

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad 7



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

alas **BAP**

Balance y Planificación de Insumos Médicos

alas **BAP**Nomenclador

AUTORES: Elaine Cuza Cuñat

Yordanger Fernández Reyna

TUTORES: Ing. Rotceh Domínguez López

Ing. Alfredo M. Guzmán Martínez

CO-TUTORES: Ing. Annia Arencibia Morales

Ing. Katia Hurtado Duvergel

Ciudad de La Habana, junio 2009

“Año del 50 aniversario del triunfo de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los 29 días del mes de 6 del año 2009.

Elaine Cuza Cuñat

Autor

Yordanger Fernández Reyna

Autor

Ing. Rotceh Domínguez López

Tutor

Ing. Alfredo M. Guzmán Martínez

Tutor

Datos del Contacto

Ing. Rotceh Domínguez López (rdominguez@uci.cu): Recién graduado de Ingeniero en Ciencias Informáticas en la UCI. Posee categoría docente de profesor instructor. Imparte la asignatura Práctica Profesional. Actualmente se desempeña como Arquitecto del área temática Sistema de Apoyo a la Salud.

Ing. Alfredo M. Guzmán Martínez (amguzman@uci.cu) Recién graduado de Ingeniero en Ciencias Informáticas en la UCI. Posee categoría docente de profesor instructor. Imparte la asignatura preparación para la prueba de nivel de Programación (PNP) y Gráfico por Computadora. Actualmente se desempeña como jefe de Negocio.

"Si una persona es perseverante, aunque sea dura de entendimiento, se hará inteligente; y aunque sea débil se transformará en fuerte."

Leonardo Da Vinci

Agradecimientos

De Elaine:

A Dios por acudir a mí en cada momento de debilidad y necesidad.

A mis hermanos Anabel y Rafaelito, a mis cuatro abuelos, a mis padres, a mi tía y a mi familia en general que han hecho posible que este mi sueño se haga realidad, apoyándome y dándome fuerzas en estos 5 años de mi carrera.

A mis amigos que me han apoyado en tiempo de tristeza y alegría, especialmente a Dayrelis, Yunelis, Sonita, Yudmila, Lisseydis, Clenda, Pepón, al chino, Walfri y Yudita A mi novio por ser comprensivo y paciente conmigo, a su familia que ha sido de gran apoyo y guía en mi carrera.

A mi compañero de tesis Yordanger por saber alentarme y darme fuerzas en los momentos más críticos y tristes del desarrollo de esta tesis.

A los Ing. Yurisel Bolmey y Annia Arencibia por su gran ayuda en el desarrollo de esta tesis.

A todos los profesores del proyecto Balance Material por habernos tendido su mano en los momentos de necesidad.

A mis vecinos Avilas-Revilla.

A Fidel y la Revolución por haberme dado la oportunidad de convertirme en una profesional.

De Yordanger:

A mi abuela Dora por hacerme sentir verdaderamente importante, por su educación y por creer en mí.

A mi mamá Zaida que ha sido mi respaldo, mis fuerzas y mis deseos de seguir adelante. A Mayra y Carmen que son como madres. A mis amigos por su apoyo en los tiempos difíciles en especial André, Andresito, Luis Javier, Leodan, Rafael, Aramis, LuisJosé, Walfrido, Maikelis, Kenia, Clenda, Yusdelis e Ilien. Ellos han sido mis padres, mis profesores pero sobre todo han sido mis hermanos apoyándome y ayudándome en todos estos años. A mi compañera de tesis Elaine por llorar mucho pero sin dejar de confiar en que podíamos terminar esta tesis.

A Marta y Angelito por hacerme sentir como un hijo y por apoyarme siempre.

A Angélica por haberme dado momentos inolvidables y convertirme en un hombre nuevo.

A Bolmey y a los profesores del área temática.

A los muchachos del proyecto, Pompa, Under, Mojena, Fabio y Yunier Félix.

A Fidel por darnos la oportunidad de probar que tenemos valor sin importar nuestra posición.

Dedicatoria

De Elaine:

Desde niña siempre soñé con este día pero el mismo no hubiera sido posible ni hubiera llegado sin la ayuda de personas que siempre estuvieron a mi lado, por lo mismo dedico mi tesis:

Especialmente a mi mamá que espero se ponga bien.

A mis hermanos Anabel y Rafaelito, que constituyen el motivo de mi deseo de superarme, para convertirme en soporte y sostén de los mismos.

A mis abuelos Rafael y Rubén ya que soy el orgullo del primero y porque no cuento con la presencia del otro aunque lo llevo en el corazón.

A mis abuelas Iraida y Rosario por llenarme de amor, cariño y comprensión.

A mi papá el cual espero haber llenado de orgullo con este triunfo.

A mi tío Lázaro por ser como mi segundo padre.

A mi sobrina Jazmín, que espero contribuir en su educación y convertirla en una persona de bien.

De Jordanger:

Para que un hombre logre algo verdaderamente importante y hermoso es necesario que algún pilar lo sustente; mi familia ha sido ese pilar gracias a la cual he llegado hasta aquí por lo que quiero dedicar mi tesis :

A mi mamá que ha sido ejemplo vivo de todo lo que una madre es capaz de hacer por su hijo.

A mi súper abuela por sacrificarse tanto por nuestra familia.

A mi abuelo por hacerme mi primer tirapiédras y por enseñarme que hasta el peor de los días tiene su lado bueno.

A mi tía Carmen por ser mi primera maestra enseñándome a leer antes de entrar a la escuela.

A Mayra por ser mi ejemplo de abnegación y sacrificio.

A María y Kike a los que espero poder ayudar en todo lo que les haga falta.

Y a todos los que de una manera u otra tuvieron que ver con el desarrollo de esta tesis.

Resumen

El presente trabajo posee como objetivo desarrollar un componente web que facilite la gestión de la información común y poco variable en el tiempo, correspondiente al producto alas **BAP** del proyecto Balance Material.

La aplicación posee como característica principal la adaptabilidad a los criterios de clasificación de la información. Además, se integra con el sistema externo componente de seguridad, consumiendo información relacionada al área temática Sistema de apoyo a la Salud (SAS) a través de servicios web.

Para el desarrollo del módulo Nomenclador se define como herramienta de modelado Enterprise Architect 7.0, la cual utiliza como lenguaje UML 2.0 y como metodología de desarrollo RUP. Para el diseño de las páginas se utilizó la herramienta Dreamweaver 8 la cual permite escribir código CSS, HTML y JavaScript.

Para la implementación se utilizó el Framework Symfony 1.2, desarrollado completamente en PHP 5, además la herramienta Zend Studio 6.0. Para el manejo de datos el cliente EMS Manager 2007 for PostgreSQL 4.1 y como gestor de base de datos PostgreSQL 8.3.

El sistema Nomenclador aporta un conjunto de beneficios como la facilitación de un componente de apoyo a otras aplicaciones que requieren la utilización de un nomenclador, agilizando considerablemente su tiempo de desarrollo. Además, brinda un mecanismo de almacenamiento y gestión de información común y poco variable, configurable y flexible ante posibles cambios.

Palabras Claves: alas **BAP**, información común y poco variable, SAS, servicios web.

Índice

Introducción	1
Capítulo 1 Fundamentación Teórica.....	4
1.1 Sistema Nacional de Salud.....	4
1.1.1 Sistema Nacional de Salud. Estrategias económicas	5
1.1.2 Informatización del Sistema Nacional de Salud	5
1.2 Estado del arte.	7
1.2.1 Soluciones informáticas existentes	7
1.3 Técnicas, tecnologías, metodologías y software usados para la solución del problema	9
1.3.1 Internet	9
1.3.2 Aplicaciones Web	10
1.3.3 Servidor Web.....	10
1.3.4 Arquitectura	11
1.3.5 SOAP	14
1.3.6 Metodología de Desarrollo RUP	14
1.3.7 Tecnologías y lenguajes usados para la realización del sistema.	15
1.4 Herramientas	21
Capítulo 2 Características del Sistema.....	23
2.1 Modelo de Dominio.....	23
2.1.1 Conceptos fundamentales	23
2.1.2 Diagrama del modelo del dominio.....	25
2.2 Propuesta del sistema	26
2.2.1 Especificación de los requerimientos del software	26
2.3 Modelo del Sistema	31
2.3.1 Definición de los actores del sistema	31
2.3.2 Diagrama de casos de usos del sistema.....	32
2.3.3 Descripción textual de los casos de uso del sistema	34
Capítulo 3 Diseño del Sistema	37
3.1 Modelo de Diseño.....	37
3.1.1 Estructura del diseño	37
3.1.2 Definición de elementos del diseño	38
2.1.3 Diagramas de clases de diseño	40

Índice

3.1.4 Diagramas de Secuencia.....	44
3.1.5 Descripción de las clases	45
3.2 Modelo de Datos	55
3.2.1 Descripción de las tablas de la base de datos del sistema	56
Capítulo 4 Implementación.....	59
4.1 Modelo de Implementación.....	59
4.1.1 Diagrama de Componentes	59
4.2 Diagrama de Despliegue	63
Conclusiones	67
Recomendaciones	68
Referencias Bibliográficas	69
Bibliografía	73
Anexos.....	77

Introducción

El bloqueo económico, comercial y financiero impuesto por los Estados Unidos contra Cuba es el más prolongado y cruel que haya conocido la historia de la humanidad, pues forma parte esencial de su política de hostilidad y agresiones contra el pueblo cubano. Su objetivo, definido desde el 6 de abril de 1960, ha sido la destrucción de la Revolución Cubana. [1]

Para ello ha desarrollado una fuerte guerra económica, que ha afectado todas las ramas de la sociedad cubana, dígase educación, salud, agricultura, turismo entre otras. Estas afectaciones se recrudecieron con el derrumbe del campo socialista, en que Cuba se vio totalmente desamparada económicamente y donde el gobierno destinó todos sus recursos al mantenimiento de los programas de primer orden y de carácter social.

Los negativos impactos de este recrudecido bloqueo sobre el sector de la salud no han detenido los múltiples avances y logros del mismo, siendo considerado y comparado el Sistema Nacional de Salud (SNS) cubano con los de las grandes potencias mundiales, como dijera el Comandante Fidel Castro Ruz:

"Una profunda revolución en los servicios de salud tiene lugar en nuestra patria." [2]

El gran avance alcanzado en esta rama se debe en gran medida a los esfuerzos destinados por el gobierno cubano para el mantenimiento del programa nacional de salud. Actualmente este programa está muy vinculado a la combinación y actualización de las nuevas Tecnologías de la Información y Comunicación (TIC), presentes ya en la sociedad cubana. Las TIC han propiciado que como medida de desarrollo, el gobierno se interese por informatizar todos los sectores económicos y sociales del país, pues las mismas ofrecen ventajas como:

- La cooperación y colaboración entre las distintas entidades.
- El aumento de la producción de los bienes y los servicios de valor.
- El surgimiento de nuevas profesiones.
- Crecimiento de las respuestas innovadoras a los retos del futuro.

Para lograr el mantenimiento y crecimiento del SNS cubano, el gobierno asigna la tarea a distintas empresas e instituciones desarrolladoras de software de comprender, interpretar, adaptar y dar respuestas a las necesidades actuales del Ministerio de Salud Pública (MINSAP) órgano rector del SNS. Ejemplo de estas empresas e instituciones los constituyen SOFTEL, DESOFT y la Universidad de las Ciencias Informáticas (UCI).

Introducción

La UCI consta de varias facultades encargadas del desarrollo de software, estos se encuentran estrechamente vinculados al crecimiento de los distintos renglones económicos y sociales del país. La facultad 7 es responsable de elaborar e implementar software para el SNS, y cuenta para esto con varias áreas temáticas. Sistemas de Apoyo a la Salud (SAS) es el área que posee como objetivos fundamentales, desarrollar software vinculados esencialmente con la salud, y brinda soporte técnico a los productos que se implementan en la misma. Balance Material (BM) es un ejemplo de proyecto desarrollado en SAS y al cual pertenece el producto software Balance y Planificación de Insumos Médicos (alas **BAP**).

Este sistema se estructura en varios módulos, el único desarrollado es el Módulo de Planificación, el cual contenía información embebida que no pertenecía a su negocio, la información sobre los materiales y las especialidades médicas. Las especialidades se dividían en especialidades y subespecialidades, al eliminarse estas últimas provocó que el sistema se reprogramara casi en su totalidad. Además era necesario reestructurar la organización de los materiales, así como agregarles nuevas características a estos. Esta funcionalidad no existía en el sistema antiguo y era imposible de adicionar, pues al ser un software a la medida, su flexibilidad ante cambios era prácticamente nula.

Tanto los materiales como las especialidades son información común y poco variable en el tiempo, por lo que se consideran nomencladores. En la actualidad no existe un mecanismo que posibilite la gestión de esta información y que además posea la característica de ser configurable y flexible a posibles cambios.

Por lo tanto el **problema a resolver** es: ¿Cómo facilitar la obtención de la información común y poco variable en el tiempo, correspondiente al producto alas **BAP** del proyecto Balance Material?

El problema planteado tiene como **objeto de estudio**: La gestión de la información correspondiente al producto alas **BAP** del proyecto BM. Se establece como **campo de acción**: La gestión de la información común y poco variable correspondiente al producto alas **BAP** del proyecto BM.

Como **objetivo general** se plantea desarrollar un componente web que facilite la gestión de la información común y poco variable en el tiempo correspondiente al producto alas **BAP** del proyecto Balance Material.

Introducción

Para dar solución al objetivo antes descrito se deben realizar las siguientes tareas:

1. Analizar cómo se gestiona la información común y poco variable en el tiempo correspondiente, al producto alas **BAP**.
2. Asimilar las herramientas definidas por el Área Temática (AT).
3. Realizar toda la documentación según el ciclo de desarrollo, basándose en la metodología del AT-SAS.
4. Diseñar las interfaces de usuarios del componente.
5. Crear la capa de presentación apoyándose en las herramientas que brinda el framework Symfony y la Yahoo User Interface.
6. Crear la capa de acceso a datos apoyándose en las herramientas que brinda el framework Symfony.
7. Implementar las funcionalidades definidas para el componente web.
8. Generar la documentación del expediente de proyecto relacionada con el componente.

La estructura del contenido del trabajo está conformada por cuatro capítulos:

En el **Capítulo 1 “Fundamentación Teórica”** se registra el estudio del estado del arte del tema en cuestión. Se analizan las tecnologías, metodología y herramientas propuestas que apoyan la solución del problema a resolver.

En el **capítulo 2 “Características del sistema”** se realiza una breve descripción del Modelo de Dominio así como de sus elementos más importantes. Además se especifican los requerimientos funcionales y no funcionales, a partir de los cuales se representan los casos de uso del sistema y la descripción de los mismos.

En el **capítulo 3 “Diseño del sistema”** se efectúan los Diagramas de Clases del Diseño, el Modelo de Datos y se describen una serie de tablas de la base de datos para una mejor comprensión de los diferentes diagramas. Además se muestra la interacción entre los actores y el sistema mediante los Diagramas de Secuencia.

En el **capítulo 4 “Implementación”** se describe la implementación de la aplicación propuesta, a través de la descripción de algunos componentes importantes del sistema. Estarán reflejados los Diagramas de Componentes y el Diagrama de Despliegue.

Capítulo 1: Fundamentación Teórica

Capítulo 1 Fundamentación Teórica

En este capítulo se realiza un estudio de los conceptos más significativos vinculados al dominio del problema, el funcionamiento del Sistema Nacional de Salud, estrategias económicas e informatización del mismo. Además se analizan las tecnologías, metodologías y herramientas propuestas a utilizar en el desarrollo del sistema.

1.1 Sistema Nacional de Salud

El SNS es el conjunto de formas, métodos y actividades que se llevan a cabo para la organización de la atención médica en un país. [3]

En Cuba el organismo rector del SNS es el Ministerio de Salud Pública (MINSAP), encargado de dirigir, ejecutar y controlar la aplicación de la política del Estado y el Gobierno en cuanto a la salud pública, el desarrollo de las ciencias médicas y la industria médico-farmacéutica. [4]

El MINSAP entre sus principios rectores tiene:

- Colaboración internacional.
- Centralización normativa y descentralización ejecutiva.
- Elaboración y proposición del sistema de información estadística en materia de salud, como parte del Sistema de Información Estadístico Nacional.

Funciones Rectoras:

- Regular y controlar la aprobación, ejecución y evaluación de las investigaciones biomédicas o de cualquier tipo que se realice directamente en seres humanos.
- Normar las condiciones higiénicas y el saneamiento del medio ambiente en aquellos aspectos que puedan resultar agresivos a la salud humana y controlar su cumplimiento a través de la inspección sanitaria estatal.
- Regular el ejercicio de la medicina y de las actividades que le son afines.
- Ejercer la evaluación, el registro, la regulación y el control de los medicamentos de producción nacional y de importación, equipos médicos, material gastable y otros de uso médico.

Funciones Específicas

- Dirigir las actividades de producción, importación, registro, distribución, almacenamiento, comercialización y exportación de medicamentos.

Capítulo 1: Fundamentación Teórica

- Dirigir las actividades de comercialización, distribución y almacenamiento de artículos y equipos médicos.
- Dirigir la actividad de comercialización de los servicios de salud, asistencia técnica, la formación y especialización de personal extranjero, así como software y literatura científica afines con la actividad propia.[5]

1.1.1 Sistema Nacional de Salud. Estrategias económicas

Independientemente de la recuperación económica y de los logros recientes alcanzados por la salud pública cubana, la limitación de recursos es un factor presente en la actividad de salud, hecho que se hace más complejo ante el recrudecimiento del bloqueo económico al que Cuba ha sido sometida por parte del gobierno de los Estados Unidos. La toma de decisiones encaminada a asignar recursos a los procesos de la salud pública es una de las tareas más frecuentes y apremiantes que deben enfrentarse en todos los ámbitos del sistema. [6]

El SNS cubano como medida preventiva ha desarrollado un conjunto de técnicas y herramientas que conducen a un intenso aprovechamiento de los recursos, además de desarrollar un profundo análisis económico sostenido sobre bases científicas e investigativas. Para su puesta en marcha el MINSAP cuenta con un departamento económico encargado de la evaluación económica y financiera del ministerio en cuestión, el cual permite reducir las arbitrariedades en la toma de decisiones sobre el uso de los recursos disponibles.

El MINSAP cuenta además con la Dirección de Planificación y Economía como parte de esta esfera de control económico, la cual tiene como objetivo fundamental la realización de los planes económicos anuales y su control. Dichos planes abarcan entre otros aspectos, la planificación de las importaciones de materiales gastables de uso médico, las inversiones en el sector de la salud y la planificación de salud. Esta última incluye la planificación de los portadores energéticos (electricidad, combustibles) de los alimentos y productos de uso no médico, el balance y el control de sus almacenes. [7]

1.1.2 Informatización del Sistema Nacional de Salud

Durante los últimos años un grupo de instituciones cubanas y el propio MINSAP han desarrollado sistemas encaminados a lograr la informatización de la salud. A partir del año 2003 se traza una nueva estrategia para alcanzar la informatización del sector alineada al proceso de informatización cubana. Se utiliza para su construcción las últimas tecnologías de la información y las comunicaciones para el desarrollo de aplicaciones, sobre una arquitectura basada en componentes y orientada a servicios, empleando para este fin software libre.

Capítulo 1: Fundamentación Teórica

A partir de la importancia y de la atención diferenciada que recibe el desarrollo de la informatización y automatización de los servicios de salud, el MINSAP ha definido como una de sus prioridades de trabajo la informatización de dicho organismo. Para ello ha convocado a un grupo de instituciones propias y del Ministerio de Informática y Comunicaciones (MIC) como SOFTEL, DESOFT y la UCI para de manera conjunta definir los proyectos a desarrollar.

Uno de estos, es Balance Material, desarrollado por estudiantes de la UCI, que tiene como objetivo fundamental lograr el balance y planificación de los insumos médicos del país. Siendo necesario desarrollar un módulo nomenclador, capaz de gestionar la información común y poco variable en el tiempo y la cual es considerada de mucha importancia para el funcionamiento del futuro sistema.

Como parte de este proceso de ordenamiento para el desarrollo, el MINSAP ha emitido bajo resolución, un grupo de políticas para la informatización del sector, algunas de las cuales se resumen a continuación:

- El proceso de informatización responde a las Políticas y Principios Socialistas.
- Esta actividad es una estrategia vital y prioritaria del Sistema Nacional de Salud.
- El desarrollo de producciones, inversiones y donaciones de sistemas informáticos debe responder a estrategias, planes de desarrollo, políticas de estandarización y proyectos que serán aprobados centralmente.
- Todos los productos y servicios se integrarán a la ciber-infraestructura del sector y se realizarán en lo fundamental sobre sistemas abiertos, arquitectura orientada a los servicios y basadas en componentes, utilizando software libre y de calidad. Deben constituirse en componentes modulares y estables, que compartan normas y cooperen entre si. La red telemática Infomed constituye la intranet de la salud pública cubana.
- La informatización debe alinearse con las tecnologías de punta y los estándares de calidad desarrollados en el mundo y adecuarse a las condiciones particulares del país.
- La superación y especialización de la informática en salud será una actividad básica para la formación de los recursos humanos.
- La seguridad informática y de contingencia son requisitos imprescindibles y responsabilidad ineludible de los productores, prestadores y usuarios.
- Para garantizar su viabilidad, sustentabilidad y mantenimiento, los programas de informatización en la salud se basarán fundamentalmente en la dirección integrada de proyectos.[8]

Capítulo 1: Fundamentación Teórica

1.2 Estado del arte

Nomenclador como concepto proviene del latín nomenclátor s. m. Lista o catálogo de nombres de personas, lugares o cosas relacionadas entre sí o que tienen algo en común. [9]

En la actualidad este es uno de los términos más utilizados en las distintas esferas económicas y sociales. La salud y la informática no han quedado exentas del mismo, pues nomenciar es la forma más organizada de clasificar, agrupar y estandarizar información. Dada la importancia de este concepto en este epígrafe se muestran ejemplos de sistemas nomencladores existentes tanto en el ámbito nacional como internacional.

1.2.1 Soluciones informáticas existentes

Componente Registro de Estudiantes

Este componente fue desarrollado por estudiantes de la UCI pertenecientes al AT-SAS, específicamente al proyecto Docencia, concerniente a la facultad 7. El mismo surge dada la necesidad de mejorar el proceso de gestión de la información de las secretarías docentes en los centros clínicos educacionales de Cuba. Este sistema está compuesto por varios módulos y posee como objetivo componentizar el módulo Matrícula del Estudiante.

Para el desarrollo de este módulo fue necesaria la utilización de un nomenclador encargado de la configuración de la plantilla Matrícula del Estudiante. Este nomenclador constituía el contenedor de los datos que conformarían la plantilla y era el encargado de que la misma fuera totalmente configurable a nivel de centro, pues brindaba datos según las necesidades de cada uno.

SLD095-EI Registro Informatizado de Salud (RIS)

El Registro Informatizado de la Salud (RIS) es la solución propuesta por el MINSAP para la informatización de la salud pública y la aplicación que contiene los nomencladores nacionales que brindan información para el funcionamiento del resto de las aplicaciones vinculadas a la salud. Más que un software, es una plataforma de aplicaciones abierta, con una interfaz de programación que permite incorporar registros que sean compatibles con ella. Es portable a diferentes sistemas operativos, tanto en los servidores como en los clientes y replicable en otros entornos. Es una plataforma en constante desarrollo y que crece en la medida en que se implementan nuevos módulos.

Componentes o módulos del RIS

- Administración
- Registro de Unidades de Salud (RUS)

Capítulo 1: Fundamentación Teórica

- Registro de Equipos Médicos y no Médicos (REQ)
- Registro de Personal de la Salud (RPS) [10]

Registro de Equipos Médicos y no Médicos (REQ)

Constituye un componente del RIS, que permite el manejo de la información relacionada con los equipos médicos y no médicos existentes en el Sistema de Salud. Posee como objetivos fundamentales el control de los equipos existentes, su estado técnico, su funcionalidad en una institución dada, municipio o provincia.

El uso de este componente permite identificar los principales problemas y limitaciones en la utilización y distribución de los equipos y tecnologías, así como la toma de decisiones para modificaciones administrativas y funcionales por una mayor eficiencia en la utilización de los recursos. Es de gran utilidad al Centro de Control Estatal de Equipos Médicos y a su función reguladora, al ofrecerle una fotografía actualizada del estado del equipamiento médico instalado en el país, lo cual le permite la evaluación permanente y sistemática del equipo en cuestión.[11]

Sistema para la Gestión de la Información de Enfermedades de Declaración Obligatoria (EDO)

El sistema permite automatizar los procesos referentes a la gestión, procesamiento y transmisión de las Enfermedades de Declaración Obligatoria (EDO) en los diferentes niveles asistenciales del SNS. El sistema surge dada la necesidad de informatizar los procesos de gestión, procesamiento y transmisión de información referente a las EDO los cuales se realizaban de forma manual.

EDO permite tener un mejor control de la información de las enfermedades detectadas en el país; además eleva la calidad de la atención médica en los distintos niveles del SNS dada su utilidad para el monitoreo y control epidemiológico, así como la confiabilidad en los datos que se obtengan por medio del sistema, evitando los errores humanos. [12]

Ámbito internacional

Sistema de Información Integrado de los Servicios Públicos de Salud (SEPS)

El sistema de Información Integrado de los Servicios Públicos de Salud (SEPS) detalla todos los procedimientos y servicios disponibles en el mercado privado de salud. SEPS para lograr cumplir con sus funciones de control, supervisión y regulación de este sector, consume

Capítulo 1: Fundamentación Teórica

información común de instituciones médicas que le permita producir conocimiento especializado, oportuno y eficiente.

El sistema, cuenta con un subsistema nomenclador que recolecta, analiza y toma decisiones rectoras en las instalaciones médicas que cuentan con su servicio. Además posee como objetivo resolver dos grandes problemas que en la actualidad rigen este sector de la salud. En primer lugar establecer estrategias eficientes de recolección de datos e intercambio de información y documentación común entre las diferentes organizaciones médicas y por último garantizar que el sistema regulador o verificador disponga en tiempo real de la información mínima, que posibilite hacer el monitoreo, supervisión y regulación del sistema. [13]

Debe mencionarse que ninguno de estos sistemas informáticos han sido seleccionado para su utilización, pues a pesar de constituir sistemas nomencladores no existe entre ellos un mecanismo estándar de gestión de información común y poco variable. Es decir, cada uno gestiona la información que necesita, condicionando de esta manera su flexibilidad ante la gestión de otro tipo de información que no está contemplada en su negocio. Debe resaltarse que uno de los sistemas estudiados es considerado propietario, siendo esta cualidad un aspecto negativo, pues se opone a la política definida por el AT-SAS, la cual trata sobre el uso en mayor grado de herramientas libres.

1.3 Técnicas, tecnologías, metodologías y software usados para la solución del problema

Las tecnologías a utilizar para la construcción del módulo Nomenclador son aquellas que han sido definidas por el Área Temática SAS, las cuales se caracterizan por ser mayormente libres. A continuación quedan expuestos aspectos importantes de las mismas.

