



Universidad de las Ciencias Informáticas

Facultad Regional “Mártires de Artemisa”

Título: Base de Datos para el Módulo de la Dirección de Control de la Inversión Extranjera del Ministerio de Comercio Exterior (MINCEX).

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

Autor:

- Yoany Manso Salazar

Tutor:

- Ing. Yuneisy Barrios Pérez

Co Tutor:

- Javier Calvo Parapar

Declaración de autoría

Declaro que soy el único autor de este trabajo y autorizo al <nombre área> de la Universidad de las Ciencias Informáticas; así como a dicho centro para que hagan el uso que estimen pertinente con este trabajo.

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

Yoany Manso Salazar

Ing. Yuneisy Barrios Pérez

Firma del Autor

Firma del Tutor

Dedicatoria

A mis padres

Agradecimientos

A mi madre Niurka Salazar Días

*Que ha sido una luchadora incansable para que yo pueda realizar mi
sueño.*

A Mirna Martínez Labrador

La cual es una guía en todos los aspectos de mi vida.

A mi familia

Que siempre me ha brindado todo su apoyo.

A Wilfredo Martínez Becerra y Mirta Labrador Labrador

Que siempre me apoyaron con sus consejos.

A mi tutor Yuneisy Barrios Pérez

*Por el apoyo, dedicación y recomendaciones oportunas para la
realización de esta tesis.*

*A todas mis amistades que estuvieron conmigo durante mi carrera
universitaria gracias por su apoyo incondicional, en especial
Armando, Evelio, Daílyn, Lisandra.*

A todos gracias

Resumen

La investigación persigue diseñar un modelo de acceso a datos que garantice el correcto almacenamiento y seguridad de la información tratada en la Dirección del Control de la Inversión Extranjera del MINCEX, en cuanto a su organización, integridad, e independencia de los datos. Se obtiene como resultado una base de datos y la capa de acceso a datos. Aportando al sistema un modelo con alta estructuración y organización de los datos así como la rapidez de las consultas realizadas a la información mediante los métodos de acceso.

En el capítulo 1 se analiza los conceptos asociados al tema en cuestión, los cuales han sido la base que fundamenta la idea de este proyecto.

En el capítulo 2 se presentan diversas herramientas que se encuentran en el mercado, las cuales se tomaron como referencia para compararlas con las herramientas a utilizar en la solución propuesta y presentar sus beneficios. También se describen las características de la herramientas a utilizadas.

En el capítulo 3 expone como se implementó el proyecto, esto incluye el diseño de la Base de Datos y la implementación de la capa de acceso a datos, cómo interactúan con el módulo de la Dirección del Control de la Inversión Extranjera del Ministerio del Comercio Exterior. También presenta el tipo de pruebas realizadas en el proyecto.

ÍNDICE

Introducción.....	1
1. CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	7
1.1 HISTORIA DE LAS BASE DE DATOS	7
1.2 CONCEPTOS ASOCIADOS AL TEMA	8
1.3 TIPOS DE BASE DE DATOS	11
1.3.1 Selección del tipo de base de datos	14
1.4. HERRAMIENTAS DE MODELADO CASE	15
1.4.1 Herramientas CASE a utilizar	17
1.5 SISTEMAS GESTORES DE BASE DE DATOS	17
1.5.1 Sistema Gestor de Base de Datos a utilizar	19
1.6 HERRAMIENTAS DE PROGRAMACIÓN	19
1.6.1 Herramienta de programación a utilizar	20
1.7 FRAMEWORKS	22
1.7.1 Framework a utilizar	23
1.8 MODELO DE ACCESO A DATOS	23
CAPÍTULO 2: DISEÑO Y ARQUITECTURA	25
2.1 DISEÑO DE BASES DE DATOS RELACIONAL	25
2.2 INFORMACIÓN A GESTIONAR	26
2.3 REQUISITOS FUNCIONALES.....	26
2.4 METODOLOGÍA PARA EL DISEÑO DE BASES DE DATOS RELACIONAL	28
2.5 CAPA DE ACCESO A DATOS.....	28
2.5.1 Mapeo de la Base de Datos	29
2.5.2 Ingeniería Inversa	29
2.6 MODELO CONCEPTUAL Y MODELO LÓGICO.....	30
2.7 MODELO ENTIDAD RELACIÓN.....	31
CAPÍTULO 3: IMPLEMENTACIÓN Y VALIDACIÓN	35
3.1 VALIDACIÓN TEÓRICA DEL DISEÑO DE LA BASE DE DATOS	35
3.2 VALIDACIÓN DE LA CAPA DE ACCESO A DATOS	35
3.3 INTEGRIDAD DE LA BASE DE DATOS	36
3.4 NORMALIZACIÓN DE LA BASE DE DATOS	36
3.5 INTEGRACIÓN DE LOS MÓDULOS EXISTENTES.....	41
3.6 ANÁLISIS DE LA SEGURIDAD Y RECUPERACIÓN DE LA BASE DE DATOS	41
3.7 ESTRUCTURA DE PAQUETES EN LA CAPA DE ACCESO	42
3.8 MÉTODOS DE LA CLASE GENERICDAO	43
3.9 PRUEBAS.....	44
CONCLUSIONES GENERALES	50
RECOMENDACIONES	51
BIBLIOGRAFÍA REFERENCIADA	52
BIBLIOGRAFÍA CONSULTADA	55

Introducción

Con el surgimiento de las Tecnologías de la Información y Comunicaciones (TIC) la sociedad se ha desarrollado y evolucionado gradualmente en todos los aspectos de la vida. Las TIC han facilitado la creación y cambios de las corrientes públicas, a través del incremento de la producción y difusión de nuevas tecnologías, utilizadas diariamente por el hombre en sus organizaciones o empresas. Constituyendo una instancia de generación de conocimiento el cual permite un proceso dinámico, continuo, acumulativo; que reelabora y modifica las competencias en el mundo. Las TIC son incuestionables y están ahí, forman parte de la cultura tecnológica que nos rodea y con la que debemos convivir. Amplían las capacidades físicas y mentales así como las posibilidades de desarrollo social.

En el año 2002 se crea la Universidad de las Ciencias Informáticas, la cual surge como uno de los programas para contribuir en gran medida a la informatización de la sociedad cubana, con protagonismo en la participación de diversos proyectos. Desde su instauración la Universidad cuenta con varias facultades. Entre estas la Facultad Regional Mártires de Artemisa, la cual asume diversas tareas encaminadas a desarrollar disímiles proyectos, uno de ellos: Informatizar el Ministerio de Colaboración Económica y la Inversión Extranjera (MINCEX). Convirtiéndose este asunto en un eslabón vertebral para la investigación, donde se impone una solución inmediata por la importancia que contiene y el rol social que desempeña en la actualidad.

Al triunfo de la Revolución Cubana constituyó una preocupación del Gobierno revolucionario el bajo nivel de las divisas extranjeras. A partir de esta inquietud, atento a la defensa y desarrollo de la economía nacional, promulgó mediante la Ley 793 del 25 de abril de 1960 la creación del Banco para el Comercio Exterior de Cuba (BANCEC), con el objeto de contribuir y colaborar con la política de comercio internacional del Estado Revolucionario y la aplicación de las medidas sobre el comercio exterior adoptadas por el Banco Nacional. El BANCEC puede

considerarse una institución de transición para el establecimiento del monopolio estatal del comercio exterior de Cuba, que se logra finalmente con la creación del Ministerio del Comercio Exterior (MINCEX), el 23 de febrero de 1961 mediante la Ley 934.

El surgimiento del MINCEX constituyó un elemento esencial en la política económica del país y dio respuesta a la necesidad del estado cubano de asumir el manejo y control de las importaciones y exportaciones de la nación. El 2 de marzo de 2009 mediante el Decreto Ley N° 264; se dispone la creación del nuevo Ministerio del Comercio Exterior y la Inversión Extranjera como Organismo de la Administración Central del Estado, asignándole como objetivo y misión esencial la de preparar y proponer al Gobierno la política integral del Estado y del Gobierno en cuanto a la actividad de comercio exterior, la creación de empresas mixtas, la colaboración económica con otros países, organizaciones y asociaciones extranjeras, y las inversiones que se negocien. Una vez aprobada dicha política por éste, le corresponde al MINCEX el dirigir, ejecutar, coordinar y controlar el cumplimiento en todas las entidades a nivel del país, sobre la bases de las estrategias de desarrollo en ellas establecidas.

El MINCEX en su estructura cuenta con un Ministro, dos Viceministros Primeros, tres Viceministro y 23 direcciones de trabajo, una de ellas es la Dirección del Control de la Inversión Extranjera (DCIE) que es la encargada de velar el cumplimiento de las normas vigentes para la inversión extranjera además de chequear que se cumpla la ley implementada con estos fines.

Dicha dirección presenta problema con el almacenamiento y recuperación de la información referente a las Asociaciones Económicas Internacionales (AEI), las directivas, el plan y el real planificado. Las solicitudes que son registradas necesitan ser almacenadas también al igual que toda la información de la dirección, leyes, normativas, resoluciones, información de las empresas y organismos. Toda información que maneja la dirección es necesario tenerla de forma organizada,

almacenada y que permita la recuperación de la misma en caso de que esta se pierda. El cúmulo de información que se recibe por vía e-mail, en archivos Excel, por entrega personal o en formato duro, trae como consecuencia el trabajo de almacenamiento es más difícil pues este procedimiento de forma manual trae muchas desventajas para los trabajadores del DCIE como impedir la tenencia de registros actualizados a causa de la demora del llenado de los datos y errores de otro tipo que se cometen en el transcurso de tal proceso. Otro aspecto que incide en la morosidad para las gestiones propias de la recepción de datos es que las consultas se realizan a través de Access lo cual es muy engorroso y complicado por la cantidad de información guardada en lugares diferentes y en distintos formatos.

Por lo anteriormente expuesto se plantea el siguiente **Problema de Investigación:** ¿Cómo mejorar la persistencia y recuperación de los datos en la DCIE del MINCEX de forma que garantice la integridad y disponibilidad en el acceso y distribución de datos?

A partir del problema de la investigación esbozado se enmarca como **Objeto de estudio** los procesos de gestión de datos, a la vez se define como **Campo de acción** bases de datos relativas a los procesos de gestión de datos.

Para darle solución al problema de investigación se trazó como esencial el siguiente **Objetivo General:** Desarrollar la base de datos para el módulo de la Dirección de Control de la Inversión Extranjera del MINCEX que garantice la integridad y disponibilidad en el acceso y distribución de los datos.

Del objetivo general planteado se derivan los siguientes **objetivos específicos:**

- ✓ Elaborar la fundamentación teórica de la investigación.
- ✓ Diseñar la base de datos para el módulo de la DCIE del MINCEX.
- ✓ Desarrollo de la capa de acceso a datos para el módulo de la DCIE del

MINCEX.

