Universidad de las Ciencias Informáticas

Facultad Regional Mártires de Artemisa

Título: *Base de Datos para los Módulos de Colaboración y Asociación de Combatientes y Familiares de los Mártires del Sistema Informativo de la Administración Provincial de Artemisa.*

Autor: Yasiel Santo Domingo.

Tutor: Ing. Martha Luisa Gala.

Co-Tutor: Lic. Yoan León Guerra.

Provincia Artemisa

Junio, 2012

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

La verdadera sabiduría está en reconocer la propia ignorancia.

Sócrates

A mis padres por apoyarme en esta travesía y estar siempre que los necesito. No sería nada sin ellos.

Agradecimientos

A mis padres, a ellos les debo mi ser.

A mi hermano por darme motivos de distracción cuando mas estresado estaba.

A toda mi familia por estar siempre pendiente de mí y ayudarme en estos largos años.

A mi novia Aylin por aguantarme en todo momento y darme animo cuando las cosas estaban más feas.

A todos mis amigos de la escuela, a los que llegaron al final y a los que no pudieron. Gracias por compartir juntos esta experiencia.

A mis tutores Martha y Yoan por guiarme en el desarrollo de este trabajo.

A todos los que de una forma u otra me han ayudado a superarme.

Muchas gracias.

DECLARACIÓN DE AUTORÍA

Declaro ser el único autor del trabajo de diploma titulado “Base de Datos para los Módulos de Colaboración y Asociación de Combatientes y Familiares de los Mártires del Sistema Informativo de la Administración Provincial de Artemisa.” y autorizo 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 Junio del año 2012.

_____

Firma del Autor

Yasiel Santo Domingo García

_____

Firma del Tutor

Ing. Martha Luisa Gala Rodríguez

_____

Firma del Cotutor

Lic. Yoan León Guerra

Resumen

En las direcciones de Colaboración y Asociación de Combatientes y Familiares de los Mártires (ACFM) existen grandes problemas a la hora de gestionar la información. Los trabajadores del centro no poseen mecanismos eficiente que les permita almacenar y recupera la información, razón por la cual el trabajo se vuelve lento e ineficiente. Para resolver este problema se realizó una investigación la cual tiene como objetivo principal desarrollar una base de datos que permita almacenar y recuperar la información en las Direcciones de Colaboración y ACFM.

En el desarrollo de la base de datos se utiliza la metodología de desarrollo SXP junto a un grupo de herramientas que permitieron obtener un producto con la calidad requerida. A partir de una serie de pruebas se valida que la solución desarrollada cumple con el objetivo trazado en la investigación. Como resultado de la investigación se obtuvo una base de datos que permite tener la información almacenada de forma íntegra, disponible y centralizada.

Palabras Claves

Base de datos, centralidad, disponibilidad, integridad, SXP

Índice

Introducción 1

Capítulo 1. Fundamentos Teóricos para desarrollo de la base de datos 9

Introducción 9

1.1- Conceptos Asociados al dominio del problema. 9

1.2- Análisis de otras soluciones existentes 17

1.3- Selección de la metodología para el desarrollo de la solución 18

1.4- Elección de las tecnologías y herramientas a utilizar. 19

Conclusiones del capítulo 26

Capítulo 2. Análisis y desarrollo de la Solución 28

Introducción 28

2.1- Objetivos estratégicos de la organización 28

2.2- Flujo actual de procesos 29

2.3- Requisitos del sistema informativo del gobierno 31

2.4- Diagramas para cada uno de los procesos en las diferentes direcciones. 34

2.5- Arquitectura del sistema 45

2.6- Solución para el acceso a los datos desde la aplicación (Capa de acceso a datos) 46

Conclusiones del Capítulo. 55

Capítulo 3: Validación de la solución propuesta 56

Introducción 56

3.1- Validación teórica del diseño de la BD. 56

3.2- Validación funcional de la base de datos y la capa de acceso a datos. 60

3.3- Revisión Realizada por el departamento de calidad. 67

3.4- Solución y Funcionalidades Obtenidas. 67

Conclusiones del capítulo 68

Conclusiones Generales 69

Recomendaciones 70

Bibliografía 71

Introducción

Los medios de información y comunicación tienen un gran impacto en la sociedad actual. La aplicación generalizada de los ordenadores y las redes hacen que cada día se haga más necesario el uso de las Tecnologías de la Información y las Comunicaciones (TIC). Estas ocupan un lugar importante en la sociedad y en la economía del siglo XXI. El concepto de TIC surge como convergencia tecnológica de la electrónica, el software y las infraestructuras de las telecomunicaciones.

El uso y el acceso a la información son el objetivo principal de las TIC. Su uso se hace cada vez más dependiente de la tecnología, los crecientes volúmenes de datos que se manejan obligan a un tratamiento con medios cada vez más sofisticados. El acceso a redes como Internet mediante ordenadores personales o la complejidad de los sistemas bancarios y de reconocimiento de rostros son pruebas evidentes del alto desarrollo con que cuentan las TIC.

En la actualidad la información se ha convertido en un recurso muy valioso. Esta marca la diferencia entre los individuos, las empresas y los gobiernos. Dominar grandes cantidades de información genera mucho conocimiento y este a su vez proporciona ventajas en cualquier esfera de la sociedad.

Para las empresas u organismos la información significa mucho en su desarrollo pues contribuye enormemente a realizar toma de decisiones y de esta forma tener grandes avances. Con el objetivo de manipular toda esta información de modo seguro y rápido se crean los sistemas de gestión de información. Estos tienen como principal objetivo la gestión de grandes cantidades de datos almacenados en bases de datos.

Las bases de datos se han convertido en el medio más utilizado para guardar la información digital. Estas tienen sus inicios en la antigüedad cuando los datos se escribían en libros y se almacenaban en bibliotecas para tenerlos organizados y

protegidos. Posteriormente su uso se desarrolló a partir de las necesidades de almacenar grandes cantidades de datos. Actualmente debido al desarrollo tecnológico de campos como la informática y la electrónica la mayoría de las bases de datos están en formato digital.

Las bases de datos han evolucionado rápidamente variando desde la forma en que se almacenan los datos hasta los métodos de obtención de estos. Poseen una independencia lógica y física de los datos, dado a la capacidad que tienen de poder modificar alguna información específica sin que afecte a los demás registros. También cuentan una forma única de proteger los datos que las hace inigualables.

Con el desarrollo de internet se crearon un sin número de bases de datos para consultar a través de servicios web. Actualmente servicios que almacenan los nombres de los países, las zonas horarias y las cuentas bancarias son algunos de los tantos utilizados por los desarrolladores en sus aplicaciones. Hoy en día el 95% de las aplicaciones web utilizan bases de datos para almacenar su información. Existen aplicaciones como Google, Wikipedia y Yahoo que se basan generalmente en búsqueda de información de forma optimizada en grandes bases de datos.

La automatización del proceso de gestión y almacenamiento de la información se ha globalizado en todas las esferas de la sociedad. En instituciones como bancos, líneas aéreas y cadenas de tiendas su uso se hace imprescindible para lograr un crecimiento sostenido. Sin dudas almacenar y gestionar grandes volúmenes de información se ha convertido en la vía para el desarrollo de la sociedad moderna.

Cuba a pesar de ser un país fuertemente bloqueado no ha estado ajena al desarrollo de las TIC. La exigencia del mercado internacional y la globalización creciente de la informática impulsan al gobierno cubano a tomar la decisión de informatizar la gestión y el almacenamiento de la información en todas las esferas del país. A raíz de esto se crean numerosos proyectos cubanos que desarrollan software para resolver este problema. Instituciones como DATYS, DeSoft y la UCI son líderes de esta rama en el país.

La Universidad de las Ciencias Informáticas (UCI) se crea en el 2002 para lograr un mayor avance en el proceso de informatizar la industria cubana e incorporar el uso de las TIC a la economía del país. Como resultado del éxito de esta universidad en el año 2007 se crea la Facultad Regional Mártires de Artemisa ubicada precisamente en la provincia de Artemisa. En esta facultad actualmente se desarrolla una aplicación para la Administración Provincial de Artemisa.

La Administración Provincial de Artemisa surge en el 2011 como resultado de la creación de la provincia Artemisa, implementando así una nueva forma de gobierno basada en direcciones. Para esto se dividió el consejo en 32 direcciones separando la jefatura gubernamental de la jefatura empresarial. Las direcciones controlan a cada una de las empresas de su especialidad. Dentro de estas direcciones se encuentran las direcciones de Colaboración y la dirección de Asociación de Combatientes y Familiares de los Mártires (ACFM).

La dirección de Colaboración se encarga de almacenar la información referente a los proyectos de Colaboración y donativos puntuales. Los proyectos después de aprobarse su financiamiento se le realiza un detallado seguimiento hasta que finalizan. También se almacena todo lo referente al desarrollo de misiones internacionalistas que brinda la provincia. Se controlan las personas que cumplen misiones, la ayuda que se le brinda a los familiares, los internacionalistas que desertan y los aportes de estas misiones a la provincia. Se tiene constancia de los estudiantes extranjeros que residen o cursan estudios en la provincia.

La dirección de ACFM está compuesta por los departamentos Inserción Laboral, Registro y Control, Seguridad Social y Asistencia Militar. En estos departamentos se gestiona toda la información sobre los combatientes de la provincia que reciben ayuda. Se le hace un tratamiento a cada combatiente, a sus familiares, se brinda ayuda económica, alimenticia, se le realizan chequeos médicos cada cierto tiempo.

Al fallecer un combatiente se garantizan todos los recursos para darle una despedida a su altura, luego se le da seguimiento a su familia para abastecerla de

lo necesario para vivir. En el departamento de Inserción Laboral se controlan todos los datos de los jóvenes que cursan el servicio militar activo. El ministerio del trabajo en conjunto con este departamento se encarga de reinsertarlos laboralmente cuando cumplan su misión dentro de las Fuerza Armadas Revolucionarias.

En ninguna de las direcciones antes mencionadas existe un correcto proceso de gestión de la información, producto a que esta se recibe por diferentes vías (correo, teléfono o mensajeros). Luego se almacena en formato duro o en archivos Excel provocando problemas de integridad al perderse o modificarse datos en el proceso. Toda la información se gestiona de forma manual por lo que se hace muy difícil tenerla disponible en el momento necesario.

Al no almacenar la información de forma centralizada los datos se encuentran duplicados en diferentes lugares, trayendo problemas para mantenerlos actualizados. Por esta razón la mayoría de las veces los trabajadores no manipulan la información real de la dirección. Todos estos contratiempos hacen que gestionar la información en estas direcciones sea un proceso lento e ineficiente.

Luego de un profundo análisis de la situación problemática existente y con el fin de solucionarla se plantea el siguiente **problema científico:** ¿Cómo elevar los niveles de integridad, disponibilidad y centralidad de los datos en el proceso de gestión de la información de las direcciones de Colaboración y ACFM de la Administración Provincial de Artemisa?

Para lograr solucionar de forma satisfactoria el problema planteado se necesita realizar un estudio exhaustivo del **objeto de estudio**: proceso de gestión de la información en las áreas de Colaboración y ACFM de la Administración Provincial de Artemisa. Específicamente en el **campo de acción**: proceso de almacenamiento y recuperación de la información en las áreas de Colaboración y ACFM.

Para resolver el problema se plantea como **objetivo general**: Desarrollar una base de datos para los módulos Colaboración y ACFM del Sistema Informativo de la Administración Provincial (SINAP) de Artemisa, de manera que aumenten los niveles de integridad, disponibilidad y centralidad de la información en sus respectivas direcciones.

Para un mayor entendimiento del objetivo general se desglosó en los siguientes **objetivos específicos:**

- Estudiar los fundamentos teórico-metodológicos para el desarrollo de las bases de datos en los sistemas de gestión de información.
- Conocer el proceso de gestión de información de las direcciones de Colaboración y ACFM.
- Evaluar los niveles de integridad y centralidad de la información en las direcciones de Colaboración y ACFM.

La idea a defender sobre la cual se sustenta la investigación es: Al integrarse una base de datos a los módulos de Colaboración y ACFM del SINAP de Artemisa, se elevaran los niveles de integridad, disponibilidad y centralidad de la información en sus respectivas direcciones.

Variable Independiente: Base de datos para los módulos de Colaboración y ACFM del SINAP de Artemisa.

Variables Dependientes: Integridad, Disponibilidad y Centralidad.

Para darle cumplimiento a los objetivos de la investigación se trazaron una serie de **Tareas de la Investigación:**

- Establecimiento de los fundamentos teórico-metodológicos para el desarrollo de la base de datos en los sistemas de gestión.

- Caracterizar el proceso de gestión de información en las direcciones de Colaboración y ACFM de la Administración Provincial de Artemisa en cuanto a Integridad, disponibilidad y centralidad de la información.

- Establecimiento de los fundamentos que deben sostener las bases de datos para los módulos de Colaboración y ACFM del SINAP de Artemisa.

- Desarrollo de bases de datos con sus respectivas capas de acceso para los módulos de Colaboración y ACFM del SINAP de Artemisa.

- Validar la contribución lograda a través de la integración de las bases de datos en los niveles de integridad, disponibilidad y centralidad de la información en el proceso de gestión de la información de las direcciones de Colaboración y ACFM.

Para el desarrollo de la solución propuesta se utilizaron los siguientes **métodos científicos**.

Métodos teóricos

- **Análisis histórico-lógico:** Posibilitó determinar las características esenciales de los sistemas de gestión gubernamental existentes y realizar el estudio de la trayectoria real de determinados elementos que servirán de guía para el desarrollo de la solución.

- **Análisis y síntesis:** A través de este método se pudo estudiar cada uno de los factores que generan la situación problemática de forma separada, para luego unirlos como un todo y descubrir las relaciones que existen entre ellos y cómo influyen de forma genérica en la problemática.

- **Inducción-deducción:** Permitió el estudio de numerosos casos particulares a través de la muestra escogida, para llegar a determinadas conclusiones con el resumen genérico de estos casos.

- **Modelación:** Facilitó la creación de diagramas que representan abstracciones, con el objetivo de explicar el proceso de almacenamiento y recuperación de los datos.

- **Enfoque de Sistema:** Se empleó para estudiar el SINAP de Artemisa y de esta forma poder entender la estructura y función de cada uno de los componentes que forman parte de los subsistemas, que a su vez forman parte del sistema general.

Métodos empíricos

- **Análisis de documentos:** Empleado para estudiar los documentos por los que se guían los especialistas de las direcciones de Colaboración y ACFM para realizar su trabajo.
- **Entrevistas:** Se usó para obtener información sobre el flujo de los procesos que se manejan en las direcciones de Colaboración y ACFM. (Anexo 1)

- **Matemático-Estadístico:** Utilizado para realizar cálculos matemáticos y definir estadísticas a partir del muestreo de diferentes aspectos (Población-muestra y pruebas funcionales) a lo largo de la investigación.

Como **población** se tomaron 5 especialistas de la dirección de ACFM y los 2 especialistas de la dirección de Colaboración. La **muestra,** seleccionada está compuesta por el 100% de la población en cada caso.

Necesidad y aporte práctico de la investigación:

Al crearse los módulos de Colaboración y ACFM del SINAP surge la **necesidad** de desarrollar una base de datos y una capa de acceso a datos que garantice el buen funcionamiento del proceso de almacenamiento y recuperación de la información.

Como **aporte práctico** se tiene: Una base de datos para los módulos de Colaboración y ACFM que garantice el correcto almacenamiento y recuperación de la información en sus respectivas direcciones.

El documento de tesis que se presenta tiene la siguiente **estructura**: Una introducción que presenta la situación problemática, la necesidad de la investigación, así como los principales elementos del diseño teórico metodológico.

Capítulo 1. Fundamentos Teóricos para el desarrollo de la base de datos: Contiene toda la teoría correspondiente a la solución propuesta, las tendencias actuales, conceptos y ventajas. También presenta una selección detallada de cada una de las herramientas usadas para el desarrollo de la solución. Así como una investigación sobre soluciones existentes similares a la que se desarrolla.

Capítulo 2. Análisis y desarrollo de la solución: En este capítulo se le da seguimiento al diseño y desarrollo de la base de datos y la capa de acceso a datos. Marcando cada pasó con una serie de artefactos y documentos.

Capítulo 3. Pruebas y validación de los resultados obtenidos: En este capítulo se realizaron una serie de pruebas para comprobar el correcto funcionamiento de la base de datos y la capa de acceso. También se valida el resultado obtenido al introducirse la base de datos a los módulos de Colaboración y ACFM del SINAP de Artemisa.

Capítulo 1. Fundamentos Teóricos para desarrollo de la base de datos

Introducción

En el capítulo se abordan los principales conceptos relacionados con el marco teórico de la solución propuesta. Se hace un estudio del arte para buscar soluciones similares a la propuesta en el trabajo. Además se investiga sobre las tecnologías, las categorías de bases de datos y tendencias actuales en cuanto al uso de ORM para el desarrollo de capas de acceso a datos. Se exponen los criterios sobre los que se basa la investigación para seleccionar las herramientas más adecuadas para el desarrollo de la solución propuesta.

1.1- Conceptos Asociados al dominio del problema.

Para poder entender parte del proceso de la investigación y ver el punto de vista del autor se hace un recuento sobre los principales conceptos asociados al objeto de estudio. A continuación se muestran los conceptos desde el punto de vista de diferentes autores para luego definir cual se asume en la investigación.

Integridad de los datos

“Consiste en garantizar la no contradicción entre los datos almacenados de modo que, en cualquier momento del tiempo, los datos almacenados sean correctos, es decir, que no se detecte inconsistencia entre los datos.” (Mato, 2006) [1]

“El termino integridad se refiere a la corrección de la información contenida en la base de datos. La integridad se asegura controlando que ciertas aserciones

[1] Del libro “Sistemas de Bases de Datos” de Rosa María Matos García, 2006, pág. 5

(restricciones de integridad, R.I) sean cumplidas por la base de datos" (Óscar Pastor, 2007)[2]

Después de indagar en los diferentes conceptos de integridad de datos el autor define en el presente trabajo que el término integridad de los datos hace referencia a "la exactitud y valides de los datos dentro del dominio al que pertenece."