1.3.1 Internet

Internet es una enorme red que conecta redes y computadoras distribuidas por todo el mundo, que permite la comunicación, transmisión y búsqueda de información sin grandes recursos tecnológicos ni económicos. Internet surge a razón de la necesidad del Ministerio de Defensa del Gobierno de los Estados Unidos de mantener sus sistemas de cómputos distantes conectados en una red llamada ARPANET. La primera razón de su desarrollo fueron motivos militares. Años más tarde el proyecto sirvió para conectar científicos que fueron desarrollando y ampliando las potencialidades de la misma. Lo que llevó su desarrollo al punto de que en el año 1972 fueran conectadas todas las agencias y proyectos del departamento de defensa y alrededor de 50 universidades.

Capítulo 1: Fundamentación Teórica

Eventualmente la Fundación Nacional de Ciencia, entidad gubernamental de los Estados Unidos, se hizo cargo de la red, conectando las redes que luego darían lugar a lo que se conoce como Internet. [14]

Con el nacimiento de Internet, nacen las denominadas aplicaciones web, que dieron un impulso definitivo a la red y constituyen una forma clara y visual de presentar información. Una página web es un documento que contiene información sobre un tema específico y que es almacenado en un servidor remoto conectado a la red mundial de información, más conocida como Internet. Este documento podrá ser consultado por cualquier persona que se pueda conectar a esta red. [15]

1.3.2 Aplicaciones Web

Las aplicaciones web son sistemas informáticos, usados por los usuarios para acceder a un servidor web a través de Internet o Intranet, y obtener así la información que se encuentra almacenada en ellos. Las aplicaciones web son muy utilizadas hoy en día debido a las ventajas que las mismas ofrecen. Algunas de las cuales se ven reflejadas a continuación:

- Compatibilidad multiplataforma. Tienen un camino mucho más sencillo para la compatibilidad con múltiples plataformas que las aplicaciones de escritorio. Varias tecnologías incluyendo Java, Flash, ASP y Ajax permiten un desarrollo efectivo de programas soportando todos los sistemas operativos principales.
- Actualización. Están siempre actualizadas con la última versión, sin necesitar de llamar la atención del usuario o interferir con sus hábitos de trabajo, con la posibilidad de que va a iniciar nuevas descargas y procedimientos de instalación.
- Inmediatez de acceso. No necesitan ser descargadas, instaladas y configuradas.
- Menos bugs (errores). Con aplicaciones basadas en web, todos los usuarios utilizan la misma versión, y los errores pueden ser corregidos tan pronto como son descubiertos.
- Múltiples usuarios concurrentes. Las aplicaciones web pueden ser utilizadas por múltiples usuarios al mismo tiempo. [16]

1.3.3 Servidor Web

Es un programa que se ejecuta de forma continua en un servidor, manteniéndose a la espera de peticiones hechas por parte de un cliente (un navegador de Internet), contesta estas peticiones devolviendo el contenido que el cliente solicita. Además utiliza el protocolo Hypertext Transfer Protocol (HTTP) el cual está diseñado para transferir los hipertextos, páginas web o páginas Hypertext Markup Language (HTML): textos complejos con enlaces, figuras,

Capítulo 1: Fundamentación Teórica

formularios, botones y objetos incrustados como animaciones o reproductores de música. Actualmente a nivel mundial existen disímiles servidores web, pero el más utilizado, por su estabilidad y su rendimiento, es el servidor web Apache. [17]

1.3.3.1 Servidor Web Apache 2.2

Comenzó a desarrollarse en 2005 pero hoy en día es el servidor web más utilizado del mundo. Representa el complemento perfecto para el desarrollo de páginas dinámicas con PHP, pues comparte con estas muchas de sus características, como gratuidad, popularidad, su sencillez de manejo y versatilidad. Puede ser instalado sobre Linux o sobre Windows. [18]

Es considerado el servidor web por excelencia pues desde su surgimiento ha demostrado que es estable y que posee mejor rendimiento que sus competidores. Además brinda algunas ventajas que son fundamentales para el desarrollo Web como son:

- **Fiabilidad:** Más del 90% de los servidores con más alta disponibilidad funcionan bajo un servidor Apache.
- **Software libre:** El servidor Apache es totalmente gratuito y es distribuido bajo la licencia de Apache en la cual se permite realizarle cambios al código fuente.
- **Extensibilidad:** Se pueden añadir módulos para ampliar aún más las capacidades del servidor. [19]

1.3.4 Arquitectura

La arquitectura, según Roger S. Pressman, es la estructura de las estructuras del sistema, la cual comprende los componentes de software, las propiedades de esos componentes visiblemente externos y las relaciones entre ellos. [20] Esta no es más que la organización de los componentes del sistema de forma que quede registrada la manera en la que colaboran y se relacionan entre ellos. Es una vía en la cual el sistema queda modelado desde distintas perspectivas con el objetivo de lograr y establecer como deberá ser construido el futuro sistema.

1.3.4.1 Modelo Vista Controlador (MVC)

Este patrón de arquitectura de software permite separar los datos de una aplicación, la interfaz de usuario y la lógica de negocio en tres componentes distintos y proporciona múltiples vistas sobre un mismo modelo de datos. El patrón MVC es utilizado frecuentemente en aplicaciones web donde se manejan diferentes interfaces de usuario y el código que provee los datos a la página es dinámico. Los tres elementos esenciales de este patrón son los siguientes:

Capítulo 1: Fundamentación Teórica

- ✓ **Modelo:** Administra el comportamiento y los datos del dominio de la aplicación, responde a requerimientos de información sobre su estado, usualmente formulados desde la vista, respondiendo a instrucciones de cambio para variar el estado de los datos, habitualmente desde el controlador.
- ✓ **Vista:** Este presenta el modelo en un formato adecuado para interactuar, usualmente un elemento de interfaz de usuario.
- ✓ **Controlador:** Este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista. **(Ver Anexo 1)**

Tanto la vista como el controlador dependen del modelo. Esta separación permite construir y probar el modelo independientemente de la representación visual. En aplicaciones web la separación entre la vista (fragmentos de código que se descargan en el navegador para este los visualice al usuario) y el controlador (componentes del lado del servidor que manejan los requerimientos a través de HTTP) esta claramente definida.

Entre las ventajas del estilo señaladas por Microsoft están las siguientes:

- ✓ **Soporte de vistas múltiples:** Dado que la vista se halla separada del modelo y no hay dependencia directa del modelo con respecto a la vista, la interfaz de usuario puede mostrar múltiples vistas de los datos de manera simultánea.
- ✓ **Adaptación al cambio:** Los requerimientos no funcionales de interfaz de usuario tienden a cambiar con mayor rapidez que las reglas del negocio. Los clientes pueden preferir distintas opciones de representación pero dado que el modelo no depende de la vista, agregar nuevas opciones o modificar las ya existentes generalmente no afecta al modelo. [21]

1.3.4.2 Cliente Servidor

Este tipo de arquitectura consiste básicamente en que un programa cliente realiza peticiones a un programa servidor y este es el encargado de responder a las mismas.

Los usuarios son los que inician la invocación del servicio en la parte cliente en la que se construye la solicitud del servicio y es enviada al servidor. El servidor recibe la solicitud del servicio por parte del cliente y devuelve los resultados de la misma en forma de respuesta a la solicitud antes realizada. [22]

Con la implantación de este modelo cada usuario, llamado cliente, tiene la posibilidad de trabajar desde cualquier lugar, solamente haciendo una solicitud al servidor y manipulando la información que este le devuelva en forma de respuesta, en dependencia de las necesidades del cliente.

Capítulo 1: Fundamentación Teórica

Bajo esta forma de comunicación, la información se puede mantener de forma centralizada, evitando así la pérdida de la misma por problemas de hardware u otros motivos relacionados con situaciones técnicas que puedan presentarse con los clientes. Este tipo de modelo es aplicable bajo cualquier circunstancia, el mismo puede implantarse tanto en Internet como en una red local (Intranet); además el procedimiento no es muy costoso ya que se puede aplicar a programas que se ejecutan sobre un solo ordenador.

Características que presenta la arquitectura Cliente Servidor

- ✓ Las funciones de cliente y servidor pueden estar en plataformas separadas, o en la misma plataforma.
- ✓ Un servidor da servicio a múltiples clientes en forma concurrente.
- ✓ Cada plataforma puede ser escalable independientemente. Los cambios realizados en las plataformas de los clientes o de los servidores, ya sean por actualización o por reemplazo tecnológico, se realizan de una manera transparente para el usuario final.
- ✓ Las funciones cliente servidor pueden ser dinámicas. Ejemplo, un servidor puede convertirse en cliente cuando realiza la solicitud de servicios a otras plataformas dentro de la red.
- ✓ El servidor presenta a todos sus clientes una interfaz única y bien definida.
- ✓ El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- ✓ El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo. [23]

Ventajas de la arquitectura Cliente Servidor

- ✓ Permiten un mejor aprovechamiento de los sistemas existentes, protegiendo la inversión. Por ejemplo, la compartición de servidores y dispositivos periféricos (como impresoras) entre máquinas clientes.
- ✓ Se pueden utilizar componentes, tanto de hardware como de software de distintos fabricantes, lo cual contribuye considerablemente a la reducción de costos y favorece la flexibilidad en la implantación y actualización de soluciones.
- ✓ Proporcionan un mejor acceso a los datos.
- ✓ La arquitectura cliente servidor elimina la necesidad de mover grandes bloques de información por la red hacia los ordenadores personales o estaciones de trabajo para su proceso. Los servidores controlan los datos, procesan peticiones y después transfieren sólo los datos requeridos a la máquina cliente. Finalmente la máquina cliente presenta

Capítulo 1: Fundamentación Teórica

los datos al usuario mediante interfaces amigables. Todo esto reduce el tráfico de la red, lo que facilita que pueda soportar un mayor número de usuarios.

- ✓ En una arquitectura como ésta, los clientes y los servidores son independientes los unos de los otros, con lo que pueden renovarse para aumentar sus funciones y capacidad de forma independiente, sin afectar al resto del sistema. [24]

1.3.5 SOAP

SOAP (Simple Object Access Protocol). Protocolo que permite la comunicación entre aplicaciones heterogéneas a través de mensajes. Está basado en XML y es el núcleo de los Web Services. Es independiente de la plataforma y del lenguaje a usar, proporciona un mecanismo estándar de empaquetado de mensajes además facilita una comunicación del estilo RPC (Remote Procedure Calls) entre un cliente y un servidor remoto.

Ventajas de SOAP:

- No está asociado con ningún lenguaje.
- No se encuentra fuertemente asociado a ningún protocolo de transporte.
- Permite la interoperabilidad entre múltiples entornos. [25]

1.3.6 Metodología de Desarrollo RUP

Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un software. Sin embargo, RUP es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, tipos de organizaciones, niveles de aptitud y tamaños de proyecto. El Proceso Unificado está basado en componentes, lo cual quiere decir que el sistema o software en construcción está formado por componentes interconectados a través de interfaces.

RUP posee tres características fundamentales que marcan el desarrollo y adaptación de la metodología en un proyecto.

- Centrado en la arquitectura, lo que posibilita que se defina la estructura fundamental del sistema y que la misma nunca se pierda.
- Guiado por los casos de uso, brinda la característica de que el desarrollo se vea marcado por una serie de construcciones organizadas por los casos de uso, posibilitando una mejor organización y avance en el desarrollo.

Capítulo 1: Fundamentación Teórica

- Iterativo e incremental, posibilita establecer las construcciones del sistema por partes y que a medida que sean terminadas dichas partes se vayan obteniendo incrementos en el desarrollo del software.

Esta metodología divide el proceso en cuatro fases, y a su vez divide estas en iteraciones a través de las cuales se van obteniendo productos que posibilitan incrementos en el software. Las fases definidas son: Fase de Inicio, Fase de Elaboración, Fase de Construcción y Fase de Transición. RUP brinda una serie de flujos de trabajos que son realizados durante las fases anteriormente mencionadas y que garantizan el orden en que deben ser ejecutadas todas las tareas que establece esta metodología, además de lograr que la realización de la aplicación sea de alta calidad, que resuelva las necesidades del usuario dentro de un cronograma y que el producto sea obtenido con el menor costo posible. RUP utiliza el UML (Lenguaje Unificado de Modelado) para representar todos los diagramas en la concepción y diseño a desarrollar. [26]

1.3.7 Tecnologías y lenguajes usados para la realización del sistema.

1.3.7.1 Ajax

El termino Ajax significa “Asynchronous JavaScript + XML” o lo que es lo mismo “JavaScript Asíncrono + XML”. En sí, no es una tecnología sino el uso coordinado de distintas tecnologías que en su conjunto brindan mayor rapidez y eficacia a las aplicaciones web, acelerando la velocidad de ejecución de las mismas. Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

Las tecnologías que conforman Ajax son:

- XHTML y CSS, para crear una presentación basada en estándares.
- DOM, para la interacción y manipulación dinámica de la presentación.
- XML, XSLT y JSON, para el intercambio y la manipulación de información.
- XMLHttpRequest, para el intercambio asíncrono de información.
- JavaScript, para unir todas las demás tecnologías.[27]

Capítulo 1: Fundamentación Teórica

Ajax es una tecnología válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores basado en estándares abiertos como JavaScript y Document Object Model (DOM).

1.3.7.2 JavaScript 1.1

JavaScript es el lenguaje de programación utilizado por los desarrolladores para brindar dinamismo a las páginas web. Una página web dinámica es aquella que incorpora efectos atractivos para los usuarios y que permite la interacción con los mismos.

Este lenguaje posee varias características, entre ellas se puede mencionar que es un lenguaje dirigido por eventos, centrado en describir objetos y funciones que responden a la activación de estos [28]. Es un lenguaje de programación interpretado, por lo que no requiere compilación, es decir, los programas escritos con JavaScript se pueden ejecutar directamente en cualquier navegador sin necesidad de procesos intermedios. [29]

1.3.7.3 HTML

HTML es el acrónimo de Hypertext Markup Language (Lenguaje de marcado de hipertexto), creado en 1989 por Tim Berners-Lee. Es originalmente un subconjunto de SGML (Standard Generalized Markup Language), especializado en la descripción de documentos en pantalla.

El proyecto inicial se basaba en una colección de etiquetas que permitían describir documentos de texto y vínculos de hipertexto que hacían posible el desplazamiento en forma jerárquica entre diferentes documentos. La facilidad de su uso y la particularidad de no ser propiedad de nadie, hizo de HTML el sistema idóneo para compartir información a través de Internet. Inicialmente su intención era que las etiquetas fueran capaces de marcar la información de acuerdo a su significado, pero por diversos motivos los creadores de los navegadores web fueron añadiendo más etiquetas HTML, dirigidas a controlar la representación de la información contenida en el documento. [30]

1.3.7.4 XML

Extensible Markup Language (XML), establecido en febrero 1998, no es más que un metalenguaje de marcado descriptivo, que posee un grupo de reglas que le permite crear sus propios elementos de marcación, los cuales pueden usarse después para describir su contenido. XML se desarrolló porque HTML no estaba diseñado para describir algunos tipos de datos que las personas querían enviar a través de la red, pues este último tenía sus etiquetas predefinidas.

Capítulo 1: Fundamentación Teórica

XML proporciona la flexibilidad y consistencia que no se podía alcanzar con HTML, pues no se tiene que forzar al contenido para que se ajuste al grupo limitado de elementos proporcionados por este. En pocas palabras, XML proporciona un mecanismo estándar para describir cualquier tipo de datos en documentos que son altamente transportables y reutilizables. [31]

1.3.7.5 CSS 2

Cascading Style Sheets (Hojas de Estilo en Cascada), es un mecanismo que describe cómo se va a mostrar un documento en la web, o incluso cómo se va a imprimir. Este mecanismo ofrece a los desarrolladores el control sobre el estilo y el formato de los documentos. Se utiliza para dar estilo a documentos HTML y XML, separando el contenido de la presentación. Estos estilos permiten a los desarrolladores Web controlar el formato de múltiples páginas web al mismo tiempo, posibilitando ante cualquier cambio en el estilo marcado para un elemento en la CSS, que sean afectadas todas las páginas vinculadas a dicha CSS en las que aparezca ese elemento. [32]

1.3.7.6 PHP 5.2

PHP (Hypertext Pre-processor) es un lenguaje de programación interpretado, creado en 1994 por Rasmus Lerdof. Es utilizado habitualmente para la creación de sitios, contenido dinámico para aplicaciones Web y aplicaciones para servidores. Con frecuencia los scripts PHP se embeben en otros códigos como HTML ampliando las posibilidades del diseñador de páginas Web. La interpretación y ejecución de estos scripts se hacen en el servidor, el cliente sólo recibe el resultado y jamás ve el código PHP. Permite conexión con todo tipo de bases de datos como MySQL, PostgreSQL, Oracle, DB2, Microsoft SQL Server, Firebird y SQLite. PHP corre sobre 7 plataformas, funciona en 11 tipos de servidores, ofrece soporte para varios Sistemas Gestores de Bases de Datos (SGBD) y contiene unas 40 extensiones estables, actualmente se encuentra en la versión 5.

Algunas de las más importantes capacidades de PHP son:

- ✓ Integración con varias bibliotecas externas, permitiendo generar documentos Portable Document Format (PDF) y Microsoft Office Excel.
- ✓ Ofrece una solución simple y universal para las paginaciones dinámicas de fácil programación.
- ✓ Soportado por una gran comunidad de desarrolladores, como producto de código abierto, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente, implicando menos costos.

Capítulo 1: Fundamentación Teórica

- ✓ Gran número de funciones predefinidas. A diferencia de otros lenguajes de programación, PHP fue diseñado especialmente para el desarrollo de páginas web dinámicas. Por ello, está dotado de un gran número de funciones que simplifican enormemente tareas habituales como descargar documentos, envío de correo electrónico, creación dinámica de imágenes y gráficos en el servidor, procesamiento de información en formularios, manipulación de cookies y sesiones, transporte de información mediante HTTP y análisis de documentos XML.
- ✓ Análisis léxico para reconocer el tipo de dato almacenado en una variable haciéndose automáticamente, permitiéndole al desarrollador no tener que especificar el tipo de dato de las variables.
- ✓ Posee un conjunto de funciones que previenen la inserción de código malicioso desde el lado del cliente hacia el servidor.[33]

Debe tenerse en cuenta que este potente lenguaje de programación también posee algunas desventajas entre ellas se pueden mencionar que todo el trabajo se realiza en el lado del servidor, no delegando responsabilidades al cliente, puede que en un momento determinado la capacidad de respuesta sea ineficiente en la medida en que las solicitudes al servidor aumenten considerablemente. Además si el código PHP se incluye en código HTML, puede que la legibilidad de este se vea afectada.

1.3.7.7 PostgreSQL 8.3

El gestor de base de datos PostgreSQL 8.3 es un sistema de base de datos de código abierto. Soporta gran parte del estándar SQL y en algunos aspectos está diseñado para que sea extensible por los usuarios. Se caracteriza por posibilitar transacciones ACID, claves foráneas, vistas, secuencias, subpeticiones, lanzadores, tipos y funciones definidos por el usuario, control de concurrencia multiversión. También posee interfaces gráficas de usuario y enlazadores para algunos lenguajes de programación. [34]

Principales Características:

- ✓ Bloqueos Consultivos: permiten el control de objetos de bases de datos a nivel de aplicación usando el motor rápido de bloqueos de PostgreSQL.
- ✓ Sentencias Preparadas: tiene nuevas interfaces administrativas y mejoras de rendimiento en sentencias preparadas.
- ✓ Criptografía: el módulo pgcrypto, soporta criptografía dentro de la base de datos. El mismo fue actualizado con los mejores algoritmos criptográficos.[35]

Capítulo 1: Fundamentación Teórica

1.3.7.8 Framework Yahoo User Interface (YUI) 2.5

El framework o librería Yahoo User Interface (YUI) es un conjunto de utilidades y controles escritos en JavaScript que se utilizan para crear aplicaciones web dinámicas complejas. La empresa Yahoo distribuye gratuitamente la librería YUI en forma de software libre y bajo la licencia BSD, que permite utilizar la librería para proyectos de cualquier tipo, incluso comercial. YUI cuenta con una gran documentación que incluye cientos de ejemplos de uso. [36]

Principales Ventajas:

- Una documentación completa.
- Las utilidades de YUI simplifican el desarrollo para la compatibilidad entre Navegadores basados en técnicas DOM, DHTML y AJAX.
- Los controles de YUI proporcionan elementos visuales atractivos para el diseño de aplicaciones web.
- Tiene la hoja de estilo Reset.css que elimina los estilos predeterminados de los navegadores web (Internet Explorer, Mozilla Firefox, Safari, Opera, entre muchos otros) para la presentación de un sitio Web.
- Todo el desarrollo de YUI, es de código abierto. Permitiendo que cualquier desarrollador web del mundo lo pueda usar libremente.
- La hoja de estilos Reset.css se adapta a casi cualquier proyecto web competente. [37]

1.3.7.9 Symfony 1.2

Symfony es un framework de desarrollo de aplicaciones web el cual brinda un conjunto de herramientas y utilidades que simplifican la construcción de las mismas. Emplea el patrón de diseño MVC con el objetivo de separar las distintas partes que conforman una aplicación web, la lógica de negocio, la lógica de la aplicación y la presentación de la aplicación.

Además proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de aplicaciones más complejas. Symfony está desarrollado completamente con PHP 5, es compatible con la mayoría de gestores de bases de datos tales como MySQL, PostgreSQL, Oracle y Microsoft SQL Server y puede ser ejecutado tanto en plataformas *nix (Unix, Linux, etc.) como en plataformas Windows. [38]

Características de Symfony

- Fácil de instalar y configurar en la mayoría de plataformas, ejemplo de esto los sistemas Windows y los *nix estándares.

Capítulo 1: Fundamentación Teórica

- Independiente del sistema gestor de bases de datos.
- Sencillo de usar, flexible.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
- Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.

1.3.7.10 UML 2.0

El Lenguaje de Modelado Unificado (UML - Unified Modeling Language) es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software. Permite una forma de modelar cosas conceptuales como lo son procesos de negocio y funciones de sistema. Además, de actividades concretas como escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de software reusables.

En las versiones previas del UML, se hacía un fuerte hincapié en que UML no era un lenguaje de programación. Un modelo creado mediante UML no podía ejecutarse. En el UML 2.0, esta asunción cambió de manera drástica y se modificó el lenguaje, de manera tal que permitiera capturar mucho más comportamiento. De esta forma, se permitió la creación de herramientas que soporten la automatización y generación de código ejecutable, a partir de modelos UML.

Ventajas de UML 2.0:

- Produce un aumento en la calidad del desarrollo.
- Reduce los costos del proyecto.
- Mejora en un 50% o más los tiempos totales de desarrollo.
- Permite especificar la estructura y el comportamiento del sistema y comunicarlo a todos los integrantes del proyecto.
- Brinda la posibilidad de obtener un "plano" del sistema.
- Permite dimensionar mejor los riesgos de un proyecto, tener un mejor rendimiento antes de construir el sistema.
- Facilita la documentación de las decisiones de la arquitectura del proyecto.
- Ofrece un mejor soporte a la planificación y control del proyecto.
- Ofrece mayor rigurosidad en la especificación.
- Permite realizar una verificación y validación del modelo realizado.
- Se pueden automatizar determinados procesos y permite generar código a partir de los modelos y viceversa.[39]

Capítulo 1: Fundamentación Teórica

1.3.7.11 Enterprise Architect 7.0

Es una herramienta de análisis de negocio y UML, orientada a objetos para el desarrollo completo del ciclo de vida. Enterprise Architect provee el límite competitivo para el desarrollo de software, administración de proyecto, administración de requerimientos y análisis de negocio. Entre sus características más importantes figuran:

Última especificación UML 2.0.

Importación/Exportación XMI 2.0.

Nuevo motor de Reporte HTML.

Perfiles y soporte de Tecnologías.

Pruebas, rastreo de recursos, mantenimiento.

Archivos compartibles o modelos basados del repositorio.

Ingeniería reversa de código fuente en 10 lenguajes.

Importar esquema de base de datos.

Importar fuente XSD y WSDL. [40]

1.4 Herramientas

Basándose en los lineamientos arquitectónicos del Área Temática SAS y para dar solución al problema planteado, el equipo de desarrollo decide utilizar para el modelado del sistema la herramienta Enterprise Architect 7.0, la cual utiliza como lenguaje UML 2.0 y como metodología de desarrollo RUP.

Para el diseño de las páginas se utiliza la herramienta Dreamweaver 8 que permite escribir código CSS, HTML y JavaScript. Para el desarrollo se utiliza el Framework Symfony 1.2, desarrollado completamente en PHP 5, además la herramienta Zend Studio 6.0 pues brinda facilidades para escribir código PHP. Para trabajar con la base de datos el EMS Manager 2007 for PostgreSQL 4.1 como cliente y como gestor de base de datos se utiliza PostgreSQL 8.3.

Capítulo 1: Fundamentación Teórica

Conclusiones

Existe un gran número de sistemas nomencladores, pero ninguno posee la característica de ser configurable y flexible ante la posible gestión de otro tipo de información que no esté contemplada en su negocio, condicionando su utilización para el desarrollo de la investigación.

El equipo decidió acoger las herramientas propuestas por el grupo de arquitectura de AT-SAS pues las mismas son consideradas herramientas y tecnologías libres y multiplataforma.

Capítulo 2: Características del Sistema

Capítulo 2 Características del Sistema

El presente capítulo tiene como objetivo la descripción de las características del sistema además de la propuesta de solución para el Módulo Nomencladores. Al existir poca identificación de los procesos del negocio se desarrolla un Modelo de Dominio, además se analizan los requerimientos funcionales y no funcionales lo cual permite un mejor entendimiento del sistema.

2.1 Modelo de Dominio

El Modelo de Dominio es la representación de los conceptos más importantes y significativos en el desarrollo de un sistema. Este representa clases conceptuales del dominio del problema, conceptos del mundo real, no de los componentes de software.

El objetivo fundamental del mismo es obtener "el panorama" de las interrelaciones de los objetos más importantes representados mediante clases. Además, desempeña un papel central en la comprensión del entorno actual y en planificación futura de la posible aplicación.

Para el modelado del módulo Nomenclador se decide realizar un Modelo de Dominio debido a que las fronteras del negocio no son bien delimitadas. Además, los conceptos y las definiciones interrelacionadas fueron obtenidos sobre la base de las necesidades existentes en el proyecto BM, pues representan objetos del mundo real que serán tratados por el sistema en elaboración.

2.1.1 Conceptos fundamentales

Para obtener una mejor visión del diagrama Modelo de Dominio, a continuación se proporciona un marco conceptual con las definiciones identificadas en la elaboración del desarrollo del software.

Nomenclador: Representa el contenedor de la información que se desea gestionar, está estructurado jerárquicamente por grupos, subgrupos e información.

Usuario: Representa una persona o individuo que interactúa con el sistema Nomenclador y el cual posee ciertos grados de accesibilidad. Encargado de la gestión de la información del sistema.

Administrador: Representa el usuario del sistema encargado de la configuración y la gestión de la información del Nomenclador, así como de la construcción de su estructura jerárquica, es decir grupos, subgrupos e información.

