

Universidad de las Ciencias Informáticas

Facultad 4



Título:

Configuración del Generador de Recuperaciones
Dinámicas para aplicaciones Web.

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

Autor(es):

Yanet Pérez Valcárcel
Belkis Y. Torres González

Tutor(es):

Msc. Raquel Vidal Soto

Consultante:

Ing. Raydel Muñoz Vidal

Ciudad de la Habana, Cuba
Julio 2007

DECLARACION DE AUTORÍA

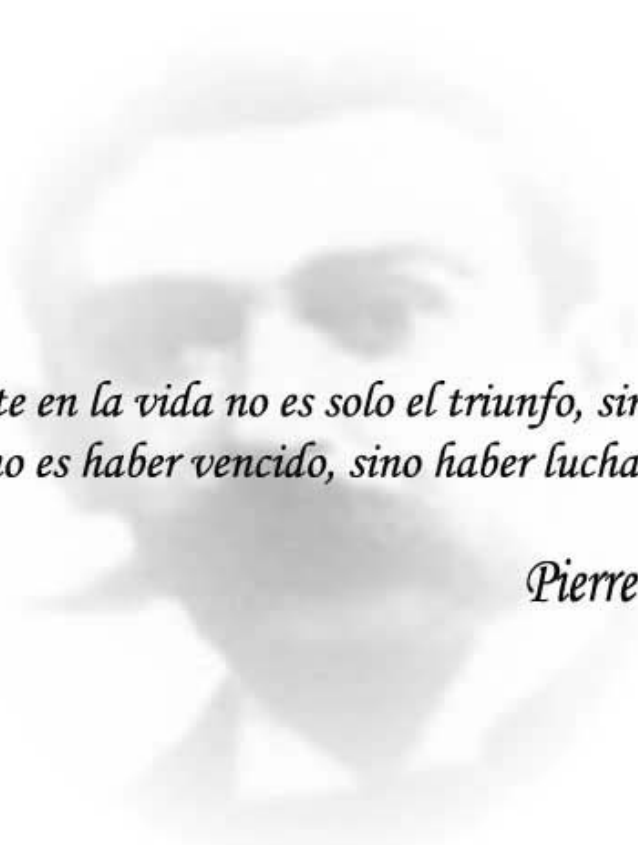
Por este medio declaramos que somos las únicas autoras de este trabajo y autorizamos al Ministerio de las Fuerzas Armadas Revolucionarias (MINFAR) y a la Universidad de Ciencias Informáticas (UCI) para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmamos la presente a los 27 días del mes de Junio de 2007.

Firma del Autor

Firma del Autor

Firma del Tutor



*"Lo importante en la vida no es solo el triunfo, sino la lucha,
lo esencial no es haber vencido, sino haber luchado bien..."*

Pierre de Coubertin

AGRADECIMIENTOS

Resulta justo, encabezar este trabajo de diploma con los merecidos agradecimientos que le pudiéramos tributar a todas aquellas personas que de una forma u otra, con su apoyo moral, con su cariño, con su presencia y hasta con una simple sonrisa nos han dado todo aquello que hemos necesitado en el momento preciso y cuando mas falta nos hacia. Mencionar nombres seria impropio del objetivo que perseguimos con estas palabras, porque podríamos incurrir en el fatal error de omitir entre líneas el nombre de alguien que humildemente haya contribuido con este derroche de sacrificio y dedicación que supone la realización de un trabajo de diploma, por eso, simplemente optamos por la sencilla manera de agrupar en estas frases, mas que los nombres, el recuerdo de todos aquellos amigos, familiares y compañeros, que se convirtieron en el eslabón fundamental de nuestras vidas en este periodo de tiempo y a quien agradecemos, profundamente, el hecho de que hoy podamos sentirnos como ingenieros de la Patria.

DEDICATORIA

A mi padre Rolando Torres González por haberme sabido guiar y trasmitirme su experiencia a lo largo de todos estos años.

A mi madre Laudelina González Naranjo por brindarme su confianza, cariño de madre, ser amiga, compañera, y consejera indisoluble de mi vida.

A toda mi familia por depositar en mí toda la confianza que un ser humano puede llevar consigo.

A mis amigos por haber sabido compartir conmigo toda su vida universitaria, la cual ha sido la etapa mas linda de mi vida.

Y a todos los que nos ayudaron de una forma u otra en la realización de este trabajo.

Yadira.

A mis padres: los verdaderos artífices de esta obra, por ser mi fuerza, mi inspiración, mi orgullo, por ser todo lo que necesito y necesitaré siempre.

A mi familia: por ser el sustento fundamental de mi vida, por construirme el rinconcito más feliz y cálido del mundo.

A mis amigos: por ayudarme y acompañarme en los momentos más difíciles, por hacerme encontrar la dicha cada momento que paso con ellos.

A todo aquel que me ha estrechado su mano, me ha dado aliento, me ha hecho sentir que la vida me sonrío.

Yanet.

RESUMEN

En la actualidad el desarrollo tecnológico ha posibilitado que se informaticen diferentes procesos de gestión de información que son necesarios para el funcionamiento óptimo de cualquier entidad. Nuestro país es partícipe de esto, pues una de las metas que se ha propuesto es la informatización de la sociedad, por lo que las diferentes empresas y organismos se encuentran desarrollando sistemas para el manejo de información.

Una de estas empresas es el Ministerio de las Fuerzas Armadas Revolucionarias (MINFAR), que debido al elevado volumen de información que presenta, pretende desarrollar herramientas informáticas dirigidas al proceso de configuración de aplicaciones que recuperan información de forma dinámica que se encuentra almacenada en los Sistemas de Gestión de Base de Datos de dicha entidad. Para así garantizar la integridad, seguridad, protección y recuperación rápida y precisa de los datos que se soliciten.

En este trabajo se identifican algunas de las principales deficiencias que están presentes en el MINFAR para la recuperación de la información almacenada. Debido a esto se propone una solución a estos disímiles problemas, la cual se describe a lo largo de este documento, el cual cuenta con cuatro capítulos los cuales complementan con su contenido la solución propuesta.

Palabras Claves:

Gestión de información, proceso de configuración, recuperación de la información.

INDICE

AGRADECIMIENTOS	II
DEDICATORIA	III
RESUMEN	IV
INTRODUCCION	1
CAPITULO I: FUNDAMENTACION TEORICA	5
1.1 Introducción	5
1.2 Sistemas informatizados existentes vinculados al campo de acción	5
1.2.1 Un marco de configuración común	6
1.3 Tendencias y tecnologías actuales de apoyo en la solución del problema	6
1.4 Metodología a utilizar	7
1.5 Lenguaje Unificado de Modelado (UML).....	7
1.6 El Proceso Unificado de Modelado (RUP)	8
1.7 Tecnología Cliente/Servidor.....	9
1.8 Lenguajes de Programación para la Web.....	9
1.8.1 Lenguajes del lado del cliente:.....	10
1.8.2 Lenguajes del lado del Servidor:.....	11
1.9 Tecnología AJAX	12
1.10 Sistemas de Gestión de Bases de Datos (SGBD)	12
1.10.1 PostgreSQL Sistema de Gestión de Bases de Datos	13
1.11 Herramientas CASE.....	15
1.11.1 Selección de la herramienta CASE a utilizar:	16
1.12 Otras Herramientas necesarias	16
1.13 Conclusiones	17

CAPITULO II: CARACTERISTICAS DEL SISTEMA	18
2.1 Introducción	18
2.2 Problema y situación problemática	18
2.3 Objetivos estratégicos de la organización.....	19
2.4 Descripción de los procesos del negocio propuestos	19
2.5 Propuesta del sistema.	20
2.6 Modelo del Dominio	20
2.6.1 Conceptos principales.....	21
2.6.2 Principales Eventos.....	21
2.6.3 Diagrama de clases del dominio	22
2.7 Requerimientos Funcionales	22
2.8 Requerimientos no funcionales.....	24
2.9 Modelo del Sistema.	26
2.9.1 Definición de los casos de uso.....	26
2.9.2 Definición de los actores.	26
2.9.3 Descripción de los Casos de Uso del Sistema.	27
2.9.4 Descripciones no expandidas:	27
2.9.5 Diagrama de Casos de Uso del Sistema	30
2.9.6 Casos de uso expandidos.....	31
2.9.7 Descripción del Caso de Uso Crear configuración inicial del sistema.	31
2.9.9 Descripción del Caso de Uso Gestionar esquema de la Base de Datos.	39
2.9.10 Descripción del Caso de Uso Gestionar tabla de un esquema.....	44
2.9.11 Descripción del Caso de Uso Gestionar relaciones.....	49
2.9.12 Descripción del Caso de Uso gestionar dependencias.....	55
2.9.13 Descripción del Caso de Uso Gestionar accesos.	61
2.9.14 Descripción del Caso de Uso Revisar consistencia.....	66

2.10 Conclusiones	70
CAPITULO III: ANALISIS Y DISEÑO DEL SISTEMA	71
3.1 Introducción	71
3.2 Análisis.....	71
3.2.1 Diagramas de clases del análisis.....	71
3.3 Arquitectura del Sistema.....	75
3.4 Diseño.....	77
3.4.1 Diagramas de Secuencia.....	77
3.4.2 Diagrama de clases del diseño.....	78
3.4.3 Mecanismo de Diseño de Seguridad.....	85
3.4.4 Mecanismo de Diseño de Acceso a Datos.....	86
3.5 Diseño de la Base de Datos.....	88
3.5.1 Diagrama de Clases Persistentes.....	88
3.5.2 Modelo de Datos.....	89
3.5.3 Diagrama Entidad Relación de la BD.....	90
3.5.4 Descripción de las tablas.....	90
3.6 Conclusiones	96
CAPITULO IV: IMPLEMENTACION Y PRUEBA	97
4.1 Introducción	97
4.2 Diagramas de Componentes	97
4.3 Diagramas de Despliegue.....	103
4.4 Conclusiones	104
CONCLUSIONES	105

RECOMENDACIONES106

BIBLIOGRAFIA107

GLOSARIO DE TÉRMINOS109

INTRODUCCION

En la actualidad, la información existente en el mundo se va haciendo cada vez más amplia, y esta es necesario que esté a la disposición de los usuarios, tanto física, como digital y en los más disímiles formatos. Por lo que se hace necesario lograr aplicaciones de software donde su objetivo fundamental sea realizar recuperaciones de información, ya que permite a las entidades acceder de forma rápida y objetiva a los datos almacenados. Y esto se puede lograr gracias al desarrollo de las nuevas Tecnologías de la Información y las Telecomunicaciones con que se cuenta. Pero para poder lograr una eficiente aplicación de recuperación de información capaz de acoplarse a una variedad de sistemas, se hace imprescindible una buena configuración de ésta, ya que es la base de cualquier acción que se realice con el objetivo de obtener una determinada información, permitiendo que los usuarios accedan de forma rápida, dinámica y precisa a los datos almacenados.

Hoy en Cuba se esta llevando a cabo un proceso de informatización por lo que diferentes empresas o entidades se encuentran desarrollando sistemas de gestión de la información. Un ejemplo de esto es el Ministerio de las Fuerzas Armadas Revolucionarias (MINFAR), que es la entidad encargada de la defensa del país y que cuenta con algunos sistemas desarrollados a raíz de este proceso de informatización. En la construcción de estos sistemas durante la etapa de análisis y diseño de las aplicaciones, por diferentes razones no se pronosticaron todos los requerimientos informativos que se debían tener en cuenta. Esto hizo que se generaran modelos estáticos de recuperación sin posibilidades de cambios, obligando a consultar los mismos, que quizás la información que te muestren sea más o menos abarcadora que la que se requiere. A medida que se utilizan las aplicaciones las necesidades que surgen son mayores que las que se le brindan al usuario en un inicio. Por esta razón se hace imprescindible la realización de mantenimientos a los sistemas, interrumpiendo así el funcionamiento óptimo de los mismos, trayendo como consecuencia un menor rendimiento en la recuperación de la información y un retraso en la obtención de esta, debido a la reestructuración del sistema de forma tal que se pueda recuperar la información deseada.

Después de un análisis de las situaciones antes expuestas; se puede decir que el **problema a resolver** es: la no existencia de una herramienta flexible que permita configurar una aplicación de recuperación dinámica de información.

Conjuntamente con el desarrollo de las redes de computadoras se han implementado diversas aplicaciones, cuyo propósito es la gestión de la información para presentarla con algún formato previamente determinado. La recuperación de información es el conjunto de tareas mediante las cuales el usuario localiza y accede a los recursos de información que son pertinentes a la solución del problema planteado.

Con el presente trabajo se propone construir una herramienta de configuración de una aplicación de recuperaciones dinámicas de información que permita un mayor aprovechamiento de la jornada laboral, logrando que el sistema se convierta en una ayuda indispensable y útil para todos.

Por tanto el **objeto de estudio** de este trabajo es el proceso de configuración de la aplicación de recuperación de la información existente en las Bases de Datos del MINFAR.

De ello se deriva que el **campo de acción** que abarca este trabajo, son los sistemas de gestión de recuperación de información existente en el MINFAR.

Como **hipótesis** se parte de la idea que, si se cuenta con una herramienta Web, basada en la configuración de la aplicación de recuperación de información; debe mejorar en dinamismo y rapidez el acoplamiento de éste a las aplicaciones desarrolladas en el MINFAR y a la vez servir como soporte para el buen funcionamiento del mismo.

El **objetivo general** de este trabajo será: desarrollar una herramienta que permita la configuración de la aplicación de recuperación de la información existente en los Sistemas de Gestión de Base de Datos del MINFAR de manera dinámica.

De acuerdo con esta propuesta se derivan los siguientes **objetivos específicos**:

- Realizar un estudio sobre formas y mecanismos de configuración de aplicaciones que recuperen información. Y proponer uno para desarrollarlo en diversas aplicaciones.
- Analizar y Diseñar una aplicación Web que pueda acoplarse fácilmente a un sistema de gestión de base de datos que permita configurar la aplicación de recuperaciones dinámicas de información.

- Proponer un proceso que garantice la integridad de la información contenida en las Bases de Datos y a la vez permita la mayor colaboración posible en las tareas de recuperación por parte de los usuarios que utilicen la aplicación.

Para cumplir con estos objetivos y resolver la situación problemática planteada, se proponen las siguientes **tareas**:

1. Estudio, descripción y formas de configuración de los sistemas para la recuperación de información.
2. Análisis del estado del arte de las tecnologías que se utilizan para llevar a cabo herramientas como la que se pretende desarrollar.
3. Selección de la metodología de análisis y diseño de sistemas informáticos que faciliten la creación y garanticen la calidad de la herramienta.
4. Selección de las herramientas para llevar a cabo la realización del proyecto, así como la elección de la plataforma en la que se desarrollará la aplicación.

La realización y puesta en práctica del Configurador de la aplicación de recuperaciones dinámicas para aplicaciones Web debe resolver el problema existente en el ámbito para el cual está destinado.

Las **ventajas** más inmediatas que posee son:

- Permite acoplar la aplicación de recuperaciones dinámicas de información a las aplicaciones Web desarrolladas en el MINFAR.
- Permite en tiempo real la reestructuración de la información almacenada de forma tal que se pueda recuperar según la necesidad de los usuarios, sin afectar su integridad.

La culminación e implantación de la herramienta Web proporcionará los siguientes **aportes prácticos**: Se contará con una herramienta flexible de configuración de la aplicación de recuperación dinámica de información. Además será fácil adaptarlo a cualquier plataforma, debido a posibles migraciones de sistemas operativos que se realicen.

Este documento consta de cuatro capítulos en los cuales se describen paso a paso todo el proceso o etapas por la que transitó nuestro trabajo. Además de presentar Introducción, Conclusiones, Recomendaciones, Bibliografía, Anexos y Glosario de términos.

En el Capítulo 1: **Fundamentación teórica**, se aborda el tema relacionado con los sistemas informatizados existentes vinculados al campo de acción; así como las técnicas, tecnologías, herramientas empleadas y la metodología para el desarrollo del software empleado.

En el Capítulo 2: **Características del sistema**, en el desarrollo de este capítulo se describe el negocio a través de un modelo de Dominio, y se hace el análisis del sistema a desarrollar. Se definen las funcionalidades del sistema y se describen detalladamente, utilizando herramientas de modelación.

En el Capítulo 3. **Análisis y diseño del sistema**, enfoca la construcción de la solución mediante diagramas de clases, de datos, y se plantean los principios para el diseño y la implementación. Aquí se construyen las funcionalidades que se definieron en el capítulo anterior.

En el Capítulo 4. **Implementación y prueba**, en el desarrollo de este capítulo se describe como se encuentra estructurado el sistema representado a través del diagrama de despliegue y el diagrama de componentes. También se documenta la realización de un conjunto de pruebas en vista a detección de diferentes errores que dificulten la eficiencia del sistema.

Capítulo

I

CAPITULO I: FUNDAMENTACION TEORICA**1.1 Introducción**

En el presente capítulo se brinda una visión general de aspectos relacionados con sistemas informatizados de recuperación de información. También se da una breve panorámica de las herramientas, técnicas y lenguajes de programación utilizada para el desarrollo de nuestra aplicación, así como la descripción de las principales definiciones asociadas al dominio del problema y que son necesarias para entender el negocio y la propuesta de solución.

Además se definen los conceptos más importantes a manipular en el resto del documento y se hace una valoración de softwares similares existentes en el mundo incluyendo una comparación entre estos. También se da una breve descripción de la metodología de análisis y diseño escogida y de las herramientas de desarrollo usadas para la confección del sistema.

1.2 Sistemas informatizados existentes vinculados al campo de acción

En el mundo actual han avanzado considerablemente las TICs y esto ha provocado que se desarrollen sistemas informatizados día a día mucho más eficientes. Por lo que actualmente existe en las instituciones de nuestro país, como por ejemplo en la FAR, una gran demanda de software para resolver disímiles problemas teóricos y prácticos. Nuestra universidad (UCI) tiene un convenio con esta institución, específicamente en la Facultad 4, donde un grupo de estudiantes tratan de resolver algunas de las deficiencias existentes, como es el caso de la configuración de recuperaciones dinámicas de información. A nivel mundial existen diversos gestores para la recuperación de la misma, uno de los más comunes que se utiliza es el Microsoft Query que conforma el paquete de herramientas de la compañía Microsoft. También se puede encontrar herramientas más potentes para realizar recuperaciones como el Crystal Report, siendo éste uno de los más populares, el Agata Report y el Jasper Report.

Pero en la investigación realizada hasta el momento no se encontró alguna herramienta de configuración de aplicaciones que recuperen información, pero si sobre la gran utilidad que presentan los configuradores en general, pues gracias a estos el cliente dispone de una guía para componer y seleccionar el producto adaptado a cada necesidad. Estas herramientas permiten el ahorro de tiempo y esfuerzo al prescriptor del producto. Por ejemplo: Urbandeco.

1.2.1 Un marco de configuración común

El configurador conocido como Measurement & Automation Explorer, presenta una vista unificada al sistema de hardware de medición soportado por el software de control de servicios. Con Measurement & Automation Explorer los usuarios pueden definir el nombre de canales para organizar señales o especificar funciones de escalamiento para convertir señales digitalizadas a cantidades medidas.

El beneficio clave del configurador es la integración con los ADEs, como Microsoft Visual Basic y Visual C++ y NI LabVIEW y Measurement Studio. Esta integración le da a los desarrolladores la habilidad de fácilmente integrar múltiples mediciones en una sola aplicación sin programación. Sin estas herramientas de configuración, los desarrolladores deben pasar mucho tiempo configurando las funciones de medición de manera programática. [10]

De aquí que se decidiera implementar un sistema de configuración para las recuperaciones dinámicas de información, y hacer un estudio de los sistemas existentes, para determinar las debilidades y las fortalezas que pueden presentar.

1.3 Tendencias y tecnologías actuales de apoyo en la solución del problema

Son diversas las tendencias, técnicas, tecnologías, metodologías y softwares usados en la actualidad para realizar aplicaciones Web; las FAR, institución para la cual se está llevando a cabo esta herramienta Web, tiene como política de desarrollo de sus aplicaciones la utilización de software libre. A continuación se muestran algunos de los conceptos y características generales de las tecnologías, metodologías y softwares utilizados en la solución de nuestro problema.

1.4 Metodología a utilizar

Actualmente los principales procesos de las empresas son realizados por sistemas informáticos, debido a esto las mismas tienen como objetivo fundamental priorizar la calidad en el desarrollo y mantenimiento del software.

Una metodología para el desarrollo de un proceso de software nos brinda mayor organización a la hora de desarrollar un sistema, y contribuye a que se obtenga un producto eficiente, logrando así la satisfacción de los clientes. Es por esto que se hace necesaria la buena elección de la metodología que va a guiar el proceso de desarrollo del sistema.

1.5 Lenguaje Unificado de Modelado (UML)

UML (Lenguaje Unificado de Modelado) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables. [1]

UML ofrece nueve diagramas en los cuales modelar sistemas.

- Diagramas de Casos de Uso para modelar los procesos del negocio.
- Diagramas de Secuencia para modelar el paso de mensajes entre objetos.
- Diagramas de Colaboración para modelar interacciones entre objetos.
- Diagramas de Estado para modelar el comportamiento de los objetos en el sistema.
- Diagramas de Actividad para modelar el comportamiento de los Casos de Uso, objetos u operaciones.
- Diagramas de Clases para modelar la estructura estática de las clases en el sistema.
- Diagramas de Objetos para modelar la estructura estática de los objetos en el sistema.
- Diagramas de Componentes para modelar componentes.
- Diagramas de Implementación para modelar la distribución del sistema.

1.6 El Proceso Unificado de Modelado (RUP)

El Proceso Racional Unificado o RUP (Rational Unified Process), es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. [2]

Presenta una forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo). Pretende implementar las mejores prácticas en Ingeniería de Software. Posee una administración de requisitos. Hace uso de arquitectura basada en componentes. Tiene en cuenta el control de cambios y la verificación de la calidad del software.

Las principales características de RUP son:

Dirigido por casos de uso:

En el Proceso Unificado los casos de uso se utilizan para capturar los requisitos funcionales y para definir los contenidos de las iteraciones. La idea es que cada iteración tome un conjunto de casos de uso o escenarios y desarrolle todo el camino a través de las distintas disciplinas: diseño, implementación, prueba, etc. [3]

Centrado en la Arquitectura:

La arquitectura de un sistema software se representa a través de las diferentes vistas del sistema en construcción como pueden ser: Vista de Casos de Uso, Vista Lógica, Vista de Procesos.

Vista de Implementación, Vista de Despliegue.

La arquitectura del software abarca:

- La organización del sistema software.
- Los elementos estructurales que compondrán el sistema y sus interfaces, con sus comportamientos.
- El estilo arquitectónico que guía esta organización.

Iterativo e Incremental:

El Proceso Unificado es un marco de desarrollo iterativo e incremental compuesto de cuatro fases denominadas Inicio, Elaboración, Construcción y Transición. Cada una de estas fases es a su vez dividida en una serie de iteraciones (la de inicio sólo consta de varias iteraciones en proyectos grandes). Estas

iteraciones ofrecen como resultado un incremento del producto desarrollado que añade o mejora las funcionalidades del sistema en desarrollo. [3]

1.7 Tecnología Cliente/Servidor

Es la tecnología que proporciona al usuario final el acceso transparente a las aplicaciones, datos, servicios de cómputo o cualquier otro recurso del grupo de trabajo y/o, a través de la organización, en múltiples plataformas. [4]

El término cliente/servidor es originalmente aplicado a la arquitectura de software que describe el procesamiento de datos entre dos o más programas: una aplicación y un servicio soportante.

El modelo Cliente/Servidor puede actuar como una sola entidad y también como entidades separadas, realizando actividades o tareas independientes. Sus funciones pueden estar en plataformas separadas, o en la misma y ser dinámicas. Cada plataforma puede ser escalable independientemente.

Se puede decir que el modelo Cliente/Servidor puede incluir múltiples plataformas, bases de datos, redes y sistemas operativos. Estos pueden ser de distintos proveedores, en arquitecturas propietarias y no propietarias y funcionando todos al mismo tiempo.

1.8 Lenguajes de Programación para la Web

Uno de los ejes fundamentales que diferencian a Internet de otros medios de comunicación es la interacción y personalización de la información con el usuario. Esto se logra por medio de algunos de los diferentes lenguajes de programación para Web que existen hoy en día. Dichos lenguajes se clasifican en dos partes fundamentales que reconocen la propia arquitectura Cliente/Servidor de esta plataforma de desarrollo: los lenguajes del lado del Servidor que se caracterizan por desarrollar la lógica de negocio dentro del Servidor, además de ser los encargados del acceso a Bases de Datos, tratamiento de la información, etc, y los lenguajes del lado del Cliente que son los encargados de aportar dinamismo a la aplicación en los navegadores.

1.8.1 Lenguajes del lado del cliente:

JavaScript

Es el lenguaje de programación que nos permite interactuar con el navegador de manera dinámica y eficaz, no requiere compilación, es utilizado principalmente en páginas Web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado el permite de forma eficiente validar formularios, detectar navegadores y mejorar el diseño, además de ser un lenguaje de fácil aprendizaje.

HTML

El HTML, es el lenguaje de marcas de texto utilizado normalmente en la WWW (World Wide Web). HTML no es propiamente un lenguaje de programación como C++, Visual Basic, etc., sino un sistema de etiquetas. HTML no presenta ningún compilador, por lo tanto algún error de sintaxis que se presente éste no lo detectará y se visualizara en la forma como éste lo entienda. [8]

El entorno para trabajar HTML es simplemente un procesador de texto, como el que ofrecen los sistemas operativos Windows (Bloc de notas), UNIX (el editor vi o ed) o el que ofrece MS Office (Word). El conjunto de etiquetas que se creen, se deben guardar con la extensión .htm o .html. Estos documentos pueden ser mostrados por los visores o "browsers" de paginas Web en Internet, como Netscape Navigator, Mosaic, Opera y Microsoft Internet Explorer. [8]

XML

Lenguaje universal de marcado para documentos estructurados y datos en la Web, más amplio, más rico y más dinámico que HTML. No solo es un lenguaje de marcado, sino también un metalenguaje que permite describir otros lenguajes de marcado.

Resumen de los lenguajes del lado del cliente:

Teniendo en cuenta los lenguajes de programación existentes que corren en el lado del cliente se determina que JavaScript es el ideal debido a que es compatible con la mayoría de los navegadores modernos, crea efectos especiales en las páginas y define interactividades con el usuario. Además de fácil aprendizaje. Al mismo tiempo HTML es un lenguaje de marcas diseñado para estructurar textos y

presentarlos en forma de hipertexto, que es el formato estándar de las páginas Web. Así como XML que es un lenguaje universal de marcado para documentos estructurados y datos en la Web, es más amplio, rico y dinámico que HTML. Por todas estas características expuestas anteriormente fue que se decidió el empleo de los mismos.

1.8.2 Lenguajes del lado del Servidor:

Personal Home Page (PHP)

PHP es un lenguaje de programación usado generalmente para la creación de contenido para sitios Web. Se trata de un lenguaje usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios Web. Debido al diseño de PHP, también es posible crear aplicaciones con una interfaz gráfica para el usuario. Permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, Postgres, Oracle, Microsoft SQL Server, lo cual permite la creación de Aplicaciones Web robustas. PHP también tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos tales como UNIX (y de ese tipo, como Linux), Windows y Mac OS X, y puede interactuar con los servidores de Web más populares. [5]



Figura 1.1 Esquema de representación del funcionamiento del PHP.

PHP es un lenguaje multiplataforma. Permite leer y manipular datos desde diversas fuentes, incluyendo datos que pueden ingresar los usuarios desde formularios HTML. Presenta la capacidad de expandir su potencial utilizando la enorme cantidad de módulos. Posee una amplia documentación. Es libre, por lo que se presenta como una alternativa de fácil acceso para todos. Permite las técnicas de

Programación Orientada a Objetos así como crear los formularios para la Web. No requiere definición de tipos de variables ni manejo detallado del bajo nivel.

Resumen del lenguaje del lado del servidor:

Se llega a la conclusión de que PHP es el óptimo para implementar la propuesta de sistema de este trabajo, debido a la velocidad de ejecución que presenta, es uno de los que tienen mayor disponibilidad de recursos en Internet, es el lenguaje más usado por los usuarios, y esto es debido también a su fácil manejo. Además de estar soportado en múltiples plataformas.

1.9 Tecnología AJAX

AJAX, acrónimo de Asynchronous JavaScript And XML (JavaScript y XML asíncronos), es una técnica de desarrollo Web para crear aplicaciones interactivas. Éstas se ejecutan en el cliente, es decir, en el navegador del usuario, y mantiene comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla. Esto significa aumentar la interactividad, velocidad y usabilidad en la misma. [9]

1.10 Sistemas de Gestión de Bases de Datos (SGBD)

Los Sistemas de gestión de base de datos son tipos de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, manipulación de datos y consulta. Tienen el propósito de manejar de manera clara, sencilla y ordenada un conjunto de información. [6]

Las principales funciones que debe cumplir un SGBD se relacionan con la creación y mantenimiento de la base de datos, el control de accesos, la manipulación de datos de acuerdo con las necesidades del usuario, el cumplimiento de las normas de tratamiento de datos, evitar redundancias e inconsistencias y mantener la integridad. Presentan facilidad de manejo de grandes volúmenes de información de manera rápida, además de la independencia del tratamiento de la misma.

