

**Universidad de las Ciencias Informáticas**  
**Facultad 7**



***Título: Análisis y Diseño del Componente de Notificación y Mensajería para productos del Área Temática Sistemas de Apoyo a la Salud***

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

**Autoras:** Lisandra Yenet Garzón Ferrer  
Marleydi Hernández Paula

**Tutores:** Ing. Karel Gómez Velázquez  
Ing. Norge Martínez Almaguer

Ciudad de La Habana, Julio de 2008  
"Año 50 de la Revolución"

## DECLARACIÓN DE AUTORÍA

Declaramos que somos las únicas autoras de este trabajo y autorizamos a la Facultad 7 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los 4 días del mes de julio del año 2008:

---

Lisandra Yenet Garzón Ferrer

***Autor***

---

Marleydi Hernández Paula

***Autor***

---

Ing. Karel Gómez Velázquez

***Tutor***

---

Ing. Norge Martínez Almaguer

***Tutor***

## DATOS DE CONTACTO

**Ing. Karel Gómez Velázquez** ([kgomez@uci.cu](mailto:kgomez@uci.cu)): Ingeniero en Ciencias Informáticas, graduado en la UCI, en el curso 2006-07. Posee la Categoría Docente de Profesor Adiestrado. Ha participado en proyectos de desarrollo de Sistemas Informáticos para la Salud desde el año 2005. Actualmente imparte asignaturas de Técnicas de Programación. Además, se desempeña como Asesor de Tecnologías y Arquitectura en la Facultad 7.

**Ing. Norge Martínez Almaguer** ([nmartinez@uci.cu](mailto:nmartinez@uci.cu)): Ingeniero en Ciencias Informáticas, graduado en la UCI, en el curso 2006-07. Posee la Categoría Docente de Profesor Adiestrado. Ha participado en proyectos de desarrollo de Sistemas Informáticos para la Salud desde el año 2005. Actualmente imparte asignaturas de Práctica Profesional. Además, se desempeña Líder de Proyecto del Área Temática Sistemas de Apoyo a la Salud.

*“La independencia no es una bandera, o un himno, o un escudo; la independencia no es una cuestión de símbolo, la independencia depende del desarrollo, depende de la tecnología, depende de la ciencia en el mundo de hoy”*

A handwritten signature in black ink, appearing to read 'Fidel Castro Ruz', with a stylized flourish at the end.

*Fidel Castro Ruz*

## **AGRADECIMIENTOS**

*A nuestro Comandante Fidel y a la Revolución por habernos dado la oportunidad de formarnos en una Universidad del Futuro.*

*A nuestros tutores Karel Gómez y Norge Martínez, por brindarnos tanta confianza, dedicación y profesionalidad.*

*A todas aquellas personas que han brindado su apoyo para el desarrollo exitoso de nuestro Trabajo de Diploma.*

*En especial a nuestras familias, por guiarnos ejemplarmente por el camino más certero, por entregarnos toda su confianza durante estos años y contribuir extraordinariamente en la formación profesional que hemos adquirido.*

*A todos ellos:*

*Gracias de todo corazón...*

## DEDICATORIA

*A las personas más importantes en mi vida: Mis mamis Lidia y Lourdes y mis papitos Marcos y Julio, por inculcarme esas ansias de superación continua y demostrarme que mientras que haya vida, habrá esperanzas.*

*A toda mi familia por depositar siempre toda su confianza, en especial a Celi, Fabi y Mily, por plasmar todas esas cosas lindas que me conforman. A mi novio Danny por su apoyo y comprensión, por ser el camino y la vida.*

*A su familia por inspirarme todos estos esfuerzos. A mis amigos por enseñarme a alcanzar mis sueños.*

*A todos Uds. va dedicado este trabajo.*

*Marleydi Hernández Paula.*

*A las personas mas maravillosas del planeta, mi madre Ileana y mi padre Juan Enrique que me han apoyado a todo lo largo de mi vida. A mis padrinos Ynelis y Alfonso y mis tíos Francis, Angel, Nelza, Zoila y Juana por ayudarme siempre y estar conmigo en todos los momentos. A mi familia en general que nunca me ha abandonado. Al amor de mi vida Marlon por tener paciencia conmigo y brindarme mucho amor y cariño. A mis amigos del alma Caty, Jany, Karina, Maylin, Joe y Eslavy por demostrarme que en el mundo aun existen amistades, un beso a todos, nunca los olvidaré. A mis compañeros de grupo por estar siempre conmigo.*

*A mi tutor Karel Gómez por ayudarme a forjarme un poco más y demostrarme que todo es posible. A todos aquellos los que hicieron posible realizar mis sueños.*

*A todos ustedes un abrazo de todo corazón.*

*Lisandra Garzón Ferrer.*

## RESUMEN

El Componente de Notificación y Mensajería se concibe a partir de las necesidades de comunicación existentes entre las diferentes instancias del Ministerio de Salud Pública (MINSAP), donde los servicios de mensajería no se encuentran disponibles para todas las personas. El objetivo del trabajo consiste en diseñar un componente que permita la notificación y el intercambio de mensajes entre usuarios y Sistemas de Apoyo a la Salud, facilitando así el intercambio de información entre estos.

Su desarrollo está basado en tecnologías libres, multiplataformas y sobre una arquitectura en capas. Se utiliza PHP 5 como lenguaje de programación, se implementa el patrón de arquitectura Modelo Vista Controlador, PostgreSQL 8.2 como Sistema de Gestión de Bases de Datos, metodología AJAX para realizar más eficientemente las peticiones al servidor y la librería Ext JS para obtener una interfaz visual. Además se emplean estándares abiertos como XML lo que permite la interoperabilidad entre aplicaciones desarrolladas sobre diferentes plataformas y el protocolo de transporte HTTP para el transporte de los datos.

Mediante este componente se permite integrar los servicios de mensajería síncrona y asíncrona, modelando Servicios Web XML que posibilitarán, a sistemas externos, el envío de notificaciones a los usuarios que interactúan con la información gestionada por estos. Fue obtenido un prototipo no funcional refinado que permitirá una correcta comprensión de los requerimientos de software especificados, lo que constituye un elemento significativo para el proceso de implementación.

**Palabras Claves:** *notificación, mensajería, síncrona, asíncrona, servicio.*

**TABLA DE CONTENIDOS**

**CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA** ..... 7

    1.1 Objeto de automatización e información manipulada ..... 7

    1.2 Antecedentes de Sistemas de Mensajería ..... 9

    1.3 Tendencias y Tecnologías actuales..... 12

    1.4 Herramientas, tecnologías, lenguajes y metodología de soporte al desarrollo..... 18

    1.5 Herramienta a utilizar ..... 25

**CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA** ..... 27

    2.1 Modelo de Dominio ..... 27

    2.2 Propuesta de Sistema..... 29

    2.3 Modelo de Casos de Uso del Sistema ..... 35

**CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA**..... 44

    3.1 Modelo de Análisis ..... 44

    3.2 Modelo de Diseño ..... 48

    3.3 Descripción de clases y atributos ..... 58

    3.4 Diseño de la Base de Datos ..... 66

    3.5 Modelo de Despliegue ..... 74

    3.6 Prototipo no Funcional ..... 75

**CONCLUSIONES** ..... 80

**RECOMENDACIONES** ..... 81

**REFERENCIAS BIBLIOGRÁFICAS** ..... 82

**BIBLIOGRAFÍA**..... 84

**ANEXOS** ..... 86

**GLOSARIO DE TÉRMINOS**..... 92

## INTRODUCCIÓN

En Cuba la informatización de la sociedad se define como el proceso de utilización ordenada y masiva de las Tecnologías de la Información y las Comunicaciones (TIC). Este proceso busca lograr más eficacia y eficiencia en la prestación de los servicios haciendo sustentable el aumento sistemático de la calidad de vida de la población. La estrategia de informatización está contenida en el Programa Rector de la Informatización de la Sociedad en Cuba, en el que se contempla: Infraestructura, Tecnologías y Herramientas, Formación Digital, Fomento de la Industria Nacional de las Tecnologías de la Información y las Comunicaciones entre otras.

Este persigue promover el uso masivo de las Tecnologías de la Información a escala nacional, teniendo en cuenta los objetivos generales estratégicos que el país se ha propuesto, buscando un desarrollo coherente y una identificación precisa de los actores de la Sociedad de la Información. Esta estrategia, como expresión del proceso revolucionario cubano, tiene al ciudadano en el centro de sus objetivos, buscando elevar su calidad de vida en su desempeño familiar, laboral, educacional, cultural, social y político y en la consecución del fortalecimiento y ampliación de los logros y beneficios que la Revolución le ha dado. [1]

La preparación de los recursos humanos especializados para las TIC es un factor clave de la estrategia cubana de Informatización. A mediados del 2002 se creó la Universidad de las Ciencias Informáticas (UCI). La UCI como universidad de nuevo tipo, primera de la Batalla de Ideas, posee un novedoso modelo de formación que combina el estudio con la producción y la investigación. Este centro tiene como objetivo fundamental crear especialistas en el campo de la informática para poder apoyar y seguir adelante con el proceso de informatización de la sociedad. Además juega un importante papel en el desarrollo de la Industria Cubana del Software pues de conjunto con otras empresas, obtienen productos y soluciones informáticas que se insertan en todos los sectores de la sociedad. [2]

La producción de la UCI está estructurada en Polos Productivos, que responden a líneas estratégicas de investigación y desarrollo identificadas. La Facultad 7 cuenta entre sus polos con uno asociado a la Informatización de los Servicios de Salud, el cual contiene el Área Temática Sistemas de Apoyo a la Salud (SAS), que se encarga del desarrollo de software que dan soporte a diferentes actividades no

asistenciales, es decir no vinculadas directamente con la atención médica a pacientes. En esta área temática se desarrollan los siguientes sistemas:

1. Gestión de Información en el Proceso de Formación de Recursos Humanos en Salud Pública.
2. Sistema para la Planificación, Balance y Distribución de suministros médicos.
3. Sistema de Información Estadística Complementario de Salud (SIEC-Salud).
4. Sistema para la Gestión y Control de los Colaboradores de la Salud.

El escenario típico de cualquier sistema informático para la salud es insertarse en un panorama donde coexisten numerosos sistemas heterogéneos y distribuidos en diferentes ubicaciones físicas. Es inevitable la integración de sus datos lo que consiste en la combinación de la información que se encuentra almacenada en componentes con estas características, proporcionado al usuario final una vista unificada de los mismos. Tal integración desde el punto de vista clínico; tiene el propósito de facilitar a los usuarios, ya sea personal médico, administrativo, investigativo o al propio paciente un conglomerado de la información obtenida a partir del paciente en el proceso de atención.

La comunicación entre los usuarios y Sistemas de Apoyo a la Salud resulta de gran importancia, debido al constante flujo de información que se establece entre los diferentes niveles de dirección del Ministerio de Salud Pública, utilizado para análisis estadístico y la toma de decisiones. Los sistemas de mensajería son unos de los medios más utilizados para este intercambio de mensajes, pues permiten mantener comunicación a larga distancia sin la presencia física de las personas. Existen dos tipos de sistemas de mensajería: Síncrona, donde los mensajes son enviados en tiempo real, en las que se encuentran los chat y foros y Asíncrona donde los mensajes son enviados de forma similar al correo postal, solo se necesita la dirección de correo del destinatario, siendo actualmente el más difundido el correo electrónico.

Los sistemas de mensajería en muchas ocasiones no resuelven los problemas de comunicación existentes entre los miembros de una institución determinada, ya que pueden presentar una serie de dificultades imposibilitando el intercambio de información y en muchas ocasiones no estando disponibles para todo el personal. Como consecuencia inmediata de esta dificultad se evidencia que toda la información que se quiera divulgar tenga que realizarse de forma física con el personal o

convocando reuniones, siendo un medio incómodo pues pueden surgir una serie de dificultades con las personas y estas ausentarse, no recibiendo la información correspondiente.

No existen sistemas capaces de establecer la conexión entre oficinas o sucursales que pertenezcan a una empresa determinada, haciéndose engorroso el trabajo de las personas que laboran en estos lugares al realizar intercambios de información, pues al no tener un medio de comunicación, en muchas ocasiones no se realiza la entrega de determinados informes en el momento requerido, manifestándose el incumplimiento.

Normalmente los sistemas de mensajería se encuentran coexistiendo de manera independiente entre sí, es decir no de forma integrada y representando la labor que desempeñan los usuarios en la organización; por lo que es necesario poseer conjuntos de credenciales diferentes para acceder a sus funcionalidades. Además no es posible que los sistemas informáticos desplegados en un dominio determinado, puedan hacer uso de estos servicios para establecer comunicación con sus usuarios y así realizar el envío de mensajes y notificaciones asociadas a las acciones que los mismos deben realizar sobre la información del sistema entre ellos cierres estadísticos, reuniones, intercambio de datos, entre otras.

En la actualidad la mayoría de los envíos de información se realizan de forma manual o por vía telefónica, trayendo consigo la lentitud de los partes informativos, la escasa comunicación entre los distintos niveles y la escasa retroalimentación de las gestiones realizadas. Es por ello que comúnmente los especialistas funcionales del sector solicitan entre sus requerimientos de software elementos que den solución a situaciones como las mencionadas anteriormente, los cuales no pertenecen a las necesidades de informatización de determinados procesos trayendo consigo que se realicen desarrollos innecesarios que provocan retrasos en los cronogramas pactados.

Actualmente el proyecto Sistema para la Planificación, Balance y Distribución de suministros médicos cuenta con un sistema de mensajería asíncrona el cual no cumple con todas las necesidades de comunicación identificadas para los productos de SAS. Debido a que no ofrece una solución estándar y la comunicación está restringida solo entre personas, no pudiendo el propio sistema en un momento determinado intercambiar mensajes con sus usuarios.

En este sentido se manifiesta el siguiente **Problema a resolver** por el Componente de Notificación y

Mensajería: ¿Cómo facilitar el intercambio de mensajes entre los usuarios de los Sistemas de Apoyo a la Salud?

**Objeto de estudio:** Proceso de notificación y mensajería entre los usuarios de los Sistemas de Apoyo a la Salud.

**Campo de acción:** Proceso de diseño de un Sistema de Notificación y Mensajería para productos del Área Temática Sistema de Apoyo a la Salud.

Como **Objetivo general** se persigue: Diseñar un componente que permita la notificación y el intercambio de mensajes entre usuarios y Sistemas de Apoyo a la Salud.

Para dar cumplimiento al objetivo anteriormente planteado se definen las siguientes **Tareas de la investigación:**

1. Analizar el Estado del Arte de los sistemas de mensajería existentes.
2. Analizar las tendencias y tecnologías actuales para realizar el proceso de diseño de dicha solución como por ejemplo PHP, XML, Servicios Web XML, AJAX, PostgreSQL, y JavaScript.
3. Asimilar de la Arquitectura definida por el MINSAP para el desarrollo de sus aplicaciones.
4. Documentar artefactos de Trabajo del Proceso Unificado de Desarrollo “*Modelo de Negocio*”, “*Gestión de Requerimientos*” y “*Análisis y Diseño*”.
5. Obtener un prototipo no funcional refinado para el Componente de Notificación y Mensajería.

De forma general con el establecimiento de este componente de Notificación y Mensajería se obtienen los siguientes **Beneficios:**

- Proporcionarle al implementador un análisis y diseño refinado así como un prototipo no funcional.

- Proporcionarle a los usuarios finales un sistema de mensajería que integre tanto la comunicación asíncrona con la asíncrona con un mismo conjunto de credenciales de acceso.
- Integración con un Componente de Seguridad que reutiliza los datos de los usuarios mediante el protocolo LDAP, provee servicios de búsqueda de usuarios y de componentes. Evitando de esta forma la duplicación de la información en la base de datos.
- Constituye un valor agregado a los productos desarrollados por el Área Temática SAS, debido a que además de resolver su negocio tendrán integrado su propio Sistema de Notificación y Mensajería.
- Publicación de Servicios Web XML para sistemas externos de que permitan el envío de mensajes asíncronos, con sus respectivas descripciones WSDL.

El presente documento se encuentra estructurado en tres capítulos. El primer capítulo, **“FUNDAMENTACIÓN TEÓRICA”** describe el contexto de desarrollo del Componente de Notificación y Mensajería argumentando las tecnologías, herramientas y metodologías a utilizar. El segundo capítulo **“CARACTERÍSTICAS DEL SISTEMA”** se detallan las reglas del negocio, los requerimientos funcionales, no funcionales, los actores y los distintos diagramas. El tercer y último capítulo **“ANÁLISIS y DISEÑO DEL SISTEMA”** se centra en la modelación detallada, donde los requerimientos funcionales se especifican y se describe como se debe implementar el sistema presentándose además la propuesta de un prototipo no funcional del componente.

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

El objetivo de este capítulo es abordar los distintos elementos que brindan la base teórico conceptual para el desarrollo del Componente de Notificación y Mensajería. Se analizan antecedentes de sistemas de mensajería y se argumentan el por qué de selección de las tecnologías, herramientas y metodologías a utilizar en el desarrollo de dicho Componente.

### 1.1 Objeto de automatización e información manipulada

Existen dos tipos de mensajería:

**Mensajería Asíncrona:** Los mensajes son enviados de forma similar al correo postal, basta conocer su dirección de correo para poderlo enviar. El servicio de mensajería asíncrona más conocido y utilizado es el correo electrónico.

**Mensajería Síncrona:** Los mensajes son enviados como si se estuviese manteniendo una conversación, es decir, en tiempo real. Se le da alta sólo a aquellos contactos que interesen, por lo que se puede conocer si un contacto de la lista personal está conectado o no, para, posteriormente, comenzar una conversación. Cada servicio de mensajería asigna un identificador a cada cliente, para que así se puedan añadir los contactos de forma personalizada. No viene a sustituir a la mensajería asíncrona sino que es un complemento.

#### 1.1.1 Características de la Mensajería Asíncrona

La mensajería asíncrona entre sus características destaca:

- Envío de mensajes a un destinatario con posibilidad de enviar el mensaje en sí y archivos adjuntos (de cualquier tipo) mediante el protocolo SMTP (Simple Mail Transfer Protocol). Para ello es necesario configurar un servidor SMTP.
- Recepción de mensajes de un origen que pueden contener lo mismo que en el envío, el mensaje en sí y archivos adjuntos, y se realiza mediante el protocolo POP (Post Office

Protocol) o mediante el protocolo IMAP (Internet Message Access Protocol). Para ello es necesario configurar un servidor POP o IMAP.

## 1.1.2 Características de la Mensajería Síncrona

Los sistemas de mensajería síncrona tienen funciones básicas además de mostrar los usuarios que hay conectados y enviar mensajes.

### Contactos:

- **Registrar y borrar usuarios de la lista de contactos propia:** Al solicitar la inclusión en la lista de contactos, se puede enviar un mensaje explicando los motivos para la admisión.
- **Agrupar los contactos:** En grupos de familia, trabajo, facultad y amigos.
- **Se puede usar un avatar (imagen que identifique a la persona):** No tiene por que ser una foto personal.

### Chat:

Puede haber varios tipos de mensajes:

- **Aviso:** Lanza un mensaje solo. No es una invitación a mantener la conversación, solo se quiere enviar una información, un ejemplo de uso de este tipo sería el mensaje del día.
- **Invitación a chatear:** Se invita a mantener una conversación en tiempo real.
- **Mensaje emergente:** Es un aviso que se despliega unos segundos y se vuelve a cerrar. No requiere atención si no se desea. Sirve como aviso breve que moleste lo mínimo posible.

## **Charlas en grupo:**

- Se pueden crear salas (grupos de charla), públicas, privadas y también permanentes o que desaparezcan al quedarse sin usuarios.
- Restringir el acceso a salas mediante invitaciones certificadas, para invitar solo a quien la persona desee.

## **1.2 Antecedentes de Sistemas de Mensajería**

En la actualidad existen diferentes programas que ejecutan estos sistemas.

### **1.2.1 Antecedentes de mensajería asíncrona y síncrona**

#### **Servicios de correo electrónico**

##### **1.2.1.1 Gmail**

Es un servicio de correo electrónico con posibilidades POP3 e IMAP, proporcionado por la empresa estadounidense Google a partir del 1 de abril de 2004. Ha captado la atención de los medios de información por sus innovaciones tecnológicas, su capacidad, y por algunas noticias que alertaban sobre la violación de la privacidad de los usuarios.

#### **Entre sus características se reflejan:**

- Permite ver los mensajes en contexto. Si recibe respuesta a un mensaje que ha enviado o recibido, Gmail la mostrará automáticamente por orden cronológico junto con el original. Esta conversación aumenta a medida que envíe y reciba respuestas, lo que facilita el seguimiento de la conversación desde una misma ubicación.
- Una gran capacidad de almacenamiento.

- No tiene grandes avisos molestos, como Yahoo o Hotmail.
- No muestra ventanas emergentes, anuncios de banners no orientados ni advertencias para adquirir urgentemente espacio de almacenamiento adicional.
- Ofrece anuncios de texto muy relevantes junto al cuerpo de los mensajes. También vínculos a páginas Web relacionadas que puedan ser de interés.
- Corrector de correos electrónicos y está totalmente en español.

### 1.2.1.2 Yahoo

Es un servicio de webmail de Yahoo. Es uno de los mayores proveedores de correo electrónico de Internet. A finales de 2006, se anunció que Correo Yahoo sería código abierto y en mayo del 2007, Yahoo comenzó a ofrecer a sus usuarios un espacio de almacenamiento de correo ilimitado.

#### **Entre sus características se reflejan:**

- Almacenamiento de correo ilimitado.
- Protección contra spam y virus.
- La publicidad se muestra en pantalla mientras se trabaja con la cuenta de correo.
- A principios del 2006, Correo Yahoo! introduce alias en su repertorio de características. Los usuarios pueden añadir un simple alias de nombre de usuario conteniendo un punto para una cuenta preexistente.
- Correo Yahoo subraya las direcciones y números de teléfonos en el correo y permite al usuario añadirlos a la libreta de direcciones.

## Programas para leer y revisar correo:

### 1.2.1.3 Outlook Express

Outlook Express es un cliente de correo electrónico producido por Microsoft para sus plataformas Windows, existiendo también versiones para otras plataformas. Outlook Express se distribuye sin costo adicional con Microsoft Internet Explorer, formando parte así de los últimos sistemas operativos de la familia Windows. Es un programa derivado de Microsoft Outlook (que forma parte de Office), pero especializado en correo electrónico y noticias de red. Permite un mejor manejo de algunas características comunes en grupos de correo electrónico y noticias de red como el manejo de solo texto.