Capítulo 2: Características del Sistema

Grupos: Representa la raíz de la estructura jerárquica de un nomenclador, conformado por subgrupos e información.

Subgrupos: Los subgrupos, jerárquicamente son antecidos por un grupo o por otro subgrupo y pueden ser contenedores de otros subgrupos o de la estructura información.

Información: Representa el final de la estructura jerárquica de un nomenclador.

Campo: Representa los distintos atributos o características con los que puede contar la estructura jerárquica de un nomenclador.

Estado: Este concepto posibilita conocer el estado actual de la información en el sistema.

Activo: Representa el estado activo de la información. La información está siendo usada en el sistema.

Pasivo: Representa el estado pasivo de la información. La información no está siendo usada en el sistema es decir, ha cambiado de estado.

Capítulo 2: Características del Sistema

2.1.2 Diagrama del modelo del dominio

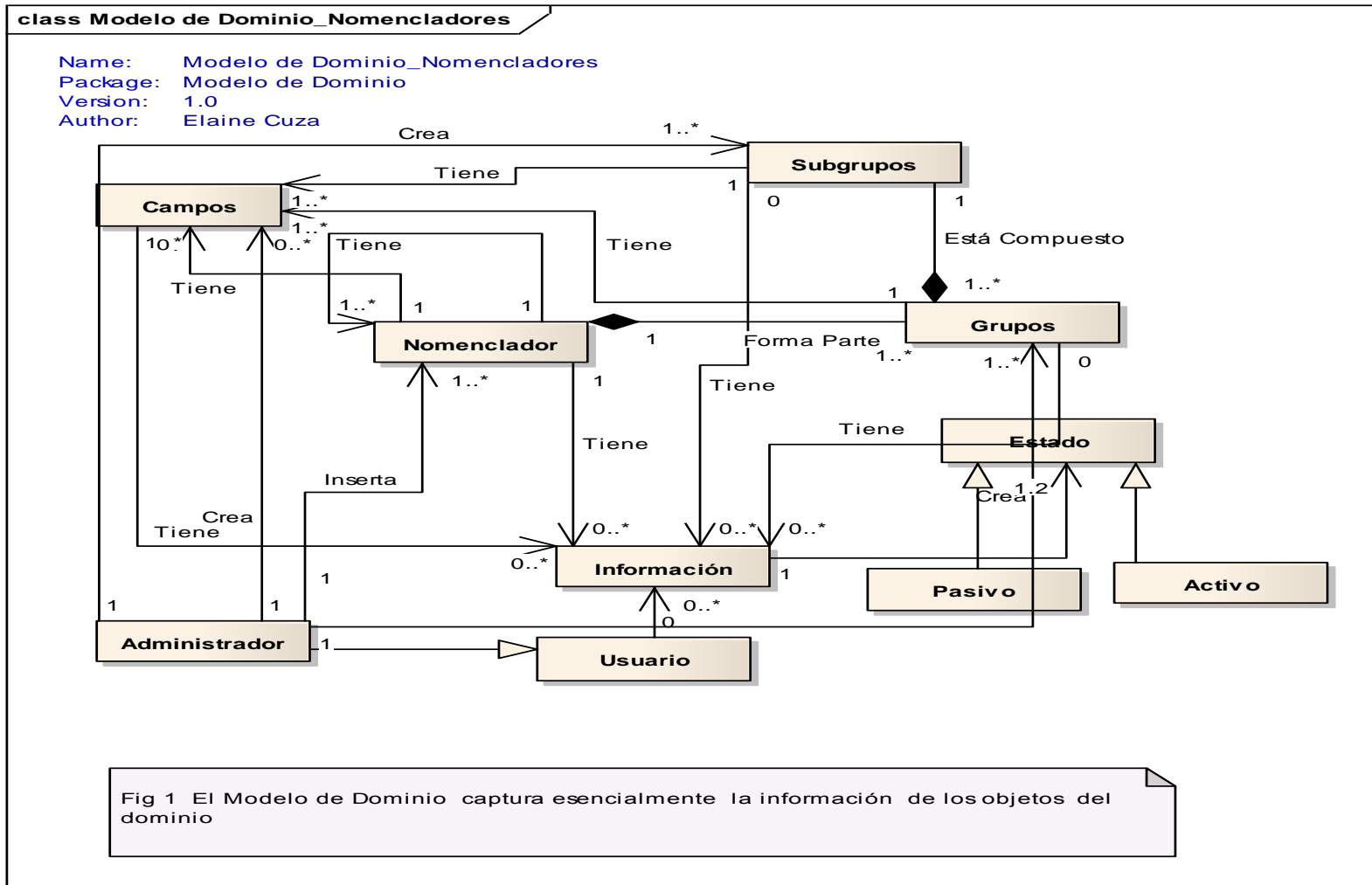


Fig. 2.1 Modelo de dominio módulo Nomenclador

Capítulo 2: Características del Sistema

2.2 Propuesta del sistema

Se propone la construcción de una aplicación capaz de nombrar información común y poco variable y que posea como característica configurabilidad y flexibilidad ante posibles cambios, la misma debe permitir:

- Configurar los campos de los elementos nombrados.
- Asignar los campos a los elementos nombrados.
- Introducir la información correspondiente de los nomencladores o codificadores.

2.2.1 Especificación de los requerimientos del software

Los requerimientos representan una condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo, son características que un sistema debe tener para cubrir algunas de las necesidades que lo motivan.

El objetivo fundamental de la captura de los requerimientos del software es traducir de forma sencilla el problema tal y como lo ve el cliente, logrando un mejor entendimiento entre el usuario final y los desarrolladores. Este entendimiento sobre lo que debe y no debe hacer el sistema permite identificar las funcionalidades requeridas y las restricciones que se imponen.

2.2.1.1 Requerimientos Funcionales (RF)

Condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente. Son acciones que el sistema debe realizar, para que se cumpla el objetivo y darle solución al problema identificado.

A continuación se muestra el listado de requerimientos obtenido luego de ser analizados los principales conceptos del Modelo de Dominio.

RF 1 Autenticar	RF 25 Seleccionar Campo
RF 2 Ver Información Adicional	RF 26 Listar Campo
RF 3 Insertar Grupo	RF 27 Cancelar Campo
RF 4 Modificar Grupo	RF 28 Insertar Información
RF 5 Eliminar Grupo	RF 29 Modificar Información
RF 6 Buscar Grupo	RF 30 Eliminar Información
RF 7 Seleccionar Grupo	RF 31 Buscar Información
RF 8 Listar Grupo	RF 32 Asignar Información

Capítulo 2: Características del Sistema

RF 9 Cancelar Grupo	RF 33 Seleccionar Información
RF 10 Actualizar Grupo	RF 34 Listar Información
RF 11 Insertar Subgrupo	RF 35 Cancelar Información
RF 12 Modificar Subgrupo	RF 36 Actualizar Información
RF 13 Eliminar Subgrupo	RF 37 Insertar Histórico
RF 14 Buscar Subgrupo	RF 38 Buscar Histórico
RF 15 Seleccionar Subgrupo	RF 39 Seleccionar Histórico
RF 16 Listar Subgrupo	RF 40 Listar Histórico
RF 17 Cancelar Subgrupo	RF 41 Cancelar Histórico
RF 18 Actualizar Subgrupo	RF 42 Generar Código
RF 19 Insertar Campo	
RF 20 Modificar Campo	
RF 21 Eliminar Campo	
RF 22 Buscar Campo	
RF 23 Asignar Campo	
RF 24 Agregar Campo	

Tabla 2.2 Requisitos funcionales

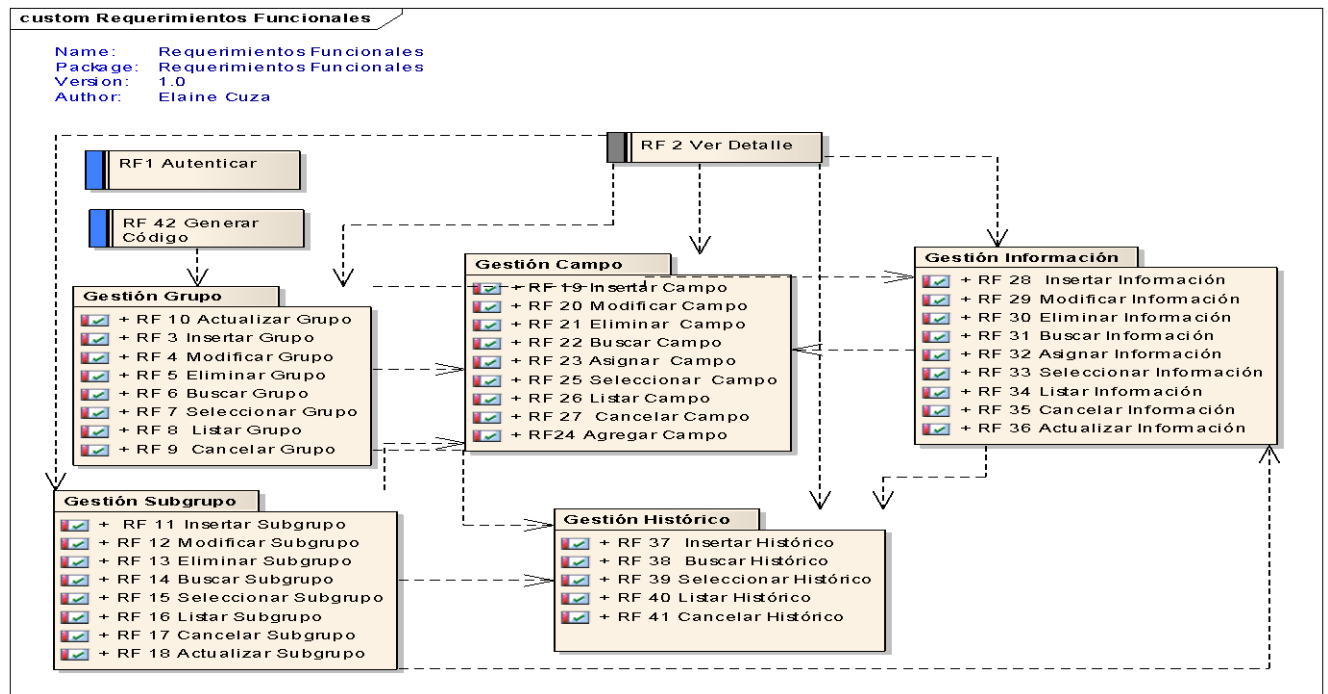


Fig. 2.3 Diagrama de los requisitos funcionales módulo Nomenclador

Capítulo 2: Características del Sistema

2.2.1.2 Requerimientos No Funcionales (RNF)

Son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. Los requerimientos no funcionales son fundamentales en el éxito del producto y normalmente están vinculados a requerimientos funcionales, es decir una vez que se conozca lo que el sistema debe hacer se puede determinar cómo ha de comportarse, qué cualidades debe tener, cuán rápido o grande debe ser. [41]

Usabilidad

Estos requerimientos describen los niveles apropiados de usabilidad, dados los usuarios finales del producto, para ello debe revisarse la especificación de los perfiles de usuarios.

RNF1: El sistema debe presentar un acceso fácil y rápido, para facilitar el uso del mismo por usuarios con conocimientos informáticos básicos, tales como: manejo de la computadora, y específicamente de un navegador web.

Seguridad

La seguridad informática puede ser tratada mediante tres aspectos esenciales:

- **Confidencialidad:**

RNF2: La información estará protegida contra accesos no autorizados utilizando mecanismos de autenticación y autorización, lográndose con la utilización de servicios web que brinda el Componente de Seguridad del Área Temática.

RNF3: La autenticación será la primera acción del usuario en el sistema y consistirá en suministrar un nombre de usuario único y una contraseña que debe ser de conocimiento exclusivo de la persona que se autentica. Si el usuario autenticado no se encuentra registrado se debe reportar un error de acceso.

- **Integridad:**

RNF4: La información podrá ser modificada solo por personal autorizado.

RNF5: Se harán validaciones de la información tanto en el cliente como en el servidor contra ataques de inyección HTML o SQL.

- **Disponibilidad:**

Capítulo 2: Características del Sistema

RNF6: Se logrará que el sistema esté disponible en horario laborable, permitiendo que en el tiempo no laborable se puedan realizar tareas de soporte tanto de hardware como de software.

Eficiencia:

RNF7: El sistema deberá ser rápido ante las solicitudes de los usuarios y en el procesamiento de la información, el tiempo de respuesta deberá ser el menor posible.

Soporte:

RNF8: Una vez terminado el sistema se realizarán procesos de despliegue, capacitación y mantenimiento de software.

RNF9: Para agilizar el tiempo de soporte se deben utilizar estándares de codificación permitiendo una asimilación rápida del código fuente.

RNF10: Se deberá realizar una documentación detalla de cada clase implementada en el código fuente.

Requisitos para la documentación de usuarios en línea y ayuda del sistema:

RNF11: Se dispondrá de un Manual de Usuario que indicará como interactuar con las funcionalidades del sistema.

Interfaces de usuario:

RNF12: El sistema debe presentar una estructura y diseño de interfaz homogénea permitiendo manejo rápido y confiable de la información.

RNF13: Se deben utilizar tonalidades de colores claros y agradables a la vista, lo cual permite una mayor aceptación por parte del usuario final.

RNF14: Todos los mensajes de error o de advertencia del sistema deben mostrarse de forma similar permitiendo esto una mayor organización y visualización de la información.

RNF15: La navegabilidad del sistema debe ser simple, permitiendo un manejo rápido y sencillo de la información a gestionar en la misma.

RNF16: El sistema emitirá diferentes mensajes, algunos informativos, otros de advertencia, los cuales servirán para informar al usuario acerca de la realización satisfactoria o no de determinadas acciones, y en otros casos, para indicar que se no se ha introducido algún dato obligatorio o que se introdujo de forma incorrecta.

Interfaces Hardware común:

RNF17: Ordenador Pentium IV.

RNF18: Se requiere tarjeta de red.

Capítulo 2: Características del Sistema

Estaciones de trabajo:

RNF19: Se requiere que tenga al menos:

- 512 MB de memoria RAM
- 10 GB de disco duro
- 3 GHz como mínimo

Hardware para los servidores:

Aplicación:

RNF20: Se requiere que tenga al menos:

- 1 GB de memoria RAM
- 40 GB de disco duro como mínimo
- 3 GHz como mínimo

Base de Datos:

RNF21: Se requiere que tenga al menos:

- 1 GB de memoria RAM
- 80 GB de disco duro como mínimo
- 3 GHz como mínimo

Software para el cliente:

RNF22: Se requiere que tenga al menos:

- En las computadoras de los clientes se requiere del navegador Web Mozilla Firefox en versión 3.0 o superior.
- Sistema operativo Windows XP Service Pack 2 o GNU/Linux distribución Ubuntu 8.04.

Software para los servidores:

Aplicación:

RNF23: Se requiere que tenga al menos:

- Servidor web Apache versión 2.2 o superior.
- PHP 5.2 y el Framework Symfony 1.2.
- Sistema operativo GNU/Linux en su distribución Debian 4 Etch preferentemente o Ubuntu 8.04.

Base de Datos:

RNF24: Se requiere que tenga al menos:

Capítulo 2: Características del Sistema

- Postgre SQL versión 8.3
- Sistema operativo instalado GNU/Linux en su distribución Debian 4 Etch preferentemente o Ubuntu 8.04.

Estándares Aplicables:

RNF26: Para la implementación del sistema se deberán seguir los estándares de codificación y diseño definidos por el área temática.

2.3 Modelo del Sistema

El modelo del sistema permite que los usuarios y los desarrolladores del software alcancen un entendimiento común relacionado con las capacidades y cualidades que debe poseer el sistema. Cada usuario del sistema será representado mediante un actor. Estos interactúan con el sistema mediante los casos de uso que representan una especificación de secuencia de acciones lógicas, incluyendo variantes que el sistema puede desarrollar y que producen un resultado observable de valor para un actor concreto.

Este modelo es considerado la base fundamental para el análisis, diseño e implementación, por lo que mientras mejor estructurado se encuentre el mismo, mucho más sencillo será el proceso de construcción de sistema de software.

2.3.1 Definición de los actores del sistema

A continuación se muestra un listado de los actores definidos para el sistema. Estos actores pueden estar representados por un usuario en concreto (una persona u otro sistema externo) que interactúe con el mismo.

Actor	Descripción
Usuario	Persona o individuo que interactúa con el sistema, el cual posee ciertos grados de accesibilidad (es decir ciertos privilegios que le son asignados según el rol que desempeñe el usuario), es el encargado de la gestión de la información común y poco variable del posible nomenclador.
Administrador	Usuario del sistema, encargado de la configuración y de la gestión de la información

Capítulo 2: Características del Sistema

	de la aplicación.
Componente de Seguridad	Sistema externo encargado de la seguridad de la aplicación, pues es el sistema responsable de gestionar los requerimientos de seguridad de los productos desarrollados por el AT. Brinda privilegios para acceder a determinados servicios web.

Tabla 2.4 Descripción de los actores del sistema

2.3.1.1 Vista Global de los Actores

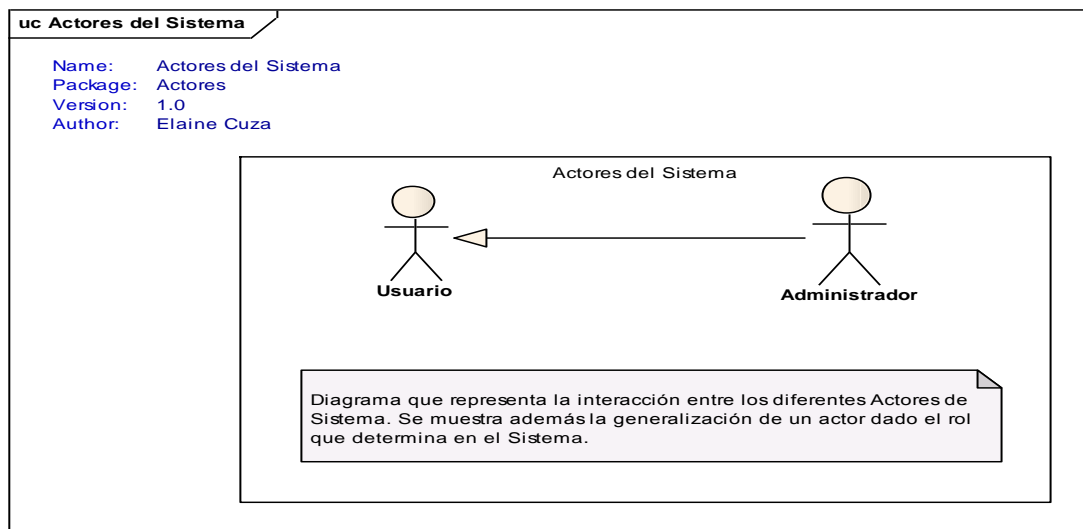


Fig. 2.5 Vista global de actores

2.3.2 Diagrama de casos de usos del sistema

A continuación se muestra el Diagrama de Casos de Uso del Sistema donde se pueden apreciar las relaciones que se establecen entre actores y casos de uso.

Capítulo 2: Características del Sistema

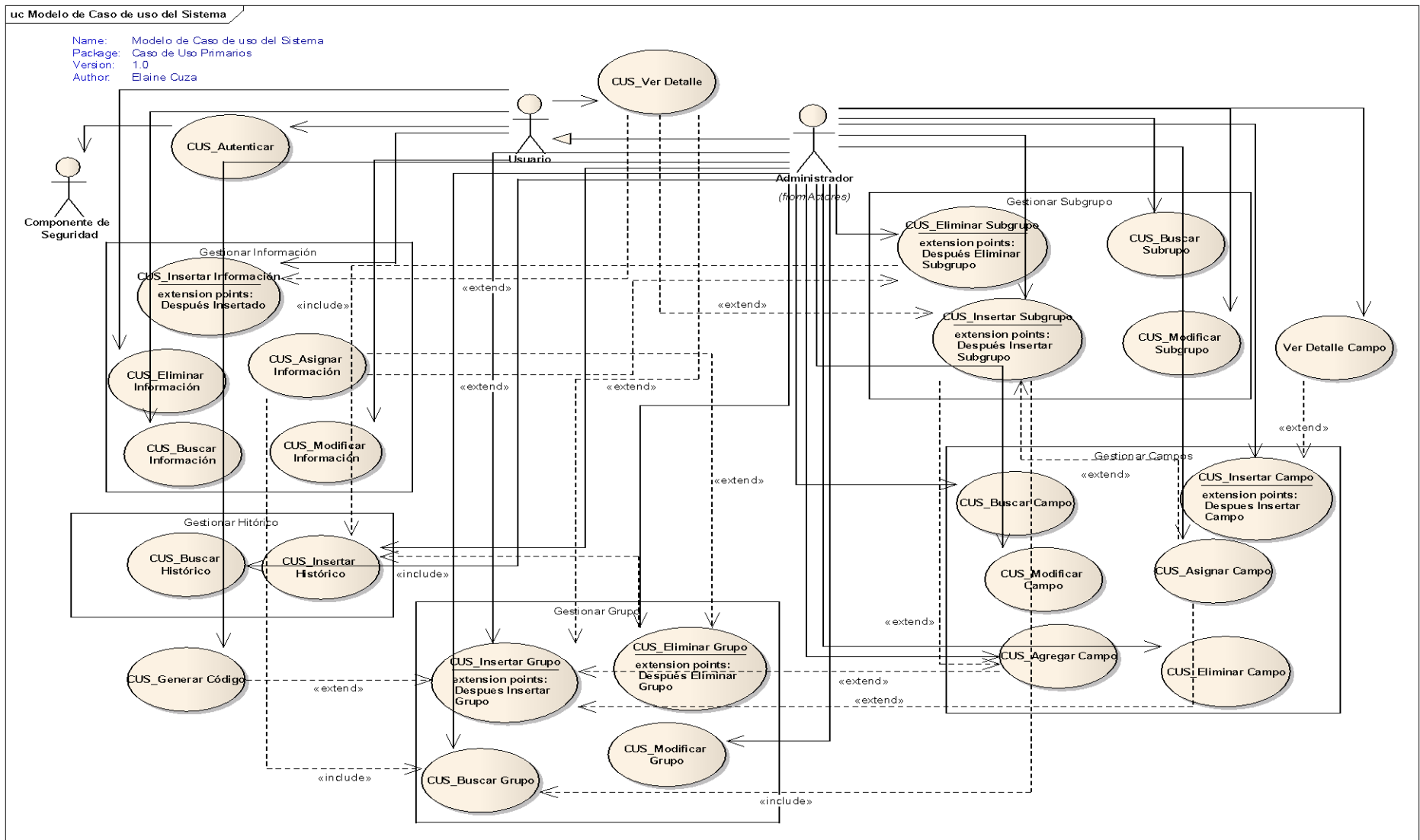


Fig. 2.6 Diagrama de casos de uso del sistema

Capítulo 2: Características del Sistema

2.3.3 Descripción textual de los casos de uso del sistema

A continuación se realiza la descripción de los casos de usos arquitectónicamente significativos. Las descripciones restantes serán mostradas en el Expediente de Proyecto.

Caso de Uso:	CUS_ Insertar Grupo
Resumen:	El caso de uso inicia cuando el administrador accede a la opción insertar grupo, el sistema brinda la posibilidad de introducir los datos para insertar el nuevo grupo, el administrador introduce los datos del grupo (nombre, código, descripción y nulo) y el sistema inserta un nuevo grupo.
Complejidad:	Alta
Prioridad:	Crítica
Precondiciones:	El usuario debe tener los permisos de acceso para introducir un grupo.
Actores:	Administrador (Inicia)
Requisitos:	RF 3, RF 6 RF 8, RF 9, RF 10
Entidades:	Grupo
Casos De uso:	CUS_Ver Detalle(Extendido), CUS_Generar Código(Extendido)
Poscondiciones:	Se insertó un grupo por el Administrador del sistema.

Tabla 2.7 Descripción textual CUS_ Insertar Grupo

Caso de Uso:	CUS_ Insertar Subgrupo
Resumen:	El caso de uso inicia cuando el administrador accede al árbol del sistema y al hacer clic secundario sobre un grupo o subgrupo elige la opción insertar subgrupo, el sistema muestra una pantalla donde brinda la posibilidad de insertar los datos del nuevo subgrupo (nombre, código, descripción y nulo).El caso de uso finaliza cuando ha sido insertado el nuevo subgrupo.
Complejidad:	Alta
Prioridad:	Crítica
Precondiciones:	El usuario debe tener los permisos de acceso para introducir un subgrupo.
Actores:	Administrador (Inicia)

Capítulo 2: Características del Sistema

Requisitos:	RF 11, RF 14, RF 16 ,RF 17, RF 18
Entidades:	Subgrupo
Casos De uso:	CUS_Ver Detalle (Extendido)
Poscondiciones:	Se insertó un subgrupo por el Administrador del sistema.

Tabla 2.8 Descripción textual CUS_Insertar Subgrupo

Caso de Uso:	CUS_Insertar Información
Resumen:	El caso de uso inicia cuando el administrador accede al árbol del sistema y al hacer clic secundario sobre un grupo o subgrupo elige la opción insertar Información, el sistema muestra una pantalla donde brinda la posibilidad de insertar los datos de la nueva Información (nombre, código, descripción y nulo).El caso de uso finaliza cuando ha sido insertada la nueva información.
Complejidad:	Alta
Prioridad:	Crítica
Precondiciones:	El usuario debe tener los permisos de acceso para introducir una Información.
Actores:	Usuario (Inicia)
Requisitos:	RF 28, RF 31, RF 34 , RF 35 , RF 36
Entidades:	Información
Casos De uso:	CUS_Ver Detalle (Extendido)
Poscondiciones:	Se insertó una información por el Administrador del sistema.

Tabla 2.9 Descripción textual CUS_Insertar Información

Caso de Uso:	CUS_Insertar Campo
Resumen:	El caso de uso inicia cuando el administrador accede a la opción insertar campo, el sistema brinda la posibilidad de introducir los datos para insertar el nuevo campo, el administrador introduce los datos del campo (nombre, tipo de campo, tamaño, nulo, tipo componente y descripción). El caso de uso termina cuando es insertado el nuevo campo en el sistema.

Capítulo 2: Características del Sistema

Complejidad:	Alta
Prioridad:	Crítica
Precondiciones:	El usuario debe tener los permisos de acceso para introducir un campo.
Actores:	Administrador (Inicia)
Requisitos:	RF 19, RF 22, RF 26, RF 27
Entidades:	Campo
Casos De uso:	CUS_Ver Detalle Campo (Extendido)
Poscondiciones:	Se insertó un campo por el Administrador del sistema.

Tabla 2.10 Descripción textual CUS_ Insertar Campos

Debe mencionarse que con el desarrollo de la investigación fueron documentados casos de uso del sistema que serán implementados en próximas iteraciones, ejemplo de esto lo constituyen:

CUS_Generar Código

CUS_Asignar Información

CUS_Insertar Histórico

CUS_Buscar Histórico

CUS_Autenticar

Conclusiones

Con el desarrollo del capítulo se obtuvo una representación del sistema que se desea desarrollar teniendo como punto de partida los RF y RNF definidos en el mismo. Se identificaron los clientes que interactuarán con el sistema teniéndose en cuenta que el usuario con mayores privilegios y que realizaría mayor cantidad de funcionalidades lo constituye el administrador. Debe mencionarse que este capítulo constituye la plataforma para el desarrollo exitoso de los sucesivos flujos de trabajo definidos por RUP, siendo estos: Diseño e Implementación.