- ✓ Validar los resultados obtenidos mediante pruebas funcionales.

Para desarrollar este trabajo se ha planteado la siguiente **idea a defender**: Con el desarrollo de la base de datos para el módulo de la Dirección de Control de la Inversión Extranjera del MINCEX se garantiza la integridad y disponibilidad en el acceso y distribución de los datos.

De ahí que se determine como **variable independiente** Base de Datos para la DCIE.

Y por otra parte como **variable dependiente** se señaló el mayor nivel de accesibilidad, disponibilidad, distribución e integridad de los datos

Porsupuesto se deriva una variable interviniente que corresponde a la confección de la base de datos (BD) para garantizar un correcto almacenamiento y recuperación de la información en la DCIE del MINCEX. Para de esta forma lograr una mayor efectividad en el cumplimiento de los objetivos.

Para dar solución a los objetivos específicos se trazaron las siguientes **tareas de la investigación**:

1. Establecimiento de los fundamentos teóricos y metodológicos para el desarrollo de las bases de datos
2. Especificación de las herramientas y tecnologías a utilizar para el desarrollo de la solución propuesta.
3. Diseño de la base de datos para el módulo de la DCIE del MINCEX.
4. Desarrollo de las capas de acceso a datos para el módulo de la DCIE del MINCEX.
5. Realización de pruebas funcionales para validar los resultados obtenidos.

Aporte Práctico:

- ✓ Base de datos para la Dirección de Control de la Inversión Extranjera del Ministerio del Comercio Exterior que garantiza la integridad, disponibilidad en el acceso y distribución de los datos.

En las diferentes etapas de este proyecto hemos empleado diversos métodos científicos tanto los teóricos como los empíricos los cuales nos respaldaron con un soporte tangible la realidad existente en el MINXEC. Seguimos un cronograma con diferentes análisis que posibilitaron las herramientas para la investigación:

Métodos teóricos

- **Análisis histórico-lógico:** Se investigó los antecedentes de la administración de datos, posibilitando el análisis de su trayectoria para una mejor comprensión de los mismos.
- **Analítico – sintético:** Este método pues permite dividir el fenómeno estudiar y unir las partes previamente analizadas. Se analizaron los aspectos más importantes para administrar una base de datos y posteriormente sintetizar y exponer los resultados obtenidos.
- **Inductivo-deductivo:** Se examinaron las características de los casos particulares, así como su comportamiento para de esta forma deducir conclusiones que conlleven a la verificación posterior en la práctica.
- **Modelación:** Para confeccionar el modelo de datos de la DCIE del MINCEX como resultado para un mejor entendimiento de la posible solución a implementar.

Métodos empíricos

- **Observación:** Permite conocer la realidad mediante la percepción directa de los objetos y fenómenos ocurridos en la DCIE del MINCEX.
- **Análisis - Documental:** Permitió el estudio y análisis de todos los documentos e informes de la dirección, para obtener un mejor entendimiento de los procesos de la misma.

La siguiente investigación está compuesta por introducción, 3 capítulos, conclusiones, recomendaciones, bibliografía y anexos a continuación especificamos una síntesis de los mismos.

Capítulo 1: En él se expone la fundamentación teórica. Se abordan los temas relacionados con las bases de datos, abarca las definiciones de conceptos, elaboración del estado del arte, definición de la metodología, herramientas, y tecnologías a usar para el desarrollo de la solución propuesta.

Capítulo 2: Un capítulo donde se muestra el diseño y arquitectura de la base de datos. Se describen los patrones de diseño a utilizar y se definen los artefactos correspondientes al rol diseñador de base de datos, como artefactos, modelo entidad relación, modelo relacional. Además, se distinguirán las reglas de transformación del modelo entidad relación y al modelo relacional.

Capítulo 3: Este último consta del Diseño y validación de la Base de Datos. Ejecución de los artefactos del diseño de las bases de datos e integración con la aplicación en la capa de persistencia.

1. Capítulo 1: Fundamentación teórica

Introducción

Este capítulo se dedica a fundamentar teóricamente el problema a resolver, en el mismo se harán análisis de los conceptos relacionados, estudio del arte, herramientas y tecnologías a utilizar con el tema en cuestión, enfatizando en la persistencia de datos.

1.1 Historia de las base de datos

La información, hoy en día, es considerada como un recurso vital para una organización, y contar con una base de dato adecuada para que la información esté disponible cuando se le requiera es muy importante. Las bases de datos han existido desde los comienzos de las civilizaciones y de echo define a las civilizaciones.

Cuando el hombre necesita guardar conocimiento o seguir el rastro de la información, lo escribe, y lo cataloga usando índices de papel. Así el libro fue el primer tipo de base de datos. Estas no eran bases de datos electrónicas, sin embargo servían para el mismo propósito. Eran usados para seguir el rastro de libros de contabilidad, conocimiento científico e histórico. Hoy cuando pensamos en bases de datos, pensamos en bases de datos electrónicas, no en estos objetos que definieron a la civilización hace varios miles de años.

El término bases de datos fue escuchado por primera vez en un simposio celebrado en California en 1963. Posteriormente, el uso de las bases de datos se desarrolló a partir de las necesidades de almacenar grandes cantidades de información. Sobre todo, desde la aparición de las primeras computadoras, el concepto de bases de datos ha estado siempre ligado a la informática. (Mato, 1999)

1.2 Conceptos asociados al tema

Administración

Técnica encargada de la planificación, organización y control de los recursos de la organización o empresa con el fin de obtener el máximo beneficio posible. (Mato, 1999)

Datos

El dato es una representación simbólica (numérica, alfabética, algorítmica, entre otros.), un atributo o característica de una entidad. Los datos describen hechos empíricos, sucesos y entidades. Los datos aisladamente pueden no contener información humanamente relevante. Los datos pueden consistir en números, estadísticas o proposiciones descriptivas. Los datos convenientemente agrupados, estructurados e interpretados se consideran que son la base de la información humanamente relevante que se pueden utilizar en la toma de decisiones, la reducción de la incertidumbre o la realización de cálculos (Robert, 2001)

Según las investigaciones de Sanguino se conceptualiza la palabra dato como "elemento o cantidad conocida que sirve de base para la resolución de un problema". Principio en el que se asienta una discusión. Elemento o base para la formulación de un juicio. (Heber, 2009)

Base de datos

Muchos autores, han planteado diferentes definiciones, sin embargo, todos siguen la misma idea, ejemplo de estas son las siguientes definiciones:

Una base de datos es un conjunto de datos persistentes que es utilizado por los sistemas de aplicación de alguna empresa dada. (Dante, 2010)

Conjunto de datos interrelacionados entre sí, almacenados con carácter más o menos permanente en la computadora. O sea, que puede considerarse una colección de datos variables en el tiempo. (Mato, 1999)

Una base de datos (BD) "es una entidad en la cual se pueden almacenar datos de manera estructurada, con la menor redundancia posible, es la recopilación de datos

en forma ordenada y concreta, donde diferentes programas y diferentes usuarios deben poder utilizar estos datos. (Cali, 2006)

Otra definición de base de datos es que es un conjunto de datos almacenados entre los que existen relaciones lógicas y ha sido diseñada para satisfacer los requerimientos de información de una empresa u organización. En una base de datos, además de los datos, también se almacena su descripción. (Murray, 2003)

Las bases de datos (BD) intentan compendiar la normativa de las distintas administraciones son un instrumento de búsqueda que facilita el acceso a la información. Una BD es una agrupación de información perteneciente a un mismo contexto y almacenados sistemáticamente para su posterior uso, constituyendo la base fundamental de cualquier sistema de información. Debido al constante desarrollo tecnológico la mayoría de las BD hoy en día se encuentran en formato digital. Permite guardar gran cantidad de información de forma organizada, funcionando como un enorme almacén para que en el momento deseado pueda ser utilizada con facilidad, permitiendo su recuperación rápida y segura.

Es posible considerar a la base de datos como una especie de armario electrónico para archivar; es decir, es un depósito o contenedor de una colección de archivos de datos computarizados. Los usuarios del sistema pueden realizar una variedad de operaciones sobre dichos archivos. (Piatini, 2005)

Elementos de una base de datos

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

Atributos: Son los diferentes campos que conforman la estructura de una base de datos.

Campos: Es la unidad más pequeña de datos.

Registro: Es un conjunto de campos o atributos relacionados entre sí.

Archivo: Es un conjunto de registros relacionados.

Sistema de gestión

Un sistema de gestión (SG) es un programa que permite crear una estructura de soporte para la creación y administración de contenidos, una interfaz que controla una o varias bases de datos. El sistema permite manejar de manera independiente el contenido y permite crear, editar, gestionar contenidos. (Mato, 1999)

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

Los SG ofrecen a los usuarios una percepción de la base de datos que está, en cierto modo, por encima del nivel del hardware y que maneja las operaciones del usuario expresadas en términos de ese nivel más alto de percepción. (Dante, 2010)

Persistencia de datos

La persistencia de datos es la representación residual de datos que han sido de alguna manera nominalmente borrados o eliminados. Este residuo puede ser debido a que los datos han sido dejados intactos por un operativo de eliminación nominal o por las propiedades físicas del medio de almacenaje. La persistencia de datos posibilita en forma inadvertida la exhibición de información sensible si el medio de almacenaje es dejado en un ambiente sobre el que no se tiene control. (Pearson, 2001)

Sistema Gestor de Base de Datos

Sistema Gestor de Base de Datos (SGBD) es el responsable de tratar todas las peticiones de información de los usuarios. Es un software que sirve de interfaz entre la BD, el usuario y las aplicaciones que la utilizan, que pretende manejar de manera clara, sencilla y ordenada un conjunto de datos. (Ribagorda, 2005)

Un sistema de gestión de base de datos es básicamente un sistema computarizado para guardar registros; es decir, es un sistema computarizado cuya finalidad general es almacenar información y permitir a los usuarios recuperar y

actualizar esa información con base en peticiones. La información en cuestión puede ser cualquier cosa que sea de importancia para el individuo u organización; en otras palabras, todo lo que sea necesario para auxiliarle en el proceso general de su administración. (Dante, 2010)

El software que permite la utilización y/o actualización de los datos en una o varias BD, desde diferentes puntos de vista a la vez, por uno o varios usuarios, se denomina Sistema Gestor de Base de Datos.

Este sistema es un conjunto de programas de propósito general, que permite a los usuarios controlar el acceso y la utilización de la base de datos para incluir, modificar o recuperar información, incluyendo prestaciones con el fin de conseguir la independencia, integridad y seguridad de los datos y la concurrencia de los usuarios.

