado de bases de datos. Tiene como objetivo representar relaciones que existen en la vida real de una forma lógica. Está compuesto principalmente por las entidades, los atributos, las interrelaciones y el dominio. El diseño del MER para la Dirección de Transporte y Seguro del MINCEX cuenta con 22 entidades interrelacionadas, las cuales tienen sus respectivos atributos y estos a su vez tienen sus respectivos dominios.

Entidades: Representa elementos u objetos de la vida real. Se representan por un rectángulo con el nombre de la entidad dentro. Se recomienda utilizar nombres en singular y que los mismos describan la entidad.

- Atributos: Son propiedades que poseen las entidades, por ejemplo un color, un nombre u otro dato que ofrezca información sobre la entidad. Se representa mediante elipses, conectados a la entidad mediante una línea.
- Dominio: Conjunto de valores posibles que puede tomar un atributo determinado.
- Interrelación: Relación, vínculo o correspondencia entre entidades.

Otros elementos que componen este tipo de diagrama son:

- Llave o clave: Atributo o conjunto de atributos de un artículo que define que

cada ocurrencia de artículo de la base de datos sea único.

- Entidad débil: Es el tipo de entidad que no puede existir por si sola, su existencia depende de otra entidad.
- Llave foránea: Conjunto de atributos común a dos entidades que sirve como relación entre las dos entidades. No es un atributo de la entidad relacionada, pero es la llave-primaria de la entidad con la cual ésta se relaciona.
- Llaves Secundarias: No puede identificar en forma única a un registro. Las llaves secundarias pueden usarse para seleccionar un grupo de registros que pertenecen a un conjunto.
- Tipo de Interrelación: Es una estructura genérica que describe un conjunto de interrelaciones de igual comportamiento. Se representa con mediante un rombo con el nombre dentro.

TIPOS DE ATRIBUTOS

- Simple o compuestos: Los atributos compuestos están formados por un conjunto de atributos, por tanto, es compuesto. Los simples, por el contrario, tienen una estructura modular.
- Multivaluados y mono valuados: Los atributos multivaluados pueden tener más de un valor para una entidad determinada. Mientras que los monos valuados tienen un solo valor.
- Derivados: Estos son atributos cuyo valor para una entidad puede ser obtenido o calculado a través del valor de otros atributos.

DIAGRAMA ENTIDAD RELACIÓN DE LA BASE DE DATOS DE LA DIRECCIÓN DE TRANSPORTE Y SEGURO DEL MINCEX.

La base de datos de la Dirección de Transporte y Seguro del MINCEX contiene 22 tablas (ver Anexo 1) en modelo físico, estas son:

- DTSmincex_Organismo: Esta clase guarda los datos de las empresas que se relacionan con cada organismo.
- DTSmincex_Producto: Esta clase contiene los datos de los productos que serán transportados dentro de los contenedores.
- DTSmincex_Empresa: Esta clase guarda los datos de las empresas que se relacionan con cada organismo.
- DTSmincex_Contrato: Esta clase representa el modelo o proforma del informe para el control de un contrato determinado, realizado por una empresa.
- DTSmincex_Contenedor: Esta entidad contiene los datos de los contenedores que contienen los productos que van a ser transportados o comercializados.
- DTSmincex_Tipo transporte: Esta clase guarda los datos del tipo de transporte usado por las empresas.
- DTSmincex_Aereo: Esta clase contiene los datos del tipo de transporte aéreo.
- DTSmincex_Terrestre: Esta clase contiene los datos del tipo de transporte terrestre.
- DTSmincex_Marítimo: Esta clase contiene los datos del tipo de transporte marítimo.
- DTSmincex_Tipo_Contenedor: Clase que contiene los tipos de contenedores según su situación, esta puede ser los contenedores que se encuentran en puerto pendiente de extracción, inhabilitados, habilitados o que hayan sido extraídos y estén pendientes a devolución.
- DTSmincex_Distribucion_por_plazos: Clase que representa los datos de la ubicación por plazos de los contenedores.

- DTSMincex_Terminales: Esta entidad contiene los datos de las terminales en las cuales se encuentran los contenedores.
- Plazos_entrega: Esta clase guarda los datos de los plazos de entrega
- DTSMincex_Vendedor: Esta entidad contiene los datos del vendedor que participa en el contrato.
- DTSMincex_Comprador: Esta entidad contiene los datos del comprador que participa en el contrato.
- DTSMincex_Pendiente_de_Ejecucion: Esta entidad representa los contratos que están pendientes a ejecución.
- DTSMincex_Disponible: Esta entidad contiene los datos de los contratos que se encuentran disponibles.
- DTSMincex_Buqueado: Esta entidad representa los datos de los contratos que ha sido buqueados.
- DTSMincex_Embarcado: Esta entidad representa los datos de los contratos que ha sido embarcados.
- DTSMincex_Despacho_Demora: La entidad en cuestión guarda los datos relacionados con las actividades de despacho de mercancías y las causas de las demoras de los mismos en caso de que estas ocurran.
- DTSMincex_Contenedores_Importacion: Esta entidad contiene los datos de los contenedores de importación que se encuentran en las terminales cubanas.
- DTSMincex_Cargas_Masivas_Generales: La entidad contiene los datos de las cargas masivas generales que son transportadas hacia las terminales y desde las terminales cubanas.

Seguidamente se muestra el diseño de algunas de estas tablas. El resto de las mismas se encuentran en el documento “Plantilla Modelo Canónico de Datos” junto

a su debida descripción.

Descripción de la entidad “DTSmincex_Organismo”.

Entidad:	DTSmincex_Organismo		
Descripción:	Esta entidad representa el los organismos, los cuales tienen una serie de empresas relacionadas, por ejemplo el ministerio del transporte.		
Relaciones:	Este tiene una relación de 1 a muchos con la entidad empresa DTSmincex_Empresa.		
Campos	Tipo de Dato	Tamaño	Descripción
id_organismo	Entero	10	Este campo contiene un número entero que representa el identificador del organismo.
nombre	Varchar	10	El campo nombre contiene el nombre del organismo.

Descripción de la entidad “DTSmincex_Producto”.