Capítulo 3: Diseño del Sistema

Capítulo 3 Diseño del Sistema

El objetivo de este capítulo es traducir los requisitos funcionales y no funcionales a especificaciones del sistema, describiéndose cómo sería implementado y contribuyendo a obtener una arquitectura sólida y estable para la futura implementación del software.

El propósito del capítulo está orientado a adquirir una comprensión de los aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos, tecnologías de distribución, concurrencias y tecnologías de interfaz de usuario. Además de crear una entrada apropiada y un punto de partida para las actividades de implementación, capturando los requisitos, interfaces y clases.

En este capítulo serán expuestos diferentes artefactos tales como el Modelo de Diseño, especificándose la estructura y definición de los elementos que posee, además de los diagramas de clases y la descripción de las clases del diseño.

3.1 Modelo de Diseño

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso, centrándose en cómo los requisitos funcionales y no funcionales junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar. Sirve de abstracción de la implementación y es utilizado como entrada fundamental a las actividades de implementación. [42]

3.1.1 Estructura del diseño

Para lograr un mayor entendimiento del diseño del sistema es necesario definir su descomposición en paquetes de clases. Dicha definición es de suma importancia a la hora de implementar el mismo, pues divide de forma organizada la estructura interna de la futura aplicación. Debe mencionarse que el diseño del subsistema Nomenclador está basado en el patrón de arquitectura MVC incorporado por el framework Symfony (**Ver Anexo 2**), el cual separa la lógica de negocio (el modelo) y la presentación (la vista) mediante el controlador que es el encargado de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista. El siguiente diagrama refleja la estructura del diseño del módulo Nomencladores.

Capítulo 3: Diseño del Sistema

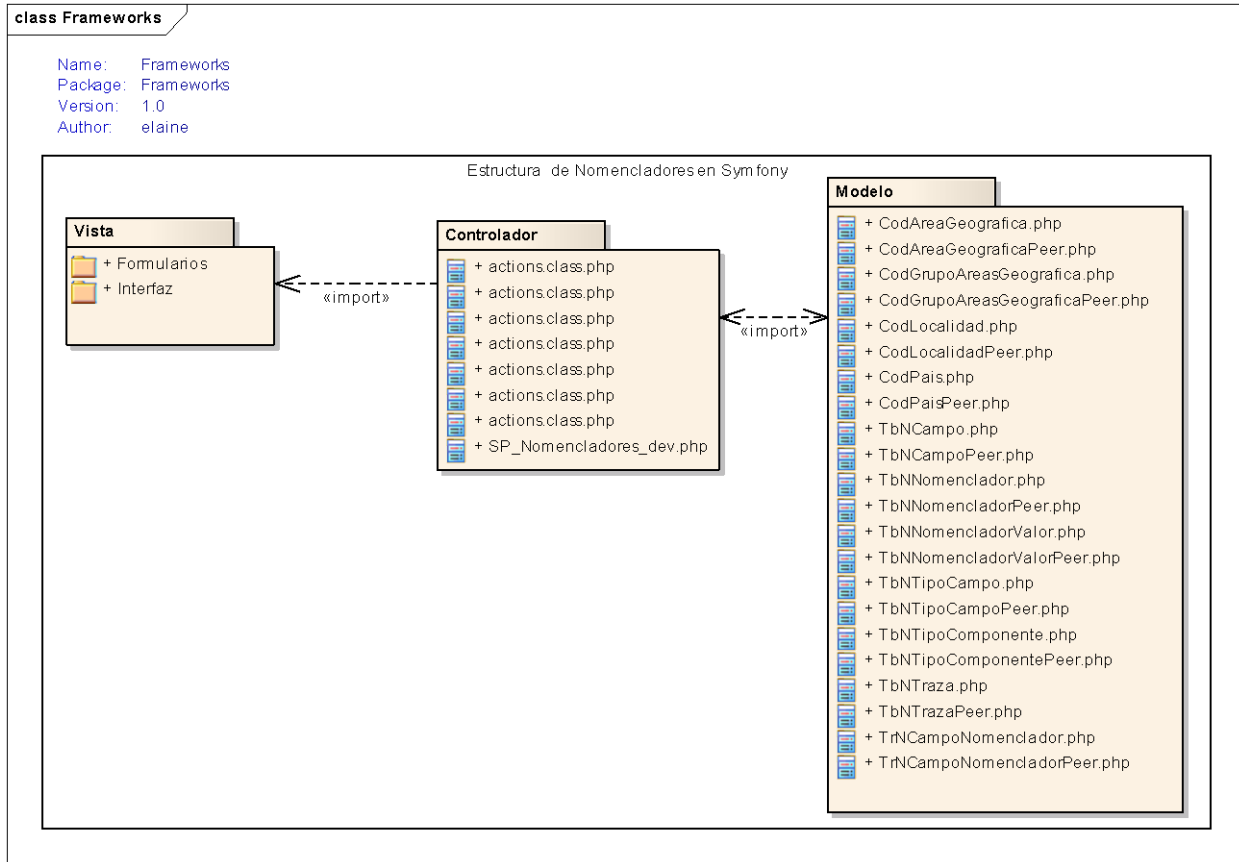


Fig. 3.1 Diagrama representativo del diseño del subsistema Nomenclador

3.1.2 Definición de elementos del diseño

El Módulo de Nomencladores constituye una aplicación web, por lo que se modelará utilizando las clases estereotipadas “Server Page”, “Client Page”, “Form” empleadas para el código servidor, código cliente y formularios respectivamente.

Clases de Diseño representadas con estereotipos web	
<p>Nombre_SP</p>	<p>Server Page: Representaría el código puramente servidor. El código servidor se encarga de construir o generar el resultado HTML que conforma el código cliente (build).</p>
	<p>Client Page: Representaría el código puramente cliente. Son interpretadas por el</p>

Capítulo 3: Diseño del Sistema

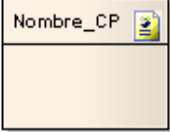
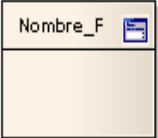
	<p>navegador. Sus atributos son las variables declaradas dentro del script, que son accesibles para cualquier función dentro de la página. Es importante destacar que una página cliente es construida por una sola página servidora, aunque podría ser un código HTML estático, caso en el cual, ningún código servidor la construye.</p>
	<p>Form: Representaría los formularios presentes en el código cliente. Los formularios envían sus datos al código servidor para ser procesados los pedidos (submit).</p>

Tabla 3.2 Clases del diseño estereotipadas

Relaciones entre Clases:

Hasta Desde	Client Page	Form	Server Page
Client Page	<<Link>> , <<redirect>>	Contiene	<<Link>> , <<redirect>>
Form	Agregado por.	---	<<Submit>>
Server Page	<<Build>> , <<redirect>>		<<Redirect>>

Tabla 3.3 Relaciones entre clases del diseño

2.1.3 Diagramas de clases de diseño

Los Diagramas de Clases del Diseño representan las clases y sus relaciones, simbolizan la parte estática del sistema. (Ver Anexo 3)

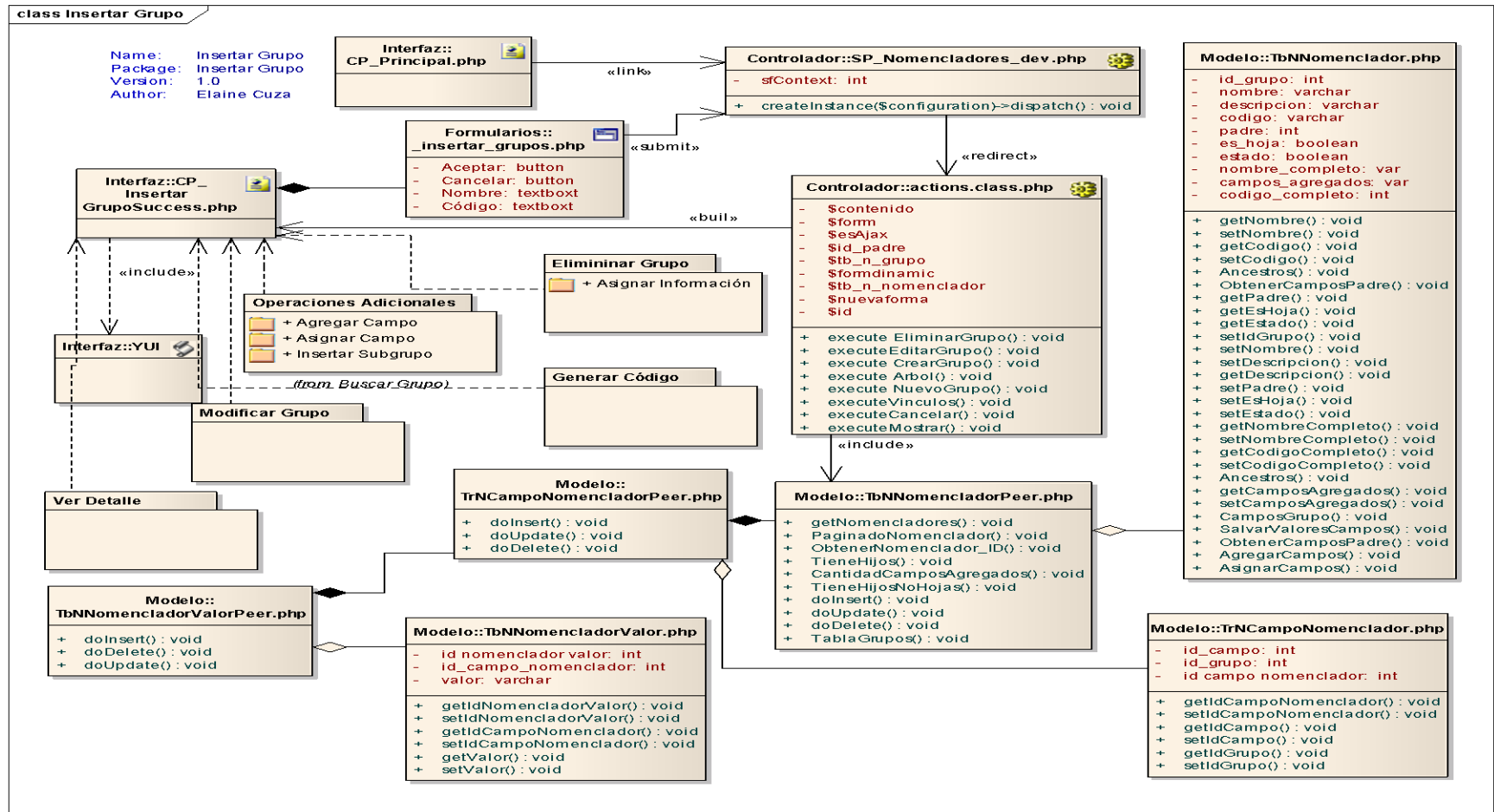


Fig. 3.4 Diagrama de clases del diseño CUS_Insertar Grupo

Capítulo 3: Diseño del Sistema

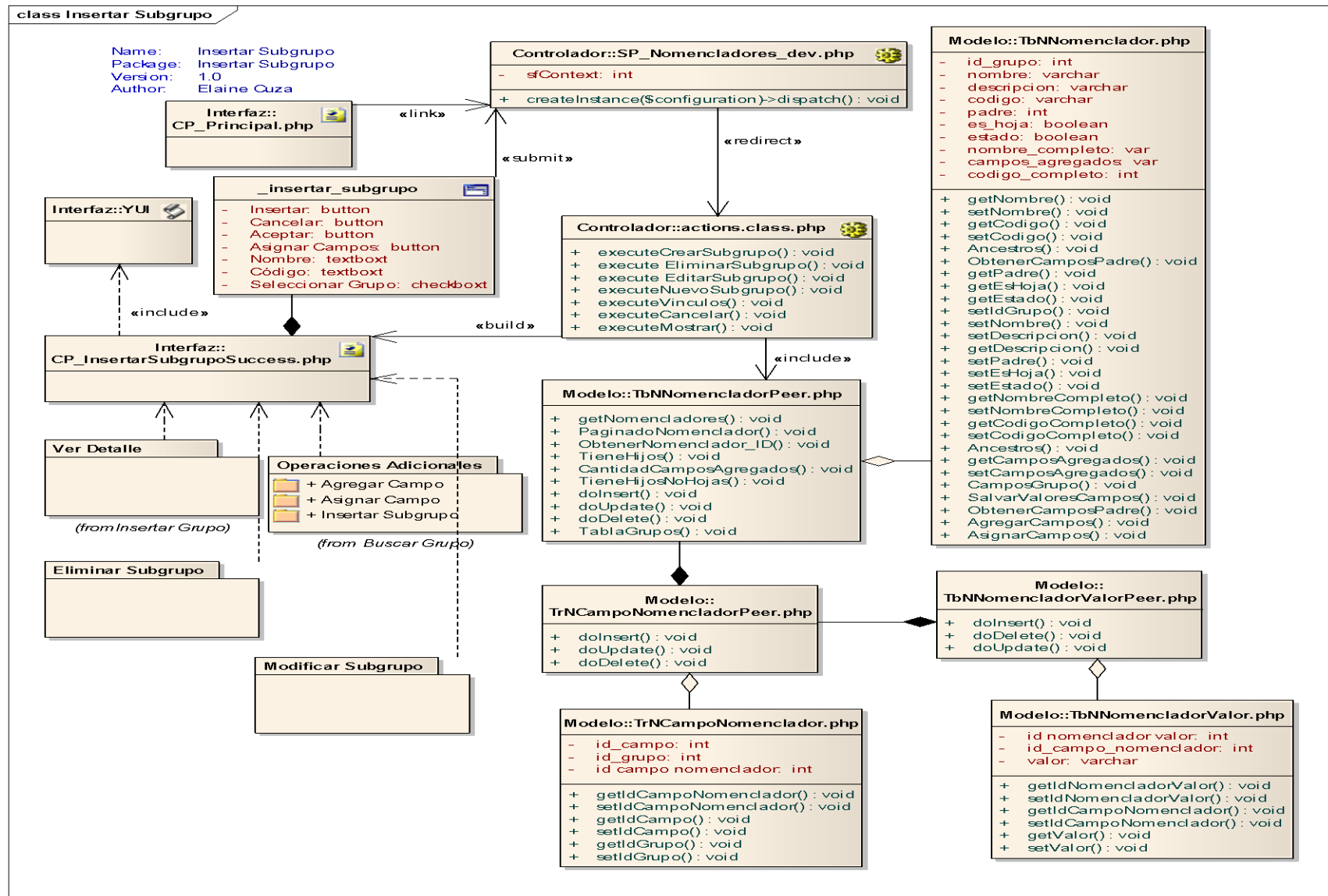


Fig. 3.5 Diagrama de clases del diseño CUS_Insertar Subgrupo

Capítulo 3: Diseño del Sistema

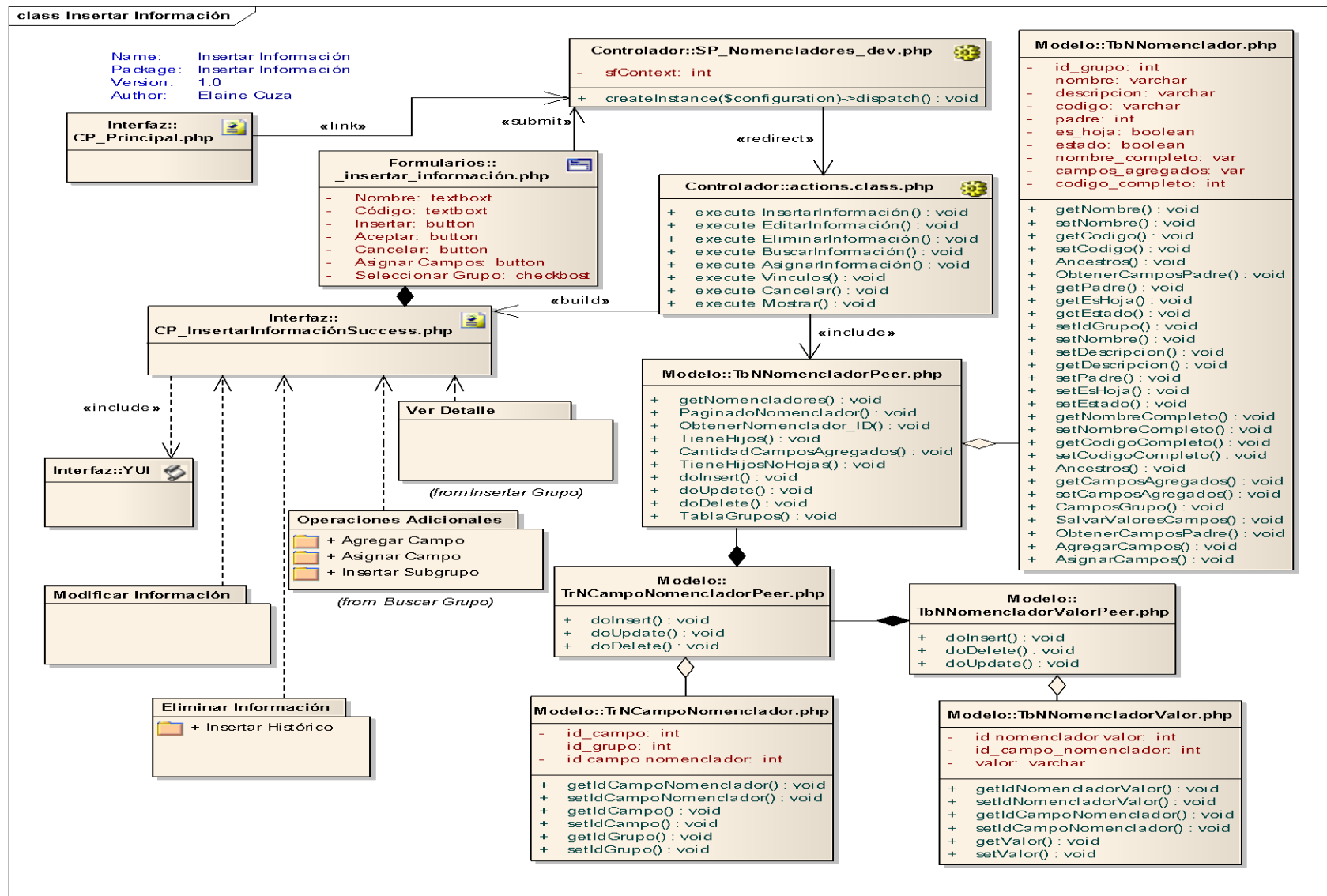


Fig. 3.6 Diagrama de clases del diseño CUS_Insertar Información

Capítulo 3: Diseño del Sistema

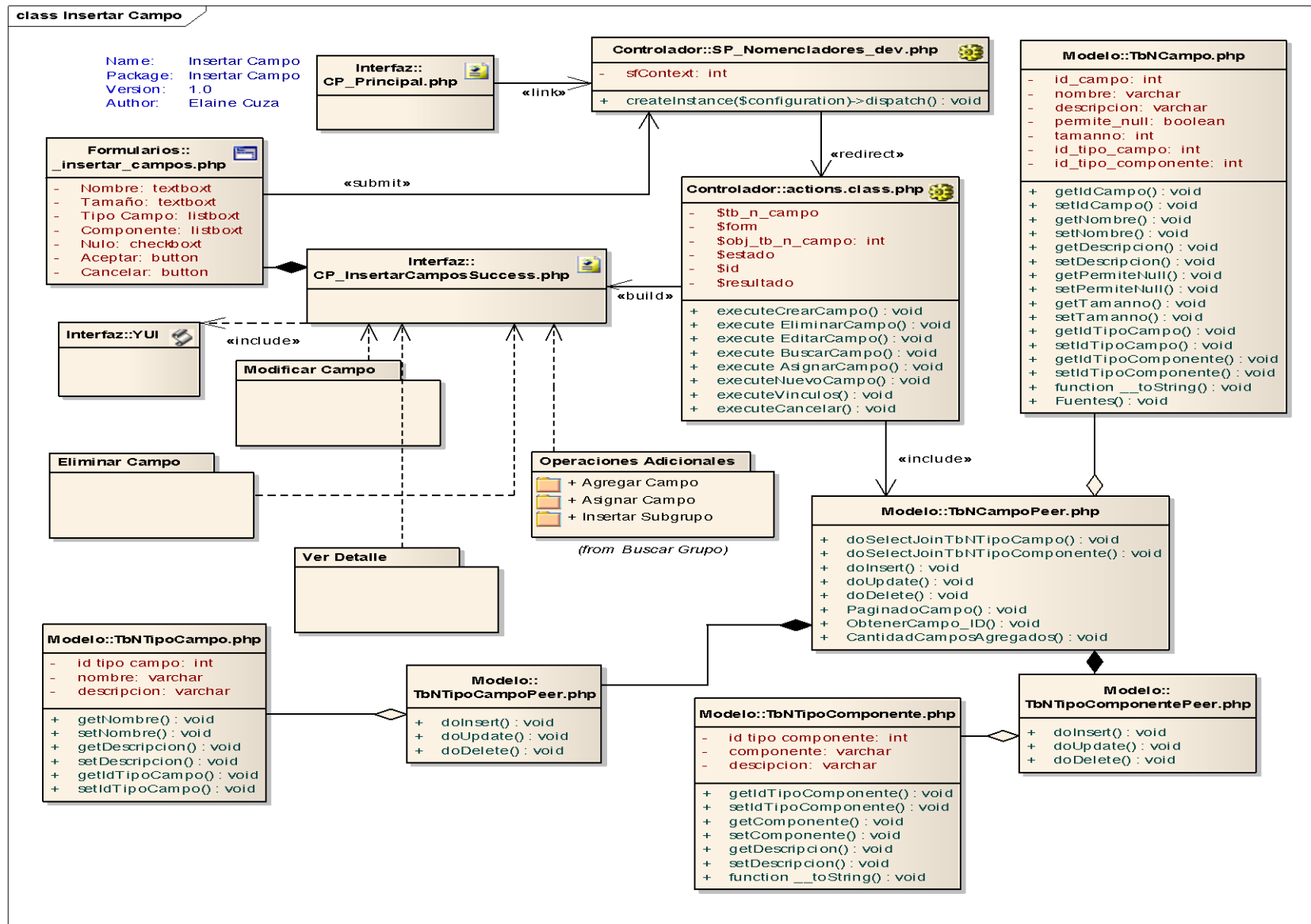


Fig. 3.7 Diagrama de clases del diseño CUS_Insertar Campos

Capítulo 3: Diseño del Sistema

3.1.4 Diagramas de Secuencia

Los diagramas de secuencia muestran las interacciones entre objetos, ordenadas en secuencias temporales durante un escenario en específico. Si los casos de uso tienen varios flujos o subflujos distintos, suele ser útil crear un diagrama de secuencia para cada uno de ellos. Los diagramas de secuencia son buenos para mostrar cuáles objetos se comunican con otros objetos y qué mensajes disparan esas comunicaciones.

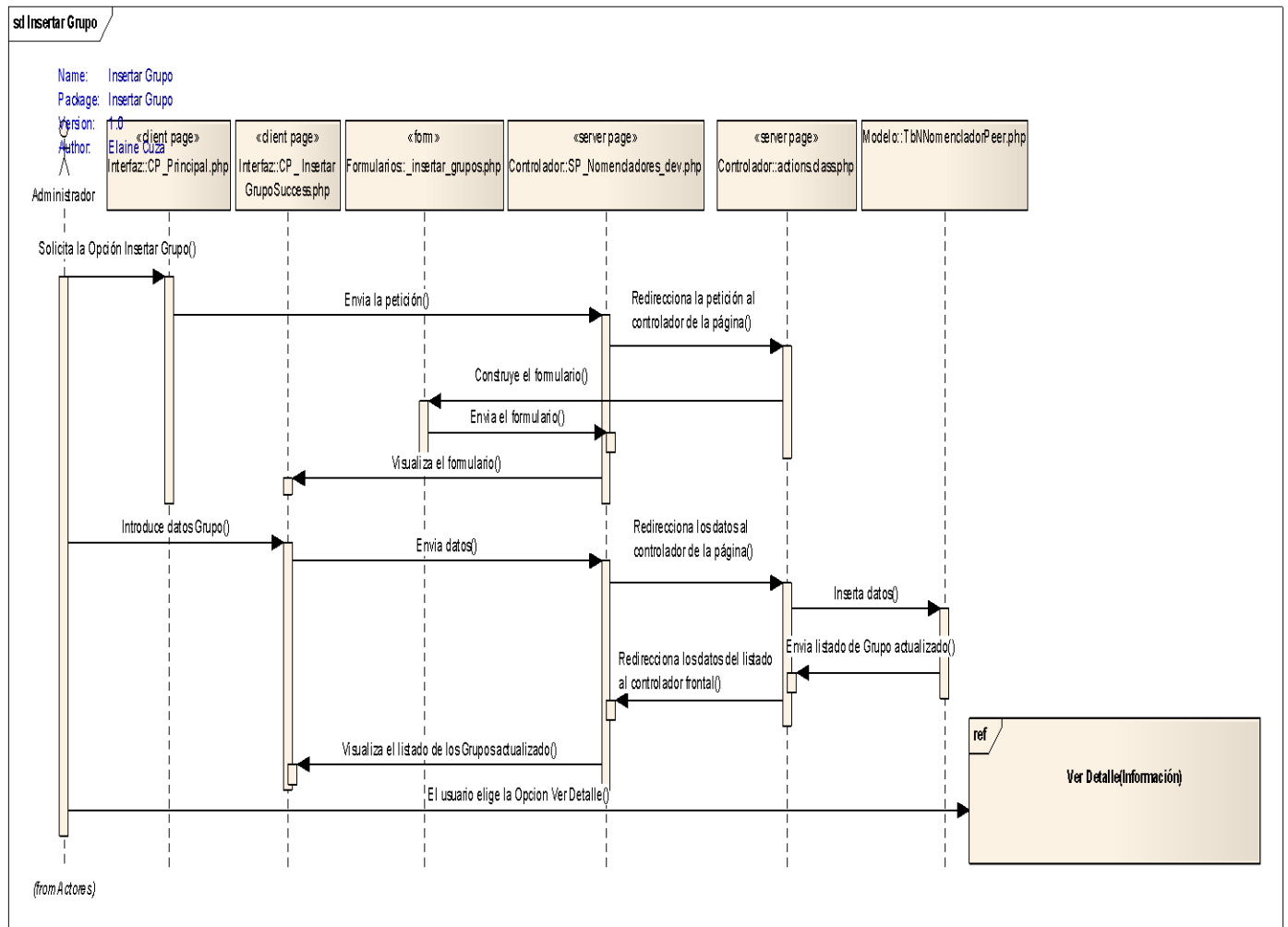


Fig. 3.8 Diagrama de secuencia CUS_Insertar Grupo

Capítulo 3: Diseño del Sistema

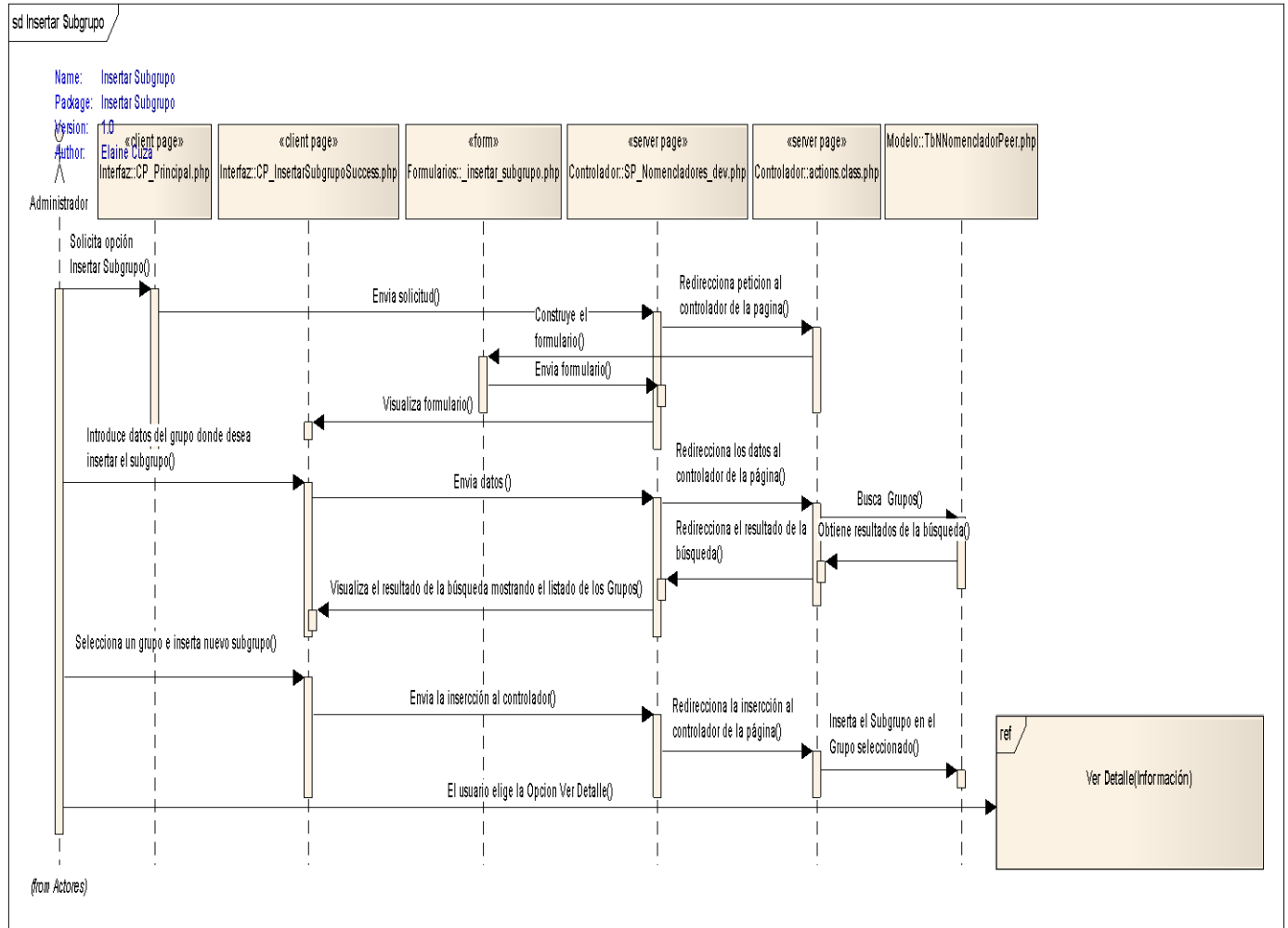


Fig. 3.9 Diagrama de secuencia CUS_Insertar Subgrupo (Ver Anexo 4).