### 1.2.1.4 Evolution

Evolution es un cliente de correo electrónico que soporta la gran mayoría de protocolos usados en la actualidad, como STMP, POP3, IMAP. Con Evolution se puede leer, escribir, gestionar correos, contactos y calendario de eventos.

#### **Características:**

- Conectividad integrada con Novell GroupWise.
- Conectividad integrada con Microsoft Exchange.
- Mejora en cuentas de correo IMAP.
- Mejoras numerosas en el calendario.
- Capacidad para S/MIME, mejora de administración de contactos.
- Mejora en la integración con el escritorio GNOME.

### 1.2.1.5 ICQ (I see you)

Es un servicio de mensajería instantánea y el primero de su tipo en ser ampliamente utilizado en Internet, mediante el cual es posible chatear y enviar mensajes instantáneos. También permite el envío de archivos, videoconferencias y charlas de voz.

### 1.2.1.6 Instant Messenger

Es un programa de mensajería instantánea de American On Line, denominado habitualmente Instant Messenger.

#### Características:

- La conversación puede ser almacenada.
- Puedes mandar mensajes a los contactos aún cuando estén desconectados y se recibirán al conectarse.
- Se puede tener todos los amigos que se desee. No hay límite de contactos.

## 1.3 Tendencias y Tecnologías actuales

A continuación se dan a conocer las distintas tecnologías, herramientas, lenguajes y metodologías de soporte al desarrollo del Componente de Mensajería; detallando por cada una de ellas sus características fundamentales y justificando el por qué su selección.

### 1.3.1 Arquitectura Cliente - Servidor

La arquitectura cliente - servidor es una nueva tendencia en el desarrollo de redes locales, que tiene como objetivo optimizar el uso tanto del hardware como del software a través de la separación de funciones: el cliente que maneja la porción de la aplicación y el servidor que administra los procesos de almacenamiento y recuperación de los datos. Puede presentarse como uno a varios clientes y uno o

más servidores, junto con un sistema operativo y una plataforma de comunicación para formar un sistema cooperativo que permita la computación distribuida, el análisis y la presentación de datos.

## Características de la arquitectura Cliente - Servidor:

- El servidor presenta una interfaz única y bien definida a todos sus clientes.
- 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.
- Los cambios en el servidor no afectan al cliente.

### 1.3.2 Modelo Vista Controlador

El patrón Modelo Vista Controlador (MVC) permite separar los datos de una aplicación, la interfaz de usuario y la lógica de negocio en tres componentes distintos, esto proporciona múltiples vistas sobre un mismo modelo de datos. Este patrón se usa frecuentemente en aplicaciones Web donde se utilicen 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: [3]

- **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 cambiar el estado de estos 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.

Tanto la vista como el controlador dependen del modelo, el cual no depende de otros conceptos o clases. Esta separación permite construir y probar el modelo independientemente de la representación visual. En aplicaciones Web por ejemplo, la separación entre la vista (navegador) y el controlador que constituye los componentes del lado del servidor que manejan los requerimientos a través de HTTP (HyperText Transfer Protocol) está muy claramente definida. 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.

Además permite definir cómo organizar el modelo de diseño en capas, que pueden estar físicas y/o lógicamente distribuidas, lo cual quiere decir que los componentes de una capa sólo pueden hacer referencia a componentes en capas inmediatamente inferiores. De igual modo es importante porque simplifica la comprensión y la organización del desarrollo de sistemas complejos, reduciendo las dependencias de forma que las capas más bajas no son conscientes de ningún detalle o interfaz de las superiores.

### 1.3.3 Arquitectura en 3 Capas

La arquitectura en 3 capas es una forma de programación que separa la lógica de negocios de la lógica de diseño. La ventaja principal de este estilo, es que el desarrollo se puede llevar a cabo en varios niveles y en caso de algún cambio, solo afecta al nivel requerido sin tener que revisar entre códigos mezclados.

#### Capas:

- **Capa de presentación:** Es con la que interactúa el usuario (hay quien la denomina "capa de usuario"), presenta el sistema al usuario, le comunica la información y captura la información del usuario dando un mínimo de. Esta capa se comunica únicamente con la capa de negocio.
- **Capa de negocio:** Es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio e incluso de lógica del negocio pues es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los

resultados y con la capa de datos, para solicitar al gestor de base de datos para almacenar o recuperar datos de él.

- **Capa de datos:** Es donde residen los datos y es la encargada de acceder a los datos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

### 1.3.4 Arquitectura Orientada a los Servicios

SOA (Service Oriented Architecture) es una estrategia de integración, donde cualquier sistema pueda exponer servicios con funcionalidades bien definidas a la aplicación que lo requiera. De esta manera, una aplicación final simplemente se provee de un conjunto de estos servicios, añade su lógica particular y le presenta una interfaz al usuario final. Una visión interna de los servicios es que los mismos funcionan como aplicaciones independientes, teniendo sus propias reglas de negocio, datos, procedimientos, operaciones y administración. Exponen toda su funcionalidad basada en una interfaz a través de mensajes, lo que implica la carencia de una interfaz de usuario. SOA proporciona una metodología y un marco de trabajo para documentar las capacidades de negocio y puede dar soporte a las actividades de integración y consolidación. [4]

A continuación se relacionan las principales ventajas de la Arquitectura Orientada a los Servicios:

- **Reusabilidad de Servicios:** Reducción considerable de tiempos y costos de desarrollo de aplicaciones al utilizar servicios disponibles ya desarrollados, para resolver problemáticas comunes a otras aplicaciones. Aumentado por esta razón la robustez del nuevo sistema, al utilizarse software ya probado.
- **Interoperabilidad de aplicaciones:** Disminución de la complejidad en el proceso de integración, pues se interactúa con elementos que se abstraen de la tecnología y ubicación de los servicios. [5]

En un ambiente SOA, los nodos de la red hacen disponibles sus recursos a otros participantes en la

red como servicios independientes a los que se tiene acceso de un modo estandarizado. La mayoría de las definiciones de SOA identifican la utilización de servicios Web empleando SOAP (Simple Object Access Protocol) y WSDL (Web Services Description Language) en su implementación. La arquitectura orientada a servicios ofrece la ventaja de una integración fluida con la tecnología existente, sea cual sea la plataforma que se utilice. De este modo tecnologías que antes funcionaban de forma aislada, se combinan con otras aplicaciones esenciales para la operación diaria del proyecto.

### 1.3.5 Arquitecturas basadas en Componentes

La complejidad de los Sistemas computacionales actuales conlleva a lograr obtener la reutilización del software existente. El desarrollo de software basado en componentes permite reutilizar piezas de código preelaborado que posibilitan realizar diversas tareas, ganándose diversos beneficios como las mejoras a la calidad, la reducción del ciclo de desarrollo y el mayor retorno sobre la inversión.

En la Arquitectura basada en componentes la interfaz constituye el elemento básico de interoperabilidad. Cada componente debe describir de forma completa las interfaces que ofrece, así como las interfaces que requiere para su operación para su correcto funcionamiento con independencia de los mecanismos internos que utilice para soportar la funcionalidad de la interfaz. Características muy relevantes de esta arquitectura son la modularidad y la reusabilidad. Sin embargo, también se requiere robustez ya que los componentes han de operar en entornos mucho más heterogéneos y diversos. El desarrollo de software basado componentes es la evolución natural de la ingeniería software para mejorar la calidad, disminuir los tiempos de desarrollo y gestionar la creciente complejidad de los sistemas. Los beneficios más importantes de esta arquitectura son los siguientes:

[6]

- **Reutilización del software:** Nos lleva a alcanzar un mayor nivel de reutilización.
- **Simplifica las pruebas:** Permite que las pruebas sean ejecutadas probando cada uno de los componentes antes de probar el conjunto completo de componentes ensamblados.
- **Mayor calidad:** Dado que un componente puede ser construido y luego mejorado continuamente por un experto u organización, la calidad de una aplicación basada en componentes mejorará continuamente.

## 1.3.5.1 XML (eXtensible Markup Language)

XML es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Permite definir la gramática de lenguajes específicos. Por lo tanto, XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades. Es un lenguaje muy similar a HTML pero su función principal es describir datos y no mostrarlos como es el caso de HTML.

## 1.3.5.2 Servicios Web XML

Un Servicio es un sistema de software diseñado para soportar interacción máquina a máquina los cuales pueden ser descritos, publicados, localizados e invocados a través de la red utilizando protocolos estándar. Los Servicios Web utilizan SOAP como protocolo para invocar llamadas remotas por su simplicidad. SOAP permite la comunicación entre aplicaciones heterogéneas, de modo que clientes de diferentes plataformas o lenguajes de programación pueden comunicarse entre sí de manera satisfactoria. [7]

Paralelamente, alrededor de los Servicios Web existen una serie de protocolos y mecanismos adicionales para facilitar tareas como el descubrimiento de servicios distribuidos a lo largo de la red o UDDI (Universal Description, Discovery and Integration), una descripción del contenido de los mensajes o WSDL el cuál define la interfaz con el servicio especificando dónde está disponible el servicio y qué protocolo de comunicaciones utilizar para consumir este, lo que significa que un archivo WSDL define todo lo necesario para escribir un programa que interactúe con un Servicio Web, especifica su finalidad, funcionalidad, forma de uso y restricciones del servicio. [8]

Los Servicios Web basados en XML permiten que las aplicaciones compartan información y que además invoquen funciones de otras aplicaciones independientemente de cómo se hayan creado las aplicaciones, cuál sea el sistema operativo o la plataforma en que se ejecutan y cuáles los dispositivos utilizados para obtener acceso a ellas. Aunque los servicios Web XML son independientes entre sí, pueden vincularse y formar un grupo de colaboración para realizar una tarea determinada. La comunicación hacia y desde un servicio, es realizada utilizando mensajes y no llamadas a métodos, como se realiza en el paradigma de la Programación Orientada a Objetos. Estos mensajes deben

contener o referenciar toda la información necesaria para ser comprendidos internamente por la aplicación que lo ha solicitado. [9]

### **1.3.5.3 Alta cohesión y bajo acoplamiento**

Los patrones Alta cohesión y Bajo acoplamiento pertenecen a la familia de GRASP (General Responsibility Assignment Software Patterns). El primero plantea que debe haber pocas dependencias entre las clases; de tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de clases, potenciando la reutilización, y disminuyendo la dependencia entre las mismas; mientras que el segundo propone que cada elemento del diseño debe realizar una labor única dentro del sistema. Ejemplos de una baja cohesión son clases que tienen muchas responsabilidades. En todas las metodologías se considera la refactorización. Uno de los elementos a refactorizar son las clases saturadas de métodos. [10]

## **1.4 Herramientas, tecnologías, lenguajes y metodología de soporte al desarrollo**

### **1.4.1 Lenguajes utilizados para el proceso de desarrollo**

#### **1.4.1.1 PHP (Hypertext Pre-processor)**

PHP es un lenguaje de programación del lado del servidor, gratuito, ágil, posee integración con varias bibliotecas externas y una amplia documentación. Es un lenguaje interpretado empleado para la creación de páginas Web dinámicas, por lo que ofrece una solución simple y universal para las paginaciones dinámicas de fácil programación. De igual forma permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, PostgreSQL y Oracle. Posee la capacidad de ser ejecutado en la mayoría de los sistemas operativos, tales como Unix (Linux o Mac OS X) así como Windows, ya que existe en versión CGI módulo para Apache que presenta la capacidad de expandir su potencial utilizando la enorme cantidad de módulos denominados extensiones.

Posee análisis léxico para reconocer el tipo de dato almacenado en una variable haciéndose automáticamente, permitiéndole al usuario no tener que separar las variables de sus valores. Es un

producto de código abierto, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente. Este lenguaje cuenta con un conjunto de funciones de seguridad que previenen la inserción de órdenes dentro de una solicitud de datos desde el cliente evitando por ejemplo, la ocurrencia de la conocida inyección de código SQL. Finalmente permite la implementación del Paradigma de la Programación Orientada a Objetos, no requiere definición de tipos de variables y presenta tratamiento de errores.

### 1.4.1.2 JavaScript

JavaScript es un lenguaje de programación interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas Web, Javascript se puede incluir en cualquier documento HTML, o todo aquel que termine traduciéndose en HTML en el navegador del cliente. JavaScript no es un lenguaje orientado a objetos propiamente dicho, pues no dispone de herencia, es más bien un lenguaje basado en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad. Todos los navegadores interpretan el código JavaScript integrado dentro de las páginas Web.

#### **Ventajas de JavaScript:**

- Los programas escritos en este lenguaje no requieren de mucha memoria ni tiempo adicional de transmisión, por ser pequeños y compactos.
- No requiere un tiempo de compilación; ya que los scripts se pueden desarrollar en un período de tiempo relativamente corto.
- Es independiente de la plataforma hardware o sistema operativo y funciona correctamente siempre y cuando exista un navegador con soporte JavaScript.

### 1.4.1.2.1 AJAX

AJAX (Asynchronous JavaScript And XML) es una metodología para el desarrollo de aplicaciones Web interactivas o RIA (Rich Internet Applications) que no constituye una tecnología en sí, sino una combinación de cuatro tecnologías ya existentes:

- XHTML (o HTML) y hojas de estilos en cascada (CSS) para el diseño que acompaña a la información.
- DOM (Document Object Model) accedido con un lenguaje de scripting por parte del usuario, especialmente implementaciones como JavaScript, para mostrar e interactuar dinámicamente con la información presentada.
- El objeto XMLHttpRequest para intercambiar datos asincrónicamente con el servidor Web. En algunos frameworks y en algunas situaciones concretas, se usa un objeto iframe en lugar del XMLHttpRequest para realizar dichos intercambios.
- XML es el formato usado comúnmente para la transferencia de vuelta al servidor, aunque cualquier formato puede funcionar, incluyendo HTML preformateado, texto plano, JSON y hasta EBML. [11]

La misma se ejecuta en el cliente, es decir, en el navegador de los usuarios y mantiene comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla. Esto significa aumentar la interactividad, velocidad y usabilidad.

### 1.4.1.2.2 Librería de componentes de diseño Ext JS 2.0

Es una librería de componentes que facilita las herramientas necesarias para la creación de aplicaciones Web con excelentes gráficos ya que posee una considerable colección de elementos para el diseño de interfaces. Brinda soporte para construir interfaces gráficas dinámicas y de fácil interacción con el usuario, para comunicar datos de forma asíncrona con el servidor, además para la interacción con diversos navegadores como: Internet Explorer, Firefox, Safari y Opera. Es importante

señalar que la Ext JS 2.0 tiene dos tipos de licencias, LGPL (Open Source) y la comercial, esta última es obligatoria si se desea obtener soporte.

Permite realizar operaciones de bajo nivel como: manejo de eventos, consultas al DOM, manejo del CSS, entre otras funciones que se realicen con la librería que se seleccione. Por lo que Ext actúa como puente entre las librerías JavaScript más usadas (Prototype, JQuery, YUI). Además ofrece algunas ventajas como son: el Modelo de Componentes, Modelo de Contenedores, las Capas, las Plantillas y las Vistas de Datos. Además soporta los navegadores: Internet Explorer 6, Firefox 1.5, Safari 2 y Opera 9 o sus versiones superiores.

Ext JS 2.0 implementa patrones de diseño como:

Patrones creacionales dentro del mismo se encuentran los patrones Builder, el cual abstrae el proceso de creación de un objeto complejo, centralizando dicho proceso en un único punto y el patrón Singleton, que garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. Cuenta además con los patrones estructurales como el Flyweight, que reduce la redundancia cuando gran cantidad de objetos poseen idéntica información y también cuenta con los patrones de comportamiento como Observer el cual define una dependencia de uno-a-muchos entre objetos, de forma que cuando un objeto cambie de estado se notifique y actualicen automáticamente todos los objetos que dependen de él.

## 1.4.2 Sistema de Gestión de Bases de Datos

Los Sistemas de Gestión de Bases de Datos (SGBD) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. El propósito general de los sistemas de gestión de base de datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante, para un buen manejo de datos.

### **Ventajas:**

- Facilidad de manejo de grandes volúmenes de información.

- Gran velocidad en muy poco tiempo.
- Independencia del tratamiento de información.
- Seguridad de la información (acceso a usuarios autorizados), protección de información, de modificaciones, inclusiones o consultas.
- No hay duplicidad de información, comprobación de información en el momento de introducir la misma.
- Integridad referencial al terminar los registros.

## 1.4.2.1 PostgreSQL 8.2.

PostgreSQL es uno de los SGBD con filosofía de código abierto, liberado bajo la licencia BSD (Berkeley Software Distribution) que aproxima sus datos a un modelo objeto - relacional. Este gestor de bases de datos además de ser potente, soporta distintos tipos de datos: además del soporte para los tipos primitivos o nativos, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes y cadenas de bits, también permite la creación de tipos propios e incorpora una estructura de datos array que constituye un conjunto o agrupación de variables del mismo tipo cuyo acceso se realiza por índices. Entre sus principales ventajas constan las siguientes:

- **Añade funciones de diversa índole:** Manejo de fechas, geométricas y orientadas a operaciones con redes.
- **Permite la declaración de funciones propias:** Así como la definición de disparadores.
- **Soporta el uso de índices, reglas y vistas:** Incluye herencia entre tablas (aunque no entre objetos, ya que no existen) por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.
- **Admite la gestión de diferentes usuarios:** Como también los permisos asignados a cada uno de ellos. [12]

PostgreSQL se puede instalar un número ilimitado de veces sin temor de sobrepasar la cantidad de licencias. Soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos. Posee confiabilidad, seguridad y flexibilidad. PostgreSQL mediante un sistema de control concurrente multiversión posibilita que mientras un proceso realiza una acción en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos.

## 1.4.2.2 Servidor Web Apache

Apache es un servidor Web posible de utilizar en distintas plataformas y entornos. Es altamente configurable de diseño modular, posibilitando que los administradores de sitios Web puedan elegir los módulos que serán incluidos y ejecutados en el servidor.

### Características de Apache:

- Es una tecnología gratuita y de código abierto, lo que proporciona transparencia en todo el proceso de instalación.
- Es prácticamente universal, por su disponibilidad en multitud de sistemas operativos.
- Posee una alta configurabilidad en la creación y gestión de logs, de este modo es posible tener un mayor control sobre lo que sucede en el servidor.

Este servidor Web tiene una fácil integración con varios lenguajes de programación como: Java, Perl y especialmente PHP. Dicha relación a dado lugar al desarrollo de aplicaciones como el APPSERV y XAMPP los cuales instalan el Apache y el PHP configurados para su uso.

## 1.4.3 HyperText Transfer Protocol (HTTP)

HTTP es un protocolo del nivel de aplicación usado para la transferencia de información entre sistemas, de forma clara y rápida. Se basa en un paradigma de peticiones y respuestas entre un cliente y un servidor. Un cliente envía una petición en forma de método y una versión de protocolo seguida de los modificadores de la petición y el servidor contesta con una línea de estado que incluye

la versión del protocolo y un código que indica éxito o error, seguido de la información del servidor en forma de mensaje y un posible contenido. Generalmente es el cliente el que inicia la comunicación HTTP y consiste en la petición de un recurso del servidor. La misma puede concebirse de forma directa al servidor o a través de intermediarios. [13]

Este protocolo permite transmitir una amplia cantidad de información de diferentes tipos y formatos, además de multimedia, audio, video, gráficos, programas e imágenes. Es un protocolo seguro y ágil. Presenta ventajas como: no almacenamiento de información sobre conexiones anteriores que hayan establecido, ya que trata cada conexión como independiente del resto de las posibles conexiones que se puedan realizar, después del establecimiento de la conexión actual.

### **1.4.4 Proceso Unificado de Desarrollo (RUP)**

El Proceso Unificado es un proceso de software genérico que puede ser utilizado para una gran cantidad de tipos de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de competencia y diferentes tamaños de proyectos. Provee un enfoque disciplinado en la asignación de tareas y responsabilidades dentro de una organización de desarrollo. Su meta es asegurar la producción de software de muy alta calidad que satisfaga las necesidades de los usuarios finales, dentro de un calendario y presupuesto predecible. Su ciclo de vida se caracteriza por ser dirigido por casos de uso, centrado en la arquitectura, iterativo e incremental.

#### **1.4.4.1 UML (Lenguaje Unificado de Modelado)**

UML es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema de software. Se utiliza para detallar los artefactos en el sistema, documentar y construir. Ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, esquemas de bases de datos y componentes de software reutilizables.

#### **UML permite:**

- Visualizar gráficamente un sistema de manera que otros puedan entenderlo.

- Especificar cuáles son las características de un sistema antes de su construcción.
- Construir sistemas diseñados a partir de modelos especificados.
- Documentar los elementos gráficos del sistema desarrollado para futuras revisiones.

## 1.5 Herramienta a utilizar

### 1.5.1 Rational Rose Enterprise Edition 2003

Herramienta que incluyen soporte para UML y constituye la mejor elección para el ambiente de modelado que soporte la generación de código a partir de modelos en Ada, ANSI C++, C++, CORBA, Java/J2EE, Visual C++ y Visual Basic. Como todos los demás productos, proporciona un lenguaje común de modelado para el equipo que facilita la creación de software de calidad más ágilmente. Esta herramienta de modelado tan empleada actualmente posee una capacidad de análisis de código eficiente, el modelado UML para trabajar en diseños de bases de datos tiene la capacidad de representar la integración de los datos y los requerimientos de aplicación a través de diseños lógicos y físicos, además de la posibilidad que brinda crear definiciones de tipo de documento XML (DTD) para el uso en la aplicación. [14]

Se utiliza para modelar un sistema antes de proceder a construirlo y cubre todo el ciclo de vida de un proyecto: concepción y formalización del modelo, construcción de los componentes, transición a los usuarios y certificación de las distintas fases del proceso de desarrollo de software. De igual modo es una herramienta poderosa de modelado visual para el análisis y diseño de sistemas.