Todo SGBD debe permitir crear y mantener una BD permitiendo especificar tipos, estructuras y restricciones, realizar consultas, actualizarlas y generar informes. Estos sistemas deben tener características u objetivos que deben cumplir, entre ellos:

Control de la redundancia: la redundancia puede tener efectos negativos o no deseados, pues puede provocar inconsistencias en los datos o duplicar el trabajo al actualizar.

Independencia: la independencia de los datos consiste en la capacidad de modificar el esquema lógico o físico de una BD sin tener que realizar cambios en las aplicaciones que sirven de ella.

Seguridad: se debe garantizar que la información se encuentra segura, estableciendo los permisos de acceso y autorización a los usuarios y grupos correspondientes. (Gil, 2000)

1.3 Tipos de Base de datos

Bases de datos jerárquicas

Como su nombre indica, almacenan su información en una estructura jerárquica. En este modelo los datos se organizan en una forma similar a un árbol (visto al

revés), en donde un nodo padre de información puede tener varios hijos. El nodo que no tiene padres es llamado raíz, y a los nodos que no tienen hijos se los conoce como hojas.

Las bases de datos jerárquicas son especialmente útiles en el caso de aplicaciones que manejan un gran volumen de información y datos muy compartidos permitiendo crear estructuras estables y de gran rendimiento. Una de las principales limitaciones de este modelo es su incapacidad de representar eficientemente la redundancia de datos.

Base de datos de red

Éste es un modelo ligeramente distinto del jerárquico; su diferencia fundamental es la modificación del concepto de nodo: se permite que un mismo nodo tenga varios padres (posibilidad no permitida en el modelo jerárquico). Fue una gran mejora con respecto al modelo jerárquico, ya que ofrece una solución eficiente al problema de redundancia de datos; pero, aun así, la dificultad que significa administrar la información en una base de datos de red ha significado que sea un modelo utilizado en su mayoría por programadores más que por usuarios finales. (García, 2005)

Bases de datos transaccionales

Son bases de datos cuyo único fin es el envío y recepción de datos a grandes velocidades, estas bases son muy poco comunes y están dirigidas por lo general al entorno de análisis de calidad, datos de producción e industrial, es importante entender que su fin único es recolectar y recuperar los datos a la mayor velocidad posible, por lo tanto la redundancia y duplicación de información no es un problema como con las demás bases de datos, por lo general para poderlas aprovechar al máximo permiten algún tipo de conectividad a bases de datos relacionales.

Bases de datos relacionales

Éste es el modelo utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. Su idea fundamental es el uso de "relaciones".

Estas relaciones podrían considerarse en forma lógica como conjuntos de datos llamados "tuplas". Pese a que ésta es la teoría de las bases de datos relacionales creadas por Codd, la mayoría de las veces se conceptualiza de una manera más fácil de imaginar. Esto es pensando en cada relación como si fuese una tabla que está compuesta por registros (las filas de una tabla), que representarían las tulpas, y campos (las columnas de una tabla).

En este modelo, el lugar y la forma en que se almacenen los datos no tienen relevancia (a diferencia de otros modelos como el jerárquico y el de red). Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar para un usuario esporádico de la base de datos. La información puede ser recuperada o almacenada mediante "consultas" que ofrecen una amplia flexibilidad y poder para administrar la información. El lenguaje más habitual para construir las consultas a bases de datos relacionales es SQL, Structured Query Lenguaje o Lenguaje Estructurado de Consultas, un estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales. (Date, 1993)

Bases de datos multidimensionales

Son bases de datos ideadas para desarrollar aplicaciones muy concretas, como creación de Cubos OLAP. Básicamente no se diferencian demasiado de las bases de datos relacionales (una tabla en una base de datos relacional podría serlo también en una base de datos multidimensional), la diferencia está más bien a nivel conceptual; en las bases de datos multidimensionales los campos o atributos de una tabla pueden ser de dos tipos, o bien representan dimensiones de la tabla, o bien representan métricas que se desean estudiar. (Riccuti, 1994).

Bases de datos orientadas a objetos

Este modelo, bastante reciente, y propio de los modelos informáticos orientados a objetos, trata de almacenar en la base de datos los objetos completos (estado y comportamiento).

Una base de datos orientada a objetos es una base de datos que incorpora todos los conceptos importantes del paradigma de objetos:

- ✓ Propiedad que permite ocultar la información al resto de los objetos, impidiendo así accesos incorrectos o conflictos.
- ✓ Herencia - Propiedad a través de la cual los objetos heredan comportamiento dentro de una jerarquía de clases.
- ✓ Polimorfismo - Propiedad de una operación mediante la cual puede ser aplicada a distintos tipos de objetos.

En bases de datos orientadas a objetos, los usuarios pueden definir operaciones sobre los datos como parte de la definición de la base de datos. Una operación (llamada función) se especifica en dos partes. La interfaz (o signatura) de una operación incluye el nombre de la operación y los tipos de datos de sus argumentos (o parámetros). La implementación (o método) de la operación se especifica separadamente y puede modificarse sin afectar la interfaz. Los programas de aplicación de los usuarios pueden operar sobre los datos invocando a dichas operaciones a través de sus nombres y argumentos, sea cual sea la forma en la que se han implementado. Esto podría denominarse independencia entre programas y operaciones. (Date, 1993)

Bases de datos documentales

Permiten la indexación a texto completo, y en líneas generales realizar búsquedas más potentes, permiten el acceso a los documentos en formato electrónico, se instalan en una red local y se consultan desde ordenadores conectados a la red. (Rodríguez, 2003)

1.3.1 Selección del tipo de base de datos

Después de realizar un estudio sobre los tipos de base de datos existentes se

evidencia que la base de datos a utilizar es la relacional dadas las disímiles ventajas que esta brinda, además de una fácil integración con la herramienta de programación a utilizar en la solución propuesta.

Herramientas y Tecnologías

Cuba ha dado a la difícil tarea de migrar la tecnología informática a software libre, por esa razón la Facultad Regional Mártires de Artemisa realiza sus proyectos mediante herramientas libres contribuyendo al desarrollo tecnológico del país.

1.4. Herramientas de modelado CASE

Las herramientas CASE (Ingeniería de Software Asistido por Computadora) son un conjunto de programas que brindan a los desarrolladores asistencia para el desarrollo del ciclo de vida de un proyecto. Con la introducción de estas herramientas se han podido confeccionar y modificar los diagramas que mejoran la calidad y productividad del software. Tienen como objetivo fundamental automatizar los aspectos clave en el proceso de desarrollo.

Dentro de las características más importantes de las herramientas CASE están:

- ✓ Alto rendimiento.
- ✓ Calidad de software desarrollado.
- ✓ Archivos de datos de conocimiento.
- ✓ Facilidades de uso.

Entre las herramientas CASE libres más utilizadas se pueden mencionar:

Umbrello: Rápida, ligera, sencilla de usar, no se pone lenta cuando los proyectos son enormes. Requiere de más opciones de generación de documentación de los diseños modelados dentro de ella. Permite generar código en diversos lenguajes y

soporta los diagramas UML estándares. (Zapata, 2005)

ArgoUML: Es una herramienta profesional de Modelado con UML, de uso libre. Incluye todos los diagramas UML 1.3, permite ingeniería inversa de código Java y generación de código. Disponible para las plataformas más importantes e integración con Eclipse. Se ha desarrollado con los criterios de código abierto y licencia libre. (Kendall, 2000)

Visual Paradigm UML: Es un Lenguaje de Modelado Unificado (UML), una herramienta profesional que implementa el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Ayuda a la construcción de aplicaciones más rápidas, con la calidad requerida y a un menor costo. Permite representar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Esta herramienta proporciona además, abundantes tutoriales, demostraciones interactivas y proyectos UML. (Zapata, 2005)

Características

- ✓ Provee todas las funcionalidades deseadas en una herramienta para modelar bajo UML.
- ✓ Genera código en varios lenguajes.
- ✓ Sus diseños son exportables a XML y pueden ser importados por algunos Frameworks.

Power Architect: Herramienta de modelado de datos que fue creada por los diseñadores de almacenamiento de datos y tiene muchas características únicas dirigidas específicamente para el arquitecto de almacenamiento de datos. Permite a los usuarios de la herramienta ingeniería inversa de bases de datos existentes,

realizar perfiles de datos en bases de datos de origen y generar automáticamente los metadatos de ETL. (Kendall, 2000)

1.4.1 Herramientas CASE a utilizar

Se decide utilizar Power Architect y Visual Paradigm UML ya que es muy fácil su uso y garantiza todas las etapas y aspectos del ciclo de desarrollo del proyecto ya que permiten mejorar la productividad en el desarrollo y mantenimiento del software. También aumentan la calidad del proyecto reduciendo el tiempo y costo del mismo. Además automatizan la documentación del proyecto ayudan a la reutilización del software, portabilidad y estandarización de la información.

1.5 Sistemas Gestores de Base de Datos

Apache Derby:

Es un SGDB escrito en Java que puede ser empotrado en aplicaciones Java y utilizado para procesos de transacciones online. Tiene un tamaño de 2 MB de espacio en disco. Inicialmente distribuido como IBM Cloudscape, Actualmente se distribuye como Sun Java DB.

PostgreSQL

PostgreSQL es uno de los mejores Sistemas Gestores de Base de Datos utilizado principalmente por la comunidad de software libre por sus cualidades: soporta el lenguaje común de acceso a los datos: SQL. Es multiplataforma y posee buenas interfaces de instalación y administración. Aproxima los datos a un modelo Objeto-Relacional, y es capaz de manejar completas rutinas y reglas. Ejemplos de su avanzada funcionalidad son consultas SQL declarativas, control de concurrencia multiversión, soporte multiusuario, transacciones y optimización de consultas. [1]

Entre sus ventajas se encuentra:

- ✓ Restauración de bases de datos en procesos paralelos, que acelera recuperación de un respaldo hasta 8 veces.

- ✓ Privilegios por columna, que permiten un control más granular de datos confidenciales.
- ✓ Configuración de ordenamiento configurable por base de datos, lo cual hace a PostgreSQL más útil en entornos con múltiples idiomas.
- ✓ Actualizaciones “en el lugar” desde 8.3 a 8.4 con muy bajo downtime, gracias al uso de pg_migrator beta.
- ✓ Nuevas herramientas de monitoreo de consultas que le otorgan a los administradores mayor información sobre la actividad del sistema.
- ✓ Nuevas herramientas de monitoreo de consultas.
- ✓ Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.
- ✓ Soporta el uso de índices, reglas y vistas.

MySQL

MySQL es un sistema de gestión de bases de datos relacional y multiusuario. Utilizado principalmente en aplicaciones web, como Drupal o phpBB. Es capaz de almacenar una enorme cantidad de datos de gran variedad y de distribuirlos para cubrir las necesidades de cualquier tipo de organización. MySQL es un software de código abierto, licenciado bajo la Licencia Pública General (GPL). (Hernández, 1999).