Entidad:	DTSmincex_Producto
Descripción:	Esta entidad contiene los datos de los productos que serán transportados dentro de los contenedores.
Relaciones:	La relación de esta tabla es con la tabla DTSmincex_Contenedor, puesto que un contenedor

	puede tener asociados varios productos.		
Campos	Tipo de Dato	Tamaño	Descripción
Id_producto	entero	10	Este campo contiene un número entero que va a identificar el producto.
pais_origen	varchar	10	Nombre del país del cual proviene el producto.
pais_destino	varchar	10	Nombre del país al cual va dirigido el producto.
observaciones	varchar	10	Apuntes que reflejan la situación de dicho producto.
nombre	varchar	10	Nombre del producto.

Descripción de la entidad “DTSmincex_Empresa”.

Entidad:	DTSmincex_Empresa		
Descripción:	Esta entidad guarda los datos de las empresas que se relacionan con cada organismo.		
Relaciones:	La relación de esta entidad es con las entidades DTSmincex_Contrato, DTSmincex_Contenedores.		
Campos	Tipo de Dato	Tamaño	Descripción
id_empresa	entero	10	Número que

			identifica a la empresa.
id_organismo	Entero	10	Identificador del organismo al cual pertenece la empresa.
nombre	varchar	10	Nombre de la empresa

Descripción de la entidad “DTSmincex_Contrato”.

Entidad:	DTSmincex_Contrato		
Descripción:	Esta entidad representa el modelo o proforma del informe para el control de un contrato determinado, realizado por una empresa.		
Relaciones:	Esta entidad está relacionada con las entidades DTSmincex_Empresa, DTSmincex_Tipo_transporte, DTSmincex_Vendedor, DTSmincex_Comprador, DTSmincex_Buqueado, DTSmincex_Disponible, DTSmincex_Embarcado, DTSmincex_Pendiente_de_ejecucion.		
Campos	Tipo de Dato	Tamaño	Descripción
id_contrato	entero	10	Número que identifica el contrato.

id_empresa	entero	10	Número que identifica a la empresa que realiza el contrato.
id_vendedor	varchar	10	Nombre de l vendedor que oferta el producto.
id_comprador	varchar	10	Nombre del comprador que compra el producto.
producto	varchar	10	Nombre del producto que se está comercializando.
condición	varchar	10	Condiciones bajo las que se realiza dicho contrato.
valor	float	10	Valor del producto.
cant_mercancia	entero	10	Cantidad de producto que se va a

			comercializar.
puerto_origen	varchar	10	Nombre del puerto en el cual se embarca el producto.
puerto_destino	varchar	10	Nombre en el puerto a donde se dirige el producto.
id_transporte	entero	10	Número que identifica el transporte a usar.
fechaBL	date	10	Fecha BL del contrato
flete_aplicado	float	10	Valor del flete aplicado
flete_indicativo_maritimo	float	10	Valor del flete indicativo marítimo
observaciones	varchar	10	Apuntes que reflejan características y condiciones del contrato.
fecha_contrato	varchar	10	Fecha en la que

			se realiza el contrato.
flete_total	entero	10	Monto correspondiente al flete total.
flete_cont	entero	10	Refiere al flete correspondiente a los contenedores.
cant_contenedor es	entero	10	Cantidad de contenedores.

Descripción de la entidad “DTSmincex_Contenedor”.

Entidad:	DTSmincex_Contenedor		
Descripción:	Esta entidad contiene lo datos de los contenedores que contienen los productos que van a ser transportados o comercializados.		
Relaciones:	Esta entidad se relaciona con las entidades DTSmincex_Empresa, DTSmincex_Contenedor_Importacion, DTSmincex_Terminales, DTSmincex_Tipo_contenedor, DTSmincex_Distribucion_por_plazos, DTSmincex_Plazos_de_Entrega, DTSmincex_Producto.		
Campos	Tipo de Dato	Tamaño	Descripción

id_contenedor	entero	10	Número que identifica el contenedor.
id_cont_importacion	entero	10	Número que identifica los contenedores de importación
id_terminales	entero	10	Número que identifica la terminal
id_plazo_de_entrega	entero	10	Rango de días establecidos para la entrega de dicho contenedor.
id_producto	entero	10	Número que identifica al producto que contiene.
id_empresa	entero	10	Número que identifica a la empresa a la que pertenece el contenedor.
id_distribución	entero	10	Número que indica los días asignados para la

			distribución de los contenedores.
lleno_o_vacio	boolean		Estado de los contenedores según mercancía que contiene.
habilitados_o_inhabilitados	boolean		Estado del contenedor según disponibilidad.
ubicación_zona_geografica	varchar	10	Ubicación de los contenedores según zona geográfica del país.
almacenes propios_o_enoc	varchar	10	Ubicación de los contenedores según se encuentren en almacenes propios o de Enoc.

2.4 NORMALIZACIÓN DE LAS BASES DE DATOS

En una base de datos se utilizan tablas bidimensionales para almacenar la información siendo esta la base del modelo relacional. Para lograr que estas tablas tengan la forma adecuada se realiza un proceso llamado normalización el cual

permite obtener relaciones de la forma plana bidimensional, esto evita que ocurran ciertas anomalías de actualización como por ejemplo la falta de consistencia de los datos.

Las tablas que se obtienen tras llevar a cabo este proceso no son más que matrices rectangulares que puede ser descritas de forma matemática y poseen propiedades tales como:

- Cada columna tiene un nombre propio.
- No se admiten filas duplicadas.
- Cada entrada de la tabla representa un ítem de datos, no hay grupos repetitivos.
- Todos los ítems de una columna son de un mismo tipo de datos (son de una misma clase).
- Tanto las filas como las columnas pueden ser consideradas en todo tipo de secuencia, independientemente del momento, sin afectar el contenido de información ni la semántica de la función que utilice la tabla. (Martin, 1989)

EXISTEN VARIAS FORMAS NORMALES (FN)

- Primera FN: Esta forma normal está referida a la apariencia de un tipo de registro. Cuando se está en presencia de esta, todas las ocurrencias de un tipo de registro deben contener el mismo número de campos. Esta forma excluye grupos y campos de repetición de variable. Solo permite valores atómicos; o lo que es lo mismo, prohíbe que un atributo tenga como valor a un conjunto de relaciones, prohíbe las relaciones dentro de las relaciones.