Disponibilidad de los datos

"Aseguramiento de que los usuarios autorizados tienen acceso cuando lo requieran a la información y sus activos asociados." (Polín, 2010)[3]

"Es la característica, cualidad o condición de la información de encontrarse a disposición de quienes deben acceder a ella, ya sean personas, procesos o aplicaciones." (Vieites, 2007)[4]

El autor de la presente investigación después de estudiar varios conceptos asume el concepto dado por Alvaro Gomez Vieites[4] ya que es el que mejor se adapta al marco de la investigación.

Centralidad

"Centralización es la acción y efecto de centralizar. Este verbo, por otra parte, refiere a reunir varias cosas en un centro común o a hacer que distintas cosas dependan de un poder central." (Colectivo, 2012)[5]

El autor de la investigación asume el concepto de centralidad antes mostrado y asume entonces que centralidad de los datos es "la reunificación de los datos de un dominio de forma centralizada."

Información

[2] Del libro "Gestión de Bases de Datos" de Óscar Pastor, Pedro Blesa Pons, 2006, Pág. 24
[3] Tomado del artículo "Confluencia con la LOPD", publicado por la revista Belt Ibérica, 20/10/04
[4] Del libro: "Enciclopedia de la Seguridad Informática", de Alvaro Gomez Vieites, 2007
[5] Tomado del sitio web "http://definicion.de/centralizacion/"

"Los datos suelen ser descritos como elementos discretos, huérfanos de contexto. Cuando los datos son contextualizados, se convierten en información" (Heidi, 2006)[6]

La información *"comprende los datos y conocimientos que se usan en la toma de decisiones"* (Ferrell O. C. y Hirt Geoffrey, 2004)[7]

En la presente investigación se toma el concepto de información dado por Heidi Toffler y Toffler Alvin en su libro "La Revolución de la Riqueza". Este concepto se toma porque define exactamente la diferencia que tienen los términos información y datos dentro del marco de la investigación.

Módulo

El autor se basa en la definición de módulo dada por José E. Gallardo Ruiz y Carmen M. García López en su libro Diseño modular en el cual expresan que un módulo es "un fragmento de un programa que se desarrolla de forma independiente del resto del programa. Esta independencia hace posible un mecanismo de compilación por separado que limita la complejidad del programa que se está desarrollando." (López, 2010)[8]

Base de Datos (BD)

"Una Base de Datos es un 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, 2006) [9]

"Una base de datos es un conjunto de datos almacenados entre los que existen relaciones lógicas y ha sido diseñada para satisfacer los requisitos de información

[6] Del libro: "La Revolución de la Riqueza", de Toffler Alvin y Toffler Heidi, 2006, Pág. 154.
[7] Del libro:" Introducción a los Negocios en un Mundo Cambiante", de Ferrell O. C. y Hirt Geoffrey, McGraw-Hill Interamericana, 2004, Pág. 121.
[8] Del libro "Diseño modular", de José E. Gallardo Ruiz y Carmen M. García López, 2010, pág. 2
[9] Del libro "Sistemas de Bases de Datos", de Rosa María Matos García, 2006, Pág. 3

de una empresa u organización. En una base de datos, además de los datos, también se almacena su descripción. " (Marqués, 2011)[10]

Basándose en los conceptos antes expuestos el autor asume que base de datos es "un conjunto de datos relacionados entre sí almacenados para satisfacer los requisitos de almacenamiento de una empresa u organización".

Clasificaciones de las Bases de Datos

Las bases de datos pueden clasificarse de varias maneras, de acuerdo al contexto que se esté manejando, la utilidad de las mismas o las necesidades que satisfagan.

Según la variabilidad de los datos almacenados se clasifican en: (Peter Rob, 2004)[11]

- **Bases de datos estáticas**

  Son bases de datos de sólo lectura, utilizadas primordialmente para almacenar datos históricos que posteriormente se pueden utilizar para estudiar el comportamiento de un conjunto de datos a través del tiempo, realizar proyecciones, tomar decisiones y realizar análisis de datos para inteligencia empresarial.

- **Bases de datos dinámicas**

  Estas son bases de datos donde la información almacenada se modifica con el tiempo, permitiendo operaciones como actualización, borrado y adición de datos, además de las operaciones fundamentales de consulta. Un ejemplo de esto puede ser la base de datos utilizada en un sistema de información de un supermercado, una farmacia, un videoclub o una empresa.

Modelos de Base de Datos

Una de las características fundamentales de las bases de datos es que proporcionan cierto nivel de abstracción de datos, al ocultar las características

[10] Del libro "Bases de datos" de Mercedes Marqués , 2011,Cap. 1, pág. 2
[11] Del libro "Sistemas de Bases de Datos" de Peter Rob, Carlos Coronel, 2004, pág. 47

sobre el almacenamiento físico que la mayoría de los usuarios no necesitan conocer. Los modelos de datos son el instrumento principal para ofrecer dicha abstracción a través de su jerarquía de niveles.

Un modelo de datos es un conjunto de conceptos que sirven para describir la estructura de una base de datos, es decir, los datos, las relaciones entre los datos y las restricciones que deben cumplirse sobre los datos. Los modelos de datos contienen también un conjunto de operaciones básicas para la realización de consultas (lecturas) y actualizaciones de datos. Además, los modelos de datos más modernos incluyen mecanismos para especificar acciones compensatorias o adicionales que se deben llevar a cabo ante las acciones habituales que se realizan sobre la base de datos. (Marqués, 2011)[12]

Las siguientes tipos de modelos fueron tomados de la definición dada por (Peter Rob, 2004)[13]

- **Modelo jerárquico**

  Una base de datos jerárquica consiste en una colección de registros que se conectan entre sí por medio de ligas. Los registros y las ligas son similares a los del modelo de red, pero en el modelo jerárquico se organiza en forma de árbol con raíz (donde la raíz es nodo ficticio); de tal manera que una base de datos jerárquica es una colección de árboles de este tipo, formando un bosque. A cada árbol con raíz se le denomina árbol de base de datos. En este modelo un registro puede repetirse en varios sitios pudiendo ocasionar los siguientes problemas: Riesgos de la inconsistencia al llevar a cabo actualizaciones. Inevitable desperdicio de espacio en el medio de almacenamiento secundario.

- **Modelo red**

[12] Del libro “Bases de datos” de Mercedes Marqués, 2011, pág. 14
[13] Del libro “Sistemas de Bases de Datos” de Peter Rob, Carlos Coronel, 2004, pág. 23

Este modelo se distingue del jerárquico porque en su concepto de nodo: permite que un mismo nodo tenga varios padres (posibilidad no permitida en el modelo jerárquico).

- **Modelo relacional**

  Una base de datos relacional es una base de datos que cumple con el modelo relacional, el cual es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. Permiten establecer relaciones entre los datos (que están guardados en tablas), y trabajar con ellos conjuntamente. Tras ser postuladas sus bases en 1970 por Edgar Frank Codd, de los laboratorios IBM en San José (California), no tardó en consolidarse como un nuevo paradigma en los modelos de base de datos. A diferencia de otros modelos como el jerárquico y el de red en este modelo no tiene importancia el lugar y la forma en que se almacenan los datos. Permitiendo una considerable ventaja al hacerlo más fácil de entender y de utilizar para usuarios ocasionales de la BD.

- **Modelo orientado a objeto**

  A finales de los 80 aparecieron las primeras BDOO (Base de datos Orientadas a Objeto), es una base de datos inteligente. Soporta el paradigma orientado a objetos almacenando datos y métodos, y no sólo datos. Está diseñada para ser eficaz, desde el punto de vista físico, para almacenar objetos complejos. Evita el acceso a los datos; esto es mediante los métodos almacenados en ella. Es más segura ya que no permite tener acceso a los datos (objetos); esto debido a que para poder entrar se tiene que hacer por los métodos que haya utilizado el programador. (Tijerina) Las Bases de Datos orientada a objetos incorporan todos los conceptos importantes del paradigma de objetos:

  - Encapsulación: 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.

❖ **Modelo documentales**

En las BD documentales cada registro se corresponde con un documento, permitiendo la indexación a texto completo y la realización de búsquedas más potentes. Tesaurus es un sistema de índices optimizado para este tipo de bases de datos.

Diseño de base de datos

"Esta etapa consta de tres fases: diseño conceptual, diseño lógico y diseño físico de la base de datos. La primera fase consiste en la producción de un esquema conceptual de los datos, que es independiente de todas las consideraciones físicas. Este modelo se refina después en un esquema lógico eliminando las construcciones que no se pueden representar en el modelo de base de datos escogido (relacional, orientado a objetos, etc.). En la tercera fase, el esquema lógico se traduce en un esquema físico para el SGBD escogido. La fase de diseño físico debe tener en cuenta las estructuras de almacenamiento y los métodos de acceso necesarios para proporcionar un acceso eficiente a la base de datos en memoria secundaria." (Marqués, 2011)[14]

Los objetivos del diseño de la base de datos son: (Marqués, 2011)[14]

- ❖ Representar los datos que requieren las principales áreas funcionales y los usuarios, y representar las relaciones entre dichos datos.

- ❖ Proporcionar un modelo de los datos que soporte las transacciones que se vayan a realizar sobre los datos.

[14] Del libro "Bases de datos" de Mercedes Marqués, 2011, pág. 99

- Especificar un esquema que alcance las prestaciones requeridas para el sistema.

Diseñador de base de datos

"Los diseñadores de la base de datos realizan el diseño de la base de datos, debiendo identificar los datos, las relaciones entre ellos y las restricciones sobre los datos y sobre sus relaciones. El diseñador de la base de datos debe tener un profundo conocimiento de los datos de la empresa y también debe conocer sus reglas de negocio. Las reglas de negocio describen las características principales sobre el comportamiento de los datos tal y como las ve la empresa. Para obtener un buen resultado, el diseñador de la base de datos debe implicar en el proceso a todos los usuarios de la base de datos, tan pronto como sea posible" (Marqués, 2011)[15]

Seguridad de la base de datos (BD)

"La seguridad de la base de datos consiste la protección de la base de datos frente a usuarios no autorizados. Sin unas buenas medidas de seguridad, la integración de datos en los sistemas de bases de datos hace que éstos sean más vulnerables que en los sistemas de ficheros. Sin embargo, los SGBD permiten mantener la seguridad mediante el establecimiento de claves para identificar al personal autorizado a utilizar la base de datos. Las autorizaciones se pueden realizar a nivel de operaciones, de modo que un usuario puede estar autorizado." (Marqués, 2011)[16]

Mapeo de objeto relacional (ORM)

"Consiste en una técnica de programación para convertir datos entre el lenguaje de programación orientado a objetos utilizado y el sistema de base de datos relacional utilizado en el desarrollo de aplicaciones. Esto posibilita el uso de las características propias de la orientación a objetos (básicamente herencia y

[15] Del libro "Bases de datos", de Mercedes Marqués, 2011, pág. 5
[16] Del libro "Bases de datos", de Mercedes Marqués, 2011, pág. 10

polimorfismo)". (Gavin King, 2004)[17]

Framework ORM

"Un Framework ORM no es más que una aplicación (librería) que los desarrolladores implementaron para automatizar el proceso de Mapeo de objeto relacional." (Gavin King, 2004) [18]

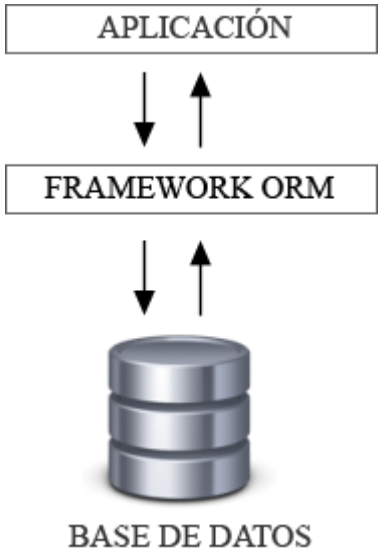

Figura. 1.1 Estructura del proceso de Mapeo de objeto relacional

1.2- Análisis de otras soluciones existentes

Cuando se realiza una investigación se debe buscar soluciones semejantes para compararlas, esta búsqueda se hace tanto a nivel internacional como nacional. En el caso de la investigación presente no se buscaron sistemas de gestión gubernamentales a nivel internacionales debido a que Cuba implementa un modelo de gobierno único.

A nivel nacional se encontró un sistema que gestiona la información sobre la seguridad social de las FAR llamado SISSFAR (Sistema Informático para la Seguridad Social de las FAR), este sistema posee soluciones similares a las del módulo de ACFM pero no cuenta con los procesos que se manejan en los

[17] Del libro "Hibernate in Action", de Gavin King, Christian Bauer, 2004, pág.23
[18] Del libro "Hibernate in Action", de Gavin King, Christian Bauer, 2004, pág.27

departamentos de Inserción Laboral y de Asistencia Militar.

Por otra parte no se encontró ningún sistema nacional que gestione datos similares a los de la dirección de Colaboración. Por estas razones se decide desarrollar una base de datos con su capa de acceso para los módulos de Colaboración y ACFM y de esta forma resolver los problemas de integridad, disponibilidad y centralidad que presentan sus respectivas direcciones.

1.3- Selección de la metodología para el desarrollo de la solución

¿Por qué SXP y no RUP, SCRUM o XP?

Cuba se ha marcado como objetivo social garantizar la transición a entornos libres pero de forma segura. Este proceso conlleva sustituir todas las aplicaciones existentes y de tipo prioritarias por aquellas alternativas que cumplan con las funcionalidades de las anteriores. El uso de una nueva herramienta consume un tiempo previo de estudio que es de vital importancia para la familiarización con la misma.

Todas las actividades que se deben realizar se les debe dedicar bastantes horas, entonces se está frente a un gran problema: ¿cómo lograr la satisfacción de un cliente en un tiempo relativamente corto y garantizando que el producto se ajuste a las normativas de calidad?

Es en este momento donde la metodología Proceso Unificado de Rational (RUP) pierde sentido; puesto que orienta todo el trabajo ingenieril a una sobre documentación del proyecto, que hace poco gestionable el mismo.

RUP es un proceso formal: Provee un acercamiento disciplinado para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo es asegurar la producción de software de alta calidad que satisfaga los requerimientos de los usuarios finales (respetando cronograma y presupuesto). Esta metodología genera mucha documentación en cada una de las iteraciones del proyecto, haciendo muy amplio el proceso de documentación.

Se define entonces el uso de una metodología Ágil que conlleve menos documentación y más tiempo de desarrollo. Se considera por su gran uso y preferencia de autores que las metodologías adecuadas para el software serían extreme programming (XP) y SCRUM.

XP fue la metodología candidata para guiar el proceso ingenieril, puesto que le precedía su alto grado de aceptación por la comunidad internacional de desarrollo ágil, además facilita una documentación más discreta y mayor dinamismo para el desarrollo. SCRUM es entonces la metodología ideal para toda la gestión de proyectos, serviría de soporte para acelerar el dinamismo que se identificó en XP, la identificación de los pequeños sprint (iteraciones) que proporcionan la base de desarrollo iterativo e incremental.

En XP no se usan casos de usos y actores y muchas veces los proyectos los necesitan para facilitar el entendimiento entonces es donde debe también integrarse con SCRUM que si los usa. Al final la fusión de SCRUM y XP sería la metodología antes mencionada y explicada SXP. (José H. Canós, 2010)[19]

1.4- Elección de las tecnologías y herramientas a utilizar.

Sistemas Gestores de Bases de Datos (SGBD)

Existen varias definiciones sobre los SGBD, algunas se muestran a continuación:

- "El software que permite la utilización de los datos almacenados en una (o varias) base(s) de datos por uno o varios usuarios desde diferentes puntos de vista y a la vez es lo que se conoce como un Sistema Gestor de Base de Datos (SGBD)" (Mato, 2006)[20]

[19] Resumen de varios epígrafes del libro "Metodologías Ágiles en el Desarrollo de Software", de José H. Canós, Patricio Letelier y Mª Carmen Penadés,2010
[20] Del libro "Sistemas de Bases de Datos", de Rosa María Matos García, 2006, Pág. 4

- Un Sistema de Gestión de Bases de Datos consiste en un "conjunto de elementos de software con capacidad de definir y utilizar una base de datos". (Capote, 2005)[21]

- "El sistema de gestión de la base de datos es una aplicación que permite a los usuarios definir, crear y mantener la base de datos, y proporciona acceso controlado a la misma." (Marqués, 2011)[22]

Luego de haber estudiado cada uno de los conceptos descritos anteriormente se tuvo afinidad con el concepto brindado por la Dra. Rosa María Matos en su libro Sistemas de Bases de Datos. Se toma este concepto porque es el que mejor describe la intención del autor en el marco de la investigación al referirse al término Sistemas gestores de bases de datos.

Análisis de los principales Sistemas Gestores de Base de Datos.

En los últimos años el desarrollo que ha experimentado el software de base de datos ha sido extraordinario. La facilidad de manejar la información, suministrando a los usuarios las herramientas necesarias que le permitan manipular de forma abstracta los datos con la existencia de múltiples entornos de desarrollo y soluciones informáticas. Dentro de los SGBD más potentes de software propietario se encuentran Oracle y Microsoft SQL Server, y dentro del software libre esta PostgreSQL y MySQL que presenta el uso de una licencia dual o doble licenciamiento.

Selección del Gestor de bases de datos

La universidad de las ciencias informáticas actualmente opta por las tecnologías de software libre por lo que los gestores privativos no son una opción para el desarrollo de la base de datos. Quedarían entonces MySQL y PostgresSQL, de estos dos se escogió PostgresSQL debido a que posee una gran escalabilidad. Es capaz de ajustarse al número de CPUs y a la cantidad de memoria que posee el

[21] Del libro "Introducción a las bases de datos", de Olga Pons Capote, 2005, Pág. 7
[22] Del libro "Bases de datos", de Mercedes Marqués, 2011, pág. 3

sistema de forma óptima, haciéndole capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta (se dice que ha llegado a soportar el triple de carga de lo que soporta MySQL).