Entre sus ventajas se encuentran:

- ✓ Tablas hash en memoria temporales
- ✓ Completo soporte para operadores y funciones en cláusulas select y where.
- ✓ Completo soporte para cláusulas group by y order by, soporte de funciones de agrupación.
- ✓ Ofrece un sistema de contraseñas y privilegios seguro mediante verificación basada en el host y el tráfico de contraseñas está cifrado al conectarse a un servidor.

- ✓ Soporta gran cantidad de datos. MySQL Server tiene bases de datos de hasta 50 millones de registros.
- ✓ En MySQL 5.0, los clientes y servidores Windows se pueden conectar usando memoria compartida.
- ✓ MySQL contiene su propio paquete de pruebas de rendimiento proporcionado con el código fuente de la distribución de MySQL.

1.5.1 Sistema Gestor de Base de Datos a utilizar

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

1.6 Herramientas de programación

NetBeans IDE es un entorno de desarrollo - una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java - pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el NetBeans IDE.

Es un producto libre y gratuito sin restricciones de uso. NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios. La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos.

Eclipse es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados, también se puede usar para otros tipos de aplicaciones cliente.

Zend Studio o **Zend Development Environment** es un completo entorno de desarrollo integrado para el lenguaje de programación PHP. Está escrito en Java, y está disponible para las plataformas Microsoft Windows, Mac OS X y GNU/Linu. Es una propuesta para el desarrollo de aplicaciones Web, actuando en la parte cliente. (<http1>)

1.6.1 Herramienta de programación a utilizar

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

1.7 Frameworks

Hibernate es una herramienta de Mapeo objeto-relacional (ORM) para la plataforma Java (y disponible también para .Net con el nombre de Hibernare) que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) o anotaciones en los beans de las entidades que permiten establecer estas relaciones. Hibernate es software libre, distribuido bajo los términos de la licencia GNU LGPL. (Bauer y King, 2005).

Entre sus ventajas:

1. Hibérnate genera las sentencias SQL y libera al desarrollador del manejo manual de los datos que resultan de la ejecución de dichas sentencias
2. Flexible en cuanto al esquema de tablas utilizado, para poder adaptarse a su uso sobre una base de datos ya existente.
3. Posee la funcionalidad de crear la base de datos a partir de la información disponible.
4. Ofrece también un lenguaje de consulta de datos llamado HQL

Apache Struts es framework libre para crear aplicaciones de web. Estas aplicaciones se diferencian de sitios web convencionales porque pueden crear una respuesta dinámica. Muchos sitios web entregan páginas sólo estáticas. Un uso de web que se le puede dar a Struts es que puede actuar recíprocamente con bases de datos y motores lógicos de negocio para personalizar una respuesta. (htt12)

Spring

El **Spring Framework** (conocido simplemente como **Spring**) es un framework de código abierto de desarrollo de aplicaciones para la plataforma Java. Es muy facilidad la integración con los estándares J2EE y herramientas comerciales

existentes, particularmente útil con aplicaciones basadas en la arquitectura MVC (Modelo Vista Controlador). (Bauer y King, 2005).

1.7.1 Framework a utilizar

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

1.8 Modelo de acceso a datos

En el modelo de acceso a datos, una capa es un nivel lógico en el cual residen componentes o aplicaciones lógicas. Las capas pueden residir en uno a más equipos o servidores, el número de capas hace referencia al número de niveles y no al número de equipos en los cuales los servicios son divididos. Las capas que generalmente se incluyen en aplicaciones son:

Capa de Cliente: Conocida como capa de Presentación es la que contiene las interfaces en las que el usuario interactúa con el sistema.

Capa de la Lógica de Negocios: Contiene la lógica que interactúa con el origen de datos. Esta capa intermedia contiene la parte de la aplicación que interactúa con los datos, por ejemplo, la creación de una cadena de conexión al origen de datos.

Conclusiones parciales

En este capítulo han sido tratados conceptos fundamentales que son la base de la investigación. Se describe detalladamente la situación problemática siendo el punto de partida para la investigación. Fueron indagados conceptos asociados al sistema, así como de base de datos y sus relaciones con términos como información y sistema gestor de base de datos; permitiendo ampliar el conocimiento

para entender los complejos procesos. Como resultado de la investigación se utilizara, Base de Datos relacional ya que son muchas las ventajas que este modelo ofrece, como SGBD utilizaremos PostgreSQL 9.0, herramienta CASE se utilizara Visual Paradigm y Power Architect para la realización del modelaje de la base de datos y los diagramas necesarios en el trabajo de diploma

Capítulo 2: Diseño y arquitectura

Introducción

Todo desarrollo de una investigación está encaminado a obtener como resultado un producto, en este caso una base de datos. En el presente capítulo se realizará la caracterización del diseño y la arquitectura a utilizar en la base de datos. Para construir el modelo de datos para la base de datos para el DCIE del MINCEX se transitó por las tres fases generales, que reflejan los detalles del diseño del almacenamiento de datos así como los mecanismos e recuperación.

2.1 Diseño de Bases de Datos Relacional

Lo primero para crear una base de datos, es un estudio de lo que se quiere almacenar en ella, como el tipo de información que se quiere almacenar en la misma, los datos a guardar, teniendo en cuenta dos aspectos: la información disponible y la información que necesitamos.

La planificación de la estructura de la base de datos, en particular de las tablas, es vital para la gestión efectiva de la misma. El diseño de la BD consiste en una descripción de cada una de las tablas, así como de los campos que las componen, el registro y los valores que contendrá cada uno de esos campos.(Jacobson, 2000)

Son muchas las consideraciones a tomar en cuenta al momento de hacer el diseño de la base de datos, quizá las más fuertes sean:

- ✓ La velocidad de acceso.
- ✓ El tamaño de la información.
- ✓ El tipo de la información.
- ✓ Facilidad de acceso a la información.
- ✓ Facilidad para extraer la información requerida.
- ✓ El comportamiento del manejador de bases de datos con cada tipo de información.

Objetivos del diseño de Bases de Datos Relacional

El objetivo principal del diseño de una base de datos relacional es generar un conjunto de esquemas de relaciones que permitan almacenar la información con un mínimo de redundancia, pero que a la vez faciliten la recuperación de la información. Además de:

- ✓ Pedir sólo lo necesario y ser explícito.
- ✓ Almacenar sólo la información relevante.
- ✓ Seleccionar el tipo de dato apropiado
- ✓ Normalizar las estructuras de tablas.
- ✓ Utilizar índices apropiadamente

2.2 Información a Gestionar

El sistema debe ser capaz de manejar toda la información relacionada con la DCIE del MINCEX, los datos asociados y catalogación de los mismos. La información a almacenar se clasifica por diferentes modelos. El modelo de Directivas, Presupuesto y Cierre Trimestre también se almacenara la información relacionada con el expediente de la AEI y las solicitudes de los Organismos. La información contenida en el sistema solo debe ser accedida por usuarios que tengan roles asignados y permiso para su consulta y su manipulación. La aplicación contara con un proceso de autenticación de usuario y control de acceso para todas las funcionalidades realizadas en la base de datos según los roles definido, para así mantener la integridad y seguridad de la información.

2.3 Requisitos funcionales

Los requisitos funcionales son declaraciones de los servicios que debe proporcionar el sistema, la manera que debe comportarse ante de entradas particulares y situaciones peculiares. (Sommerville, 2005).

El presente trabajo de diploma da solución a los siguientes requisitos funcionales de la DCIE del MINCEX. Estos requisitos fueron definidos por los analistas luego de realizar el levantamiento de información en la Dirección de Control de la Inversión

Extranjera.

RF1 Insertar Datos de la Solicitud

RF2 Actualizar Datos de la Solicitud

RF3 Buscar Datos de la Solicitud

RF4 Mostrar Datos de la Solicitud

RF5 Archivar Solicitud.

RF6 Aprobar Solicitud.

RF7 Rechazar Solicitud.

RF8 Insertar Presupuesto

RF9 Actualizar valores del Presupuesto

RF10 Buscar valores del presupuesto

RF11 Mostrar valores del Presupuesto

RF12 Realizar Reportes de cada uno de los indicadores del Presupuesto.

RF13 Exportar reportes de cada uno de los indicadores del Presupuesto.

RF14 Insertar datos del Capital y Financiamiento

RF15 Actualizar datos del Capital y Financiamiento

RF16 Realizar Reportes de cada uno de los indicadores del Capital y Financiamiento.

RF17 Exportar reportes de cada uno de los indicadores del Capital y Financiamiento.

RF18 Realizar Reportes de cada uno de los indicadores del Cierre del Proyecto.

RF19 Exportar reportes de cada uno de los indicadores del Cierre del Proyecto.

RF20 Insertar indicadores de las directivas.

RF21 Actualizar indicadores de las directivas.

RF22 Buscar indicadores de las directivas.

RF23 Mostrar indicadores de las directivas.

RF24 Realizar reportes de cada uno de los indicadores de las directivas.

RF25 Exportar reportes de cada uno de los indicadores de las directivas.

RF26 Insertar indicadores de las AEI

RF27 Actualizar indicadores de las AEI

RF28 Buscar indicadores de las AEI

RF29 Mostrar indicadores de las AEI.

2.4 Metodología para el diseño de Bases de Datos Relacional

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

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

- Determinación de entidades y atributos.
- Normalización de entidades.
- Determinación de relaciones.
- Obtención del modelo lógico global de los datos.
- Diseño físico de la BD.

Cuando se realiza el diseño de la BD para un sistema determinado, es necesario establecer los datos a tomar en cuenta y las dependencias funcionales existentes entre ellos. Esto se obtiene luego de realizada la etapa de análisis del sistema y partiendo de lo obtenido en esta. (Mato, 1999).

2.5 Capa de Acceso a Datos

Es la capa destinada a acceder a los datos, mediante el uso de Objetos de Acceso a Datos (DAOs). Los datos están guardados en un sistema de representación relacional, por lo tanto los DAOs se encargan de obtener representaciones orientadas a objetos que serán usados por los servicios. Con el uso de esta capa aseguramos la independencia entre la lógica del negocio y el sistema de representación de los datos, pudiendo cambiarse la representación de los datos sin

afectar a la lógica de la aplicación. Usaremos tantos DAOs como módulos existen en el sistema.