### 1.5.2 Zend Studio for Eclipse

Zend Studio for Eclipse es desarrollado por la empresa Zend para apoyar el desarrollo PHP por medio de la IDE Eclipse. El software está destinado a todo tipo de desarrolladores en PHP, pero principalmente será de gran utilidad para aquellos profesionales en la materia, debido a las posibilidades que este brinda. El mismo permite mejorar la calidad de los proyectos en PHP, agilizar los ciclos de desarrollos, incluye herramientas para edición, debugging, análisis, optimización y bases de datos, e incluso soportando los procesos del desarrollo a través de la programación ágil. Está basado en Zend Studio y el proyecto Eclipse PHP Developers Tools (PDT), y se convierte así en el IDE

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

---

para PHP más potente del mercado, ofreciendo así al desarrollador profesional de PHP la potencia de Zend Studio y el soporte multilenguaje de Eclipse y su enorme conjunto de extensiones. [15]

Aporta las ventajas de Eclipse/PDT mejoradas como son: una comunidad de numerosos usuarios y desarrolladores, simplificación de la instalación, un gran número de plugins, soporte multilenguaje en una única herramienta (Eclipse), coloreado de sintaxis PHP, autocompletado de código e inspección de métodos y atributos, además de soporte avanzado de depuración de scripts PHP (permitiendo debugging local como remoto).

En este capítulo se analizó todo el contexto sobre el análisis del estado del arte de sistemas de notificación y mensajerías, así como de las tendencias, tecnologías, lenguajes y herramientas actuales para realizar el proceso de diseño del Componente de Notificación y Mensajería, especificando el por qué de su uso.

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Este capítulo plasma la propuesta de solución para el Componente de Notificación y Mensajería. Se determina desarrollar un Modelo de Dominio por la poca estructuración de los procesos de Negocio, donde se representa un marco conceptual y las relaciones entre estas definiciones. Se justifican todos los usuarios que interactuarán con el Componente de Notificación y Mensajería, también se exponen los primeros casos de uso, se estructura el Diagrama de Casos de Uso del Sistema y se agrupan además los requerimientos funcionales y no funcionales.

### 2.1 Modelo de Dominio

El modelo de Dominio o Modelo Conceptual es una representación visual de los conceptos u objetos del mundo real. Representa clases conceptuales del dominio del problema. El modelo desarrollado no se trata de un conjunto de diagramas que describen clases de software u objetos de software con responsabilidades, sino que puede considerarse como un diccionario visual de las abstracciones relevantes, vocabulario e información del dominio. El modelo de dominio es global, es decir se realiza para todos los casos de uso y no para uno en particular.

**El Modelo de Dominio se representa mediante UML como un Diagrama de Clases en los que se muestra:**

- Conceptos u objetos del dominio del problema: clases conceptuales.
- Asociaciones entre las clases conceptuales.
- Atributos de la clase conceptuales.

En el Modelo de Dominio no se muestra comportamiento. Las clases conceptuales pueden tener atributos pero no métodos. Cualquiera que sea la solución de casos de uso que se haya elegido, los conceptos e ideas propias del dominio del problema son las mismas; un mismo modelo contempla cualquiera de las soluciones analizadas. En el desarrollo del Componente de Notificación y Mensajería debido a que no pudieron ser identificados los procesos de negocio se decide realizar un Modelo de Dominio, por la poca estructuración y fronteras del mismo.

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

---

### 2.1.1 Conceptos fundamentales

Se proporciona un marco conceptual para mejor comprensión del Modelo de Dominio con las definiciones identificadas en el proceso de desarrollo del Componente de Notificación y Mensajería.

**Usuario:** Persona o Sistema que tendrá el acceso a realizar las funcionalidades que forman parte del Componente de Notificación y Mensajería.

**Persona:** Usuario que interactúa con la aplicación enviando mensajes Síncronos o Asíncronos a otros usuarios persona.

**Sistema:** Usuario capaz de enviar mensajes Asíncronos a los usuarios persona.

**Mensaje Síncrono:** Tipo de mensaje enviado en tiempo real, es decir, como si se estuviera estableciendo una conversación.

**Mensaje Asíncrono:** Tipo de mensaje enviado de forma similar al correo postal.

**Sala de Conferencia:** Salas de conversación entre varios usuarios persona, para realizar reuniones u otras actividades de importancia.

**Historial:** Mensajes guardados en la aplicación enviados por los usuarios.

**Listas de Usuarios:** Carpetas creadas por los usuarios personas en la mensajería asíncrona donde se recoge un grupo de otros usuarios persona.

**Contacto:** Carpetas creadas por los usuarios personas en la mensajería asíncrona donde se recoge datos adicionales de un solo usuario persona.

**Organismo:** Grupos a los que pertenecen los usuarios persona en la mensajería síncrona.

## 2.1.1.1 Diagrama del Modelo de Dominio

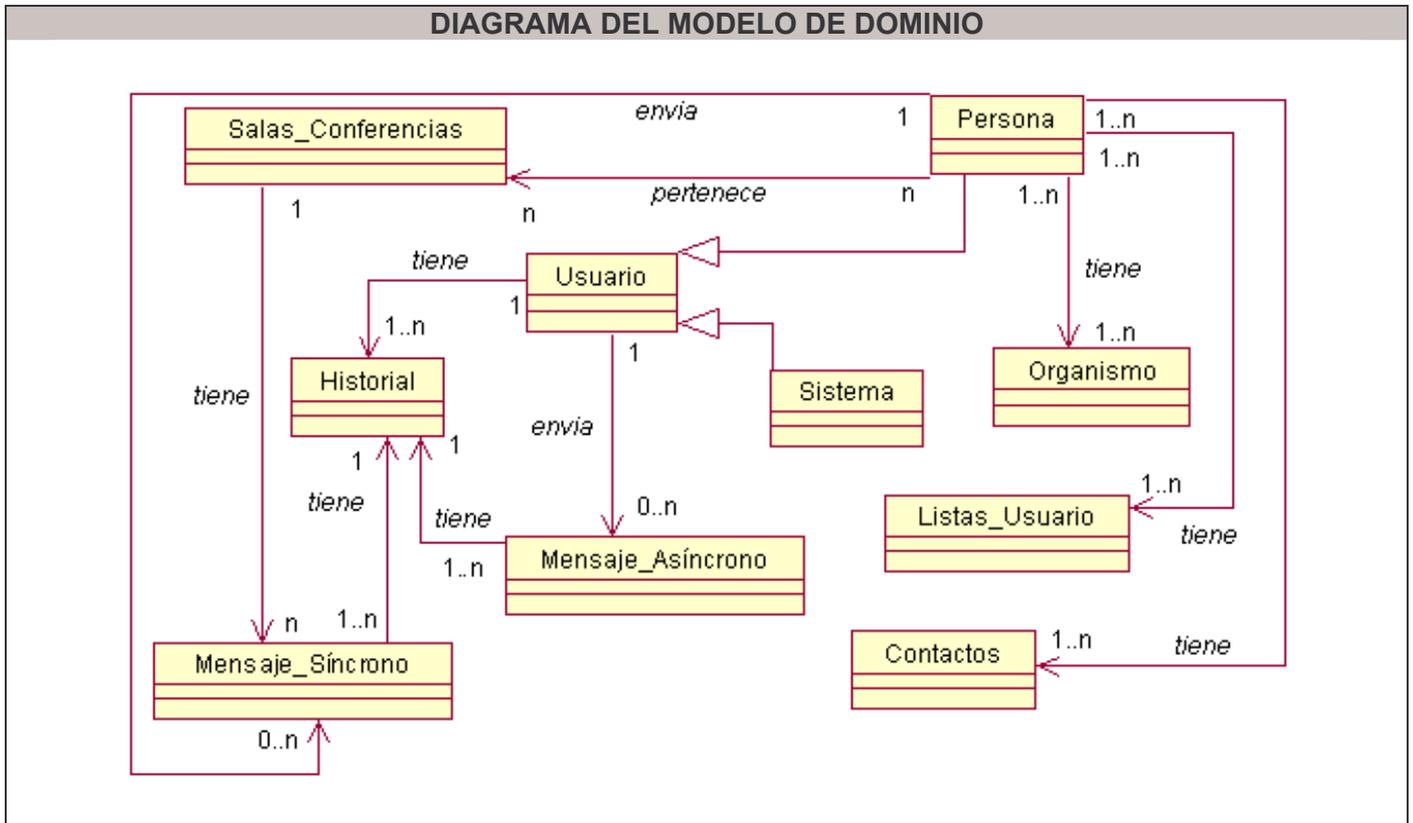


Figura 2.1 Diagrama del Modelo de Dominio.

## 2.2 Propuesta de Sistema

Un sistema está comprendido por varios elementos: hardware, software y personas; la ingeniería de sistemas ayuda a traducir las necesidades del cliente en un modelo del sistema que utiliza estos elementos. Este proceso comienza tomando la visión global definida en el Modelo de Dominio realizándose un análisis del mismo con el fin de establecer todos los requerimientos básicos, cada uno de los conceptos o definiciones identificados serán objetos de estudio individualmente. La ingeniería de software demanda una intensa comunicación entre el cliente y los desarrolladores, realizándose un grupo de actividades denominada gestión de requerimientos. Una vez que los requerimientos hayan sido identificados el Modelo del Sistema puede ser realizado.

### **2.2.1 Especificación de los requerimientos del software**

Una especificación de requerimientos de software define de forma precisa el producto de software que se va a construir. Es una descripción completa del comportamiento del sistema a desarrollar. Los requerimientos del software define de forma precisa el producto de software, para esto se definen una serie de requerimientos funcionales que el sistema debe ser capaz de cumplir, además se detallan los requerimientos no funcionales que son cualidades del sistema e incluye un conjunto de casos de uso que describen todas las interacciones que los usuarios tendrán con el software.

#### **2.2.1.1 Requerimientos Funcionales**

Los requerimientos funcionales describen las acciones que un sistema debe cumplir. Por esto los mismos deben estar claros y libres de ambigüedades. Por lo general se describen mejor a través del modelo de Caso de uso y los Casos de uso como tal. Este tipo de requerimiento especifica algo que el sistema entregado debe ser capaz de realizar, por lo tanto los requerimientos funcionales especifican el comportamiento de entrada y salida del sistema, es decir define el comportamiento interno del software: cálculos, detalles técnicos, manipulación de datos y otras funcionalidades específicas que muestran cómo los casos de uso serán llevados a la práctica. A continuación se listan los requerimientos funcionales que tendrá el Componente de Notificación y Mensajería.

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

LISTA DE REQUERIMIENTOS FUNCIONALES	
<b>RF1</b> Autenticar <b>RF2</b> Enviar Mensaje Síncrono <b>RF3</b> Eliminar Mensaje Síncrono <b>RF4</b> Enviar Mensaje Asíncrono <b>RF5</b> Eliminar Mensaje Asíncrono <b>RF6</b> Crear Lista de Usuario <b>RF7</b> Eliminar Lista de Usuario <b>RF8</b> Modificar Lista de Usuario <b>RF9</b> Adicionar Usuario a las lista <b>RF10</b> Crear Grupo de usuarios <b>RF11</b> Eliminar Grupo de usuarios	<b>RF12</b> Modificar Grupo de usuarios <b>RF13</b> Crear Contacto <b>RF14</b> Eliminar Contacto <b>RF15</b> Modificar Contacto <b>RF16</b> Buscar Lista Usuario <b>RF17</b> Buscar Contacto <b>RF18</b> Buscar Usuario <b>RF19</b> Buscar Organismo <b>RF20</b> Crear Sala de Reunión <b>RF21</b> Listar Sala de Reunión

Tabla 2.1 Lista de Requerimientos Funcionales.

A continuación se representa la lista de los casos de uso del sistema, que no es más que la forma en que los usuarios interactúan con el sistema. Los casos de uso del sistema son la agrupación de los requerimientos funcionales mencionados anteriormente. En tal sentido los casos de uso del sistema identificados son los siguientes:

DEFINICIÓN DE CASOS DE USO	
<b>CUS1_Autenticar</b>	<b>CUS7_Buscar_Lista_Usuario</b> <b>RF13</b> Buscar Listas Usuario
<b>CUS2_Gestionar_Mensaje_Síncrono</b> <b>RF2</b> Enviar Mensaje Síncrono <b>RF3</b> Eliminar Mensaje Síncrono	<b>CUS8_Buscar_Contacto</b> <b>RF17</b> Buscar Contacto
<b>CUS3_Gestionar_Mensaje_Asíncrono</b> <b>RF4</b> Enviar Mensaje Asíncrono	<b>CUS9_Buscar_Usuario</b> <b>RF18</b> Buscar Usuario

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

<b>RF5</b> Eliminar Mensaje Asíncrono	
<b>CUS4_Gestionar_Lista_Usuario</b> <b>RF6</b> Crear Lista de Usuarios <b>RF7</b> Eliminar Lista de Usuarios <b>RF8</b> Modificar Lista de Usuarios <b>RF9</b> Adicionar Usuarios a la lista	<b>CUS10_Buscar_Organismo</b> <b>RF19</b> Buscar Organismo
<b>CUS5: Gestionar_Grupo_Usuario</b> <b>RF10</b> Crear Grupo de usuarios <b>RF11</b> Eliminar Grupo de usuarios <b>RF12</b> Modificar Grupo de usuarios	<b>CUS11_Gestionar_Sala_Reunión</b> <b>RF20</b> Crear Sala de Reunión <b>RF21</b> Visualizar Sala de Reunión
<b>CUS6_Gestionar_Contacto</b> <b>RF13</b> Crear Contacto <b>RF14</b> Eliminar Contacto <b>RF15</b> Modificar Contacto	

Tabla2.2 Definición de Casos de Uso.

### 2.2.1.2 Requerimientos No Funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Son los requerimientos que imponen restricciones al diseño o funcionamiento del sistema; tal como requerimientos de funcionamiento, estándares de calidad, o requerimientos del diseño. Son aspectos del sistema visibles para el usuario, que no están relacionados de forma directa con el comportamiento funcional del sistema. Especifican propiedades que debe tener el sistema, como restricciones de ambiente, transición, performance, dependencias de plataformas y confiabilidad.

### 2.2.1.2.1 Rendimiento

Los requerimientos clasificados en esta categoría están relacionados con tiempos de respuesta estimados, requeridos y esperados para la ejecución en línea de procesos del sistema, teniendo como base la plataforma tecnológica y escenarios específicos a los que en teoría el sistema estará expuesto y frente a los que deberá responder.

**RNF1** Garantizar una buena velocidad de navegación y que las respuestas del sistemas tengan una velocidad aceptable para los usuarios.

### 2.2.1.2.2 Seguridad

Requerimientos relacionados con la confidencialidad de los datos en la transmisión y en el almacenamiento, junto con las necesidades del sistema para evitar intrusiones no autorizadas al mismo y la capacidad para seguir eventos que comprometan esta seguridad a través del tiempo.

**RNF2** Contar con un mecanismo de seguridad basado en los principios de Autenticación, Autorización y Auditoría.

**RNF2.1** La autenticación será la primera acción del usuario en el sistema.

### 2.2.1.2.3 Disponibilidad

Requerimientos relacionados con la capacidad del sistema para estar disponible para los usuarios, esto se refleja en el tiempo estimado, esperado y requerido por el usuario para que el sistema esté disponible.

**RNF3** Los usuarios autorizados tendrán acceso a la información en todo momento.

### 2.2.1.2.4 Portabilidad

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

---

Requerimientos que describen la capacidad del sistema para migrar de una plataforma hardware a otra sin que esto represente mayores traumatismos para el cliente del mismo, teniendo en cuenta los requisitos técnicos presentados y las generalidades naturales de configuración del sistema.

**RNF6** Garantizar que el sistemas sea compatible con distintos navegadores y sistemas operativos.

### 2.2.1.2.5 Capacidad

Requerimientos de tecnología que permitirán al sistema ejecutar sus funciones de manera que responda a las expectativas del usuario en términos de eficiencia y eficacia en la respuesta de las operaciones. Se tienen en cuenta también las características, que harán que sea posible que el sistema funcione de manera estable durante un tiempo determinado, planeando crecimiento del mismo.

**RNF7** Soportar varias conexiones al mismo tiempo.

### 2.2.1.2.6 Usabilidad

Requerimientos que determinan las características generales de la capa de presentación del sistema en cuanto a las características de diseño gráfico de la misma, además de las facilidades para que el uso del sistema por parte del usuario final.

**RNF8** La interfaz de usuario debe de realizarse con un diseño sencillo y fácil de usar.

### 2.2.1.2.7 Reusabilidad

Requerimientos que consideran la capacidad de los componentes del sistema de prestar servicios a otros sistemas, de tal manera que el componente como valor adicional al cumplimiento de sus funciones pueda prestar sus servicios a otros sistemas sin que esto implique modificaciones o redefiniciones en dicho componente.

**RNF9** Proporcionar servicios públicos para el envío de mensajería asíncrona a sistemas externos mediante Servicios Web XML.

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

---

**RNF10** Describir los Servicios Web teniendo en cuenta el estándar WSDL (Web Services Description Language).

### 2.3 Modelo de Casos de Uso del Sistema

El Modelo de Caso de Uso del Sistema proporciona la entrada fundamental para el análisis, diseño e implementación, por lo que mientras mejor estructurado se encuentre el mismo, más sencillo será el proceso de construcción de sistema de software. El modelado de casos de uso es una técnica efectiva y a la vez simple para modelar los requerimientos del sistema desde la perspectiva del usuario. Este Modelo permite que los desarrolladores de software y los clientes lleguen a un entendimiento sobre los requerimientos, es decir, sobre las condiciones y posibilidades que debe cumplir el sistema.

#### 2.3.1 Definición de los actores

El Modelo de Casos de Uso describe, además, lo que hace el sistema para cada tipo de usuario. Estos actores pueden estar representados por un usuario en concreto, humano u otro sistema, que interactúe de alguna forma con el sistema de desarrollo. A continuación se definen los actores del sistema:

ACTOR	JUSTIFICACIÓN
<b>Usuario</b>	Persona o Sistema que tendrán el acceso a realizar las funcionalidades que forman parte del Componente de Notificación y Mensajería.
<b>Persona</b>	Usuario que interactúa con la aplicación enviando mensajes Síncronos o Asíncronos a otros usuarios persona.
<b>Sistema</b>	Usuario capaz de enviar mensajes Asíncronos a los usuarios personas.
<b>Componente de Seguridad</b>	Sistema externo que brinda servicios de búsqueda de usuarios persona y autenticación.

Tabla 2.3 Descripción de los actores del Sistema.

### 2.3.1.1 Vista especializada de los actores de Sistema

En el siguiente diagrama, se puede observar como se separan *Persona* y *Sistema* debido a sus diferencias, pero que a su vez se comportan como Usuarios. En la propuesta de actores del Componente de Notificación y Mensajería, se separan ya que *Persona* es el usuario que interactúa con la aplicación enviando mensajes a otros usuarios personas y *Sistema* es el usuario capaz de enviar notificaciones a los usuarios personas.

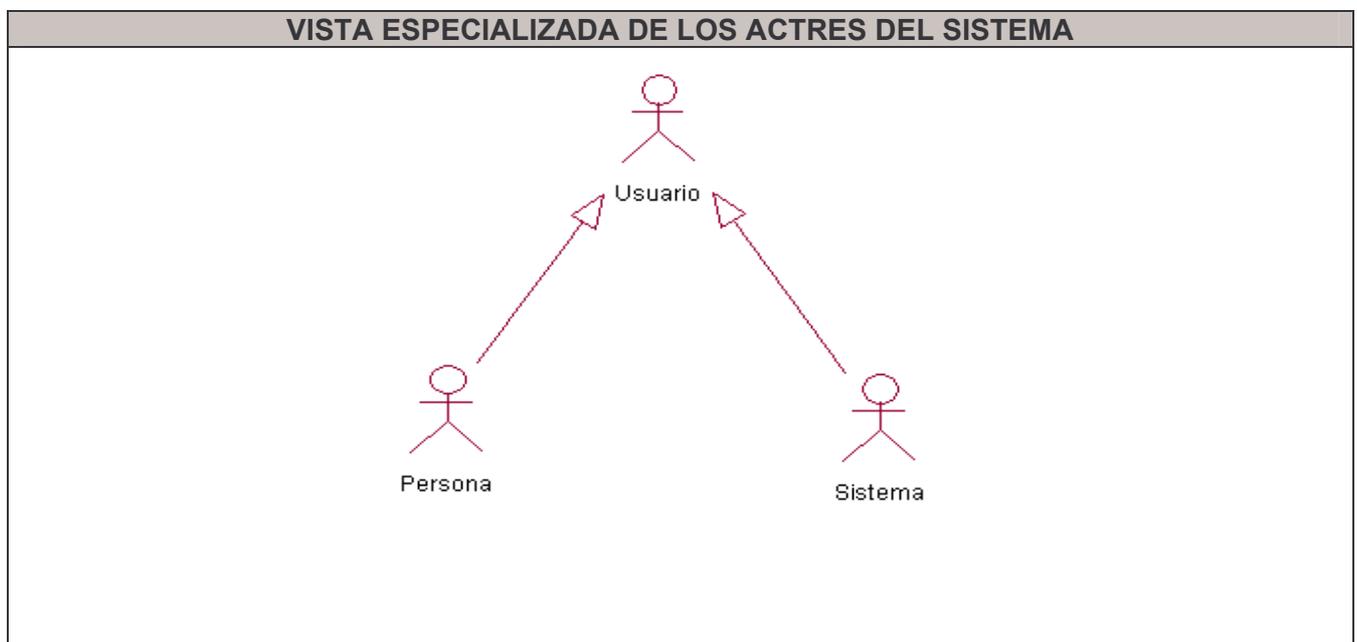


Figura 2.2 Vista especializada de los actores del Sistema.

### 2.3.2 Diagrama de Casos de Uso del Sistema

A continuación se muestra el Diagrama de Casos de Uso observándose las relaciones que se establecen entre los actores y los casos de uso, es decir son descritas las propiedades estructurales del sistema.