El diseño de esta base de datos cumple con la primera forma normal puesto que los atributos de cada una de las tablas son atómicos, no siendo ni multivaluados ni compuestos.

- Segunda FN: La primera condición que debe cumplir una tabla de la base de datos para encontrarse en segunda forma normal es, primeramente, estar en primera forma normal. Luego se puede decir que se está en presencia de la

segunda forma normal si todo atributo no primo A de R depende funcionalmente de manera total de la clave primaria de R. Dicho de otra manera, ninguno depende parcialmente de toda clave de R. Esta forma normal puede ser violada cuando un campo no-clave proporciona un hecho sobre un aparte de la clave.

- Tercera FN: Lo primero que debe cumplir una tabla para estar en 3FN es que se encuentre en 2FN, luego que cada atributo de la relación que no está contenido dentro de la llave primaria dependa solamente de la llave primaria y no de ningún otro atributo, o lo que es lo mismo, tiene como objetivo eliminar las dependencias transitivas. Una dependencia transitiva es aquella en la cual las columnas que no son llave son dependientes de otras columnas que tampoco son llave. Por lo se puede decir que una tabla está en tercera forma normal si y solo si los campos de la tabla dependen únicamente de la clave, dicho en otras palabras los campos de las tablas no dependen unos de otros.
- Cuarta FN: se está en presencia de esta forma normal cuando un tipo de registro no contiene dos o más hechos multivalor independientes acerca de una entidad. Además el registro debe satisfacer a la tercera FN.
- Quinta FN: se puede afirmar que un tipo de registro se encuentra en quinta FN cuando su contenido de información no puede ser reconstruido a partir de varios tipos de registro más pequeños.

2.5 CONCLUSIONES PARCIALES:

En este capítulo se han presentado y descrito en detalle algunas de las entidades que forman parte del diseño de la base de datos, así como a sus atributos y el tipo de dato que almacenan en cada caso (al resto de las entidades solo se hace mención, puesto que la descripción de estas se encuentra de manera más detallada en la Plantilla Modelo Canónico de Datos). Este diseño cubre los requisitos funcionales planteados para la base de datos expuestos también en este.

CAPÍTULO 3: IMPLEMENTACIÓN Y VALIDACIÓN DE LA CAPA DE ACCESO A DATOS.

3.1 INTRODUCCIÓN:

En este capítulo se aborda la implementación de los artefactos del diseño de la base de datos e integración con la aplicación en la capa de persistencia. Se plantea, el análisis hecho sobre la integridad de los datos, la redundancia y la seguridad de los datos dentro de una base de datos. Se expone una breve explicación del patrón de diseño DAO el cual es utilizado en la implementación de la capa de acceso a datos que se desarrolló. Además, se exponen y se analizan las pruebas realizadas para validar las funcionalidades que el sistema debe cumplir.

3.2 VALIDACIÓN TEÓRICA

El buen diseño de una base de datos para todo sistema es fundamental, para ello se debe tener en cuenta una serie de aspectos fundamentales que lo garanticen. Estos aspectos son, la normalización del diseño, la integridad y seguridad de los datos y el análisis de la redundancia de la información. Estos garantizan que a los datos no tenga acceso el personal no autorizado, además que, la información no sufra cambios indebidos por causa de un mal control de acciones.

INTEGRIDAD DE LOS DATOS

Cuando se habla de integridad se refiere a la completitud y la corrección en los datos de una base de datos. Al ser modificada la información de una base de datos, ya sea de forma directa o mediante el uso de sentencias SQL tales como: INSERT, UPDATE o DELETE. En las mismas se pueden presentar problemas con la integridad e inconsistencia de sus datos. En relación con este aspecto, PostgreSQL brinda confiabilidad, pues posee escritura adelantada de registros, lo cual evita la ocurrencia de pérdidas en caso de fallas de hardware, de energía o del sistema operativo que se utiliza. Por lo que se puede afirmar que una de las funciones principales del modelo de datos relacional es preservar la integridad de la

información almacenada en la mayor medida posible. Además, es conveniente que exista al menos una mínima redundancia para evitar inconsistencias en la misma. (García, y otros, 2008).

INTEGRIDAD DE LA CLAVE

Ningún atributo de una clave candidata puede tomar valores nulos, además, estos valores van a ser únicos para cada tupla. Esto se garantiza a través del uso de restricciones *PRIMARY KEY*.

INTEGRIDAD DEL DOMINIO

Cuando se habla de integridad de dominio se refiere a la integridad de los datos correspondiente a una entrada de datos para una columna determinada, o lo que es lo mismo, esta restringe los valores que puede tomar un atributo respecto a su dominio. Un ejemplo de estas restricciones son: las restricciones *CHECK*, *FOREIGN KEY*; definiciones *DEFAULT*; definiciones *NOT NULL* y reglas.

INTEGRIDAD REFERENCIAL

Este tipo de integridad garantiza que una entidad siempre se tenga relación con otras entidades válidas. Además, implica que los datos sean correctos, lo cual sugiere que no existan: duplicaciones, pérdida de datos o relaciones mal resueltas. Las base de datos relacionales poseen esta propiedad, pues el software gestor (en este caso el PostgreSQL) se encarga de su cumplimiento.

INTEGRIDAD SEMÁNTICA

Sobre los dominios, o sobre los atributos, se pueden definir restricciones al diseñar una base de datos, pero dichas restricciones pueden ser violadas por algunas operaciones. El sistema gestor de base de datos PostgreSQL permite restricciones de varios tipos como: de unicidad, no nulas, de actualización en cascada, entre otras, con la utilización de una declaración adecuada y con una gran flexibilidad. El gestor puede de una manera fácil controlar las transacciones y detectar violaciones de integridad, puesto que estas reglas de integridad son almacenadas en el diccionario como parte adicional de la descripción de los datos

(esto se conoce como “*control centralizado de la semántica*”), lo cual las hace muy entendibles y manejables, facilitando de esta forma su mantenimiento.