3.1.5 Descripción de las clases

Consecutivamente serán expuestas algunas clases que han sido identificadas para la posterior implementación del sistema. De esta manera se poseerá un mejor entendimiento sobre el funcionamiento del mismo.

Descripción de las clases controladoras

Nombre: SP_Nomencladores_dev.php
Tipo: Server Page
Descripción: Esta clase es la encargada de redireccionar las peticiones del usuario a las clases actions.php.

Capítulo 3: Diseño del Sistema

Métodos	
Nombre	Descripción
createInstance(\$configuration)->dispatch	Este método permite realizar la configuración del sistema.
Variables	
Nombre	Descripción
sfContext	En esta variable almacena la configuración del sistema.
Relaciones: Esta clase mantiene una relación de tipo redirect con las clases actions.php del sistema.	

Tabla 3.10 Descripción de la clase SP_Nomencladores_dev.php

Nombre: actions.php	
Tipo: Server Page	
Descripción: Esta clase es la encargada de controlar las acciones realizadas sobre la entidad Grupo.	
Métodos	
Nombre	Descripción
function executeNuevogrupo	Este método permite mostrar al usuario una nueva interfaz para la creación de un nuevo grupo.
function executeCrearGrupo	Este método permite insertar un nuevo grupo y actualizar los datos del sistema.
function executeEditarGrupo	Este método permite modificar los datos de un grupo.
function executeEliminarGrupo	Este método permite eliminar un grupo.
function executeArbol	Este método permite la construcción del árbol de información.
function executeVinculos	Este método contiene los diferentes vínculos a los que se puede acceder a partir de esta clase.
function executeCancelar	Este método permite cancelar una acción que se realice sobre la entidad grupo.

Capítulo 3: Diseño del Sistema

function executeMostrarNomencladores	Este método permite mostrar los grupos que existen en el sistema.
function executeGrupoCampos	Este método permite agregar o asignar campos a un grupo
Variables	
Nombre	Descripción
\$contenido	Esta variable hace referencia a los nodos que contendrá el nuevo árbol de información.
\$esAjax	Esta variable hace referencia al objeto creado por Ajax para realizar las peticiones del usuario.
\$form	Esta variable representa un objeto de una forma.
\$id_padre	Esta variable representaría una variable global, a través de la cual se puede acceder al nodo padre de un grupo.
\$tb_n_grupo	Representa un objeto de tipo grupo.
\$formdinamic	Representa un objeto de una forma dinámica.
\$tb_n_nomenclador	Representa un objeto de tipo grupo.
\$nuevaforma	Esta variable representa un objeto de una forma.
\$id	Representa el id de un grupo.
Relaciones: Esta clase mantiene una relación de tipo redirect con la clase SP_Nomencladores_dev.php que constituye el controlador	

Tabla 3.11 Descripción de la clase actions.php relacionada a la entidad grupo.

Nombre: actions.php	
Tipo: Server Page	
Descripción: Esta clase es la encargada de controlar las acciones realizadas sobre la entidad Campo	
Métodos	
Nombre	Descripción
function executeNuevoCampo	Este método permite mostrar al usuario una

Capítulo 3: Diseño del Sistema

	nueva interfaz para la creación de un nuevo campo.
function executeCrearCampo	Este método permite insertar un nuevo campo y actualizar los datos del sistema.
function executeEditarcampo	Este método permite modificarlos los datos de un campo.
function executeEliminarcampo	Este método permite eliminar un campo.
function executeBuscarCampos	Este método permite la búsqueda de la entidad campo.
function executeVinculos	Este método contiene los diferentes vínculos a los que se puede acceder a partir de esta clase.
function executeCancelar	Este método permite cancelar una acción que se realice sobre la entidad campo.
function executeMostrar	Este método permite mostrar los campos.
function PaginadoCampo	Este método permite realizar el paginado de los campos.
function executeAsignarCampos	Este método permite agregar y asignar campos.
Variables	
Nombre	Descripción
\$tb_n_campo	Esta variable representa un objeto de tipo campo.
\$form	Esta variable representa un objeto de tipo formulario.
\$obj_tb_n_campo	En esta variable se obtiene un objeto de tipo campo mediante un id.
\$estado	Esta variable representa el posible estado de los campos.
\$id	Permite conocer el id del grupo al cual se le agregará o asignará campo.
\$resultado	Guarda los campos.
Relaciones: Esta clase mantiene una relación de tipo redirect con la clase	

Capítulo 3: Diseño del Sistema

SP_Nomencladores_dev.php que constituye el controlador frontal de la aplicación.

Tabla 3.12 Descripción de la clase actions.php relacionada a la entidad campo.

Nombre: actions.php	
Tipo: Server Page	
Descripción: Esta clase es la encargada de mostrar los detalles de las entidades del nomenclador.	
Métodos	
Nombre	Descripción
function executeVerDetalleGrupo	Este método permite ver los detalles de las entidades grupos, subgrupos e información.
function executeVerDetalleCampo	Este método permite ver los detalles de la entidad campo.
Variables	
Nombre	Descripción
\$tb_n_grupo	Esta variable representa un objeto de tipo grupo.
\$hijos	Esta variable representa los hijos que pueden tener un grupo o subgrupo.
\$cant_hijos	Esta variable representa la cantidad de hijos que puede tener un grupo.
\$campos_agreg	Esta variable representa los campos agregados o asignados.
\$cant_campos_agreg	Esta variable representa la cantidad de campos agregados o asignados.
\$tb_n_campo	Esta variable representa un objeto de tipo

Capítulo 3: Diseño del Sistema

	campo.
\$grupos_agreg	Esta variable representa los grupos agregados del sistema.
\$cant_grupos_agreg	Esta variable representa la cantidad de grupos agregados del sistema.
Relaciones: Esta clase mantiene una relación de tipo redirect con la clase SP_Nomencladores_dev.php que constituye el controlador frontal de la aplicación.	

Tabla 3.13 Descripción de la clase actions.php relacionada a los ver detalles de las distintas entidades del sistema.

Descripción de las clases del modelo

Nombre: TbNNomencladorPeer.php	
Tipo: Acceso a datos	
Descripción: En esta clase se representa la parte de acceso de datos de la tabla tb_n_nomenclador, la misma constituye una clase hija de la BaseTbNNomenclador, heredando funcionalidades tales como los insert, delete, update entre otras.	
Métodos	
Nombre	Descripción
function getNomencladores	Este método devuelve los valores de la tabla. tb_n_nomenclador.
function getNomencladores	Este método permite obtener la estructura del paginado de la tabla tb_n_nomenclador.
function ObtenerNomenclador_ID	Este método permite obtener el id de la estructura del nomenclador deseado.
function TieneHijos	Este método permite conocer los hijos de una de las estructuras del nomenclador.
function CantidadCamposAgregados	Este método permite conocer la cantidad de campos agregados o asignados de una de

Capítulo 3: Diseño del Sistema

	las estructuras creadas en la tabla tb_n_nomenclador.
function TieneHijosNoHojas	Este método permite conocer si una de las estructuras del nomenclador tiene hijo y si no es hoja.
function TablaGrupos	Este método permite obtener el listado de los grupos existentes en el sistema.
function doInsert	Este método permite insertar una nueva estructura en la tabla tb_n_nomenclador.(Esta función es heredada de la clase base)
function doDelete	Este método permite eliminar una de las estructuras en la tabla tb_n_nomenclador.(Esta función es heredada de la clase base)
function doUpdate	Este método permite actualizar los datos de una nueva estructura en la tabla tb_n_nomenclador.(Esta función es heredada de la clase base)
Relaciones: Esta clase mantiene una relación de tipo incluida con las actions.php que hacen referencia a la tabla tb_n_nomenclador.	

Tabla 3.14 Descripción de la clase TbNNomencladorPeer.php

Nombre: TbNNomenclador.php	
Tipo: Acceso a datos	
Descripción: Esta clase representa la parte de acceso a datos de la tabla de la BD tb_n_nomenclador constituye un clase hija de BaseTbNNomenclador, heredando sus atributos y métodos	
Atributos	
Nombre	Descripción

Capítulo 3: Diseño del Sistema

var \$campos_agregados	Este atributo representa en el sistema los campos agregados.
var \$nombre_completo	Este atributo representa el nombre de los mismos.
var \$codigo_completo	Este atributo representaría el código que identificaría este campo.
Métodos	
function getNombreCompleto	Este método devuelve el nombre de la estructura solicitada de la tabla tb_n_nomenclador.
function setNombreCompleto	Este método permite cambiar el nombre de la estructura de la tabla seleccionada.
function getCodigoCompleto	Este método devuelve el código de la estructura solicitada de la tabla tb_n_nomenclador.
function setCodigoCompleto	Este método permite cambiar el código de la estructura de la tabla seleccionada.
public function getCamposAgregados	Este método devuelve los valores de los campos agregados.
function CamposGrupo	Este método devuelve los campos de cierta estructura de la tabla tb_n_nomenclador.
function SalvarValoresCampos	Este método permite salvar los valores de los campos de una de las estructuras de la tabla tb_n_nomenclador.
function AgregarCampos	Este método permite agregar campos a una de las estructuras de la tabla tb_n_nomenclador.
function AsignarCampos	Este método permite asignar campos a una de las estructuras de la tabla tb_n_nomenclador.
function __toString	Este método devuelve el nombre de un

Capítulo 3: Diseño del Sistema

	grupo.
Relaciones: Esta clase mantiene una relación de agregación con la clase TbNNomencladorPeer.	

Tabla 3.15 Descripción de la clase TbNNomenclador.php

Nombre: TbNCampoPeer.php	
Tipo: Acceso a datos	
En esta clase se representa la parte de acceso de datos de la tabla tb_n_campo. La misma representa una clase hija de la BaseTbNCampo heredando funcionalidades tales como los insert, delete, update entre otras.	
Métodos	
Nombre	Descripción
function PaginadoCampo	Este método permite obtener el paginado de los campos del sistema.
function ObtenerCampo_ID	Este método permite obtener un campo dado un id.
function CantidadCamposAgregados	Este método permite obtener la cantidad de campos del sistema.
function doInsert	Este método permite insertar un nuevo campo en la tabla tb_n_campo.(Esta función es heredada de la clase base)
function doDelete	Este método permite eliminar un campo en la tabla tb_n_campo.(Esta función es heredada de la clase base)
function doUpdate	Este método permite actualizar los datos de uno de los campos de la tabla tb_n_campo.(Esta función es heredada de la clase base)
Relaciones: Esta clase mantiene una relación de tipo incluida con las actions.php que hacen	

Capítulo 3: Diseño del Sistema

referencia a la tabla tb_n_campo.

Tabla 3.16 Descripción de la clase TbNCampoPeer.php

Nombre: TbNCampo.php	
Tipo: Acceso a datos	
Descripción: Esta clase representa la parte de acceso a datos de la tabla tb_n_nomenclador, constituye la clase hija de BaseTbCampo, heredando sus atributos y métodos.	
Métodos	
function Fuentes	Este método devuelve el tipo de componente con que puede contar el campo.
function __to String	Este método devuelve el nombre de un campo.
Relaciones: Esta clase mantiene una relación de agregación con la clase TbNCampoPeer.	

Tabla 3.17 Descripción de la clase TbNCampo.php

3.2 Modelo de Datos

Describe la representación lógica y física de los datos persistentes. Es el artefacto resultante de la actividad de diseño de base de datos.

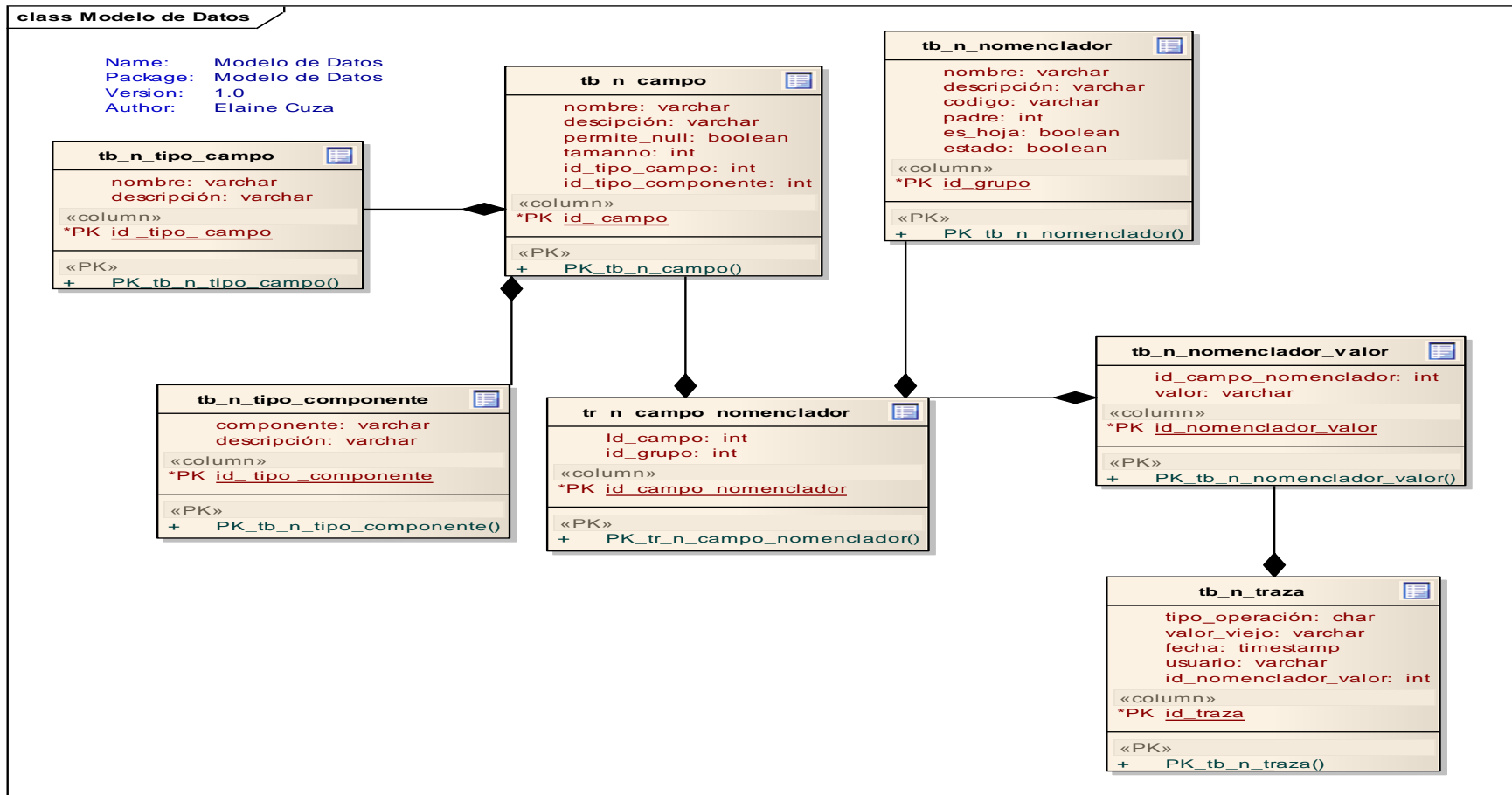


Fig. 3.18 Modelo de datos

Capítulo 3: Diseño del Sistema

3.2.1 Descripción de las tablas de la base de datos del sistema

Nombre: tb_n_campo	
Descripción: Almacena información que permite la representación de los campos como componentes de la interfaz gráfica de usuario (IGU), ejemplo: el campo Nombre se le asocia para representarlo (textbox), el tipo de dato del campo (string).	
Atributo	Tipo
id_campo	integer
nombre	character(variable)
descripcion	character(variable)
permite_null	boolean
tamaño	integer
id_tipo_componente	integer
id_tipo_campo	integer
id_campo	integer

Tabla 3.19 Descripción de la tabla tb_n_campo

Nombre: tb_n_nomenclador	
Descripción: Almacena información referente a los grupos de elementos, subgrupo de elementos y los elementos. El objetivo de esta tabla es nomenciar estructuras dinámicas del negocio.	
Atributo	Tipo
id_grupo	integer
nombre	character(variable)
descripcion	character(variable)
codigo	character(variable)
padre	Integer
es_hoja	Boolean
estado	Boolean

Tabla 3.20 Descripción de la tabla tb_n_nomenclador

Nombre: tr_n_campo_nomenclador

Capítulo 3: Diseño del Sistema

Descripción: Almacena información referente a los campos que se encuentran en tb_n_campo que pueden describir a un grupo, subgrupo o elemento, definido en la tabla tb_n_nomenclador.	
Atributo	Tipo
id_campo_nomenclador	integer
Id_grupo	integer
id_campo	integer

Tabla 3.21 Descripción de la tabla tr_n_campo_nomenclador

Nombre: tb_n_nomenclador_valor	
Descripción: Almacena información referente a los valores del campo en un grupo descrito en la tabla tr_n_campo_nomenclador.	
Atributo	Tipo
id_nomenclador_valor	integer
id_campo_nomenclador	integer
valor	character(variable)

Tabla 3.22 Descripción de la tabla tb_n_nomenclador_valor

Nombre: tb_n_traza	
Descripción: Almacena información referente a los valores históricos tomados por un campo que pertenezca a un grupo, subgrupo o material. Esto se logra por la relación que existe con la tabla tb_n_nomenclador_valor, y la implementación de un disparador (trigger) que es el encargado de insertar los datos en la tabla.	
Atributo	Tipo
id_traza	integer
tipo_operacion	character(1)
valor_viejo	character(variable)
valor_nuevo	character(variable)
fecha	timestamp without time zone
usuario	character(variable)
id_gsm_valor	integer
id_traza	integer

Capítulo 3: Diseño del Sistema

Tabla 3.23 Descripción de la tabla tb_n_traza

Nombre: tb_n_tipo_campo	
Descripción: Almacena información referente a los tipos de datos posible que pueda tomar un campo.	
Atributo	Tipo
id_tipo_campo	integer
nombre	character(variable)
descripcion	character(variable)

Tabla 3.24 Descripción de la tabla tb_n_tipo_campo

Nombre: tb_n_tipo_componente	
Descripción: Almacena información referente a los tipos de componentes que se pueden utilizar para representar los campos en la IGU.	
Atributo	Tipo
id_tipo_componente	integer
componente	character(variable)
descripcion	character(variable)

Tabla 3.25 Descripción de la tabla tb_n_tipo_componente

Conclusiones

Con la realización de este capítulo fue obtenido el Modelo de Diseño del sistema Nomenclador, obteniéndose todos los artefactos definidos para el desarrollo del flujo: los Diagramas de Clases del Diseño, los Diagramas de Secuencias y el Modelo de Datos. Además, se representó la realización física de los casos de uso, materializándose los requerimientos planteados para el sistema y brindando un mayor entendimiento del sistema que se quiere desarrollar.

Capítulo 4: Implementación

Capítulo 4 Implementación

El presente capítulo tiene como objetivo fundamental detallar la arquitectura definida durante el diseño. Se implementan las clases definidas en el capítulo anterior en término de componente. Además, se obtienen los Diagramas de Componente y de Despliegue como parte del Modelo de Implementación del sistema, proporcionando una visión de cómo quedará construida y distribuida la aplicación.

4.1 Modelo de Implementación

El Modelo de Implementación es comprendido por un conjunto de componentes y subsistemas que constituyen la composición física de la implementación del sistema. Entre los componentes se encuentran: datos, archivos, ejecutables, código fuente y los directorios. Fundamentalmente, se describe la relación que existe desde los paquetes y clases del modelo de diseño a subsistemas y componentes físicos. [43]

4.1.1 Diagrama de Componentes

Un diagrama de componentes representa como un sistema es dividido en componentes y muestra las dependencias entre estos. Los Diagramas de Componentes pueden ser usados para modelar y documentar la arquitectura del sistema y son utilizados en la construcción del mismo. Uno de los usos principales es que puede servir para ver qué componentes pueden compartirse entre sistemas o entre diferentes partes de un sistema.