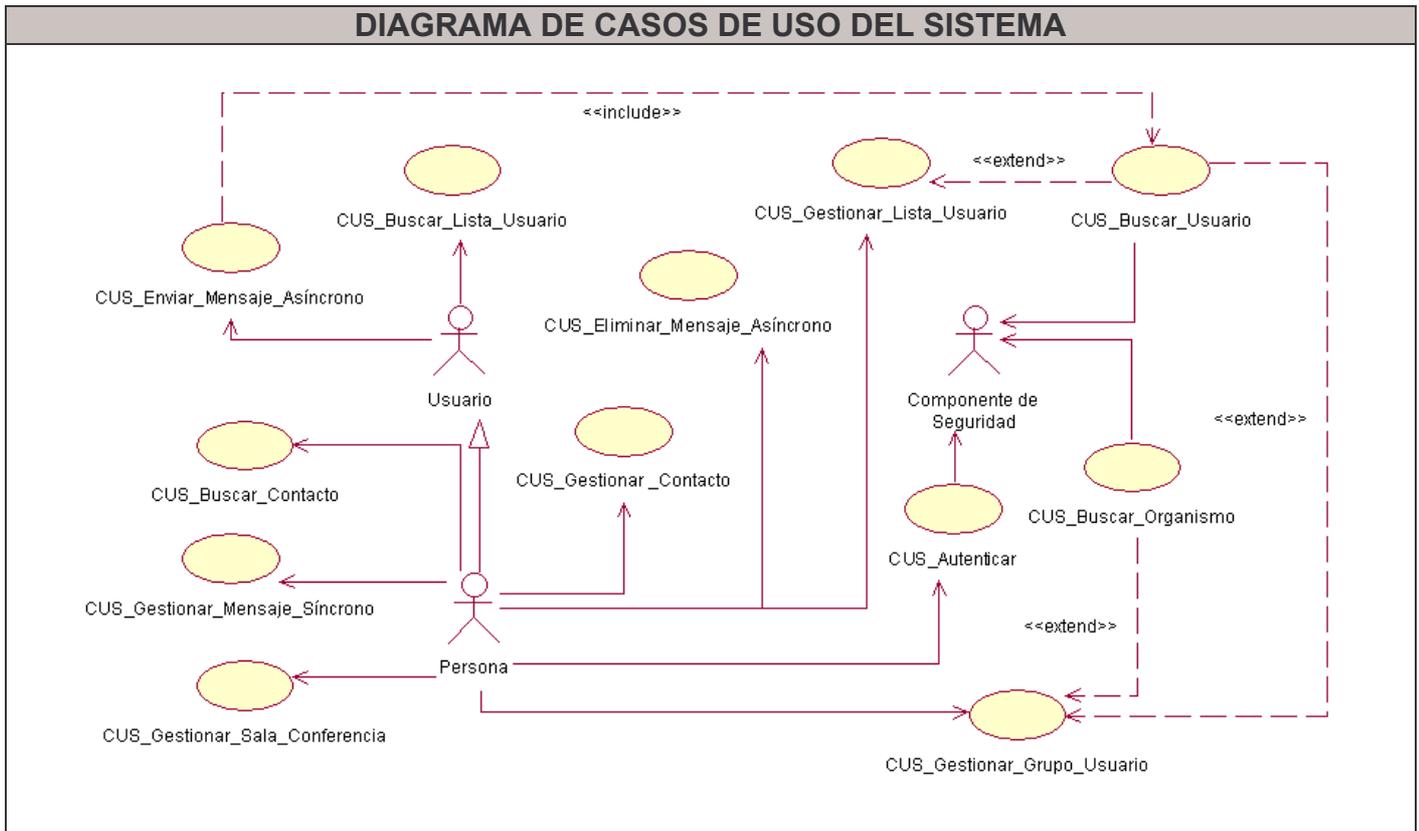


Figura 2.3 Diagrama de Casos de Uso del Sistema.

### 2.3.2.1 Descripción textual de los Casos de Uso

La descripción textual de un caso de uso, representa los procesos o flujos de funcionalidades que son objeto de automatización mediante el mismo, incluyendo cómo comienza, termina e interactúan los actores con este. A continuación se proporciona una descripción breve de los casos de uso identificados para el Componente de Notificación y Mensajería.

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

### CUS1\_Autenticar

<b>CUS1_Autenticar</b>	
<b>Propósito:</b>	Permitir la entrada al sistema.
<b>Actores:</b>	<b>Usuario</b> (Inicia)
<b>Resumen:</b>	El caso de uso se inicia cuando un usuario persona accede al Componente de Notificación y Mensajería, introduciendo su nombre de usuario y contraseña.
<b>Precondición:</b>	El nombre de usuario y contraseña deben ser correctas.
<b>Poscondición:</b>	El usuario persona accede al sistema.
<b>Referencias:</b>	<b>Componente de Seguridad, RF1</b>
<b>Prioridad:</b>	Crítico

Tabla 2.4 Descripción textual *CUS1\_Autenticar*.

### CUS2\_Gestionar\_Mensaje\_Síncrono

<b>CUS2_Gestionar_Mensaje_Síncrono</b>	
<b>Propósito:</b>	Permitir que el usuario persona interactúe con la mensajería síncrona.
<b>Actores:</b>	<b>Usuario</b> (Inicia)
<b>Resumen:</b>	El caso de uso se inicia cuando un usuario persona desea enviar un mensaje síncrono a otros usuarios persona o eliminar su historial de mensajes. El caso de uso finaliza una vez que se realicen una de las acciones mencionadas.
<b>Precondición:</b>	El usuario persona debe acceder a la mensajería síncrona y tener el nombre de usuario los destinatarios.
<b>Poscondición:</b>	El usuario accede a la mensajería síncrona.
<b>Referencias:</b>	<b>RF2, RF3</b>
<b>Prioridad:</b>	Crítico

Tabla 2.5 Descripción textual *CUS2\_Gestionar\_Mensaje\_Síncrono*

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

### CUS3\_Gestionar\_Mensaje\_Asíncrono

<b>CUS3_Gestionar_Mensaje_Asíncrono [include: CUS_Buscar_Usuario ]</b>	
<b>Propósito:</b>	Permitir que el usuario persona interactúe con la mensajería asíncrona.
<b>Actores:</b>	<b>Usuario</b> (inicia)
<b>Resumen:</b>	El caso de uso se inicia cuando un usuario persona desea redactar o enviar un correo electrónico a otros usuarios. El caso de uso finaliza una vez que se realicen una de las acciones mencionadas.
<b>Precondición:</b>	El usuario debe acceder a la mensajería asíncrona y tener la dirección de correo del o los destinatarios.
<b>Poscondición:</b>	El usuario accede a la mensajería asíncrona.
<b>Referencias:</b>	<b>RF4, RF5, RF13 ,RF15</b>
<b>Prioridad:</b>	Crítico

Tabla 2.6 Descripción textual CUS3\_Gestionar\_Mensaje\_Asíncrono [include: CUS\_Buscar\_Usuario]

### CUS4\_Gestionar\_Lista\_Usuario

<b>CUS4_Gestionar_Lista_Usuario [extend: CUS_Buscar_Usuario]</b>	
<b>Propósito:</b>	Permitir que el usuario persona pueda gestionar sus listas de usuarios.
<b>Actores:</b>	<b>Persona</b> (inicia)
<b>Resumen:</b>	El caso de uso se inicia cuando un usuario persona desea crear listas de usuarios, adicionar usuarios persona o modificar estos grupos. El caso de uso finaliza una vez que se realice una de las acciones mencionadas.
<b>Precondición:</b>	El usuario debe conocer algún dato los que desea adicionar a su carpeta.
<b>Poscondición:</b>	El usuario realiza las gestiones de la listas de distribución.
<b>Referencias:</b>	<b>F6, RF7, RF8, RF9, RF18</b>
<b>Prioridad:</b>	Crítico

Tabla 2.7 Descripción textual CUS4\_Gestionar\_Lista\_Usuario [extend: CUS\_Buscar\_Usuario]

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

### CUS5\_Gestionar\_Grupo\_Usuario

<b>CUS5_Gestionar_Grupo_Usuario [incluye CUS_Buscar_Usuario, CUS_Buscar_Organismo]</b>	
<b>Propósito:</b>	Permitir que el usuario persona pueda gestionar sus grupos de usuarios.
<b>Actores:</b>	<b>Persona</b> (inicia)
<b>Resumen:</b>	El caso de uso se inicia cuando un usuario persona desea crear grupos de usuarios, adicionar persona al grupo, eliminar un grupo o un usuario de el o modificar estos grupos. El caso de uso finaliza una vez que se realicen una de las acciones mencionadas.
<b>Precondición:</b>	El usuario debe conocer el nombre de los usuarios persona que desee agregar.
<b>Poscondición:</b>	El usuario realiza las gestiones de los grupos de usuarios.
<b>Referencias:</b>	<b>RF10, RF11, RF12, RF19</b>
<b>Prioridad:</b>	Crítico

Tabla 2.8 Descripción textual CUS5\_Gestionar\_Grupo\_Usuario [incluye: CUS\_Buscar\_Usuario, CUS\_Buscar\_Organismo]

### CUS6\_Gestionar\_Contacto

<b>CUS6_Gestionar_Contacto</b>	
<b>Propósito:</b>	Permitir que el usuario persona pueda gestionar sus contactos.
<b>Actores:</b>	<b>Persona</b> (inicia)
<b>Resumen:</b>	El caso de uso se inicia cuando un usuario persona desea crear un contacto, eliminarlo o modificarlo. El caso de uso finaliza una vez que se realicen una de las acciones mencionadas.
<b>Precondición:</b>	El usuario debe conocer los datos de los que desea adicionar a su carpeta.
<b>Poscondición:</b>	El usuario realiza las gestiones del contacto.
<b>Referencias:</b>	<b>RF13, RF14, RF15</b>
<b>Prioridad:</b>	Crítico

Tabla 2.9 Descripción textual CUS6\_Gestionar\_Contacto

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

### CU7\_Buscar\_Lista\_Usuario

<b>CUS7_Buscar_Lista_Usuario</b>	
<b>Propósito:</b>	Permitir que el usuario pueda buscar sus listas de usuarios.
<b>Actores:</b>	<b>Persona</b> (inicia)
<b>Resumen:</b>	El caso de uso se inicia cuando un usuario persona desea buscar las listas de usuarios que tiene creadas para modificarlas o eliminarlas.
<b>Precondición:</b>	El usuario debe acceder a la mensajería síncrona o asíncrona.
<b>Poscondición:</b>	El usuario encuentra la lista deseada.
<b>Referencias:</b>	<b>RF13</b>
<b>Prioridad:</b>	Secundario

Tabla 2.10 Descripción textual *CUS7\_Buscar\_Lista\_Usuario*

### CUS8\_Buscar\_Contacto

<b>CUS8_Buscar_Contacto</b>	
<b>Propósito:</b>	Permitir que el usuario pueda buscar sus contactos.
<b>Actores:</b>	<b>Persona</b> (inicia)
<b>Resumen:</b>	El caso de uso se inicia cuando un usuario persona desea buscar los contactos que tiene creados para modificarlas o eliminarlas.
<b>Precondición:</b>	El usuario debe acceder a la mensajería asíncrona.
<b>Poscondición:</b>	El usuario encuentra el contacto deseado.
<b>Referencias:</b>	<b>RF17</b>
<b>Prioridad:</b>	Secundario

Tabla 2.11 Descripción textual *CUS8\_Buscar\_Contacto*

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

### CUS9\_ *Buscar\_ Usuario*

<b>CUS9_ <i>Buscar_ Usuario</i></b>	
<b>Propósito:</b>	Permitir que se pueda buscar los usuarios cuando se necesite.
<b>Actores:</b>	<b>Persona</b> (inicia)
<b>Resumen:</b>	El caso de uso se inicia cuando un usuario persona desea enviar un mensaje a uno o más usuarios del sistema.
<b>Precondición:</b>	El usuario debe acceder a la mensajería síncrona o asíncrona.
<b>Poscondición:</b>	Se encuentra el usuario deseado.
<b>Referencias:</b>	<b>RF18</b>
<b>Prioridad:</b>	Secundario

Tabla 2.12 Descripción textual *CUS9\_ Buscar\_ Usuario*

### CUS10\_ *Buscar\_ Organismo*

<b>CUS10_ <i>Buscar_ Organismo</i></b>	
<b>Propósito:</b>	Permitir que se pueda buscar los organismos cuando se necesite.
<b>Actores:</b>	<b>Persona</b> (inicia)
<b>Resumen:</b>	El caso de uso se inicia cuando un usuario persona desea agregar un grupo al sistema.
<b>Precondición:</b>	El usuario debe acceder a la mensajería síncrona o asíncrona.
<b>Poscondición:</b>	Se encuentra el usuario deseado.
<b>Referencias:</b>	<b>RF18</b>
<b>Prioridad:</b>	Secundario

Tabla 2.13 Descripción textual *CUS9\_ Buscar\_ Organismo*

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

---

### CUS11\_Gestionar\_Sala\_Reunión

<b>CUS10_Gestionar_Sala_Reunión</b>	
<b>Propósito:</b>	Permitir que la persona gestione las salas de reunión que le pertenezca.
<b>Actores:</b>	<b>Persona</b> (inicia).
<b>Resumen:</b>	El caso de uso se inicia cuando un usuario persona desea crear una sala de reunión para conversar en tiempo real con otros usuarios de un tema específico. El caso de uso finaliza cuando el usuario crea satisfactoriamente una sala.
<b>Precondición:</b>	El usuario persona debe acceder a la mensajería síncrona.
<b>Poscondición:</b>	El usuario persona crea la sala de reunión.
<b>Referencias:</b>	<b>RF20, RF21</b>
<b>Prioridad:</b>	Secundario

Tabla 2.14 Descripción textual *CUS10\_Gestionar\_Sala\_Reunión*

En este capítulo se abordó a las características del Componente de Notificación y Mensajería detallando algunos conceptos que son de gran utilidad para su comprensión. Se describen los requerimientos funcionales que debe cumplir; los no funcionales que el sistema deberá tener. Además se describen los casos de uso del sistema y se realiza su correspondiente diagrama para entender mejor la interacción de los actores con dicho componente

## CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

En este capítulo se traducen los requerimientos a una especificación que describe cómo implementar el sistema. El análisis consiste en obtener una visión del sistema, de modo que sólo se interesa por los requerimientos funcionales. El diseño es un refinamiento del análisis que tiene en cuenta los requerimientos no funcionales, o sea cómo cumple el sistema sus objetivos. El diseño debe ser suficiente para que el sistema pueda ser implementado sin ambigüedades.

### 3.1 Modelo de Análisis

En el Modelo de Análisis se estructuran los requerimientos de manera que se facilite su comprensión, preparación, modificación y en general su mantenimiento. En este se refinan y estructuran los requerimientos obtenidos con anterioridad, profundizando el equipo del proyecto en el dominio de la aplicación lo que les permitirá una mayor comprensión del problema para modelar la solución.

#### 3.1.1 Diagrama de Clases del Análisis

En este epígrafe se definen las clases del análisis:

- **Clase Entidad:** Modelan información que posee larga vida y que es a menudo persistente.
- **Clase Controladora:** Coordinan la realización de uno o unos pocos casos de uso coordinando las actividades de los objetos que implementan la funcionalidad del caso de uso.
- **Clase Interfaz:** Modelan la interacción entre el sistema y sus actores.

A continuación se muestran los diagramas de clases del análisis del Componente de Notificación y Mensajería:

## 3.1.1.1 Diagrama de clases del análisis *Autenticar*

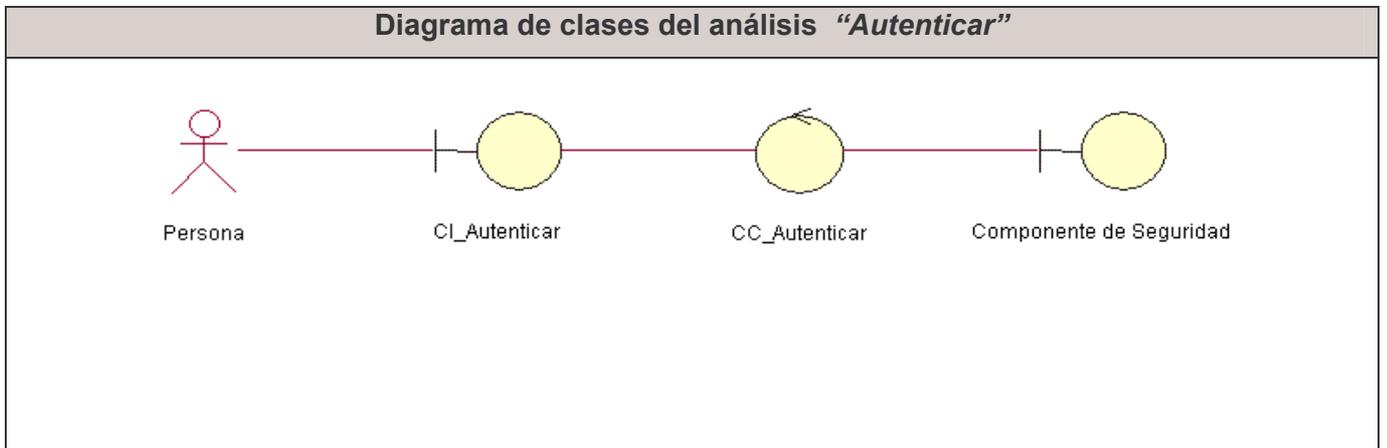


Fig 3.1 DCA\_Autenticar

## 3.1.1.2 Diagrama de clases del análisis *Gestionar\_Mensaje\_Síncrono*

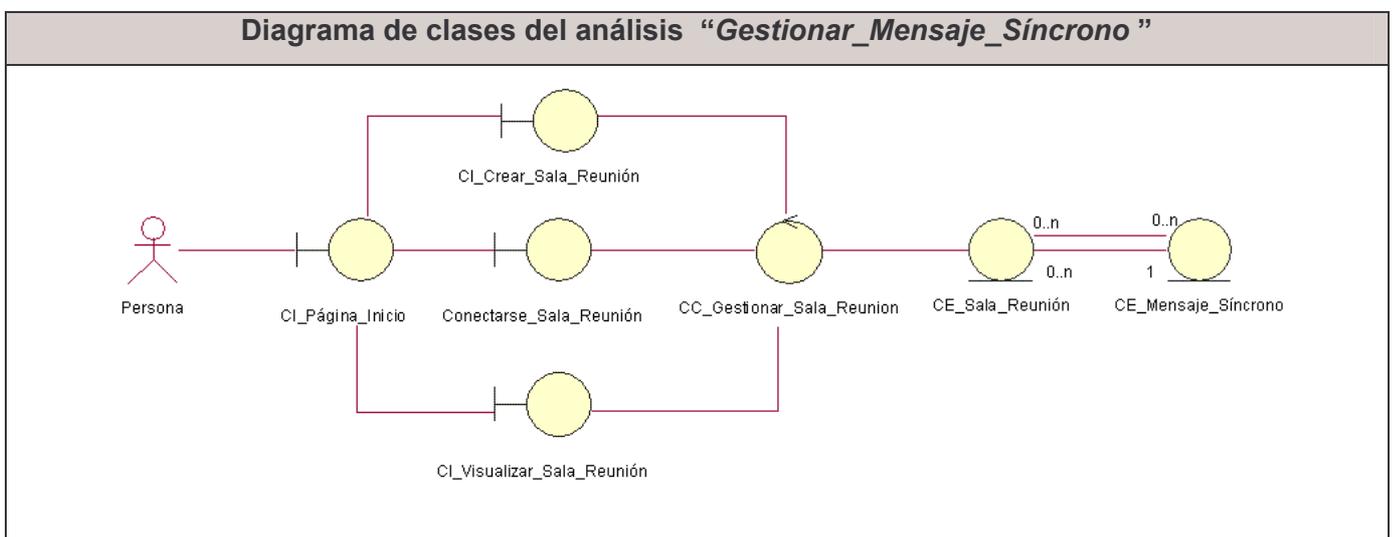


Fig 3.2 DCA\_Gestionar\_Mensaje\_Síncrono

# CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

## 3.1.1.3 Diagrama de clases del análisis *Gestionar\_Mensaje\_Asyncrono*

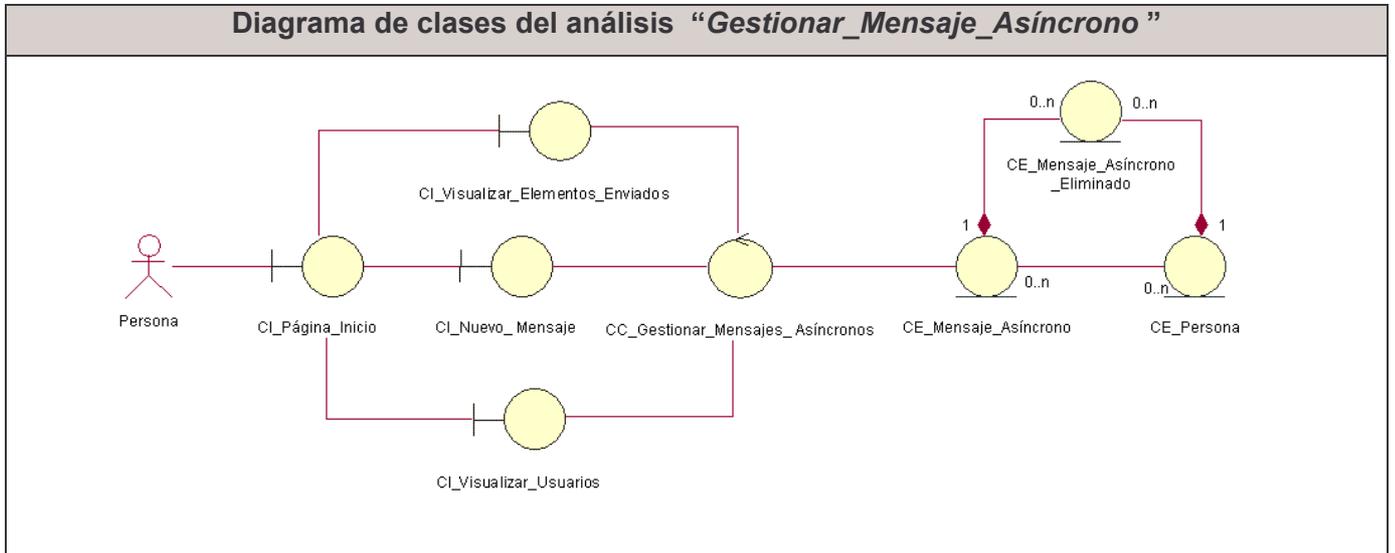


Fig 3.3 DCA\_Gestionar\_Mensaje\_Asyncrono

## 3.1.1.4 Diagrama de clases del análisis *Gestionar\_Grupo\_Usuario*

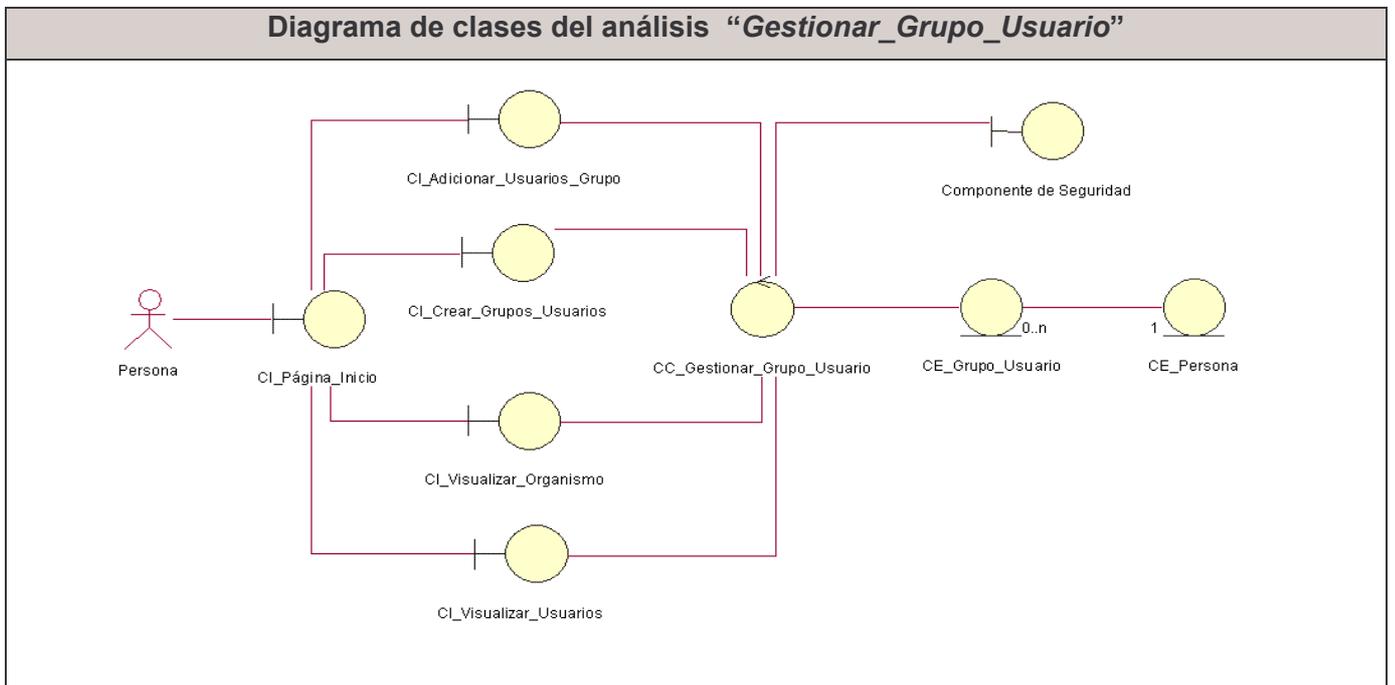


Fig 3.4 DCA\_Gestionar\_Grupo\_Usuario

# CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

## 3.1.1.5 Diagrama de clases del análisis *Gestionar\_Contacto*

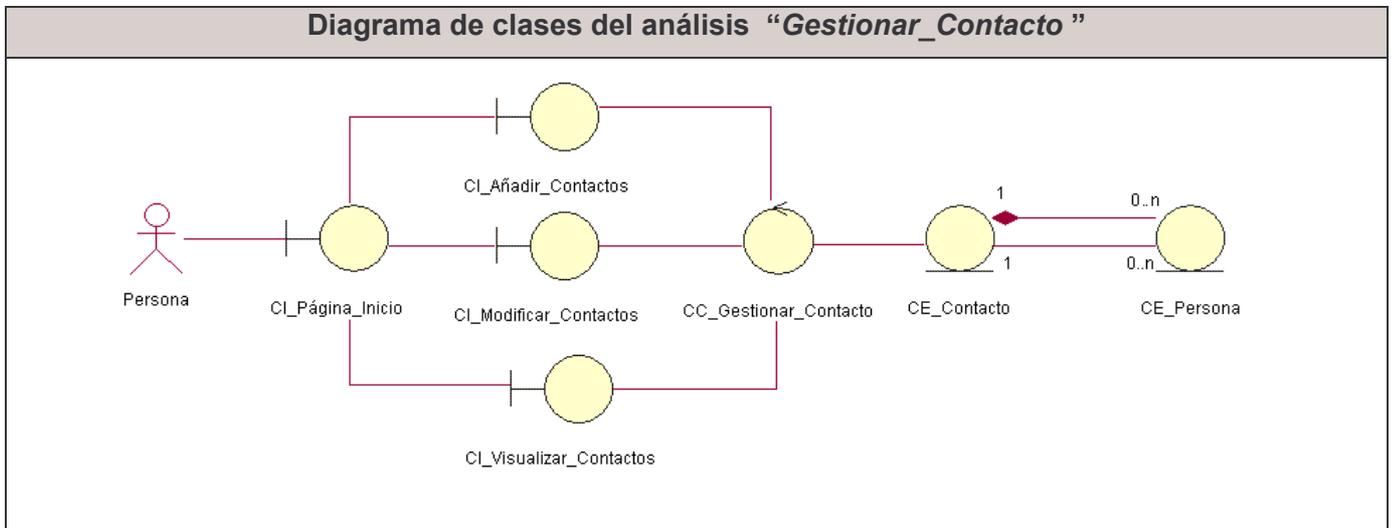


Fig 3.5 DCA\_Gestionar\_Contacto

## 3.1.1.6 Diagrama de clases del análisis *Gestionar\_Lista\_Usuario*

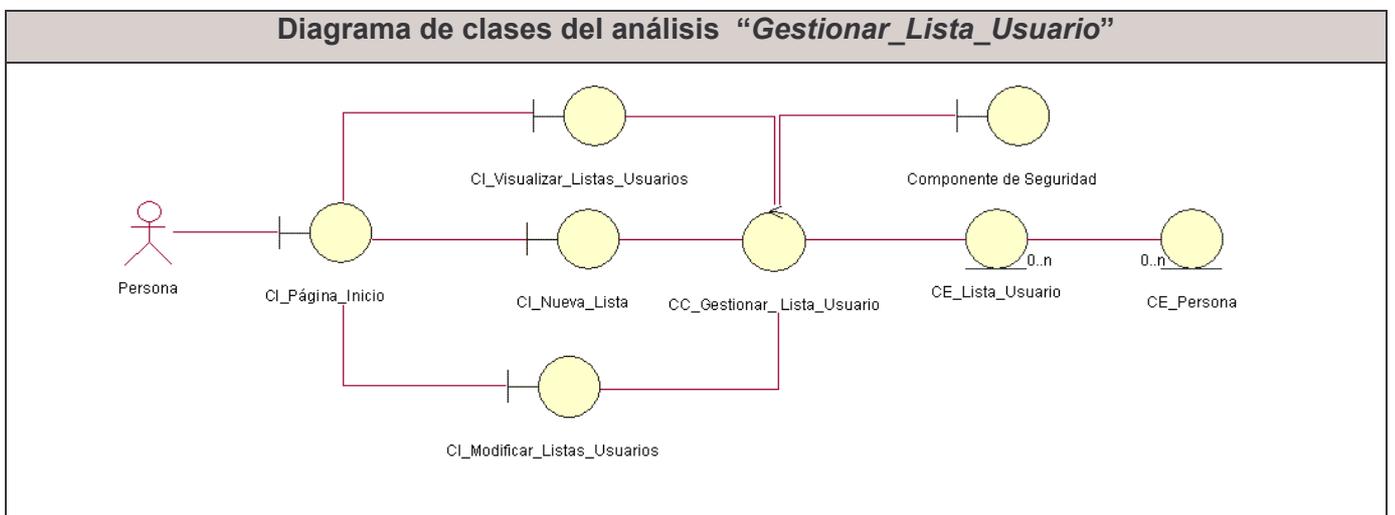


Fig 3.6 DCA\_Gestionar\_Lista\_Usuario

## 3.1.1.7 Diagrama de clases del análisis *Gestionar\_Sala\_Reunión*

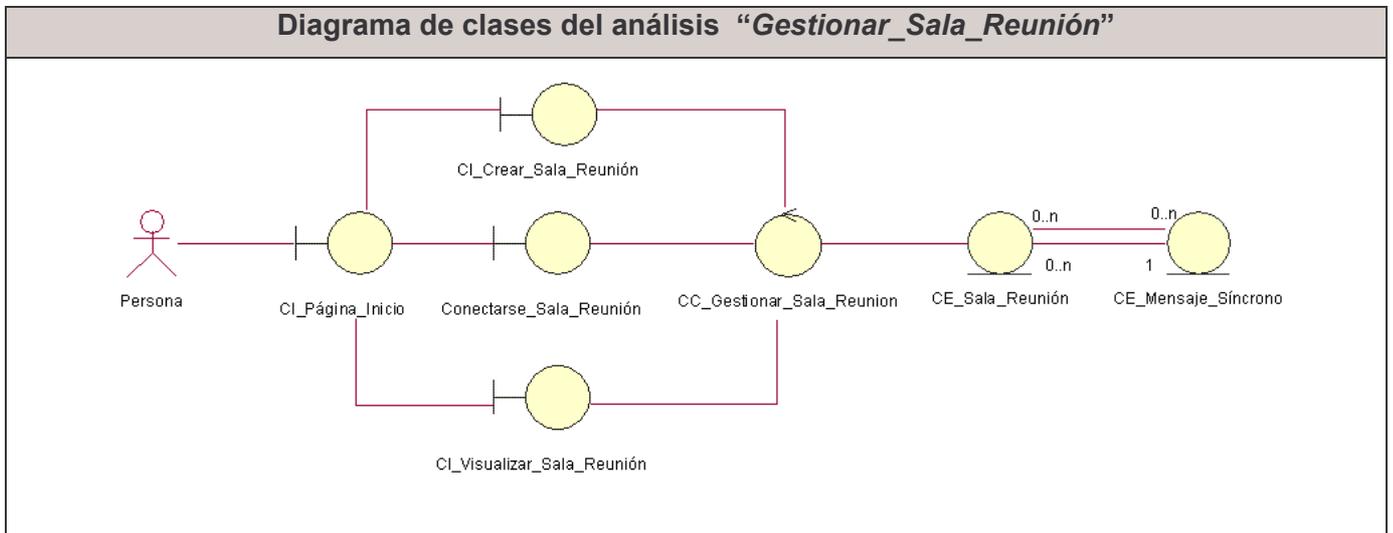


Fig 3.7 DCA\_Gestionar\_Sala\_Reunión

## 3.2 Modelo de Diseño

El diseño es el centro de atención al final de la fase de elaboración y el comienzo de las iteraciones de construcción. El Modelo de Diseño está muy cercano al de implementación, lo que es natural para guardar y mantener el modelo de diseño a través del ciclo de vida completo del software. En el diseño se modela el sistema y determina su forma para que soporte todos los requerimientos, incluyendo los no funcionales. Una entrada esencial en el diseño es el resultado del análisis, o sea el Modelo de Análisis, que proporciona una comprensión detallada de los requerimientos. Los propósitos del diseño se definen en transformar los requerimientos en un diseño de como el sistema debe ser, desarrollar una robusta arquitectura del sistema y adaptar el diseño para que se corresponda con el entorno de implementación, diseñando sus funcionalidades.

## 3.2.1 Estructuración del Modelo de Diseño

La descomposición en subsistemas es importante para la arquitectura, debido a que estos constituyen la estructura fundamental del producto del software. Esta descomposición se realiza antes de empezar el Modelo de Diseño definiendo sus interfaces y dependencias.

Un Diagrama de Subsistemas de Servicios es un conjunto de clases agrupadas por compartir funcionalidades similares y las relaciones que se establecen entre sí. Una de las características primordiales de estos subsistemas es su capacidad de reutilización. A continuación se muestran estos subsistemas:

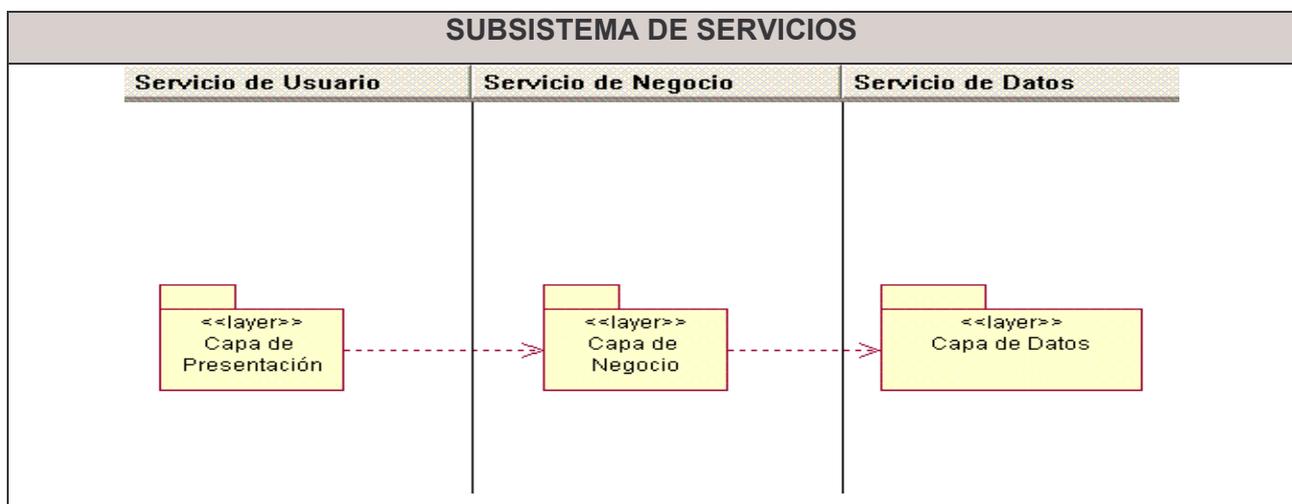


Fig 3.8 Subsistemas de Servicios

Para una mayor comprensión de los Diagramas de Clases del Diseño se realiza la descripción de sus capas:

**Capa de Presentación:** Esta capa intercambia información con los actores, siendo esta la única capa con la que interactúa el mismo. Recibe las peticiones de los usuarios la cual es trasladada a la Capa de Negocio. Contiene los ficheros mediante los cuales son invocados los métodos de la Capa de Negocio, entre ellos pueden ser mencionados los documentos JavaScript que se emplean en las validaciones y tratamiento de errores o excepciones.

**Capa de Negocio:** Esta capa establece la comunicación entre la capa de presentación y la capa de datos, encargada de recibir y responder cada petición de los usuarios. La misma recibe las peticiones

## CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

de los usuarios realizadas a través de la Capa de Presentación. Además se establecen las reglas del negocio definidas y se ejecutan los programas.

**Capa de Datos:** Esta capa recibe las solicitudes de almacenamiento o recuperación de información realizada a través de la capa de negocio. Está formada por el gestor de base de datos PostgreSQL.

### 3.2.2 Definición de los elementos del diseño

Las extensiones Web son muy utilizadas para realizar el diseño en UML de las aplicaciones web. Para esto se utilizan diferentes estereotipos para el modelado. Se tienen las clases “Client Page”, “HTML Form” y “Server Page” empleadas para el código cliente, formularios y código servidor respectivamente. Además se representan ficheros contenedores de sentencia script y clases que representan la lógica de negocio y la capa de acceso a datos del Componente de Notificación y Mensajería.

A continuación se representan las relaciones de los estereotipos Web: [16]

Hasta	Client Page	Form	Server Page
Desde			
Client Page	<<link>>, <<redirect>>	aggregation	<<link>>, <<redirect>>
Form	aggregate by		<<submit>>
Server Page	<<build>>, <<redirect>>		<<include>>, <<redirect>>

Se realizó una nomenclatura para cada clase del diseño la cual se describe a continuación:

CP\_<NombreClaseCliente> para las clases clientes, FORM\_<NombreClaseFormulario> para las clases formularios, SP\_<NombreClaseServidora> para las clases servidoras, <NombreClaseControladora.php> para las clases controladoras y CAD\_<NombreClaseAccesoDatos.php> para la clase de acceso a datos.