ANÁLISIS DE REDUNDANCIA

Un modelo completamente normalizado no debería contener redundancia alguna, si se introduce redundancia entonces se estaría ante el peligro de inconsistencia y sería necesario la utilización de triggers para asegura la consistencia cuando alguna acción con relación a la información sea ejecutada, dígase insertar, eliminar o actualizar. No obstante en ocasiones para lograr del sistema una rápida respuesta en cuanto a pedidos de información es necesaria la existencia de una redundancia controlada, lo cual se traduce en un menor tiempo de repuesta en operaciones muy usadas por la aplicación, por ejemplo, las sentencias SELECT. Los beneficios de introducir redundancia pueden ser impresionantes, pero debido al costo que ello implica, se debería usar estrategias como esta, solo sí esto diera como resultado una ventaja clara.

SEGURIDAD DE LOS DATOS

Cuando se analiza la seguridad de los datos se debe tener en cuenta diferentes fallos, los cuales pueden ser de tipo lógico, físico y humanos, los mismo tienden a alterar de forma indebida y descontrolada los datos de una base de datos. Con el fin de mantener estos datos seguros se deben trazar y seguir una serie de estrategias como por ejemplo: se deben crear vistas y procedimientos de almacenado los cuales van a ser llamados desde la aplicación evitando introducir directamente el código SQL en la consulta. Además, se deben hacer copias de resguardo o backups de la BD de forma periódica para poderla restablecer ante todo tipo de daño que sufra la misma. El servidor de base de datos deberá estar en un lugar protegido del clima, acceso de personal no autorizado, y de ataques indirectos como: la interrupción malintencionada de servicios (muchas veces es más fácil interrumpir el servicio que atacar directamente al sistema), dicho local deberá estar preferiblemente separado del resto de las máquinas donde trabajarán los usuarios.

3.2 CAPA DE ACCESO A DATOS

La capa de acceso a datos es la encargada de la persistencia de la información y la comunicación con el gestor de base de datos. En esta capa se encuentra los objetos que encapsulan la lógica de acceso a datos (DAO), cuya implementación extienden de las clases de soporte del framework Spring para el uso de este patrón usando el framework ORM Hibernate. En esta capa se encuentran las clases entidades que representan las clases del dominio.

PATRONES DE DISEÑO DE SOFTWARE

Antes que nada es bueno señalar que un patrón de diseño representa experiencia y conocimiento que ingenieros y programadores han adquirido en la solución de problemas comunes. Dichos patrones son una solución que ha sido probada para dar respuesta a un problema general de diseño, en un contexto determinado. Si al seguir alguno de estos patrones se añaden accesos directos o se incrementa la cantidad de código, disminuye la capacidad de comprensión del diseño o de una implementación. Si se regula la modularidad, se simplifica la descripción y se separan de forma satisfactoria los conceptos, entonces se incrementa la comprensión de lo que se está desarrollando.

En el presente trabajo de diploma estos patrones sirven de ayuda al diseño, mediante el cual se obtiene una solución de una forma rápida, segura y flexible. Además, proveen una especificación de los objetos y clases, sus relaciones y el objetivo del diseño beneficiando así la documentación y el mantenimiento del sistema, en este caso el Sistema Informático para los Procesos de Gestión de Información del MINCEX.

DATA ACCESS OBJECT (DAO):

Patrón de Diseño J2EE que centraliza todo el acceso a datos en una capa independiente, aislándolo del resto de la aplicación. Sus principales beneficios son que reduce la complejidad de los objetos de negocio al abstraerlos de la implementación real de la comunicación con la fuente de datos y que permite una migración más fácil de fuente de datos.

Un DAO realiza operaciones atómicas contra la base de datos, nunca son necesarias las transacciones. Por ejemplo, de ello se puede mencionar las búsquedas por clave, la creación, la actualización y el borrado de un registro, obtener todos los registros u otra operación que se vaya a realizar con cierta frecuencia. Por lo general, se crea un DAO por cada objeto que vaya a ser utilizado en la aplicación, en este trabajo específicamente lo que se hace es que se crea un DAO genérico nombrado **GenericDao.java** (ver figura No.2), el mismo posee todos los métodos necesarios que permiten la manipulación de los datos de la base de datos, de forma particular el objeto con el cual se trabaja en cada uno de los métodos va a ser un Entity de Hibernate, que se corresponde con las clases que se obtienen del mapeo realizado a la base de datos correspondiente y se encuentra en el paquete entity y este a su vez dentro del paquete model (ver figura No.3).

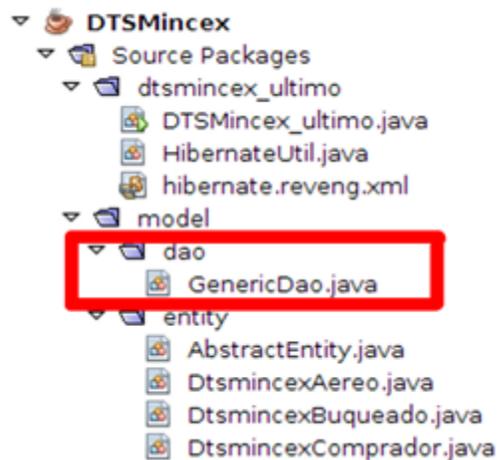


Figura No.2 Clase GenericDao.java

A continuación aparecen los métodos que posee la clase **GenericDao.java**, así como una breve explicación de la función de cada uno con vistas a un mejor entendimiento del código y la forma a través de la cual van a ser manipulada la información.