El diagrama que se muestra a continuación, tiene como objetivo figurar la estructura y organización general del Módulo de Nomencladores, en términos de componentes:

Capítulo 4: Implementación

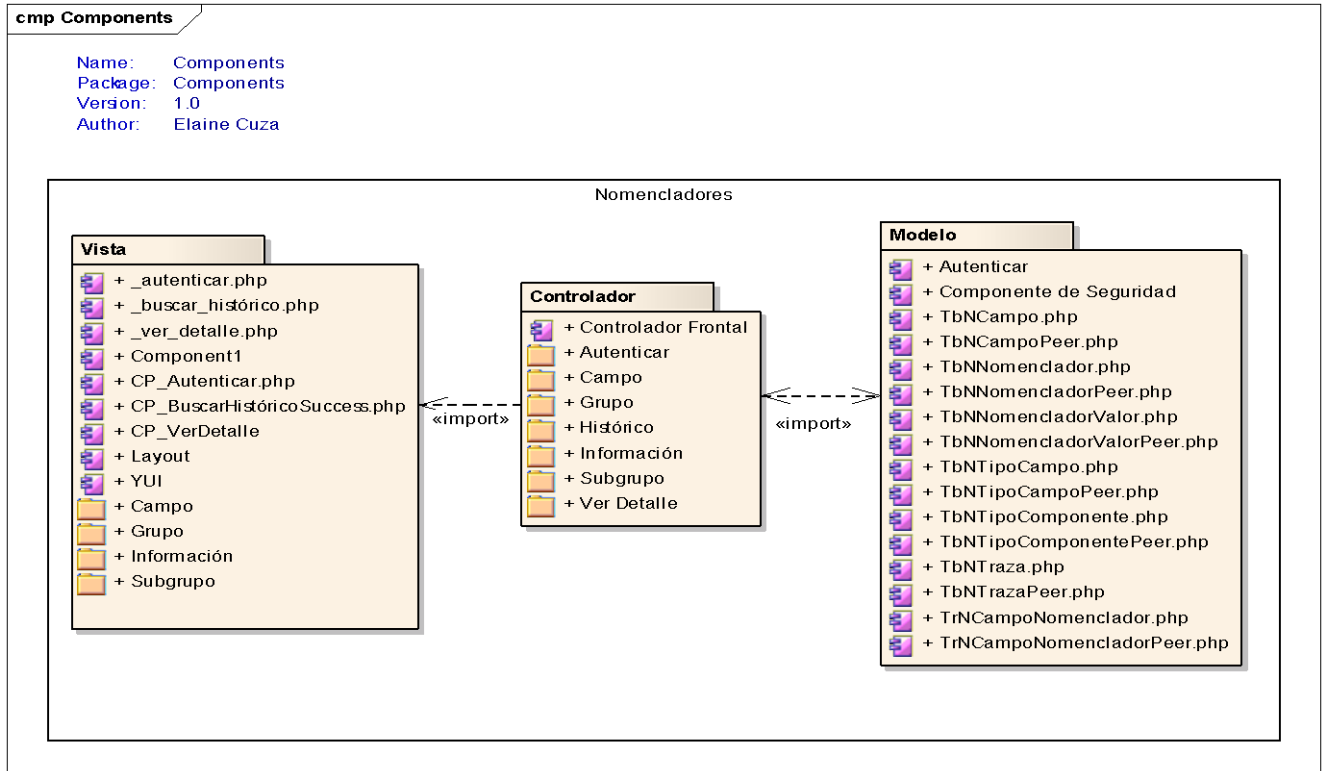


Fig. 4.1 Organización en paquetes de componentes

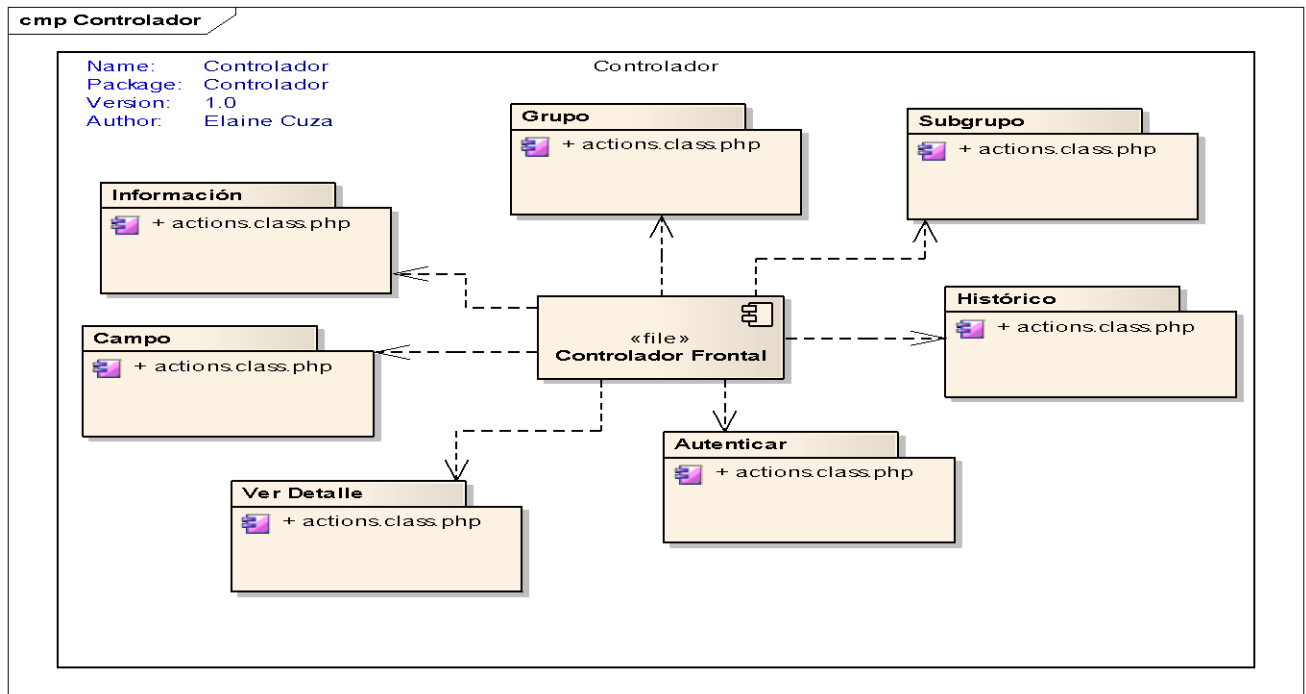


Fig. 4.2 Diagrama de componente paquete controlador

Capítulo 4: Implementación

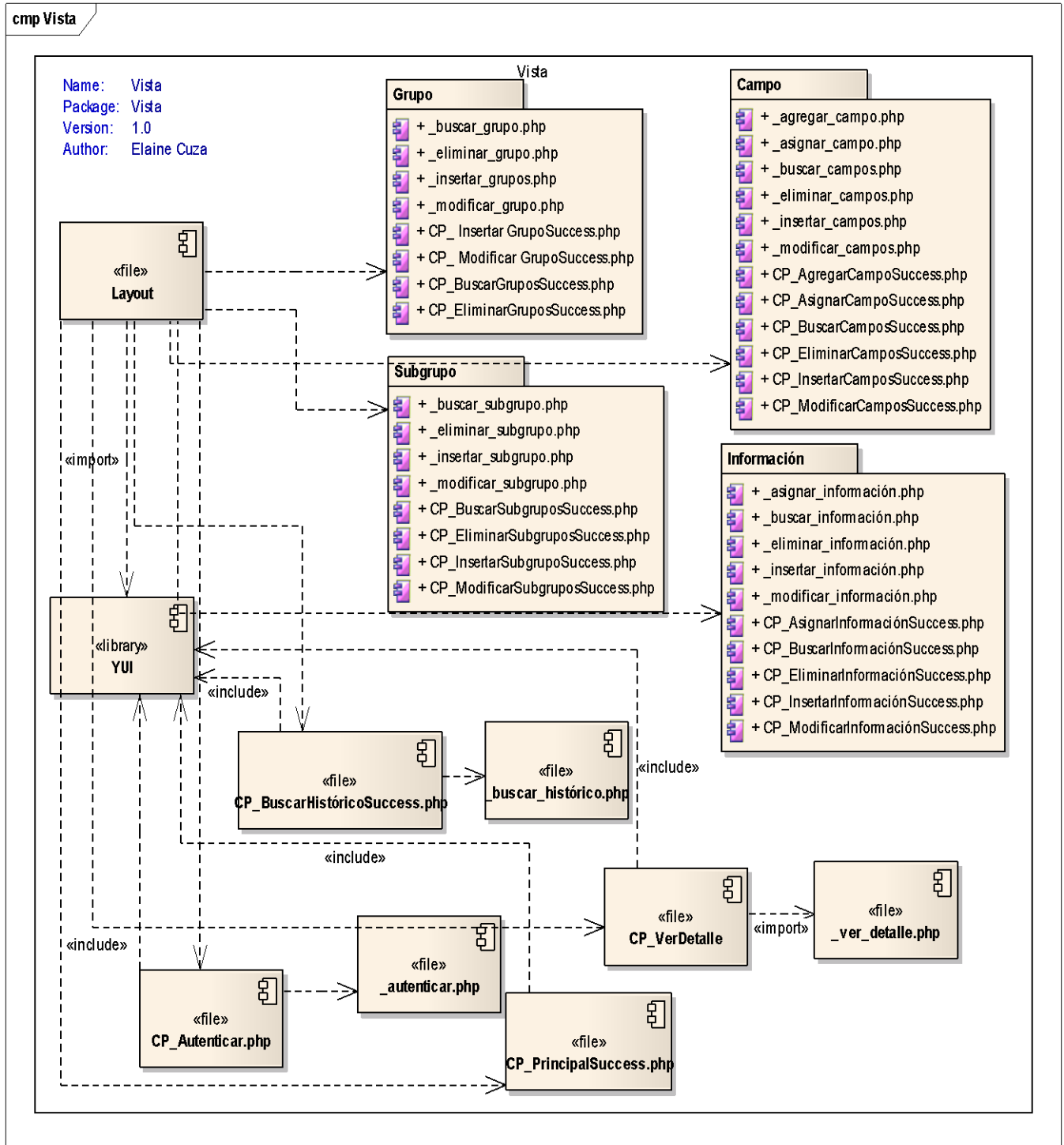


Fig. 4.3 Diagrama de componente paquete vista

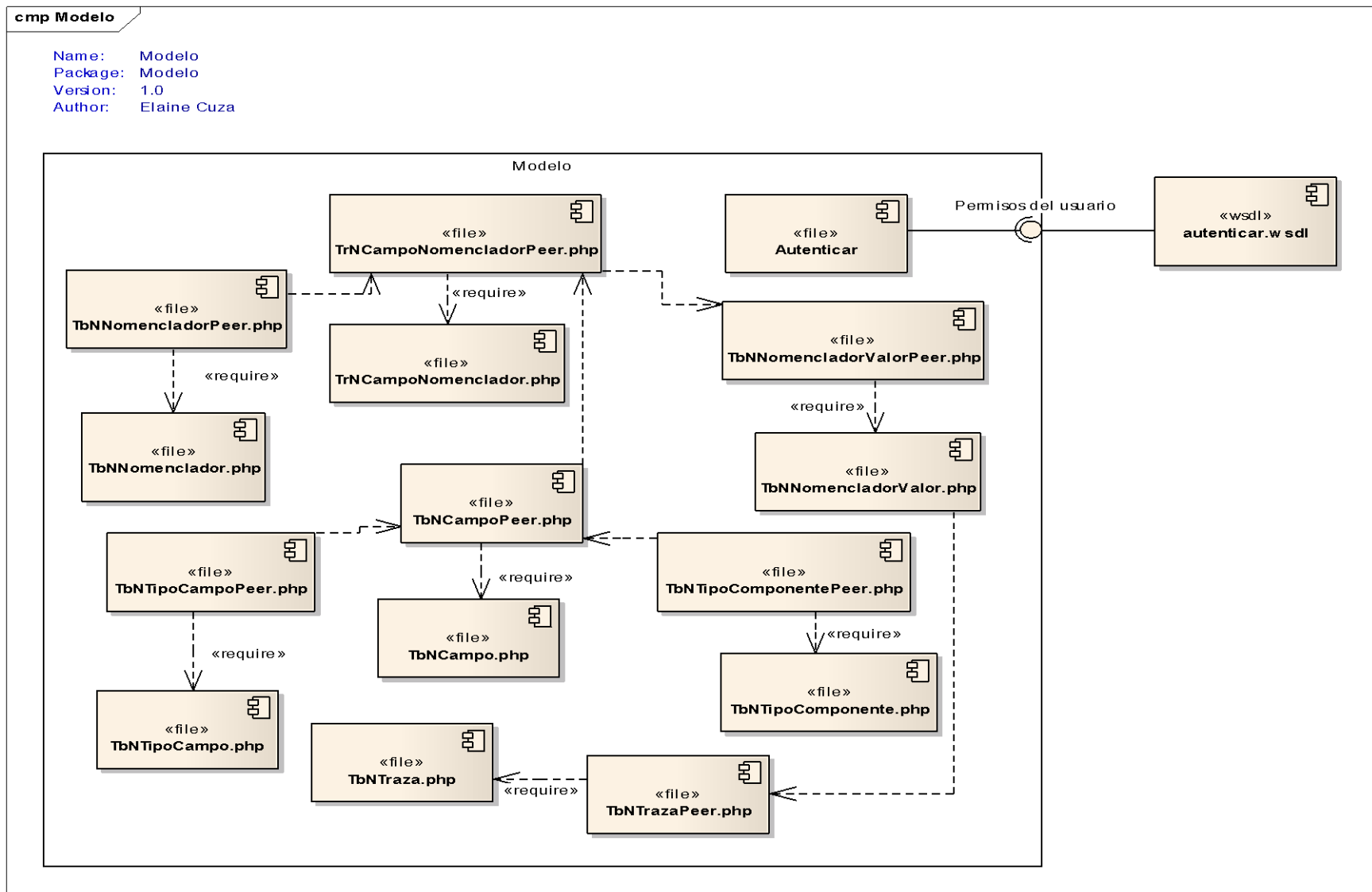


Fig. 4.4 Diagrama de componente paquete modelo

Capítulo 4: Implementación

El módulo de Nomencladores se encuentra estructurado por tres paquetes: el paquete vista, el paquete controlador y el paquete modelo. Estos representan la separación lógica del sistema según el MVC, patrón clásico del diseño web e incorporado por el framework Symfony para el desarrollo de aplicaciones.

- En el paquete vista son implementados los componentes que permiten al usuario interactuar con la aplicación. Para ello cuenta con las páginas clientes y la librería YUI, la cual es utilizada para crear aplicaciones web dinámicas complejas.
- En el paquete controlador se encuentran implementados los componentes encargados de procesar las interacciones del usuario con la aplicación y los componentes encargados de realizar los cambios apropiados en el modelo o en la vista. El controlador frontal ofrece un punto único de entrada y salida para toda la aplicación, además se encarga de gestionar las peticiones realizadas en la aplicación. Otro componente que forma parte de este paquete Controlador son las acciones, clases que incluyen el código específico del controlador de cada página.
- El paquete modelo está constituido por las clases Peer quienes contienen todo el código relacionado con el manejo de los datos del sistema, además se pueden encontrar las Clases.php que representarían la tabla de la base de datos sobre la cual se trabajaría.

4.2 Diagrama de Despliegue

El Diagrama de Despliegue es utilizado para modelar el hardware empleado en las implementaciones de sistemas y las relaciones entre sus componentes. El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos del cómputo. Cada nodo representa un recurso; en el caso del despliegue del módulo Nomenclador se recomienda contar con tres nodos procesadores:

- Nodo PC Cliente a través del cual los clientes se conectarán al sistema y podrán operar el mismo. Esta PC deberá contar con el sistema operativo Windows XP o GNU/Linux además del navegador Mozilla Firefox 3.0 que le permitirá al cliente tener acceso a la aplicación, además será necesario que este nodo cuente con un procesador Intel(R) Pentium 4 CPU 3.00GHz, con memoria 512 MB y 1GB de disco duro como mínimo.
- Nodo Servidor Aplicación: El mismo deberá contar con un procesador Intel(R) Pentium 4 CPU 3.00GHz, con memoria 1GB y 40 GB de disco duro como mínimo. Además, se

Capítulo 4: Implementación

necesitará contar con el software base GNU/Linux Distribución Debian 4 Etch o Ubuntu 8.04, el servidor de aplicaciones Apache2.2, Symfony 1.2 y el lenguaje de programación PHP 5.

- **Nodo Servidor Base de Datos:** Este servidor es necesario que cuente con procesador Intel(R) Pentium 4 CPU 3.00GHz, con memoria 1GB y 80 GB de disco duro como mínimo. Deberá utilizar como software base GNU/Linux Distribución Debian 4 Etch o Ubuntu 8.04 y como Sistema Gestor de Base de Datos PostgreSQL 8.3.

Capítulo 4: Implementación

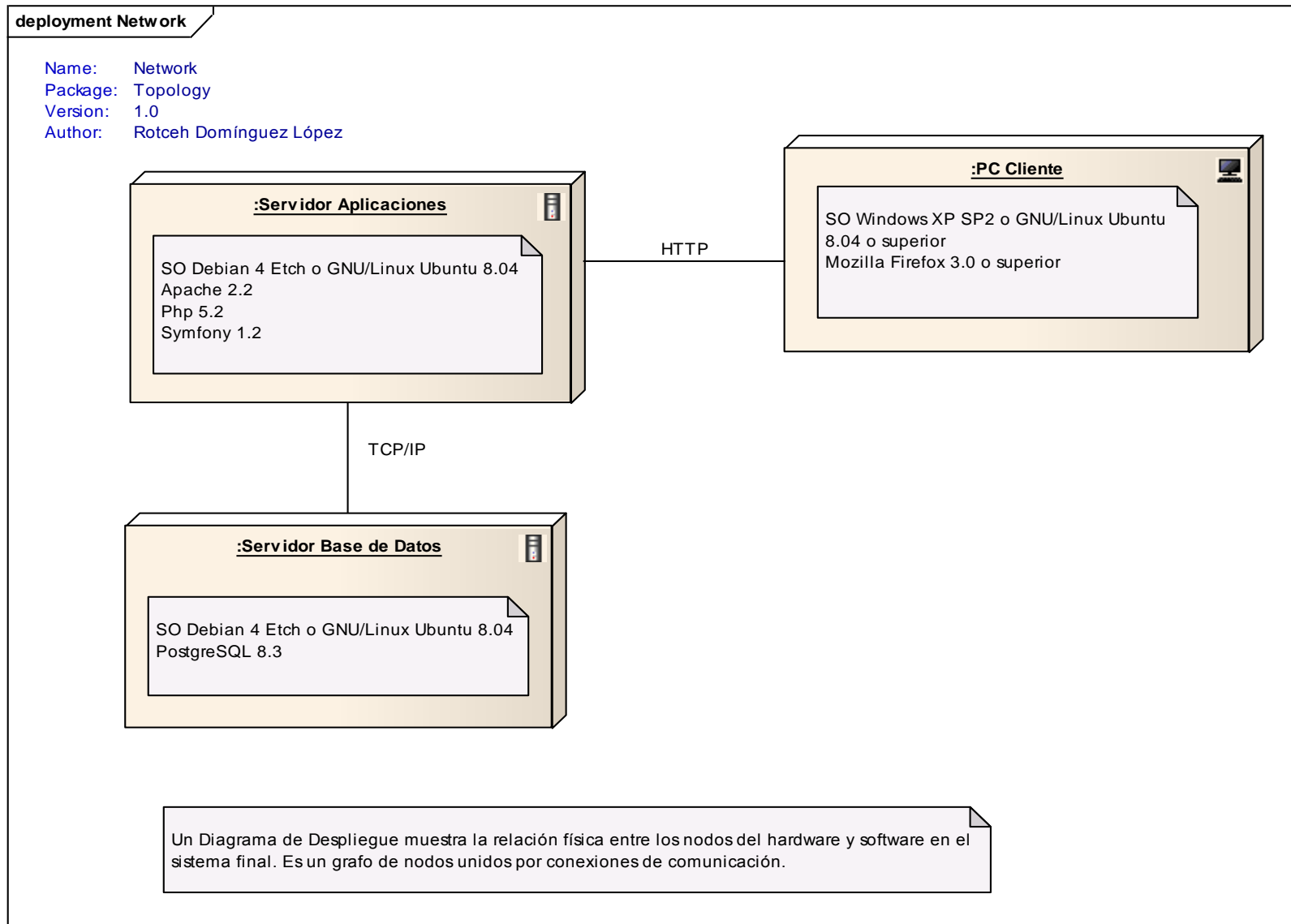


Fig. 4.5 Diagrama de despliegue

Capítulo 4: Implementación

Conclusiones

Como resultado del desarrollo del capítulo antes expuesto fue logrado el Modelo de Implementación del sistema Nomenclador. Donde fueron mostrados los Diagramas de Componentes y Despliegue, que representan cómo construir y distribuir el sistema, permitiendo la obtención de un producto que cumple con las funcionalidades propuestas.

Conclusiones

En la presente investigación se cumplió con el objetivo y las tareas propuestas. Se obtuvieron las siguientes conclusiones:

- Existe un gran número de sistemas nomencladores, pero ninguno posee la característica de ser configurable y flexible ante la posible gestión de otro tipo de información que no esté contemplada en su negocio.
- Existen múltiples herramientas que pudieran dar solución a la problemática planteada, sin embargo el equipo decidió acoger las herramientas propuestas por el grupo de arquitectura de AT-SAS pues las mismas son consideradas en su mayoría herramientas y tecnologías libres y multiplataforma.
- Como resultado del modelado de los flujos de negocio, requerimientos, diseño e implementación se obtuvo el diseño de un software que cumple con los requisitos definidos para la construcción de la aplicación.
- Se generaron todos los artefactos relacionados con el análisis y diseño definidos por la metodología de desarrollo RUP, los mismos constituyeron la base para la implementación del sistema.
- Con el desarrollo de la investigación se obtuvo la primera versión del sistema al **BAPNomenclador**. El cual constituye un mecanismo de gestión y almacenamiento de información común y poco variable, que posee como características la configurabilidad y flexibilidad ante posibles cambios.

Recomendaciones

El equipo de desarrollo recomienda incorporar nuevas funcionalidades al sistemas, tales como:

- Al eliminar un grupo o subgrupo, el sistema debe ser capaz de asignar la información embebida en estos a otra estructura del nomenclador.
- Generar código dinámicamente.
- Gestionar la información histórica de las estructuras del sistema.
- Establecer la comunicación entre posibles nomencladores creados.
- Crear reportes de cada una de las estructuras del nomenclador.
- Implementar los casos de uso identificados por el equipo de desarrollo y que fueron documentados.

Referencias Bibliográficas

Referencias Bibliográficas

1. voltairenet.org.Red de Prensa No Alineados. [Online] [Cited: 11 5, 2008.] <http://www.voltairenet.org/article129780.html>.
2. Discurso pronunciado por el Presidente de la República de Cuba, Fidel Castro Ruz, en el acto de inauguración de las obras del extraordinario programa de salud ya en marcha, que se lleva a cabo en Cuba. [Online] [Cited: 11 5, 2008.] <http://www.cuba.cu/gobierno/discursos/2003/esp/f070403e.html>.
3. Marín Díaz, Miguel Enrique. Fundamentos del Sistema de Salud Pública en Cuba para estudiantes de Informática. [book auth.] Miguel Enrique Marín Díaz. La Habana : s.n., 2006.
4. Ministerio de Salud Pública. *Gaceta Oficial de la República de Cuba. Edición Ordinaria*. La Habana : Oficial, 2004.
5. Avila de la Cruz, Lisset del Carmen; Bolmey Romero, Yurisnel and Domínguez López, Rotceh. *Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas.Sistema Gestión de Información en el Proceso de Planificacóon de Materiales Gastables de Uso Médico del Ministerio de Salud Pública(MINSAP)*. [Online] 2008. [Cited: 11 5, 2008.] <http://biblioteca.uci.cu/sbd/biuci/index.html>.
6. Gálves Gonzalez, Ana María. La Evaluación Económica en Salud en Cuba.Instrumento para la toma de desiciones. [Online] [Cited: 1 19, 2009.] <http://www.bibliociencias.cu/gsd/collect/tesis/index/assoc/HASH0109/7fdde813/8c2a976.dir/doc.pdf>.
7. Infomed. [Online] [Cited: 1 19, 2009.] [http://www.dne.sld.cu/minsap/..](http://www.dne.sld.cu/minsap/)
8. Delgado Ramos, Ariel; Cabrera Hernández, Mirna and Rodríguez Díaz, Alfredo. Estrategia de Informatización del Sistema Nacional de Salud. [Online] [Cited: 1 21, 2009.]http://informatica2009.sld.cu/Members/mirnacabrera/estrategia-de-informatizacion-del-sistema-nacional-de-salud/at_download/trabajo..
9. The Free Dictionary. [Online] [Cited: 1 25, 2009.] <http://es.thefreedictionary.com/nomencl%c3%a1to>.
10. Informática en Salud. [Online] [Cited: 1 25, 2009.] <http://informatica2009.sld.cu/Members/denis/nomencladores-nacionales-de-recursos-y-servicios-para-la-informatizacion-de-la-atencion-medica-en-el-sistema-nacional-de-salud>.
11. Rodríguez Díaz, Alfredo. Resgistro de equipos Médicos y no Médicos.Situación Actual. [Online] [Cited: 1 26, 2009.] <http://cencomed.sld.cu/socbio2007/trabajos/pdf/conferencias/aguiar/viernes/alfredo.pdf>.

Referencias Bibliográficas

12. Informática 2009. Sistema para la gestión de enfermedades de declaración obligatoria. [Online] [Cited: 1 27, 2009.]
<http://informatica2009.sld.cu/Members/Thunder/sistema-para-la-gestion-de-la-informacion-de-enfermedades-de-declaracion-obligatoria>.
13. Superintendencia de Entidades prestadoras de Salud. [Online] [Cited: 2 3, 2009.]
http://www.seps.gob.pe/servicios/nomenclador/nomenclador_descripcion.aspx?opcion=12&seccion=179.
14. Qué es Internet? [Online] [Cited: 2 3, 2009.]
<http://www.civila.com/desenredada/que-es.html>.
15. *Ídem ref 14.*
16. Corsi, Miguel and Good, Robin. Beneficios de las Aplicaciones basadas en la Web. [Online] [Cited: 2 5, 2009.]
http://www.masternewmedia.org/es/aplicaciones_web/temas_de_aplicaciones_web/Beneficios_De_Las_Aplicaciones_Basadas_En%20_Web_Y_El_Anuncio_De_Microsoft_De_La_Era_En_Vivo.htm.
17. Portal de Salud de Cuba. [Online] [Cited: 2 7, 2009.]
http://www.sld.cu/sistema_de_salud/aspectos.html#misi3n.
18. Gómez Velázquez, Karel; Gómez Gómez, Leonardo and Arencibia Morales, Annia. *Trabajo de Diploma para optar por el Título de Ingeniero Informático. Centro de Control para el Sistema de Información para la Salud*. La Habana : s.n., 2007.
19. Cinfone. [Online] [Cited: 2 8, 2009.] <http://hospedaje.cinfonet.com/?web=apache>.
20. Presman, Roger S. Ingeniería de Software. Un Enfoque Práctico. La Habana : Félix Varela. Vol. Parte 1, 2005.
21. Legido Hernández, Marcos. Las variables en JavaScript. *Las variables en JavaScript*. [Online] 2005. [Cited: 2 10, 2009.]
http://www.wikilearning.com/caracteristicas_de_javascript-wkccp-3529-3.htm.
22. Herramientas Web para la enseñanza. Protocolos de Comunicación. [Online] [Cited: 2 10, 2009.] <http://neo.lcc.uma.es/evirtual/cdd/tutorial/aplicacion/cliente-servidor.html>.
23. Instituto Nacional Estadísticas Informáticas. Características del Modelo C/S. [Online] [Cited: 2 10, 2009.]
<http://www.inei.gob.pe/biblioiniepub/bancopub/inf/lib5038/carac.HTM>.
24. *Ídem ref 23.*
25. desarrolloweb.com. [Online] [Cited: 2 18, 2009.]
<http://www.desarrolloweb.com/articulos/1557.php>.

Referencias Bibliográficas

26. Anacleto Valerio, Adrián. epidata consulting. [Online] [Cited: 2 19, 2009.] http://www.epidataconsulting.com/tikiwiki/tiki-read_article.php?articleId=15.
27. Eguíluz Pérez, Javier. LibrosWeb.es.Introducción a AJAX . [Online] [Cited: 2 17, 2009.] <http://www.librosweb.es/ajax/capitulo1.html>
28. Maestros de la Web. ¿Qué es Javascript? [Online] [Cited: 2 17, 2009.] <http://www.maestrosdelweb.com/editorial/%C2%BFque-es-javascript/>.
29. Eguiluz Pérez, Javier. LibrosWeb.es. Introducción a JavaScript. [Online] [Cited: 2 17, 2009.] <http://www.librosweb.es/javascript/capitulo1.html>.
30. Minnick, Chris and Chelsea, Valentine. *XHTML Serie Práctica*. Nueva York.Estados Unidos : s.n., 2000. Pág. 3, 4 y, 5.
31. Minnick, Chris and Chelsea, Valentine. *XHTML Serie Práctica*. Nueva York. Estados Unidos : s.n., 2007. Pág. 7.
32. Consorcio World Wide Web (W3C). [Online] [Cited: 2 18, 2009.] <http://www.w3c.es/divulgacion/guiasbreves/HojasEstilo>.
33. Gallego Vázquez, José Antonio. *Desarrollo Web con PHP y MySQL*. Madrid. España : Ediciones Anaya Multimedia, 2003. Pág 20.
34. Postgresql. [Online] [Cited: 2 18, 2009.] <http://www.postgresql.org/about/press/presskit82.html.es>.
35. La Web de Programador. [Online] [Cited: 2 18, 2009.] http://www.lawebdelprogramador.com/news/mostrar_new.php?id=72&texto=PostgreSQL&n1=394636&n2=1&n3=0&n4=0&n5=0&n6=0&n7=0&n8=0&n9=0&n0=0.
36. CSS Framework. [Online] [Cited: 2 18, 2009.] <http://xynova.com/css/YuiLibrary.htm>.
37. Eguíluz Pérez, Javier. librosweb.es.CSS avanzado. [Online] [Cited: 2 19, 2009.] http://www.librosweb.es/css_avanzado/capitulo5/el_framework_yui.html.
38. Zaninotto, François and Potencier, Fabien. librosweb.es.Symfony 1.2, la guía definitiva. [Online] [Cited: 2 19, 2009.] http://www.librosweb.es/symfony_1_2/capitulo1/symfony_en_pocas_palabras.html.
39. *idem ref 5*.
40. sparx systems. [Online] [Cited: 2 19, 2009.] <http://www.sparxsystems.com.ar/new/...>
41. JACOBSON, Ivar James and BOOCH., Grady. *El lenguaje unificado de modelado*. Estados Unidos : s.n., 2000.
42. Teleformacion.uci.cu.Conferencia Ingeniería de Software I curso 2008-2009 . [Online] [Cited: 3 10, 2009.] <http://teleformacion.uci.cu/mod/resource/view.php?id=13165>.