2.5.1 Mapeo de la Base de Datos

El mapeo objeto-relacional (más conocido por su nombre en inglés, Object-Relational mapping, o sus siglas O/RM, ORM, y O/R mapping) es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional, utilizando un motor de persistencia. En la solución propuesta para la BD del DCIE se realiza en mapeo de la base de datos. Para así poder crear los Objetos de Acceso a Datos (DAO) para acceder a la base de datos. Esto posibilita el uso de las características propias de la orientación a objetos (básicamente herencia y polimorfismo). Esto trae como ventaja rapidez en el desarrollo. Abstracción de la base de datos. Reutilización, Seguridad, Mantenimiento del código. Lenguaje propio para realizar las consultas (Bauer y King, 2005).

2.5.2 Ingeniería Inversa

La ingeniería inversa se ha definido como el proceso de construir especificaciones de un mayor nivel de abstracción partiendo del código fuente de un sistema software o cualquier otro producto se puede utilizar como punto de partida cualquier otro elemento de diseño. La aplicación de ingeniería inversa nunca cambia la funcionalidad del software sino que permite obtener productos que indican cómo se ha construido el mismo.

La finalidad de la ingeniería inversa es la de desentrañar los misterios y secretos de los sistemas en uso a partir del código. Para ello, se emplean una serie de herramientas que extraen información de los datos, procedimientos y arquitectura del sistema existente. (Cejas, 2006)

Esto brinda ventajas como:

- ✓ Reducir la complejidad del sistema

- ✓ Generar diferentes alternativas
- ✓ Recuperar y/o actualizar la información perdida
- ✓ Facilitar la reutilización

2.6 Modelo Conceptual y Modelo Lógico

Los dos modelos fundamentales de datos son el conceptual y el lógico. Ambos son conceptuales en el sentido de que convierten parámetros del mundo real en abstracciones que permiten entender los datos sin tener en cuenta la física de los mismos. El modelo conceptual se define exteriormente al SGBD, realizándose, manualmente o con la ayuda de algún sistema automatizado, la transformación entre el modelo conceptual y el lógico global. El proceso de modelación conceptual es denominado también modelación semántica, ya que con estos modelos se pretende reflejar, en mayor medida, la semántica o significado de los datos y sus interrelaciones.

Para una mejor comprensión del diseño de la base de datos propuesta se descubrirá el modelo lógico de datos, en él se ilustran las entidades lógica clave y sus relaciones que son necesarias para satisfacer los requisitos del sistema para datos persistentes que sean coherentes con la arquitectura de la aplicación.

sus interrelaciones y propiedades.

El DER de la DCIE del MINCEX, está compuesto por varios modelos por lo que se expone dividido en 3 modelos.