- Método SaveOrUpdate (salvar o actualizar): este método recibe como parámetro un objeto entity o lo que es lo mismo un objeto de tipo la clase correspondiente a la tabla de la base de datos donde se quiera insertar los datos,

pues sería bueno recordar que el uso de Hibernate permite crear clases de las tablas de las BD y además manipular los datos de las misma si tener la necesidad de trabajar directamente en la base de datos, y que cuando se habla de un objeto este no es más que los datos de una fila completa de las tablas que se encuentra en la base de datos. Habiendo entrado el dato, este se insertará en caso de que no esté, de lo contrario se actualizará la información.

```
public void saveOrUpdate (Object entity) {  
    getHibernateTemplate ().saveOrUpdate (entity);  
  
}
```

- Método Save (salvar): este método recibe un parámetro de igual tipo que el método que se explica arriba, solo que este realiza una única función y es la de insertar la información.

```
public void save (Object entity) {  
    getHibernateTemplate ().save (entity);  
  
}
```

- Método Update (actualizar): mediante la entrada por parámetros de un objeto entity, o sea de tipo la clase (tabla de la base de datos) la cual se quiere modificar sus datos se realiza la actualización de los datos específicos de ese objeto.

```
public void update (Object entity) {  
    getHibernateTemplate ().update (entity);  
}
```

- Método Delete (eliminar): este método toma como entrada también un objeto entity y lo elimina de la tabla correspondiente.

```
public void delete (Object entity) {  
    getHibernateTemplate ().delete (entity);  
}
```

- Método Find (buscar): a través de este método se obtiene una lista de los datos de la clase que se especifica en el parámetro de entrada.

```
public <T> List<T> find (Class<T> entityClass) {  
    return getHibernateTemplate ().loadAll (entityClass);  
}
```

- Método FindByPK (buscar por llave primaria): este método recibe como parámetros de entrada un objeto entity de tipo la clase o tabla donde se quiere buscar un datos determinado el cual tiene como identificador el otro parámetro de que es de tipo string.

```
public Object findByPK (Class entityName, String id) {
```

```

try {
    return getHibernateTemplate ().get (entityName, id);
} catch (Exception e) {
    return getHibernateTemplate ().get (entityName, Integer.parseInt (id));
}
}

```

- Método FindByParamsAndValues (buscar por varios parámetros y valores): este método permite la búsqueda de un objeto determinado dependiendo de los parámetros de búsqueda entrados.

```

public <T> List<T> findByParamsAndValues (Class<T> entityClass, String []
params, Object [] values) {

```

```

    List<T> list = new Array List<T> ();
    if (params.length > 0 && values.length > 0) {
        DetachedCriteria criteria = DetachedCriteria.forClass (entityClass);
        criteria.add (Restrictions.eq (params [0], values [0]));

        if (params.length > 1 && values.length > 1) {
            for (int i = 1; i < params.length; i++) {
                criteria.add (Restrictions.eq (params[i], values[i]));
            }
        }

        list = getHibernateTemplate ().findByCriteria (criteria);
    }

    return list;
}

```

- Método FindByParamAndValue (buscar por parámetro y valor): este método permite la búsqueda de un dato determinado teniendo como referencia la clase o tabla a la cual pertenece, el nombre de la columna en la que se encuentra y el valor que corresponde a ese campo en específico.

```

public <T> List<T> findByParamAndValue (Class<T> entityClass, String
param, Object value) {

```

```

    DetachedCriteria criteria = DetachedCriteria.forClass (entityClass);
    criteria.add (Restrictions.eq (param, value));

    return getHibernateTemplate ().findByCriteria (criteria);}

```

- Método FindByExample (buscar por ejemplo): este método se utiliza para

realizar una búsqueda de un objeto dado, de manera que se pueda saber si existe o no.

```
public AbstractEntity findByExample (Object exampleEntity) {  
    return (AbstractEntity) getHibernateTemplate ().findByExample  
(exampleEntity);  
}
```

- Método DeleteAll (eliminar todo): este método permite eliminar un conjunto de atributos de un objeto.

```
public <T> void deleteAll (Collection<T> collection) {  
    getHibernateTemplate ().deleteAll (collection) ;}
```

A continuación se muestran las clases que se obtienen al realizar el mapeo, a las cuales se ha hecho mención con anterioridad, y que están contenidas en el paquete entity.

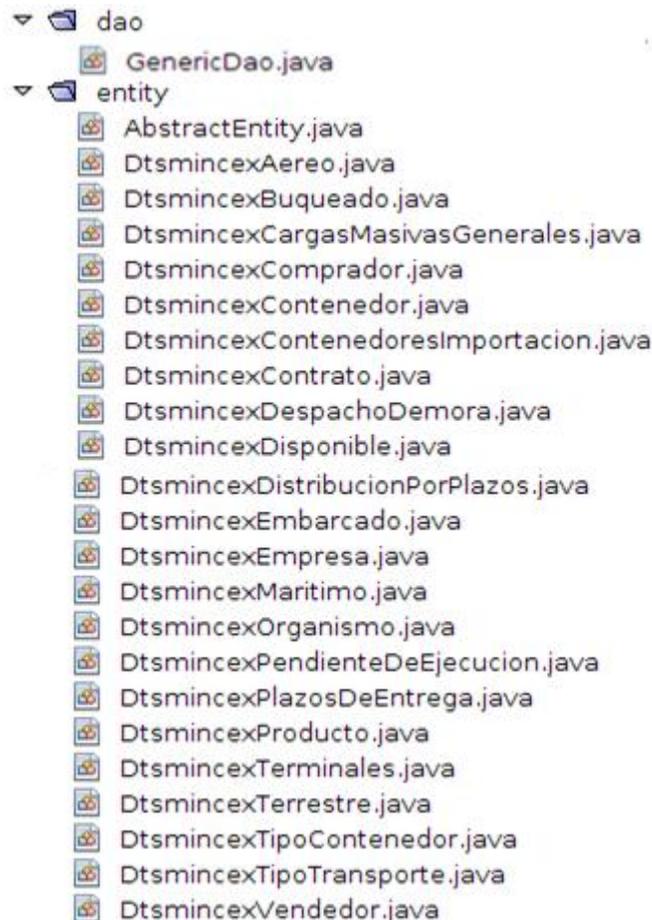


Figura No.3 Clases entity.

Estas contienen atributos, los cuales corresponden a las columnas de cada una de las tablas contenidas dentro de la base de datos; además de estos atributos las clases contienen los métodos necesarios para obtener y modificar cada uno de estos atributos.

Dentro del paquete entity se encuentra también la clase **AbstractEntity.java** de la cual heredan todas las demás que tengan dentro de sus atributos un identificador auto incrementable.