Referencias Bibliográficas

43. merinde.rinde. [Online] [Cited: 3 20, 2009.]

http://merinde.rinde.gob.ve/index.php?option=com_content&task=view&id=105&Itemid=297.

Bibliografía

- Anacleto Valerio, Adrián. epidata consulting. [Online] [Cited: 2 19, 2009.] http://www.epidataconsulting.com/tikiwiki/tiki-read_article.php?articleId=15.
- Arias Sojo, Jorge; Coto García, Jany and Cruz Abat, Aimet Anolan. *Trabajo de Diploma para Optar por el Título de Ingeniero en Ciencias Informáticas. Componente Registro de Estudiantes*. La Habana : s.n., 2008.
- Avila Cantallops, Víctor Manuel. Trabajo de Diploma para Optar por el Título de Ingeniero en Ciencias Informática. Sistema para la planificación de materiales gastables de uso médico (Módulo de Planificación). La Habana : s.n., 2007.
- Avila de la Cruz, Lisset del Carmen; Bolmey Romero, Yurisnel and Domínguez López, Rotceh. *Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas. Sistema Gestión de Información en el Proceso de Planificación de Materiales Gastables de Uso Médico del Ministerio de Salud Pública(MINSAP)*. [Online] 2008. [Cited: 11 5, 2008.] <http://biblioteca.uci.cu/sbd/biuci/index.html>.
- Cinfone. [Online] [Cited: 2 8, 2009.] <http://hospedaje.cinfonet.com/?web=apache>.
- Consorcio World Wide Web (W3C). [Online] [Cited: 2 18, 2009.] <http://www.w3c.es/divulgacion/guiasbreves/HojasEstilo>.
- Corsi, Miguel and Good, Robin. Beneficios de las Aplicaciones basadas en la Web. [Online] [Cited: 2 5, 2009.] http://www.masternewmedia.org/es/aplicaciones_web/temas_de_aplicaciones_web/Beneficios_De_Las_Aplicaciones_Basadas_En%20Web_Y_El_Anuncio_De_Microsoft_De_La_Era_En_Vivo.htm.
- CSS Framework. [Online] [Cited: 2 18, 2009.] <http://xynova.com/css/YuiLibrary.htm>.
- Delgado Ramos, Ariel; Cabrera Hernández, Mirna and Rodríguez Díaz, Alfredo. Estrategía de Informatización del Sistema Nacional de Salud. [Online] [Cited: 1 21, 2009.] http://informatica2009.sld.cu/Members/mirnacabrera/estrategia-de-informatizacion-del-sistema-nacional-de-salud/at_download/trabajo.
- desarrolloweb.com. [Online] [Cited: 2 18, 2009.] <http://www.desarrolloweb.com/articulos/1557.php>.
- Discurso pronunciado por el Presidente de la República de Cuba, Fidel Castro Ruz, en el acto de inauguración de las obras del extraordinario programa de salud ya en marcha, que se lleva a cabo en Cuba. [Online] [Cited: 11 5, 2008.] <http://www.cuba.cu/gobierno/discursos/2003/esp/f070403e.html>. 2.

Bibliografía

- Eguiluz Pérez, Javier. LibrosWeb.es. Introducción a JavaScript. [Online] [Cited: 2 17, 2009.] <http://www.librosweb.es/javascript/capitulo1.html>.
- Eguiluz Pérez, Javier. librosweb.es.CSS avanzado. [Online] [Cited: 2 19, 2009.] http://www.librosweb.es/css_avanzado/capitulo5/el_framework_yui.html.
- Eguiluz Pérez, Javier. LibrosWeb.es.Introducción a AJAX . [Online] [Cited: 2 17, 2009.]<http://www.librosweb.es/ajax/capitulo1.html>
- Expósito González, Elsy and Mileisis Torres Qui, Karina. *Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informática. Propuesta de un procedimiento de pruebas de software en el área temática Sistema de Apoyo a la Salud*. La Habana : s.n., 2008.
- Fernández Cedeño, Karel and Rafael Ortega Valen, Trany. *Trabajo de Diploma para Optar por el Título de Ingeniero en Ciencias Informáticas. Sistema de Información Estadística Complementario de Salud. Módulo: Actividades de Cirugía y otras Atenciones y Servicios*. La Habana : s.n., 2008.
- Free Download Manager. [Online] [Cited: 3 27, 2009.] http://www.freedownloadmanager.org/es/downloads/diagrama_de_despliegue_libre.
- Gálves Gonzalez, Ana María. La Evaluación Económica en Salud en Cuba.Instrumento para la toma de desiciones. [Online] [Cited: 1 19, 2009.] <http://www.bibliociencias.cu/gsd/collect/tesis/index/assoc/HASH0109/7fdde813/8c2a976.dir/doc.pdf>.
- Gómez Velázquez, Karel; Gómez Gómez, Leonardo and Arencibia Morales, Annia. *Trabajo de Diploma para optar por el Título de Ingeniero Informático. Centro de Control para el Sistema de Información para la Salud*. La Habana : s.n., 2007.
- Herramientas Web para la enseñanza. Protocolos de Comunicación. [Online] [Cited: 2 10, 2009.] <http://neo.lcc.uma.es/evirtual/cdd/tutorial/aplicacion/cliente-servidor.html>.
- Infomed. [Online] [Cited: 1 19, 2009.] [http://www.dne.sld.cu/minsap/..](http://www.dne.sld.cu/minsap/)
- Informática 2009.Sistema para la gestion de enfermedades de declaracion obligatoria. [Online] [Cited: 1 27, 2009.] <http://informatica2009.sld.cu/Members/Thunder/sistema-para-la-gestion-de-la-informacion-de-enfermedades-de-declaracion-obligatoria>.
- Informática en Salud. [Online] [Cited: 1 25, 2009.] <http://informatica2009.sld.cu/Members/denis/nomencladores-nacionales-de-recursos-y-servicios-para-la-informatizacion-de-la-atencion-medica-en-el-sistema-nacional-de-salud>.

Bibliografía

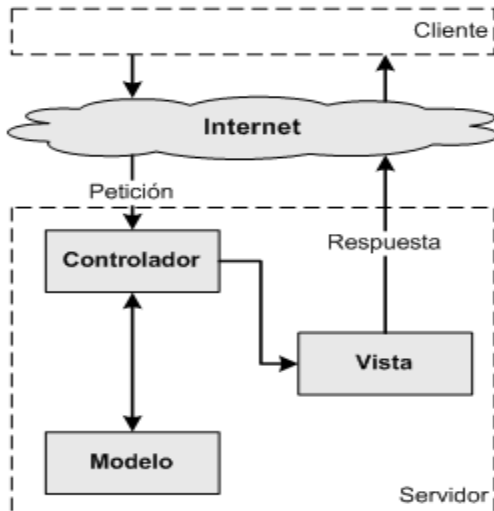
- Instituto Nacional Estadísticas Informáticas. Características del Modelo C/S. [Online] [Cited: 2 10, 2009.] <http://www.inei.gob.pe/biblioiniepub/bancopub/inf/lib5038/carac.HTM>.
- JACOBSON, Ivar James and BOOCH., Grady. El lenguaje unificado de modelado. Estados Unidos : s.n., 2000.
- La Web de Programador. [Online] [Cited: 2 18, 2009.] http://www.lawebdelprogramador.com/news/mostrar_new.php?id=72&texto=PostgreSQL&n1=394636&n2=1&n3=0&n4=0&n5=0&n6=0&n7=0&n8=0&n9=0&n0=0.
- Legido Hernández, Marcos. Las variables en JavaScript. Las variables en JavaScript. [Online] 2005. [Cited: 2 10, 2009.] http://www.wikilearning.com/caracteristicas_de_javascript-wkccp-3529-3.htm.
- Maestros de la Web. ¿Qué es Javascript? [Online] [Cited: 2 17, 2009.] <http://www.maestrosdelweb.com/editorial/%C2%BFque-es-javascript/>.
- Marín Díaz, Miguel Enrique. Fundamentos del Sistema de Salud Pública en Cuba para estudiantes de Informática. [book auth.] Miguel Enrique Marín Díaz. La Habana : s.n., 2006.
- merinde.rinde. [Online] [Cited: 3 20, 2009.] http://merinde.rinde.gob.ve/index.php?option=com_content&task=view&id=105&Itemid=297.
- Ministerio de Salud Pública. Gaceta Oficial de la República de Cuba. Edición Ordinaria. La Habana : Oficial, 2004.
- Minnick, Chris and Chelsea, Valentine. XHTML Serie Práctica. Nueva York. Estados Unidos : s.n., 2007. Pág. 7.
- Minnick, Chris and Chelsea, Valentine. XHTML Serie Práctica. Nueva York. Estados Unidos : s.n., 2007. Pág. 3, 4 y, 5.
- Portal de Salud de Cuba. [Online] [Cited: 2 7, 2009.] http://www.sld.cu/sistema_de_salud/aspectos.html#misi3n.
- PostgreSQL. [Online] [Cited: 2 18, 2009.] <http://www.postgresql.org/about/press/presskit82.html.es>.
- Presman, Roger S. Ingeniería de Software. Un Enfoque Práctico. La Habana : Félix Varela. Vol. Parte 1, 2005.
- Qué es Internet? [Online] [Cited: 2 3, 2009.] <http://www.civila.com/desenredada/que-es.html>.

Bibliografía

- Rodríguez Díaz, Alfredo. Registro de equipos Médicos y no Médicos.Situación Actual. [Online] [Cited: 1 26, 2009.] <http://cencomed.sld.cu/socbio2007/trabajos/pdf/conferencias/aguiar/viernes/alfredo.pdf>.
- sparx systems. [Online] [Cited: 2 19, 2009.] <http://www.sparxsystems.com.ar/new/...>
- Superintendencia de Entidades prestadoras de Salud. [Online] [Cited: 2 3, 2009.] http://www.seps.gob.pe/servicios/nomenclador/nomenclador_descripcion.aspx?opcion=12&seccion=179.
- Teleformacion.uci.cu.Conferencia Ingeniería de Software I curso 2008-2009 . [Online] [Cited: 3 10, 2009.] <http://teleformacion.uci.cu/mod/resource/view.php?id=13165>.
- The Free Dictionary. [Online] [Cited: 1 25, 2009.] <http://es.thefreedictionary.com/nomencl%c3%a1to>.
- voltairenet.org.Red de Prensa No Alineados. [Online] [Cited: 11 5, 2008.]<http://www.voltairenet.org/article129780.html>.
- Zaninotto, François and Potencier, Fabien. librosweb.es.Symfony 1.2, la guía definitiva. [Online] [Cited: 2 19, 2009.] http://www.librosweb.es/symfony_1_2/capitulo1/symfony_en_pocas_palabras.html.

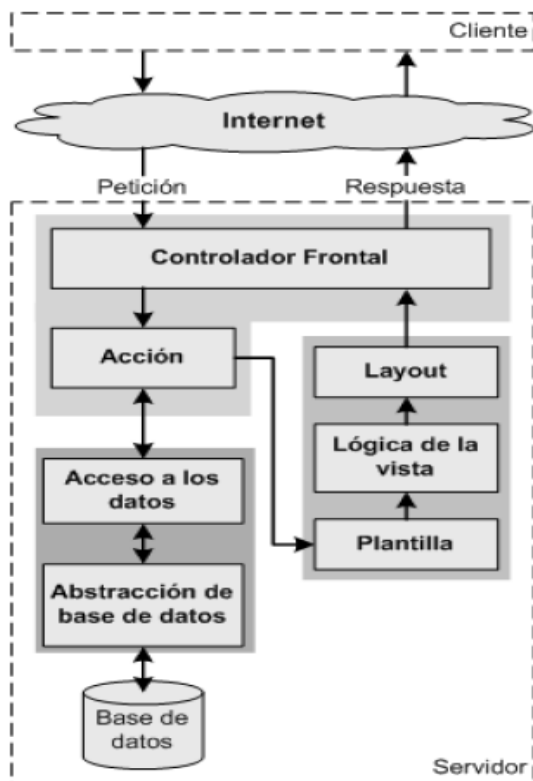
Anexos

Anexo 1 Modelo Vista Controlador



Arquitectura MVC

Anexo 2 MVC según Symfony



Implementación del patrón MVC según el framework Symfony.