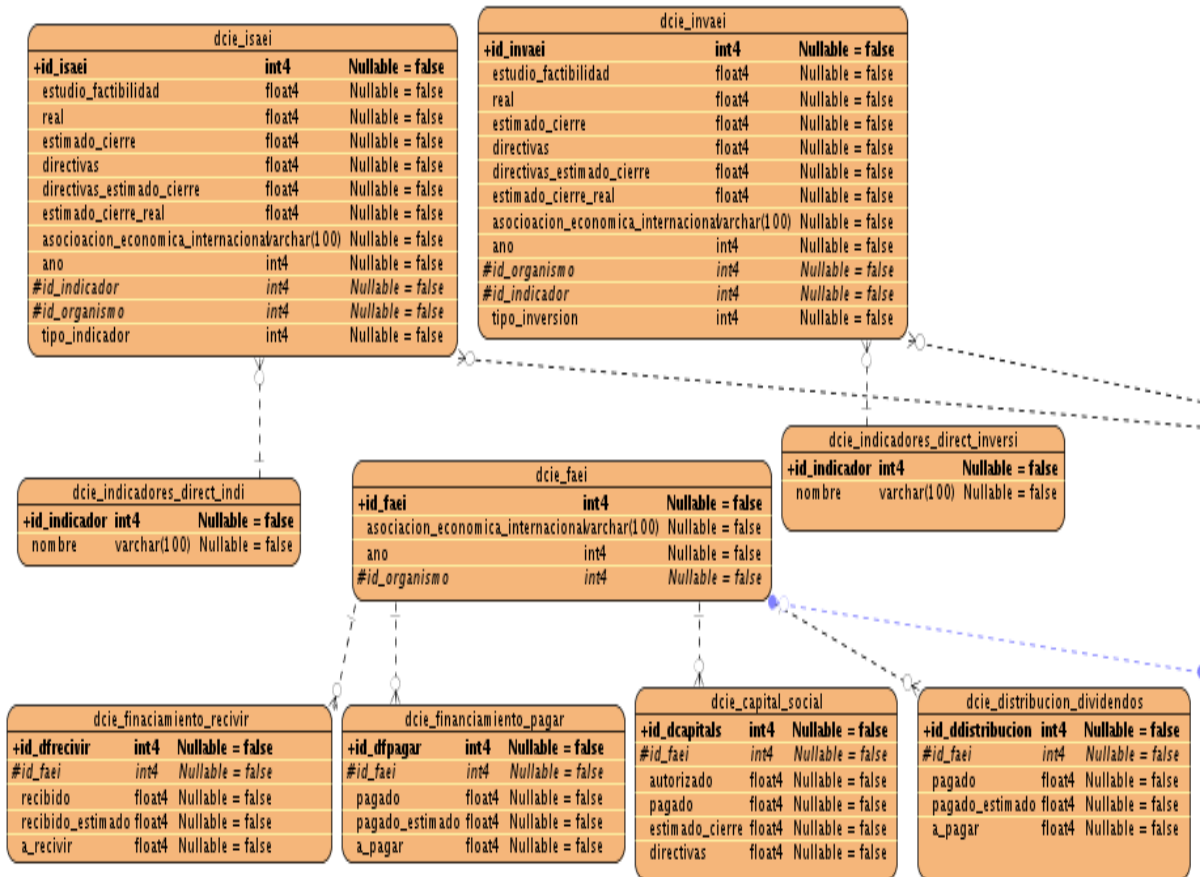


Fig. 2 Modelo Entidad Relación perteneciente al modelo directivas

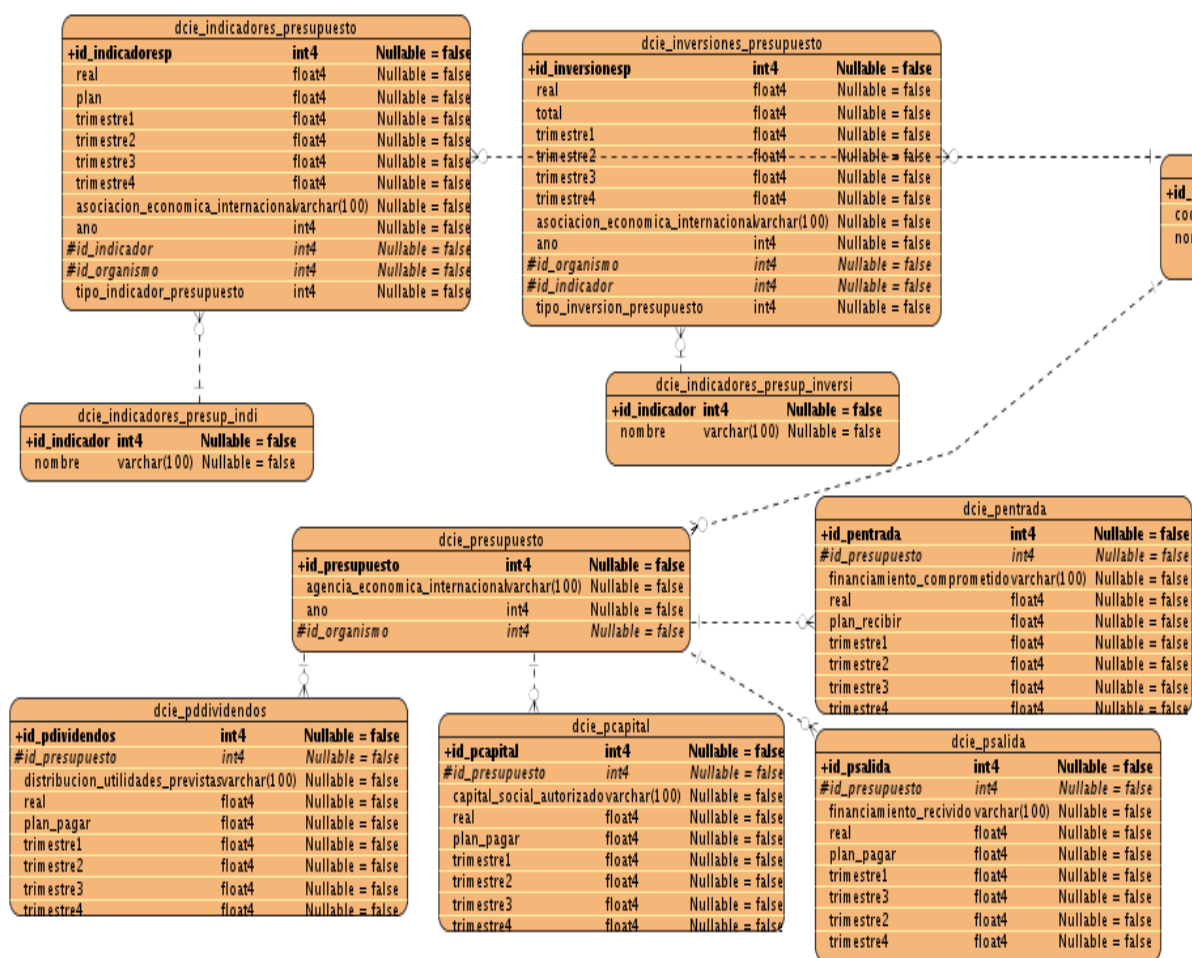


Fig. 3 Modelo Entidad Relación perteneciente al Modelo Presupuesto

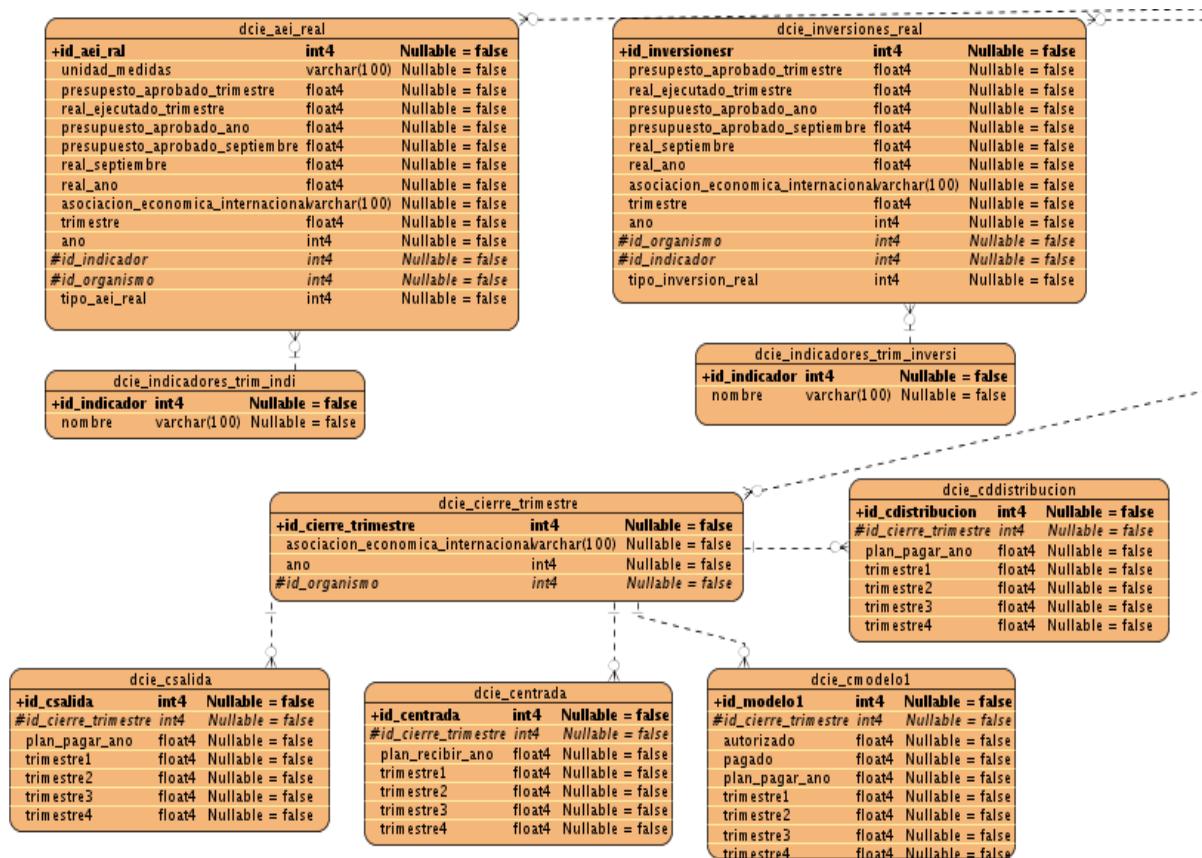


Fig. 4 Modelo Entidad Relación perteneciente al Modelo Cierre Trimestre

Conclusiones parciales

El diseño propuesto para la base de datos satisface los requisitos funcionales de la DCIE, resuelve las insuficiencias de versiones anteriores. A lo largo del capítulo se explica su estructura y ventajas que lo hacen adaptable y escalable a diferentes entornos de ejecución. En este punto se tiene una descripción clara y precisa del modelo entidad relación, lo que servirá de documentación y contribuirá en la toma de decisiones de líderes y desarrolladores del sistema. Esto garantiza que el diseño de las estructuras de la base de datos y el gestor utilizado para el desarrollo de la misma soporta el almacenamiento de los niveles de información requeridos para el comienzo del funcionamiento de la base de datos.

Capítulo 3: Implementación y validación

Introducción

En el presente capítulo se aborda la validación teórica y funcional del diseño de la base de datos. Inicialmente se recogen aspectos como la integridad, seguridad, redundancia de la información. En un segundo momento se tratan aspectos como las pruebas realizadas a la base de datos para comprobar su correcto y eficiente funcionamiento.

3.1 Validación Teórica del diseño de la Base de Datos

Como en todo sistema de información, se deberá verificar que la base de datos y las aplicaciones funcionan correctamente. Para realizar la validación del diseño de la BD se establece un análisis detallado de la integridad de la información, características y funcionalidades, que aseguren la calidad de almacenamiento y disponibilidad de los datos. Con su tratamiento se evitan errores de entrada introducidos por usuarios descuidados o cualquier otra circunstancia de intento de violar la información existente en la base de dato.

3.2 Validación de la capa de acceso a datos.

El desarrollo de la capa de acceso a datos, presenta un proceso importante, en la solución propuesta la misma garantiza la consistencia, integridad y seguridad de los datos, Este último representa entre los aspectos mencionados el más importante debido a que garantiza el control de acceso a los datos por las personas autorizadas y además que estos datos no se dañen debido a acciones que no hayan sido previamente controladas.

3.3 Integridad de la Base de Datos

Una vez que se define la estructura de datos del modelo relacional, se definen las reglas de integridad que los datos almacenados en dicha estructura deben cumplir para garantizar que sean correctos. Al definir cada atributo sobre un dominio se impone una restricción sobre el conjunto de valores permitidos para cada atributo. Hay además dos reglas de integridad muy importantes que son restricciones que se deben cumplir en todas las bases de datos relacionales. Estas reglas son la de integridad de entidad y las de integridad referencial.

La integridad es uno de los factores más importantes a la hora de realizar el diseño de una base de datos esta se divide en diferentes aspectos.

Integridad referencial: Garantiza interrelaciones válidas entre entidades. Implica que los datos sean correctos y sin duplicaciones, datos perdidos o relaciones mal resueltas.

Integridad de Entidad: Define una fila como entidad única para una tabla determinada en la base de datos. La integridad de entidad exige la integridad de las columnas de los identificadores o la clave principal de una tabla, mediante índices y restricciones únicas, o restricciones de llaves primarias. Las restricciones de entidades aseguran la integridad de las entidades que son modeladas por el sistema.

3.4 Normalización de la Base de Datos

La normalización es la expresión formal del modo de realizar un buen diseño y provee los medios necesarios para describir la estructura lógica de los datos en un sistema de información. La normalización es el proceso mediante el cual se transforman datos complejos a un conjunto de estructuras de datos más pequeñas, que además de ser simples y más adaptables, son más fáciles de mantener. También se puede entender la normalización como una serie de reglas que sirven para ayudar a los diseñadores de bases de datos a desarrollar un esquema que minimice los problemas de lógica. Cada regla está basada en la que antecede. Se

puede concluir planteando que la normalización es el proceso en el cual se va comprobando el cumplimiento de una serie de reglas, o restricciones; cada regla que se cumple aumenta el grado de normalización ayudando a evitar anomalías en la actualización y mejorando la independencia de los datos.

El diseño de base de datos conceptual proporciona el esquema de relaciones y sus restricciones de integridad que sirve como punto inicial para el proceso de implementación de la base de datos. El proceso de normalización es un estándar que básicamente, es un proceso de conversión de las relaciones evitando:

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

Proceso de Normalización

El proceso de normalización va reduciendo un conjunto de relaciones, a una forma más sencilla y simple. Consiste en ir descomponiendo relaciones en otras relaciones con un mayor grado de sencillez en términos de las operaciones que se ejecutan sobre ellas. El proceso incluye varias fases que se realizan en orden. La realización de la 2da. Fase supone que se ha concluido la 1ra y así sucesivamente. Tras completar cada fase se dice que la relación está en: 1FN, 2FN, 3FN, FNBC. Se dice que una determinada relación está en una forma normal si satisface un cierto conjunto de restricciones.

Formas Normales

Existen básicamente seis formas normales incluyendo la Forma Normal BOYCE-COOD. Cada una de estas formas tiene sus propias reglas. Cuando una base de datos se conforma a un nivel, se considera normalizada a esa forma de normalización. No siempre es una buena idea tener una base de datos conformada en el nivel más alto de normalización, puede llevar a un nivel de complejidad que pudiera ser evitado si estuviera en un nivel más bajo de normalización.

En general, las primeras tres formas normales son suficientes para cubrir las necesidades de la mayoría de las bases de datos. El creador de estas tres primeras formas normales o reglas fue Dr. Edgar Frank Codd7.

Primera Forma Normal (1FN)

Formalmente una relación está en 1FN si cumple la propiedad de que sus dominios no tienen elementos que a su vez, sean conjuntos. Los valores que puede tomar un atributo de una relación son los elementos contenidos en su correspondiente dominio. Si se permitiera que un elemento del dominio de un atributo fuera un conjunto, entonces dicho atributo pudiera tomar como valor ese conjunto de valores.

Eso implicaría que en una posición fila, columna habría un conjunto de valores y no un único valor, como debe ocurrir en el modelo relacional. Se puede traducir lo dicho anteriormente como: Una tabla está en primera Forma Normal si y solo si:

- ✓ Todos los atributos son atómicos. Un atributo es atómico si los elementos del dominio son indivisibles, mínimos.
- ✓ La tabla contiene una clave primaria.
- ✓ La tabla no contiene atributos nulos.
- ✓ Si no incluye ningún grupo repetitivo (un grupo repetitivo es un atributo que contiene un conjunto de valores y no un único valor).

Lo que se persigue en realidad al normalizar a este nivel es:

- ✓ Eliminar los grupos repetitivos de las tablas individuales.
- ✓ Crear una tabla separada por cada grupo de datos relacionados.
- ✓ Identificar cada grupo de datos relacionados con una clave primaria.

Segunda Forma Normal (2FN)

Una relación R se dice que está en **2FN** si está en **1FN** y si, y solo si, los atributos no llaves (ni primarias, ni candidatas) de R , si los hubiese, son funcional y completamente dependientes de la llave primaria de R [2]. La regla de la 2da forma normal establece que todas las dependencias parciales se deben eliminar y separar dentro de sus propias tablas. Una dependencia parcial es un término que describe a aquellos que no dependen de la llave primaria de la tabla para identificarlos.

Lo que persigue en realidad al normalizar a este nivel es: Crear tablas separadas para aquellos grupos de datos que se aplican a varios registros. Relacionar estas tablas mediante una clave externa.

Tercera Forma Normal (3FN)

Una relación R está en 3FN si está en 2FN y si, y solo si, los atributos no llaves son independientes de cualquier otro atributo no llave primaria. Esto es lo mismo que decir que se deben eliminar las dependencias transitivas de atributos no llaves respecto a la llave primaria, estando ya la relación en 2FN[2]. Una tabla está normalizada en esta forma si todas las columnas que no son llave funcionalmente dependientes por completo de la llave primaria y no hay dependencias transitivas. Una dependencia transitiva es aquella en la cual existen columnas no llave que dependen de otras columnas que tampoco son llave. Cuando las tablas están en 3ra. Forma Normal se previenen errores de lógica cuando se insertan o borran registros. Cada columna en una tabla está identificada de manera única por la llave primaria, y no deben datos repetidos. Esto provee un esquema limpio y elegante,

que es fácil de trabajar y expandir. Lo que se persigue en realidad al normalizar a este nivel:

- ✓ Creando tablas para los atributos no llaves que no dependen transitivamente de la llave primaria (y los atributos que no se analizan por ser atributos llaves pertenecientes a claves candidatas).
- ✓ Creando una tabla para los atributos no llaves que dependen transitivamente de la llave primaria a través de otro atributo o conjunto de atributos no llave primaria (que no son parte de la llave primaria). La llave primaria de la relación así formada será el atributo o conjunto de atributos a través de los cuales existe la dependencia transitiva.