3.3 CASOS DE PRUEBAS

Los métodos implementados en la capa de acceso a datos fueron sometidos a pruebas unitarias, con el objetivo de verificar que funcionasen de la forma correcta en el módulo de la Dirección de Transporte y Seguro. Se realizaron nueve Casos de Pruebas, uno para cada método que fue implementado en la clase GenericDAO. A continuación aparecen una serie de tablas donde se reflejan los aspectos que se tuvieron en cuenta para la realización de esta prueba a los métodos para la persistencia de objetos.

Método:	save	
Condiciones:	Debe existir el objeto de la entidad que se desee salvar.	
Objetivo de la Prueba	Resultado Esperado	Resultado de la Prueba
Insertar en la BD el objeto que fue pasado por parámetro.	Se espera que el objeto pasado por parámetro se almacene en la tabla correspondiente a la BD.	El objeto pasado por parámetro se almacenó satisfactoriamente.

Tabla 1. Prueba al Método save.

Método:	saveOrUpdate	
Condiciones:	Debe existir el objeto de la entidad que se desea salvar o actualizar.	
Objetivo de la Prueba	Resultado Esperado	Resultado de la Prueba

Insertar o actualizar en la BD el objeto que fue pasado por parámetro.	Se espera que no existir los objetos en la base de datos sean guardados y si existen entonces actualizarlos en las tablas y columnas correspondientes en la base de datos.	Los objetos fueron actualizados correctamente en la base de datos.
---	--	--

Tabla 2. Prueba al Método saveOrUpdate.

Método:	find	
Condiciones:	Se le debe especificar la entidad que se desea buscar.	
Objetivo de la Prueba	Resultado Esperado	Resultado de la Prueba
Obtener el listado de la entidad especificada.	Se espera obtener una lista con todos los registros existentes en la entidad especificada.	Fue obtenido satisfactoriamente la lista con los registros de la entidad especificada.

Tabla 3. Prueba al Método find.

Método:	findByPK	
Condiciones:	Se le debe especificar la entidad y el valor de la llave primaria serializable.	
Objetivo de la Prueba	Resultado Esperado	Resultado de la Prueba
Obtener el objeto de la entidad especificada al que le corresponde la llave primaria indicada.	Se espera obtener el objeto que le corresponde la llave primaria indicada.	Fue obtenido el objeto especificado.

Tabla 4. Prueba al Método findByPK.

Método:	findByParamAndValue	
Condiciones:	Se le debe especificar la entidad, el parámetro y el valor por el que se desea buscar.	
Objetivo de la Prueba	Resultado Esperado	Resultado de la Prueba

Obtener el objeto de la entidad especificada que cumpla que el valor del parámetro se igual al especificado.	Se espera obtener el objeto cuyo valor del parámetro que se va a buscar sea igual al especificado.	Se obtuvo el objeto que presenta dicho parámetro con el valor indicado.
---	--	---

Tabla 5. Prueba al Método findByParamAndValue.

Método:	findByParamsAndValues	
Condiciones:	En la entidad debe existir dichos parámetros	
Objetivo de la Prueba	Resultado Esperado	Resultado de la Prueba
Obtener el objeto de la entidad especificada que cumpla que el valor de los parámetros se igual a los especificados.	Se espera obtener el objeto cuyos valores de los parámetros que se va a buscar sean iguales a los especificados.	Se obtuvo el objeto que presenta los parámetros con los valores indicados.

Tabla 6. Prueba al Método findByParamsAndValues.

Método:	update	
Condiciones:	El objeto de la entidad que se desea actualizar debe existir.	
Objetivo de la Prueba	Resultado Esperado	Resultado de la Prueba
Actualizar los atributos del objeto de la entidad especificada.	Se espera que los atributos del objeto sean actualizados.	Los atributos del objeto fueron actualizados.

Tabla 7. Prueba al Método update.

Método:	delete	
Condiciones:	El objeto de la entidad especificada debe existir.	
Objetivo de la Prueba	Resultado Esperado	Resultado de la Prueba
Eliminar el objeto de la entidad especificada.	Se debe eliminar el objeto de la entidad.	El objeto de la entidad fue eliminado.

Tabla 8. Prueba al Método delete.

Método:	deleteAll	
Condiciones:	En la entidad deben existir al menos uno o varios objetos.	
Objetivo de la Prueba	Resultado Esperado	Resultado de la Prueba
Eliminar todos los objetos de la entidad especificada.	Se espera eliminar todos los objetos de la entidad.	Fueron eliminados los objetos de la entidad especificada.

Tabla 9. Prueba al Método deleteAll.

3.4 RESULTADOS Y FUNCIONALIDADES OBTENIDOS

RESULTADOS

Después de analizar el resultado de las pruebas y finalizado este trabajo se tiene como resultado la capa de acceso a datos del módulo de la dirección de Transporte y Seguro en su versión 1.0. Este módulo forma parte del Sistema Informático para el Proceso de Gestión de Información del MINCEX. El mismo permite la gestión de la información que es manipulada en esta dirección, optimizando la manera en que se realizaban estas actividades anteriormente.

FUNCIONALIDADES

Algunas de las principales funcionalidades que posee dicho módulo son mencionadas a continuación y responden a los requisitos funcionales definidos, permitiendo:

- Gestionar los datos correspondientes a la proforma de los contratos realizados por las empresas relacionadas con cada organismo.
- Gestionar la priorización de mercancías de importación, mediante la inserción, modificación o la búsqueda.
- Realizar gestión de la información mensual de los niveles de fletes marítimos.
- Gestionar Resumen de certificado de actividad de contenedores por el

MINCEX.

- Permitir que se gestione la información de los contenedores propios extraídos en el puerto.
- Gestionar Plan de Arribo de Contenedores.
- Realizar reporte de contenedores y situación de cargas viejas en puerto.
- Obtener el reporte del plan mensual y semanal de las operaciones de carga.
- Realizar reporte de Plan diario de extracción de contenedores.