PostgreSQL 9.1

Fue el pionero en muchos de los conceptos existentes en el sistema objeto-relacional actual, incluido, más tarde en otros sistemas de gestión comerciales. PostgreSQL es un sistema objeto-relacional, ya que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. (John C. Worsley, 2002)[23]

Lenguaje de Programación

En la actualidad los lenguajes de programación para el desarrollo de aplicaciones Web se clasifican en dos grupos Cliente/Servidor. Dentro de los lenguajes del lado del Servidor los más utilizados: PERL, ASP, JAVA, PHP. Estos lenguajes permiten desarrollar lógica del negocio dentro del servidor, y posibilitan el acceso a las bases de datos y el procesamiento de la información. Como la capa de acceso a datos que se desea desarrollar es parte de una aplicación web que utiliza para su implementación el framework JwebSocket y este a su vez esta implementado en Java, se requiere entonces el uso del lenguaje de programación Java.

Java

Java es un lenguaje de programación y la primera plataforma informática creada por Sun Microsystems en 1995. Es la tecnología subyacente que permite el uso de programas punteros, como herramientas, juegos y aplicaciones de negocios. Java se ejecuta en más de 850 millones de ordenadores personales de todo el mundo y en miles de millones de dispositivos, como dispositivos móviles y aparatos de televisión. (Desarrolladores, 2012)[24]

[23] Resumen de varios párrafos consultados en el libro “Practical Postgrest SQL”
[24] Tomado del sitio oficial de java http://www.java.com/es/download/faq/whatis_java.xml, 10/2/2012

Necesidad del uso de ORM

A la hora de trabajar con programación orientada a objeto y bases de datos siempre surge el problema de concordancia entre este paradigma y el modelo relacional utilizados por las bases de datos. El modelo relacional trata con relaciones, tuplas y conjuntos, es muy matemático por naturaleza. Sin embargo, el paradigma orientado a objetos trata con objetos, sus atributos y relaciones entre objetos.

Cuando se quiere hacer que los objetos sean persistentes utilizando para ello una base de datos relacional, uno se da cuenta de que hay una desavenencia entre estos dos paradigmas, la también llamada diferencia objeto-relacional. Un mapeador objeto-relacional (ORM) nos ayudará a evitar esta diferencia.

Ventajas del uso de la técnica ORM: (Gavin King, 2004)

- ✓ Rapidez en el desarrollo: La mayoría de las herramientas actuales permiten la creación del modelo por medio del esquema de la base de datos, leyendo el esquema, nos crea el modelo adecuado.

- ✓ Abstracción de la base de datos: Al utilizar un sistema ORM, lo que conseguidos es separarnos totalmente del sistema de base de datos que utilicemos, y si en un futuro se tiene que cambiar de motor de bases de datos, tendremos la seguridad de que este cambio no nos afectará a nuestro sistema, siendo el cambio más sencillo.

- ✓ Reutilización: Nos permite utilizar los métodos de un objeto de datos desde distintas zonas de la aplicación, incluso desde aplicaciones distintas.

- ✓ Seguridad: Los ORM suelen implementar sistemas para evitar tipos de ataques como pueden ser los SQL injections.

- ✓ Mantenimiento del código: Nos facilita el mantenimiento del código debido a la correcta ordenación de la capa de datos, haciendo que el mantenimiento del

código sea mucho más sencillo.

- ✓ Lenguaje propio para realizar las consultas: Estos sistemas de mapeo traen su propio lenguaje para hacer las consultas, lo que hace que los usuarios dejen de utilizar las sentencias SQL para que pasen a utilizar el lenguaje propio de cada herramienta.

Desventajas del uso de la técnica ORM

- ✓ Tiempo utilizado en el aprendizaje: Este tipo de herramientas suelen ser complejas por lo que su correcta utilización lleva un tiempo que hay que emplear en ver el funcionamiento correcto y ver todo el partido que se le puede sacar.
- ✓ Aplicaciones algo más lentas: Esto es debido a que todas las consultas que se hagan sobre la base de datos, el sistema primero deberá de transformarlas al lenguaje propio de la herramienta, luego leer los registros y por último crear los objetos.

Análisis de las herramientas para el ORM

Las herramientas ORM han madurado mucho durante los últimos años, muchas se utilizan en sistemas grandes con muchas transacciones por segundo. El rendimiento ha sido un talón de Aquiles de estas herramientas pero muchos de los problemas de rendimiento han sido superados utilizando cache, guardando los datos frecuentemente utilizados en memoria etc. La ganancia en tiempo de desarrollo es muy grande para desechar este tipo de herramientas, en algunos casos no es suficiente el rendimiento y en esos casos la herramienta debe permitir hacer consultas SQL directamente.

De los diferentes ORM para Java se decidió optar por hibernate siendo esa una herramienta que ha conseguido en un tiempo record una excelente reputación en la comunidad de desarrolladores posicionándose claramente como el producto OpenSource (código abierto) líder en este campo gracias a sus prestaciones,

buena documentación y estabilidad. Es valorado por muchos incluso como solución superior a productos comerciales dentro de su enfoque, siendo una muestra clara de su alta calidad.

Hibernate 3.0

"Hibernate es una herramienta de mapeo objeto/relacional para entornos Java. El término de mapeo objeto/relacional (ORM) se refiere a la técnica de mapear una representación de datos desde un modelo de objeto a un modelo de datos relacionales con un esquema basado en SQL. Hibernate no solamente se ocupa del mapeo desde las clases Java a las tablas de las bases de datos (y desde los tipos de datos de Java a los tipos de datos de SQL), sino que también facilita la consulta y recuperación de datos. Esto puede reducir de manera importante el tiempo de desarrollo que se tomaría con el manejo de datos de forma manual en SQL y JDBC." (Gavin King, 2009)[25]

Entorno de desarrollo integrado (IDE)

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes. Entre los principales IDEs disponibles para el lenguaje de programación Java tenemos.

Netbeans 7.1

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. Tiene un gran

[25] Del libro "Documentación de referencia de Hibernate" de Gavin King, 2009, pág. 1

completamiento de código y ofrece un asistente para generar los archivos POJOS necesarios para en uso de Hibernate. (NetBeans, 2011)[26]

Herramientas CASE (*Computer Aided Software Engineering*)

"Las Herramientas CASE, son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas brindan ayuda en todos los aspectos del ciclo de 20 vida de desarrollo del software en tareas como el proceso de realizar el diseño de un proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores. "(Rumbaugh, 2000)[27]

Herramientas CASE seleccionada

Visual Paradigm 6.4

Es una herramienta CASE que utiliza UML (Lenguaje Unificado de Modelado) como lenguaje de modelado, soporta el diagrama entidad-relación. Está disponible en varias ediciones: Enterprise, Professional, Community, Standard, Modeler y Personal, que permite escoger la más idónea para el usuario según su necesidad. Esta herramienta se distribuye bajo licencia gratuita y comercial, entre las principales características que presenta Visual Paradigm están:

Producto de calidad, multiplataforma, soporta aplicaciones web, varios idiomas, fácil de instalar y actualizar, compatibilidad entre ediciones, soporta a miles de usuarios trabajando sobre el mismo proyecto y permite que cada uno vea los cambios realizados en tiempo real. Esta herramienta tiene habilitado UML 2.1, un estándar ampliamente utilizado actualmente en las empresas para el modelado de software. Permita obtener el diagrama Entidad-Relación a partir del diagrama de clases persistentes o viceversa. (Sitio Oficial)[28]

[26] Tomado del sitio oficial de Netbeans "http://netbeans.org/features/index.html",15/2/2012
[27] Del libro "El Proceso Unificado de Desarrollo de Software. Madrid", de G. Booch y J. Rumbaugh, 2000
[28] Tomado del sitio oficial del Visual Paradigm "http://www.visual-paradigm.com",15/2/2012

Herramientas para el control de versiones

Cuando se desarrolla software, y en especial cuando se hace en equipo, es fundamental contar con un mecanismo que permita gestionar el código que producimos. La gestión de los cambios que se producen sobre el software se convierte en algo indispensable cuando varios programadores trabajan sobre el mismo código.

El utilizar una herramienta de este tipo facilita la labor de integrar el código de los diferentes participantes y sobre todo, permite llevar un control de los cambios que se han realizado sobre cada uno de los componentes software. En el desarrollo de la capa de acceso a datos se utiliza el Subversion 1.6 que es el software utilizado a nivel de institución. Para gestionar el contenido del servidor Subversion se utiliza la herramienta RapidSVN 0.12 la cual brinda muchas facilidades para el manejo de versiones por parte del cliente.

Otras herramientas.

PGAdmin lll 1.14

Es una aplicación gráfica para gestionar el gestor de bases de datos PostgreSQL, siendo la más completa y popular con licencia Open Source. Está escrita en C++ usando la librería gráfica multiplataforma wxWidgets, lo que permite que se pueda usar en Linux, FreeBSD, Solaris, Mac OS X y Windows. Es capaz de gestionar versiones a partir de la PostgreSQL 7.3 ejecutándose en cualquier plataforma, así como versiones comerciales de PostgreSQL como Pervasive Postgres, EnterpriseDB, Mammoth Replicator y SRA PowerGres. (pgAdmin, 2011)[29]

Conclusiones del capítulo

Al hacer un estudio sobre los principales conceptos y herramientas vinculadas al marco teórico de la solución propuesta. Se definieron cuáles son los conceptos asumidos por el autor para la investigación. Se escogió la

[29] Tomado de la página oficial de la herramientas “http://www.pgadmin.org/”

metodología y las herramientas necesarias para el desarrollo de bases de datos y capas de acceso a datos.

Como resultado de la investigación que se realizó sobre soluciones de software para la gestión de información gubernamental. Surge la necesidad de desarrollar una base de datos y una capa de acceso a datos para poder almacenar y recuperar la información en los módulos de Colaboración y ACFM.

Capítulo 2. Análisis y desarrollo de la Solución

Introducción

En este capítulo se hará referencia a todas las decisiones y pasos que se realizaron para diseñar e implementar la base de datos y la capa de acceso a datos. Para un mayor entendimiento de la solución, en este capítulo se muestra la estructura del SINAP de Artemisa. Se explica con ejemplos y diagramas los objetivos que persiguen cada una de las direcciones, la forma en que se automatizaran y los procedimientos necesarios para su implementación.

2.1- Objetivos estratégicos de la organización

La Administración Provincial de Artemisa es la encargada de controlar todos los recursos con que cuenta la provincia para lograr este objetivo se dividió en 32 direcciones dentro de las que se encuentran la dirección Colaboración y la dirección de ACFM.

La dirección de Colaboración tiene como objetivos estratégicos.

- Gestiona la información referente a los proyectos de colaboración y donativos puntuales que se reciben en la provincia.
- Se controlan las personas que cumplen misiones en otras naciones, la ayuda que se le brinda a los familiares, los que desertan y los aportes de estas misiones a la provincia.
- Se tiene constancia de los estudiantes extranjeros que residen o cursan estudios en la provincia.
- Gestionar los datos de los viajes que realizan los dirigentes al exterior.
- Controla la información de proyectos nacionales en los que participa la provincia.
- Se tiene constancia de los datos sobre las empresas mixtas que radican en Artemisa.

- Se gestionan las propuestas de hermanamiento con otros países, las delegaciones extranjeras que visitan la provincia y las actividades desarrolladas en el exterior por autoridades que mantienen vínculos con nuestra provincia.

Objetivos estratégicos de la dirección de ACFM.

Objetivos del departamento de Inserción laboral

En este departamento se controlan todos los datos de los jóvenes que cursan el servicio militar activo para en conjunto con el ministerio del trabajo reinsertarlos laboralmente cuando cumplan su misión dentro de las Fuerza Armadas Revolucionarias.

Objetivos del departamento de Registro y Control

Atiende el estado, ubicación y control del potencial por categorías (Ejército Rebelde, Lucha Clandestina, Pensionados de las Fuerzas Armadas Revolucionarias (F.A.R.), Impedidos Físicos de las F.A.R., Familiares de los Mártires, Viudas e Hijos), así como el seguimiento y control de las necesidades económicas, sociales y de salud que plantean.

Objetivos del departamento de Seguridad Social

Tramita y controla la expedición de los medios de pago (chequeras) que reciben los beneficiarios que se acogen al sistema de seguridad social de la F.A.R y las solicitudes de medios de pagos por extravíos.

Objetivos del departamento Asistencia Militar

En este departamento se gestiona toda información referente a los medios que utilizan para darles una adecuada despedida a los combatientes que fallecen.

2.2- Flujo actual de procesos

En estas direcciones se recibe la información de diferentes lugares y por diferentes vías, luego se almacena en formato duro o tablas con formato Excel en diferentes

lugares del disco duro de la pc y se realizan reportes en formato Excel para la dirección del gobierno.

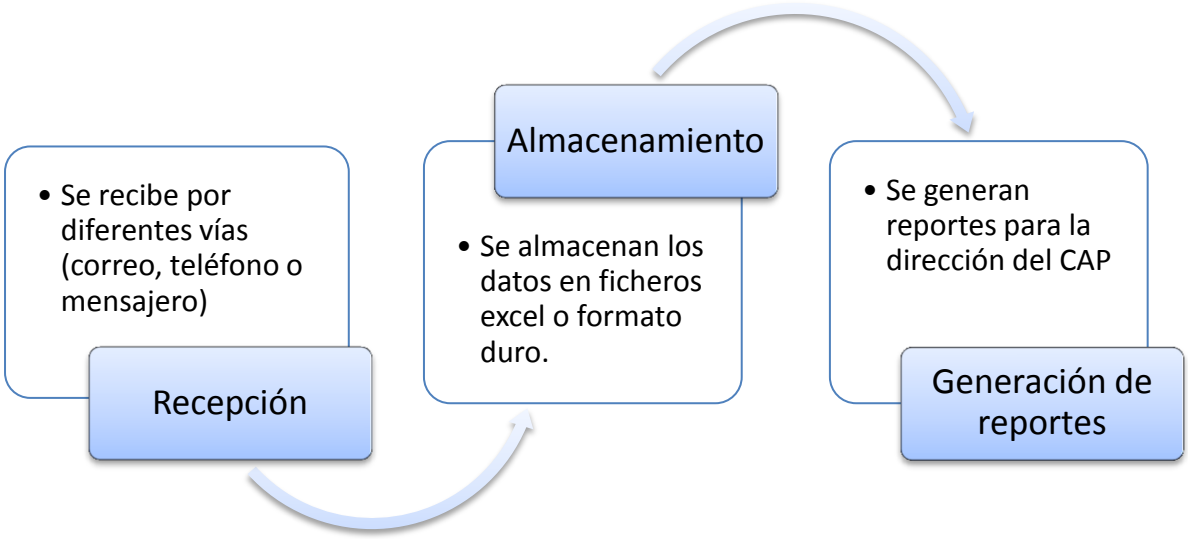

Figura. 2.1 Flujo actual de procesos

Análisis crítico de la ejecución de los procesos

Los procesos de las direcciones de Colaboración y ACFM se realizan manualmente sobre formato duro o ficheros Excel, acumulándose grandes cantidades de información, haciéndose muy complicado organizarla sin que se produzcan duplicados o pérdidas. Los reportes son ineficientes y poco fiables ya que se es casi imposible filtrar una información dentro de todo este volumen de datos.

Procesos objeto de automatización.

En la dirección de Colaboración se automatizaran los siguientes procesos.

- Gestión de los proyectos financiados, en ejecución y finalizados.
- Gestión de todos los datos referentes a los colaboradores, sus sanciones, atención a familiares y los desertores.
- Gestión de información sobre las empresas mixtas.
- Gestión de datos de los estudiantes extranjeros que cursan estudios en la provincia.
- Gestión de los datos de proyectos nacionales en los que participa la provincia.

- Gestión de los datos de visitas extranjeras en la provincia, propuestas de hermandad y actividades desarrolladas en el exterior por autoridades que mantienen vínculos con nuestra provincia.

En la dirección de ACFM se automatizaran los siguientes procesos**.**

- Gestión de los jóvenes dados de baja del servicio militar.
- Gestión de los datos generados por el fallecimiento de un combatiente.
- Gestión de los datos de los beneficiarios y de los tipos de beneficiarios.
- Gestión de los problemas que presentan los beneficiarios.
- Gestión de las pensiones que reciben los beneficiarios.
- Gestión de los chequeos médicos y las visitas que se le realizan a los beneficiarios.

2.3- Requisitos del sistema informativo del gobierno

Volumen de Información a Gestionar

El volumen de información que se maneja en las direcciones de Colaboración y ACFM es grande por lo que la base de datos debe ser capaz de almacenar toda esta información de forma eficiente ya que estos datos estarán en constante cambio.

Características de la información

La información que se maneja en las direcciones de Colaboración y ACFM es prioritaria para el desarrollo de la provincia. En la dirección de Colaboración se manejan muchos datos calculables que necesitan de conocimientos básicos de contabilidad para poder organizar la información de forma correcta en la base de datos.

También se manipula información de cuentas bancarias y gastos de los proyectos de colaboración en la provincia, estos datos deben ser consultados y modificados por una persona debidamente autorizada. En la dirección de ACFM la información

está distribuida en diferentes departamentos teniendo cada uno diferentes clasificaciones de los beneficiarios dependiendo del estado en que se encuentra este dentro del sistema.

Nivel de demanda de servicios concurrentes

De forma general la base de datos permitirá un alto número de servicios concurrentes, pero en la práctica el nivel de demanda exigido para su uso es bajo. La información que se maneja en las direcciones está en constante cambio por lo que los datos son prácticamente variados diariamente desde diferentes lugares.

También es frecuente que los dirigentes del gobierno consulten los datos almacenados para tomar decisiones. Normalmente el número de servicios concurrentes no debe superar los 30 por lo que no supone un problema gestionarlos.

Mecanismos de aseguramiento de la integridad de la información.