Los Sistemas de Gestión de Bases de Datos presentan diversos inconvenientes entre los que están el elevado costo de actualización del hardware y software, un mal diseño y un mal adiestramiento a los usuarios puede originar problemas en un futuro. Además se generan campos vacíos en exceso.

Los SGBD más utilizados en el mundo son Oracle, MySQL, Microsoft SQL Server, PostgreSQL, entre otros.

La información es representada a través de tuplas, las cuales describen al fenómeno, proceso o ente de la realidad objetiva que se está analizando y se representan a través de tablas.

1.10.1 PostgreSQL Sistema de Gestión de Bases de Datos

PostgreSQL es un Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS) que ha sido desarrollado de varias formas desde 1977.

PostgreSQL está considerado como el sistema de bases de datos de código abierto más avanzado del mundo. Posee muchas características que tradicionalmente sólo se podían ver en productos comerciales de alto calibre.

Selección del Sistema de Gestión de Bases de Datos (SGBD)

Se seleccionó el **PostgreSQL** como herramienta para almacenar y gestionar los datos de nuestro sistema.

¿Qué características tiene PostgreSQL que lo hace nuestra elección?

PostgreSQL ofrece muchas ventajas para una compañía o negocio, respecto a otros sistemas de bases de datos, en cuanto a:

1- Instalación ilimitada

Es frecuente que las bases de datos comerciales sean instaladas en más servidores de lo que permite la licencia. Algunos proveedores comerciales consideran a esto la principal fuente de incumplimiento de licencia. Con PostgreSQL, nadie puede demandarlo por violar acuerdos de licencia, puesto que no hay costo asociado a la licencia del software.

Esto trae consigo ventajas adicionales como:

- ✓ Modelos de negocios más rentables con instalaciones a gran escala.
- ✓ No existe la posibilidad de ser auditado para verificar cumplimiento de licencia en ningún momento.
- ✓ Flexibilidad para hacer investigación y desarrollo sin necesidad de incurrir en costos adicionales de licenciamiento.

2- Ahorros considerables en costos de operación

Este software ha sido diseñado y creado para tener un mantenimiento y ajuste mucho menor que los productos de los proveedores comerciales, conservando todas las características, estabilidad y rendimiento.

Además de esto, los programas de entrenamiento son reconocidamente mucho más costo-efectivos, manejables y prácticos en el mundo real que aquellos de los principales proveedores comerciales.

3- Estabilidad y confiabilidad legendarias

En contraste a muchos sistemas de bases de datos comerciales, es extremadamente común que compañías reporten que PostgreSQL nunca ha presentado caídas en varios años de operación de alta actividad.

4- Extensible

El código fuente está disponible para todos sin costo. Si un equipo necesita extender o personalizar PostgreSQL de alguna manera, pueden hacerlo con un mínimo esfuerzo, sin costos adicionales. Esto es complementado por la comunidad de profesionales y entusiastas de PostgreSQL alrededor del mundo que también extienden PostgreSQL todos los días.

5- Multiplataforma

PostgreSQL está disponible en casi cualquier Unix (34 plataformas en la última versión estable), y una versión nativa de Windows está actualmente en estado beta de pruebas.

6- Diseñado para ambientes de alto volumen

PostgreSQL usa una estrategia de almacenamiento de filas llamada MVCC para conseguir una mejor respuesta en ambientes de grandes volúmenes. Los principales proveedores de sistemas de bases de datos comerciales usan también esta tecnología, por las mismas razones.

7- Herramientas gráficas de diseño y administración de bases de datos.

Existen varias herramientas gráficas de alta calidad para administrar las bases de datos (pgAdmin, pgAccess) y para hacer diseños de bases de datos (Tora, Data Architect).

Además PostgreSQL ofrece una serie de características técnicas, al igual que otros gestores, que permiten un mejor trabajo con las bases de datos; entre estas se puede encontrar:

- ✓ Replicación (soluciones comerciales y no comerciales) que permiten la duplicación de bases de datos maestras en múltiples sitios de réplica.

- ✓ Interfaces nativas para ODBC, JDBC, C, C++, PHP, Perl, TCL, ECPG, Python y Ruby.
 - ✓ Reglas
 - ✓ Vistas
 - ✓ Triggers
 - ✓ Unicode
 - ✓ Secuencias
 - ✓ Herencia
 - ✓ Outer Joins
 - ✓ Sub-selects
 - ✓ Una API abierta
 - ✓ Procedimientos almacenados
 - ✓ Soporte nativo SSL
 - ✓ Lenguajes procedurales
 - ✓ Índices parciales y funcionales
 - ✓ Soporte para consultas con UNION, UNION ALL y EXCEPT
 - ✓ Extensiones para SHA1, MD5, XML y otras funcionalidades
 - ✓ Herramientas para generar SQL portable para compartir con otros sistemas compatibles con SQL.
 - ✓ Sistema de tipos de datos extensible para proveer tipos de datos definidos por el usuario y rápido desarrollo de nuevos tipos.
- ✓ Funciones de compatibilidad para ayudar en la transición desde otros sistemas menos compatibles con SQL. [7]

1.11 Herramientas CASE

Las Herramientas CASE son un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del Ciclo de Vida de desarrollo de un Software.

¿Por qué se debería usar herramientas CASE de modelado con UML?

Las herramientas de modelado con UML ofrecen beneficios para todos los que participan en un proyecto, en la medida que los sistemas que se construyen se vuelven complejos. Estas nos permiten aplicar la metodología de análisis y diseño orientados a objetos y abstraernos del código fuente, en un

nivel donde la arquitectura y el diseño se tornan obvios y fáciles de entender y modificar. En presencia de un proyecto grande, es imprescindible el uso de una herramienta CASE.

Ejemplos de herramientas CASE:

- ASADAL - Herramienta CASE especializada en Sistemas de Tiempo Real.
- System Architect, herramientas CASE para Análisis y Diseño, incluye técnicas estructuradas y orientadas a objetos.
 - CRADLE, conjunto de herramientas CASE integradas que dan soporte a la Planificación estratégica, Análisis y Diseño.
 - PowerDesigner 7.0: herramienta CASE de Análisis y Diseño incluye capacidades de generación relacional y con orientación a objetos.
 - SilverRun: Conjunto integrado de herramientas CASE para el modelado de negocios.
 - Rational Rose, herramienta CASE para Análisis y Diseño basándose en el Proceso Unificado de Rational (RUP).
 - Visual Paradigm, herramienta CASE para Análisis y Diseño, utiliza el Lenguaje Unificado de Modelado (UML)

1.11.1 Selección de la herramienta CASE a utilizar:

La herramienta CASE seleccionada fue el Visual Paradigm porque nos brinda un entorno agradable de creación de diagramas para UML, diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad; posibilita el uso de un lenguaje común para todo el equipo de desarrollo facilitando la comunicación, brinda capacidades de ingeniería directa e inversa, modelo y código que permanece sincronizado en todo el ciclo de desarrollo, nos da la disponibilidad de múltiples versiones para cada necesidad y la de integrarse en los principales IDEs y es multiplataforma.

1.12 Otras Herramientas necesarias

Como se pretende implementar una aplicación Web para confeccionar la propuesta de este trabajo, se hace necesario tener en cuenta la utilización de un editor de páginas Web, y una herramienta para el trabajo con las imágenes.

Para estas funciones la elección no ha sido muy difícil, ya que la herramienta de creación de sitios Web más utilizada en la actualidad es el *Macromedia Dreamweaver 8*. Con esa herramienta se podrá desarrollar cualquier sitio Web personal con características de sitio profesional y utilizar casi todos los recursos de la Web, así como realizar aplicaciones que se ejecuten en servidor y hacer vinculaciones dinámicas de datos. Además de contar con un soporte para aplicaciones PHP y utilización de bases PostgreSQL, también cuenta con un amplio soporte para la creación y utilización de CSS (*Cascading Style Sheets*) para lograr un diseño fácil y óptimo.

Finalmente se escogió el *Adobe Photoshop 7* como herramienta principal para crear las imágenes del sistema, ya que se considera la aplicación estándar para el tratamiento digital de imágenes.

1.13 Conclusiones

En el presente capítulo se ha tratado de forma general y resumida, lo relacionado al estado del arte a nivel internacional, nacional y en nuestra universidad, así como, las tendencias, técnicas, tecnologías y metodologías de desarrollo de aplicaciones Web actuales que se utilizan en la solución de nuestro problema.

Se realizó un análisis completo de las tecnologías que serán utilizadas a lo largo del desarrollo del sistema propuesto, y se fundamentaron las elecciones del lenguaje, el sistema gestor de Base de Datos y la metodología a utilizar. Una vez conocidas las herramientas óptimas y los conceptos a utilizar se puede empezar a desarrollar la propuesta de sistema.

A square graphic with a grey background and a black border. Inside, the word "Capítulo" is written in a bold, black, serif font at the top. Below it, the Roman numeral "II" is written in a large, bold, black, serif font.**CAPITULO II: CARACTERISTICAS DEL SISTEMA****2.1 Introducción**

En el presente capítulo se describen las características que presenta el sistema, para esto se verán los procesos del negocio que tienen que ver con el objeto de estudio, a partir del análisis de éstos se puede percibir que debido a la falta de información existente y a la poca estructuración de esos procesos, se necesita definir conceptos que se pueden agrupar en un Modelo de Dominio para capturar correctamente los requisitos y poder construir un sistema eficiente.

Con esta captura correcta de los requisitos funcionales y no funcionales se hace una concepción general del sistema, y se logra identificar mediante un Diagrama de Casos de Uso, las relaciones de los actores que interactúan con el sistema y las secuencias de acciones con las que interactúan.

El objetivo del Proceso Unificado, dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental, es guiar a los desarrolladores de cualquier sistema de software en la implementación y distribución eficiente de sistemas que se ajusten a las necesidades de los clientes.

2.2 Problema y situación problemática

Hoy en Cuba se esta llevando a cabo un proceso de informatización por lo que las diferentes empresas o entidades se encuentran desarrollando sistemas de gestión de información. En la construcción de estos sistemas durante la etapa de análisis y diseño de las aplicaciones, por diferentes razones no se pronostican todos los requerimientos informativos que se deben tener en cuenta. Esto hace que se generen modelos estáticos de recuperación sin posibilidades de cambios, obligando a consultar los mismos que quizás muestren información más o menos abarcadora que la que se requiere. A medida que se utilizan las aplicaciones las necesidades que surgen son mayores que las que se brindan al usuario en un inicio. Por esta razón se hace imprescindible la realización de mantenimientos a los

sistemas, interrumpiendo así el funcionamiento óptimo del mismo, trayendo como consecuencia un menor rendimiento en la recuperación de la información y de manera inmediata el trabajo del desarrollador.

Después de un análisis de las situaciones antes expuestas; se puede decir que el **problema a resolver** es: la no existencia de una herramienta flexible que permita configurar la aplicación de la recuperación de información que se desee obtener.

2.3 Objetivos estratégicos de la organización

La organización tiene como objetivo estratégico preparar al país para responder en caso de cualquier tipo de agresión que se presente. Debido a esto es que el trabajo se centra en la realización de una herramienta Web que le permita a la aplicación de recuperaciones dinámicas de información adaptarse a cualquier sistema existente en el MINFAR, garantizando así que se produzca una eficiente y rápida recuperación de la información que se solicite.

2.4 Descripción de los procesos del negocio propuestos

Para obtener una descripción detallada de los procesos del negocio que se relacionan con el campo de acción, se hace necesario centrar la atención en los procesos de recuperación y la relación entre las aplicaciones mediante las cuales se realiza esta recuperación en el MINFAR.

A través de las técnicas de modelado que propone UML, se puede comprender mejor como se lleva a cabo la recuperación de información en el entorno al cual se hace referencia. El primer paso dentro del modelado del negocio es la identificación de los diferentes procesos del negocio que existen en la organización dirigidos al tema en cuestión, este paso es muy importante pues es el que establece los límites del modelado.

Actualmente cualquier persona previamente autorizada que desee recuperar algún tipo de información que se encuentra almacenada, necesita acceder a esta a través de consultas realizadas directamente al sistema de almacenamiento que se está empleando. En caso de aparecer complicaciones o presentar un bajo nivel de conocimientos, de manera que no pueda realizar la consulta, existe un encargado o técnico informático al cual debe consultar.

El mantenimiento de la información almacenada es realizado por los administradores o determinada persona con plenos permisos para guardar, transformar, insertar o eliminar cualquier dato.

2.5 Propuesta del sistema.

Para llevar a cabo el proceso de la recuperación eficiente y rápida de información mediante la configuración realizada a través de una herramienta Web, inicialmente el configurador, que no es más que el usuario autorizado a configurar una aplicación determinada para futuras recuperaciones de información, debe de autenticarse, pasando los datos del servidor de la Base de Datos en la que se quiere recuperar. Luego de esto, si los datos introducidos fueron correctos y además responde como administrador del sistema, se crea un esquema en esa misma Base de Datos llamado recuperaciones, que es por donde se van a construir en tiempo de ejecución las consultas para la recuperación de la información que se necesite. Este esquema de recuperaciones va a contener las características físicas de la Base de Datos a la que se conectó. Posibilitando así la configuración necesaria para una futura y eficiente recuperación. El usuario en su configuración puede administrar acceso, es decir, definir los usuarios que van a tener permiso o no a diversas tablas. También se puede revisar consistencia, que no es más que la forma de comprobar los cambios que pudieron haber ocurrido en la Base de Datos.

2.6 Modelo del Dominio

Considerando las descripciones de los procesos anteriormente comentados, se llega a la conclusión de que el negocio que se está estudiando tiene muy bajo nivel de estructuración, con soluciones muy diversas y dispersas, aunque todas llevan el mismo propósito de satisfacer una necesidad de recuperar información.

Para ello se utilizará un modelo del dominio, ya que permite de manera visual mostrar los principales conceptos que se manejan en el dominio del sistema en desarrollo. Esto ayuda a los usuarios, clientes y desarrolladores e interesados a utilizar un vocabulario común para poder entender el contexto en que se emplaza el sistema. Para capturar correctamente los requisitos y poder construir un sistema eficiente se necesita tener un firme conocimiento del funcionamiento del objeto de estudio. Este modelo va a contribuir posteriormente a identificar algunas clases que se utilizarán en el sistema.

2.6.1 Conceptos principales

Entidad: organización a la que pertenece el especialista.

Especialista: cualquier persona calificada y autorizada para gestionar o manipular la información que se quiere recuperar.

Aplicación Informática: sistema informático utilizado por el especialista para gestionar la información que se quiere recuperar.

Datos: conjunto organizado e integrado de información almacenada y clasificada en computadora o papel para su posterior consulta, actualización o cualquier tarea de mantenimiento mediante operaciones específicas.

Datos Digitales: representan datos almacenados en formato digital.

Datos Papel: representan datos almacenados en papel.

2.6.2 Principales Eventos

Pertenece: proceso que identifica al especialista especificando la entidad a la que pertenece.

Gestiona: proceso mediante el cual el especialista organiza la información de tal forma que se pueda recuperar la misma.

Configura: proceso mediante el cual el especialista transforma una aplicación informática para dar respuesta a una petición de información.

Accede: proceso mediante el cual la aplicación informática busca la información necesaria para dar respuesta a una petición.

2.6.3 Diagrama de clases del dominio

El modelo del dominio se describe mediante diagramas UML, específicamente con un diagrama de clases conceptuales significativas en el dominio del problema.

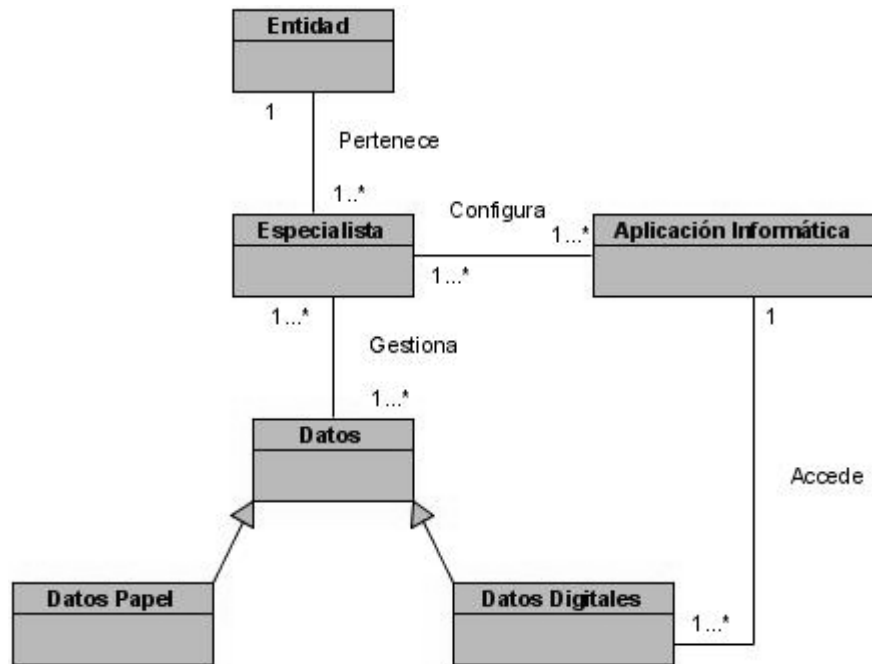


Fig. 2.1 Diagrama de clases del dominio

2.7 Requerimientos Funcionales

Una vez conocidos los conceptos que rodean al objeto de estudio, se debe analizar: ¿Qué debe hacer el sistema para que se cumplan los objetivos planteados al inicio de este trabajo?, para ello se enumeran, a través de requerimientos funcionales, las acciones que el sistema deberá ser capaz de realizar para satisfacer al usuario. Dentro de ellos se incluyen las acciones que podrán ser ejecutadas por el usuario, las acciones ocultas que debe realizar el sistema y las condiciones extremas a determinar por el sistema.

R1 Crear configuración inicial del sistema.

1.1 Autenticarse.

1.2 Obtener diseño físico de la BD.

R2 Gestionar Base de Datos.

2.1 Mostrar esquemas.

2.2 Actualizar esquemas.

2.3 Eliminar esquema.

2.4 Agregar esquema.

R3 Gestionar esquema de la Base de Datos.

3.1 Mostrar tablas del esquema seleccionado.

3.2 Actualizar tablas del esquema seleccionado.

3.3 Eliminar tabla del esquema seleccionado.

3.4 Agregar tabla en el esquema seleccionado.

R4 Gestionar tabla de un esquema.

4.1 Mostrar campos de la tabla seleccionada.

4.2 Actualizar campos de la tabla seleccionada.

4.3 Eliminar campo de la tabla seleccionada.

4.4 Agregar campo en la tabla seleccionada.

R5 Gestionar relaciones

5.1 Mostrar relaciones.

5.2 Insertar nueva relación.

5.3 Eliminar relación.

5.4 Modificar relación.

R6 Gestionar dependencias

6.1 Mostrar dependencias.

6.2 Crear nueva dependencia.

6.3 Eliminar dependencia.

6.4 Modificar dependencia.

R7 Gestionar accesos.

7.1 Mostrar sistemas.

7.2 Mostrar roles.

7.3 Mostrar tablas de un esquema seleccionado, definir tablas por roles y obligatorias.

7.4 Definir campos por roles.

R8 Revisar consistencia.

2.8 Requerimientos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto usable, rápido o confiable.

Apariencia o interfaz externa:

- El sistema debe tener una interfaz fácil de usar y amigable para que pueda ser utilizada sin mucho entrenamiento por el usuario.
- Estará diseñado para resolución de 800x600, aunque deberá verse en cualquier resolución superior a esta.
- Empleo de los colores: gris, blanco y azul principalmente, que son los definidos en los estándares del proyecto.

Usabilidad:

- El sistema debe estar disponible las 24 h del día.
- El sistema podrá ser usado por cualquier persona que posea conocimientos mínimos de base de datos y conocimientos medios en el manejo de la computadora.

Soporte:

- Se necesita un servidor de Bases de Datos que soporte grandes volúmenes de datos y velocidad de procesamiento.
- Un servidor Web que soporte y ejecute el código PHP 5.0 con PDO incluido.
- Los tiempos de respuesta y velocidad de procesamiento de la información serán rápidos.
- Por parte del cliente se requiere un navegador capaz de interpretar JavaScript.

Portabilidad:

- El sistema debe ser multiplataforma, haciendo énfasis en Linux y Windows.

Seguridad:

- Autenticación (Contraseña de acceso.)
- Las funcionalidades del sistema dependen directamente del acceso que tenga el usuario.
- Protección contra acciones no autorizadas o que puedan afectar la integridad de los datos.
- Verificación sobre acciones irreversibles (eliminaciones).

Legales:

.El sistema se basa en el manual de normas y principios establecidos por el MINFAR.

Software:

Para el cliente:

- Navegador Mozilla Firefox.
- Sistema operativo Linux, Windows 98 o superior.

Para el servidor:

- Sistema operativo Windows Advancer Server (2000 o superior) o Linux en cualquiera de sus distribuciones.
- Un servidor Apache 2.0 o superior con módulo PHP con PDO incluido, este debe estar configurado con la extensión "pgsql" incluida.
- Un servidor de base de datos PostgreSQL 8.0 o superior.

Hardware:

Para el servidor:

- Requerimientos mínimos: Procesador Pentium III a 1GHz de velocidad de procesamiento y 1Gb de memoria RAM.

- Al menos 40Gb de espacio libre en disco duro.
- Tarjeta de red.

Para el cliente:

- Requerimientos mínimos: Procesador Pentium II a 133Mhz con 128 Mb de memoria RAM.
- Tarjeta de red.

Restricciones para el diseño e implementación:

- Utilizar los estándares establecidos (codificación, diseño, entre otros)
- Emplear como servidores Web y de bases de datos Apache y PostgreSQL respectivamente.
- Utilizar como lenguaje del lado del servidor al PHP 5.0 con PDO incluido y del lado del cliente el JavaScript.

2.9 Modelo del Sistema.

El modelo del sistema es la base sobre la cual se desarrollan otros componentes. Se detalla en diagramas de bajo nivel que muestran características adicionales del sistema. Cada proceso puede desglosarse en diagramas de flujos de datos cada vez más detallados. Repitiéndose esta secuencia hasta que se obtienen suficientes detalles para que el analista comprenda la parte del sistema que se encuentra bajo investigación.

2.9.1 Definición de los casos de uso.

Los casos de uso se han convertido en la técnica más utilizada a nivel mundial para el levantamiento y la comunicación clara y eficiente de los requisitos para el desarrollo de sistemas.

Los casos de uso son un artefacto clave en el Proceso Unificado de desarrollo de software, ya que son el depósito principal de los requisitos funcionales que gobiernan el diseño, la construcción, las pruebas, y muchos otros aspectos de este proceso.

2.9.2 Definición de los actores.

Los actores no son ninguna parte del sistema, ellos representan a cualquiera o algo que debe interactuar con el sistema. Un actor puede que ser: usuarios, máquinas u otros sistemas.

Actores	Justificación
Configurador	Representa a una persona con acceso al sistema de configuración donde podrá definir diferentes eventos que serán utilizados para las recuperaciones dinámicas.

Tabla 2.1 Definición de los actores.

2.9.3 Descripción de los Casos de Uso del Sistema.

La descripción de los casos de usos del sistema, nos ayuda a entender bien qué es lo que el sistema debe hacer. Es decir, describen un uso del sistema y cómo este interactúa con el usuario. Esto se puede hacer de forma breve o extendida.

2.9.4 Descripciones no expandidas:

CU-1	Crear configuración inicial del sistema.
Actor	Configurador
Descripción	El caso de uso se inicia cuando el configurador introduce los datos para su autenticación, para que luego de esto se cree el esquema de recuperaciones, guardándose en él, el diseño físico de la Base de Datos. El caso de uso termina cuando el configurador cierra el sistema o accede a otras opciones de la aplicación.
Referencia	R1

Tabla 2.2 Crear configuración inicial del sistema.

CU-2	Gestionar Base de Datos.
Actor	Configurador

Descripción	El caso de uso se inicia cuando el configurador desea actualizar los esquemas pertenecientes a la Base de Datos en el esquema de recuperaciones. También se puede eliminar y/o agregar esquemas. El caso de uso termina cuando el configurador cierra el sistema o accede a otras opciones de la aplicación.
Referencia	R2

Tabla 2.3 Gestionar Base de Datos.

CU-3	Gestionar esquema de la Base de Datos.
Actor	Configurador
Descripción	El caso de uso se inicia cuando el configurador desea actualizar las tablas pertenecientes a un esquema de la Base de Datos en el esquema de recuperaciones. También se puede eliminar y/o agregar tablas del esquema seleccionado. El caso de uso termina cuando el configurador cierra el sistema o accede a otras opciones de la aplicación.
Referencia	R3

Tabla 2.4 Gestionar esquema de la Base de Datos.

CU-4	Gestionar tabla de un esquema.
Actor	Configurador
Descripción	El caso de uso se inicia cuando el configurador desea actualizar los campos pertenecientes a una tabla de un esquema determinado. También se puede eliminar y/o agregar campos. El caso de uso termina cuando el configurador cierra el sistema o accede a

	otras opciones de la aplicación.
Referencia	R4

Tabla 2.5 Gestionar tabla de un esquema.

CU-5	Gestionar relaciones.
Actor	Configurador
Descripción	El caso de uso inicia cuando el configurador desea crear una nueva relación, de acuerdo a su requerimiento puede modificar y/o eliminar alguna relación que él haya creado con anterioridad. El caso de uso termina cuando el configurador cierra el sistema o accede a otras opciones de la aplicación.
Referencia	R5

Tabla 2.6 Gestionar relaciones.

CU-6	Gestionar dependencias.
Actor	Configurador
Descripción	El caso de uso inicia cuando el configurador desea crear una nueva dependencia, de acuerdo a su requerimiento puede modificar y/o eliminar alguna dependencia. El caso de uso termina cuando el configurador cierra el sistema o accede a otras opciones de la aplicación.
Referencia	R6

Tabla 2.7 Gestionar dependencias.

CU-7	Gestionar accesos.
Actor	Configurador
Descripción	El caso de uso se inicia cuando el configurador desea

	definir las tablas y los campos por roles, y las tablas obligatorias. El caso de uso termina cuando el configurador cierra el sistema o accede a otras opciones de la aplicación.
Referencia	R7

Tabla 2.8 Gestionar accesos.

CU-8	Revisar consistencia.
Actor	Configurador
Descripción	El caso de uso se inicia cuando el configurador desea revisar consistencia, para esto verifica si han ocurrido cambios en la Base de Datos. El caso de uso termina cuando el configurador cierra el sistema o accede a otras opciones de la aplicación.
Referencia	R8

Tabla 2.9 Revisar consistencia.

2.9.5 Diagrama de Casos de Uso del Sistema

El diagrama de Casos de Usos del Sistema describe las interacciones del sistema con su entorno, identificando los Actores, que representan los diferentes roles desempeñados por los usuarios del sistema, y los Casos de Uso, que corresponden a la funcionalidad que el sistema ofrece a sus usuarios.

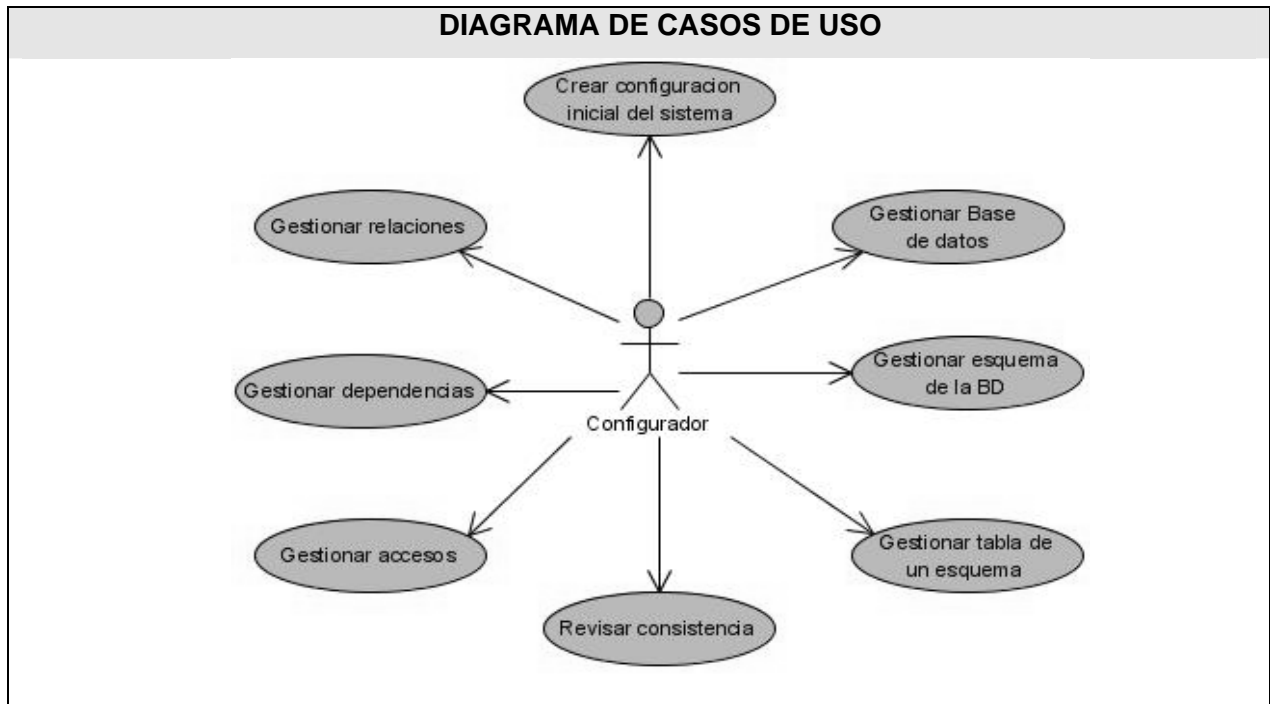



Fig. 2.2 Diagrama de casos de uso

2.9.6 Casos de uso expandidos.

Mediante los casos de uso expandidos se describe paso a paso la secuencia de eventos que los actores realizan para completar un proceso a través del sistema. Este sería el último paso en el análisis para pasar a la construcción de la solución propuesta.