3.5 CONCLUSIONES PARCIALES:

Después de realizadas las pruebas pertinentes al sistema, en este caso al módulo correspondiente a la Dirección de Transporte y Seguro del MINCEX, y analizando los resultados que han sido obtenidos, se puede ver claramente que han sido implementadas las funcionalidades que dan respuesta a todos los requerimientos planteados en un inicio. Esto permite que la información esté centralizada, y que se pueda recuperar cuando se desee.

CONCLUSIONES GENERALES

Una vez culminado el presente trabajo de diploma, se obtuvo como resultado la base de datos y la capa de acceso a datos correspondiente al módulo de la Dirección de Transporte y Seguro del Ministerio de Comercio Exterior y la Inversión Extranjera. Tomando como base el diseño metodológico de la investigación, se arribó a las siguientes conclusiones:

- Mediante los fundamentos teóricos metodológicos se hizo un estudio del arte para identificar soluciones a fines al resultado de la investigación y se describieron las herramientas y tecnologías para el desarrollo.
- Se implementó la BD y la capa de acceso a datos, teniendo como premisa fundamental los requisitos identificados en la Dirección de Transporte y Seguro.
- Para comprobar la viabilidad del sistema se hicieron pruebas unitarias a la base de datos, arribando estas resultados satisfactorios.

RECOMENDACIONES

Después de haber cumplido los objetivos propuestos, y teniendo en cuenta el resultado obtenido de las pruebas realizadas, se recomienda:

- Utilizar el Sistema Gestor de Base de Datos PostgreSQL para gestionar la información en otros sistemas.
- Utilizar las herramientas seleccionadas en el presente trabajo para el manejo de la base de datos realizada.
- Para garantizar el soporte de la base de datos, se recomienda documentar todo el proceso de diseño con un manual de ayuda para los usuarios. Así como, la posibilidad de emitir sus quejas y sugerencias a los desarrolladores de la herramienta, ya sea por correo o por teléfono. Se debe realizar mantenimiento al BD y darle solución a todo tipo de problema que surja en la misma.

REFERENCIAS BIBLIOGRÁFICAS

- **Anzaldo, Juan. 2005.** Breve Historia de las Bases de Datos. [En línea] 2005.
- **2002.** Documentación de PostgreSQL. [En línea] 2002. <http://www.postgresql.org/docs>.
- **2009.** Documentación UPO / porque utilizar bases de datos. [En línea] 2009. <http://docupo.pbworks.com>.
- **Firebird. 2008.** Firebird. [En línea] 2008. <http://www.firebird.com.mx/modules/news/>.
- **García, Yissell María Castro y Villar, Roberto Llerena. 2008.** Propuesta de diseño para la base de datos de Akademos 2.0. 2008.
- **Henrik, K. 2009.** SCRUM y XP.ed. [En línea] 2009. <http://infoq.com/minibooks/scrum-xp-fromthetrenches>.
- **2008.** Hibernate. [En línea] 2008. <https://forja.rediris.es/docman/>.
- **2005.** Metodologías de desarrollo del software. [En línea] 2005. <http://www.willydev.net/descargas/prev/TodoAgil.pdf>.
- **Peñalver, Romero. 2008.** *GM Trabajo de diploma: Metodología ágil para proyecto de software libre*. Ciudad de La Habana: Universidad de las Ciencias Informáticas, 2008.
- **Derby, Apache. 2007.** Apache Derby. [En línea] 21 de mayo de 2007. <http://db.apache.org/derby/>.
- **EEM System. 2009.** PostgreSQL EEM System. [En línea] 19 de enero de 2009. <http://eemsystems.com/web/psql>.
- **MySQL. 2008.** MySQL. About MySQL. MySQL. [En línea] 18 de enero de 2008. <http://www.mysql.com/about/>.
- **SYSTEMS, EEM. 2008.** Postgres EEM SYSTEMS. [En línea] 18 de enero de 2008. <http://eemsystems.com/web/psql>.

BIBLIOGRAFÍAS

- **Alcalde, Alejandro Lino Carralero. 2010.** Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas. *Diseño e implementación de la base de datos del Sistema Unificado de la Fuerza de Trabajo Calificada “GeForza”*. Ciudad de la Habana: s.n., 2010.
- **Andino, Henry Maturell. 2009.** Diseño de base de datos para el Sistema de Producción de la Empresa de Perforación y Extracción de Petróleo de Occidente (EPEPO. Ciudad de la Habana: s.n., 2009.
- **Ballester, Lidier González. 2009.** Trabajo de diploma para optar por el título de Ing. en Ciencias informáticas. *Diseño de una base de datos para el control de los RRHH en los polos productivos de la facultad 9*. Ciudad de la Habana: s.n., 2009.
- **Claudia Deco, Cristina Bender. 2010.** *Tendencias actuales de investigación en Bases de Datos*. 2010.
- **Cuesta, Ernesto Alejandro Figueredo. 2011.** Diseño de la base de datos del procedimiento Ordinario de la materia Penal del proyecto Tribunales Populares Cubanos. Ciudad de la Habana: s.n., 2011.
- **Elian Luis Fernández de la Pera, Danis López Naranjo. 2007.** Trabajo para optar por el título de Ing. en Ciencias Informáticas. “DESARROLLO DE LA CAPA DE ACCESO A DATOS PARA LOS MÓDULOS DE ADMINISTRACIÓN Y NOMENCLADORES DEL SISTEMA DE GESTIÓN DE INVENTARIO DE ALMACEN. Ciudad de la Habana: s.n., 2007.
- **Martínez, Sisley Jiménez. 2011.** Trabajo de diploma para optar por el título de Ing. de las Ciencias Informáticas. *Diseño e implementación de la base de datos para el subsistema de Planificación material y financiera, del paquete de aplicaciones ERP-Cuba Cedrux*. Ciudad de la Habana: s.n., 2011.
- **Ortiz, Vladimir Creagh. 2010.** Diseño e implementación de la base de datos para generar una Base de Conocimiento de Patrones”. Ciudad de la Habana:

s.n., 2010.