Para garantizar la seguridad en la base de datos se cuenta con varios roles dentro de los que tenemos:

Administrador de base de datos: Tiene permitida todas las operaciones, puede conceder privilegios y crear nuevos roles.

Usuario del sistema: Usuario con derecho a crear, borrar, modificar y consultar los datos en la base de datos.

Consultor: Usuarios con derecho solo a consultar los reportes, en este caso se encuentran los directivos del gobierno.

Requisitos funcionales y no funcionales

La captura de requisitos cumple un papel importante en la realización de un proyecto pues especifican las condiciones o capacidades que el sistema debe cumplir y las restricciones sobre las cuales debe operar el producto. El cumplimiento debido de los requisitos logra la satisfacción de ambas partes en el desarrollo de software.

Requisitos no funcionales

Los requisitos no funcionales son propiedades o cualidades que el sistema debe cumplir, estos ni describen información a guardar, ni funciones a realizar, a través de los mismos se debe obtener un producto atractivo, usable, rápido y confiable. Para el desarrollo de la base de datos se recibe como requisito no funcional la necesidad de un servidor de base de datos con conexión de red, con al menos 1 GB de memoria RAM y 160 GB de disco duro y con el gestor de base de datos PostgreSQL 9.1.0.

La capa de acceso a datos necesita un servidor de aplicaciones con conexión de red, con 1 GB de RAM y con más de 60 GB de HDD. Se requiere tener instalado en la PC del servidor la versión 7 de la máquina virtual de Java. La aplicación está diseñada para su funcionamiento constante, requiriendo la conectividad apropiada para asegurar el servicio que brinda durante los siete días de la semana y 24 horas del día. Todas las acciones deben ser disponibles solo para usuarios autenticados por medio de un sistema de seguridad basado en roles para asignar privilegios a los mismos.

Requisitos funcionales

Son las capacidades o condiciones que el sistema debe cumplir.

Para el desarrollo de la base de datos y de la capa de acceso se cuenta con 256 requisitos funcionales (Anexo 2) a implementar. Estos requisitos fueron creados a partir de la descripción de cada uno de los procesos objetos de automatización.

Clasificación de la base de datos propuesta

La base de datos que se desea desarrollar le da soporte a un sistema de gestión de información por lo que los datos están en constante cambio. Por esta razón el tipo de base de datos que se escogió según la variación de sus datos es dinámica.

Para el desarrollo de la base de datos se usó el modelo relacional. Este modelo permite una absoluta libertad en las relaciones entre las tablas, es el más usado en la actualidad para representar problemas reales y para realizar la administración de datos de forma dinámica.

2.4- Diagramas para cada uno de los procesos en las diferentes direcciones.

Para entender de forma organizada y fácil los diagramas de clases persistentes y de entidad relacional se separaron por procesos. A continuación se muestran estos diagramas pertenecientes a cada uno de los procesos.

- Diagrama de clases persistentes para el proceso gestión de los proyectos financiados, en ejecución y finalizados

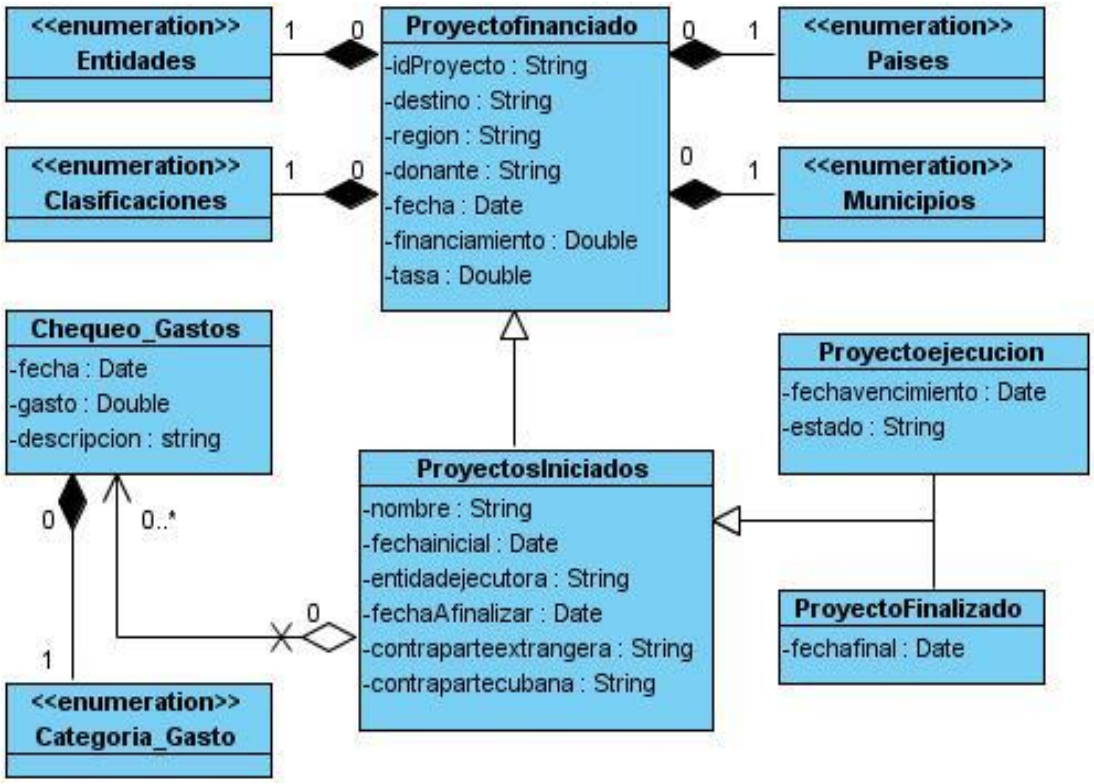

Figura. 2.2 Gestión de los proyectos financiados, en ejecución y finalizados

Aquí se describen los estados por los que pasa un proyecto dentro de la dirección. Los proyectos se financian, luego se separó en iniciado que puede ser en ejecución para los proyecto que se ejecutan actualmente y finalizado para los que ya terminaron. Los proyectos iniciados cuentan con una lista de chequeos de gastos que se le realizan de forma periódica.

- Modelo lógico de datos para el proceso de gestión de proyectos financiados, en ejecución y finalizados.

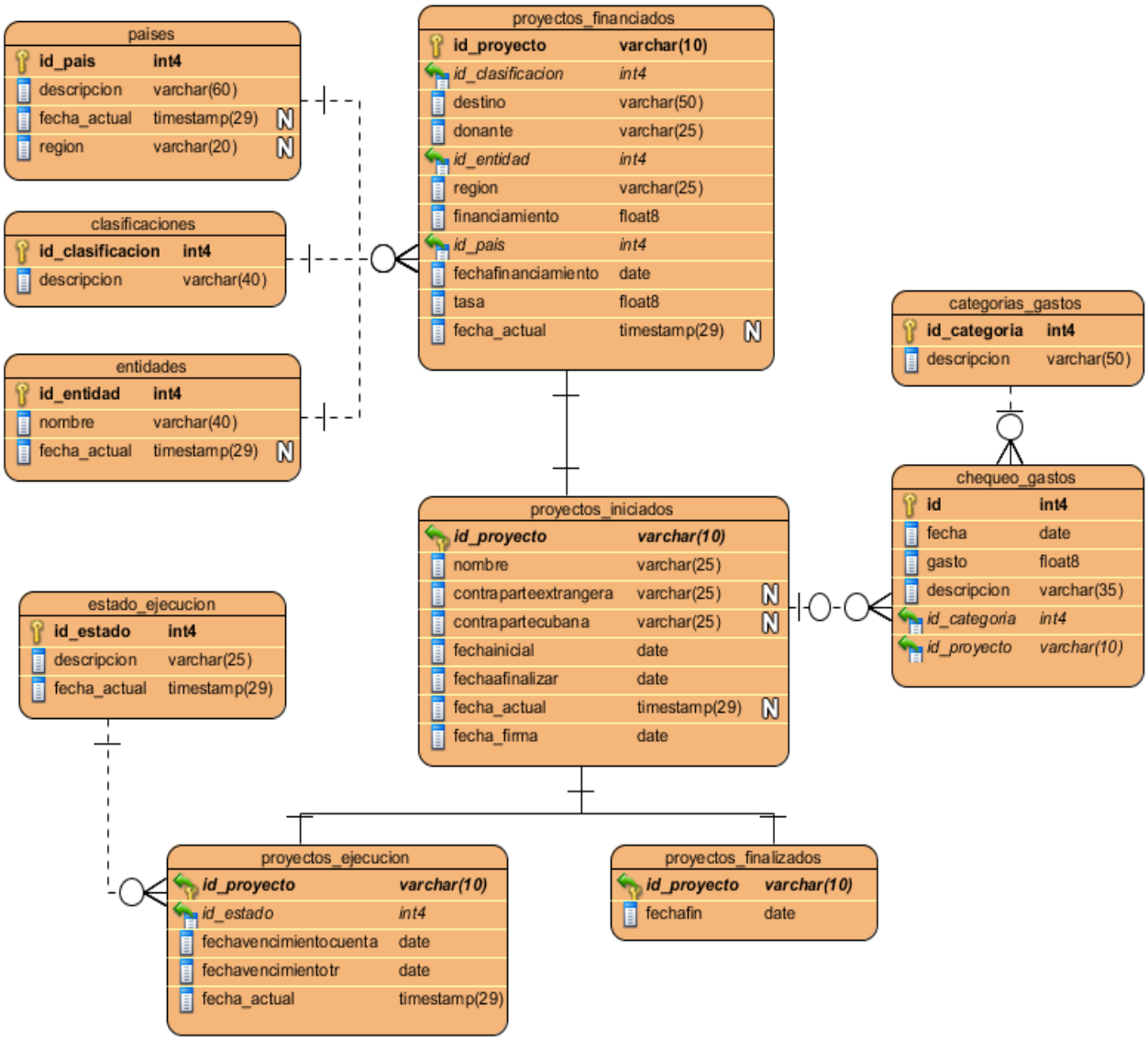

Figura. 2.2 Diagrama ER para proyectos financiados, en ejecución y finalizados

Al financiarse un proyecto se almacena en la tabla proyectos_financiados luego cuando se inician se agregan datos en las tablas proyectos_iniciados y proyectos_ejecucion cogiendo el identificador primario de la tabla proyectos_financiados. Al finalizar un proyecto se deben mantener los datos de este durante 5 años por lo que se almacenan en la tabla proyectos_finalizados los

datos más relevantes de la tabla proyectos_ejecucion. La tabla proyectos_iniciados tiene una relación de uno a muchos con la tabla chequeos_gastos pasando como llave foránea el identificador del proyecto hacia a tabla chequeo_gastos. Las tablas países, clasificaciones, entidades, estado_ejecucion y gategoria_gasto son tablas nomencladores que almacenan las clasificaciones, las entidades, el estado de ejecución de un proyecto y las categorías que puede tener un gasto.

- Diagrama de clases persistentes para el proceso gestión los colaboradores, sus sanciones, atención a familiares y los desertores.

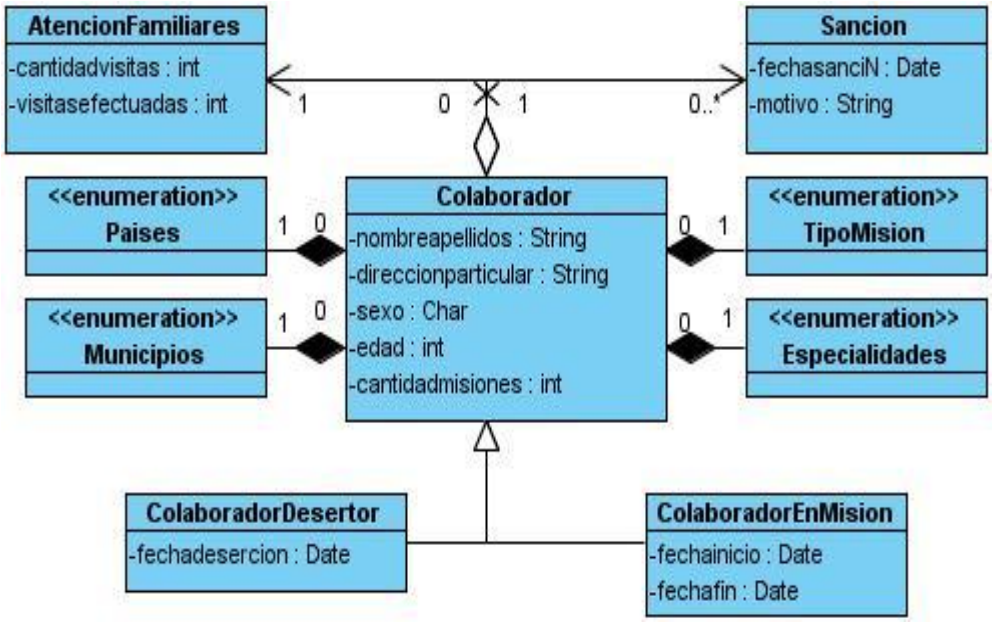

Figura. 2.3 Gestión los colaboradores, sus sanciones, atención a familiares y los desertores.

Describe los datos de los colaboradores y la forma en que se gestionan. Existen dos tipos de colaboradores. Los que se encuentran en misión y los que desertaron. De estos colaboradores se tiene una lista con las sanciones que se le han aplicado y se gestionan los datos referentes a la atención a sus familiares.

- Modelo lógico de datos para el proceso gestión los colaboradores, sus sanciones, atención a familiares y los desertores.

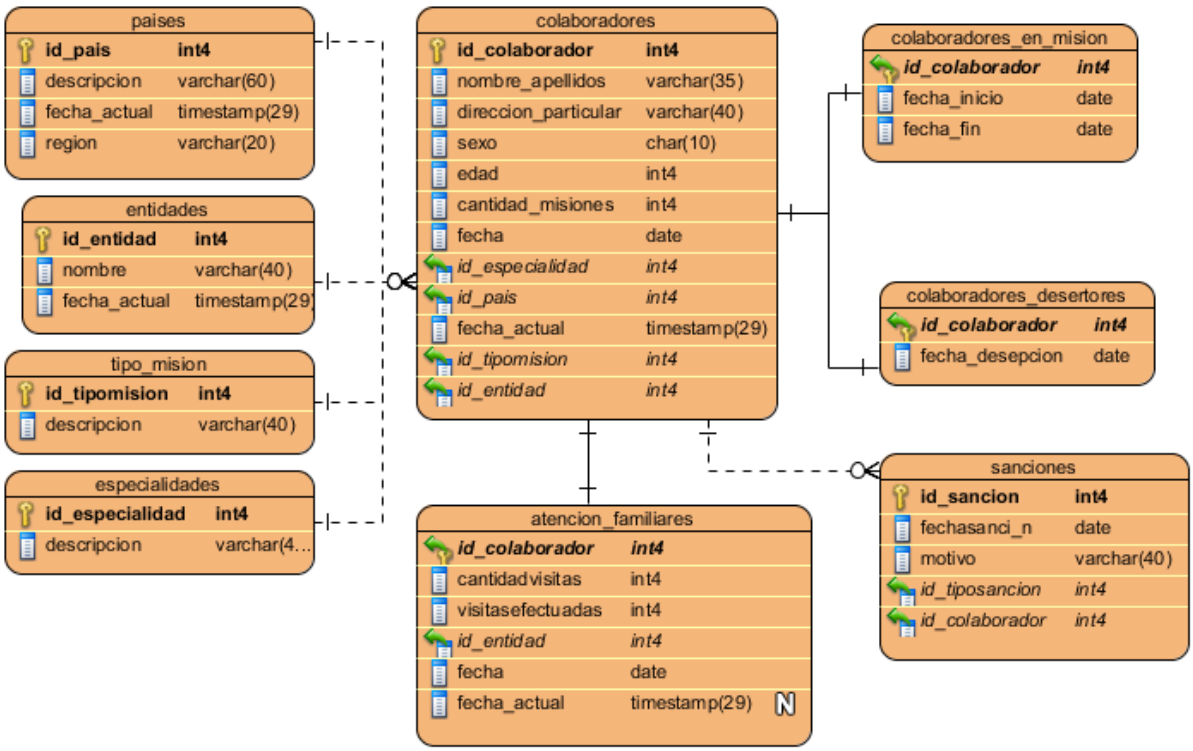

Figura. 2.4 Diagrama ER para la gestión de colaboradores

El proceso tiene como tabla principal la tabla colaboradores de la cual se recibe como llave primaria en las tablas colaboradores_en_mision, colaboradores_desertores y atención_familiares. Los colaboradores se almacenan en la tabla colaboradores luego cuando salen de misión se le agregan datos que se guardan en tabla colaboradores_en_mision, si el colaborador deserta se elimina de colaboradores_en_mision y se guardan datos en colaboradores_desertores. Cuando un colaborador esta en misión se le realiza una atención a los familiares y se almacena en dicha tabla. La tabla colaboradores tiene relaciones de mucho a uno con los nomencladores especialidades, tipo_mision, entidades y países por lo que recibe como llave foránea en identificador de cada una de estas tablas.

- Diagrama de clases persistentes proceso gestión de los estudiantes extranjeros que cursan estudios en la provincia.

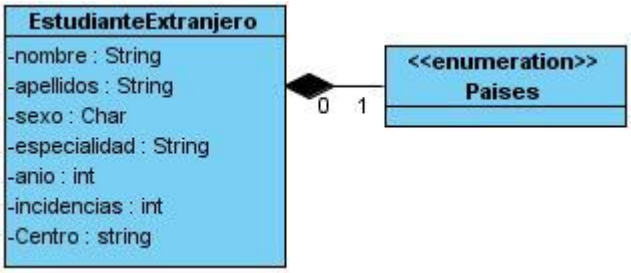

Figura. 2.5 Gestión de los estudiantes extranjeros

A la provincia llegan anualmente una serie de estudiantes extranjeros que cursaran estudios en el país aquí se describen todos los atributos que se necesitan saber acerca de estos.

- Modelo lógico de datos para el proceso gestión de los estudiantes extranjeros que cursan estudios en la provincia.