2.9.7 Descripción del Caso de Uso Crear configuración inicial del sistema.

Caso de uso:	Crear configuración inicial del sistema.
Actores:	Configurador (inicia)
Propósito:	Crear el esquema donde se va a almacenar el diseño físico de la Base de Datos una vez obtenido.
Resumen	El caso de uso se inicia cuando el configurador introduce los datos para su autenticación, para que luego de esto se cree el esquema de recuperaciones, guardándose en él, el diseño físico de la Base de

	Datos. El caso de uso termina cuando el configurador cierra el sistema o accede a otras opciones de la aplicación.
Precondiciones:	
Poscondiciones:	Ha quedado creado el esquema de recuperaciones en la Base de Datos. Se ha obtenido el diseño físico de la Base de Datos en el esquema de recuperaciones.
Tipo:	Real y expandido.
Responsabilidades:	R1
Casos de uso relacionados:	
Interfaz 1	 <p>A: nombre: nombip, tipo: varchar (text para insertar el nombre o el IP donde va a estar la BD a la que se quiere conectar)</p> <p>B: nombre: usuario, tipo: varchar (text para insertar el usuario con el que se quiere conectar a una BD)</p> <p>C: nombre: contraseña: varchar (text para insertar la contraseña con la que se quiere conectar a una BD)</p> <p>D: nombre: nomb, tipo: varchar (text para insertar el nombre de la BD a la que se quiere conectar)</p> <p>E: nombre: aceptar (button para indicar al sistema la conformidad con los</p>

datos y dar curso a las acciones).	
Curso normal de eventos	
Acción del actor	Respuesta del sistema
<p>2. Introduce los datos para la autenticación como son: Nombre o IP de la BD a la que se quiere conectar (1-A), usuario (1-B), contraseña (1-C), nombre de la base de datos (1-D) y acciona el botón <Aceptar> (1-E).</p>	<p>1. Solicita autenticación.</p> <p>3. Se conecta a la Base de Datos.</p> <p>4. Comprueba si no existe el esquema de recuperaciones.</p> <p>5. Comprueba si puede crear el esquema de recuperaciones.</p> <p>6. Crea el esquema de recuperaciones.</p> <p>7. Almacena en este esquema el diseño físico obtenido de la Base de Datos.</p> <p>8. Almacena en este esquema los módulos y roles que obtiene mediante el servicio de seguridad.</p> <p>9. Muestra la Interfaz inicial del sistema.</p>
Cursos alternos	
<p>Línea 3: Muestra mensaje de error diciendo “No se puede conectar a la Base de Datos”.</p> <p>Línea 5: Muestra la Interfaz inicial del sistema.</p> <p>Línea 6: Muestra mensaje de error diciendo que “No tiene permisos en la Base de Datos para poder crear el esquema recuperaciones” y se mantiene en la interfaz de autenticarse para que vuelva a introducir los</p>	

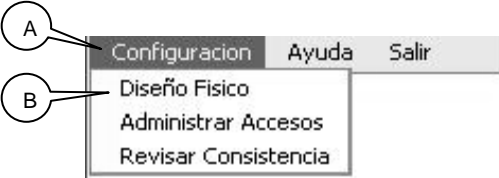
datos.

Tabla 2.10 Crear configuración inicial del sistema.

2.9.8 Descripción del Caso de Uso Gestionar Base de Datos.

Caso de uso:	Gestionar Base de Datos.
Actores:	Configurador (inicia).
Propósito:	Actualizar los esquemas existentes en la Base de Datos almacenada en el esquema de recuperaciones.
Resumen	El caso de uso se inicia cuando el configurador desea actualizar los esquemas pertenecientes a la Base de Datos en el esquema de recuperaciones. También se puede eliminar y/o agregar esquemas. El caso de uso termina cuando el configurador cierra el sistema o accede a otras opciones de la aplicación.
Precondiciones:	Debe haber sido autenticado como configurador. Debe estar almacenado el diseño de la Base de Datos en el esquema de recuperaciones.
Poscondiciones:	Han quedado actualizados los esquemas de la Base de Datos en el esquema de recuperaciones.
Tipo:	Real y expandido.
Responsabilidades:	R2
Casos de uso relacionados:	

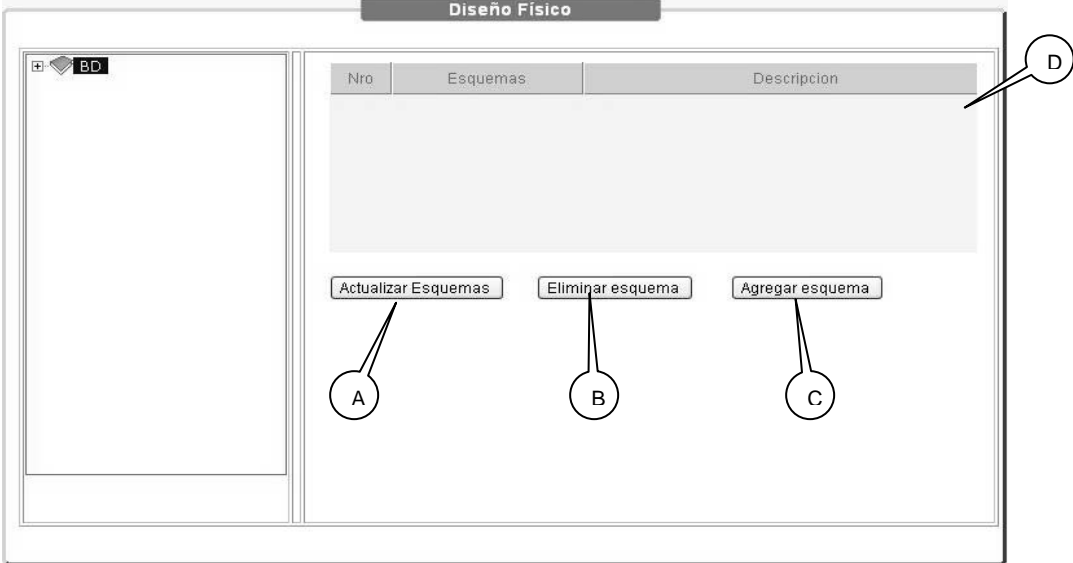
Interfaz 2



A: menú desplegable que permite seleccionar las diferentes configuraciones ha realizar.

B: indica una de las configuraciones que se pueden realizar.

Interfaz 3



A: nombre: actualizar esquema (button para indicar al sistema que se desea actualizar todos los esquemas existentes en la BD).

B: nombre: eliminar esquema (button para indicar al sistema que se desea eliminar un esquema).

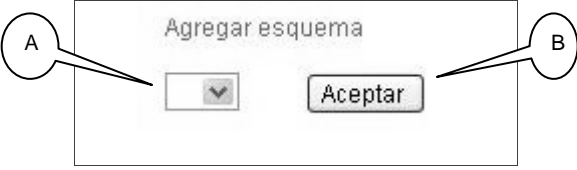
C: nombre: aceptar (button para indicar al sistema que se desea agregar un esquema).

D: nombre: esquemas (Grid para mostrar la lista de nombres de los esquemas existentes en la BD, y una breve descripción de cada uno de ellos)

Curso normal de eventos para el caso de uso

Acción del actor	Respuesta del sistema
<p>1. Selecciona la opción Configuración del menú principal. (Interfaz 2-A)</p> <p>3. Selecciona la opción Diseño Físico (2-B)</p> <p>5. El actor decide: Si desea actualizar el diseño de los esquemas pertenecientes a la Base de Datos en el esquema de recuperaciones ver sección 1.</p> <p>Si desea eliminar esquema ver sección 2.</p> <p>Si desea agregar esquema ver sección 3.</p>	<p>2. Muestra menú con las opciones Diseño Físico, Administrar Accesos y Revisar Consistencia.</p> <p>4. Muestra la interfaz 3.</p>
Sección 1	
Acción del actor	Respuesta del sistema
<p>5. Acciona el botón <Actualizar esquemas> (3-A)</p>	<p>6. Muestra mensaje diciendo al usuario si está</p>

<p>7. Selecciona "Aceptar".</p>	<p>seguro que desea actualizar el diseño de los esquemas pertenecientes a la Base de Datos en el esquema de recuperaciones.</p> <p>8. Realiza la actualización del diseño de los esquemas pertenecientes a la Base de Datos en el esquema de recuperaciones.</p> <p>9. Muestra el grid (3-D) con todos los esquemas existentes en la Base de Datos.</p>
Sección 2	
Acción del actor	Respuesta del sistema
<p>5. Selecciona del grid (3-D) el esquema que desea eliminar.</p> <p>6. Acciona el botón <Eliminar> (3-B).</p> <p>9. Selecciona "Aceptar".</p>	<p>7. Comprueba si no existe el esquema en la Base de Datos.</p> <p>8. Muestra mensaje diciendo al usuario si está seguro que desea eliminar la selección.</p> <p>10. Elimina el esquema.</p> <p>11. Muestra el grid (3-D) con los esquemas restantes de la Base de Datos.</p>
Sección 3	

<p>Interfaz 4</p> 	
<p>A: nombre: agresq (lista/menu para seleccionar el esquema que se desea agregar)</p> <p>B: nombre: aceptar (button para indicar al sistema que quiere agregar el esquema).</p>	
Acción del actor	Respuesta del sistema
<p>5. Acciona el botón <Agregar esquema> (3-C).</p> <p>7. Escoge el esquema que desea agregar.</p> <p>8. Acciona el botón <Aceptar> (4-B).</p>	<p>6. Muestra la interfaz 4 con el lista/menu (4-A) de todos los esquemas de la Base de Datos.</p> <p>9. Agrega el esquema.</p> <p>10. Muestra el grid (3-D) con los esquemas existentes.</p>
<p align="center">Cursos alternos</p>	
<p>Sección 1:</p> <p>Línea 7: Si el actor responde "Cancelar" el sistema no ejecuta la acción, y no se actualiza los esquemas.</p> <p>Sección 2:</p> <p>Línea 8: En caso que el esquema seleccionado exista en la Base de Datos, muestra mensaje "El esquema seleccionado no se puede eliminar por existir en</p>	

la Base de Datos”, y no se elimina dicho esquema.

Línea 9: Si el actor responde “Cancelar” el sistema no ejecuta la acción, y no elimina el esquema seleccionado.

Sección 3:

Línea 10: En caso que el esquema seleccionado exista, muestra mensaje “El esquema seleccionado ya existe”, y no agrega dicho esquema.

Tabla 2.11 Gestionar Base de Datos.

2.9.9 Descripción del Caso de Uso Gestionar esquema de la Base de Datos.

Caso de uso:	Gestionar esquema de la Base de Datos.
Actores:	Configurador (inicia).
Propósito:	Actualizar las tablas pertenecientes a un esquema existente en la Base de Datos almacenada en el esquema de recuperaciones.
Resumen	El caso de uso se inicia cuando el configurador desea actualizar las tablas pertenecientes a un esquema de la Base de Datos en el esquema de recuperaciones. También se puede eliminar y/o agregar tablas del esquema seleccionado. El caso de uso termina cuando el configurador cierra el sistema o accede a otras opciones de la aplicación.
Precondiciones:	Debe haber sido autenticado como configurador. Debe estar almacenado el diseño de la Base de Datos en el esquema de recuperaciones.
Poscondiciones:	Han quedado actualizadas las tablas pertenecientes a un esquema de la Base de Datos almacenada en el esquema de recuperaciones.
Tipo:	Real y expandido.

Responsabilidades:	R3
Casos de uso relacionados:	

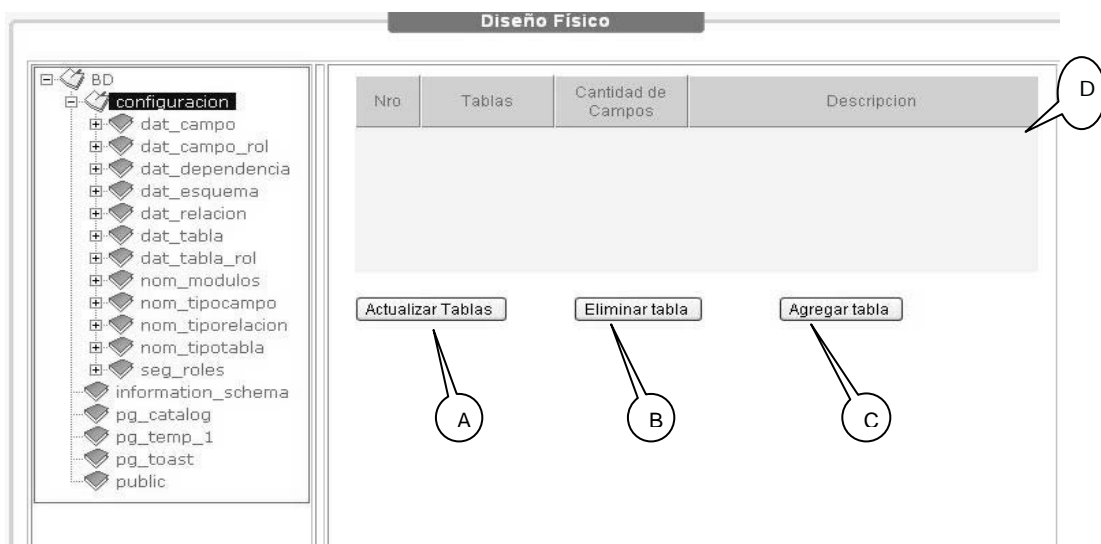
Interfaz 2



A: menú desplegable que permite seleccionar las diferentes configuraciones ha realizar.

B: indica una de las configuraciones que se pueden realizar.

Interfaz 5




A: nombre: actualizar tabla (button para indicar al sistema que se desean actualizar todas las tablas existentes en el esquema seleccionado de la BD).

B: nombre: eliminar tabla (button para indicar al sistema que se desea eliminar una tabla del esquema seleccionado de la BD).

<p>C: nombre: agregar tabla (button para indicar al sistema que se desean agregar una tabla en el esquema seleccionado de la BD).</p> <p>D: nombre: tablas (Grid para mostrar la lista de nombres de las tablas existentes en el esquema seleccionado de la BD, y una breve descripción de cada una de ellas).</p>	
Curso normal de eventos para el caso de uso	
Acción del actor	Respuesta del sistema
<p>1. Selecciona la opción Configuración del menú principal (2-A).</p> <p>3. Selecciona la opción Diseño Físico (2-B).</p> <p>5. Selecciona el esquema que desea gestionar.</p> <p>7. El actor decide: Si desea actualizar el diseño de las tablas pertenecientes al esquema seleccionado de la Base de Datos almacenada en el esquema de recuperaciones ver sección 1. Si desea eliminar tabla ver sección 2. Si desea agregar tabla ver sección 3.</p>	<p>2. Muestra menú con las opciones Diseño Físico, Administrar Accesos y Revisar Consistencia.</p> <p>4. Muestra la interfaz 3.</p> <p>6. Muestra la interfaz 5.</p>

Sección 1	
Acción del actor	Respuesta del sistema
<p>7. Acciona el botón <Actualizar tablas> (5-A).</p> <p>9. Selecciona "Aceptar".</p>	<p>8. Muestra mensaje diciendo al usuario si está seguro que desea actualizar el diseño de las tablas pertenecientes al esquema seleccionado de la Base de Datos almacenada en el esquema de recuperaciones.</p> <p>10. Realiza la actualización del diseño de las tablas pertenecientes al esquema seleccionado de la Base de Datos almacenada en el esquema de recuperaciones.</p> <p>11. Muestra el grid (5-D) con todas las tablas pertenecientes al esquema actualizado.</p>
Sección 2	
Acción del actor	Respuesta del sistema
<p>7. Selecciona del grid (5-D) la tabla que desea eliminar.</p> <p>8. Acciona el botón <Eliminar tabla> (5-B).</p> <p>11. Selecciona "Aceptar".</p>	<p>9. Comprueba si no existe la tabla en la Base de Datos.</p> <p>10. Muestra mensaje diciendo al usuario si está seguro que desea eliminar la selección.</p> <p>11. Elimina la tabla.</p> <p>12. Muestra el grid (5-D) con las tablas restantes.</p>
Sección 3	

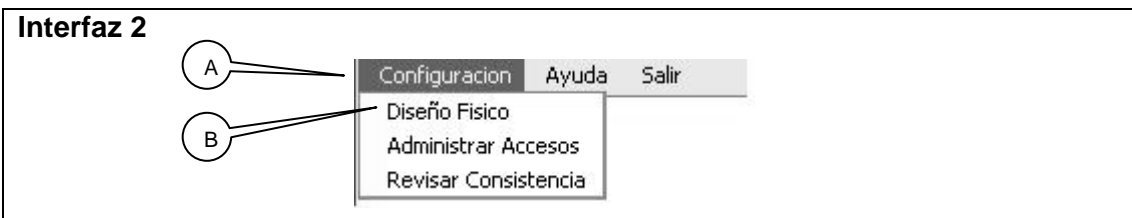
<p>Interfaz 6</p> 	
<p>A: nombre: agretab (lista/menú para seleccionar la tabla que se desea agregar)</p> <p>B: nombre: aceptar (button para indicar al sistema que quiere agregar la tabla).</p>	
<p>Acción del actor</p>	<p>Respuesta del sistema</p>
<p>6. Acciona el botón <Agregar tabla (5-C).</p> <p>8. Escoge la tabla que desea agregar.</p> <p>9. Acciona el botón <Aceptar> (6-B).</p>	<p>7. Muestra la interfaz 6 con el lista/menú (A) de todas las tablas pertenecientes al esquema seleccionado de la Base de Datos.</p> <p>10. Agrega la tabla.</p> <p>11. Muestra el grid (5-D) con las tablas existentes en el esquema seleccionado.</p>
<p>Cursos alternos</p>	
<p>Sección 1:</p> <p>Línea 9: Si el actor responde “Cancelar” el sistema no ejecuta la acción, y no actualiza las tablas.</p> <p>Sección 2:</p> <p>Línea 10: En caso que la tabla seleccionada exista en la Base de Datos, muestra mensaje “La tabla seleccionada no se puede eliminar por existir en la Base de Datos”, y no se elimina dicha tabla.</p> <p>Línea 11: Si el actor responde “Cancelar” el sistema no ejecuta la acción, y no elimina la tabla seleccionada.</p> <p>Sección 3:</p>	

Línea 12: En caso que la tabla seleccionada exista, muestra mensaje “La tabla seleccionada ya existe”, y no agrega dicha tabla.

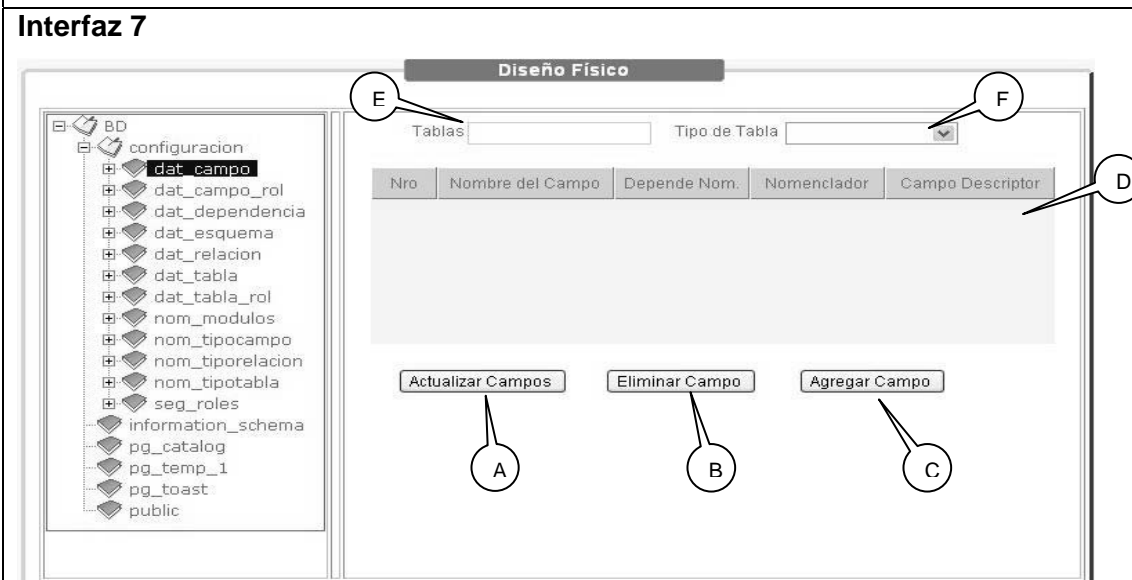
Tabla 2.12 Gestionar esquema de la Base de Datos.

2.9.10 Descripción del Caso de Uso Gestionar tabla de un esquema.

Caso de uso:	Gestionar tabla de un esquema
Actores:	Configurador (inicia).
Propósito:	Actualizar los campos pertenecientes a una tabla de un esquema existente en la Base de Datos almacenada en el esquema de recuperaciones.
Resumen	El caso de uso se inicia cuando el configurador desea actualizar los campos pertenecientes a una tabla de un esquema determinado. También se puede eliminar y/o agregar campos. El caso de uso termina cuando el configurador cierra el sistema o accede a otras opciones de la aplicación.
Precondiciones:	Debe haber sido autenticado como configurador. Debe estar almacenado el diseño de la Base de Datos en el esquema de recuperaciones.
Poscondiciones:	Han quedado actualizados los campos pertenecientes a una tabla de un esquema determinado de la Base de Datos almacenada en el esquema de recuperaciones.
Tipo:	Real y expandido.
Responsabilidades:	R4
Casos de uso relacionados:	




A: menú desplegable que permite seleccionar las diferentes configuraciones ha realizar.
B: indica una de las configuraciones que se pueden realizar.



A: nombre: actualizar campos (button para indicar al sistema que se desean actualizar todos los campos existentes en la tabla seleccionada perteneciente a un esquema de la BD).
B: eliminar campo (button para indicar al sistema que se desea eliminar un campo existente en la tabla seleccionada perteneciente a un esquema de la BD).
C: agregar campos (button para indicar al sistema que se desea agregar un campo en la tabla seleccionada perteneciente a un esquema de la BD).
D: nombre: campos (Grid para mostrar la lista de nombres de los campos existentes en una tabla seleccionada del esquema de la BD, una breve descripción de cada uno de ellos y si depende o no de un nomenclador y cual es).

<p>E: nombre: tabla, tipo: varchar (text que te muestra el nombre de la tabla a la que se le quiere dar una categoría determinada)</p> <p>F: nombre: tiptabla (lista/menú para seleccionar la categoría que tendrá la tabla seleccionada)</p>	
Curso normal de eventos para el caso de uso	
Acción del actor	Respuesta del sistema
<p>1. Selecciona la opción Configuración del menú principal (2-A).</p> <p>3. Selecciona la opción Diseño Físico (2-B).</p> <p>5. Selecciona un esquema.</p> <p>6. Selecciona una tabla perteneciente al esquema escogido anteriormente.</p> <p>8. El actor decide: Si desea actualizar el diseño de los campos pertenecientes a la tabla seleccionada del esquema escogido con anterioridad de la Base de Datos almacenada en el esquema de recuperaciones ver</p>	<p>2. Muestra menú con las opciones Diseño Físico, Administrar Accesos y Revisar Consistencia.</p> <p>4. Muestra la interfaz 3.</p> <p>7. Muestra la interfaz 7.</p>

<p>sección 1.</p> <p>Si desea eliminar campo ver sección 2.</p> <p>Si desea agregar campo ver sección 3.</p>	
Sección 1	
Acción del actor	Respuesta del sistema
<p>8. Acciona el botón <Actualizar campos> (7-A).</p> <p>10. Selecciona "Aceptar".</p>	<p>9. Muestra mensaje diciendo al usuario si está seguro que desea actualizar el diseño de los campos pertenecientes a una tabla almacenada en el esquema de recuperaciones.</p> <p>11. Realiza la actualización de los campos de la tabla seleccionada perteneciente al esquema escogido con anterioridad almacenado en el esquema de recuperaciones.</p> <p>12. Muestra el grid (7-D) con todos los campos pertenecientes a esa tabla.</p>
Sección 2	
Acción del actor	Respuesta del sistema
<p>8. Selecciona del grid (7-D) el campo que desea eliminar.</p> <p>9. Acciona el botón <Eliminar> (7-B).</p>	<p>10. Comprueba si no existe el campo en la Base de Datos.</p> <p>11. Muestra mensaje diciendo al usuario si está</p>


<p>12. Selecciona "Aceptar".</p>	<p>seguro que desea eliminar la selección.</p> <p>13. Elimina el campo.</p> <p>14. Muestra el grid (7-D) con los campos restantes.</p>
<p>Sección 3</p>	
<p>Interfaz 8</p> 	
<p>A: nombre: agrecamp (lista/menú para seleccionar el campo que se desea agregar)</p> <p>B: nombre: aceptar (button para indicar al sistema que quiere agregar el campo).</p>	
<p>Acción del actor</p>	<p>Respuesta del sistema</p>
<p>8. Acciona el botón <Agregar campo> (7-C).</p> <p>10. Escoge el campo que desea agregar.</p> <p>11. Acciona el botón <Aceptar> (8-B).</p>	<p>9. Muestra la interfaz 8 con el lista/menú (A) de todos los campos de la tabla.</p> <p>12. Agrega el esquema.</p> <p>13. Muestra el grid (7-D) con los campos existentes.</p>
<p>Cursos alternos</p>	

<p>Sección 1:</p> <p>Línea 10: Si el actor responde “Cancelar” el sistema no ejecuta la acción, y no actualiza los campos.</p> <p>Sección 2:</p> <p>Línea 11: En caso que el campo seleccionado exista en la Base de Datos, muestra mensaje “El campo seleccionado no se puede eliminar por existir en la Base de Datos”, y no se elimina dicho campo.</p> <p>Línea 12: Si el actor responde “Cancelar” el sistema no ejecuta la acción, y no elimina el campo seleccionado.</p> <p>Sección 3:</p> <p>Línea 12: En caso que el campo seleccionado exista, muestra mensaje “El campo seleccionado ya existe”, y no agrega dicho campo.</p>
--

Tabla 2.13 Gestionar tabla de un esquema Descripción del Caso de Uso

2.9.11 Descripción del Caso de Uso Gestionar relaciones.

Caso de uso:	Gestionar relaciones.
Actores:	Configurador (inicia).
Propósito:	Definir relaciones incompletas entre tablas de la Base de Datos en el esquema de recuperaciones.
Resumen	El caso de uso inicia cuando el configurador desea definir una relación, de acuerdo a su requerimiento puede modificar y/o eliminar alguna relación que él haya creado con anterioridad. El caso de uso termina cuando el configurador cierra el sistema o accede a otras opciones de la aplicación.
Precondiciones:	Debe haber sido autenticado como configurador. Debe estar almacenado el diseño de la Base de Datos en el esquema de recuperaciones. En caso de que quiera eliminar o modificar una

	relación, esta debe estar previamente definida en el esquema de recuperaciones y debe haber sido creada por él mismo.
Poscondiciones:	Las relaciones han sido definidas.
Tipo:	Real y expandido.
Responsabilidades:	R5
Casos de uso relacionados:	
Interfaz 2	 A screenshot of a software menu. The menu has three main items: 'Configuración', 'Ayuda', and 'Salir'. The 'Configuración' item is expanded, showing a sub-menu with three options: 'Diseño Físico', 'Administrar Accesos', and 'Revisar Consistencia'. Callout 'A' points to the 'Configuración' item, and callout 'B' points to the 'Diseño Físico' option in the sub-menu.
	<p>A: menú desplegable que permite seleccionar las diferentes configuraciones ha realizar.</p> <p>B: indica una de las configuraciones que se pueden realizar.</p>

Interfaz 9

A: nombre: **esquema1**, tipo: **varchar** (text que muestra el esquema donde se quiere establecer la relación)

B: nombre: **esquema** (lista/menú para seleccionar el esquema con el que se desea establecer la relación)

C: nombre: **tabla1**, tipo: **varchar** (text que muestra la tabla donde se quiere establecer la relación)

D: **tabla** (lista/menú para seleccionar la tabla con la que se desea establecer la relación)

E: **campo1** (lista/menú para seleccionar el campo con el que se desea establecer la relación)

F: **campo2** (lista/menú para seleccionar el campo con el que se desea establecer la relación)

G: nombre: **insertar** (button para indicar al sistema que quiere insertar la relación).