- **Perón, Yusniel Yanes. 2009.** Trabajo de diploma para optar por el título de Ing. de las Ciencias Informáticas. Ciudad de la Habana: s.n., 2009.
- **Rodríguez, Héctor Julio. 2006.** Historia de las bases de datos en Ciencias de la Información. [En línea] 2006. <http://hjr227.googlepages.com/home>.
- **Rodríguez, Raúl Mató. 2009.** Trabajo de Diploma para optar por el título de Ing. de las Ciencias Informáticas. *Diseño de Bases de Datos para la Empresa de Gas Manufacturado*. Ciudad de la Habana: s.n., 2009.
- **Velthuis, Mario Piattini. 2000.** Líneas de evolución de las bases de datos. España: Universidad de Castilla-La Mancha, 2000.
- **Wilmar Barban Otaño, Danoy Alonso Llerena. 2009.** Trabajo de Diploma para optar por el título de Ing. en Ciencias Informáticas. *Sistema Informático para la Gestión de Información de los Recursos Materiales y Programas-Proyectos de las Coordinaciones Regionales de Prevención del Delito*. Ciudad de la Habana: s.n., 2009.
- **Silberschatz, A, Korth, H y Sudarshan., S. 2002.** Fundamentos de bases de datos (4ª ed.). Madrid: s.n., 2002.
- **Yenisleidy Rodríguez Martínez, Madisleidy Casanova Hernández. 2011.** Trabajo para optar por el título de Ing. de las Ciencias Informáticas. *Diseño e implementación de una herramienta web métrica libre para el análisis personalizado de sitios webs para GEWEB*. Ciudad de la Habana: s.n., 2011.

GLOSARIO DE TÉRMINOS

- **Base de Datos:** Desde el punto de vista informático, la base de datos es un sistema formado por un conjunto de datos almacenados en discos que permiten el acceso directo a ellos y un conjunto de programas que manipulen ese conjunto de datos.
- **Base de Datos Relacional:** Es una base de datos en donde todos los datos visibles al usuario están organizados estrictamente como tablas de valores, y en donde todas las operaciones de la base de datos se realizan sobre estas tablas. Estas bases de datos son percibidas por los usuarios como una colección de relaciones normalizadas de diversos grados que varían con el tiempo.
- **DAO:** Patrón de Diseño J2EE que centraliza todo el acceso a datos en una capa independiente, aislándolo del resto de la aplicación.
- **Entidad:** Se refiere a cualquier concepto del mundo real con una existencia independiente.
- **Framework:** Un *framework* denota un conjunto de objetos que definen un diseño abstracto para solucionar un conjunto de problemas relacionados. Puede incluir programas, bibliotecas de objetos o lenguaje interpretado, por lo que su uso facilita la elaboración de sistemas informáticos.
- **Hibernate:** Objetivo principal es facilitar la persistencia de los datos Java en bases de datos relacionales y al mismo tiempo permite realizar consultas de estas bases de datos para obtener objetos. Este software se distribuye bajo la licencia GNU LGPL.
- **Integridad:** En términos de datos se refiere a la corrección y completitud de los datos en una base de datos.
- **Licencia GPL:** La Licencia Pública General de GNU o más conocida por su nombre en inglés GNU General Public License o simplemente su acrónimo del inglés GNU GPL, es una licencia creada por la Fundación de Software

Libre a mediados de los 80, y está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el *software* cubierto por esta licencia es *software* libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

- **Llaves foráneas o foreign key:** Expresan relaciones entre objetos representados, incluyendo en el esquema de una relación atributos de otra. Utilizado para relacionar tablas.
- **Llave primaria o primary key:** Conjunto de atributos de su esquema que son elegidos para servir de identificador unívoco de sus tuplas.
- **Modelo entidad-relación:** Es una técnica para el modelado de datos utilizando diagramas de entidad-relación.
- **Normalización de bases de datos:** Consiste en aplicar una serie de reglas a las relaciones obtenidas tras el paso del modelo entidad-relación al modelo relacional. Se usa para evitar la redundancia de los datos, problemas de actualización de los datos en las tablas, proteger la integridad de los datos.
- **Patrones:** Antes que nada es bueno señalar que un patrón de diseño representa experiencia y conocimiento que ingenieros y programadores han adquirido en la solución de problemas comunes. Dichos patrones son una solución que ha sido probada para dar respuesta a un problema general de diseño, en un contexto determinado.
- **Query:** Consiste en una cadena de consulta, normalmente se utilizan para: insertar, actualizar o editar valores de la base de datos.
- **Spring:** Framework distribuido de forma libre, su código es abierto, el consumo de recursos es mínimo lo cual indica que el costo de utilización de este por parte de la aplicación que se realiza y para el sistema y hardware que lo soporta es despreciable. Otra de las características de este framework es que las clases de una aplicación basada en Spring generalmente no dependen de las clases específicas de este, lo que lo califica como no intrusivo

- **Sistema Gestor de base de datos:** Es un tipo de *software* específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.
- **Servidor:** En informática, un servidor es un tipo de software que realiza ciertas tareas en nombre de los usuarios. El término servidor ahora también se utiliza para referirse al computador en el cual funciona ese software, una máquina cuyo propósito es proveer datos de modo que otras máquinas puedan utilizar esos datos.
- **Software libre:** Software que puede ser distribuido, modificado, redistribuido, copiado y usado libremente. Que un software sea libre no quiere decir que sea gratuito, error que viene de la traducción Free Software, pues free en inglés significa lo mismo libre que gratuito.
- **Tabla:** En las bases de datos, una tabla significa el tipo de modelado de los datos, donde se guardan los datos recolectados por su programa. Su estructura general se asemeja a la vista general de un programa de hoja de cálculo.
- **Tupla:** Es la representación de una fila en una de las tablas que se está almacenando datos. Y las cuales serán llamadas por los administradores de base de datos en el tiempo de ejecución de un sistema.
- **Triggers:** Un trigger o disparador en una base de datos, es un procedimiento que se ejecuta cuando se cumple una condición establecida al realizar una operación de inserción (INSERT), actualización (UPDATE) o borrado (DELETE).

Anexo 2: Metodología de Desarrollo de Software a utilizar, SXP.



Anexo 3: Fases de la Metodología de Desarrollo de Software XP.