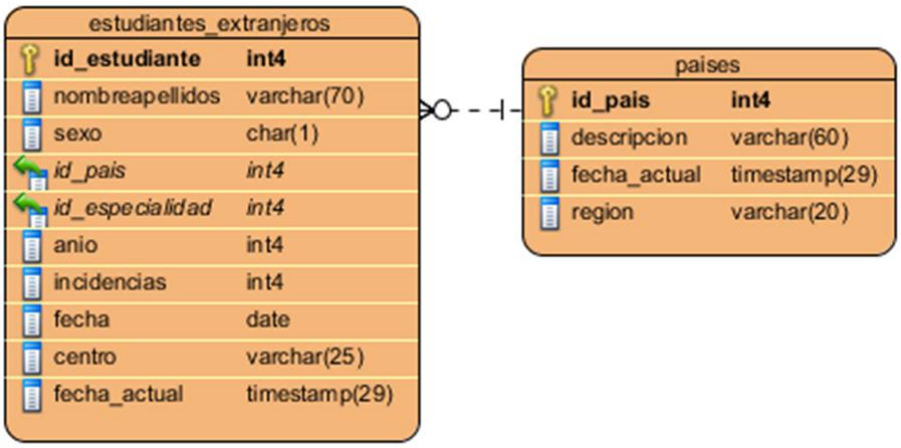

Figura. 2.6 Diagrama ER del proceso de gestión de los estudiantes extranjeros

La tabla estudiantes_extranjeros almacena todos los datos necesarios para controlar los estudiantes extranjeros que estudian en la provincia. Tiene relación con el nomenclador países que identifica a través de su id a que país pertenece el estudiante.

- Diagrama de clases persistentes proceso gestión de información sobre las empresas mixtas.

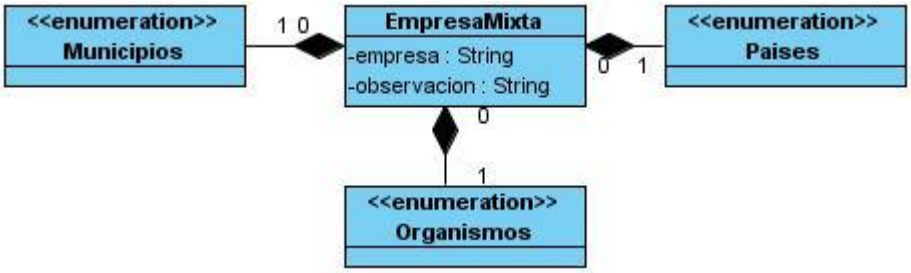

Figura. 2.7 Gestión de información sobre las empresas mixtas

Actualmente en la provincia radican varias empresas de propiedad mixta. Al gobierno le es de suma importancia controlar de que país es el propietario a que organismo pertenece, el municipio en que se encuentra ubicada y una serie de datos más.

- Modelo lógico de datos para el proceso gestión de información sobre las empresas mixtas.

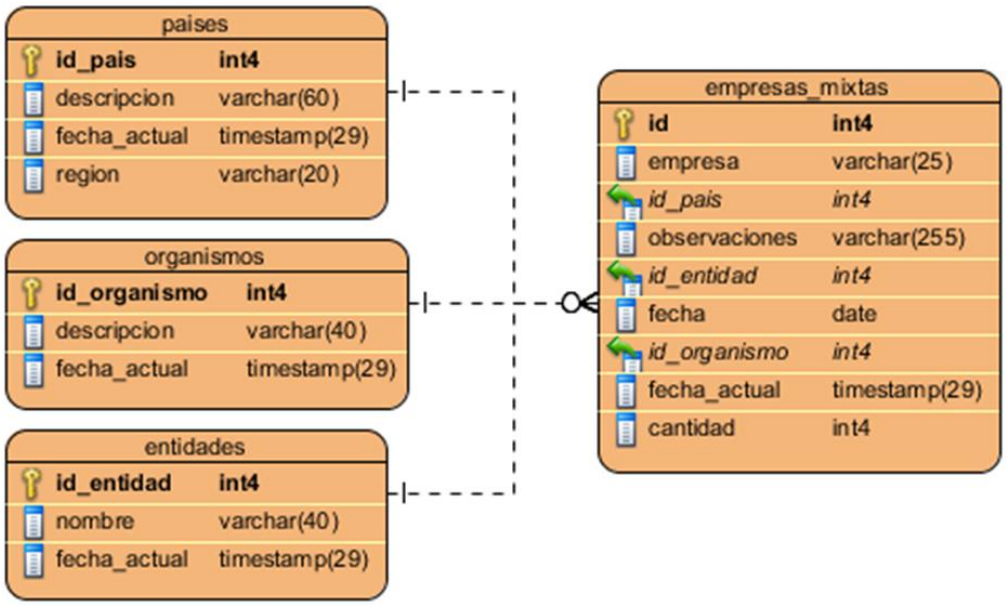

Figura. 2.8 Diagrama ER para la gestión de información sobre las empresas mixtas

En la tabla empresas_mixtas se almacenan los datos referentes a las empresas mixtas que existen en la provincia. Esta tabla tiene relación con los nomencladores países, organismos y entidades los cuales indican en cada tupla a través de su llave primaria el valor que referencia dentro de sus datos.

- Diagrama de clases persistentes para el proceso gestión de los datos de proyectos nacionales en los que participa la provincia

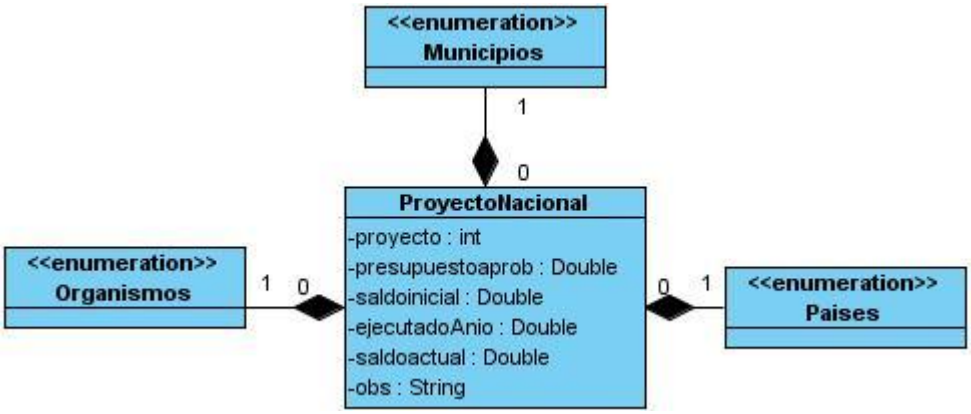

Figura. 2.9 Gestión de los datos de proyectos Nacionales en los que participa la provincia

La provincia de Artemisa colabora en varios proyectos a nivel nacional. Mensualmente la dirección del gobierno realiza un informe para que la dirección de Colaboración controle la ejecución de esos proyectos.

- Modelo lógico de datos para el proceso gestión de los datos de proyectos nacionales en los que participa la provincia.

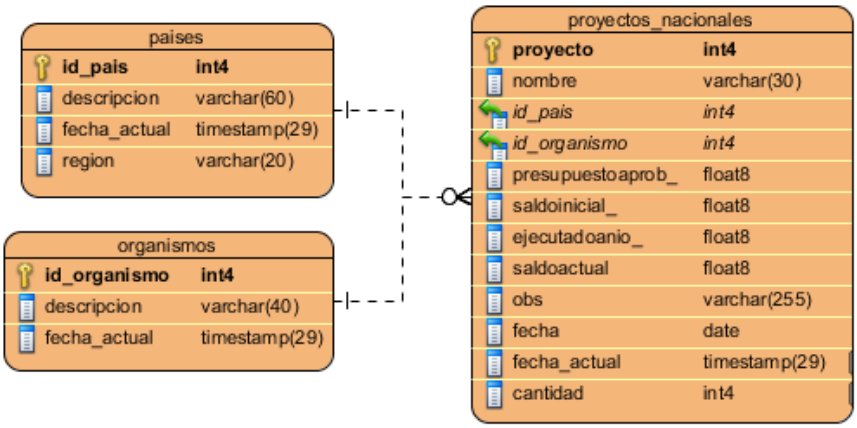

Figura. 2.10 Diagrama ER para la gestión de los proyectos Nacionales

Figura 2.10 Diagrama ER para la gestión de los datos de proyectos Nacionales en los que participa la provincia

La tabla proyectos_nacionales guarda cada uno de los atributos que generan los proyectos nacionales que existen en la provincia. Esta tabla tiene relación con los

nomencladores países y organismos que indican a través de sus llaves foráneas a que tupla se hace referencia en cada caso.

Diagramas de los principales procesos de la dirección de ACFM.

- Diagrama de clases persistentes para el proceso de gestión de los jóvenes dados de baja del servicio militar.

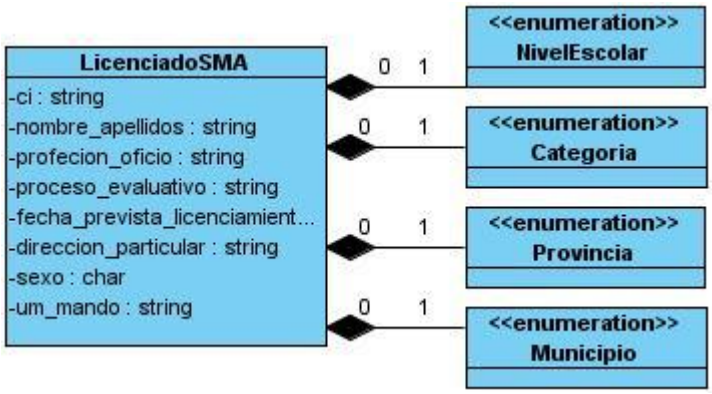

Figura. 2.11 Gestión de los jóvenes dados de baja del servicio militar.

Al termina el servicio militar activo la dirección de ACFM tiene la tarea en conjunto con el ministerio del trabajo de proporcionarle propuestas laborales a los recién salidos. La dirección debe llevar un control estricto de los recién licenciados.

- Modelo lógico de datos para el proceso de gestión de los jóvenes dados de baja del servicio militar.

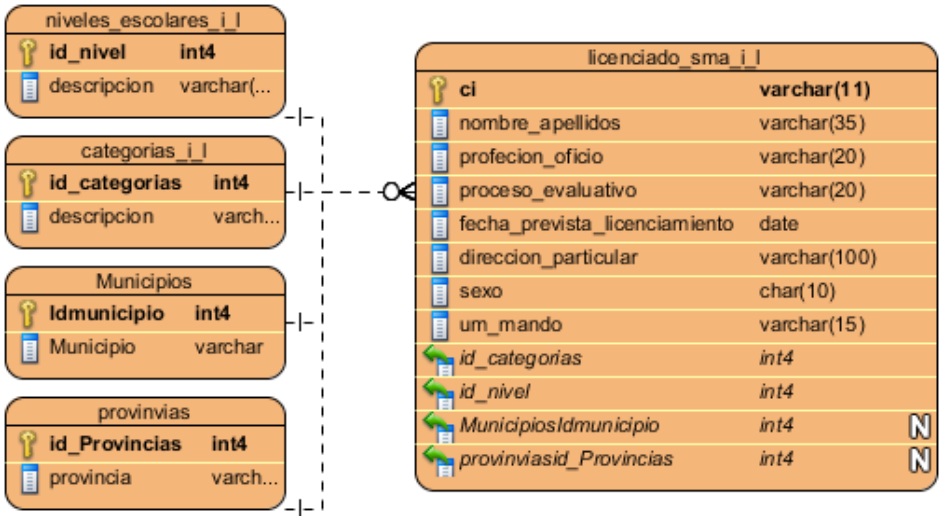

Figura. 2.12 Diagrama ER para la gestión de los jóvenes dados de baja del servicio militar.

En la entidad licenciado_sma_i_l se guardan los datos referentes a los licenciados del servicio militar activo. Para saber el nivel escolar, la categoría, el municipio y la provincia se tiene relación con los nomencladores relativos.

- Diagrama para el proceso de gestión de los datos de los beneficiarios y de los tipos de beneficiarios.

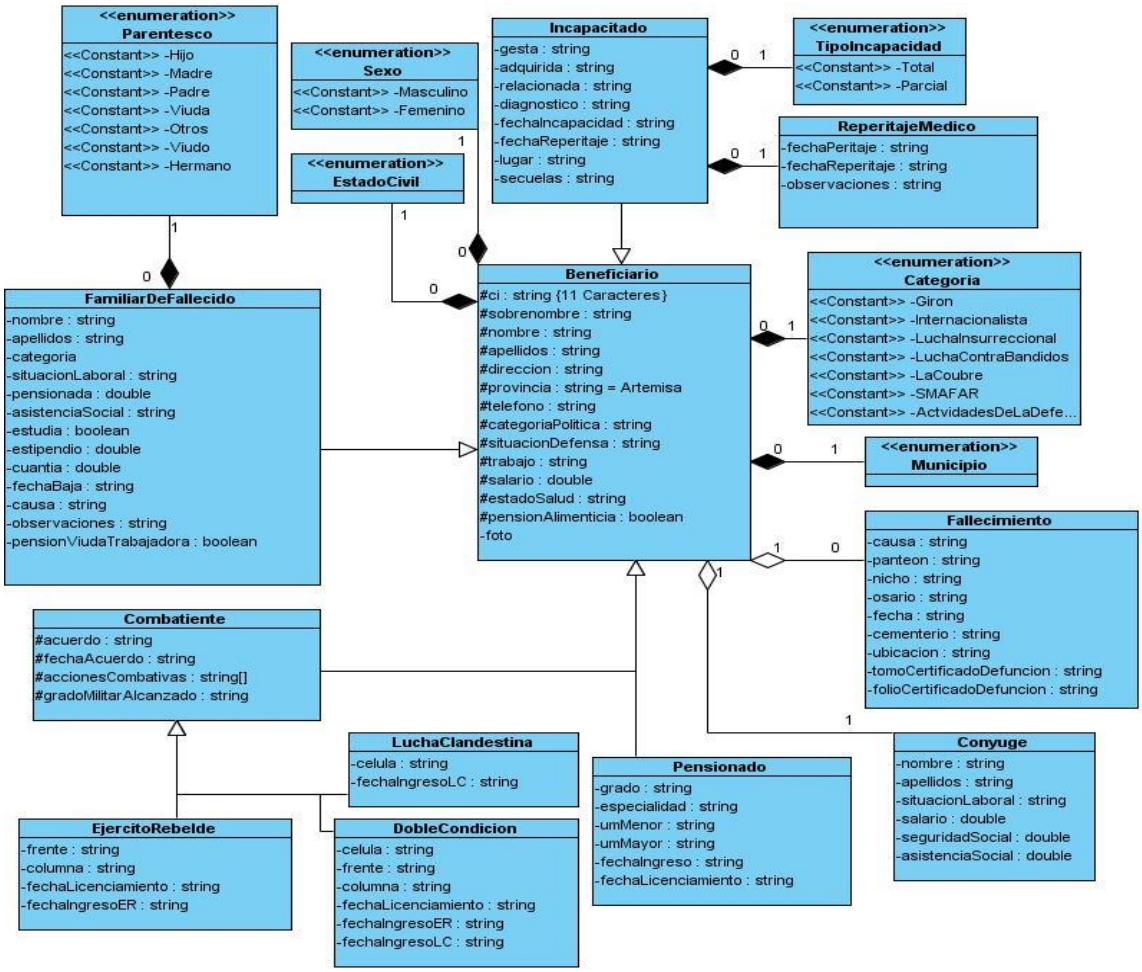

Figura. 2.13 Gestión de los beneficiarios y sus tipos

Los beneficiarios pueden ser de cuatro tipos Pensionados, Familiar de fallecido, Incapacitado y Combatiente. Los combatientes tienen tres tipos Ejército rebelde, Lucha Clandestina y Doble condición, cada una de estas clasificaciones tienen sus propiedades específicas y se relacionan con una serie de clases que describen de forma adecuada sus atributos. El sistema tiene que ser capaz de definir de qué tipo de beneficiario se trata en cada ocasión.

- Modelo lógico de datos para el proceso de gestión de los datos de los beneficiarios y de los tipos de beneficiarios.