Análisis de la redundancia de la información

La redundancia hace referencia al almacenamiento de los mismos datos varias veces en diferentes lugares. La redundancia de datos puede provocar problemas como:

- Incremento del trabajo: Como un mismo dato está almacenado en dos o más lugares, esto hace que cuando se graben o actualicen los datos, deban hacerse en todos los lugares a la vez.
- Desperdicio de espacio de almacenamiento: Ya que los mismos datos están almacenado en varios lugares distintos, ocupando así más bytes del medio de almacenamiento. Este problema es más evidente en grandes bases de datos.
- Inconsistencia de datos: Esto sucede cuando los datos redundantes no son iguales entre sí. Esto puede suceder, por ejemplo, cuando se actualiza el dato en un lugar pero el dato duplicado en otro lugar no actualizado.

Por todo lo antes planteado se puede llegar a la conclusión que la base de datos para el módulo de la dirección del control de la inversión extranjera del MINCEX no cuenta con redundancia de datos puesto que a la misma además de aplicársele las formas normales hasta la tercera forma normal se hizo énfasis en el diseño del modelo entidad-relación de la base de datos para precisamente evitar la redundancia de los datos, exceptuando la redundancia de datos controlada, que se emplea para mejorar el rendimiento en la base de datos y disminuir el tiempo de acceso a los datos.

3.5 Integración de los módulos existentes.

El proyecto Sistema Informativo del MINCEX se encuentra dividido en varios módulos, entre estos se encuentra el módulo de la DCIE. El módulo se conecta con su esquema correspondiente (dir_cie) en la base de datos con el objetivo de gestionar la información mediante la capa de acceso a datos. Una vez se haya establecido la conexión con las entidades en la base de datos se procede a la integración de la BD con el servidor del módulo para poder acceder a los datos.

3.6 Análisis de la seguridad y recuperación de la base de datos

Los datos constituyen un recurso valioso que debe ser estrictamente controlado gestionado al igual que cualquier otro recurso corporativo. La seguridad garantiza el acceso autorizado a los datos, para interrumpir cualquier intento de acceso no autorizado, ya sea por error del usuario o por mala intención. Las bases de datos, en particular, deben tener un sistema de seguridad sólido para controlar las actividades que pueden realizarse y determinar qué información puede verse y cuál puede modificarse. Un sistema de seguridad sólido, asegura la protección de datos sin tener en cuenta cómo los usuarios obtienen el acceso a la base de datos.

Los Sistemas Gestores de Bases de Datos actuales, garantizan en cierta medida el control de acceso a la base de datos, por parte de los usuarios, se realice de acuerdo a los privilegios que tengan asignados. En el caso de PostgreSQL permite realizar configuraciones para controlar los permisos a los usuarios.

Para seguridad en la base de datos de la Dirección del Control de la Inversión Extranjera del MINCEX se definió control total a todas las tablas de la base de datos solamente a usuarios que respondieran al rol de Administrador.

3.7 Estructura de paquetes en la Capa de Acceso

Dado un conjunto de clases abstractas relacionadas, el patrón AbstractEntity permite el modo de crear instancias de estas clases abstractas desde el correspondiente conjunto de subclases concretas. Proporciona una interfaz para crear objetos relacionados o dependientes sin especificar su clase concreta. El patrón AbstractEntity puede ser muy útil para permitir a un programa trabajar con una variedad compleja de entidades externas, tales como diferentes sistemas de ventanas con una funcionalidad similar.

La capa de acceso a datos de la DCIE poseerá 2 paquetes los cuales contendrán todo lo referente a la base de datos que se le realiza el mapeo relacional.

Paquete dao

Posee la clase GenericDao en la cual están implementados todos los métodos para acceder a la BD. A la clase GenericDao se le creará solamente una instancia de tipo GenericDao para cada vez que se le solicite. A través de esa instancia devuelta se podrá acceder a los DAOs, que en consecuencia también sólo se crearán una vez y no andaremos creando un mismo DAOs cada vez que lo necesitamos. Así que, además consta de métodos abstractos que crean cada uno de los DAOs que usamos en la aplicación.

Paquete entity

Este paquete posee todas las tablas de la base de datos mapeadas y llevadas al NetBeans. A través de los DAOs y los métodos de acceso de estas clases se podrá acceder a los datos que estas poseen y utilizarlos según convenga al usuario.

3.8 Métodos de la clase GenericDao

Todos los métodos de la clase GenericDao son importantes y cumplen una gran función para el desarrollo de la capa de acceso a datos.

Los métodos de la clase GenericDao son:

Métodos	Descripción
<pre>public void save(AbstractEntity entity){ getHibernateTemplate().save(entity); }</pre>	Salva la instancia persistente de la entidad dada.
<pre>public void update(AbstractEntity entity) { getHibernateTemplate().saveOrUpdate(entity);}</pre>	Actualiza la instancia persistente de la entidad dada
<pre>public <T> List<T> find(Class<T> entityClass) { return getHibernateTemplate().loadAll(entityClass);}</pre>	Busca una clase dada y retorna una lista con las instancias encontradas.
<pre>public AbstractEntity findByExample(AbstractEntity exampleEntity) return AbstractEntity)getHibernateTemplate().findByExample(exa mpleEntity);}</pre>	Se le pasa como parámetro la entidad que se desea buscar y retorna las instancias de esa clase.
<pre>public <T> AbstractEntity findByPK(Class<T> entityName, long id){ return (AbstractEntity)getHibernateTemplate().get(entityName, id);}</pre>	Dado una clase y el identificador, retorna las coincidencias encontradas
<pre>public <T> AbstractEntity findByPK(Class<T> entityName, String name){ return (AbstractEntity)getHibernateTemplate().get(entityName, name);}</pre>	Dado una clase y el nombre, retorna las coincidencias encontradas.

<pre> 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); } </pre>	<p>Busca todos los objetos de una clase dada que cumplan con un parámetro y su valor, retorna una lista con los registros que cumplan con la búsqueda.</p>
---	--

Fig. 5 Métodos de la clase GenericDao

3.9 Pruebas

Pruebas de aceptación

El cliente es quién determina si la propuesta realizada es factible para sus procesos de negocio, por ello se realizaron una serie de actividades encaminadas a validar la arquitectura y el modelo de datos de la Base de Datos.

Definición de modelos conceptuales con el cliente: Para definir las entidades referentes a cada uno de los procesos que se manejan en la DCIE fue necesario realizar un análisis del negocio existente en la dirección, es por ello que se realizaron reuniones con los clientes y el personal del proyecto encargado de la realización del levantamiento de requisitos.

Múltiples encuentros técnicos para la revisión del modelo de la BD: el modelo de datos ha sufrido múltiples revisiones por parte del personal encargado en el proyecto, encaminadas a lograr que se cumpla con las necesidades del negocio.

Entrega a Calisoft del modelo de BD propuesto para su validación parcial: durante la revisión realizada por Calisoft al módulo de la Dirección del Control de la Inversión Extranjera del MINCEX fue entregado el modelo canónico de datos, el cual no fue objeto de inconformidad, por lo que queda aprobado el mismo, según las normas de calidad de la Facultad Regional Mártires de Artemisa.

Pruebas unitarias

Las pruebas realizadas a la base de datos mediante la capa de acceso a datos fueron planeadas para identificar posibles problemas en la BD, verificar el funcionamiento de los objetos de acceso a datos y su tiempo de respuesta. Estas fueron elegidas principalmente de acuerdo a su demanda; es decir, se probaron aquellos que serán ejecutados por un número considerable de usuarios al mismo tiempo, pudiendo poner en riesgo el desempeño y tiempo de respuesta del sistema.

Tomando en cuenta el funcionamiento del sistema. Las pruebas fueron realizadas de manera local en una computadora personal (PC) con procesador Pentium 4 a 2.0 GHz. y 512 Mb. de memoria RAM. Los resultados de la pruebas pueden variar de acuerdo al contexto en que se realicen éstas.

A continuación se presentan los métodos que se emplearon en las pruebas.

Método:	save (insertar).	
Condiciones:	Debe existir un objeto de la clase entidad especificada.	
Objetivo de la Prueba	Resultado Esperado	Resultado de la Prueba
- Insertar un objeto de la entidad especificada correspondiente a la tabla en la base de datos.	- Se espera que el objeto sea insertado en la tabla correspondiente en la base de datos.	- El objeto fue insertado satisfactoriamente. -Se insertaron los datos correspondiente a las tabla de la base de datos - Mensaje al usuario "Datos Insertados Correctamente"

Fig. 6. Método Insertar.

The screenshot shows an IDE with three tabs: `Vivienda1.java`, `Dmediotransportetetransporte.java`, and `Denunciadogeneralvivienda.java`. The `Vivienda1.java` tab is active, showing the following code:

```
26
27
28
29
30     Resource resource = new FileSystemResource("/C:/Users/postgres/Desktop/daily/hibernate-persistence.xml");
31     XmlBeanFactory bean = new XmlBeanFactory(resource);
32
33     dao = (GenericDao) bean.getBean("genericDAO");
34
35     Denunciadogeneralvivienda e= new Denunciadogeneralvivienda(8,4,"15-02-2012");
36     dao.save(e);
37
38 }
39 }
```

Below the code editor is the `Output - vivienda1 (run)` window, which displays the following log output:

```
INFO: org.hibernate.cfg.SettingsFactory buildSettings
INFO: Structured second-level cache entries: disabled
Apr 22, 2012 8:50:48 PM org.hibernate.cfg.SettingsFactory buildSettings
INFO: Echoing all SQL to stdout
Apr 22, 2012 8:50:48 PM org.hibernate.cfg.SettingsFactory buildSettings
INFO: Statistics: disabled
Apr 22, 2012 8:50:48 PM org.hibernate.cfg.SettingsFactory buildSettings
INFO: Deleted entity synthetic identifier rollback: disabled
Apr 22, 2012 8:50:48 PM org.hibernate.cfg.SettingsFactory buildSettings
INFO: Default entity-mode: pojo
Apr 22, 2012 8:50:48 PM org.hibernate.cfg.SettingsFactory buildSettings
INFO: Named query checking : enabled
Apr 22, 2012 8:50:48 PM org.hibernate.impl.SessionFactoryImpl <init>
INFO: building session factory
Apr 22, 2012 8:50:50 PM org.hibernate.impl.SessionFactoryObjectFactory addInstance
INFO: Not binding factory to JNDI, no JNDI name configured
Hibernate: select nextval ('mod_infyserv_vivienda.denunciadogeneralvivienda_idenunciadogeneral_seq')
Hibernate: insert into mod_infyserv_vivienda.denunciadogeneralvivienda (codigo, fecha, id_municipio, idenunciadogeneral) values (?, ?, ?, ?)
BUILD SUCCESSFUL (total time: 14 seconds)
```

Fig. 10. Ejemplo de la ejecución satisfactoria de la función save (insertar).