H: nombre: relaciones (Grid para mostrar la lista de nombres de los esquemas, tablas y campos entre los cuales existen relaciones).

Nro	Esquema 1	Tabla 1	Campo 1	Esquema 2	Tabla 2

<p>I: nombre: eliminar (button para indicar al sistema que quiere eliminar la relación).</p> <p>J: nombre: modificar (button para indicar al sistema que quiere modificar la relación).</p>	
<p>Curso normal de eventos para el caso de uso</p>	
Acción del actor	Respuesta del sistema
<p>1. Selecciona la opción Configurar del menú principal (2-A).</p> <p>3. Selecciona la opción Diseño Físico (2-B) de la interfaz I.</p> <p>5. Selecciona un esquema.</p> <p>6. Selecciona una tabla del esquema escogido.</p> <p>7. Selecciona de la tabla escogida relaciones.</p> <p>9. El actor decide: Si desea insertar una relación ver sección 1. Si desea eliminar una relación ver sección 2. Si desea modificar una relación ver sección 3.</p>	<p>2. Muestra menú con las opciones Diseño Físico, Administrar Accesos y Revisar Consistencia.</p> <p>4. Muestra la interfaz 3.</p> <p>8. Muestra la interfaz 9.</p>
<p>Sección 1</p>	

Acción del actor	Respuesta del sistema
<p>10. Selecciona el esquema (9-A), tabla (9-C) y el campo (9-E) del primer miembro de la relación que desea crear.</p> <p>11. Selecciona el esquema (9-B), tabla (9-D) y el campo (9-F) del segundo miembro de la relación.</p> <p>12. Acciona el botón <Insertar> (9-G).</p>	<p>9. Muestra la interfaz 9 solicitando que seleccione el esquema1 (9-A), tabla1 (9-C), campo1 (9-E), esquema2 (9-B), tabla2 (9-D) y campo2 (9-F) para crear una nueva relación.</p> <p>13. Comprueba si la relación existe.</p> <p>14. Inserta la nueva relación en el esquema de recuperaciones.</p> <p>15. Muestra el grid (9-H) con la relación creada en la lista de relaciones existentes.</p>
Sección 2	
Acción del actor	Respuesta del sistema
<p>9. Selecciona del grid (9-H) de la interfaz II la relación que desea eliminar.</p> <p>10. Acciona el botón <Eliminar> (9-I).</p>	<p>11. Comprueba si la relación fue creada por el configurador.</p> <p>12. Muestra mensaje preguntando al usuario si</p>

13. Selecciona "Aceptar".	<p>está seguro que desea eliminar la selección.</p> <p>14. Elimina la relación del esquema de recuperaciones.</p> <p>15. Muestra el grid (9-H) con las relaciones existentes.</p>
Sección 3	
Acción del actor	Respuesta del sistema
<p>9. Selecciona del grid (9-H) la relación que desea modificar.</p> <p>10. Acciona el botón < Modificar > (9-J).</p> <p>13. Modifica los datos de la relación seleccionada.</p> <p>14. Acciona el botón < Aceptar > (9-K).</p> <p>16. Selecciona "Aceptar".</p>	<p>11. Comprueba si la relación fue creada por el configurador.</p> <p>12. Se muestra en la interfaz 9 los datos correspondientes de la relación seleccionada.</p> <p>15. Muestra mensaje diciendo al usuario si está seguro que desea modificar.</p> <p>17. Comprueba si la relación existe.</p> <p>18. Modifica la relación en el esquema de recuperaciones.</p> <p>19. Muestra el grid (9-H) con las relaciones existentes.</p>
Cursos alternos	

<p>Sección 1:</p> <p>Línea 14: El sistema muestra mensaje de error diciendo que “Ya existe la relación”.</p> <p>Sección 2:</p> <p>Línea 12: Si la relación es original se mostrará un mensaje. “No se puede eliminar relación por no haber sido creada por el configurador”.</p> <p>Línea 13: Si el actor responde “Cancelar” el sistema no ejecuta la acción, y no elimina la relación seleccionada.</p> <p>Sección 3:</p> <p>Línea 12: Si la relación es original se mostrara un mensaje. “No se puede modificar relación por no haber sido creada por el configurador”.</p> <p>Línea 16: Si el actor responde “Cancelar” el sistema no ejecuta la acción, y no modifica la relación.</p> <p>Línea 18: Si ya existe una relación con los mismos elementos, se mostrará un mensaje de error “Ya existe esa relación”.</p>

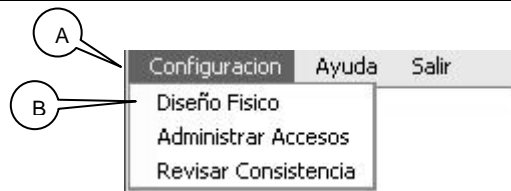
Tabla 2.14 Gestionar relaciones.

2.9.12 Descripción del Caso de Uso gestionar dependencias.

Caso de uso:	Gestionar dependencias
Actores:	Configurador (inicia).
Propósito:	Definir dependencias entre campos.
Resumen	El caso de uso inicia cuando el configurador desea crear una nueva dependencia, de acuerdo a su requerimiento puede modificar y/o eliminar alguna dependencia. El caso de uso termina cuando el configurador cierra el sistema o accede a otras opciones de la aplicación.
Precondiciones:	Debe haber sido autenticado como configurador. Debe estar almacenado el diseño de la Base de Datos en el esquema de recuperaciones.

	En caso de que quiera eliminar o modificar una dependencia, esta debe estar previamente definida en el esquema de recuperaciones y debe haber sido creada por él mismo.
Poscondiciones:	Han quedado actualizadas las dependencias.
Tipo:	Real y expandido.
Responsabilidades:	R6
Casos de uso relacionados:	

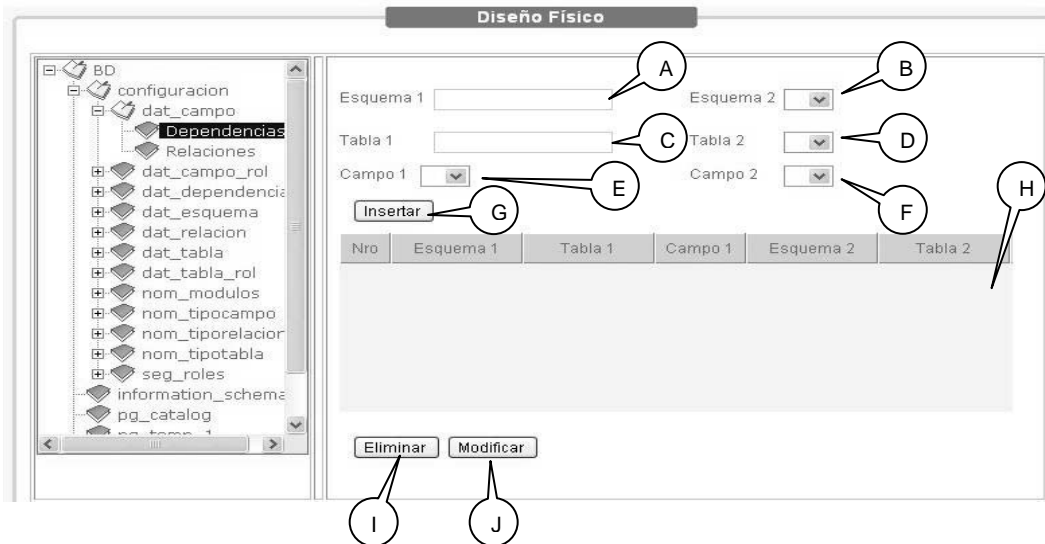
Interfaz 2



A: menú desplegable que permite seleccionar las diferentes configuraciones ha realizar.

B: indica una de las configuraciones que se pueden realizar.

Interfaz 10



A: nombre: esquema1, tipo: varchar (text que muestra el esquema donde se quiere establecer la dependencia)

B: nombre: esquema2 (lista/menú para seleccionar el esquema con el que se desea establecer la dependencia)

C: nombre: tabla1, tipo: varchar (text que muestra la tabla en la que se quiere establecer la dependencia)

D: tabla2 (lista/menú para seleccionar la tabla con la que se desea establecer la dependencia)

E: campo1 (lista/menú para seleccionar el campo con el que se desea establecer la dependencia)

F: campo2 (lista/menú para seleccionar el campo con el que se desea establecer la dependencia)

G: nombre: insertar (button para indicar al sistema que quiere insertar la dependencia).

H: nombre: relaciones (Grid para mostrar la lista de nombres de los esquemas, tablas y campos entre los cuales existen dependencia).

I: nombre: eliminar (button para indicar al sistema que quiere eliminar la dependencia).

J: nombre: modificar (button para indicar al sistema que quiere modificar la dependencia).

Curso normal de eventos para el caso de uso

Acción del actor	Respuesta del sistema
1. Selecciona la opción Configurar del menú principal (2-A).	2. Muestra menú con las opciones Diseño Físico, Administrar Accesos y Revisar Consistencia.
3. Selecciona la opción Diseño Físico (2-B).	
5. Selecciona un	4. Muestra la interfaz 3.


<p>esquema.</p> <p>6. Selecciona una tabla del esquema escogido.</p> <p>7. Selecciona de la tabla escogida dependencia.</p> <p>9. El actor decide: Si desea insertar una dependencia ver sección 1.</p> <p>Si desea eliminar una dependencia ver sección 2.</p> <p>Si desea modificar una dependencia ver sección 3.</p>	<p>8. Muestra la interfaz 10.</p>
<p>Sección 1</p>	
<p>Acción del actor</p>	<p>Respuesta del sistema</p>
<p>10. Selecciona el esquema (10-A), tabla (10-C) y el campo (10-E) del primer miembro de la dependencia que desea crear.</p> <p>11. Selecciona el esquema (10-B), tabla</p>	<p>9. Muestra la interfaz 10 solicitando que seleccione esquema1 (10-A), tabla1 (10-C), campo1 (10-E), esquema2 (10-B), tabla2 (10-D) y campo2 (10-F) para crear una nueva dependencia.</p>

<p>(10-D) y el campo (10-F) del segundo miembro de la dependencia.</p> <p>12. Acciona el botón <Insertar> (10-G).</p>	<p>13. Comprueba si la dependencia existe.</p> <p>14. Inserta la nueva dependencia en el esquema de recuperaciones.</p> <p>15. Muestra el grid (10-H) con la dependencia creada en la lista de dependencias existentes.</p>
Sección 2	
Acción del actor	Respuesta del sistema
<p>9. Selecciona del grid (10-H) la dependencia que desea eliminar.</p> <p>10. Acciona el botón <Eliminar> (10-I).</p> <p>12. Selecciona "Aceptar".</p>	<p>11. Muestra mensaje preguntando al usuario si está seguro que desea eliminar la selección.</p> <p>13. Elimina la dependencia del esquema de recuperaciones.</p> <p>14. Muestra el grid (10-H) con las dependencia existentes.</p>
Sección 3	
Acción del actor	Respuesta del sistema
<p>9. Selecciona del grid (10-H) la dependencia que desea modificar.</p>	

<p>10. Acciona el botón < Modificar > (10-J).</p> <p>12. Modifica los datos de la dependencia seleccionada.</p> <p>13. Acciona el botón < Aceptar > (10-K).</p> <p>15. Selecciona "Aceptar".</p>	<p>11. Se muestra en la interfaz 10 los datos correspondientes de la dependencia seleccionada.</p> <p>14. Muestra mensaje diciendo al usuario si está seguro que desea modificar.</p> <p>16. Comprueba si la dependencia existe.</p> <p>17. Modifica la dependencia en el esquema de recuperaciones.</p> <p>18. Muestra el grid (10-H) con las dependencias existentes.</p>
Cursos alternos	
<p>Sección 1:</p> <p>Línea 14: El sistema muestra mensaje de error diciendo que "Ya existe la dependencia".</p> <p>Sección 2:</p> <p>Línea 12: Si el actor responde "Cancelar" el sistema no ejecuta la acción, y no elimina la dependencia seleccionada.</p> <p>Sección 3:</p> <p>Línea 15: Si el actor responde "Cancelar" el sistema no ejecuta la acción, y no modifica la dependencia.</p> <p>Línea 17: Si ya existe una dependencia con los mismos elementos, se mostrará un mensaje de error "Ya existe esa dependencia".</p>	

Tabla 2.15 Gestionar dependencias

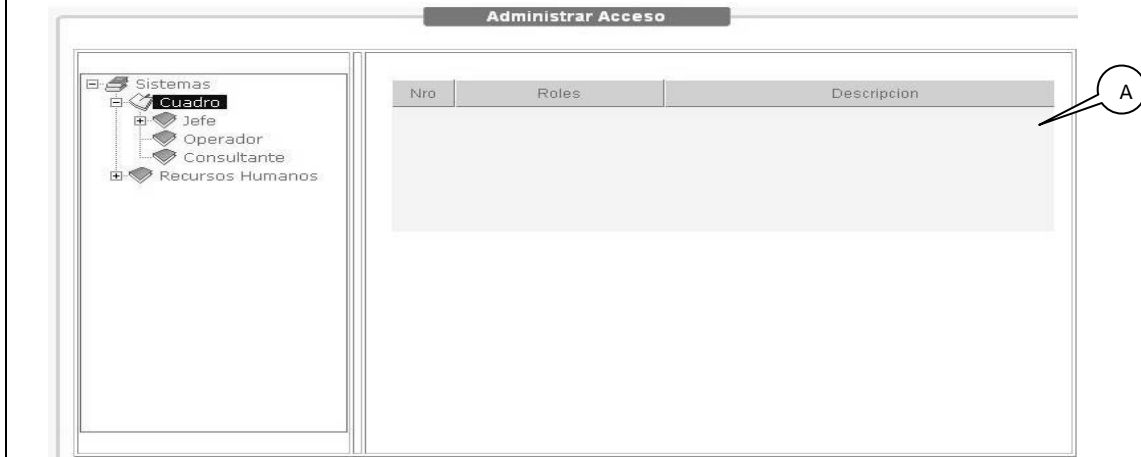
2.9.13 Descripción del Caso de Uso Gestionar accesos.

Caso de uso:	Gestionar accesos.
Actores:	Configurador (inicia).
Propósito:	Definir las tablas y los campos por roles, y las tablas obligatorias.
Resumen	El caso de uso se inicia cuando el configurador desea definir las tablas y los campos por roles, y las tablas obligatorias. El caso de uso termina cuando el configurador cierra el sistema o accede a otras opciones de la aplicación.
Precondiciones:	Debe haber sido autenticado como configurador. Debe estar almacenado el diseño de la Base de Datos en el esquema de recuperaciones.
Poscondiciones:	Han quedado definidas las tablas y los campos por roles, y las tablas obligatorias.
Tipo:	Real y expandido.
Responsabilidades:	R7
Casos de uso relacionados:	
Interfaz 2	
	
<p>A: menú desplegable que permite seleccionar las diferentes configuraciones ha realizar.</p> <p>B: indica una de las configuraciones que se pueden realizar.</p> <p>C: indica una de las configuraciones que se pueden realizar.</p>	
Interfaz 11	



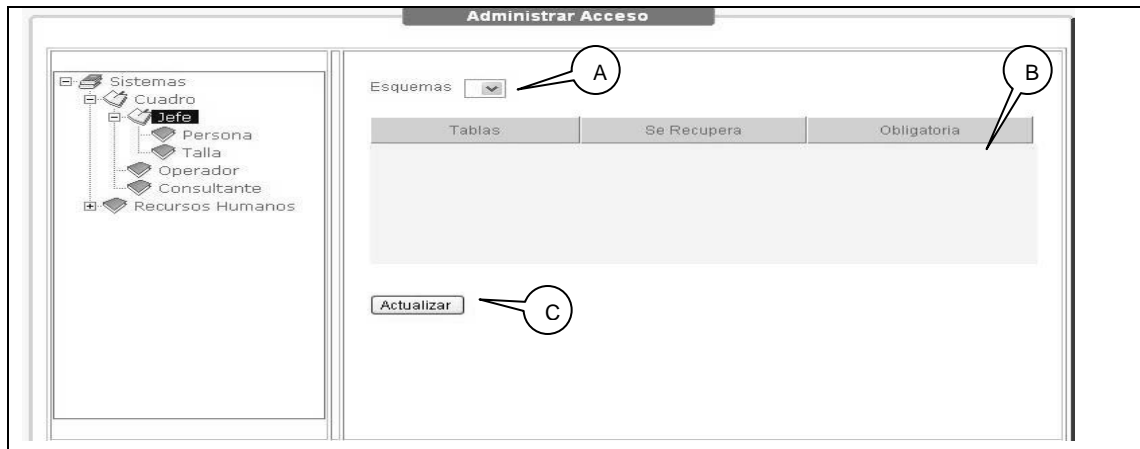
A: nombre: sistemas (Grid para mostrar la lista de nombres de los sistemas existentes).

Interfaz 12



A: nombre: roles (Grid para mostrar la lista de nombres de los roles existentes).

Interfaz 13



- A: nombre: tablas** (lista/menú para seleccionar el esquema al que se le desea administrar el acceso).
- B: nombre: sistemas** (Grid para mostrar la lista de nombres de los tablas que se recuperan o no, y cuales con las obligatorias).
- C: nombre: actualizar** (button para indicar al sistema que quiere actualizar la dependencia).

Curso normal de eventos para el caso de uso

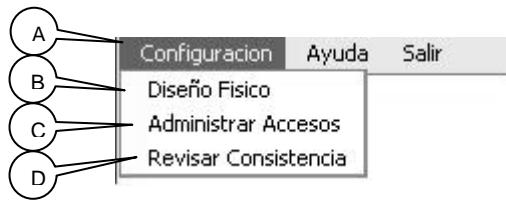
Acción del actor	Respuesta del sistema
1. Selecciona la opción Configurar del menú principal (2-A). 3. Selecciona la opción Administrar Accesos (2-C). 5. Selecciona el sistema.	2. Muestra menú con las opciones Diseño Físico, Administrar Accesos y Revisar Consistencia. 4. Muestra el grid (11-A) con todos los sistemas existentes. 6. Muestra el grid (12-A) con todos los roles existentes de dicho sistema.

<p>7. Selecciona el rol que desea configurar.</p> <p>9. El actor decide: Si desea definir las tablas por roles que se van a recuperar y las obligatorias ver sección 1.</p> <p>Si desea definir los campos por roles que se van a recuperar de una determinada tabla ver sección 2.</p>	<p>8. Muestra la interfaz 13.</p>
Sección 1	
Acción del actor	Respuesta del sistema
<p>9. Escoge el esquema (13-A).</p> <p>11. Marca las tablas de las cuales se va a recuperar información y las que van a ser obligatorias.</p> <p>12. Acciona el botón <Actualizar> (13-C)</p>	<p>10. Muestra el grid (13-B) con las tablas correspondiente al esquema seleccionado.</p> <p>13. Guarda las tablas definidas para recuperar información.</p>
Sección 2	

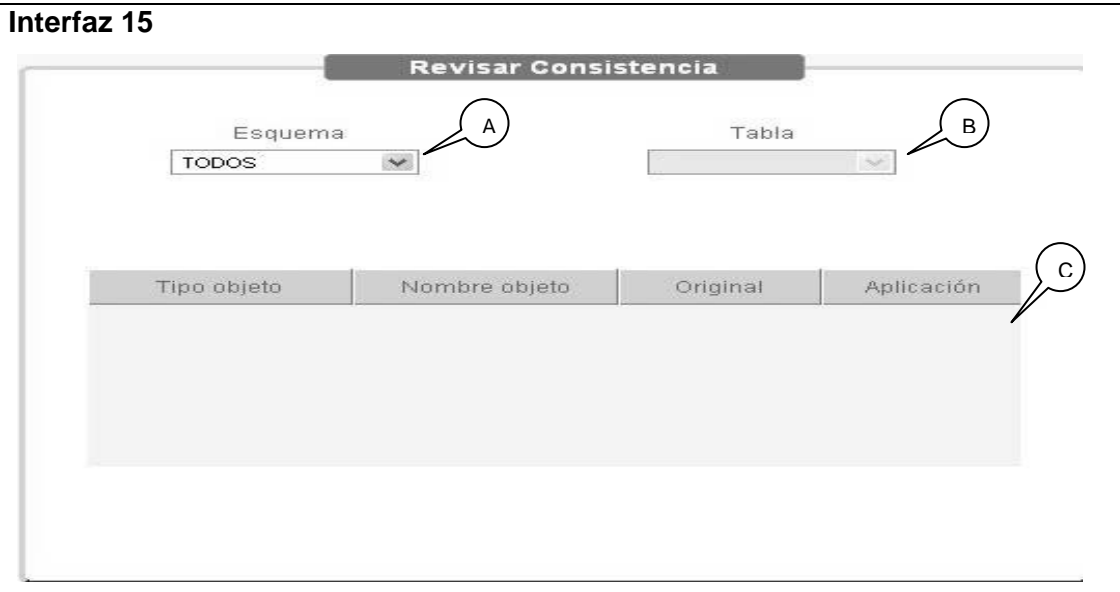
Interfaz 14	
<p>A: nombre: sistemas (Grid para mostrar la lista de nombres de los tablas que se recuperan o no, y cuales con las obligatorias).</p> <p>B: nombre: actualizar (button para indicar al sistema que quiere actualizar la dependencia).</p>	
Acción del actor	Respuesta del sistema
<p>9. Escoge la tabla que desea definirle los campos que va a tener acceso.</p> <p>11. Marca los campos de los cuales se va recuperar información.</p> <p>12. Acciona el botón <Actualizar> (14-B).</p>	<p>10. Muestra el grid (14-A) con los campos pertenecientes a la tabla seleccionada.</p> <p>13. Guarda los campos definidos para recuperar información.</p>
Cursos alternos	

Tabla 2.16 Gestionar Acceso.

2.9.14 Descripción del Caso de Uso Revisar consistencia.

Caso de uso:	Revisar consistencia.
Actores:	Configurador (inicia).
Propósito:	Verificar si han ocurrido cambios en la Base de Datos con respecto a los datos almacenados en el esquema de recuperaciones.
Resumen	El caso de uso se inicia cuando el configurador desea revisar consistencia, para esto verifica si han ocurrido cambios en la Base de Datos. El caso de uso termina cuando el configurador cierra el sistema o accede a otras opciones de la aplicación.
Precondiciones:	Debe haber sido autenticado como configurador. Debe existir la Base de Datos. Debe estar almacenado el diseño de la Base de Datos en el esquema de recuperaciones.
Poscondiciones:	Se ha comprobado si esta actualizada la Base de Datos en el esquema de recuperaciones.
Tipo:	Real y expandido.
Responsabilidades:	R8
Casos de uso relacionados:	
Interfaz 2	
 <p>El diagrama muestra un menú desplegable con un fondo gris. El menú está abierto y muestra cuatro opciones de texto: 'Configuración', 'Ayuda', 'Salir' y 'Revisar Consistencia'. A la izquierda del menú, hay cuatro círculos etiquetados con las letras A, B, C y D. Líneas conectan cada círculo con una de las opciones del menú: A apunta a 'Configuración', B a 'Diseño Físico', C a 'Administrar Accesos' y D a 'Revisar Consistencia'.</p>	
A: menú desplegable que permite seleccionar las diferentes configuraciones ha realizar.	

B: indica una de las configuraciones que se pueden realizar.
C: indica una de las configuraciones que se pueden realizar.
D: indica una de las configuraciones que se pueden realizar.



A: nombre: esquema (lista/menú para seleccionar el o los esquemas al que se le desea revisar la consistencia).
B: nombre: tabla (lista/menú para seleccionar la o las tablas a las que se le desean revisar su consistencia).
C: nombre: objeto (Grid para mostrar la lista de nombres de los objetos físicos y los tipos de ellos, que se encuentran en la BD original y en la de la aplicación).

Curso normal de eventos

Acción del actor	Respuesta del sistema
1. Selecciona la opción Configurar del menú principal (2-A). 3. Selecciona la opción Revisar Consistencia (2-D).	2. Muestra menú con las opciones Diseño Físico, Administrar Accesos y Revisar Consistencia.

<p>5. El actor decide: Si desea revisar consistencia de todos los esquemas ver sección 1.</p> <p>Si desea revisar consistencia de todas las tablas de un esquema ver sección 2.</p> <p>Si desea revisar consistencia de todos los campos de una tabla de un esquema determinado ver sección 3.</p>	<p>4. Muestra la interfaz 15.</p>
Sección 1	
Acción del actor	Respuesta del sistema
<p>5. Selecciona de la lista/menú (15-A) la opción (todos).</p>	<p>6. Muestra el grid (15-C) que contiene todos los esquemas con sus respectivos nombres y su estado de existencia presente en la Base de Datos y en el esquema de recuperaciones.</p>
Sección 2	
Acción del actor	Respuesta del sistema
<p>5. Selecciona de la</p>	

<p>lista/menú (15-A) la opción de un esquema determinado.</p> <p>7. Selecciona de la lista/menú (15-B) de la opción (todas).</p>	<p>6. Muestra la lista/menú (15-B).</p> <p>8. Muestra el grid (15-C) que contiene todas las tablas del esquema seleccionado; con sus respectivos nombres y su estado de existencia presente en la Base de Datos y en el esquema de recuperaciones.</p>
<p>Sección 3</p>	
<p>Acción del actor</p>	<p>Respuesta del sistema</p>
<p>5. Selecciona de la lista/menú (15-A) la opción de un esquema determinado.</p> <p>7. Selecciona de la lista/menú (15-B) la opción de una tabla determinada.</p>	<p>6. Muestra la lista/menú (15-C).</p> <p>8. Muestra el grid (15-C) que contiene todos los campos de la tabla seleccionada; con sus respectivos nombres y su estado de existencia presente en la Base de Datos y en el esquema de recuperaciones.</p>
<p>Cursos alternos</p>	

Tabla 2.17 Revisar consistencia.

2.10 Conclusiones

En este capítulo quedaron plasmadas las características que presenta el sistema, los problemas existentes en la entidad, y se hace una propuesta de solución a dichos problemas. Teniendo en cuenta lo dicho anteriormente se pudo apreciar que debido al bajo nivel de estructuración que presenta el negocio que se está estudiando, se llegó a la conclusión de la realización de un modelo de dominio ya que muestra los principales conceptos que se manejan en el sistema en desarrollo. Este trabajo ha expuesto cómo el modelado del dominio puede facilitar la identificación de los requisitos tanto funcionales como no funcionales del sistema. Además, de las descripciones de las funcionalidades del mismo.

A square graphic with a grey background. At the top, the word "Capítulo" is written in a bold, black, sans-serif font. Below it, the Roman numeral "III" is displayed in a large, bold, black, serif font.**CAPITULO III: ANALISIS Y DISEÑO DEL SISTEMA****3.1 Introducción**

El análisis y diseño de aplicaciones informáticas de gestión debe abordarse, con técnicas y metodologías adecuadas, acompañadas por una precisa gestión de proyectos y una eficaz gestión de la calidad. Así mismo, es importante poder contar con el soporte de entornos y herramientas adecuadas, que faciliten la tarea del profesional informático y de los usuarios a la hora de desarrollar sistemas de información.

El modelo de análisis y diseño del sistema es muy importante en el desarrollo de software dirigido por modelos ya que constituye la vista lógica de la arquitectura del software. Es en este capítulo donde se ajusta el resultado del análisis a las tecnologías y lenguajes que serán utilizados, además de darle forma a la arquitectura del sistema. Para ello los componentes de la aplicación se tratan como clases, y utilizando las extensiones del UML, se pueden presentar a través de diagramas de clases Web.

3.2 Análisis

En el análisis es donde se identifican las necesidades del cliente, se definen los conceptos del sistema para establecer su viabilidad, se asigna funciones al Hardware, Software y otros elementos del Sistema, y crea una definición del sistema que forme el fundamento de todo el trabajo de Ingeniería.

Definición del modelo de análisis. Modelo de clases de análisis**3.2.1 Diagramas de clases del análisis**

Un diagrama de clases es una colección de elementos de un modelo estático declarativo, tales como clases, interfaces, y sus relaciones, conectados como un grafo entre sí y con sus contenidos. Este diagrama de clases sirve para definir el sistema que se quiere modelar.