## 3.2.3 Diagrama de Clases del Diseño

### 3.2.3.1 Diagrama de clases del diseño *Autenticar*

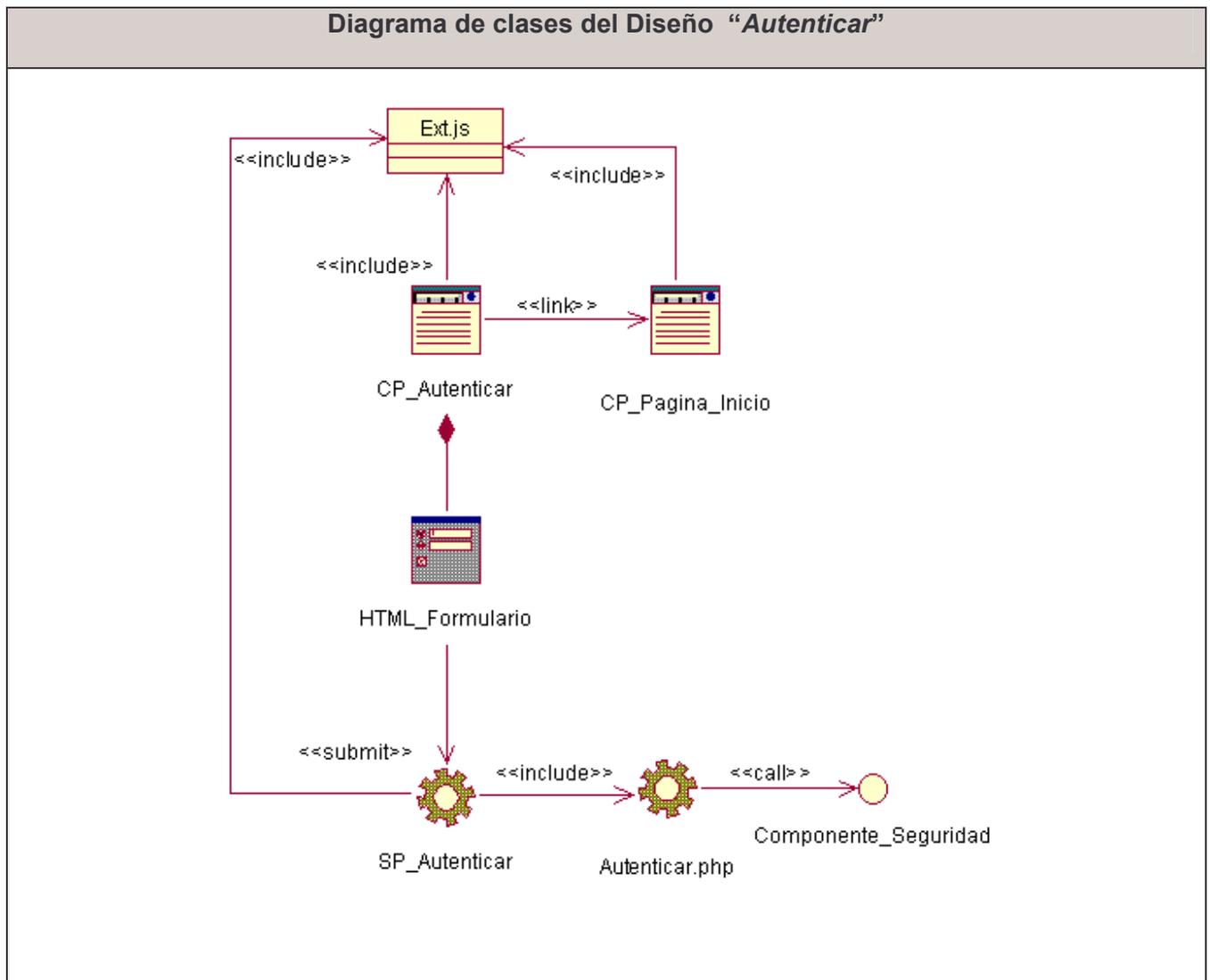


Fig 3.9 DCD\_Autenticar

## 3.2.3.2 Diagrama de clases del diseño *Gestionar\_Mensaje\_Sincrono*

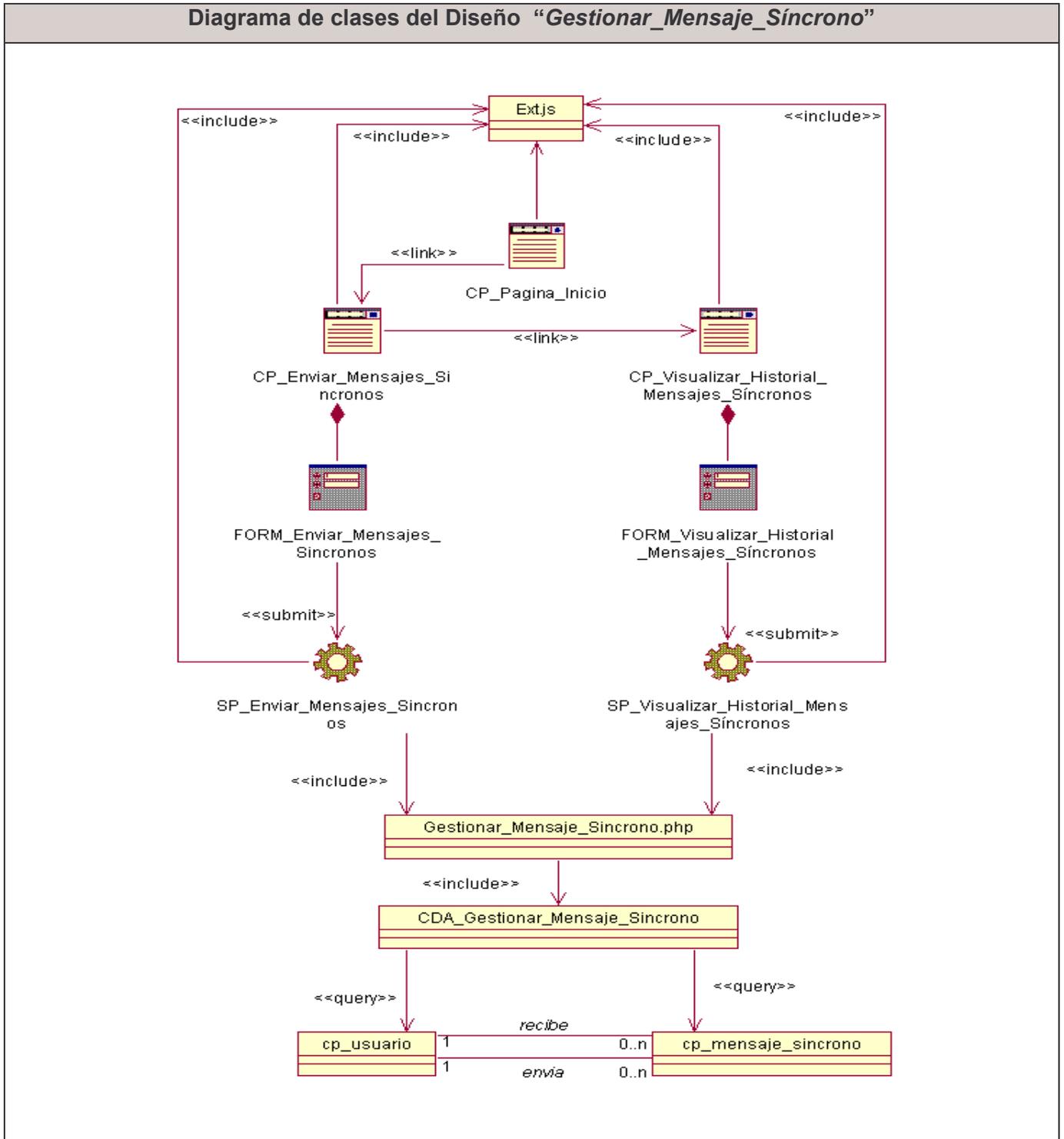


Fig 3.10 DCD\_Gestionar\_Mensaje\_Sincrono

## 3.2.3.3 Diagrama de clases del diseño *Gestionar\_Mensaje\_Asyncrono*

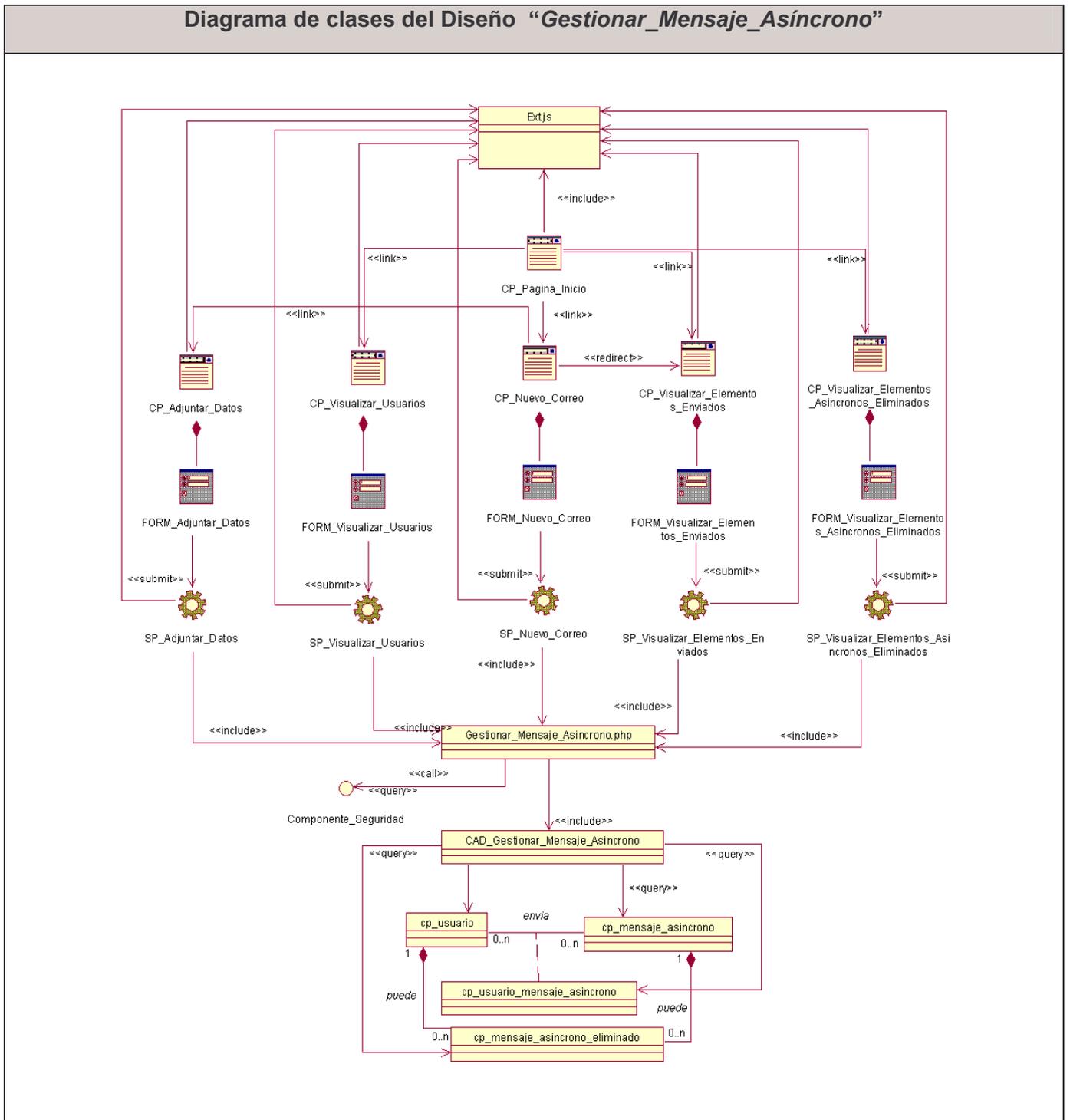


Fig 3.11 DCD\_Gestionar\_Mensaje\_Asyncrono

## 3.2.3.4 Diagrama de clases del diseño *Gestionar\_Grupo\_Usuario*

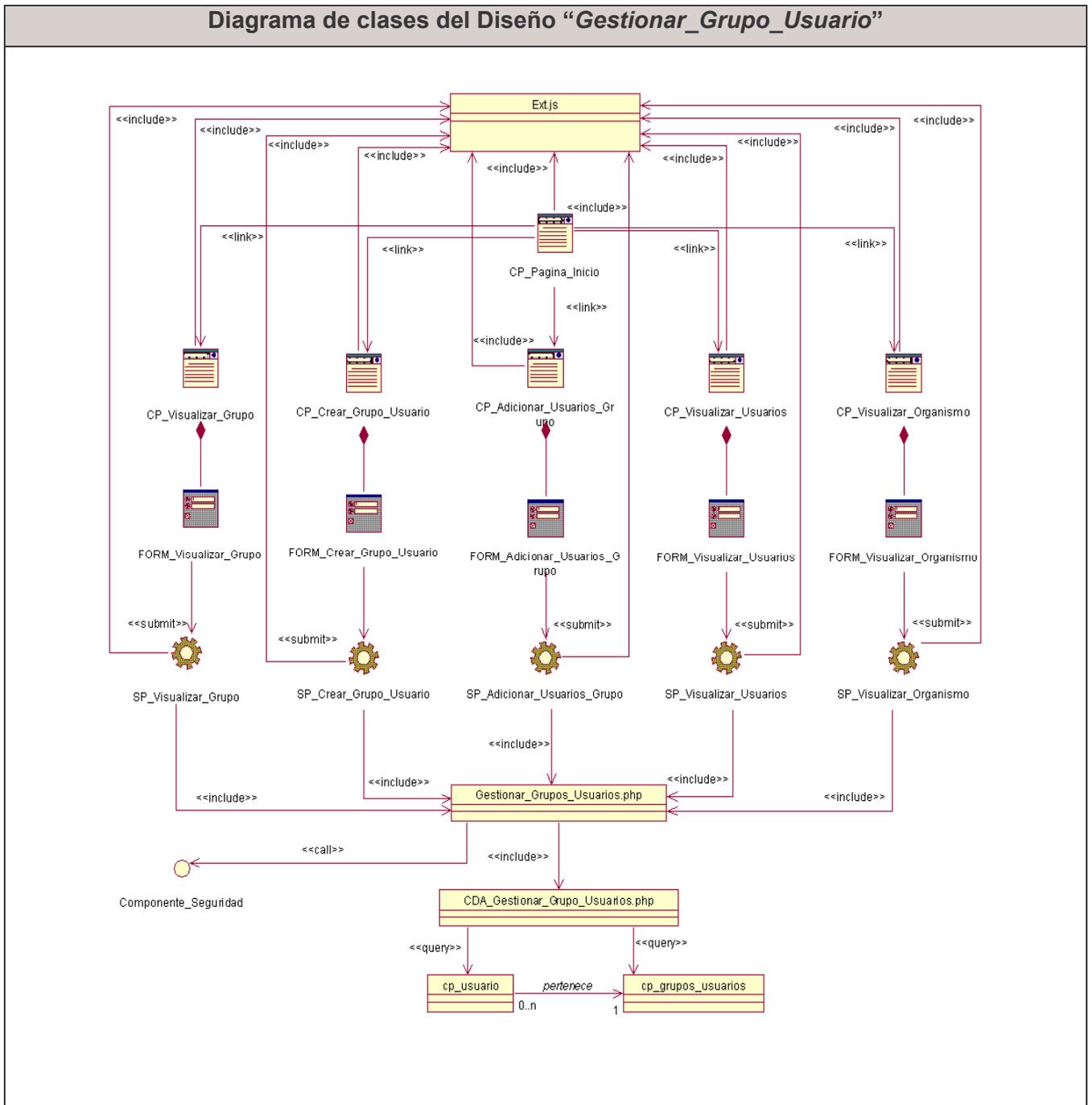


Fig 3.12 DCD\_Gestionar\_Grupo\_Usuario

## 3.2.3.5 Diagrama de clases del diseño *Gestionar\_Contacto*

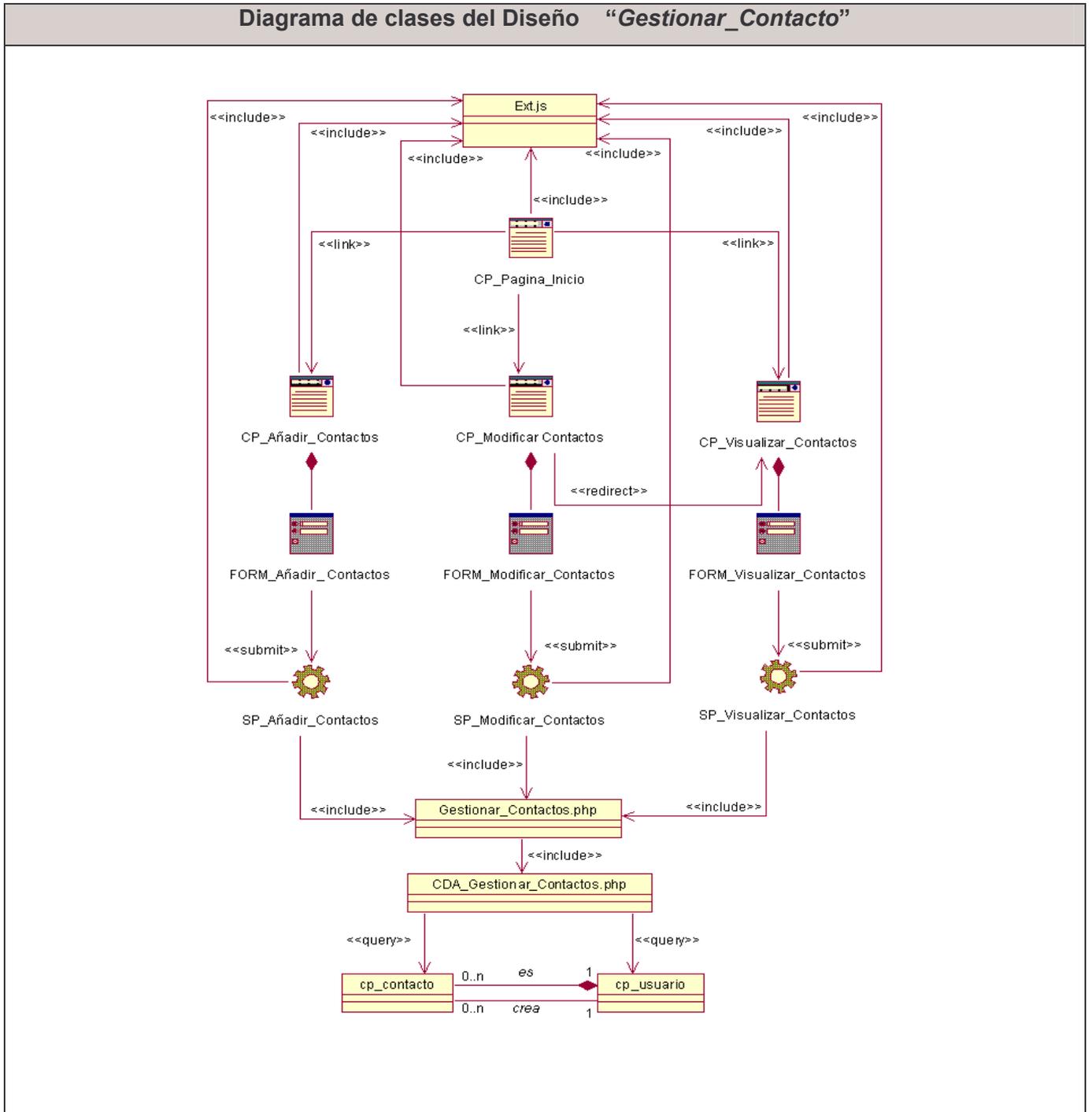


Fig 3.13 DCD\_Gestionar\_Contacto

## 3.2.3.6 Diagrama de clases del diseño *Gestionar\_Lista\_Usuario*

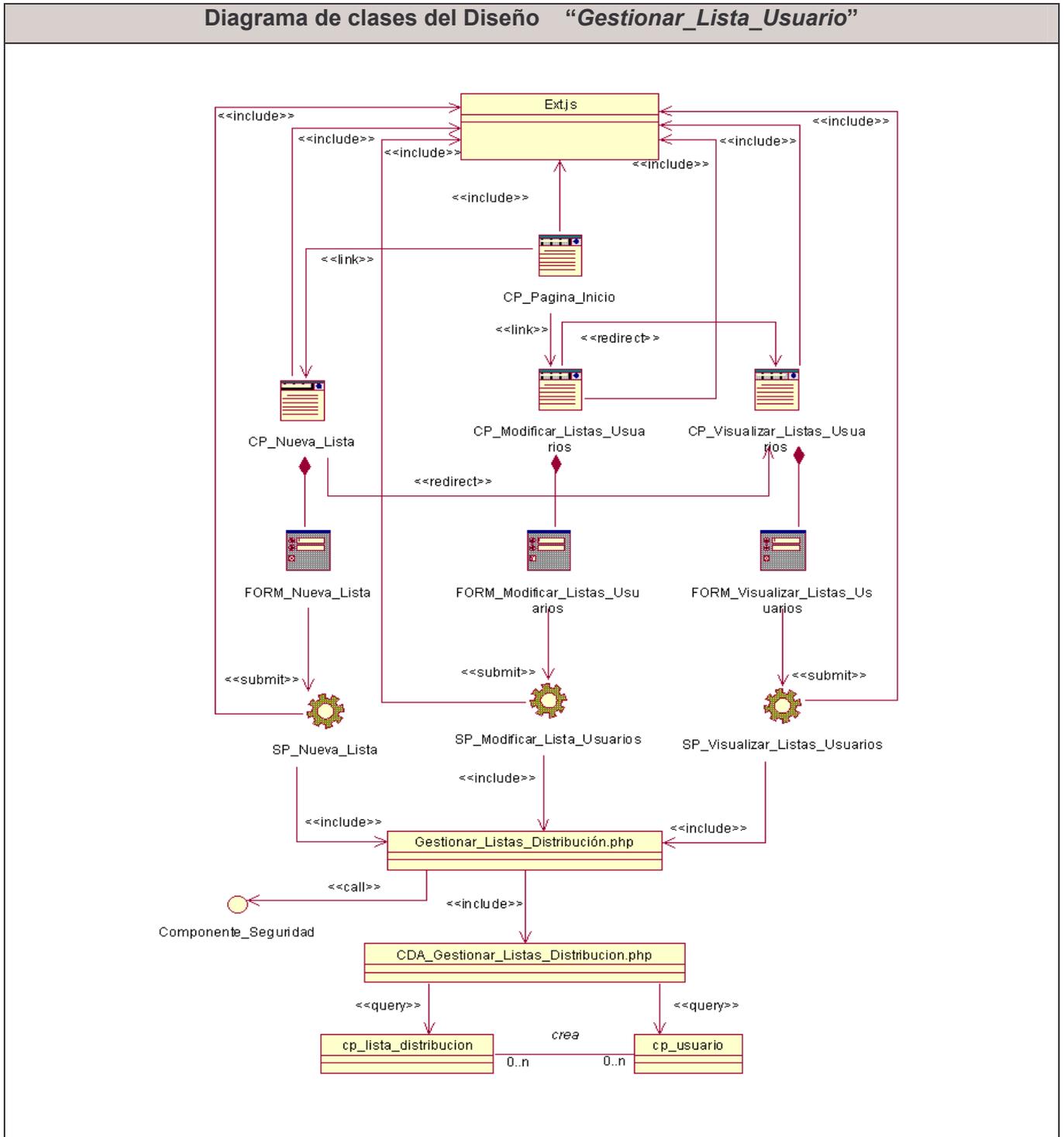


Fig 3.14 DCD\_Gestionar\_Lista\_Usuario

## 3.2.3.7 Diagrama de clases del diseño *Gestionar\_Sala\_Reunión*

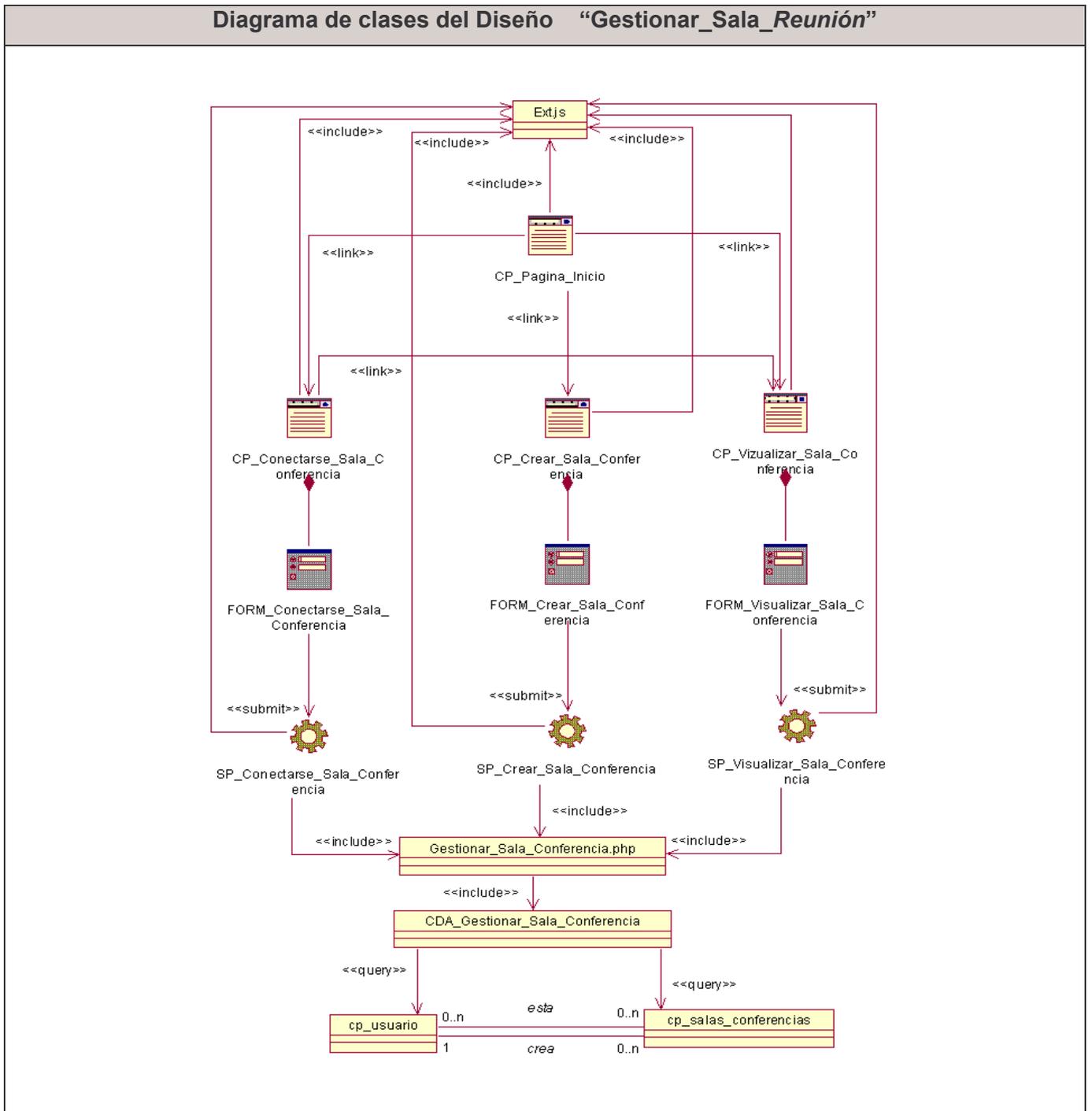


Fig 3.15 DCD\_Gestionar\_Sala\_Reunión

### 3.3 Descripción de clases y atributos

En este epígrafe se explican las clases del diseño que han sido identificadas para una mejor comprensión del Componente de Notificación y Mensajería.

#### 3.3.1 Descripción de páginas clientes

<b>Nombre:</b>	CP_Autenticar
<b>Tipo de clase:</b>	Página Cliente
<b>Descripción:</b>	Constituye la primera página Web con la que interactúa el usuario registrado en el Componente de Seguridad. Constituye la página de bienvenida y tiene como objetivo capturar el nombre de usuario y contraseña para poder acceder al componente de Notificación y Mensajería. Es utilizada en el caso de uso Autenticar.