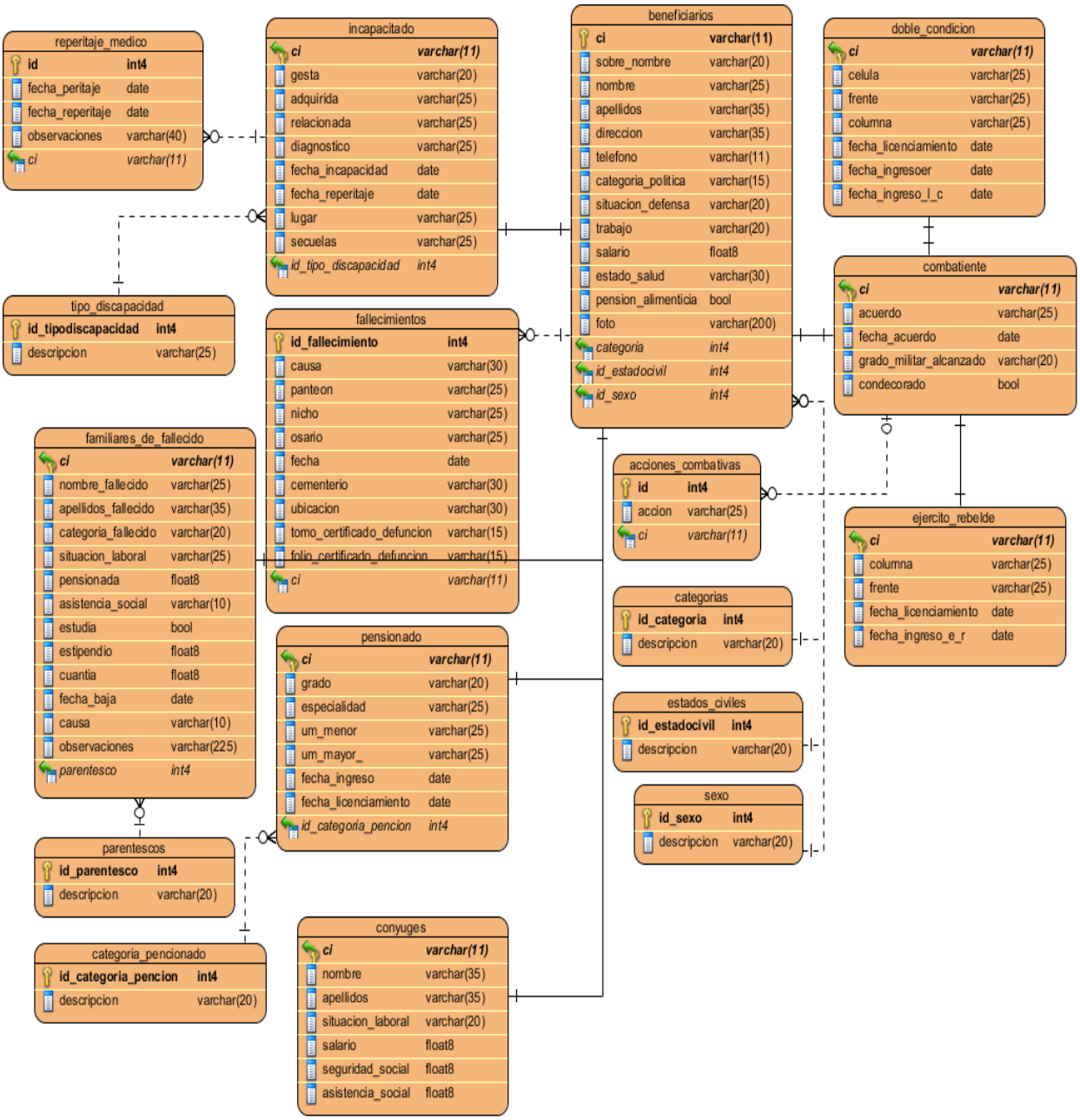

Figura. 2.14 Diagrama ER para la gestión de los beneficiarios y sus tipos

En la tabla beneficiarios se almacenan los datos básicos de los beneficiarios. Para especificar en cada caso otros datos depende del tipo de beneficiario que sea se almacenara en las tablas incapacitados, pensionado, familiar_de_fallecido y combatientes. Este último también necesita que se almacenen datos en las tablas ejercito_rebelde, lucha_clandestina o doble_condicion. Cada una de estas tablas

tiene relación con diferentes entidades en la que necesita almacenar datos específicos.

2.5- Arquitectura del sistema

Arquitectura de software (AS)

"La AS es, a grandes rasgos, una vista del sistema que incluye los componentes principales del mismo, la conducta de esos componentes según se la percibe desde el resto del sistema y las formas en que los componentes interactúan y se coordinan para alcanzar la misión del sistema.**"** (Reynoso, 2004)[30]

Cada módulo del SINAP está diseñado bajo la arquitectura n-capas, teniendo en este caso 4 capas las cuales se muestran en la siguiente figura.

Figura. 2.15 Arquitectura N-Capas

Capa de Presentación: Muestra la interfaz a través de la cual se comunica la aplicación con el cliente. Está desarrollada a partir de la estructura brindada por el framework JavascriptMVC. En esta capa se garantiza toda la validación y la lógica de negocio a través de vistas para que el cliente interactúe con el sistema.

[30] Del libro "Introducción a la Arquitectura de Software", de Carlos Billy Reynoso, 2004, pág. 9

Capa de lógica de Negocio: Esta capa tiene como principal objetivo la comunicación de los datos entre la capa de presentación y la capa de acceso a datos. En este caso lo hace a través de la tecnología jWebsocket aplicando las reglas de negocio definidas.

Capa de Acceso a Datos: En esta capa se implementan las entidades correspondientes a la base de datos. Se usa el framework Hibernate para realizar el mapeo de objeto relacional y de esta forma poder acceder a los datos de la capa de datos de forma orientada a objeto. Para separar las entidades del proceso de persistencia se implementó el patrón DAO ***(Data Access Object)*** con la ayuda del framework Spring. Este framework brinda grandes facilidades para la implementación del DAO e integrarlo con Hibernate.

Capa de Persistencia: Está compuesta por una base de datos montada sobre el gestor de base de datos PostgresSQL. Esta base de datos permite almacenar los datos pertenecientes al módulo que le corresponde.

La solución propuesta abarca las capas de acceso a datos y la de persistencia. Desarrollando la base de datos para los módulos de colaboración y ACFM y una capa de acceso que permite el acceso y recuperación de los datos almacenados en la capa de persistencia.

2.6- Solución para el acceso a los datos desde la aplicación (Capa de acceso a datos)

Estándar de código utilizado

En el desarrollo de la capa de acceso a datos se decidió a nivel de proyecto el uso del estándar de código CamelCase (Anexo 3**)**

Mapeo de objeto relacional (ORM)

Al tener los diagramas de clases que describen los procesos que se van a automatizar, se genera la base de datos y se procede a mapearla. Para mapear la base de datos se crean archivo POJOS (***Plain Old Java Object)*** que son meta clases.

En hibernate existen dos formas de crear los archivos POJOS, con archivos XML y con anotaciones. Con archivos XML tenemos el archivo POJO junto a un archivo XML que sirve de intermediario para enlazar cada atributo de la clase con su equivalente en la base de datos.

El mapeo a través de anotaciones se realiza dentro de la misma entidad, se le especifica por medio de metadatos los atributos que tienen su equivalente en la base de datos. En la aplicación se decidió mapear por medio de anotaciones ya que es un método más sencillo y no genera tantos ficheros cuando manejamos grandes cantidades de entidades.

Para la generación de los archivos POJOS se utilizó el IDE netbeans 7.1.0 que tiene asistente que permite crear las clases POJO con solo configurar la conexión a la base de datos. Los archivos creados tendrían la siguiente estructura.

```
1.  @Entity
2.  @Table(name="Nombre de la tabla"
3.  ,schema="Nombre del escquema"
4.  )
5.  public class Nombre de la clase  implements java.io.Serializable {
6.
7.  /*Atributos de la clase*/
8.  private long  id;
9.
10. public Constructor() {
11. }
12. /*Métodos Get y Set de cada uno de los atributos junto los metadatos
    correspondiente*/
13. }
```

La configuración de la tabla hace uso en este caso de las siguientes anotaciones:

javax.persistence.Entity (@Entity): Esta es la etiqueta JPA para designar a un componente de entidad.

javax.persistence.Table (@Table): Esta etiqueta JPA estándar permite definir el nombre de la tabla en base de datos hacia la que se mapeará la clase. En caso de

no definir esta anotación, se tomará el nombre de la clase como el nombre de la tabla.

javax.persistence.Column (@Column) que es la anotación estándar JPA para definir el nombre de la columna en la base de datos asociada a esta columna. Si no se define esta anotación, el nombre de la columna será igual al nombre de la propiedad. El nombre de la propiedad puede derivarse a partir de un método getter o setter.

Si el campo al que se desea acceder es campo llave primaria en la base de datos se le añade @id en la parte superior del metadatos anterior.

javax.persistence.GeneratedValue (@GeneratedValue) es la anotación estándar para definir estrategias para la generación de llaves primarias. En este caso se hace uso de un campo de identidad. Hibernate define facilidades adicionales al estándar para definir estrategias de generación de identidad; estas facilidades se pueden definir por medio de anotaciones propias de Hibernate.

La herencia entre entidades persistentes

La herencia es uno de los mecanismos más poderosos que nos proporciona la programación orientada a objeto. En hibernate existen diferentes formas de trabajar con la herencia.

- Una tabla por toda la jerarquía de clases: En esta estrategia se genera una sola tabla en la que se guardan todas las instancias del árbol completo de herencia y para diferenciar que tipo de objeto queremos construir designamos diferentes constructores.

- Una tabla por cada subclase (joins): En esta estrategia de herencia se creará una tabla por cada una de las clases que conformen nuestro árbol de herencia. Cada una de las clases y subclases que declaren atributos persistentes,

incluyendo clases abstractas e interfaces, tendrá su propia tabla. Cada una de las entidades tendrá los atributos que lo diferencian de su padre.

- Una tabla por cada clase concreta (uniones): En estrategia se genera una tabla por cada una de las entidades no-abstractas que tenga nuestra aplicación. Sin embargo cada tabla tendrá una columna para cada uno de los atributos de la clase de la entidad que almacena, propios y heredados. O sea, que la tabla mantendrá los atributos de la clase que mapea, junto con los atributos que hereda de su clase padre.

En la aplicación se optó por la estrategia una tabla por cada subclase ya que esta elimina la redundancia de los datos y a su vez se organizan con mayor facilidad. Para poder utilizar esta técnica tiene que realizarse un diseño en la base de datos que lo facilite.

Gestión de las entidades generadas.

Para gestionar todas las entidades y de esta forma tener independencia de la capa de datos, implementamos el patrón DAO que encapsula el acceso a la base de datos. Por lo que cuando la capa de lógica de negocio necesite interactuar con esta, va a hacerlo a través de la API que le ofrece DAO. En la aplicación se implementó una clase genérica que hereda de la clase HibernateDaoSupport implementada en el framework spring.

El HibernateDaoSupport brinda la ventaja de trabajar con las secciones de hibernate de forma automática y sin ningún tipo de gestión por parte del administrador de bases de datos. En la clase GenericDao se crearon las funciones necesarias para trabajar de forma general con todas las entidades, ahora si hiciera falta alguna funcionalidad para una entidad específica se implementa una nueva clase dao que herede del dao genérico. Dentro de las funciones que tiene el DAO genérico tenemos.

Nombre	Parámetros	Retorno	Descripción

saveOrUpdate	Object object		Este método recibe como parámetro un objeto, verifica que el objeto este almacenado en la base de datos, si esta lo actualiza sino lo salva.
save	Object object		Salva el objeto pasado por parámetro.
find	Class<T> entityClass	List<T>	Retorna una lista de todas las tuplas de la base de datos mapeadas como objetos, se le pasa como parámetro el tipo de dato que se desea buscar.
findByHQL	String hqlQuery	List<T>	Retorna una lista genérica con el resultado de una consulta HQL pasada por parámetros.
findByPK	Class<T>, Serializable	Object	Busca y retorna un objeto a través de la llave primaria y el tipo.
delete	Object object		Elimina un objeto pasado por parámetro.
deleteAll	Collection<T> collection		Elimina todas las tuplas de una tabla recibiendo como parámetro una lista de objetos.
findByParamAndValue	Class<T> entityClass, String param, Object value	List<T>	Devuelve la lista de objetos que cumplan con una condición a partir del tipo, nombre de un campo y su valor.

findByParamsAndValues	Class<T> entityClass, String[] params, Object[] values	List<T>	Retorna todos los objetos que cumplan con una serie de atributos y valores pasados por parámetro.

Tabla 2.1 Descripción de la clase GenericDao

Organización por paquete

Para tener mayor organización en la aplicación se separó la lógica de negocio de las entidades quedando separadas en diferentes paquetes.

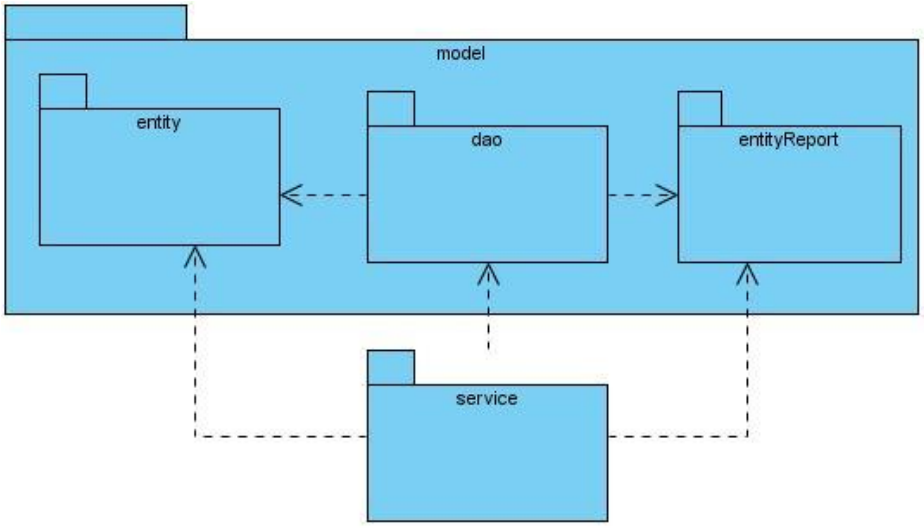

Figura. 2.16 Diagrama de paquetes genérico para los dos módulos

Breve descripción de los paquetes.

Paquete entity: En este paquete se encuentran todas las entidades del modelo.

Paquete dao: En este paquete se encuentran las entidades que se encargan de gestionar las entidades del paquete entity, también devuelven los datos necesarios para los reportes.

Paquete entityReport: Las entidades que se encuentran en este paquete tienen los campos necesarios mostrar en cada reporte, se crearon para facilitar el trabajo a la hora de mostrar la información.

Paquete services: Aquí se encuentran los servicios que gestionan las entidades del paquete entity a través de los elementos del paquete dao.

Diagramas de componentes para los módulos de Colaboración y ACFM

Paquete dao del módulo de Colaboración

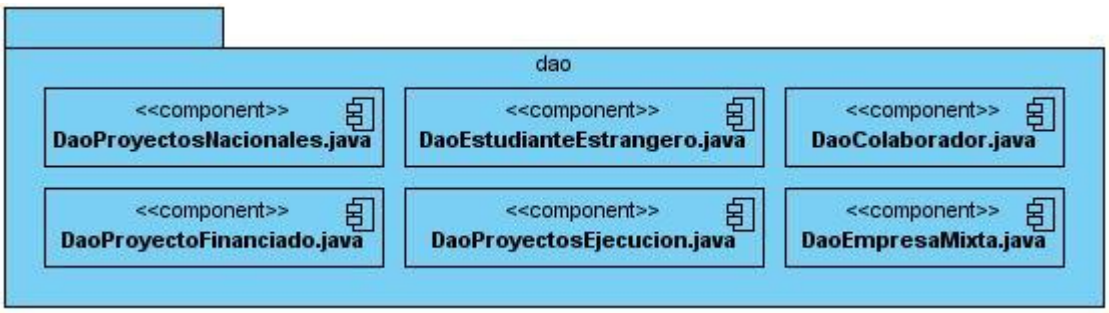

Figura. 2.17 Paquete dao del módulo de Colaboración

Descripción de los componentes del paquete dao del módulo de Colaboración (Anexo 4)

Paquete entity del módulo de Colaboración

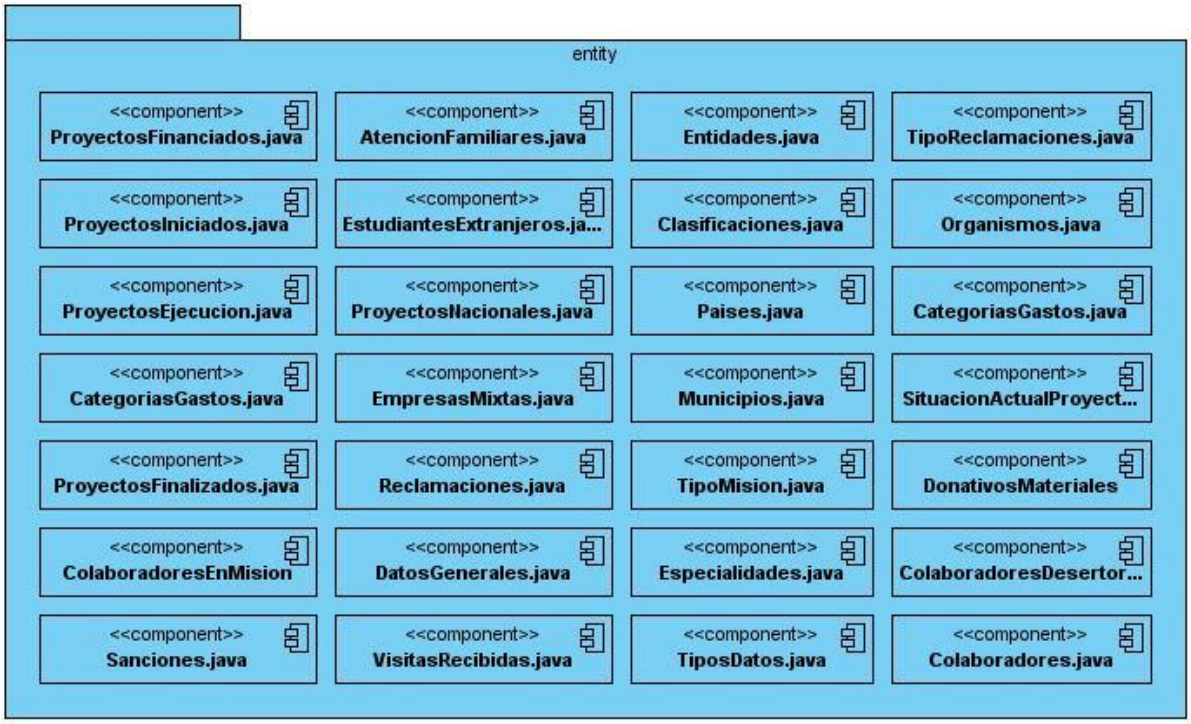

Figura. 2.18 Paquete entity del módulo de Colaboración

Descripción de los componentes del paquete entity del módulo de Colaboración (Anexo 5)