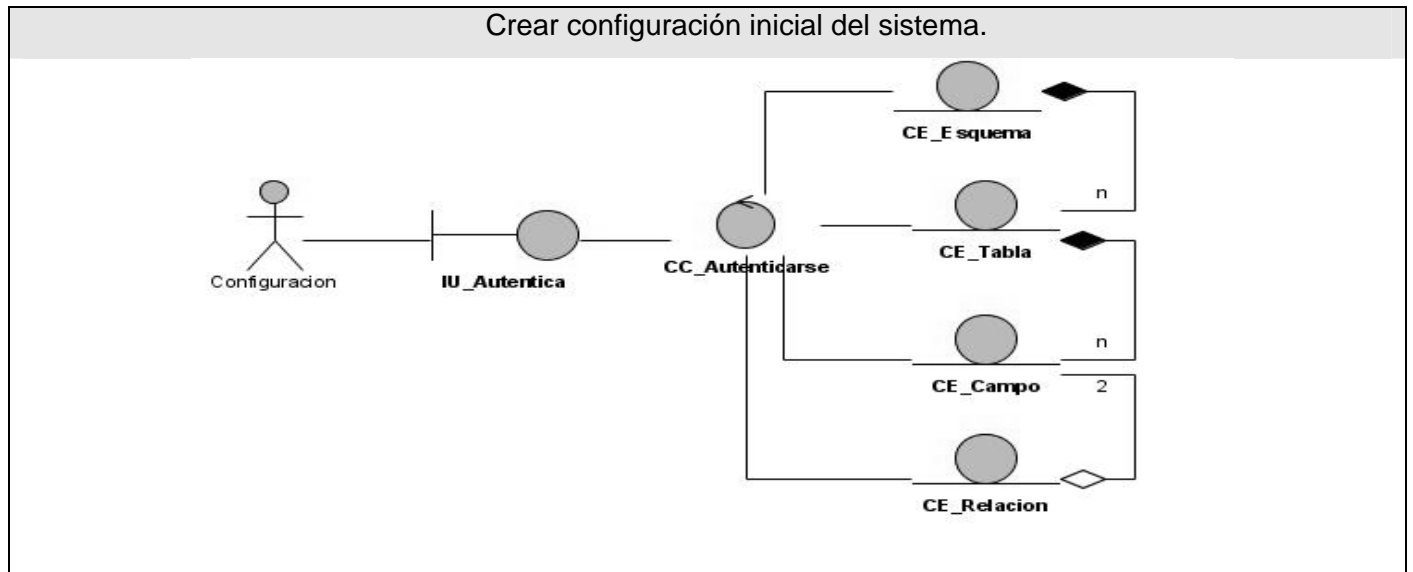


Fig. 3.1 Crear configuración inicial del sistema.

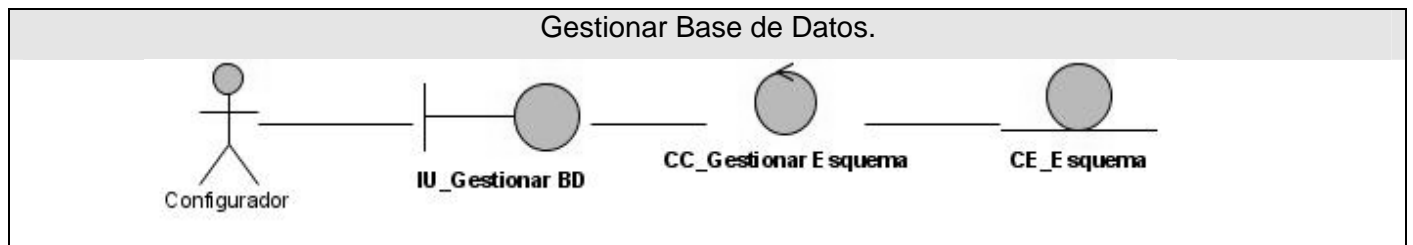


Fig. 3.2 Gestionar Base de Datos.

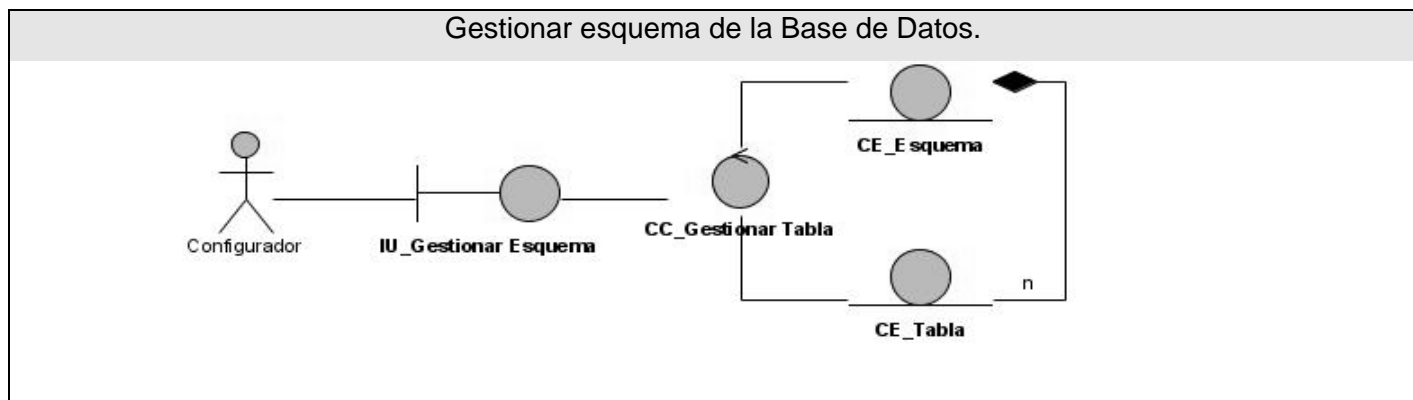


Fig. 3.3 Gestionar esquema de la Base de Datos.

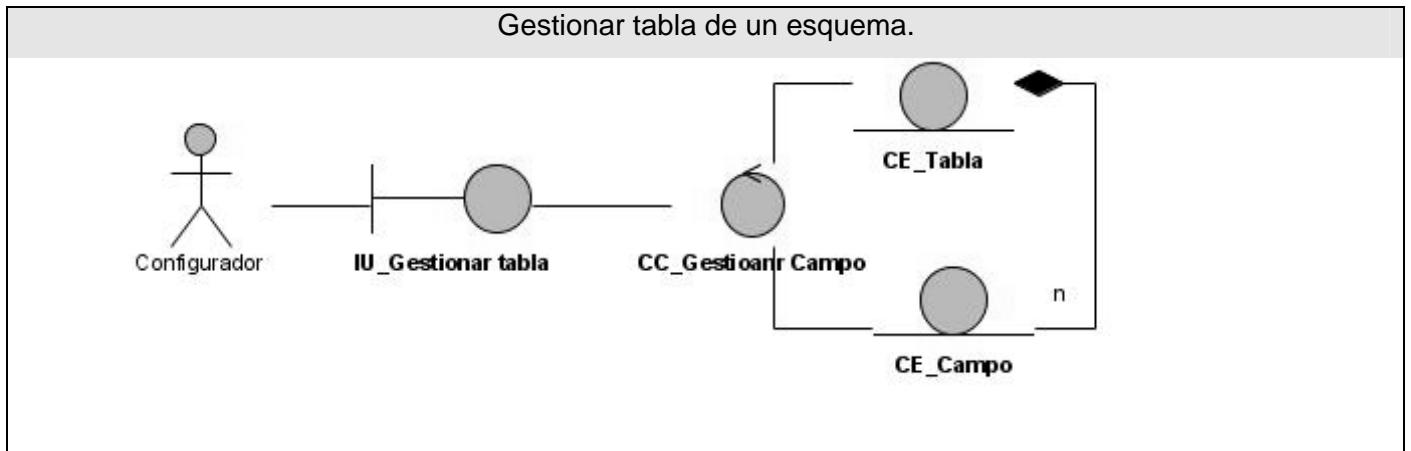


Fig. 3.4 Gestionar tabla de un esquema.

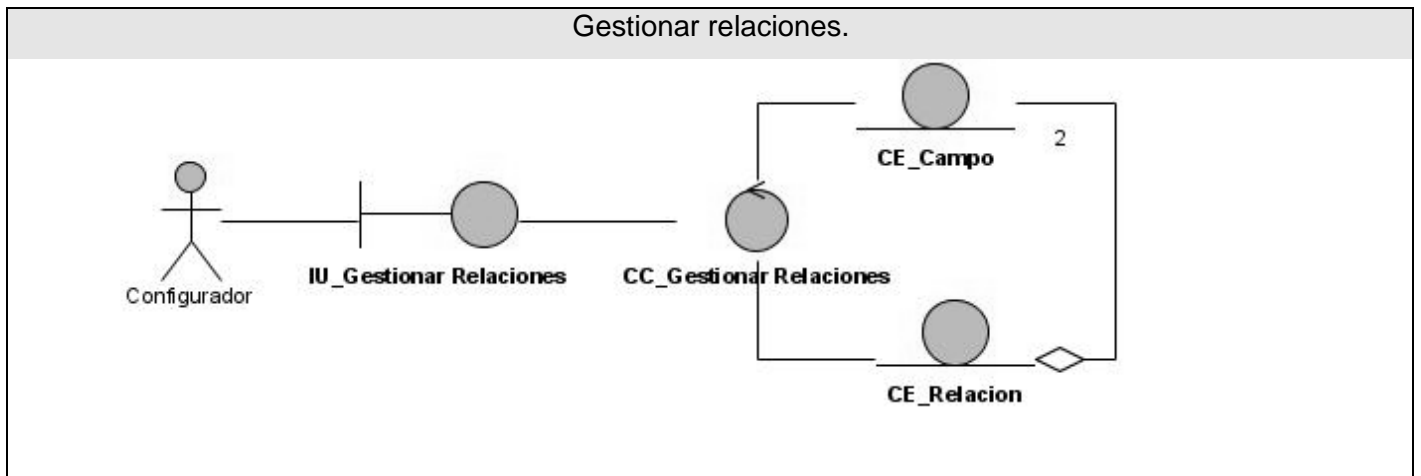


Fig. 3.5 Gestionar relaciones.

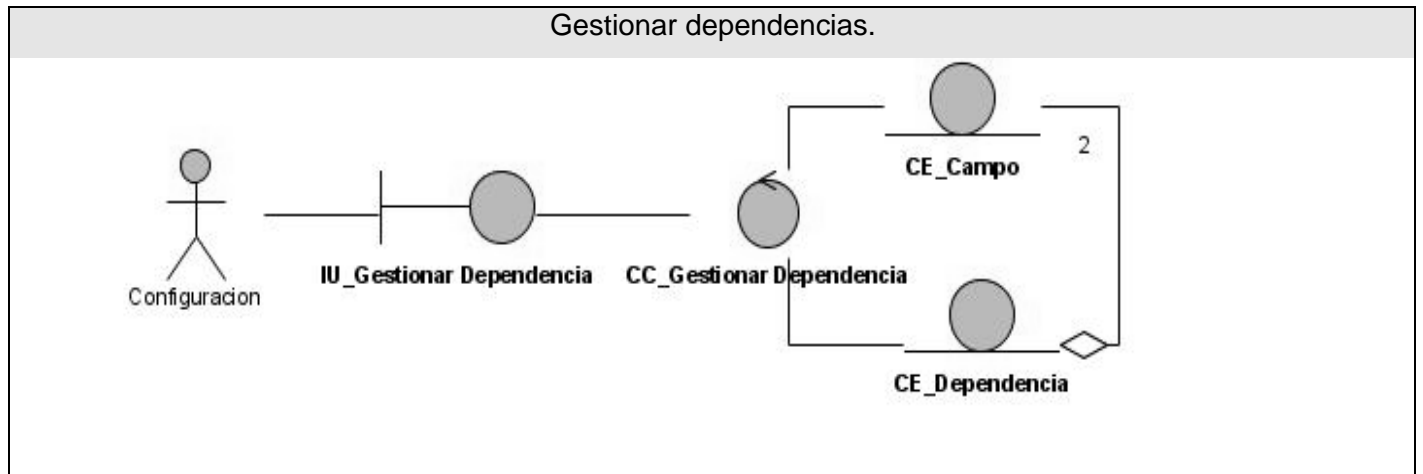


Fig. 3.6 Gestionar dependencias.

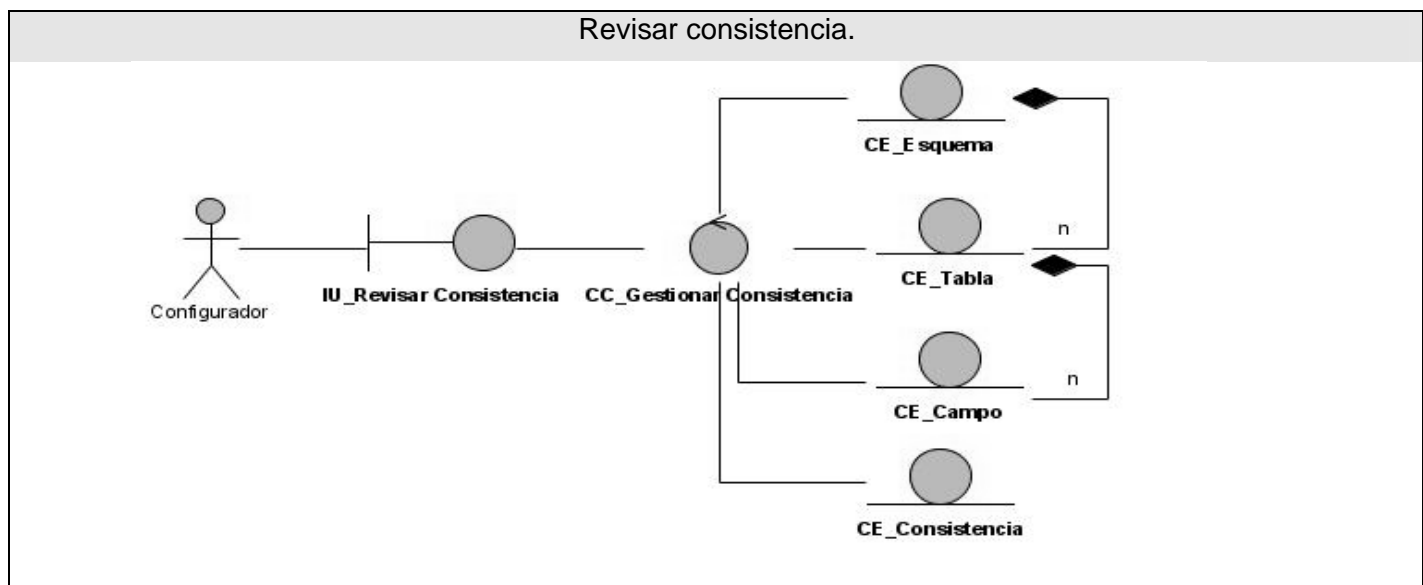


Fig.3.7 Revisar consistencia.

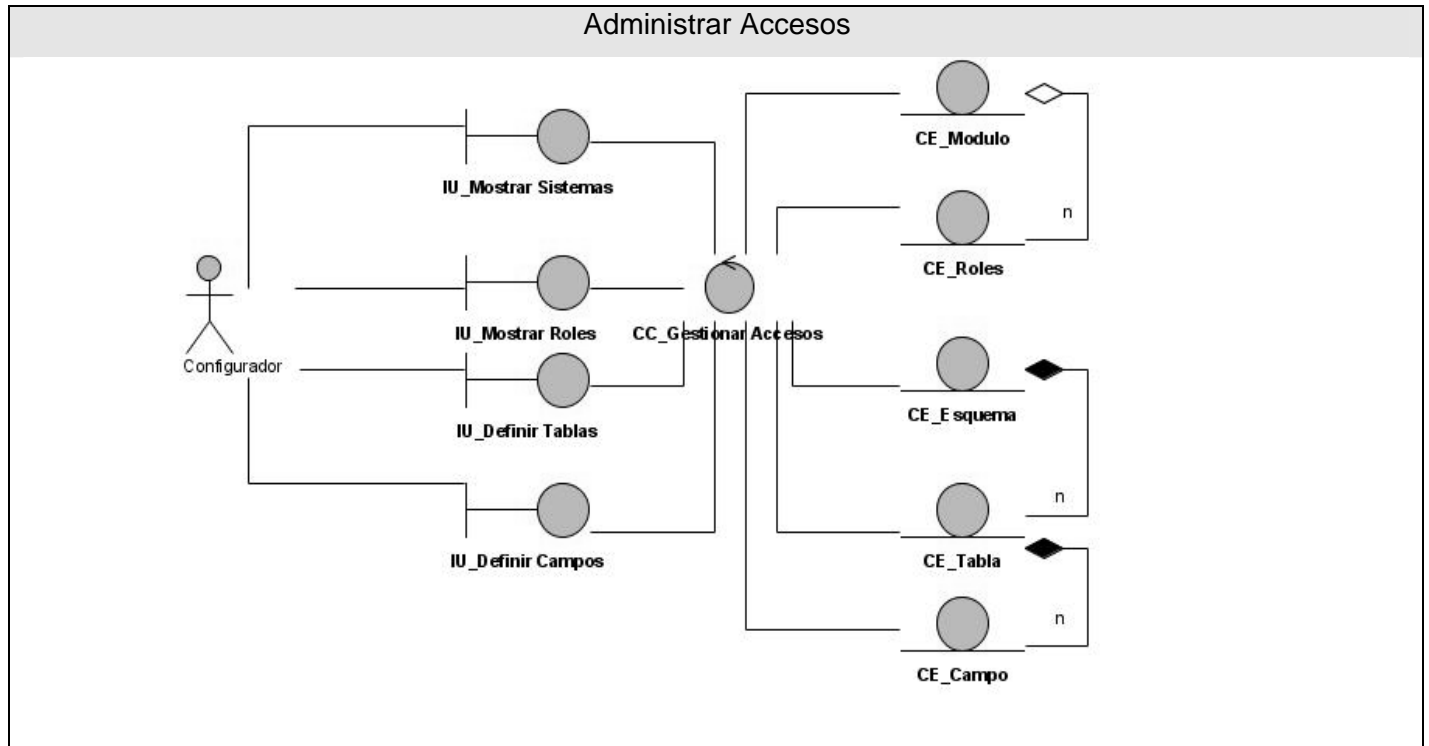


Fig. 3.8 Administrar Accesos.

3.3 Arquitectura del Sistema

Con el objetivo de comprender y organizar mejor el sistema se hace necesario definir una arquitectura que sea robusta y estable. La arquitectura que se empleará para el desarrollo de esta aplicación responde a la Arquitectura en 3 capas. Las capas son agrupaciones lógicas de los componentes de software que constituyen una aplicación o negocio. Y estas ayudan a diferenciar entre la clase de tareas desarrolladas por los componentes y hacen fácil el diseño de la reutilización. El uso de esta arquitectura en 3 capas permite mejorar la encapsulación, pues brinda más flexibilidad y escalabilidad. Además de que mejora la seguridad. Proporcionando una buena organización y estructuración entre los distintos niveles de abstracción, donde un cambio en uno de estos niveles no debe implicar grandes cambios en los restantes.

[11]

Arquitectura en 3 capas:

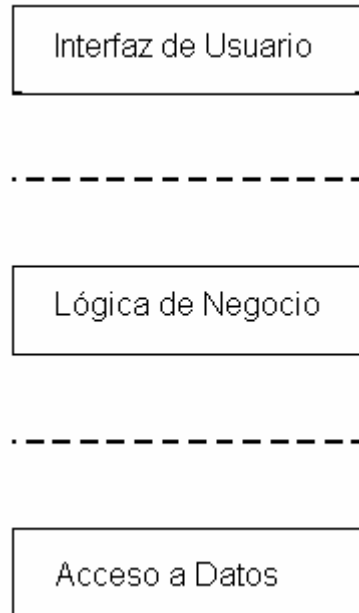


Fig. 3.9 Modelo de Arquitectura en 3 capas

Capa de Presentación: Esta formada por los Componentes de Interfaz de Usuario, y los componentes de proceso de Interfaz de Usuario. Los componentes de Interfaz de Usuario pueden ser vistos como la parte con la cual interactúa el usuario.

Capa de Lógica de Negocio: El comportamiento de la aplicación es definido por los componentes que modelan la lógica de negocio. Estos componentes reciben las acciones a realizar a través de la capa de presentación, y llevan a cabo las tareas necesarias utilizando la capa de acceso a datos para manipular la información del sistema.

Capa de Acceso a Datos: Contiene las clases que interactúan con la Base de Datos. Estas clases surgen como una necesidad de mantener la cohesión o clases altamente especializadas que ayuden a reducir la dependencia entre las clases y los datos. [12]

3.4 Diseño

En el Diseño de Sistemas se define el proceso de aplicar ciertas técnicas y principios con el propósito de definir un Sistema, con suficientes detalles como para permitir su interpretación y realización física.

3.4.1 Diagramas de Secuencia

Un Diagrama de Secuencias contribuye a la descripción de la dinámica del sistema en términos de la interacción entre sus objetos. Los diagramas de interacción son usados para identificar las clases y los métodos que ellos entregan.

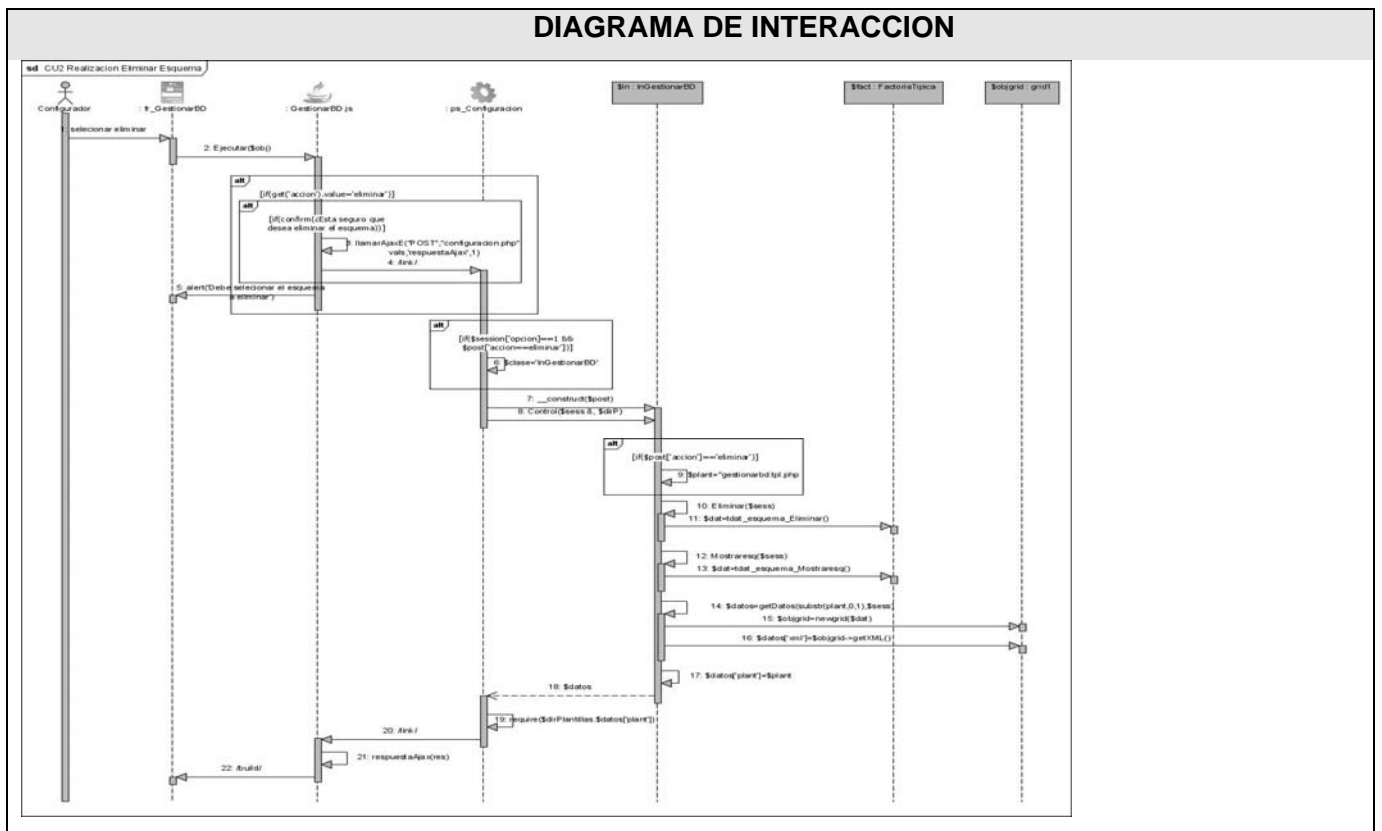


Fig. 3.10 CU2 Realización Eliminar Esquema.

Ver Diagramas de secuencias, Anexos 1

3.4.2 Diagrama de clases del diseño

Para obtener un nivel correcto de abstracción y detalle que permita obtener un resultado final, es mejor modelar los artefactos del sistema, es decir, modelar las páginas, los enlaces entre ellas, todo el código que irá creando las páginas, así como el contenido dinámico de estas una vez que estén en el navegador del cliente; estos son los artefactos que se necesitan modelar para que el desarrollador los implemente luego y obtener así el producto final.

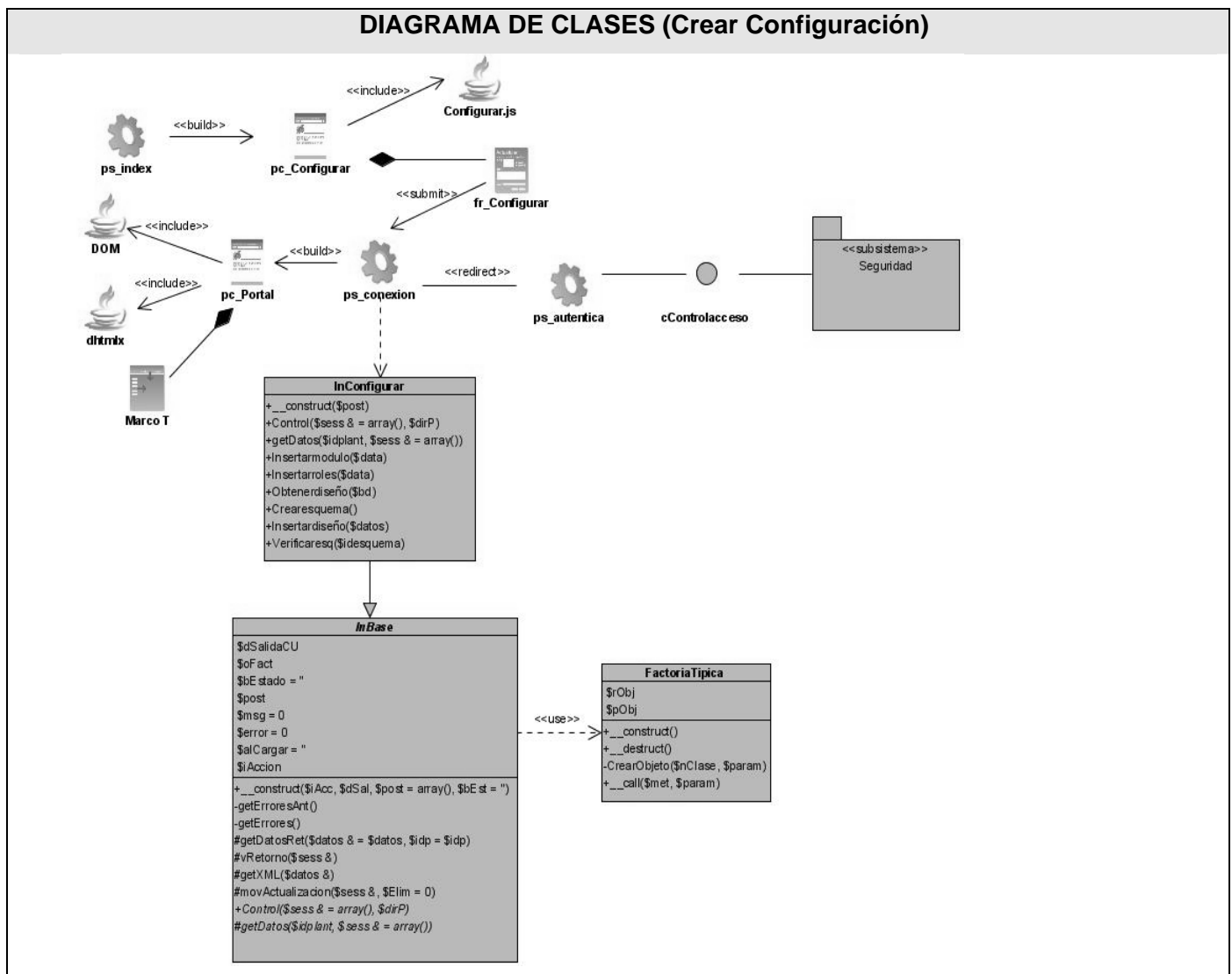


Fig. 3.11 Diagrama de clases (Crear Configuración)