Tabla 3.1 Descripción página cliente *CP\_Autenticar*

<b>Nombre:</b>	CP_Página_Inicio
<b>Tipo de clase:</b>	Página Cliente
<b>Descripción:</b>	Constituye la página de presentación del Componente de Notificación y Mensajería. Tiene como objetivo mostrarle al cliente el sistema con el cual va a interactuar. Es utilizada por todos los casos de uso excepto Autenticar.

Tabla 3.2 Descripción página cliente *CP\_Página\_Inicio*

<b>Nombre:</b>	CP_Añadir_Contactos
<b>Tipo de clase:</b>	Página Cliente
<b>Descripción:</b>	Página que brinda la posibilidad al usuario de crear los contactos que este desee, que no son más que datos adicionales de un usuario. Es utilizado por el caso de uso Gestionar_Contacto.

Tabla 3.3 Descripción página cliente *CP\_Añadir\_Contactos*

## CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

---

<b>Nombre:</b>	CP_Modificar_Contactos
<b>Tipo de clase:</b>	Página Cliente
<b>Descripción:</b>	Página que brinda la posibilidad al usuario de realizar cambios a los contactos creados por él. Es utilizado por el caso de uso Gestionar_Contacto.

Tabla 3.4 Descripción página cliente *CP\_Modificar\_Contactos*

<b>Nombre:</b>	CP_Visualizar_Contactos
<b>Tipo de clase:</b>	Página Cliente
<b>Descripción:</b>	Página que brinda la posibilidad al usuario de ver todos los contactos creados por él. Es utilizado por el caso de uso Gestiona_Contacto.

Tabla 3.5 Descripción página cliente *CP\_Visualizar\_Contactos*

<b>Nombre:</b>	CP_Visualizar_Elementos_Asíncronos_Eliminados
<b>Tipo de clase:</b>	Página Cliente
<b>Descripción:</b>	Página que brinda la posibilidad al usuario de ver las listas de distribución, contactos y mensajes asíncronos eliminados por el usuario. Es utilizado por el caso de uso Gestionar_Mensaje_Asíncrono.

Tabla 3.6 Descripción página cliente *CP\_Visualizar\_Elementos\_Asíncronos\_Eliminados*

<b>Nombre:</b>	CP_Nueva_Lista
<b>Tipo de clase:</b>	Página Cliente
<b>Descripción:</b>	Página que brinda la posibilidad al usuario de crear listas de usuarios, para mandar mensajes a usuarios que él tiene guardado en sus listas. Es utilizado por el caso de uso Gestionar_Lista_Usuario.

Tabla 3.7 Descripción página cliente *CP\_Nueva\_Lista*

## CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

---

<b>Nombre:</b>	CP_Modificar_Lista_Usuario
<b>Tipo de clase:</b>	Página Cliente
<b>Descripción:</b>	Página que brinda la posibilidad al usuario de realizar cambios en sus listas de distribución. Es utilizado por el caso de uso Gestionar_Lista_Usuario.

Tabla 3.8 Descripción página cliente *CP\_Modificar\_Lista\_Usuario*

<b>Nombre:</b>	CP_Visualizar_Lista_Usuario
<b>Tipo de clase:</b>	Página Cliente
<b>Descripción:</b>	Página que brinda la posibilidad al usuario de ver todas las listas de usuarios creadas por él. Es utilizado por el caso de uso Gestionar_Lista_Usuario.

Tabla 3.9 Descripción página cliente *CP\_Visualizar\_Listas\_Usuario*

<b>Nombre:</b>	CP_Adjunto
<b>Tipo de clase:</b>	Página Cliente
<b>Descripción:</b>	Página que brinda la posibilidad al usuario de ver en la pc del usuario el archivo que este desea adjuntar en un mensaje asíncrono. Es utilizado por el caso de uso Gestionar_Mensaje_Asíncrono.

Tabla 3.10 Descripción página cliente *CP\_Adjunto*

<b>Nombre:</b>	CP_Visualizar_Usuarios
<b>Tipo de clase:</b>	Página Cliente
<b>Descripción:</b>	Página que brinda la posibilidad al usuario de ver varios usuarios en caso de que este desee enviarle un mensaje asíncrono a otro usuario que no se sabe su dirección de correo electrónico y existan otros usuarios con el mismo nombre. Es utilizado por el caso de uso Gestionar_Mensaje_Asíncrono.

Tabla 3.11 Descripción página cliente *CP\_Visualizar\_Usuarios*

## CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

---

<b>Nombre:</b>	CP_Enviar_Nuevo_Mensaje
<b>Tipo de clase:</b>	Página Cliente
<b>Descripción:</b>	Página que brinda la posibilidad al usuario de enviar un mensaje asíncrono a otros usuarios. Es utilizado por el caso de uso Gestionar_Mensaje_Asíncrono.

Tabla 3.12 Descripción página cliente *CP\_Enviar\_Nuevo\_Mensaje*

<b>Nombre:</b>	CP_Enviar_Mensajes_Síncronos
<b>Tipo de clase:</b>	Página Cliente
<b>Descripción:</b>	Página que brinda la posibilidad al usuario de enviar mensajes síncronos a otros usuarios. Es utilizado por el caso de uso Gestionar_Mensaje_Síncrono.

Tabla 3.13 Descripción página cliente *CP\_Enviar\_Mensajes\_Síncronos*

<b>Nombre:</b>	CP_Visualizar_Historial_Mensaje_Síncronos
<b>Tipo de clase:</b>	Página Cliente
<b>Descripción:</b>	Página que brinda la posibilidad al usuario de ver los mensajes síncronos enviados a otros usuarios y viceversa. Es utilizado por el caso de uso Gestionar_Mensaje_Síncrono.

Tabla 3.14 Descripción página cliente *CP\_Visualizar\_Historial\_Mensaje\_Síncronos*

<b>Nombre:</b>	CP_Conectarse_Salas_Reuniones
<b>Tipo de clase:</b>	Página Cliente
<b>Descripción:</b>	Página que brinda la posibilidad al usuario de conectarse a una sala de conferencia elegida por él. Es utilizado por el caso de uso Gestionar_Sala_Reunión.

Tabla 3.15 Descripción página cliente *CP\_Conectarse\_Salas\_Reuniones*

## CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

---

<b>Nombre:</b>	CP_Crear_Salas_Reuniones
<b>Tipo de clase:</b>	Página Cliente
<b>Descripción:</b>	Página que brinda la posibilidad al usuario de crear salas de reuniones donde podrá conversar con otros usuarios en tiempo real de un tema de interés. Es utilizado por el caso de uso Gestionar_Sala_Reunión .

Tabla 3.16 Descripción página cliente *CP\_Crear\_Salas\_Reuniones*

<b>Nombre:</b>	CP_Visualizar_Salas_Reuniones
<b>Tipo de clase:</b>	Página Cliente
<b>Descripción:</b>	Página que brinda la posibilidad al usuario de ver todas la salas de reuniones almacenadas en el Componente de Notificación y Mensajería. Es utilizado por el caso de uso Gestionar_Sala_Reunión.

Tabla 3.17 Descripción página cliente *CP\_Visualizar\_Salas\_Reuniones*

<b>Nombre:</b>	CP_Crear_Grupos_Usuarios
<b>Tipo de clase:</b>	Página Cliente
<b>Descripción:</b>	Página que brinda la posibilidad al usuario de crear en la mensajería síncrona grupos donde se almacenaran usuarios al cual este desea de enviarles mensajes síncronos. Es utilizado por el caso de uso Gestionar_Grupo_Usuario.

Tabla 3.18 Descripción página cliente *CP\_Crear\_Grupo\_Usuario*

<b>Nombre:</b>	CP_Adicionar_Usuarios_Grupos
<b>Tipo de clase:</b>	Página Cliente
<b>Descripción:</b>	Página que brinda la posibilidad al usuario de agregar a otros usuarios a un grupo determinado. Es utilizado por el caso de uso Gestionar_Grupo_Usuario.

Tabla 3.19 Descripción página cliente *CP\_Adicionar\_Usuario\_Grupo*

### 3.3.2 Descripción de las clases del negocio

<b>Nombre:</b>	Gestionar_Contactos.php
<b>Tipo de clase:</b>	Clase del negocio
<b>Descripción:</b>	Clase controladora ejecutada en el lado del servidor que permite realizar todas las acciones de los contactos. Es donde se implementan los programas que serán ejecutados. Es utilizado por el caso de uso Gestionar_Contacto.

Tabla 3.20 Descripción clases del negocio *CN\_Gestionar\_Contactos.php*

<b>Nombre:</b>	Gestionar_Mensaje_Síncrono.php
<b>Tipo de clase:</b>	Clase del negocio
<b>Descripción:</b>	Clase controladora ejecutada en el lado del servidor que permite realizar todas las acciones de los mensajes síncronos, Es donde se implementan los programas que serán ejecutados. Es utilizado por el caso de uso Gestionar_Mensaje_Síncrono.

Tabla 3.21 Descripción clases del negocio *CN\_Gestionar\_Mensaje\_Síncrono.php*

<b>Nombre:</b>	Gestionar_Mensaje_Asíncrono.php
<b>Tipo de clase:</b>	Clase del negocio
<b>Descripción:</b>	Clase controladora ejecutada en el lado del servidor que permite realizar todas las acciones de los mensajes asíncronos, Es donde se implementan los programas que serán ejecutados. Es utilizado por el caso de uso Gestionar_Mensaje_Asíncrono.

Tabla 3.22 Descripción clases del negocio *CN\_Gestionar\_Mensaje\_Asíncrono.php*

<b>Nombre:</b>	Gestionar_Listas_Usuarios.php
<b>Tipo de clase:</b>	Clase del negocio
<b>Descripción:</b>	Clase controladora ejecutada en el lado del servidor que permite realizar todas las acciones de las listas de distribución. Es donde se implementan los programas que serán ejecutados. Es utilizado por el caso de uso Gestionar_Lista_Usuario.

Tabla 3.22 Descripción clases del negocio *CN\_Gestionar\_Listas\_Usuarios.php*

## CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

---

<b>Nombre:</b>	Gestionar_Grupos_Usuarios.php
<b>Tipo de clase:</b>	Clase del negocio
<b>Descripción:</b>	Clase controladora ejecutada en el lado del servidor que permite realizar todas las acciones de los grupos de usuarios, Es donde se implementan los programas que serán ejecutados. Es utilizado por el caso de uso Gestionar_Grupo_Usuario.

Tabla 3.23 Descripción clases del negocio *CN\_Gestionar\_Grupos\_Usuarios.php*

<b>Nombre:</b>	Gestionar_Salas_Reunión.php
<b>Tipo de clase:</b>	Clase del negocio
<b>Descripción:</b>	Clase controladora ejecutada en el lado del servidor que permite realizar todas las acciones de las salas de conferencias, Es donde se implementan los programas que serán ejecutados. Es utilizado por el caso de uso Gestionar_Sala_Reunión.

Tabla 3.24 Descripción clases del negocio *CN\_Gestionar\_Salas\_Reunión.php*

### 3.3.3 Descripción de las clases de acceso a datos

<b>Nombre:</b>	CAD_Gestionar_Contactos.php
<b>Tipo de clase:</b>	Clase de acceso a datos
<b>Descripción:</b>	Clase de acceso a datos ejecutada en el lado del servidor donde se realizan las consultas a las distintas clases involucradas. Es utilizado por el caso de uso Gestionar_Contacto.

Tabla 3.25 Descripción clase de acceso a datos *CAD\_Gestionar\_Contactos.php*

<b>Nombre:</b>	CAD_Gestionar_Mensaje_Síncrono.php
<b>Tipo de clase:</b>	Clase de acceso a datos
<b>Descripción:</b>	Clase de acceso a datos ejecutada en el lado del servidor donde se realizan las consultas a las distintas clases involucradas. Es utilizado por el caso de uso Gestionar_Mensaje_Síncrono.

Tabla 3.26 Descripción clase de acceso a datos *CAD\_Gestionar\_Mensaje\_Síncrono.php*

## CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

---

<b>Nombre:</b>	CAD_Gestionar_Mensaje_Asíncrono.php
<b>Tipo de clase:</b>	Clase de acceso a datos
<b>Descripción:</b>	Clase de acceso a datos ejecutada en el lado del servidor donde se realizan las consultas a las distintas clases involucradas. Es utilizado por el caso de uso Gestionar_Mensaje_Asíncrono.

Tabla 3.27 Descripción clase de acceso a datos *CAD\_Gestionar\_Mensaje\_Asíncrono.php*

<b>Nombre:</b>	CAD_Gestionar_Listas_Usuarios.php
<b>Tipo de clase:</b>	Clase de acceso a datos
<b>Descripción:</b>	Clase de acceso a datos ejecutada en el lado del servidor donde se realizan las consultas a las distintas clases involucradas. Es utilizado por el caso de uso Gestionar_Lista_Usuario.

Tabla 3.28 Descripción clase de acceso a datos *CAD\_Gestionar\_Listas\_Usuarios.php*

<b>Nombre:</b>	CAD_Gestionar_Grupos_Usuarios.php
<b>Tipo de clase:</b>	Clase de acceso a datos
<b>Descripción:</b>	Clase de acceso a datos ejecutada en el lado del servidor donde se realizan las consultas a las distintas clases involucradas. Es utilizado por el caso de uso Gestionar_Grupo_Usuario.

Tabla 3.29 Descripción clase de acceso a datos *CAD\_Gestionar\_Grupos\_Usuarios.php*

<b>Nombre:</b>	CAD_Gestionar_Sala_Reunión.php
<b>Tipo de clase:</b>	Clase de acceso a datos
<b>Descripción:</b>	Clase de acceso a datos ejecutada en el lado del servidor donde se realizan las consultas a las distintas clases involucradas. Es utilizado por el caso de uso Gestionar_Sala_Reunión.

Tabla 3.30 Descripción clase de acceso a datos *CAD\_Gestionar\_Sala\_Reunión.php*

### 3.4 Diseño de la Base de Datos

Para poder realizar un buen diseño de Base de Datos se tienen en cuenta una serie de pasos para que este resulte con la mayor calidad posible.

#### 3.4.1 Definición de Clases Persistentes

Todas las clases identificadas en el dominio del análisis no son persistentes. La persistencia es la capacidad de un objeto de mantener su valor en el espacio y en el tiempo. Es responsabilidad del diseñador definir qué clases son las que deben ser persistentes. En este proceso se recomienda aplicar algunas reglas, ellas son:

- Cuando una clase que está formada por otras clases es persistente, automáticamente las clases componentes también son persistentes. Lo contrario no se cumple necesariamente.
- Cuando una clase hija de una jerarquía es persistente, automáticamente son persistentes sus ancestros en el árbol de jerarquía. Lo contrario no se cumple necesariamente.
- Cuando se define como persistente a una clase que agrupa a objetos de un mismo tipo de clase base (se refiere a las clases listas, colecciones, registros), entonces automáticamente son persistentes todas las clases hijas a partir de la clase base, incluyendo a la clase base.
- Cuando hay herencia múltiple, esta debe ser resuelta antes si el medio de almacenamiento ha utilizar no soporta este concepto. La solución más factible es que la clase hija herede de la clase, de la que redefine sus métodos y añada un atributo pasivo del tipo de la otra clase de la que heredaba.

En fases anteriores dentro del ciclo de vida del desarrollo de un proyecto, puede que se hayan definido clases que coordinan el trabajo de varias clases, las llamadas clases controladoras. Se recomienda no definir las como persistentes porque conceptualmente no lo son.

### 3.4.1.1 Diagrama de Clases Persistentes

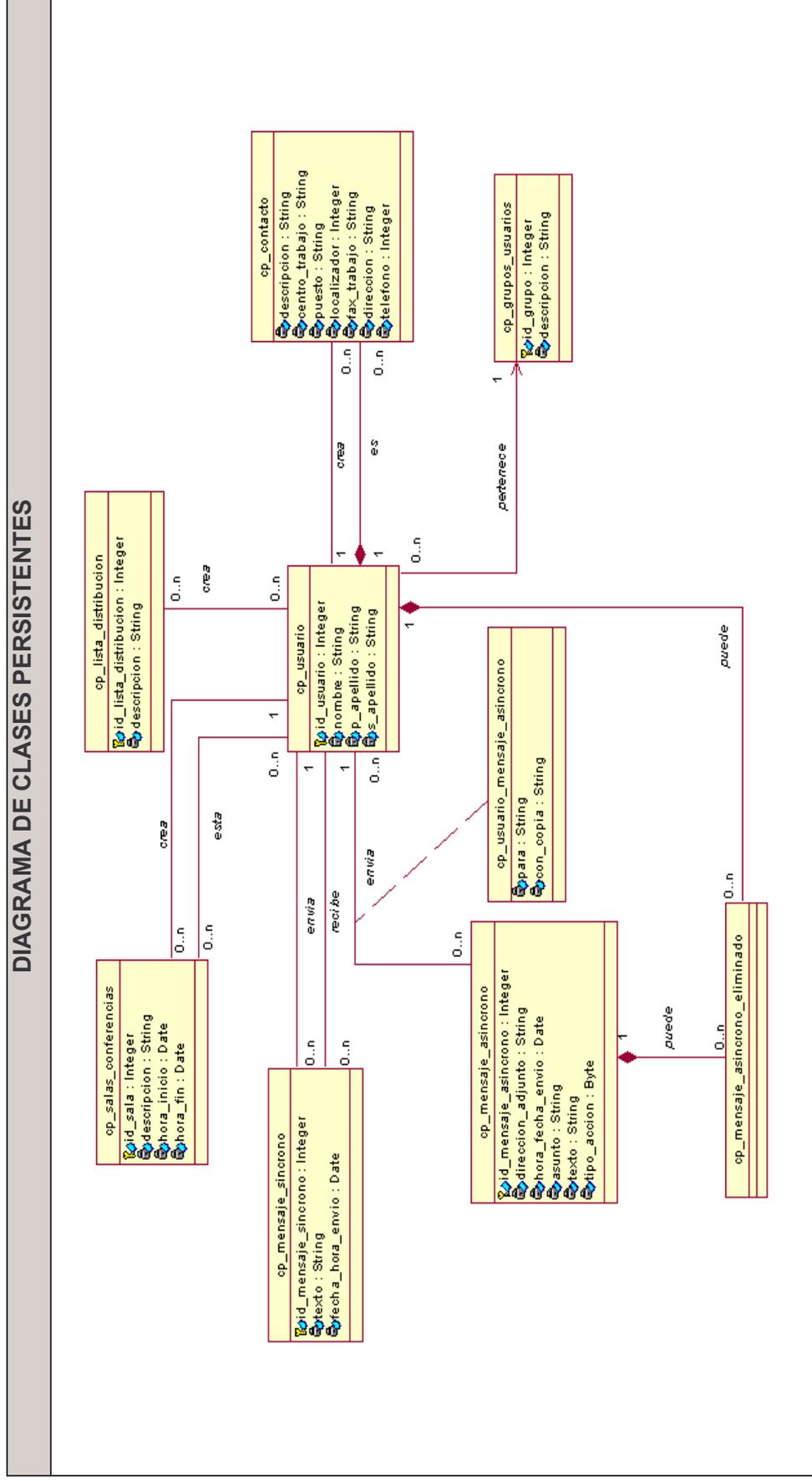


Figura 3.16 Diagrama de clases persistentes.

3.4.2 Modelo de Datos

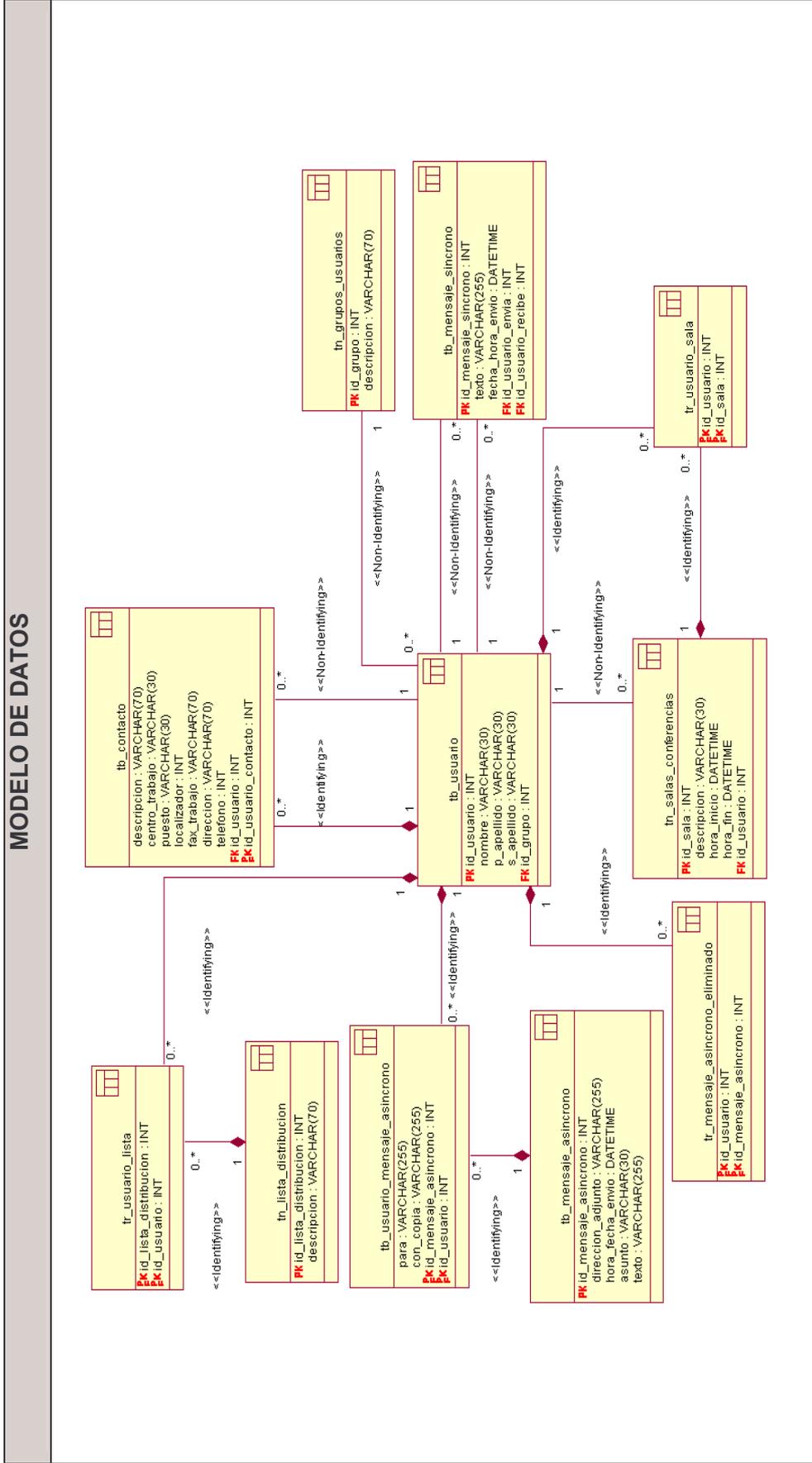


Figura 3.17 Modelo de Datos

### 3.4.2.1 Descripción de las tablas de la base de datos

En esta sección se describen las tablas de la Base de Datos a partir del modelo antes representado. Se les da la notación de `tb_<NombreTabla>` a las tablas de la base de datos, `tn_<NombreTabla>` a las tablas nomencladores y `tr_<NombreTabla>` a las nuevas tablas que surgen con motivo de una relación entre varias tablas.