Se muestra la forma de declarar e insertar un objeto en la entidad seleccionada. Como resultado de la ejecución de la función save se insertó el objeto en la tabla correspondiente en la base de datos

Editar Datos - edfdf (10.208.3.27:5432) - si_min			
Archivo Editar Vista Herramientas Ayuda			
Sin límite			
	id_organis [PK] integ	codigo character varying(100)	nombre character varying(100)
1	2	10204	MINAL
2	26	1020	Ministerio del Transporte
3	27	10208	Ministerio de Cultura
4	28	123	Ministerio de Salud
5	29	789	Ministerio de la Agricultu
*			

Fig. 8. Resultado de la ejecución de la función save (insertar) en la tabla organismo.

Método:	update (actualizar).	
Condiciones:	Debe existir un objeto de la clase entidad especificada.	
Objetivo de la Prueba	Resultado Esperado	Resultado de la Prueba
- Actualiza un objeto de la entidad especificada correspondiente a la tabla en la base de datos.	- Se espera que el objeto sea actualizado en la tabla correspondiente en la base de datos.	- El objeto fue actualizado satisfactoriamente. -Se actualizaron los datos correspondiente a las tabla de la base de datos - Mensaje al usuario "Datos Actualizados Correctamente"

Fig. 9. Método Actualizar.

```

42
43
44     DesiOrgAdmCentralEstado dorg = (DesiOrgAdmCentralEstado) dao.findByPK(DesiOrgAdmCentralEstado.class, 2)
45     dorg.setNombre("Ministerio del Interior");
46     dorg.setSiglas("MINIT");
47     dao.update(dorg);
48
49 }
50
51 }
52

```

Output - MincexFinal (run)
 Apr 8, 2012 5:01:50 PM org.hibernate.cfg.SettingsFactory buildSettings
 INFO: Deleted entity synthetic identifier rollback: disabled
 Apr 8, 2012 5:01:50 PM org.hibernate.cfg.SettingsFactory buildSettings
 INFO: Default entity-mode: pojo
 Apr 8, 2012 5:01:50 PM org.hibernate.cfg.SettingsFactory buildSettings
 INFO: Named query checking : enabled
 Apr 8, 2012 5:01:50 PM org.hibernate.impl.SessionFactoryImpl <init>
 INFO: building session factory
 Apr 8, 2012 5:01:51 PM org.hibernate.impl.SessionFactoryObjectFactory addInstance
 INFO: Not binding factory to JNDI, no JNDI name configured
 Hibernate: select desiorgadm0_id_oace as idl_6_0_, desiorgadm0_nombre as nombre6_0_, desiorgadm0_siglas as siglas6_0_ from dir_esi.desi_org
 Hibernate: update dir_esi.desi_org_adm_central_estado set nombre=?, siglas=? where id_oace=?
 BUILD SUCCESSFUL (total time: 3 seconds)

Fig. 10. Ejemplo de la ejecución satisfactoria de la función update (actualizar).

A continuación se muestra la forma de actualizar un objeto en la entidad deseada. Como resultado de la ejecución de la función update se tiene el objeto actualizado en la tabla correspondiente en la base de datos.

Editar Datos - edfdf (10.208.3.27:5432) - sj_mincex -

Archivo Editar Vista Herramientas Ayuda

Sin límite

	id_organismos [PK] integ	codigo character varying(100)	nombre character varying(100)
1	2	10204	MINAL
2	26	1020	Ministerio del Transporte
3	27	10208	Ministerio de Cultura
4	28	123	Ministerio de Salud Publica
5	29	789	Ministerio de la Agricultura
*			

Fig. 11. Resultado de la ejecución de la función update en la tabla organismo.

Conclusiones parciales

Luego de haber confeccionado el modelo de la BD y la implementación de la capa de acceso y realizar las pruebas de validación se puede decir que se cumplió el objetivo propuesto dándole respuesta al problema planteado. La base de datos obtenida contiene una estructuración de los datos que posibilitan una organización y uniformidad propia del modelo, la cual logra la disponibilidad de información mediante los métodos de acceso a datos con un tiempo de respuesta vertiginoso.

Conclusiones Generales

Luego de haber concluido el diseño de la base de datos y la implementación de la capa de acceso para el módulo de la Dirección del Control de la Inversión Extranjera del Ministerio del Comercio Exterior, se puede decir que se obtuvieron los resultados esperados.

- Se logró un diseño del modelo lógico de la base de datos minimizando la información redundante en el mismo.
- Se implementó el modelo físico de la base de datos con restricciones de integridad que garantizan que los datos involucrados o modificados sean los correctos.
- Se obtuvo un diagrama de clases persistentes donde las relaciones entre las tablas fueron construidas adecuadamente.
- Se implementó la capa de acceso a dato garantizando la correcta disponibilidad e integridad de los datos almacenados en al BD.

Recomendaciones

- Optimizar las funcionalidades en la BD para la persistencia de objetos.
- Utilizar el contenido de esta investigación como referencia para el desarrollo de próximas BD en la Universidad de las Ciencias Informáticas.
- Aplicar las concepciones adoptadas para el desarrollo de la capa de acceso a datos en otros proyectos que utilicen las mismas tecnologías.
- Lograr la integración con el mercado de datos Control de la Inversión Extranjera del MINCEX para apoyar la toma de decisiones de los directivos.

Bibliografía Referenciada

(Bartle, 2009) Phil Bartle (2009). *INFORMACIÓN PARA LA GESTIÓN y gestión de la información.* [En línea] 2009. [Citado el: 5 de abril de 2012.] <http://www.scn.org/mpfc/modules/mon-miss.htm>.

(Cejas, 2006) C.B; Crespillo, O.G; Jiménez F., M.J.; Ramírez G, C.; Sánchez G, C.; Sánchez N., C (2006). *Tipos de Lenguajes de Programación.*

(Dante, 2010) Gloria Dante (2010) *Gestión de la Información.* [En línea] 2010. [Citado el: 3 de abril de 2012.] <http://www.fahce.unlp.edu.ar/academica/Areas/bibliotecologia/Catedras/gestindeinformacin/>.

(Date, 1993) Date, C. (1993). *Introducción a los Sistemas de bases de datos.* Addison-Wesley, Wilmington, E.U.A.

(García, 2005) Ignacio García (2005) título: *Base de datos Modelo red general*

(Gil, 2000) Gil Fidel, Albrigo Javier, Do Rosario, Javier 2000: *Sistema de Gestión de Base de Datos*

(Heber, 2009) Romero Heber. (2009). *Gestión del conocimiento y la información en el Polo de Gestión Universitaria.* Universidad de las Ciencias Informáticas.

(Hernández, 1999) Vegas Hernández, J. (1999). *Un Sistema de Recuperación de Información sobre Estructura y Contenido.* PhD tesis, Universidad de Valladolid, Valladolid, España.

(Jacobson, 2000) Jacobson Ivar, Booch Grady y Rumbaugh, James. 2000. *El proceso Unificado Del Desarrollo de Software.*

(Kendall, 2000) Joaquín Kendall 2000 *Análisis Y Diseño De Sistemas 3ª. Edición*

(Mato, 1999), Rosa María. (1999). *Sistemas de Bases de Datos. La Habana: Pueblo y Educación.*

(Murray, 2003) Pablo Murray. 2003. *GESTIÓN – INFORMACIÓN - CONOCIMIENTO. Escuela de Bibliotecarios de la Biblioteca Nacional de Buenos Aires. Buenos Aires, Argentina: s.n., 2003.*

(Ribagorda, 2005) Arturo Ribagorda (2005). *Seguridad en bases de datos. Madrid : Fundación Dintel, 2005.*

(Rodríguez, 2003) Luis Rodríguez (2003). *Bases de datos documentales*

(Sommerville, 2005) Ian Sommerville. 2005. *Ingeniería del Software. Madrid, España: Editorial PEARSON EDUCACIÓN.*

(Zapata, 2005) Giraldo, L. and Y. Zapata (2005). *"Herramientas de desarrollo de ingeniería de software para Linux." Monitoria de Ingesoft.*

[1]PostgreSQL, Sitio oficial de PostgreSQL. 2011.

[Bauer y King, 2005] Bauer, C. y G. King. (2005). *Hibernate in Action. A guide to the concepts and of object/relational mapping. Manning Publications Co.*

[Riccuti,1994] Riccuti, Mike (1994). "Multidimensional Analysis: Winning the Competitive Game".

Joaquin Caselles 2007: Curso de lenguaje SQL

Mario Piattini, Adoración de Miguel, Marcos Esperanza. DISEÑO DE BASES DE DATOS RELACIONALES.

Pearson C.J. (2001). Educación Introducción a los Sistemas de Bases de Datos – Date.

Robert L kruse (2001) Libro: Estructura de Datos y Diseño de Programas

Zhao J, and D Thomas 2005: "Comparación de Herramientas de modelado UML: Enterprise Architect y Rational Rose."

Bibliografía consultada

Akif, M., S. Brodhead, et al. (2001). Java y XML. Referencia para programadores.

Ambler, 2002.Ambler, S. (2002)."Mapping Objects to Relational Databases".

Bauer, C. y G. King. (2005). Hibernate in Action. A guide to the concepts and of object/relational mapping. Manning Publications. **HARWRYSZKIEWYCZ, I T.**

Análisis y diseño de base de datos. Editorial Megabyte. Noriega Editores. México. 1994. pp29/31

LAUDON, Kenneth C 1996. Administración de los sistemas de información. 3ra. Edición.

Hoeller, J. (2007). "ClassHibernateDaoSupport" Extraído el 27 de Febrero, 2007 de <http://www.springframework.org/docs/api/org.springframework.orm.hibernate/support/HibernateDaoSupport.html>.

Hommel, S. (1999). "Conversiones de Código para el lenguaje de programación JAVA." Extraído el 2 de Febrero, 2007 de <http://usuarios.lycos.es/manualesjava/manuales/convencionscodigojava/convencionescodigojava.pdf>.

King, G. (2006). "Hibernate Reference Documentation 3.2.2". http://www.hibernate.org/hib_docs/v3/reference/en/pdf/hibernate_reference.pdf.

Martín, James.2004. Análisis y Diseño Orientado a Objetos.

Microsystem, S. (2004). "Interface Serializable." Java 2 Platform Standard <http://java.sun.com/j2se/1.5.0/docs/api/java/io/Serializable.html>.

Panduranga, S. N. (2005). Pro Hibernate: From Professional to Expert.

Richardson, C. (2006). Pojos in Action: Developing Enterprise Applications with Lightweight Frameworks.