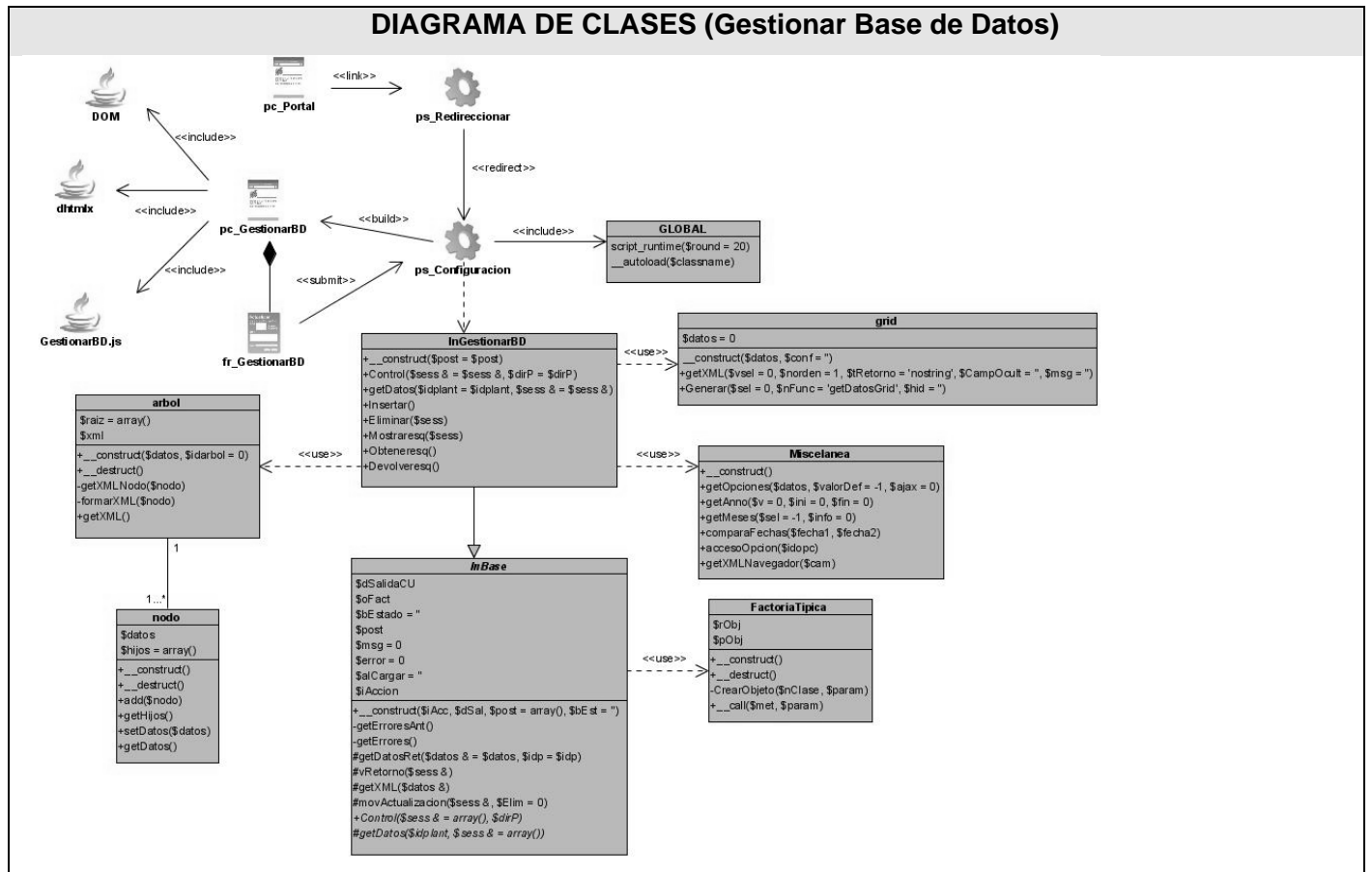


Fig. 3.12 Diagrama de clases (Gestionar Base de Datos)

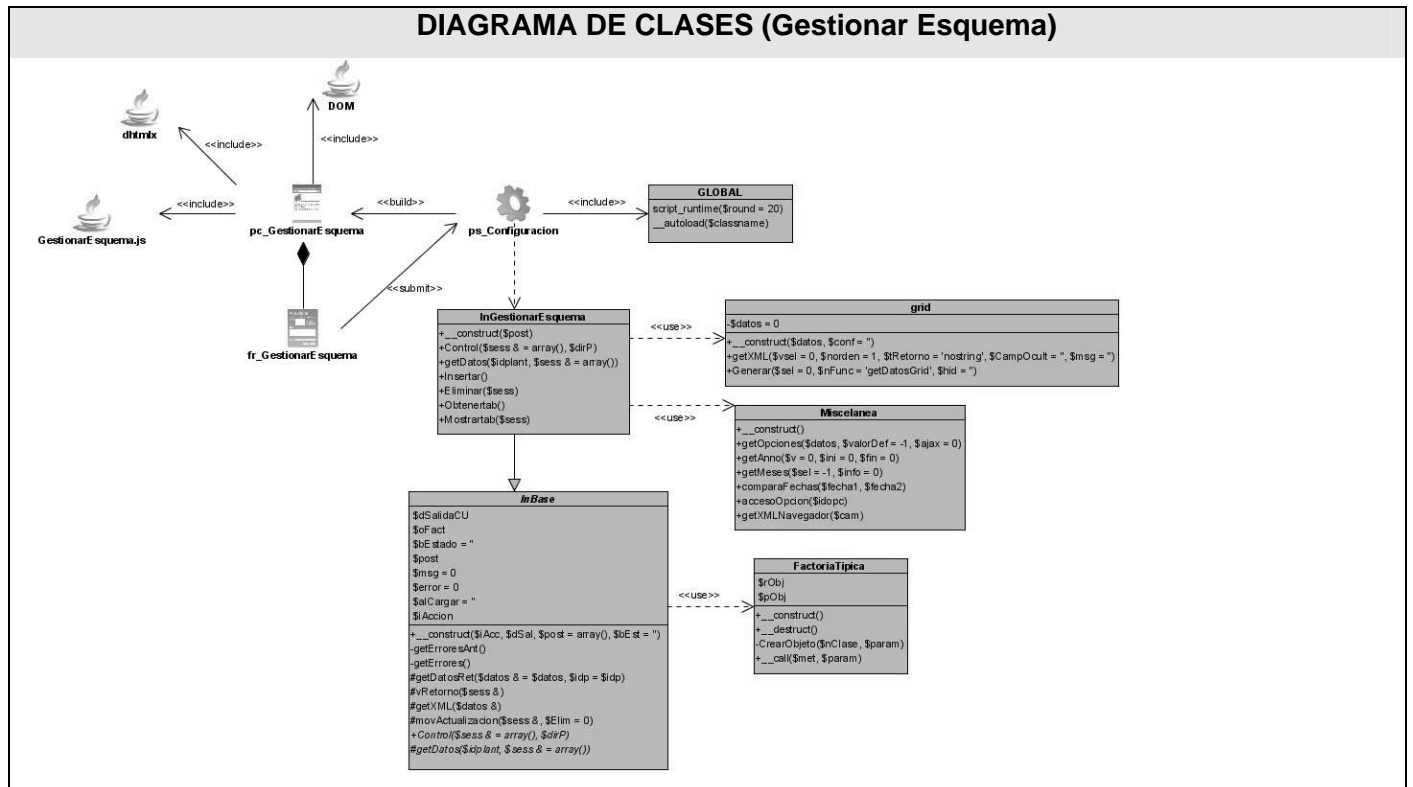


Fig. 3.13 Diagrama de clases (Gestionar Esquema)

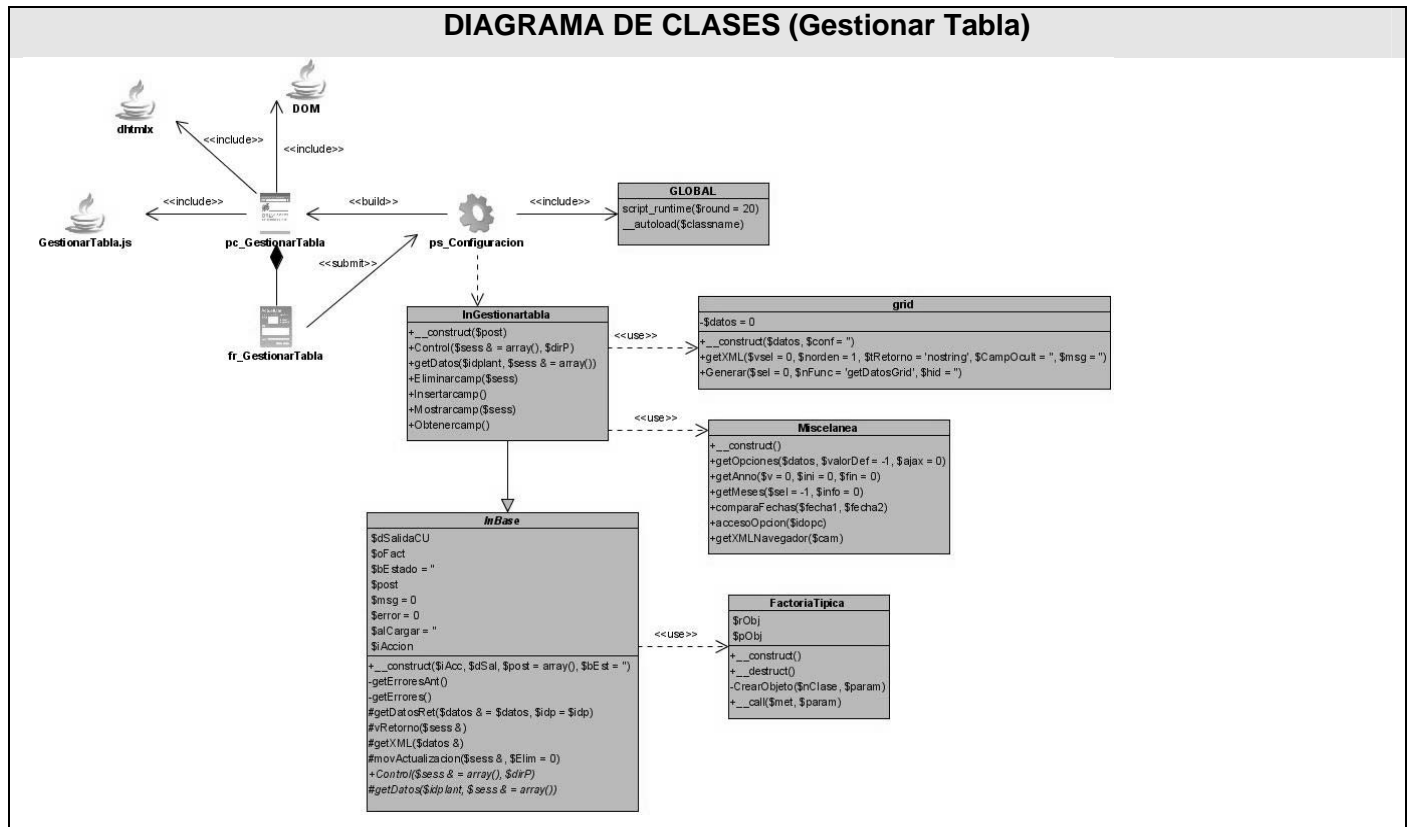


Fig. 3.14 Diagrama de clases (Gestionar Tabla)

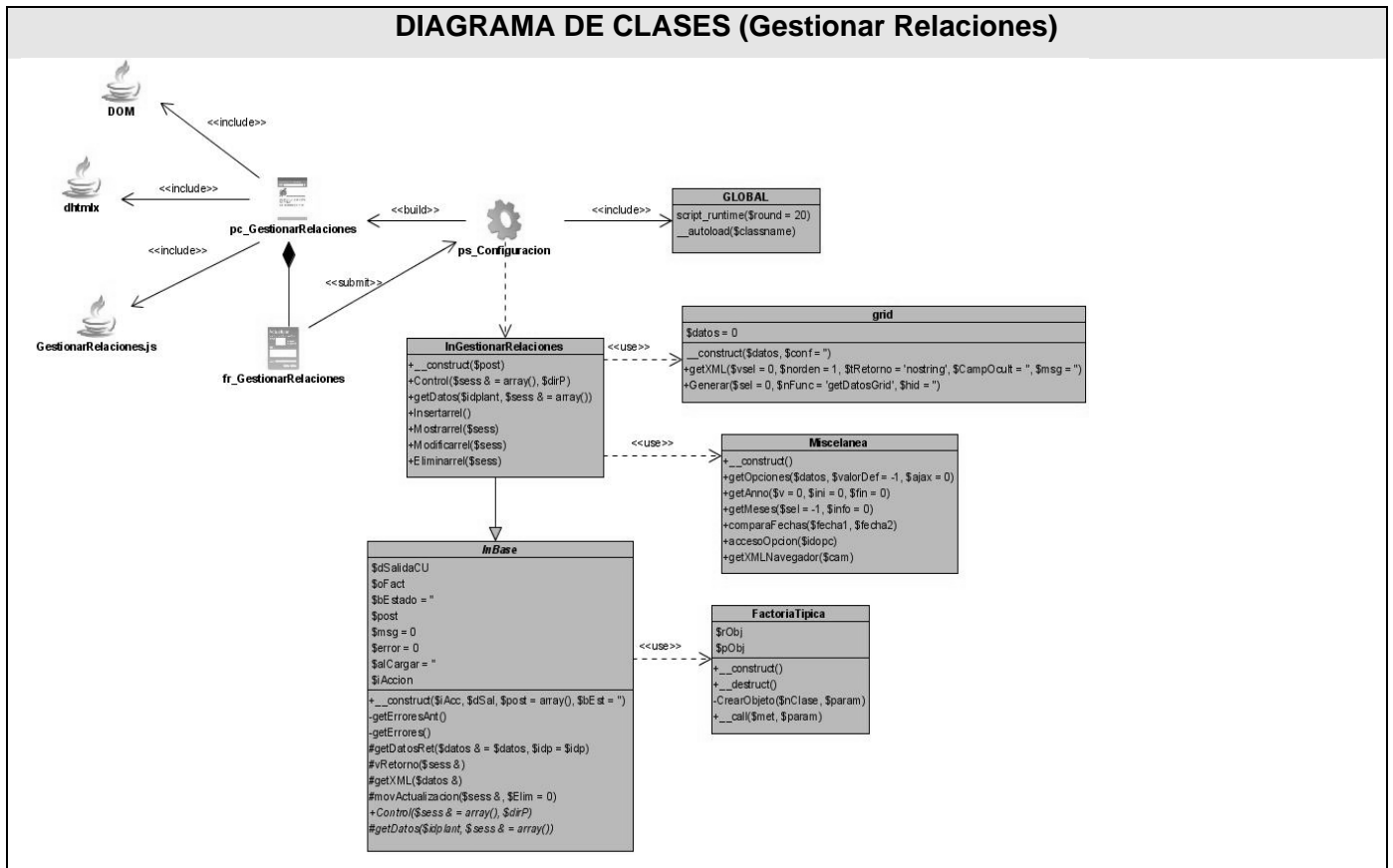


Fig. 3.15 Diagrama de clases (Gestionar Relaciones)

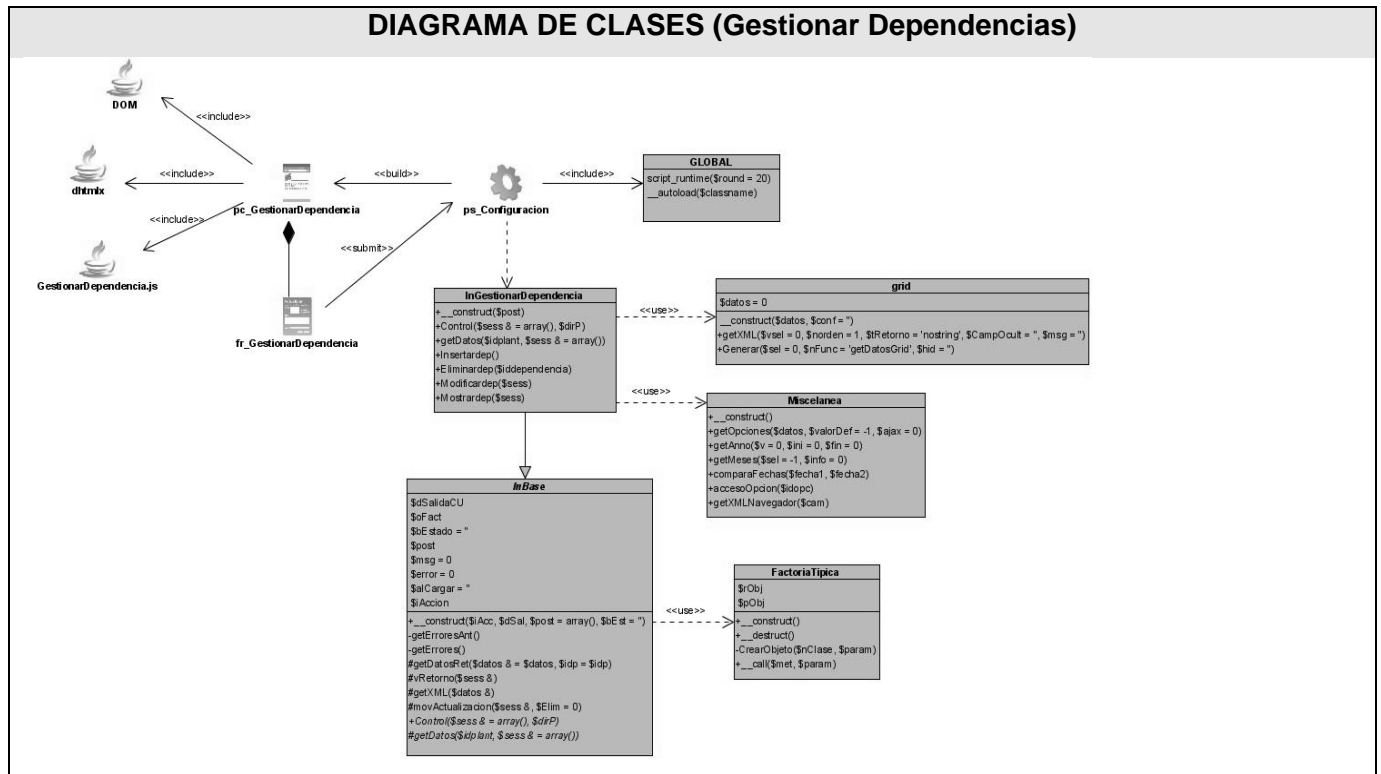


Fig. 3.16 Diagrama de clases (Gestionar Dependencias)

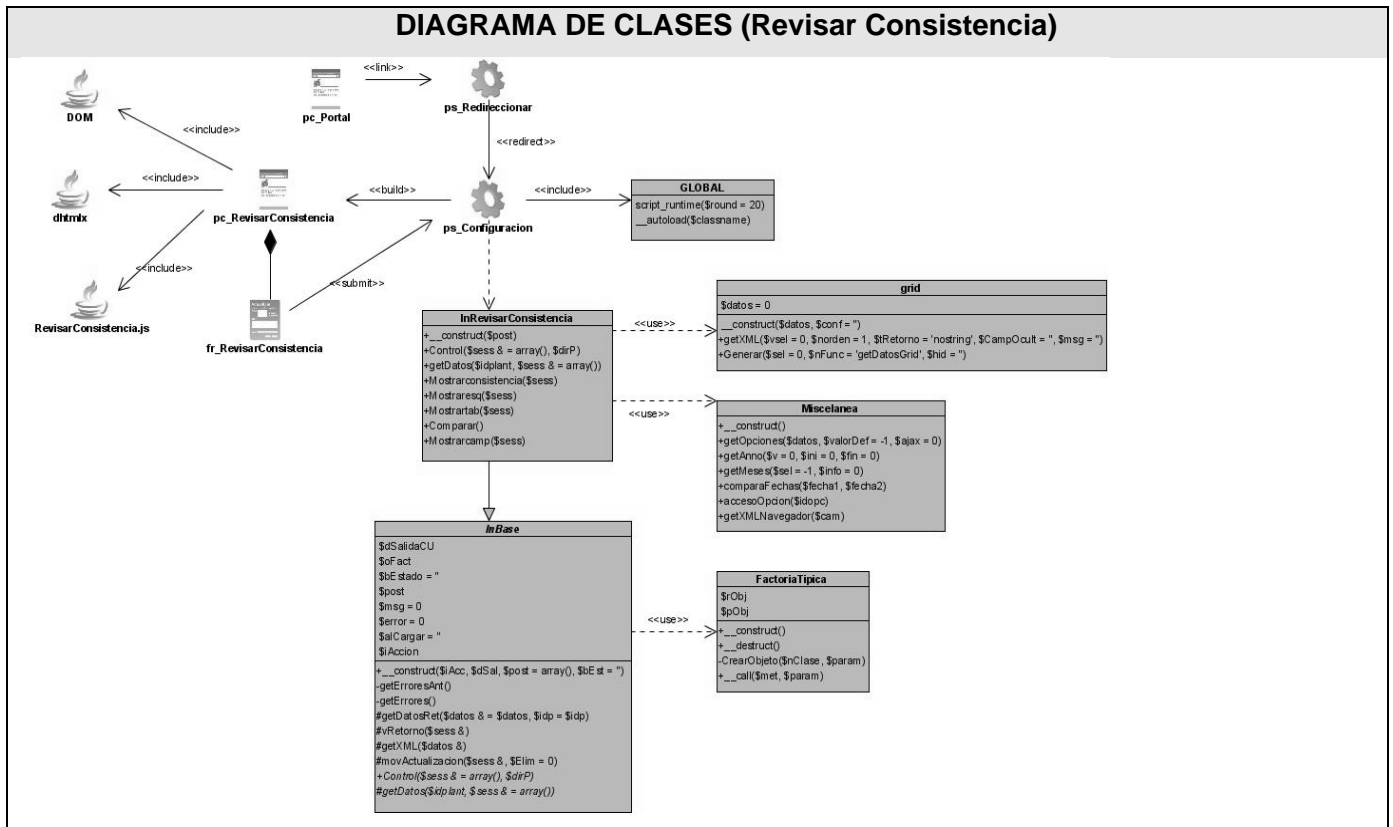


Fig. 3.18 Diagrama de clases (Revisar Consistencia)

3.4.3 Mecanismo de Diseño de Seguridad

En el diagrama que se representa a continuación se muestra la vista estática de la solución propuesta para manejar la seguridad. Mediante este mecanismo y haciendo uso del método RetArreglo se obtiene los módulos y roles de una determinada Base de Datos después de pasarle la cadena de conexión.

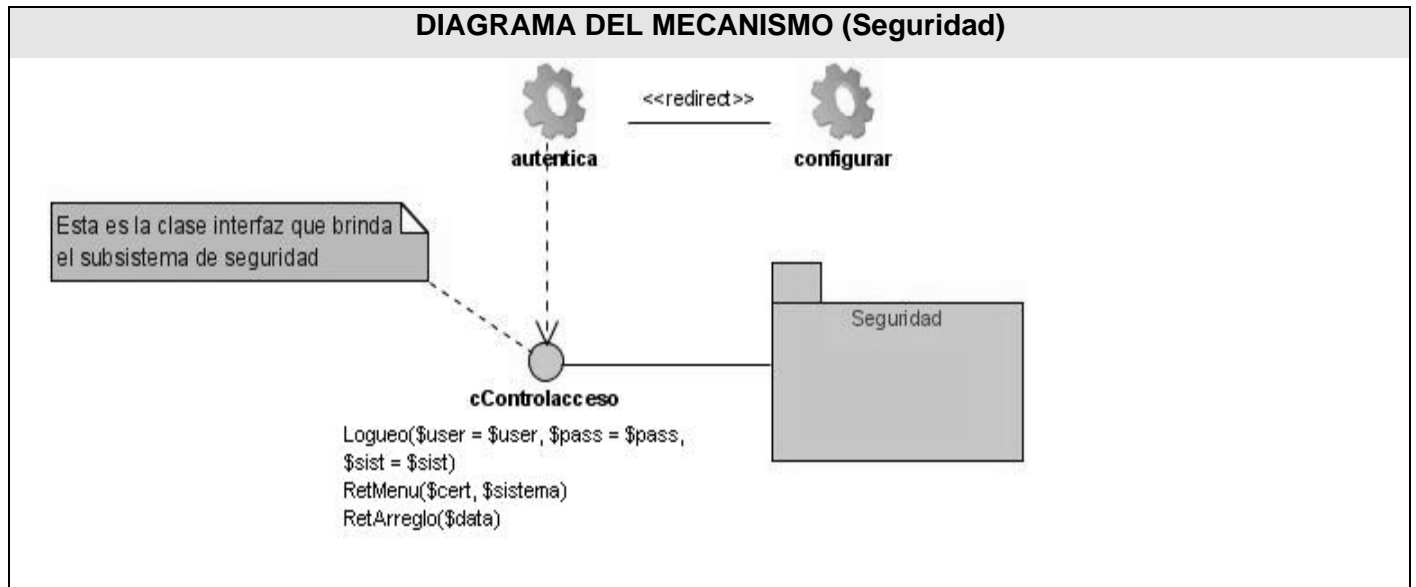


Fig. 3.19 Diagrama del mecanismo (Seguridad)

3.4.4 Mecanismo de Diseño de Acceso a Datos

En el diagrama que se representa a continuación se muestra la vista estática de la solución propuesta para manejar el acceso a datos del sistema. En él se encuentran una serie de clase las cuales cumplen con determinadas responsabilidades para asegurar un buen funcionamiento de este mecanismo.

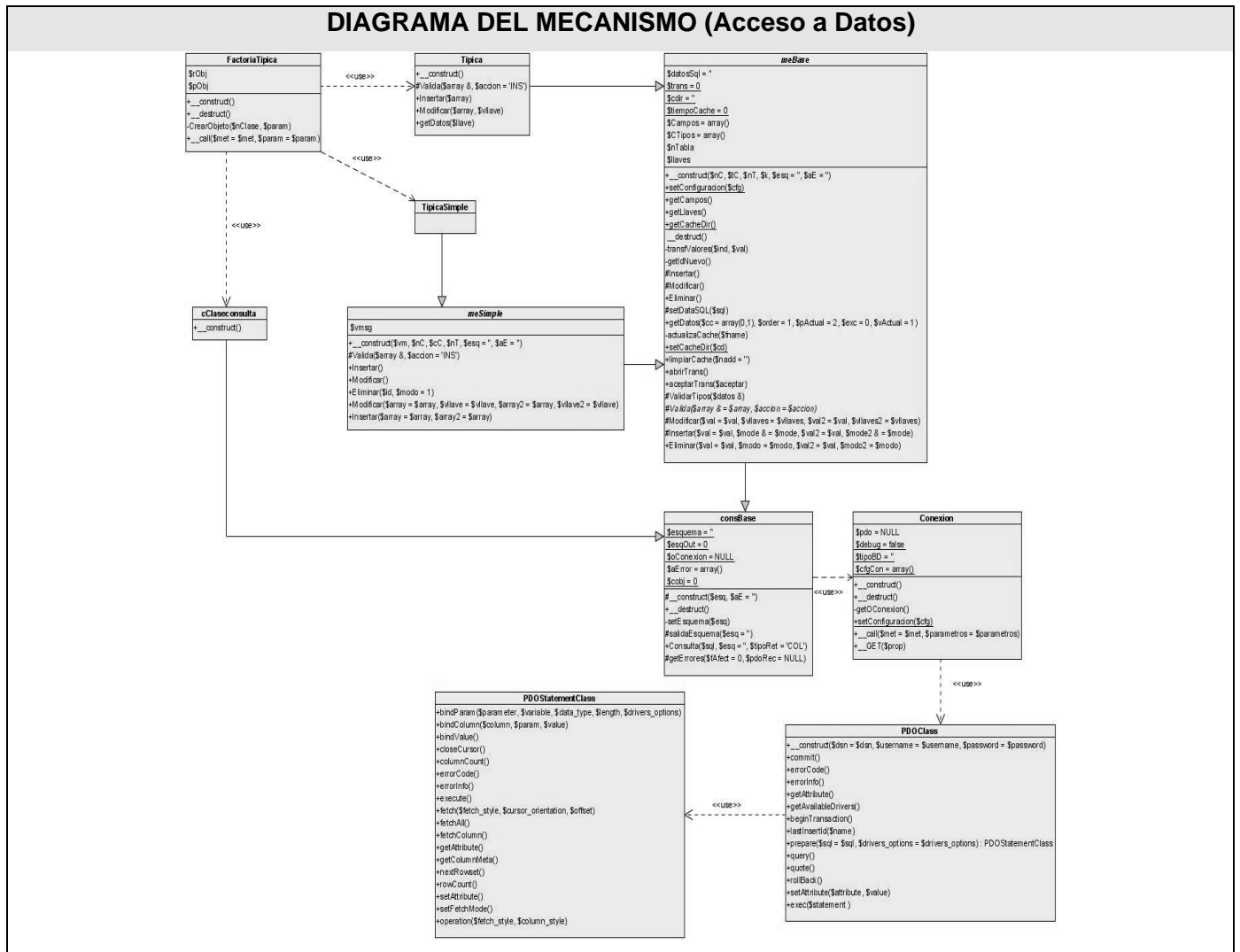


Fig. 3.20 Diagrama del mecanismo (Acceso a Datos)

Este mecanismo de diseño se ha implementado para controlar el acceso a los datos, pues se han identificado objetos que se involucran siempre en este tipo de ejecución y se efectúan, además, un conjunto de operaciones comunes a la mayoría de los Casos de Uso.

Factoría típica: Clase que implementa la interfaz del modelo de persistencia con el resto de los subsistemas. A través de esta clase se crean y se manipulan los objetos de las típicas simples, los

nomencladores y las demás típicas. Es una puerta entre la capa de Acceso a Datos y la capa de Lógica de Negocio.