Componentes del paquete entityReport del módulo Colaboración

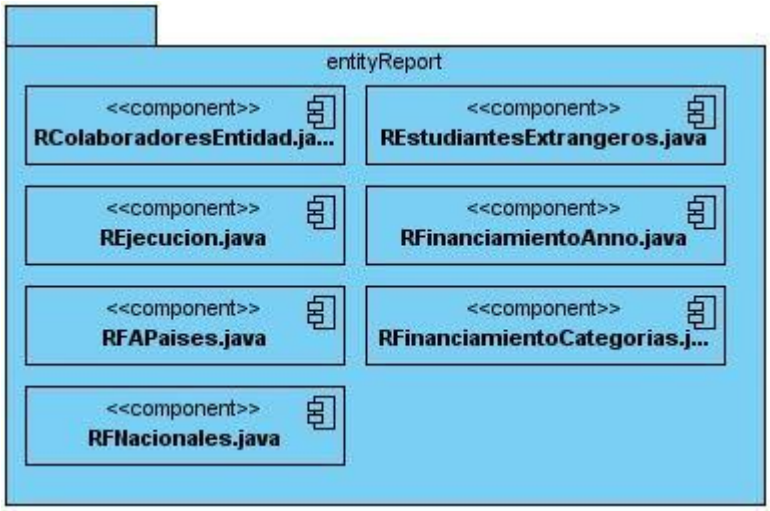

Figura. 2.19 Paquete entityReport del módulo Colaboración

Descripción de los componentes del paquete entityReport del módulo Colaboración (Anexo 6)

Componentes del módulo ACFM

Componentes del paquete dao del módulo ACFM

Figura. 2.20 Paquete dao del módulo de ACFM

Descripción de los componentes del paquete dao de la dirección ACFM (Anexo 7)

Componentes del paquete entity del módulo ACFM

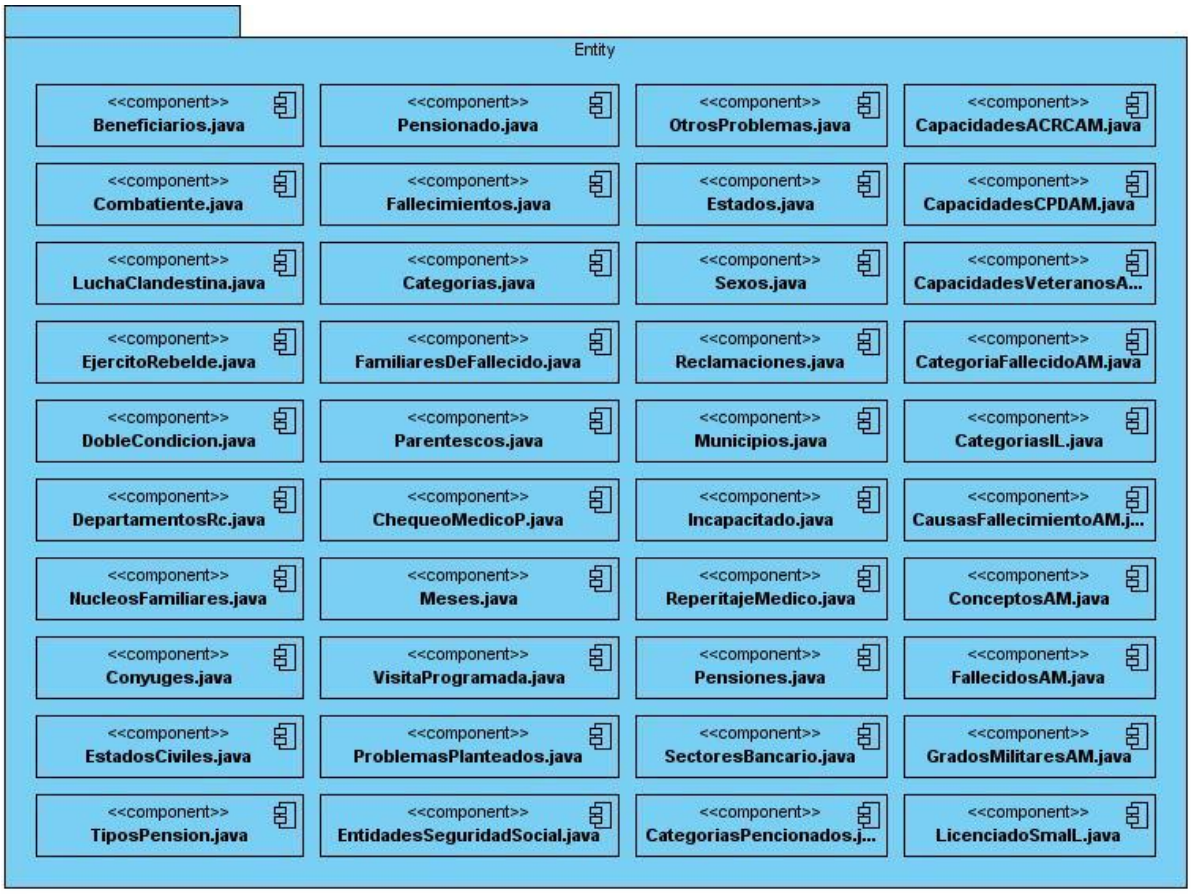

Figura. 2.21 Paquete entity del módulo de ACFM

Descripción de los componentes del paquete entity de la dirección ACFM (Anexo 8)

Componentes del paquete entityReport del módulo ACFM

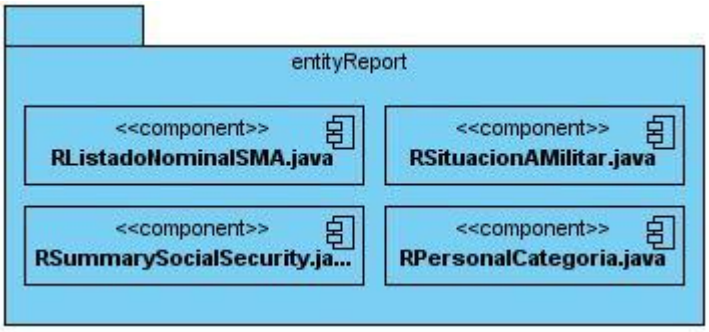

Figura. 2.22 Paquete entityReport del módulo ACFM

Descripción de los componentes del paquete entityReport del módulo ACFM (Anexo 9)

Conclusiones del Capítulo.

La propuesta de solución expuesta en este capítulo abarcó varios aspectos necesarios para el completo desarrollo de la base de datos y la capa de acceso a datos. Después de un estudio sobre el flujo actual de procesos en las direcciones de Colaboración y ACFM se resumieron los principales procesos objeto de automatización, para de esta forma definir cada uno de los requisitos funcionales que debe cumplir la solución desarrollada.

Se obtuvo una base de datos y una capa de acceso para los módulos de Colaboración y ACFM como propuesta de solución de la presente investigación. La implementación de la solución se realizó basándose en todo momento de los diagramas y procedimientos expuesto en el capítulo. La solución quedo desarrollada completamente implementando cada uno de los requisitos funcionales especificados.

Capítulo 3: Validación de la solución propuesta

Introducción

En este capítulo se realizará la validación y las pruebas necesarias para garantizar el buen funcionamiento del diseño y la implementación de la base de datos y su capa de acceso a datos para los módulos de Colaboración y ACFM.

3.1- Validación teórica del diseño de la BD.

Integridad de los datos

En una base de datos la integridad puede perderse de muchas maneras diferentes para garantizar la no contradicción entre los datos almacenados existen una serie de reglas de integridad que la estructura deben cumplir para garantizar que son correctos.

Al definir cada atributo sobre un dominio se impone una restricción sobre el conjunto de valores permitidos para cada atributo. A este tipo de restricciones se les denomina restricciones de dominios. Hay además dos reglas de integridad muy importantes que son restricciones que se deben cumplir en todas las bases de datos relacionales y en todos sus estados (las reglas se deben cumplir todo el tiempo). Estas reglas son la regla de integridad de entidades y la regla de integridad referencial. (Marqués, 2011)[31]

Regla de integridad de entidades: Se aplica a las claves primarias de las relaciones base: ninguno de los atributos que componen la clave primaria puede ser nulo. (Marqués, 2011)[31]

Regla de integridad referencial: Se aplica a las claves ajenas: si en una relación hay alguna clave ajena, sus valores deben coincidir con valores de la clave primaria a la que hace referencia, o bien, deben ser completamente nulos. (Marqués, 2011)[31]

[31] Del libro “Bases de datos”, de Mercedes Marqués, 2011, pág. 25

Revisión de integridad en el diseño de la base de datos de Colaboración y ACFM

En la base de datos para el módulo ACFM se encontraron los siguientes errores.

- En las tablas Problemas_Planteados, Combatiente y Fallecidos_A_M se encontraron que pueden ser nulos y no estaba requerido el campo como Allows nulls.

- Las relaciones entre las tablas Beneficiarios - Nucleos_Familiares y Beneficiarios - Pensiones tenían la cardinalidad de uno a muchos y realmente la relación debe ser de uno a uno.

- En las relaciones Beneficiarios - Reclamaciones y Beneficiarios – Conyuges se encontraron problemas a la hora de eliminar porque no tenían habilitado el eliminar en cascada, lo que ocasionaba errores a la hora de eliminar un beneficiario que tuviera relaciones con alguna de estas tablas.

En la base de datos perteneciente al módulo de Colaboración se encontraron los siguientes errores.

- En la tabla Proyectos_Nacionales se tenía definido como id el nombre del proyecto, provocando problemas al existir proyectos con el mismo nombre. Para solucionarlo se creó un nuevo campo que genera un id autoincrementable para identificar cada tupla.

- Las tablas Proyectos_Iniciados y Estudiantes_Extranjeros tenían campos que pueden ser nulos y no estaba requerido el campo como Allows nulls.

- Las relaciones Colaboradores - Sanciones, Colaboradores - Atencion_Familiares y Proyectos_Iniciados - Chequeo_Gastos no tenían marcada la opción eliminar en cascada.

Normalización de la base de datos

La normalización es una técnica para diseñar la estructura lógica de los datos de un sistema de información en el modelo relacional, desarrollada por E. F. Codd en

1972. Es una estrategia de diseño de abajo a arriba: se parte de los atributos y éstos se van agrupando en tablas según su afinidad. Aquí no se utilizará la normalización como una técnica de diseño de bases de datos, sino como una etapa posterior a la correspondencia entre el esquema conceptual y el esquema lógico, que elimine las dependencias entre atributos no deseadas. En la mayoría de las ocasiones, una base de datos completamente normalizada no proporciona la máxima eficiencia. (Marqués, 2011)[32]

Primera forma normal

Una tabla está en primera forma normal (1fn) si, y sólo si, todos los dominios de sus atributos contienen valores atómicos, es decir, no hay grupos repetitivos. Un grupo repetitivo es un atributo que puede tener múltiples valores para cada fila de la relación. Son los atributos que tienen forma de tabla. Si una tabla no está en 1fn, hay que eliminar de ella los grupos repetitivos. La forma de eliminar los grupos repetitivos consiste en poner cada uno de ellos como una tabla aparte, heredando la clave primaria de la tabla en la que se encontraban. La clave primaria de esta nueva tabla estará formada por la combinación de la clave primaria que tenía cuando era un grupo repetitivo y la clave primaria que ha heredado en forma de clave ajena. Se dice que conjunto de tablas se encuentra en 1fn si ninguna de ellas tiene grupos repetitivos.

Segunda forma normal

Una tabla está en segunda forma normal (2fn) si, y sólo si, está en 1fn y, además, cada atributo que no forma parte de la clave primaria es completamente dependiente de la clave primaria. La 2fn se aplica a las tablas que tienen claves primarias compuestas por dos o más atributos. Si una tabla está en 1fn y su clave primaria es simple (tiene un solo atributo), entonces también está en 2fn. Las tablas que no están en 2fn pueden sufrir anomalías cuando se realizan actualizaciones sobre ellas. Para pasar una tabla en 1fn a 2fn hay que eliminar las dependencias

[32] Del libro “Bases de datos”, de Mercedes Marqués, 2011, pág. 102

parciales de la clave primaria. Para ello, se eliminan los atributos que son funcionalmente dependientes y se ponen en una nueva tabla con una copia de su determinante. Su determinante estará formado por los atributos de la clave primaria de los que depende.

Tercera forma normal

Una tabla está en tercera forma normal (3fn) si, y sólo si, está en 2fn y, además, cada atributo que no forma parte de la clave primaria no depende transitivamente de la clave primaria. La dependencia x −→ z es transitiva si existen las dependencias x −→ y, y −→ z, siendo x, y, z atributos o conjuntos de atributos de una misma tabla. Aunque las relaciones en 2fn tienen menos redundancias que las relaciones en 1fn, todavía pueden sufrir anomalías frente a las actualizaciones. Para pasar una relación en 2fn a 3fn hay que eliminar las dependencias transitivas. Para ello, se eliminan los atributos que dependen transitivamente y se ponen en una nueva relación con una copia de su determinante (el atributo o atributos no clave de los que depende).

Forma normal de Boyce-Codd

Una tabla está en la forma normal de Boyce-Codd (bcfn) si, y sólo si, todo determinante es una clave candidata. La 2fn y la 3fn eliminan las dependencias parciales y las dependencias transitivas de la clave primaria. Pero este tipo de dependencias todavía pueden existir sobre otras claves candidatas, si éstas existen. La bcfn es más fuerte que la 3fn, por lo tanto, toda tabla en bcfn está en 3fn. La violación de la bcfn es poco frecuente ya que se da bajo ciertas condiciones que raramente se presentan. Se debe comprobar si una tabla viola la bcfn en caso de tener dos o más claves candidatas compuestas que tienen al menos un atributo en común.

La base de datos para los módulos de Colaboración y ACFM está en tercera forma normal ya que cumple con todos los requisitos exigidos para estar en esta fase. El diseño no se llevó hasta la forma Boyce-Codd porque al tener la base de datos tan

normalizada se hace más difícil agrupar la información a la hora de obtenerla. Al estar en tercera forma normal la base de datos tienen eliminada la redundancia de datos, lo valores atómicos y las dependencias transitivas.

Análisis de la seguridad de la base de datos

La seguridad de una base de datos es un atributo importante para mantener la integridad y disponibilidad de la información. La seguridad garantiza la inmunidad de las bases de datos a ataques del exterior. Para mantener el esquema de la base de datos a salvo se realizó un resguardo (***backup***) y se guardó en un lugar seguro. A través del gestor de base de datos se asignaron un grupo de roles para permitir el acceso a insertar modificar, eliminar y consultar los datos.

Una vez integrada la base de datos a la aplicación el administrador de la base de datos es el encargado de asignar los roles dependiendo del usuario. También podrá asignar mediante el gestor las direcciones ip pueden acceder a la base de datos.

3.2- Validación funcional de la base de datos y la capa de acceso a datos.

Pruebas de volumen y rendimiento

Las pruebas de volumen son pruebas típicas de entornos que utilicen bases de datos. Las mismas se realizan para analizar el comportamiento del sistema o base de datos con volúmenes de datos almacenados lo más similar posible a los esperados en la explotación real del sistema. Para el sistema en cuestión la BD se probó con datos aleatorios generados a través de un método implementado que inserto 100000 objetos en la base de datos.

```
GenericDao dao=new GenericDao();
for (int i = 0; i < 100000; i++) {
    Paises p=new Paises(i+"t");
    dao.save(p);
}
```

Figura. 3.1 Método utilizado para insertar grandes cantidades de objetos de tipo países

Al introducir los datos no se presentaron problemas de límite de capacidad, ni de volumen de datos. Las llaves autogeneradas no se salieron del rango especificado, ni se detectaron problemas con los tipos de datos definidos en el paso de diseño. Lo anteriormente planteado garantiza que el gestor utilizado, el diseño de las estructuras de la base de datos implementadas y la capa de acceso a datos soportan completamente el almacenamiento de altos niveles de información. Para validar la rapidez de respuesta de la capa de acceso se midió el tiempo al insertar 100000 objetos en la tabla países demorándose 2 minutos con 40 segundos.

Prueba de Carga

Una prueba de carga se realiza generalmente para observar el comportamiento de una aplicación bajo una cantidad de peticiones esperada. Esta carga puede ser el número esperado de usuarios concurrentes utilizando la aplicación y que realizan un número específico de transacciones durante el tiempo que dura la carga. Esta prueba puede mostrar los tiempos de respuesta de todas las transacciones importantes de la aplicación.

Esta prueba se utilizó para ver los tiempos de respuesta de la base de datos junto con su capa de acceso. Se ejecutaron una serie de peticiones a la base de datos a través de criterias creadas en la capa de acceso que se convierten en consultas SQL para ejecutarse en la base de datos. También se ejecutaron consultas en lenguaje HQL a través de un método implementado.

Nombre	Listar fallecidos
Objetivo	Obtener un listado de fallecidos de forma rápida y cumpla con las restricciones impuestas.
	Se ejecutó un criteria que devolvió un listado con 10 fallecidos agrupados por categoría, por causa de fallecimiento y por grado militar. Como condiciones tiene que la fecha de fallecimiento

Descripción	tiene que estar dentro de un rango introducido y que el municipio sea el requerido por parámetro, luego cuenta la cantidad de tuplas que se agrupen en cada caso y devuelve el resultado.

Código	`DetachedCriteria criteria = DetachedCriteria.forClass(FallecidosAM.class)`
	`.setProjection( Projections.projectionList()`
	`.add(Projections.groupProperty("categoriaFallecidoAM"))`
	`.add( Projections.groupProperty("causasFallecimientoAM"))`
	`.add( Projections.groupProperty("gradosMilitaresAM"))`
	`.add( Projections.rowCount(),"cantidad"));`
	`criteria.add( Restrictions.between("fechaFallecimiento",`
	`FechaInicial, FechaFin))`
	`.add(Restrictions.eq("municipios", municipio));`

Resultados	El resultado de la prueba fue satisfactorio obteniéndose el listado esperado en un tiempo de 200 ms

Tabla 3.1 Caso de prueba con criteria

Nombre	Listar RFAPaises (Reporte de financiamiento actual por países)
Objetivo	Obtener un listado que representa el reporte de financiamiento actual por países de forma rápida y cumpla con las restricciones impuestas.
Descripción	Se ejecutó una consulta HQL que devolvió un listado de 20 objetos de tipo RFAPaises que tiene los atributos de los reportes de financiamientos actuales de los países. La consulta contiene otra su consulta que suma los gastos de un proyecto. También para que el reporte se pueda obtener el día especificado se compara el mes y el año para que devuelva los que están financiados hasta ese mes.