Anexo 3 Diagramas de Clases

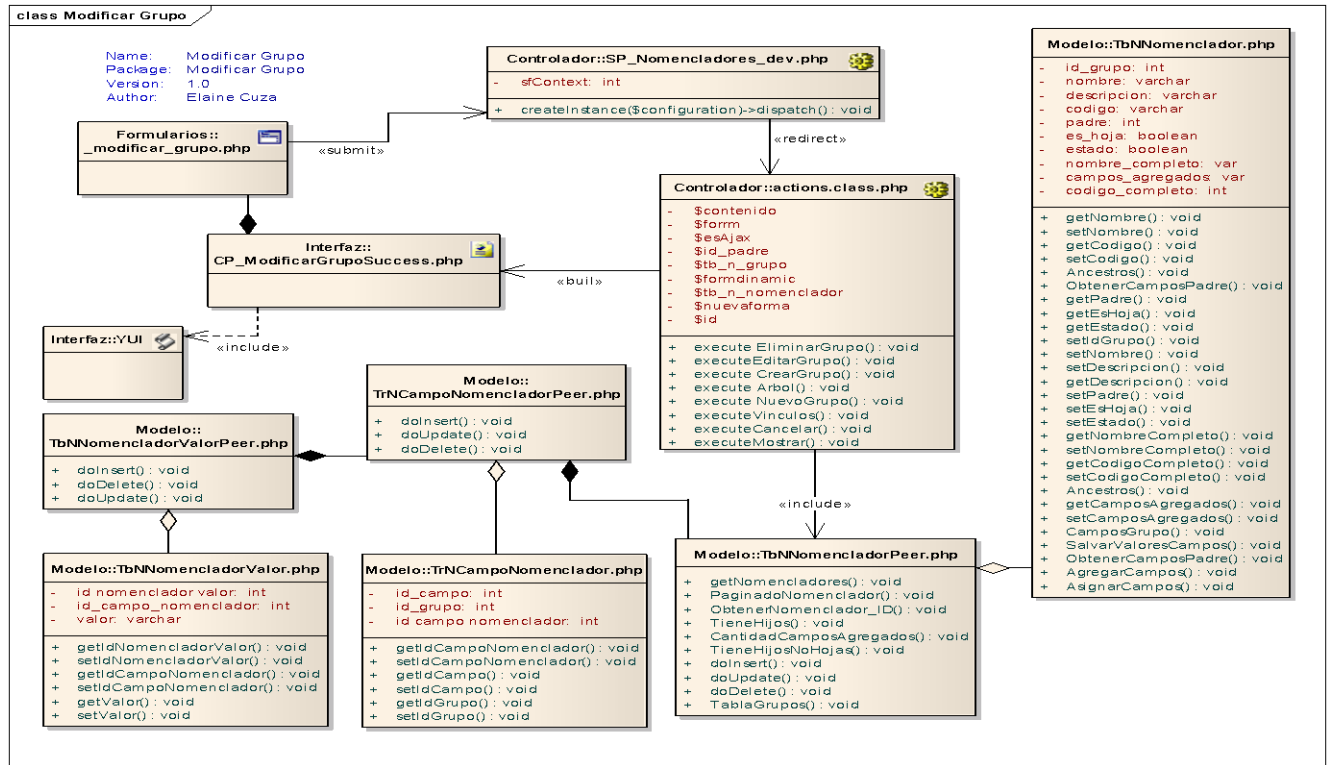


Diagrama de clase CUS_Modificar Grupo

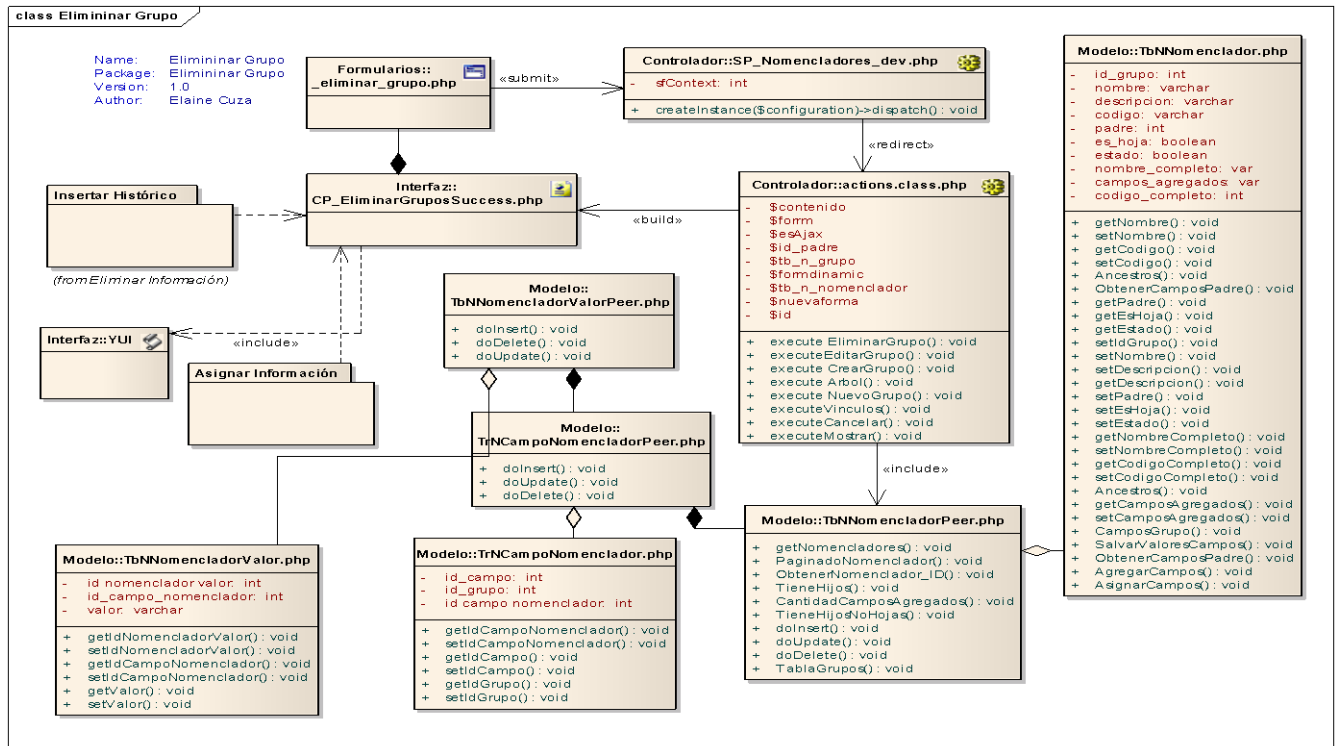


Diagrama de clase CUS_Eliminar Grupo

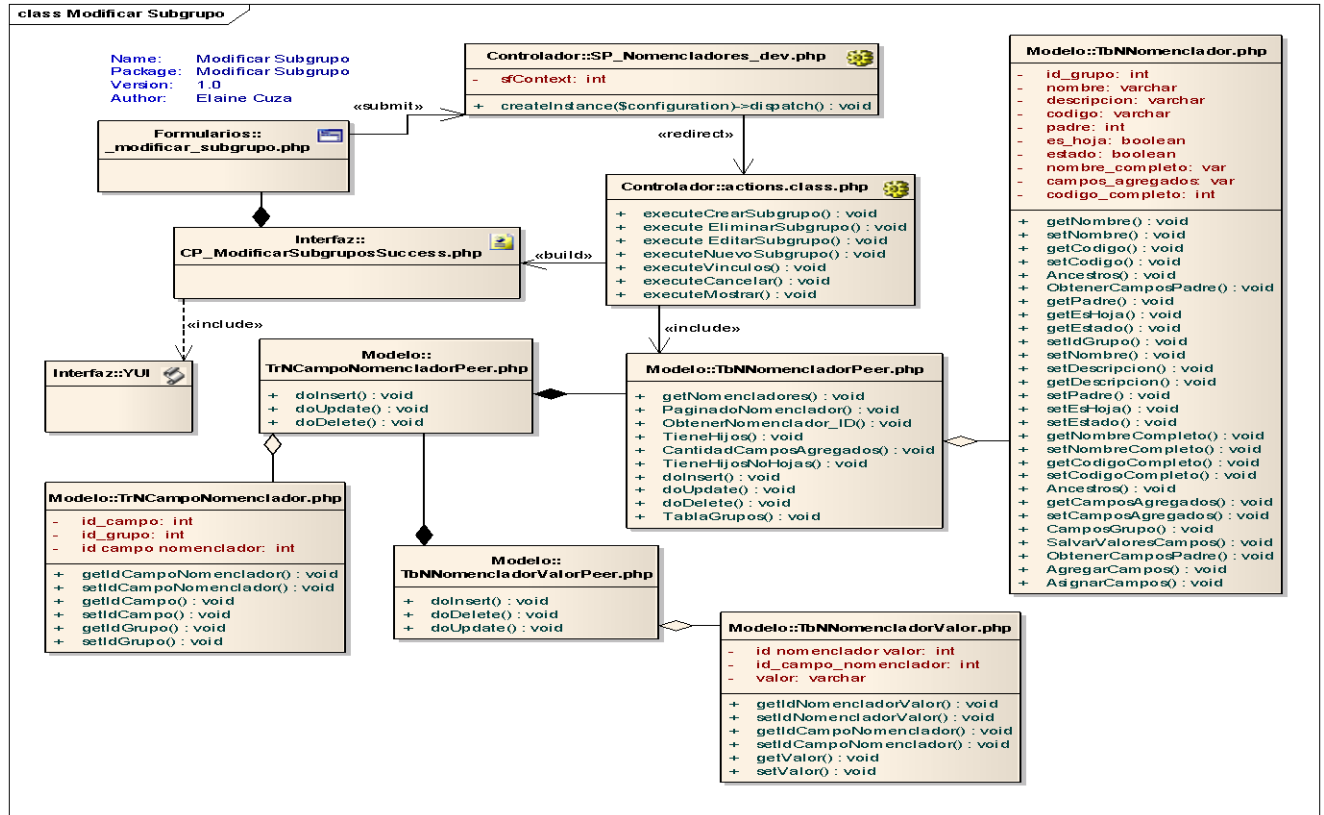


Diagrama de clase CUS_Modificar Subgrupo

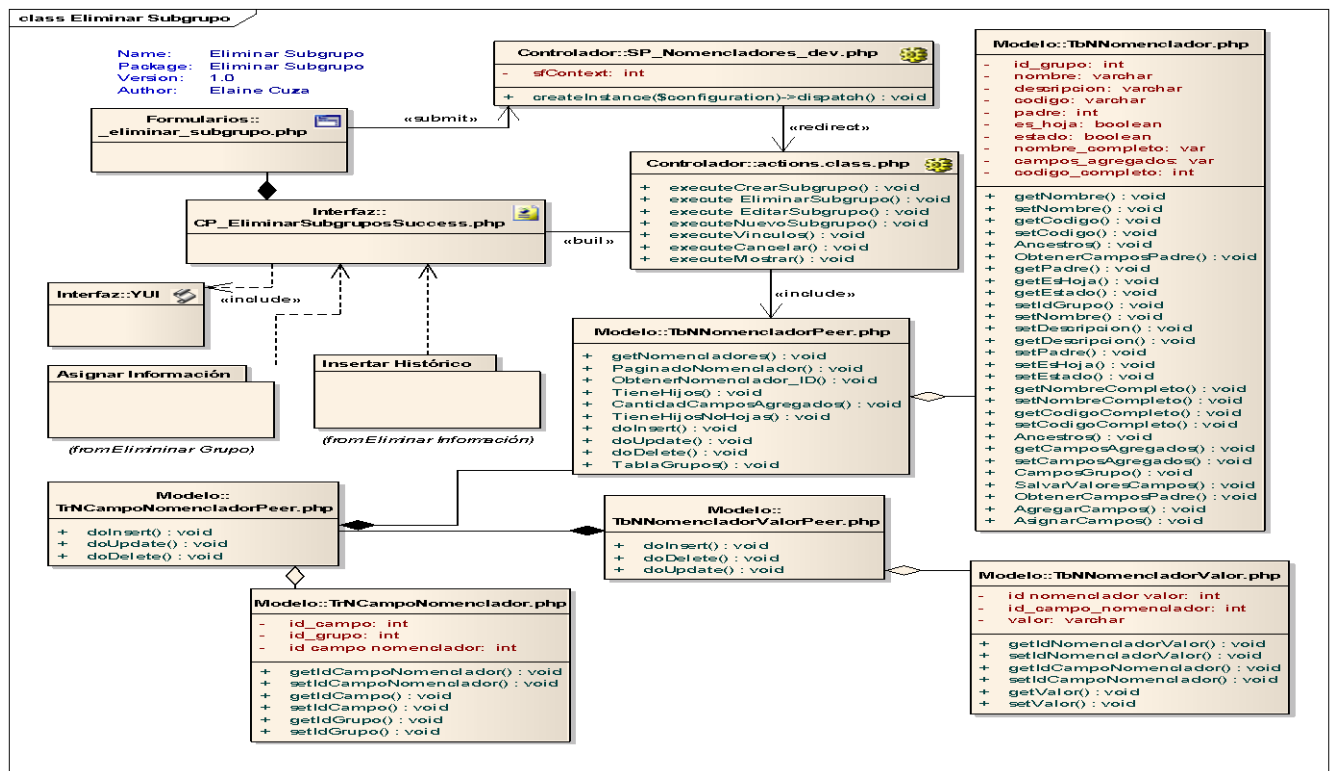


Diagrama de clase CUS_Eliminar Subgrupo

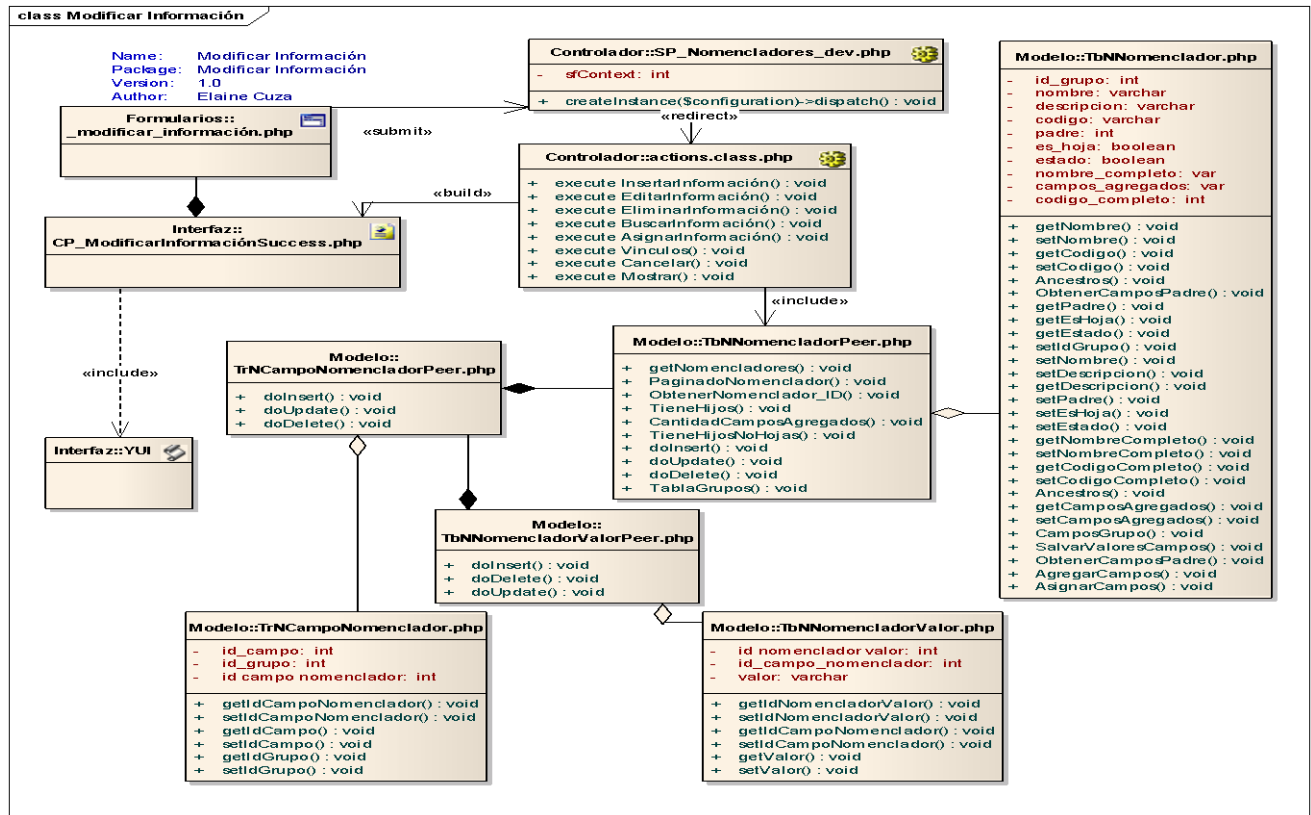


Diagrama de clase CUS_Modificar Información

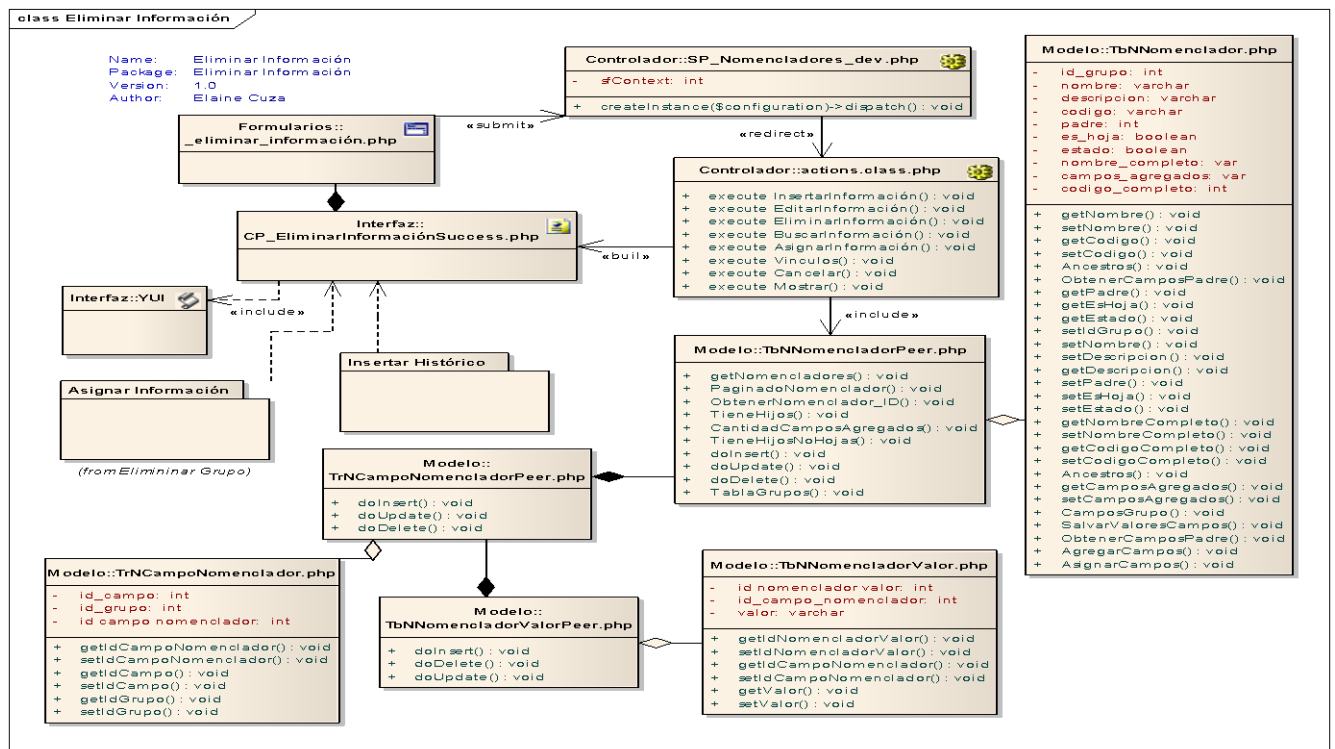


Diagrama de clase CUS_Eliminar Información

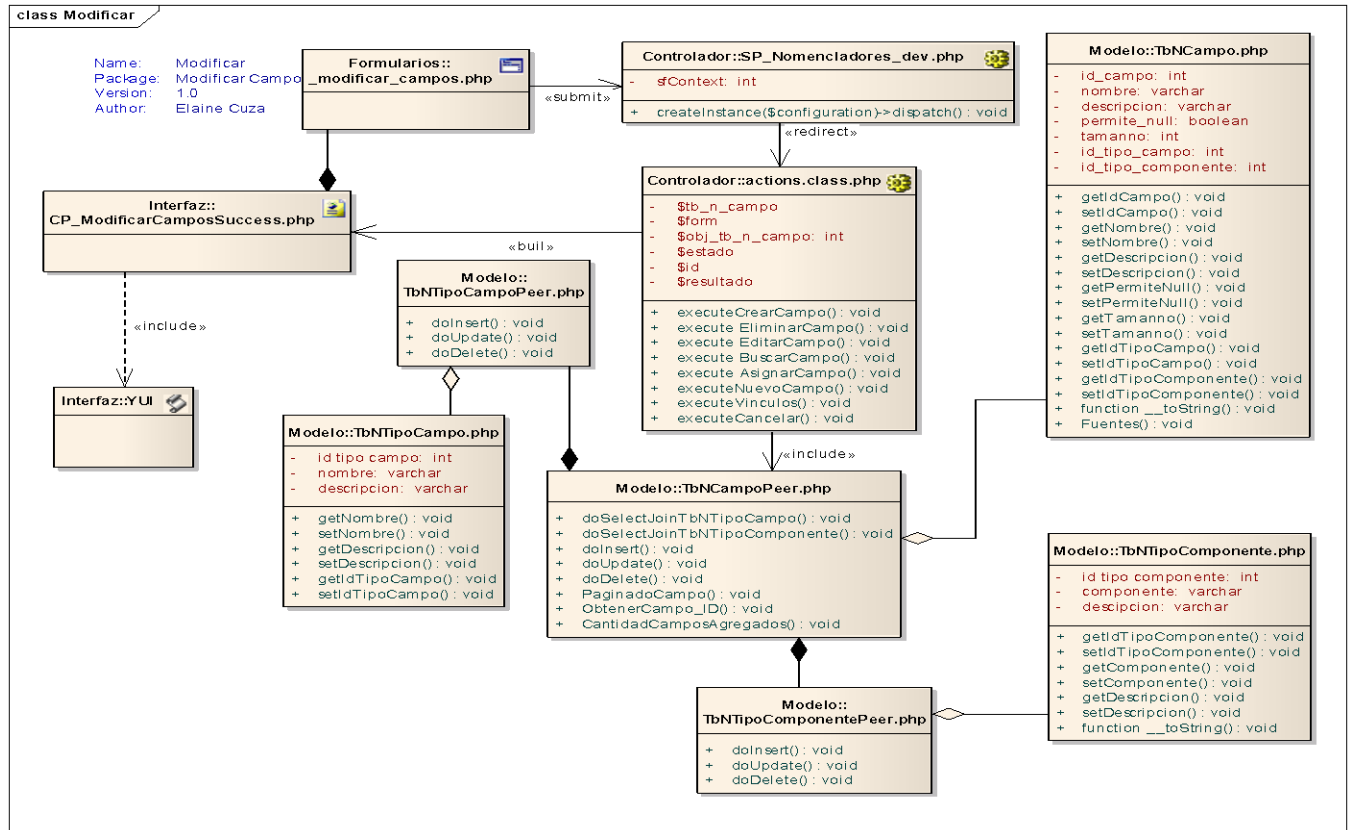


Diagrama de clase CUS_Modificar Campo

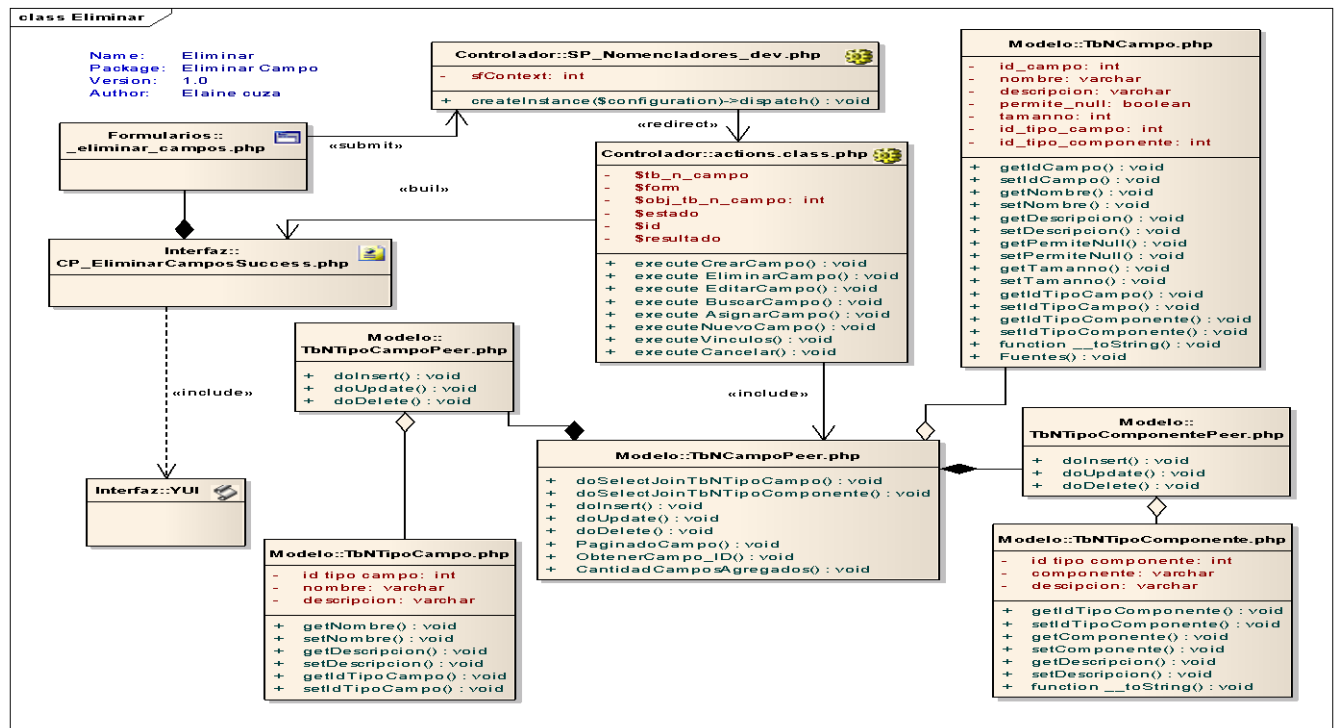


Diagrama de clase CUS_Eliminar Campo

Anexo 3 Diagramas de Secuencia

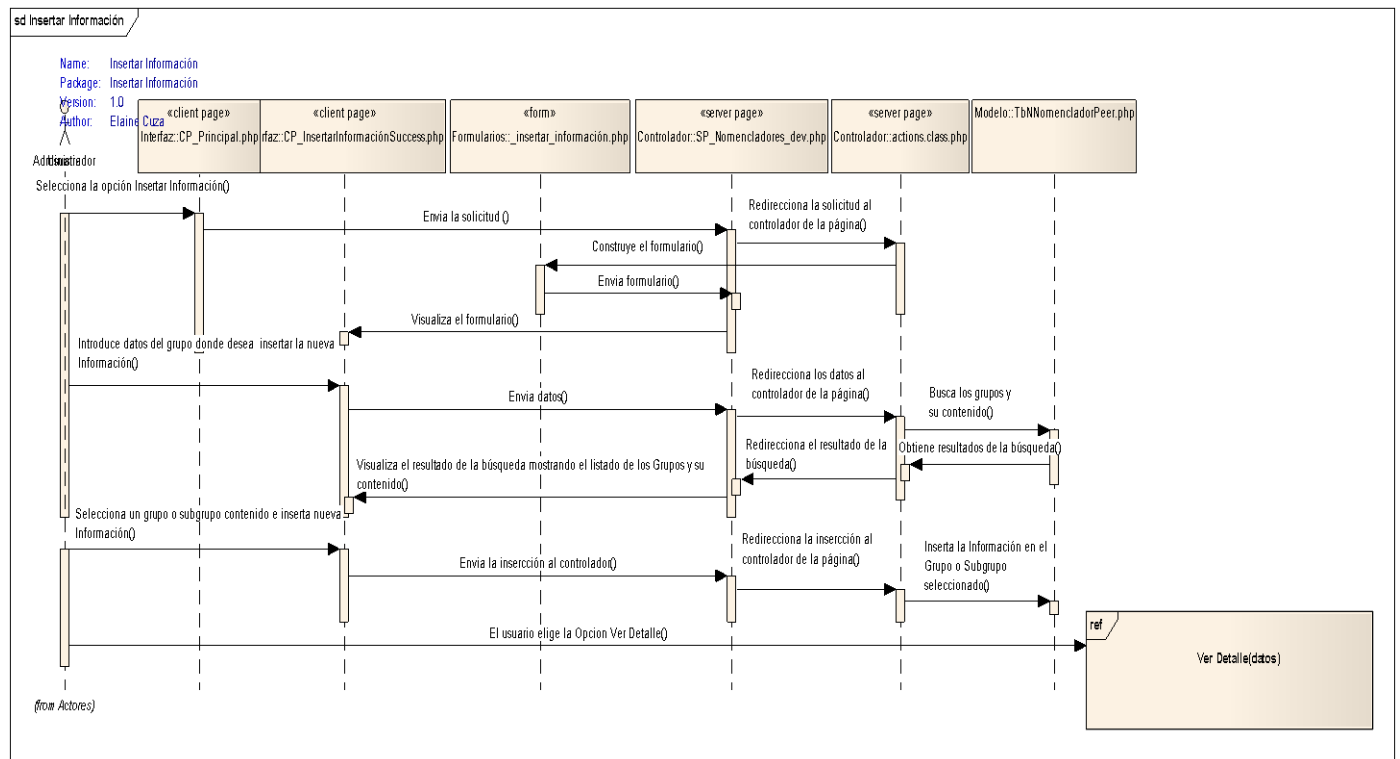


Diagrama de secuencia CUS_Insertar Información

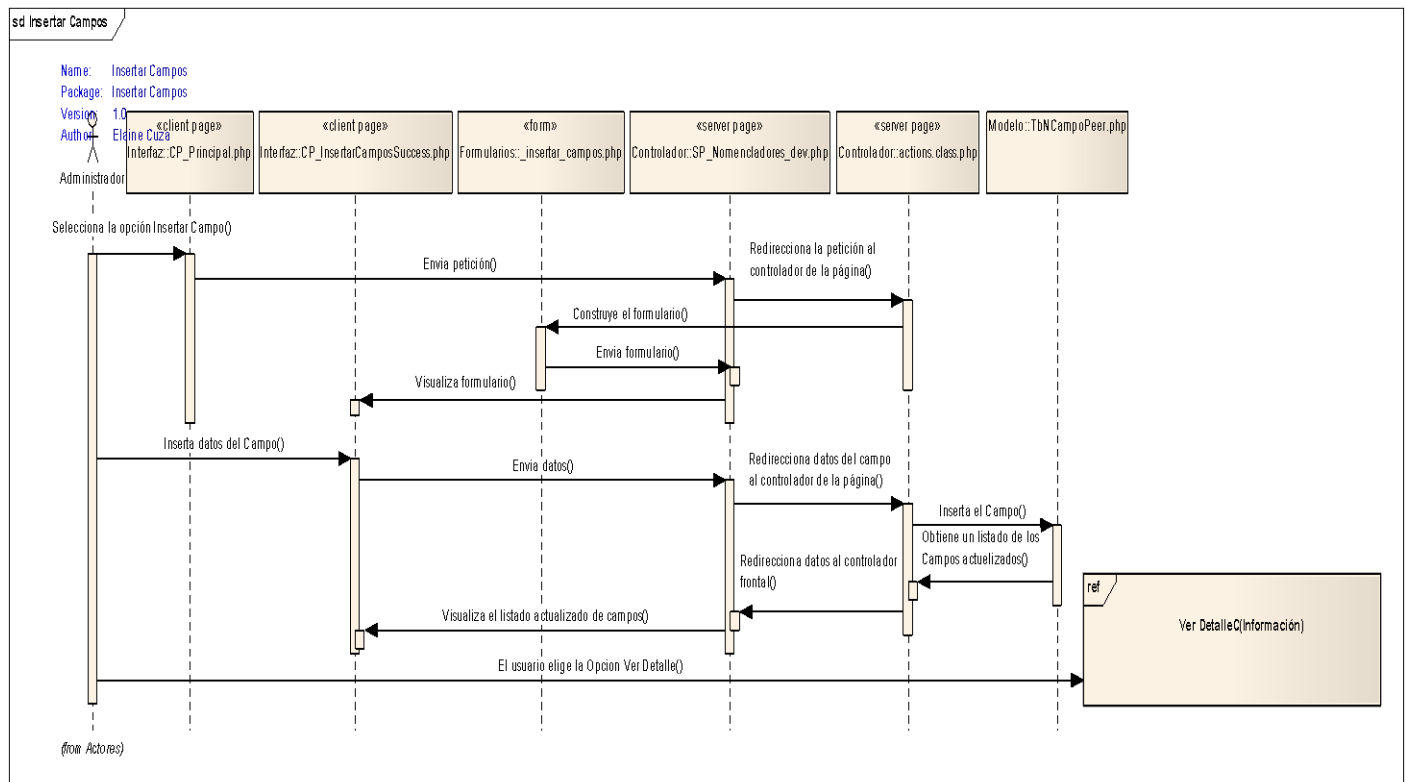
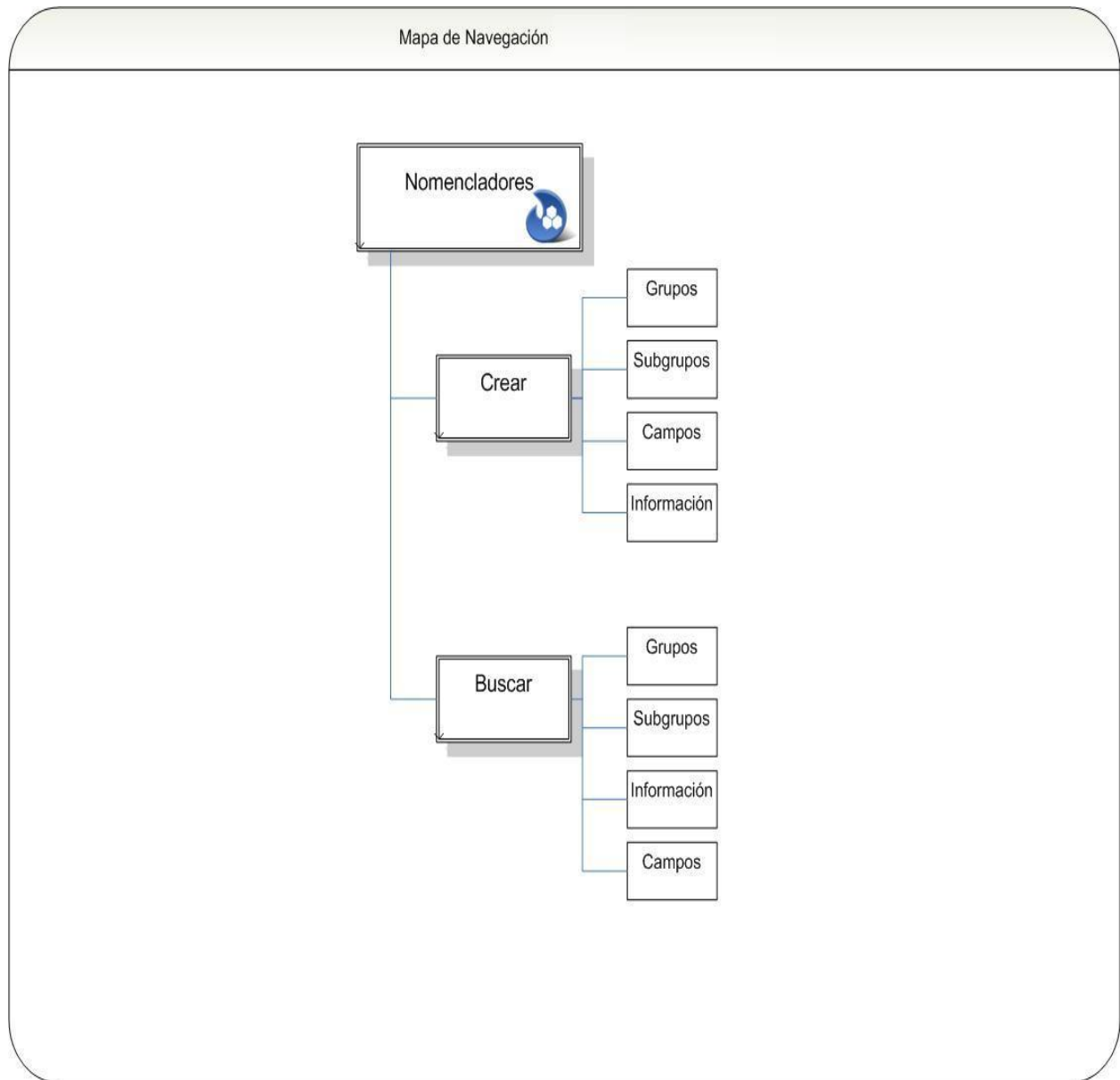
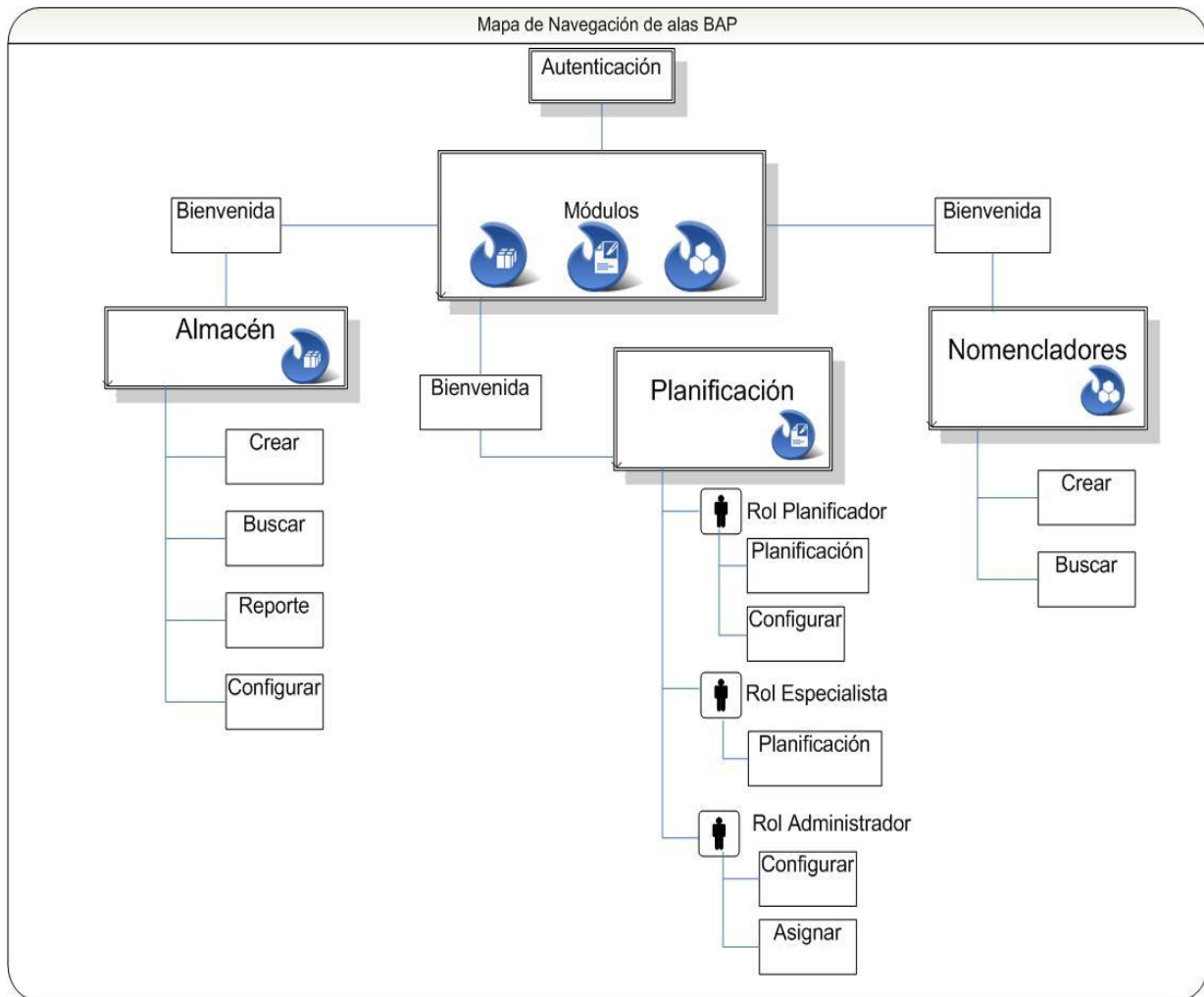


Diagrama de secuencia CUS_Insertar Campos

Anexo 5 Mapas de Navegación



Mapa de navegación módulo Nomenclador



Mapa de navegación alas BAP