Conexión: Esta clase es la encargada de establecer la conexión con el servidor de la base de datos a través de un objeto PDO de la librería de PHP.

consBase: Esta clase es la base en toda la jerarquía de Acceso a Datos y es empleada para aportar contenido dinámico a las plantillas. Encapsula el objeto conexión.

meBase: Clase abstracta que hereda de consBase, es la base para el resto de las que implementan funcionalidades para el trabajo con las entidades del sistema a implementar.

meSimple: Clase abstracta heredera de meBase, y la vez base para la implementación de las típicas que responderán a los nomencladores simples** del modelo de persistencia dado. Redefine las operaciones básicas con la funcionalidad de Validación dada.

** Entidades cuya estructura responde al siguiente patrón: idALGO, ALGO, actual. Donde ALGO representa la descripción del atributo principal de los nomencladores clásicos.

3.5 Diseño de la Base de Datos

En el diseño de la base de datos se modela el tratamiento de la información con carácter persistente dentro del sistema.

Varios son los métodos y alternativas para modelar la persistencia de los datos, incluyendo una gran variedad de herramientas de modelado. La propuesta actual es modelar la persistencia de los datos a partir de los diagramas de clase, con herramientas modernas que realizan una traducción del modelo de clases a un modelo de datos relacional.

Se propone construir dos modelos para la representación de los datos persistentes: el Modelo Lógico de Datos y el Modelo Físico de Datos. Estos dos modelos proporcionan una flexibilidad óptima para el soporte de la informatización entre el Modelo de Análisis y Diseño, y la Base de Datos Física.

3.5.1 Diagrama de Clases Persistentes

Este modelo debe ser usado siempre que se modelen datos persistentes, en sistemas que manipulan gran cantidad de información, contenida en medios de almacenamiento persistente.

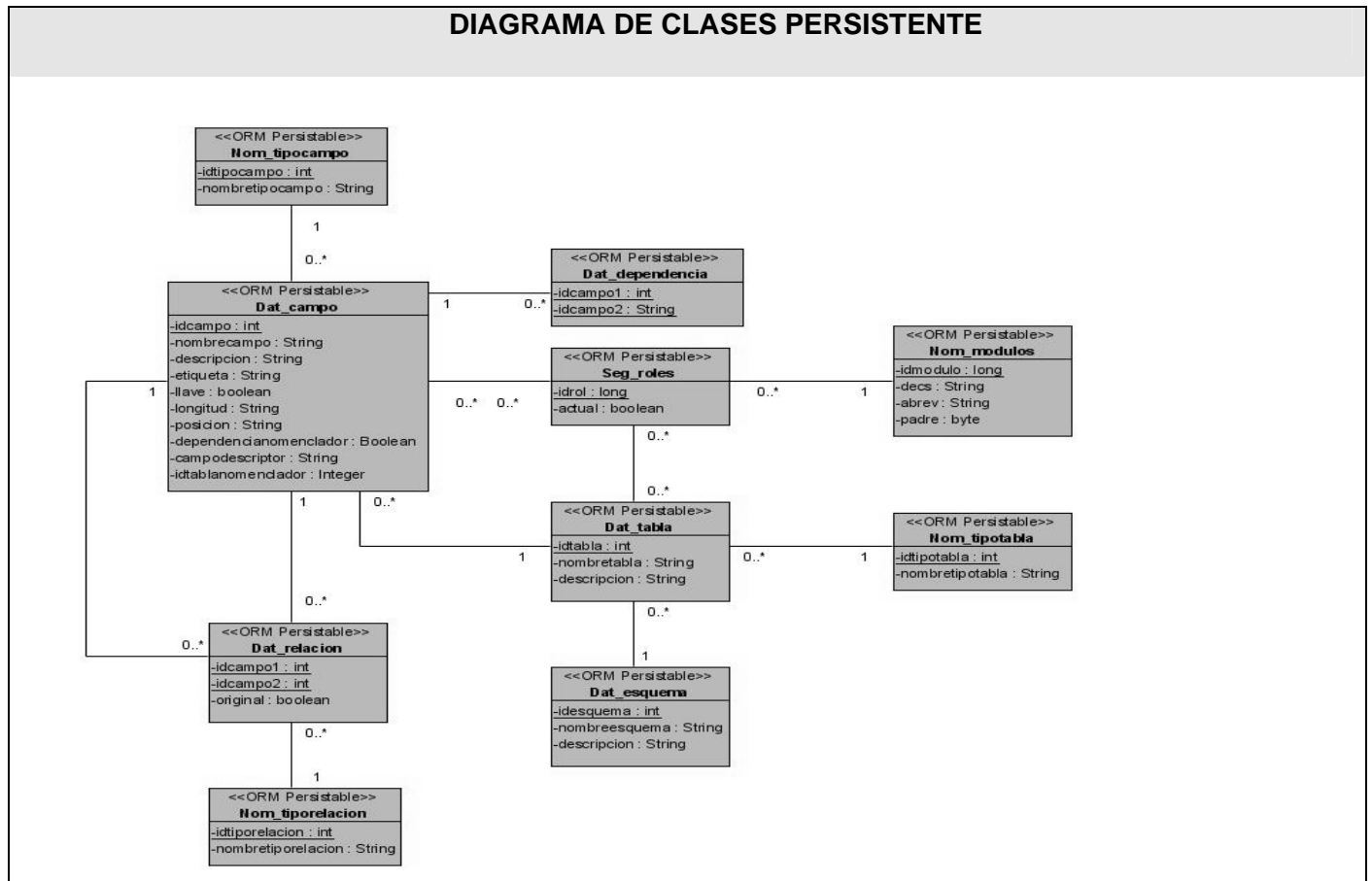


Fig. 3.21 Diagrama de clases persistente

3.5.2 Modelo de Datos

El modelo físico de los datos contiene un conjunto de tablas que conforman la base de datos. Este modelo constituye entonces la representación física del modelo de clases persistentes visto anteriormente. Las herramientas de modelado permiten la obtención de este modelo directamente a partir de las clases de los modelos anteriores.

Cuando se crea el Modelo Físico de la Base de Datos el diseñador debe seleccionar el sistema apropiado de Base de Datos, en nuestro caso el PostGreSQL. No obstante, para lograr que la aplicación sea fácil de migrar, debería construirse utilizando un estándar que sirva para múltiples bases de datos.

3.5.3 Diagrama Entidad Relación de la BD.

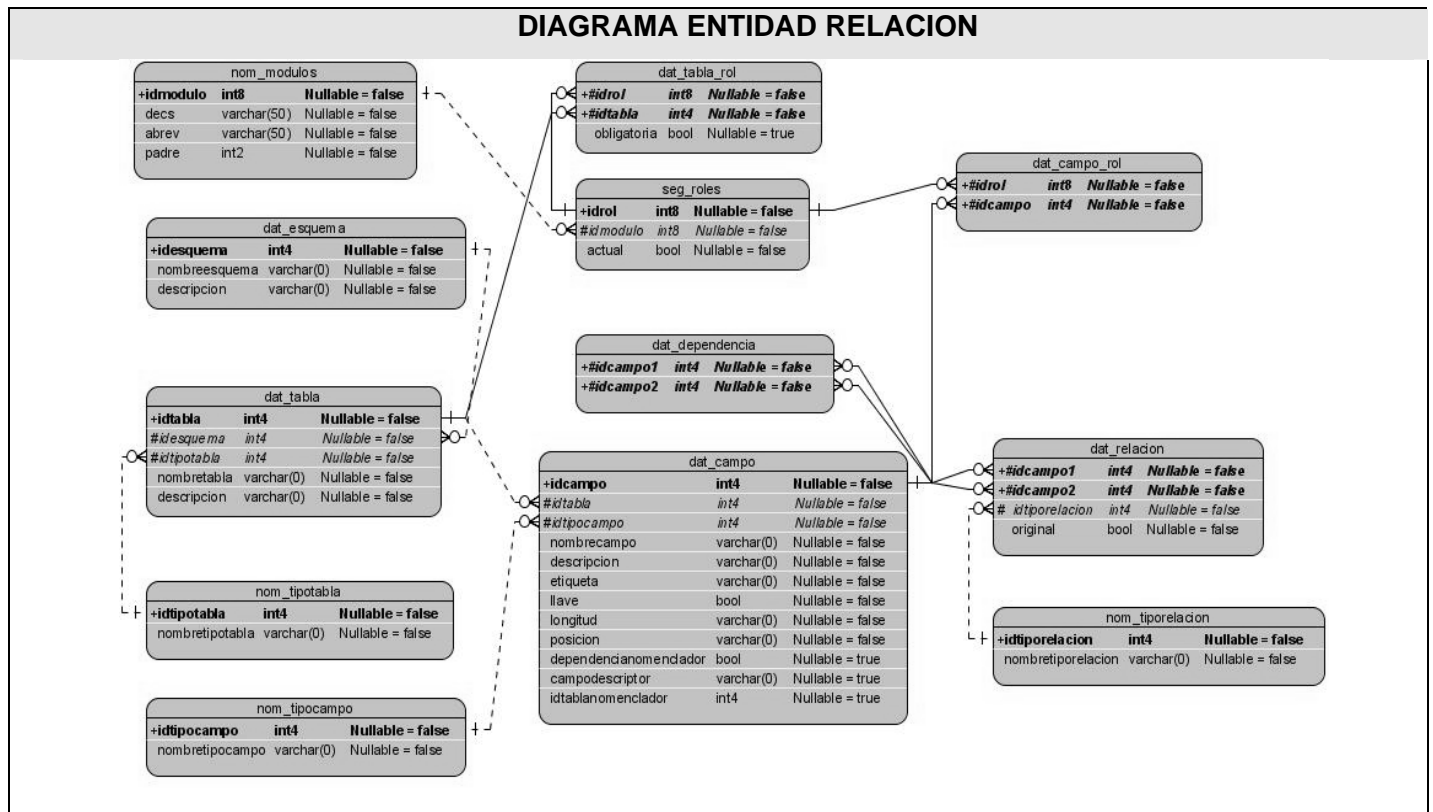


Fig. 3.22 Diagrama entidad relación

3.5.4 Descripción de las tablas.

Nombre: dat_esquema		
Descripción: Esta tabla almacena todos los esquemas de una Base de Datos con sus respectivas características.		
Atributo	Tipo	Descripción
idesquema	integer	El atributo identificador de los esquemas de la Base de Datos como llave primaria.
nombreesquema	varchar	Nombre del esquema que almacene.

descripcion	varchar	Breve información sobre el esquema que almacene.
-------------	---------	--

Tabla 3.1 dat_esquema

Nombre: dat_tabla		
Descripción: Esta tabla almacena todas las tablas de una Base de Datos con sus respectivas características.		
Atributo	Tipo	Descripción
idtabla	integer	El atributo identificador de las tablas de la Base de Datos como llave primaria.
idesquema	integer	El atributo identificador como llave foránea que representa el esquema al que pertenece la tabla almacenada.
idtipotabla	integer	El atributo identificador como llave foránea que representa la relación entre la tabla y el tipo de tabla.
nombretabla	varchar	Nombre de la tabla que almacene.
descripcion	varchar	Breve información sobre la tabla que almacene.

Tabla 3.2 dat_tabla

Nombre: dat_relacion		
Descripción: Esta tabla almacena todas las relaciones de una Base de Datos, además de las definidas por el usuario.		
Atributo	Tipo	Descripción
idcampo1	integer	El atributo identificador del primer campo que compone una relación como llave primaria.
idcampo2	integer	El atributo identificador del segundo campo que compone una relación como llave primaria.
idtiporelacion	integer	El atributo identificador como llave foránea que

		representa la correspondencia entre la relación y el tipo de relación.
original	boolean	Si la relación fue guardada de la Base de Datos almacenada o fue creada por el usuario.

Tabla 3.3 dat_relacion

Nombre: dat_campo		
Descripción: Esta tabla almacena todos los campos de una Base de Datos con sus respectivas características.		
Atributo	Tipo	Descripción
idcampo	integer	El atributo identificador de los campos de la Base de Datos como llave primaria.
idtabla	integer	El atributo identificador como llave foránea que representa la tabla a la que pertenece el campo almacenado.
idtipocampo	integer	El atributo identificador como llave foránea que representa la relación entre el campo y el tipo de campo.
idtablanomenclador	integer	Nombre de la tabla nomenclador de la cual depende.
nombrecampo	varchar	Nombre del campo que almacene.
descripción	varchar	Breve información sobre el campo que almacene.
etiqueta	varchar	Almacena el nombre que tomará el atributo campo.
longitud	varchar	La cantidad de caracteres que puede admitir el atributo campo.
posición	varchar	Es el lugar que va a tener ese atributo campo en la tabla.
campodescriptor	varchar	El nombre del campo descriptor de la tabla nomenclador que se utiliza para decodificar el

		idcampo.
llave	boolean	Si el atributo campo es o no el identificador de una tabla.
dependencianomenc lador	boolean	Si depende o no de un nomenclador para ser decodificado.

Tabla 3.3 dat_campo

Nombre: dat_dependencia		
Descripción: Esta tabla almacena todas las dependencias definidas por el usuario para facilitar la decodificación de la información.		
Atributo	Tipo	Descripción
idcampo1	integer	El atributo identificador del primer campo que compone la dependencia como llave primaria.
idcampo2	integer	El atributo identificador del segundo campo que compone la dependencia como llave primaria.

Tabla 3.4 dat_dependencia

Nombre: seg_roles		
Descripción: Esta tabla almacena todos los roles que pueden acceder a las aplicaciones existentes.		
Atributo	Tipo	Descripción
idrol	bigint	El atributo identificador de los roles que existen como llave primaria.
idmodulo	integer	El atributo identificador como llave foránea que representa el módulo al que pertenece el rol almacenado.
actual	bit(1)	Representa el estado en que se encuentra el rol.

Tabla 3.5 seg_roles

Nombre: nom_modulos		
Descripción: Esta tabla almacena todos los módulos que existen en las aplicaciones del proyecto.		
Atributo	Tipo	Descripción
idmodulo	bigint	El atributo identificador de los módulos que existen como llave primaria.
decs	varchar(50)	La descripción del módulo.
abrev	varchar(10)	La abreviatura del nombre del módulo.
padre	smallint	Identificador del módulo superior.

Tabla 3.6 nom_modulos

Nombre: nom_tipotabla		
Descripción: Esta tabla almacena los tipos de tablas que existen.		
Atributo	Tipo	Descripción
idtipotabla	integer	El atributo identificador del tipo de tabla como llave primaria.
nombretipotabla	varchar	Nombre del tipo de tabla que puede ser.

Tabla 3.7 nom_tipotabla

Nombre: nom_tipocampo		
Descripción: Esta tabla almacena los tipos de campos que existen.		
Atributo	Tipo	Descripción
idtipocampo	integer	El atributo identificador del tipo de campo como llave primaria.
nombretipocampo	varchar	Nombre del tipo de campo que puede ser.

Tabla 3.8 nom_tipocampo

Nombre: nom_tiporelacion		
Descripción: Esta tabla almacena los tipos de relaciones que existen.		
Atributo	Tipo	Descripción
idtiporelacion	integer	El atributo identificador del tipo de relación como llave primaria.
nombretiporelacion	varchar	Nombre del tipo de relación que puede ser.

Tabla 3.9 nom_tiporelacion

Nombre: dat_campo_rol		
Descripción: Esta tabla contiene los identificadores de las tablas <code>seg_rol</code> y <code>dat_campo</code> , ya que entre estas existe una relación de muchos a muchos.		
Atributo	Tipo	Descripción
idrol	integer	El atributo identificador de los roles como llave primaria.
idcampo	integer	El atributo identificador de los campos como llave primaria.

Tabla 3.10 dat_campo_rol

Nombre: dat_tabla_rol		
Descripción: Esta tabla contiene los identificadores de las tablas <code>seg_rol</code> y <code>dat_tabla</code> , ya que entre estas existe una relación de muchos a muchos.		
Atributo	Tipo	Descripción
idrol	integer	El atributo identificador de los roles como llave primaria.
idtabla	integer	El atributo identificador de las tablas como llave primaria.
obligatoria	boolean	Indica si la tabla es de consulta obligada para ese módulo.

Tabla 3.11 dat_tabla_rol

3.6 Conclusiones

En este capítulo se abarcó lo perteneciente a las vistas estáticas y de implementación correspondientes a la notación UML. De igual manera fueron señalados los contenidos que sobre los principios de diseño debían añadirse a los requisitos funcionales mencionados en el capítulo anterior, para de esta forma completar el análisis realizado.

**Capítulo
IV****CAPITULO IV: IMPLEMENTACION Y PRUEBA****4.1 Introducción**

En el presente capitulo se comienza a desarrollar la fase de construcción, donde el flujo de trabajo mas importante es la implementación. La base para el inicio de esta fase es el resultado que se haya obtenido en el diseño en el capitulo anterior, partiendo de esto es que se implementa el sistema a través de diagramas de componentes y la correspondencia de las clases con los mismos; y mediante el modelo de despliegue que describe los nodos físicos en los que funcionará la aplicación.

4.2 Diagramas de Componentes

Presenta elementos tangibles. Se utiliza, para describir la estructura física del código de la aplicación en términos de sus componentes (código fuente, binario o ejecutable) y sus dependencias.

A continuación se presenta el diagrama de componentes de estructura donde el subsistema de Configuración tiene tres paquetes, cuya distribución se verá referenciada en el diagrama de componentes general donde se podrá apreciar cada una de las partes que tienen estos paquetes en la arquitectura.

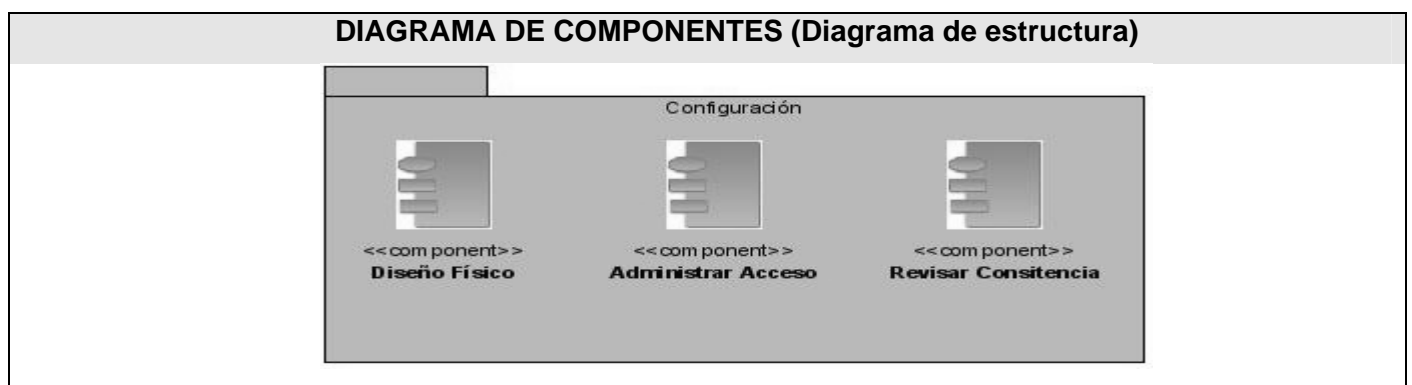


Fig. 4.1 Diagrama de componentes (Estructura)

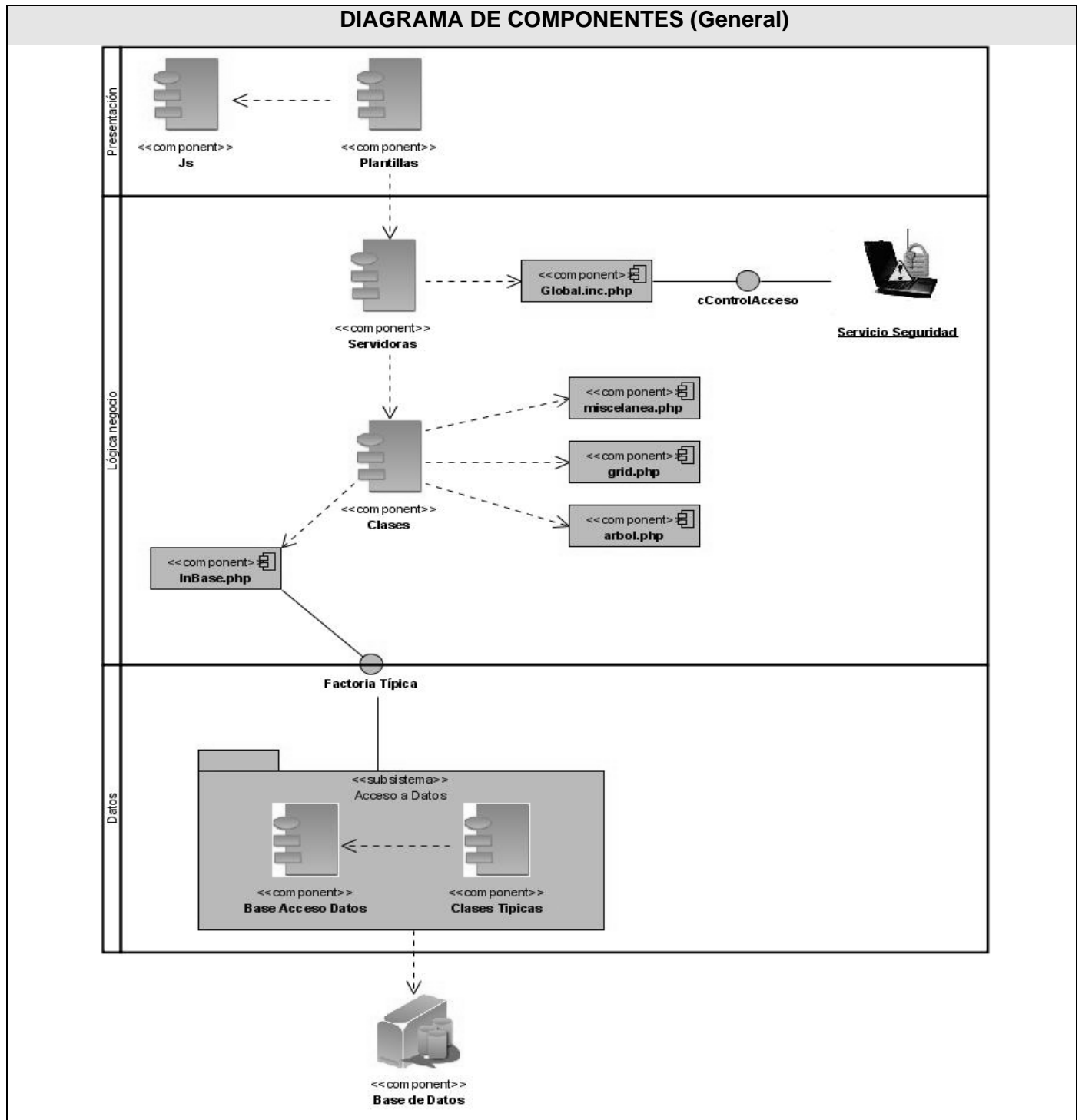


Fig. 4.2 Diagrama de componentes (General)

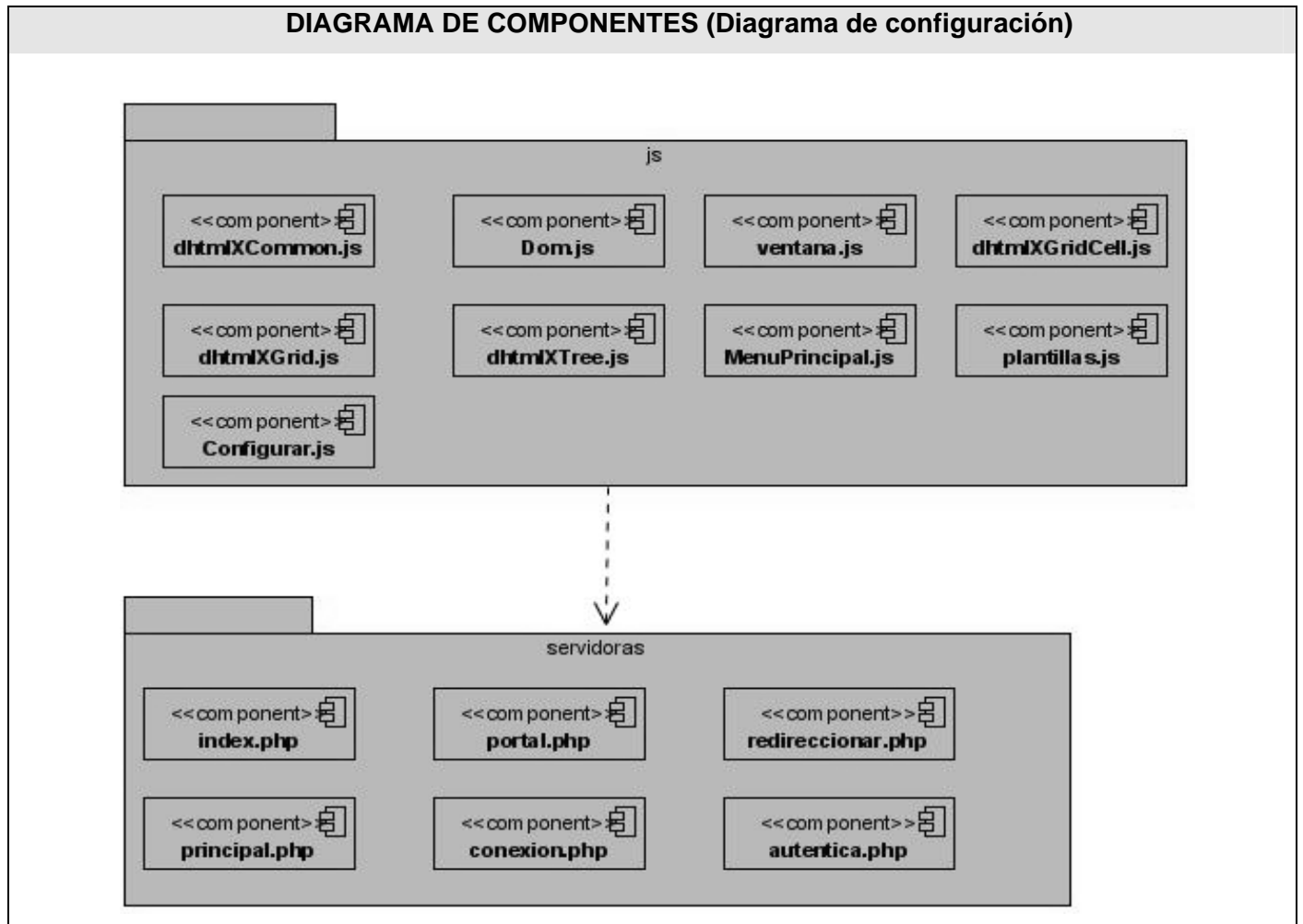


Fig. 4.3 Diagrama de componentes (Configuración)

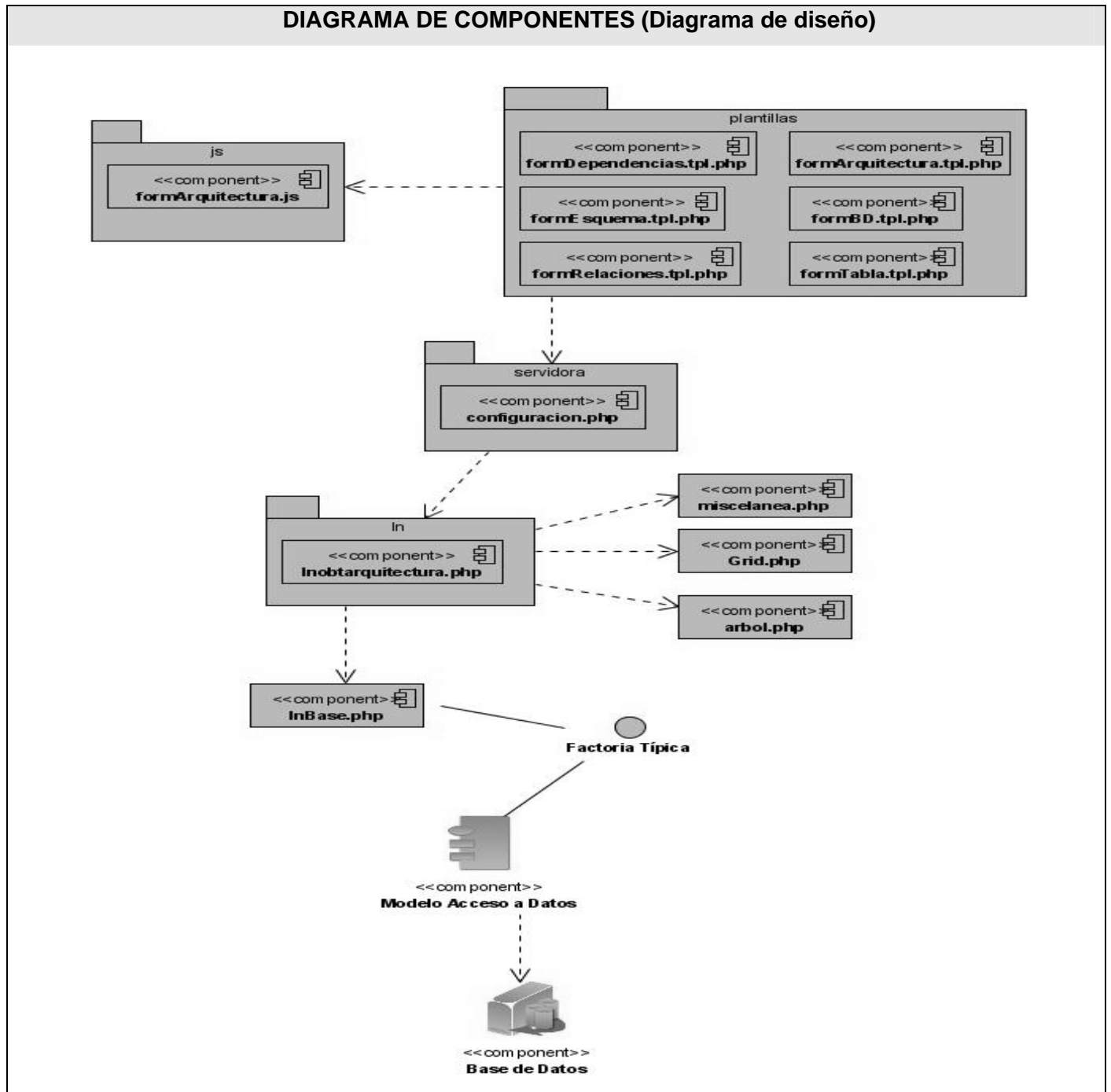


Fig. 4.4 Diagrama de componentes (Diseño)