Código	`String query="Select new RFAPaises"`
	`+ "(pe.paises.descripcion,count(*),"`
	`+ "sum(pe.financiamiento),(select sum(cg.gasto) "`
	`+ "FROM ChequeoGastos as cg where cg.proyectosIniciados.paises"`
	`+ ".descripcion=pe.paises.descripcion and ((YEAR(cg.fecha)<"`
	`+anno+") or ((YEAR(cg.fecha)="+anno+") and "`
	`+ "(MONTH(cg.fecha)<="+mes+"))))) FROM ProyectosEjecucion as pe "`
	`+ "GROUP BY pe.paises.descripcion";`
	`return findByHQL(query);`

Resultados	El resultado de la prueba fue satisfactorio obteniéndose el listado esperado en un tiempo de 250 ms

Tabla 3.2 Caso de prueba con HQL

Pruebas de Sistema

Las pruebas del sistema deben enfocarse en requisitos que puedan ser tomados directamente de casos de uso, reglas y funciones de negocios. El objetivo de estas pruebas es verificar el ingreso, procesamiento y recuperación apropiado de datos, y la implementación apropiada de las reglas de negocios. Para realizar pruebas funcionales se agruparon los requisitos funcionales correspondientes a los procesos de salvar, actualizar, buscar, buscar por llave primaria y eliminar.

Se realizaron 3 iteraciones corrigiéndose en cada caso los problemas detectados en la anterior. En la gráfica que se muestra a continuación se puede ver el porciento de requisitos funcionando en cada una de las iteraciones. Como se puede observar en la iteración 3 todos los requisitos asociados a los procesos mostrados funcionan correctamente.

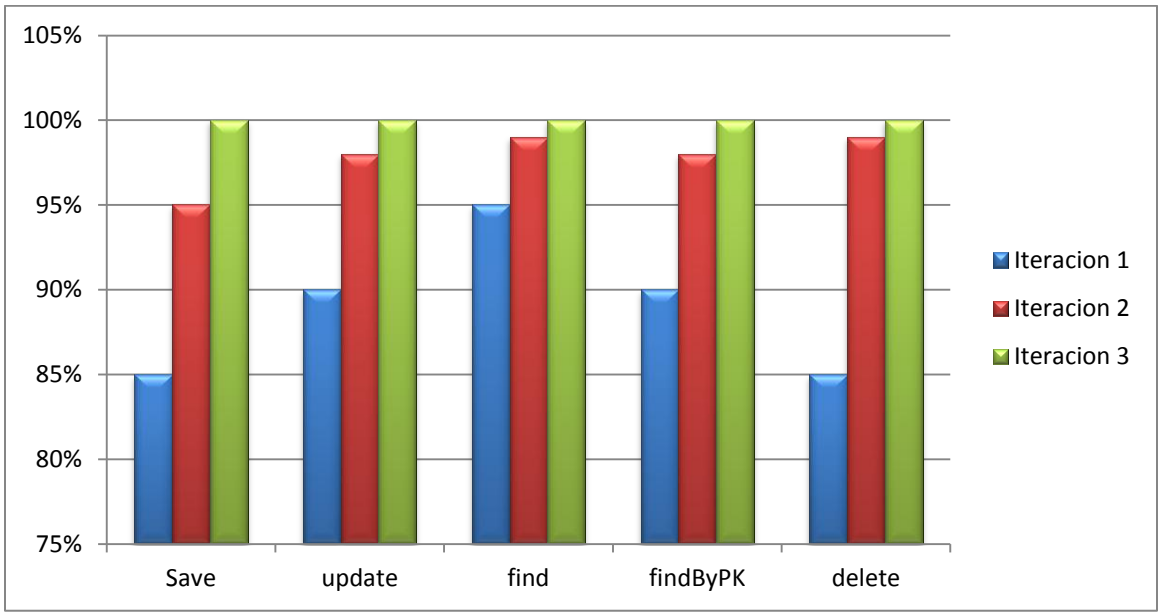

Figura. 3.2 Gráfica que muestra el porciento de los requisitos funcionando asociados a cada proceso

Ejemplo de algunas pruebas realizadas a la entidad Categorias

Prueba para revisar si inserta correctamente: Para el proceso de inserción se utilizó un formulario creado en java para de esta forma insertar los datos correspondientes en las entidades. No se inserta el campo id debido a que es un campo autogenerado por la base de datos.

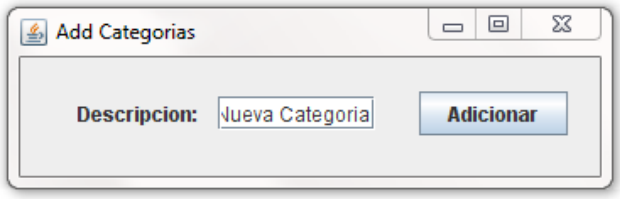

Figura. 3.3 Formulario que inserta una categoría

Código utilizado para insertar una categoría

```
private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {
    GenericDao dao=new DaoSeguridadSocial();
    Categorias categoria=new Categorias(jTextField1.getText());
    dao.save(categoria);
}
```

Figura. 3.4 Fragmento de código usado para insertar una categoría

Evidencia en la base de datos de los datos insertados

	id_categoria [PK] serial	descripcion character varying(20)
1	1	Giron
2	2	Nueva Categoria
*		

Figura. 3.5 Tabla que muestra el dato insertado en tabla

Proceso para listar: Para listar los datos existentes en la base de datos se usó un formulario hecho en java.

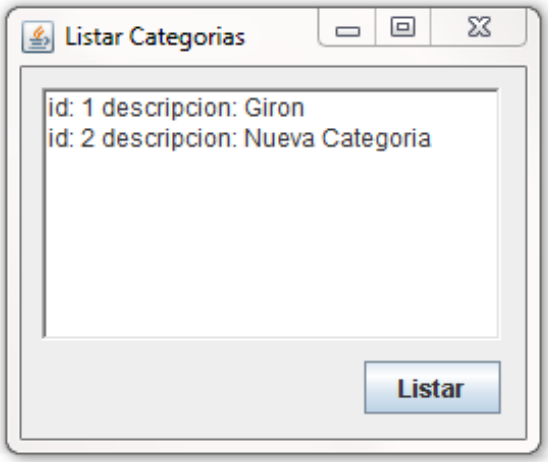

Figura. 3.6 Formulario que se utilizó para listar

Código utilizado para listar las categorías

```
private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {
  GenericDao dao=new GenericDao();
  List<Categorias> categorias=dao.find(Categorias.class);
    for (int i = 0; i < categorias.size(); i++) {
        list1.addItem("id: "+categorias.get(i).getIdCategoria()+" descripcion: "+categorias.get(i).getDescripcion());
    }
}
```

Figura. 3.7 Código que se utilizó para listar

Proceso de borrado: Para poder realizar el borrado de una tupla en la base de datos se realizó primero la función listar para luego seleccionar lo que se desea.

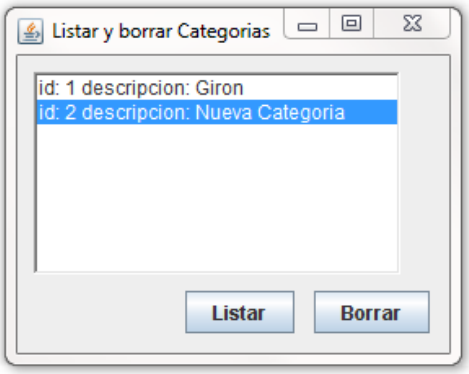

Figura. 3.8 Formulario que se utilizó para borrar

Código utilizado para borrar una categoría

```
private void jButton2MouseClicked(java.awt.event.MouseEvent evt) {
  GenericDao dao=new GenericDao();
  List<Categorias> categorias=dao.find(Categorias.class);
  dao.delete(categorias.get(list1.getSelectedIndex()));
}
```

Figura. 3.9 Código que se utilizó para borrar

Evidencia en la base de datos de los datos borrados

	id_categoria [PK] serial	descripcion character var
1	1	Giron
*		

Figura. 3.10 Tabla categorías sin la tupla eliminada

Prueba para actualizar una categoría: Para esta prueba usamos el mismo formulario listar solo que esta vez le añadimos un botón actualizar y un cuadro de texto para que pueda entrar el nuevo nombre. Como resultado de esta prueba se espera que cambien la descripción de la categoría seleccionada.

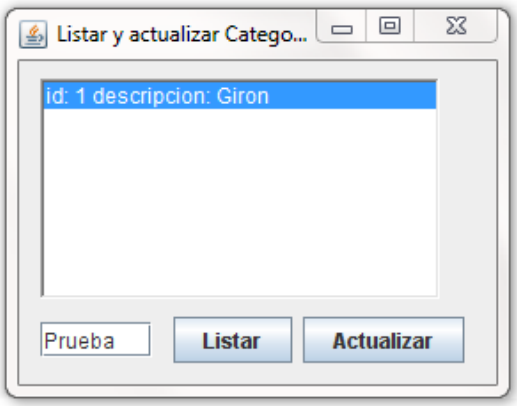

Figura. 3.11 Formulario para actualizar categorías

Código utilizado para actualizar categoría

```
private void jButton2MouseClicked(java.awt.event.MouseEvent evt) {
  GenericDao dao=new GenericDao();
  List<Categorias> categorias=dao.find(Categorias.class);
  categorias.get(list1.getSelectedIndex()).setDescripcion(jTextField1.getText())
  dao.update(categorias.get(list1.getSelectedIndex()));
}
```

Figura. 3.12 Código utilizado para actualizar

Evidencia en la base de datos de los datos actualizados

	id_categoria [PK] serial	descripcion character var
1	1	Prueba
*		

Figura. 3.13 Tabla categorías con el dato actualizado

3.3- Revisión Realizada por el departamento de calidad.

Para garantizar la calidad de la solución obtenida se realiza una inspección por parte de los expertos del departamento de calidad. La solución superó todas las pruebas hechas por el departamento. Razón por la cual se le otorga un aval certificando la eficiencia y calidad requerida para su integración con el SINAP. De esta forma también se avala la capacidad de la solución para resolver los problemas de integridad, disponibilidad y centralidad existentes en las direcciones de Colaboración y ACFM.

3.4- Solución y Funcionalidades Obtenidas.

Se obtuvo una base de datos que permite almacenar la información de las direcciones de Colaboración y ACFM de forma íntegra y centralizada. También se obtiene una capa de acceso a datos que tiene funciones principales como insertar, eliminar, actualizar, listar y buscar toda la información almacenada en la base de datos.

Conclusiones del capítulo

En el capítulo anterior se tomaron en cuenta rasgos importantes para obtener una base de datos y una capa de acceso a datos consistentes. Para validar teóricamente el diseño de la BD se midió la integridad de los datos encontrando varios errores que fueron debidamente corregidos. Con el propósito de evitar la redundancia de datos, los campos atómicos y la transitividad se revisó detalladamente si el diseño estaba normalizado correctamente.

Se analizó la seguridad de los datos describiendo los factores imprescindibles para lograr un almacenamiento confiable. Para comprobar un correcto funcionamiento del proceso se usaron pruebas de volumen de carga y pruebas del sistema. Se llegó a la conclusión que al menos los elementos probados no serán objeto de error en el futuro.

Conclusiones Generales

En el presente trabajo se realizó una investigación referente al proceso de gestión de información en las direcciones de Colaboración y ACFM. El estudio trajo resultados negativos debido al bajo nivel de integridad, disponibilidad y centralidad de la información existente en estas direcciones.

Al hacer un estudio sobre los principales conceptos y herramientas vinculadas al marco teórico de la solución propuesta, se escogió la metodología y las herramientas necesarias para el desarrollo de bases de datos y capas de acceso a datos. Como resultado de la investigación que se realizó sobre soluciones de software para la gestión de información gubernamental surge la necesidad de desarrollar una base de datos y una capa de acceso, para poder almacenar y recuperar la información en los módulos de Colaboración y ACFM.

Se caracterizó la información gestionada en cada uno de los procesos de las diferentes direcciones. De esta forma se establecieron los fundamentos necesarios para el desarrollo de la base de datos y la capa de acceso. Con el desarrollo de la base de datos y la capa de acceso a datos se logró elevar los niveles integridad, disponibilidad y centralidad de la información en el proceso de almacenamiento y recuperación de la información.

Para validar el buen funcionamiento de la base de datos y evaluar los niveles de integridad, disponibilidad y centralidad se realizaron pruebas teóricas y funcionales. Quedando demostrado que la solución propuesta es la indicada para resolver los problemas de almacenamiento y recuperación de la información existente en las direcciones de Colaboración y ACFM.

Recomendaciones

Se recomienda desplegar el sistema en la administración provincial para obtener la opinión del usuario después de un tiempo de uso.

Se recomienda desplegar el sistema en otras administraciones provinciales para brindarle un mayor aporte al país.

En próximas versiones implementar el mapeo de relaciones cíclicas en el framework JwebSocket. Para de esta forma dar soporte al uso de un ORM.

Implementar la clase Dao genérico que sea puramente de hibernate para de esta forma eliminar la dependencia de Spring.

Hacer extensivo el uso del visual Paradigm para hacer el diagrama entidad relacional y poder desechar la herramienta Power Architec

Bibliografía

Artides Visbal, Sara M. 2009. La gestión documental, de información y el conocimiento en la empresa. *La gestión documental, de información y el conocimiento en la empresa.* [En línea] 2009. [Citado el: 12 de 05 de 2012.] http://bvs.sld.cu/revistas/aci/vol19_5_09/aci02509.html.

C.J.Date. 2003. *Introducción a los Sistemas de bases de datos.* Ciudad de la Habana : Felix Varela, 2003. 2.

Capote, Olga Pons. 2005. *Introducción a las bases de datos.* s.l. : Thomson Editores Spain, 2005. 84-9732-396-3.

Colectivo. 2012. Definicion. *Definicion.* [En línea] 25 de 5 de 2012. http://definicion.de/centralizacion/.

Desarrolladores. 2012. Sitio Oficial. *Sitio Oficial.* [En línea] 29 de 1 de 2012. http://www.java.com/es/download/faq/whatis_java.xml.

Ferrell O. C. y Hirt Geoffrey, McGraw-Hill Interamericana. 2004. *Introducción a los Negocios en un Mundo Cambiante.* 2004.

Gavin King, Christian Bauer. 2004. *Hibernate in Action. A guide to the concepts and of object/relational mapping.* s.l. : Manning Plublications Co., 2004.

Gavin King, Christian Bauer, Max Rydahl Andersen, Emmanuel Bernard, y Steve Ebersole. 2009. *Documentación de referencia de Hibernate.* 2009.

Heidi, Toffler Alvin y Toffler. 2006. *La Revolución de la Riqueza.* 2006.

John C. Worsley, Joshua D.Drake. 2002. *Practical Postgrest SQL.* s.l. : Ellite Volckhausen, 2002. 1-56592-846-6.

José H. Canós, Patricio Letelier yMª Carmen Penadés. 2010. *Métodologías Ágiles en el Desarrollo de Software.* Valencia : Universidad Politécnica de Valencia, 2010.

Korth, Henry F, Silberschatz, Abraham. 1991. *Database System Concepts.* New York : McGraw-Hill, 1991. 0071008047.

Len Bass, Paul Clements, Rick Kazman. 2003. *Software Architecture in Practice(2nd edition).* s.l. : Addison-Wesley, 2003.

López, José E. Gallardo Ruiz y Carmen M. García. 2010. *Diseño modular.* Málaga : Facultad de Ciencias Matemáticas, 2010.

Marqués, Mercedes. 2011. *Bases de datos.* s.l. : Colección Sapientia, 2011. 978-84-693-0146-3.

Mato, Rosa Maria. 2006. *Sistemas de Bases de Datos.* Ciudad de la Habana, Cuna : Felix Varela, 2006. 1.

—. 2006. *Sistemas de Bases de Datos.* Ciudad de la Habana, Cuna : Felix Varela, 2006. 1.

Miguel Castaño, Mario G. Piattini Velthuis. 2004. *Fundamentos y modelos de bases de datos.* s.l. : RA-MA, 2004. 8478973613, 9788478973613.

NetBeans. 2011. *NetBeans.* [En línea] Oracle Corporation, 2011. http://netbeans.org/features/index.html.

Óscar Pastor, Pedro Blesa Pons. 2007. *Gestión de Bases de Datos.* s.l. : Servivio de publicaciones, 2007. 84-7721-861-7.

Paradimg, Visual. 2012. Visual Paradimg. *Visual Paradimg.* [En línea] 22 de 4 de 2012. http://www.visual-paradigm.com.

Peter Rob, Carlos Coronel. 2004. *Sistemas de Bases de Datos.* 2004. 970-686-286-2.

pgAdmin. 2011. pgAdmin. [En línea] 2011. http://www.pgadmin.org/.

Polín, Ignacio Bruna López. 2010. *Confluencia con la LOPD.* s.l. : BELT IBÉRICA S.A. , 2010.

Real Academia Española. Diccionario de la Lengua Española Vigésima Segunda Edición. *REAL ACADEMIA ESPAÑOLA.* [En línea] [Citado el: 31 de Enero de 2012.] http://buscon.rae.es/draeI/SrvltConsulta?TIPO_BUS=3&LEMA=g%C3%A9nesis.

Reynoso, Carlos Billy. 2004. *Introducción a la Arquitectura de Software.* BUENOS AIRES : UNIVERSIDAD DE BUENOS AIRES, 2004.

Rumbaugh, G. Booch y J. 2000. *El Proceso Unificado de Desarrollo de Software.* Madrid : Pearson Educación, S. A, 2000.

Sitio Oficial. Visual Paradigm. [En línea] [Citado el: 11 de 12 de 2011.] http://www.visual-paradigm.com.

Vieites, Alvaro Gomez. 2007. *Enciclopedia de la Seguridad Informática.* 2007.