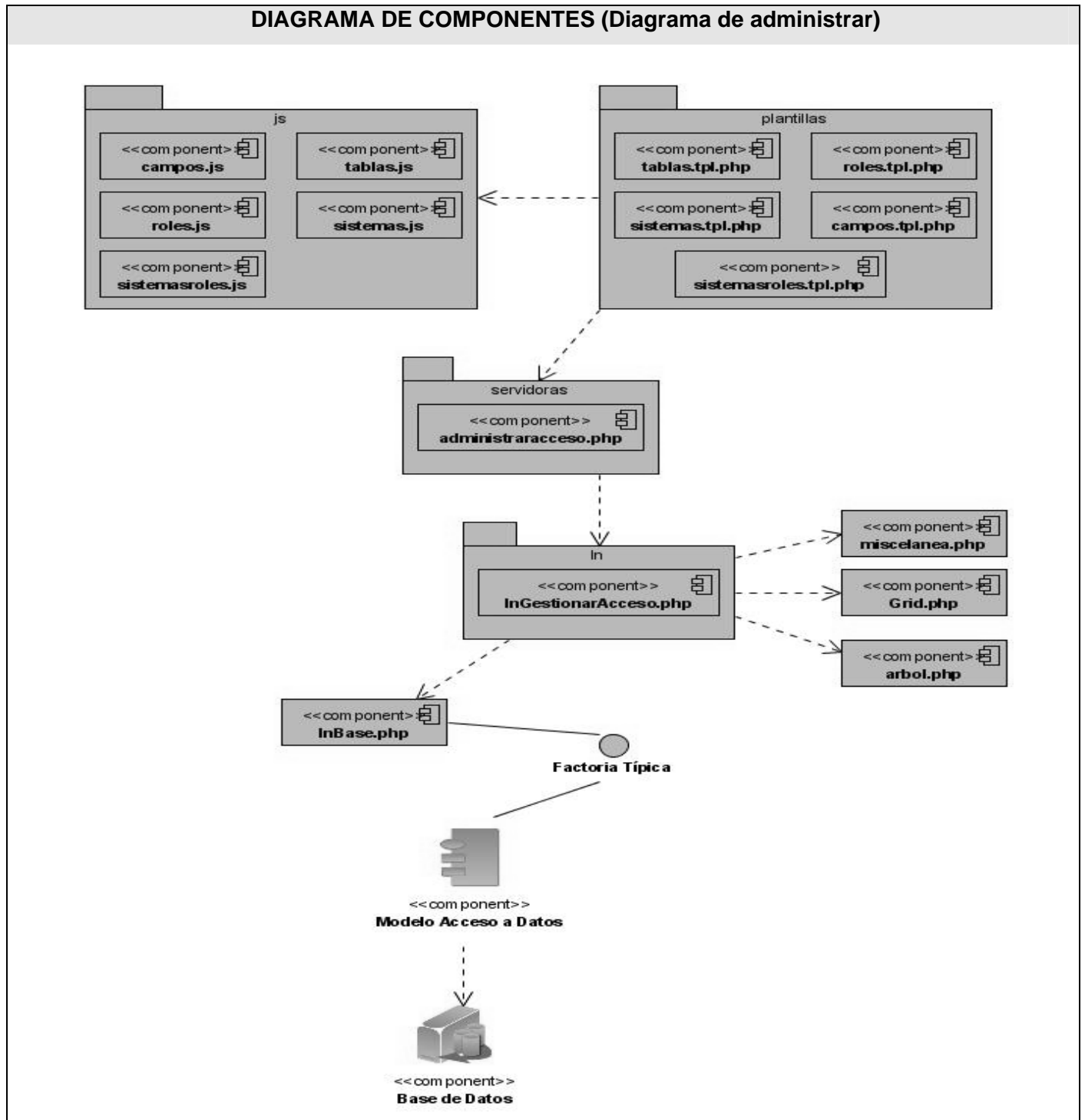


Fig. 4.5 Diagrama de componentes (Administrar)

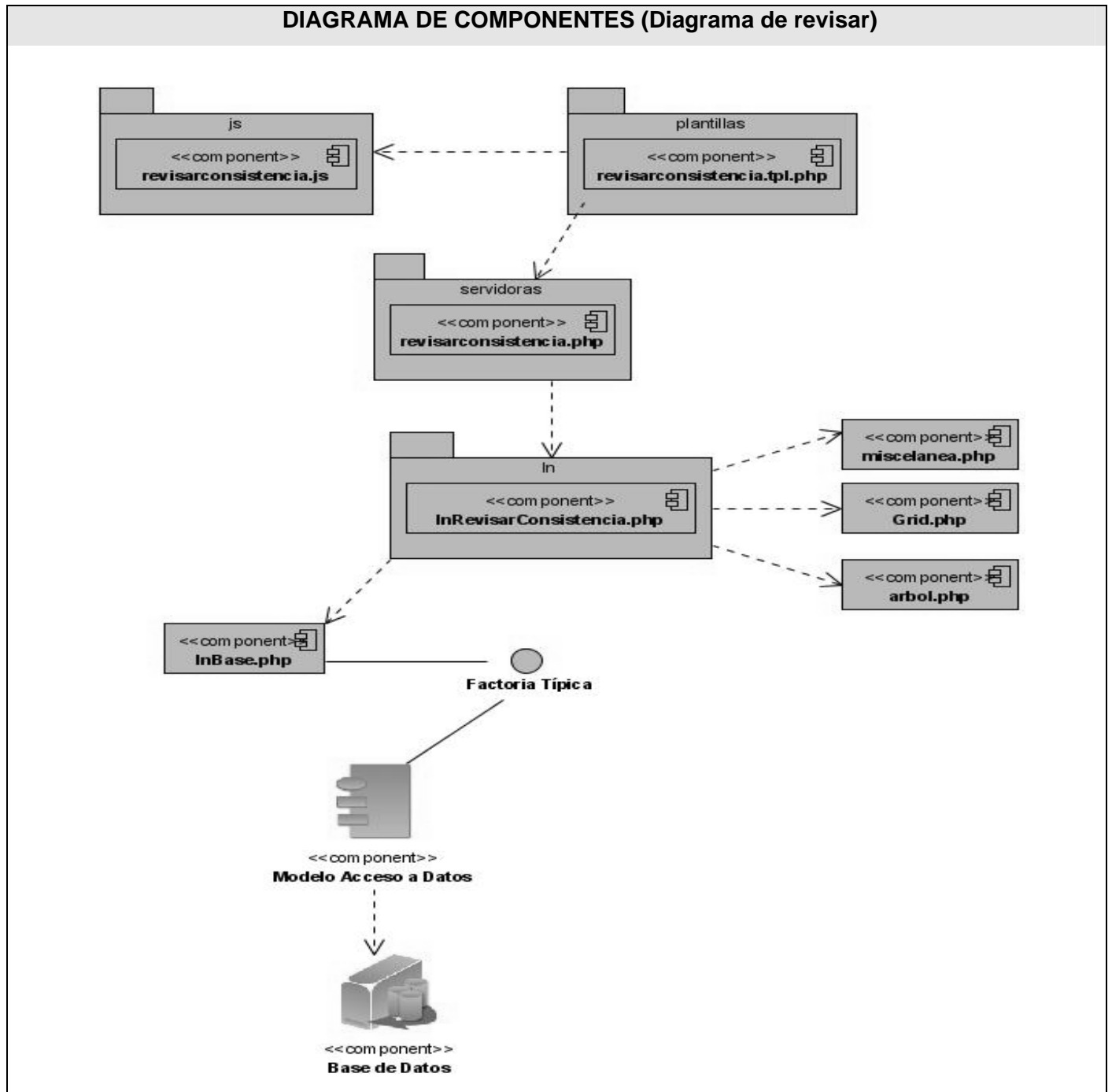


Fig. 4.6 Diagrama de componentes (Revisar)

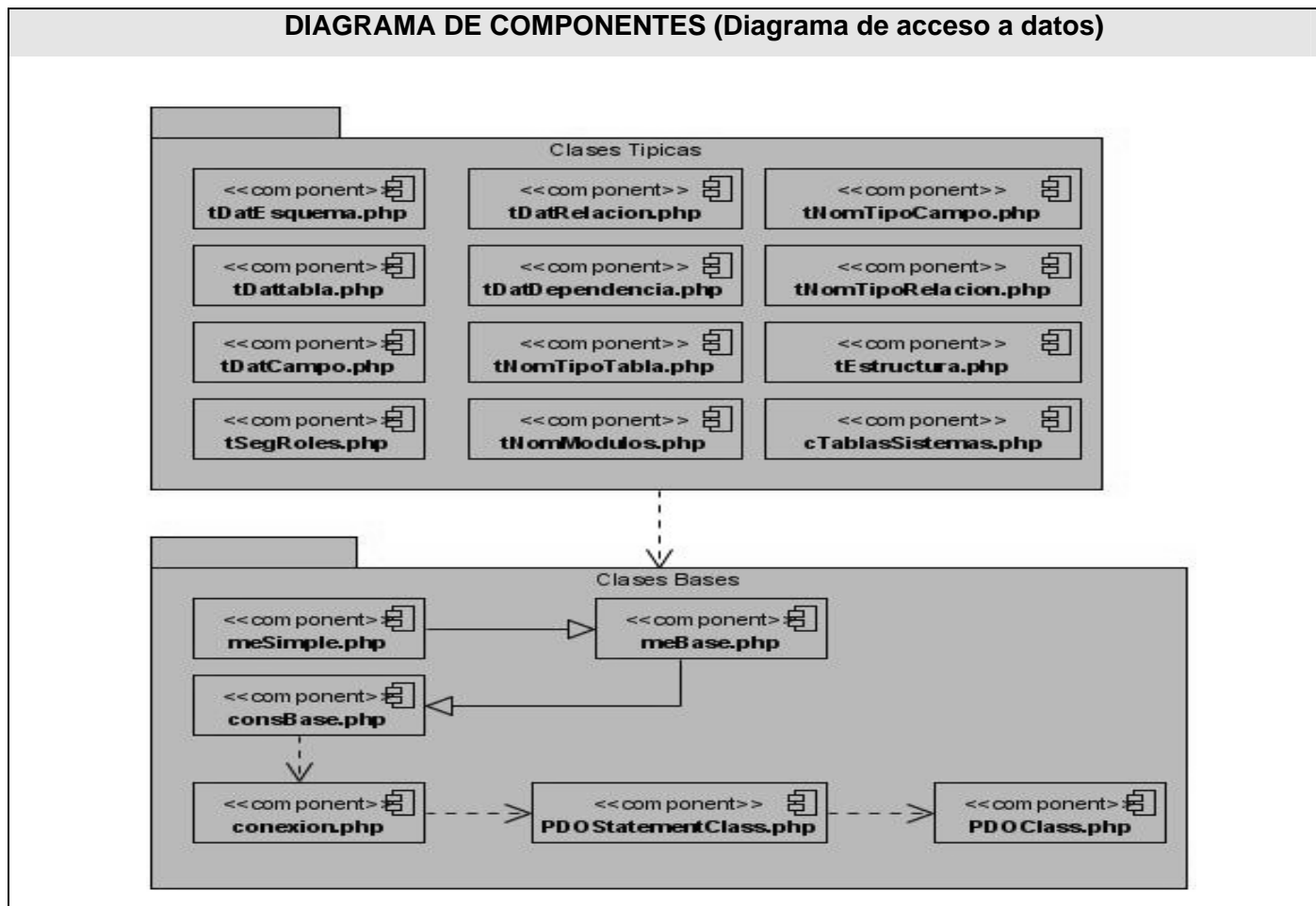


Fig. 4.7 Diagrama de componentes (Acceso a Datos)

4.3 Diagramas de Despliegue

Muestran nodos, conexiones, componentes y objetos. Los nodos representan objetos físicos con recursos computacionales como procesadores y periféricos; pueden mostrarse como una clase o una instancia, por lo que su nombre sigue la misma sintaxis establecida para clases y objetos. Las conexiones son asociaciones de comunicación entre los nodos, y se etiquetan con un estereotipo que identifica el protocolo de comunicación o la red utilizada. Los componentes son archivos de código ejecutable, que residen y se ejecutan dentro de un nodo; se pueden representar relaciones de dependencia entre los componentes que, de manera similar a las dependencias entre paquetes, corresponden al uso de servicios.

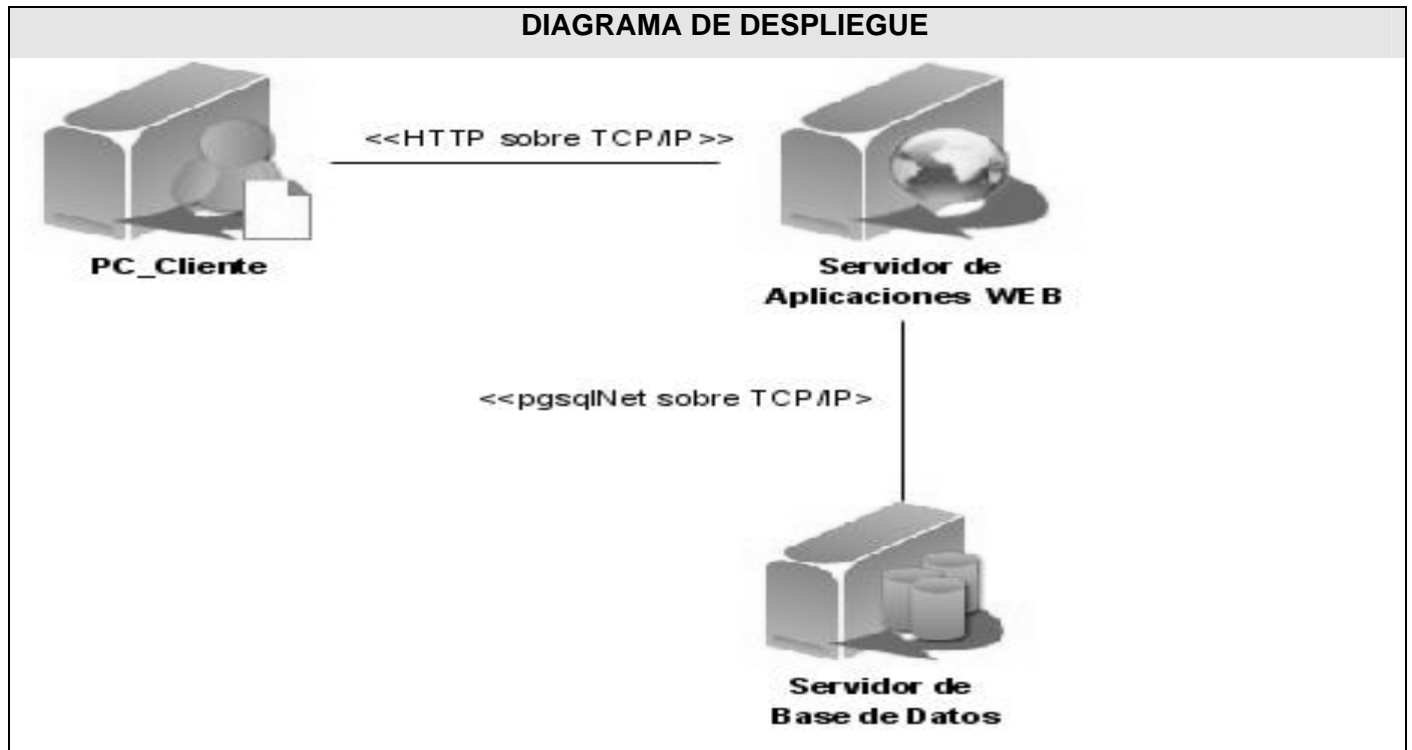


Fig. 4.8 Diagrama de Despliegue

4.4 Conclusiones

En este capítulo se abarcó lo perteneciente a las vistas de implementación correspondientes a la notación UML. De igual manera fueron señalados los contenidos que sobre los principios de diseño debían añadirse a los requisitos funcionales mencionados en el capítulo 2, para de esta forma completar los análisis al respecto. Con este capítulo culmina la modelación completa del Configuración del Generador de Recuperaciones Dinámicas para aplicaciones Web. Y se ha hecho alusión a todas las vistas del Lenguaje Unificado de Modelado (UML).

CONCLUSIONES

Con este trabajo se hace una propuesta de solución a los problemas de recuperación de información presentes en el MINFAR, con la que se logrará una vía más fácil, rápida y precisa en el procesamiento de la información que se desee obtener.

Por lo que se investigó de forma exhaustiva los procesos que se llevan a cabo para realizar la configuración de aplicaciones Web de recuperaciones dinámicas de información, áreas de conocimiento de ingeniería de software, metodologías ágiles y soluciones prácticas de desarrollo.

Con todos los problemas que actualmente existen en la entidad MINFAR a la hora de realizar las recuperaciones, se debe señalar que el surgimiento de esta aplicación trae consigo incontables beneficios, citando como principales: la integración de la información de los distintos departamentos, así como la estabilidad y actualidad de la información que se quiera obtener; trayendo esto consigo el logro de un eficiente manejo de la información.

La propuesta de la herramienta se desarrolló siguiendo la metodología RUP, y se utilizaron representaciones para la modelación de todas las fases del proyecto.

El sistema resultante está provisto de un ambiente cómodo, fácil de entender, que cumple los estándares de diseño y utiliza técnicas modernas de programación orientada a objetos.

Por todo lo expuesto anteriormente se concluye que se cumplieron los objetivos trazados. Se incluyen una serie de recomendaciones a futuras iteraciones.

RECOMENDACIONES

Sin duda no es éste un trabajo estático; los modelos y la solución que se propone habrán de revisarse y actualizarse periódicamente para reflejar los cambios que tan dinámicamente se dan en las disciplinas abordadas, ha quedado claro que la propuesta puede ser mucho más ambiciosa. Por ello se recomienda lo siguiente:

- ✓ Incitamos a todos los que crean poder colaborar con el desarrollo de este software expresen sus ideas y colaboren con los autores en la regulación e implantación del mismo.
- ✓ Hacer este trabajo extensible a todas las aplicaciones Web que necesiten recuperar información y que utilicen como SGBD PostGreSQL.
- ✓ Hacerlo extensible a otros gestores de Base de Datos como por ejemplo: SQLServer, Oracle, etc.

BIBLIOGRAFIA**Citada:**

- [1] PDF El Lenguaje Unificado de Modelado (UML), Enrique Hernández Orallo(ehernandez@disca.upv.es), 2007.
- [2] Ingeniería de Software. Proceso Unificado, jzavalar@yahoo.com , 2002. [http://www.angelfire.com/scifi/jzavalar/apuntes/IngSoftware.html#ProcUnificado]
- [3] Ivar Jacobson, Grady Booch y James Rumbaugh. 2004. El Proceso Unificado de Desarrollo de Software. (Vol. I). La Habana. : Editorial Félix Varela., 2004.
- [4] Instituto Nacional de Estadísticas e Informática, Econ. Félix Murillo Alfaro Jefe tecnología cliente servidor, 2007. [http://www.inei.gob.pe/biblioineipub/bancopub/inf/Lib5038/indice.htm]
- [5] Wikipedia La enciclopedia libre. PHP, 2007. [http://es.wikipedia.org/wiki/PHP]
- [6] Introducción a la Documatica. Los sistema de base de datos y los SGBD, Jesús Tramullas y Kronos, 2000. [http://www.tramullas.com/documatica/2-4.html]
- [7] Raydel Muñoz Vidal y Michel López. Recuperaciones Dinámicas. Universidad de las Ciencias Informáticas. Ciudad de La Habana, 2006.22, 23 y 24
- [8] Monografías.com. Lenguaje de programación para paginas Web HTML, 2007. [http://www.monografias.com/trabajos7/html/html.shtml]
- [9] AJAX un nuevo acercamiento a Aplicaciones Web, 2005. [http://www.uberbin.net/archivos/internet/ajax-un-nuevo-acercamiento-a-aplicaciones-web.php]
- [10] National Instruments Corporation. Ambientes de Desarrollo de Aplicaciones o ADE y Software que Facilita la Integración, 2007. [http://digital.ni.com/worldwide/latam.nsf/web/all/041812AB16B5511686256F34007CB70A]
- [11] PDF Diseño de la arquitectura y los servicios Web. Productos Web y su calidad. Dra. Sofía Álvarez Cárdenas, sofia@ceis.cujae.edu.cu, Instituto Superior Politécnico José Antonio Echeverría (ISPJAE), Centro de Estudios de Ingeniería de Sistemas (CEIS), 2007.
- [12] Desarrollo de una Aplicación en tres Capas con VS .NET. Joel Francia H, 2007. [http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/art140.asp]

Consultada:

[*] Wikipedia La enciclopedia libre. Lenguaje Unificado de Modelado, 2007.
[http://es.wikipedia.org/wiki/Lenguaje_Unificado_de_Modelado]

[*] Wikipedia La enciclopedia libre. Proceso Unificado de Rational, 2007.
[http://es.wikipedia.org/wiki/Proceso_Unificado_de_Rational]

[*] Ivar Jacobson, Grady Booch y James Rumbaugh. 2004. El Proceso Unificado de Desarrollo de Software. (Vol. I). La Habana. : Editorial Félix Varela., 2004.

[*] Wikipedia La enciclopedia libre. Sistema de gestión de base de datos, 2007.
[http://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_base_de_datos]

[*] Wikipedia La enciclopedia libre. Ajax, 2007. [<http://es.wikipedia.org/wiki/AJAX>]

[*] Aplicaciones en capas, 2007.

http://oness.sourceforge.net/proyecto/html/ch03s02.html#three_layers_web

[*] PDF WebSA (Web Software Architecture), 2007.

GLOSARIO DE TÉRMINOS

1. **Adobe Photoshop:** Aplicación informática de edición y retoque de imágenes bitmap, jpeg, gif, etc, elaborada por la compañía de software Adobe inicialmente para computadores Apple pero posteriormente también para plataformas PC con sistema operativo Windows.

2. **Agata Report:** Generador de reportes multi-plataforma, una herramienta de consulta que se conecta a varias Bases de Datos, como PostgreSQL, MySQL, Oracle, y permite exportar los reportes en formatos como PostScript, HTML, XML, PDF o CSV (StarCalc, Excel). Permite definir niveles de datos, subtotales y totales para el relatorio. Permite crear documentos, como cartas y conjugar dinámicamente con los datos provenientes del reporte, así como crear etiquetas de direccionamiento y hasta generar un diagrama ER completo a partir de su banco de datos.

3. **Apache:** Programa de servidor HTTP Web de código abierto. Fue desarrollado en 1995 y actualmente es uno de los servidores Web más utilizados en la red. Usualmente corre en UNIX, Linux, BSD y Windows. Es un poderoso paquete de servidor Web con muchos módulos que se le pueden agregar y que se consiguen gratuitamente en el Internet. Uno de sus competidores es Microsoft.

4. **Carandini:** Herramienta que facilitará a prescriptores técnicos, la localización de modelos que, siendo de familias o gamas diferentes, pueden ser compatibles entre sí. Este configurador ampliará el uso de los productos, ofreciendo nuevas posibilidades a arquitectos e ingenieros.

5. **CASE:** (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) 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.

6. **Crystal Report:** Herramienta para generar reportes en Visual Basic, es orientado al usuario final; es decir que un ejecutivo pueda crear sus propios informes sin necesidad de asistencia de un desarrollador.

7. **CSS:** Cascade Style Sheet. Conjunto de instrucciones HTML que definen la apariencia de uno o más elementos de un conjunto de páginas Web con el objetivo de uniformizar su diseño.

8. **Internet:** Red mundial, de redes de computadoras. Es una interconexión de redes grandes y chicas alrededor del mundo. El Internet empezó en 1962 como una red para los militares llamada ARPANet, para que en sus comunicaciones no existan "puntos de falla". Con el tiempo fue creciendo hasta convertirse en lo que es hoy en día, una herramienta de comunicación con decenas de miles de redes de computadoras

unidas por el protocolo TCP/IP. Sobre esta red se pueden utilizar múltiples servicios como por ejemplo e-mails, WWW, etc.

9. Jasper Report. Es una librería para la generación de informes. Está escrita en java y es "gratuita". El funcionamiento consiste en escribir un xml donde se recogen las particularidades del informe. Este xml lo tratan las clases del Jasper para obtener una salida. Esta salida puede ser un PDF, XML, HTML, CSV, XLS, RTF, TXT Otra ventaja de utilizar Jasper Report es que se integra perfectamente con el JFreeChart que es una librería gratuita para la generación de todo tipo de graficas.

10. Macromedia Dreamweaver: Editor de páginas Web, creado por Adobe. Es el programa de este tipo más utilizado en el sector del diseño y la programación Web, por sus funcionalidades, su integración con otras herramientas como Adobe Flash. Tiene soporte tanto para edición de imágenes como para animación a través de su integración con otras herramientas.

11. Measurement & Automation: Explorador que permite tener el acceso al software y a la configuración de los diferentes dispositivos conectados. Permite configurar el hardware y el software de NI. Crea y edita canales virtuales, tareas, interfaces y escalas virtuales. Detecta los dispositivos e instrumentos conectados al sistema.

12. Microsoft SQL Server: Sistema de gestión de bases de datos relacionales (SGBD) basada en el lenguaje SQL, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea. Así de tener unas ventajas que a continuación se pueden describir.

13. MySQL: Es uno de los Sistemas Gestores de Bases de Datos más populares. Su ingeniosa arquitectura lo hace extremadamente rápido y fácil de personalizar. La extensiva reutilización del código dentro del software y una aproximación minimalística para producir características altamente funcionales, ha dado lugar a un sistema de administración de base de datos de alta velocidad, compactación, estabilidad y facilidad de despliegue. Puede utilizarse gratuitamente y su código fuente está disponible.

14. Oracle: Es una herramienta cliente/servidor para la gestión de Bases de Datos. Es manejador de base de datos relacional que hace uso de los recursos del sistema informático en todas las arquitecturas de hardware, para garantizar su aprovechamiento al máximo en ambientes cargados de información. Oracle corre en PCs, microcomputadoras, mainframes y computadoras con procesamiento paralelo masivo. Oracle Corporation tiene desde Bases de Datos (Oracle) hasta Sistemas de Gestión. Esta entre los primeros del mundo como compañía de software.

15. PHP: Hypertext Preprocessor. Lenguaje de script diseñado para la creación de páginas Web activas, muy popular en Linux, destaca por su capacidad de ser embebido en el código HTML.

16. PostgreSQL: Es un motor de base de datos, es servidor de base de datos relacional libre, permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos.

17. Software: Se refiere a programas en general, aplicaciones, juegos, sistemas operativos, utilitarios, antivirus, etc. Lo que se pueda ejecutar en la computadora.

18. UML: Lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables. Es importante remarcar que UML es un "lenguaje" para especificar y no un método o un proceso, se utilizan para definir un sistema de software, para detallar los artefactos en el sistema y para documentar y construir, es el lenguaje en el que está descrito el modelo. UML cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos de las entidades representadas.

19. Urbandeco: Configurador que permite comparar y seleccionar los productos más adecuados para cada necesidad, así como sus diferentes complementos. Este destinado a arquitectos, permite la configuración del producto deseado y la selección de los diferentes complementos, obteniendo todos los datos técnicos y comerciales del producto configurado, incluso su precio actualizado online.

20. Web: Sistema de documentos de hipertexto enlazados y accesibles a través de Internet. Con un navegador Web, un usuario visualiza páginas Web que pueden contener texto, imágenes u otros contenidos multimedia, y navega a través de ellas usando hiperenlaces.

21. ADEs: Herramienta de programación para crear nueva formas de manejos de aplicaciones.