<b>Nombre: tb_contacto</b>			
<b>Descripción:</b> Tabla que almacenará a los datos de los contactos que posee un usuario.			
<b>Atributo</b>	<b>Tipo</b>	<b>Nulo</b>	<b>Descripción</b>
descripcion	VARCHAR(70)	Si	Descripción del contacto o información que gestiona el mismo.
centro_trabajo	VARCHAR(30)	NO	Centro de trabajo al cual pertenecerá el contacto.
puesto	VARCHAR (30)	NO	Cargo que ocupa el contacto en su centro de trabajo.
localizador	INT	Si	Localizador o Bipeer que posee el contacto.
fax_trabajo	VARCHAR (70)	Si	Fax que posee el contacto en su centro de trabajo.
direccion	VARCHAR (70)	NO	Dirección particular o del trabajo del contacto
telefono	INT	NO	Teléfono particular o del trabajo del contacto
id_usuario	INT	NO	Identificador único para el usuario, de la tabla <b>tb_usuario (FK)</b> .
id_usuario_contacto	INT	NO	Identificador único de la tabla Contacto. <b>(PK/FK)</b> .

Tabla 3.31 Descripción de la Tabla `tb_contacto`.

## CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

<b>Nombre: tb_usuario_mensaje_asincrono</b>			
<b>Descripción:</b> Tabla que almacenará los remitente y los destinatarios a los cuales son enviados mensajes asíncronos.			
<b>Atributo</b>	<b>Tipo</b>	<b>Nulo</b>	<b>Descripción</b>
para	VARCHAR(255)	NO	.Campo donde se especifica los destinatarios del mensaje
con_copia	VARCHAR(255)	NO	Campo donde se especifica otros destinatarios del mensaje que el usuario desee.
id_mensaje_asincrono	INT	NO	Identificador único para el usuario, de la tabla <b>tb_mensaje_asincrono (PK/FK)</b> .
id_usuario	INT	NO	Identificador único para el usuario, de la tabla <b>tb_usuario (PK/FK)</b> .

Tabla 3.32 Descripción de la Tabla tb\_usuario\_mensaje\_asincrono

<b>Nombre: tb_mensaje_sincrono</b>			
<b>Descripción:</b> Tabla que almacenará los datos de los mensajes Síncronos.			
<b>Atributo</b>	<b>Tipo</b>	<b>Nulo</b>	<b>Descripción</b>
id_mensaje_sincrono	INT	NO	Identificador único de los mensajes Síncronos. <b>(PK)</b>
texto	VARCHAR(255)	NO	Texto que tendrán los mensajes Síncronos.
fecha_hora_envio	DATETIME	NO	Hora y fecha en la que el mensaje Síncrono haya sido enviado o recibido
id_usuario_envia	INT	NO	Identificador único para el usuario que envía mensajes Síncronos, de la tabla <b>tb_usuario. (FK)</b>
id_usuario_recibe	INT	NO	Identificador único para el usuario que recibe mensajes Síncronos, de la tabla <b>tb_usuario. (FK)</b>

Tabla 3.33 Descripción de la Tabla tb\_mensaje\_sincrono.

## CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

<b>Nombre: tb_usuarios</b>			
<b>Descripción:</b> Tabla que almacenará información de los usuarios del Componente de Notificación y Mensajería.			
<b>Atributo</b>	<b>Tipo</b>	<b>Nulo</b>	<b>Descripción</b>
id_usuario	INT	NO	Identificador único para el usuario <b>(PK)</b> .
nombre	VARCHAR(30)	NO	Nombre del usuario. Este atributo se obtiene del Componente de Seguridad.
p_apellido	VARCHAR(30)	NO	Primer apellido del usuario. Este atributo se obtiene del Componente de Seguridad.
s_apellido	VARCHAR(30)	NO	Segundo apellido del usuario. Este atributo se obtiene del Componente de Seguridad..
id_grupo	INT	NO	Identificador único de la tabla <b>tn_grupos_usuarios (FK)</b> .

Tabla 3.34 Descripción de la Tabla tb\_usuarios.

<b>Nombre: tb_mensaje_asincrono</b>			
<b>Descripción:</b> Tabla donde se almacenarán los datos de los mensajes Asíncronos.			
<b>Atributo</b>	<b>Tipo</b>	<b>Nulo</b>	<b>Descripción</b>
id_mensaje_asincrono	INT	NO	Identificador único para el mensaje Asíncrono. <b>(PK)</b>
direccion_adjunto	VARCHAR(255)	Si	Dirección del adjunto que tendrá el mensaje Asíncrono.
hora_fecha_envio	DATETIME	NO	Hora y Fecha en que el mensaje Asíncrono haya sido recibido enviado.
asunto	VARCHAR (30)	Si	Asunto que tendrá el mensaje Asíncrono.
texto	VARCHAR(255)	NO	Texto que tendrá el mensaje Asíncrono.

Tabla 3.35 Descripción de la Tabla tb\_mensaje\_asincrono.

## CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

<b>Nombre: tn_salas_conferencias</b>			
<b>Descripción:</b> Tabla que almacenará las salas de conferencias creadas por los usuarios.			
<b>Atributo</b>	<b>Tipo</b>	<b>Nulo</b>	<b>Descripción</b>
id_sala	INT	NO	Identificador único para la salas de conferencias <b>(PK)</b>
descripcion	VARCHAR(30)	NO	Descripción de la sala de conferencia.
hora_inicio	DATETIME	NO	Hora en que se inicia las conferencias en las salas.
hora_fin	DATETIME	NO	Hora en que se terminan las conferencias en las salas o se cierra una sala.
id_usuario	INT	NO	Identificador único para el usuario, de la tabla <b>tb_usuario (FK)</b> .

Tabla 3.36 Descripción de la Tabla tn\_sala\_reunión.

<b>Nombre: tn_grupos_usuarios</b>			
<b>Descripción:</b> Tabla que almacenará los datos de los usuarios que pertenecen a un grupo de usuarios.			
<b>Atributo</b>	<b>Tipo</b>	<b>Nulo</b>	<b>Descripción</b>
id_grupo	INT	NO	Identificador único de los grupos de usuarios. <b>(PK)</b> .
descripcion	VARCHAR(70)	NO	Descripción de los grupos de usuarios.

Tabla 3.37 Descripción de la Tabla tn\_grupos\_usuarios.

<b>Nombre: tn_lista_usuario</b>			
<b>Descripción:</b> Tabla que almacenará los datos de las listas de distribución que tendrán los usuarios.			
<b>Atributo</b>	<b>Tipo</b>	<b>Nulo</b>	<b>Descripción</b>
id_lista_distribucion	INT	NO	Identificador único para las listas de usuarios. <b>(PK)</b>
descripcion	VARCHAR 70)	NO	Descripción de la lista de usuario.

Tabla 3.38 Descripción de la Tabla tb\_lista\_usuario.

## CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

<b>Nombre: tr_usuario_lista</b>			
<b>Descripción:</b> Tabla que almacenara las listas de usuario que tiene un usuario.			
<b>Atributo</b>	<b>Tipo</b>	<b>Nulo</b>	<b>Descripción</b>
id_lista_usuario	INT	NO	Identificador único para el usuario en la tabla. <b>tr_usuario_lista (PK/FK)</b>
id_usuario	INT	NO	Identificador único para el usuario, de la tabla <b>tb_usuario (PK/FK)</b>

Tabla 3.39 Descripción de la Tabla tr\_usuario\_lista.

<b>Nombre: tr_mensaje_asincrono_eliminado</b>			
<b>Descripción:</b> Tabla que almacenará los datos de los mensajes eliminados.			
<b>Atributo</b>	<b>Tipo</b>	<b>Nulo</b>	<b>Descripción</b>
id_mensaje_asincrono	INT	NO	Identificador único para el mensaje Síncrono en la tabla <b>tb_mensaje_asincrono. (PK/FK).</b>
id_usuario	INT	NO	Identificador único para el usuario en la tabla <b>tb_usuario. (PK/FK).</b>

Tabla 3.40 Descripción de la Tabla tr\_mensaje\_asincrono\_eliminado.

<b>Nombre: tr_usuarios_sala</b>			
<b>Descripción:</b> Tabla que almacenará los datos de los usuarios que pertenecen a las salas de conferencias.			
<b>Atributo</b>	<b>Tipo</b>	<b>Nulo</b>	<b>Descripción</b>
id_sala_	INT	NO	Identificador único la sala de reunión de la tabla <b>tb_sala_reunión(PK/FK)</b>
id_usuario	INT	NO	Identificador único para el usuario en la tabla <b>tb_usuario. (PK/FK).</b>

Tabla 3.41 Descripción de la Tabla tr\_usuario\_sala

## 3.5 Modelo de Despliegue

El Modelo de Despliegue representa la estructura lógica del Componente de Notificación y Mensajería, es decir como va a estar distribuida su arquitectura. A continuación se representa como quedaría esta distribución.

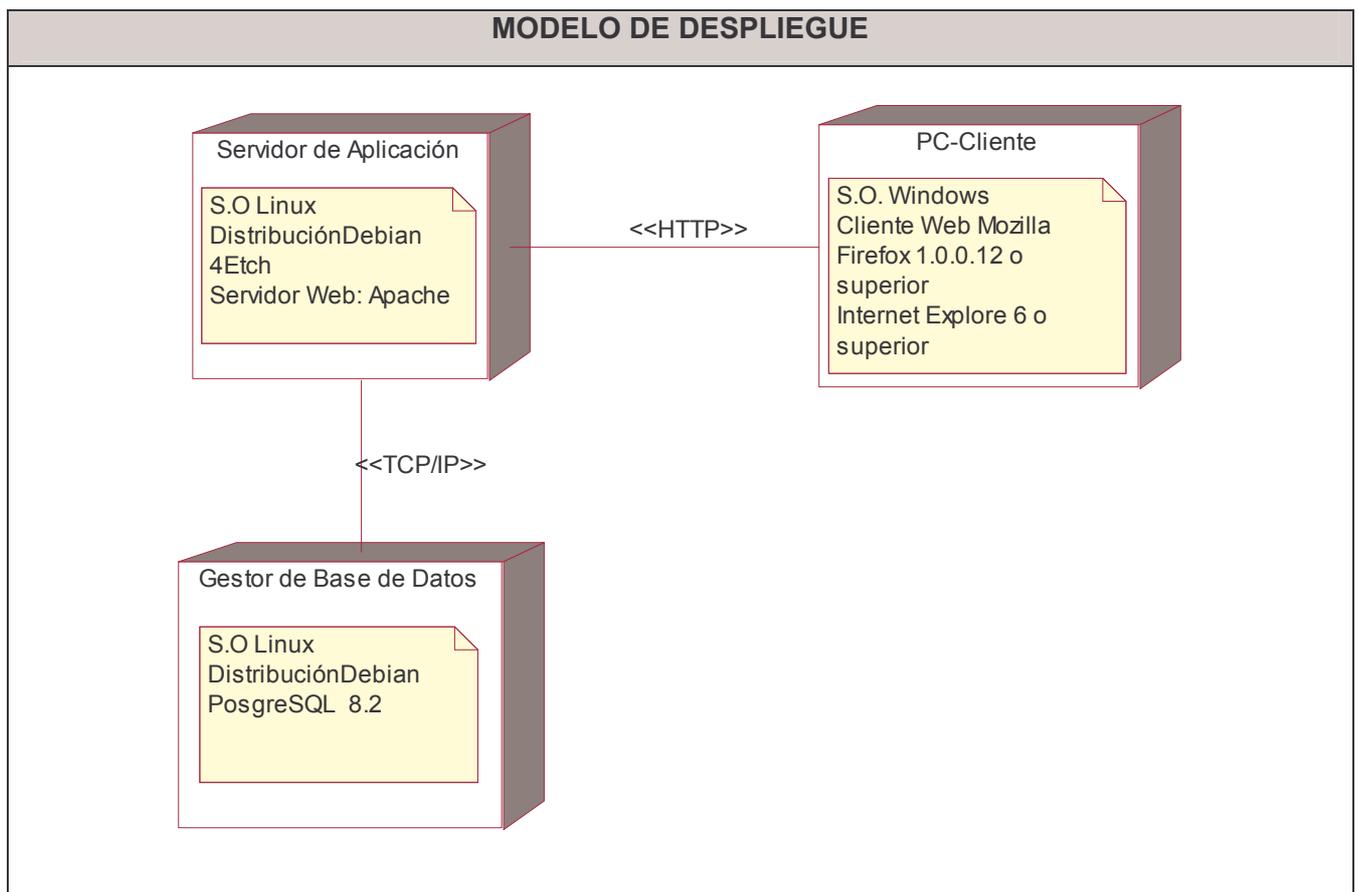


Figura 3.18 Modelo de Despliegue.

### 3.6 Prototipo no Funcional

Un prototipo no funcional es el diseño de una interfaz de usuario, el cual no tiene funcionalidades implementadas, pero sí las distintas páginas o componentes al cual tendrá acceso el usuario y las relaciones entre estas. A continuación se muestran las páginas principales del prototipo no funcional del Componente de Notificación y Mensajería.



Figura 3.19 Página de autenticación.

**A:** Región donde el usuario persona introduce su nombre de usuario y contraseña para acceder a la aplicación.

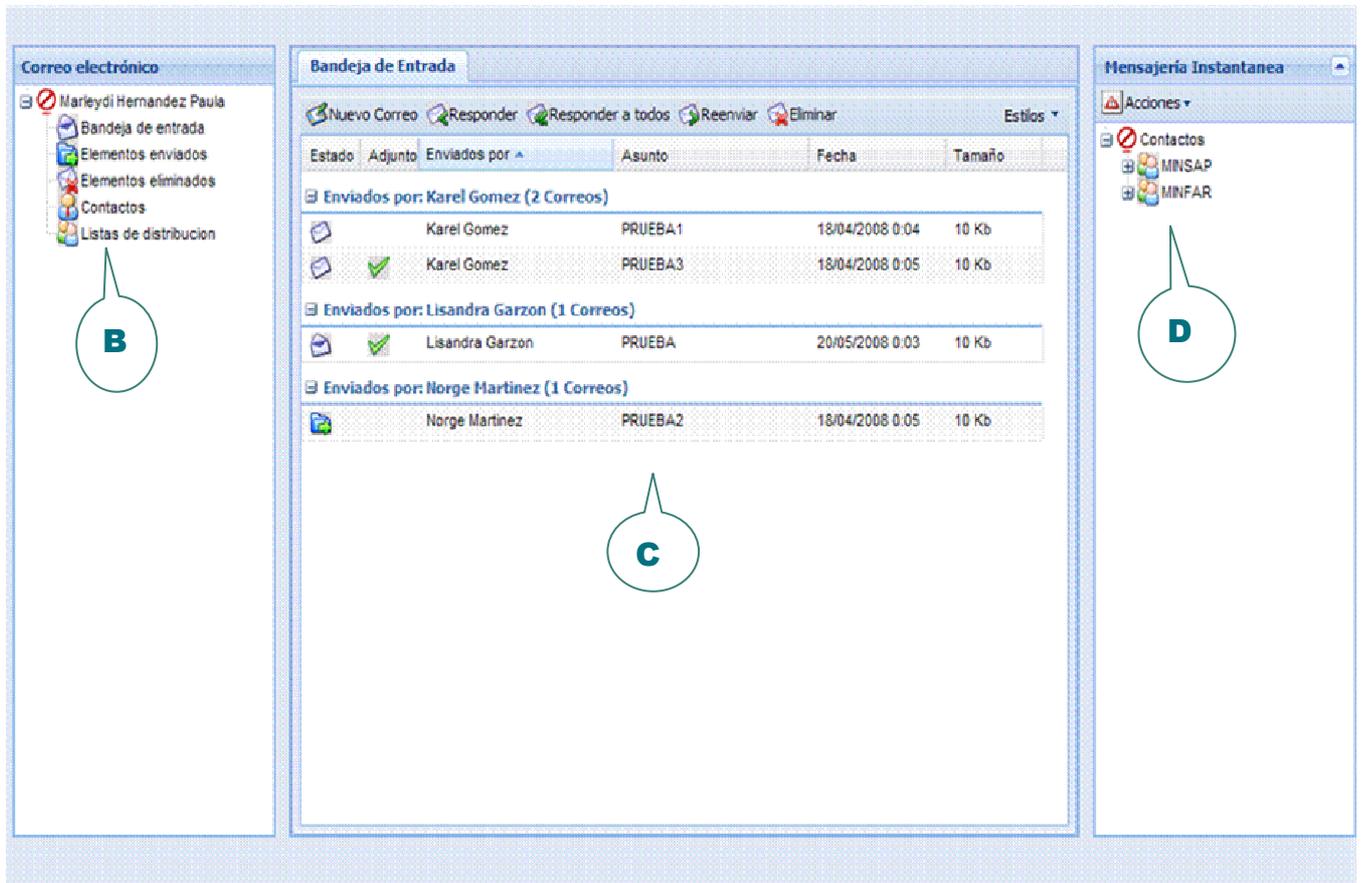


Figura 3.20 Página de Inicio.

**B:** Región donde el usuario persona puede escoger las opciones relacionadas de la mensajería asíncrona.

**C:** Región donde se muestra el contenido según la opción que haya seleccionado el usuario persona.

**D:** Región donde al usuario persona se le muestra el menú de Acciones y los diferentes grupos de usuarios que tiene su mensajería síncrona.

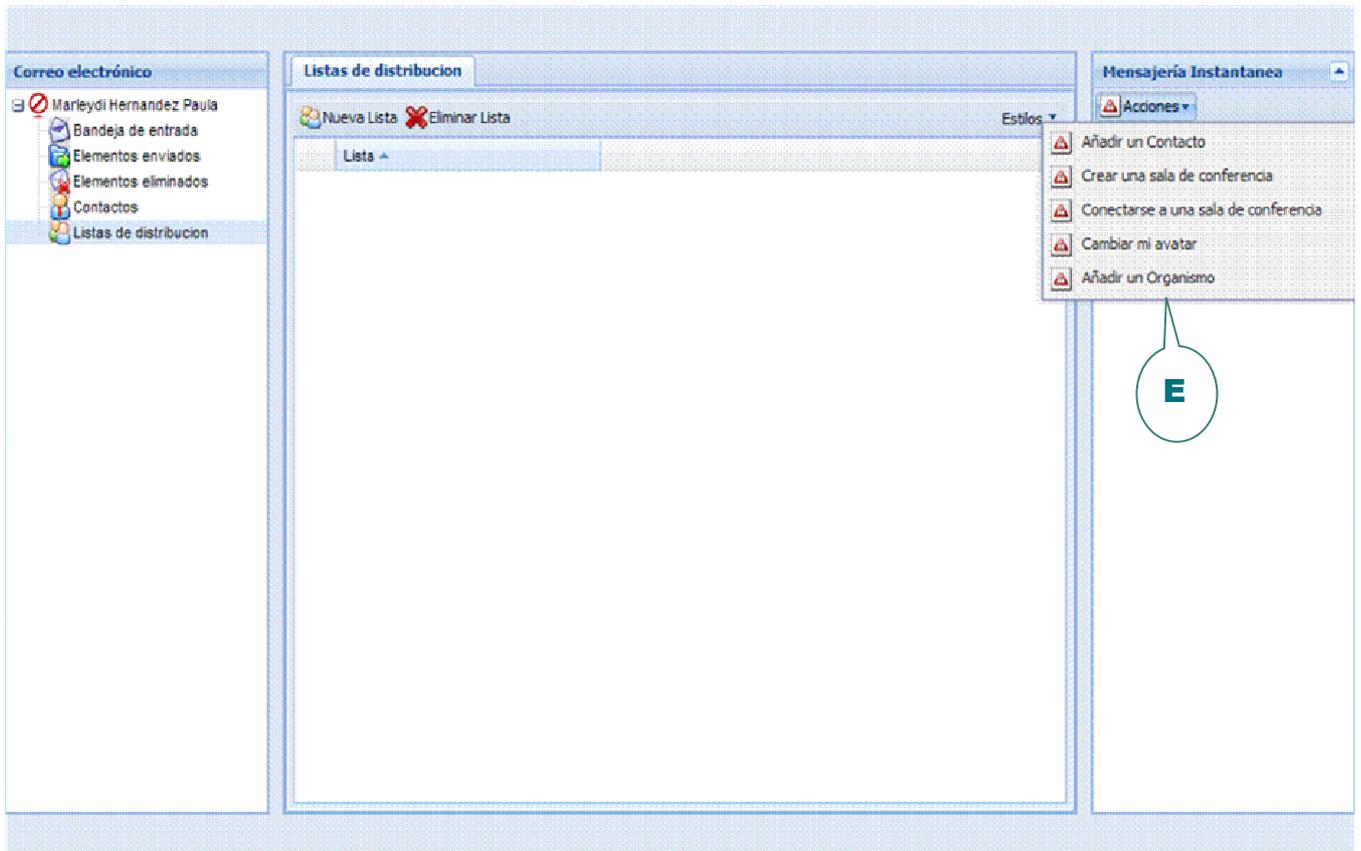


Figura 3.20 Página de Acciones.

**E:** Región donde se muestran al usuario persona las opciones que contiene el menú de Acciones de su mensajería síncrona.

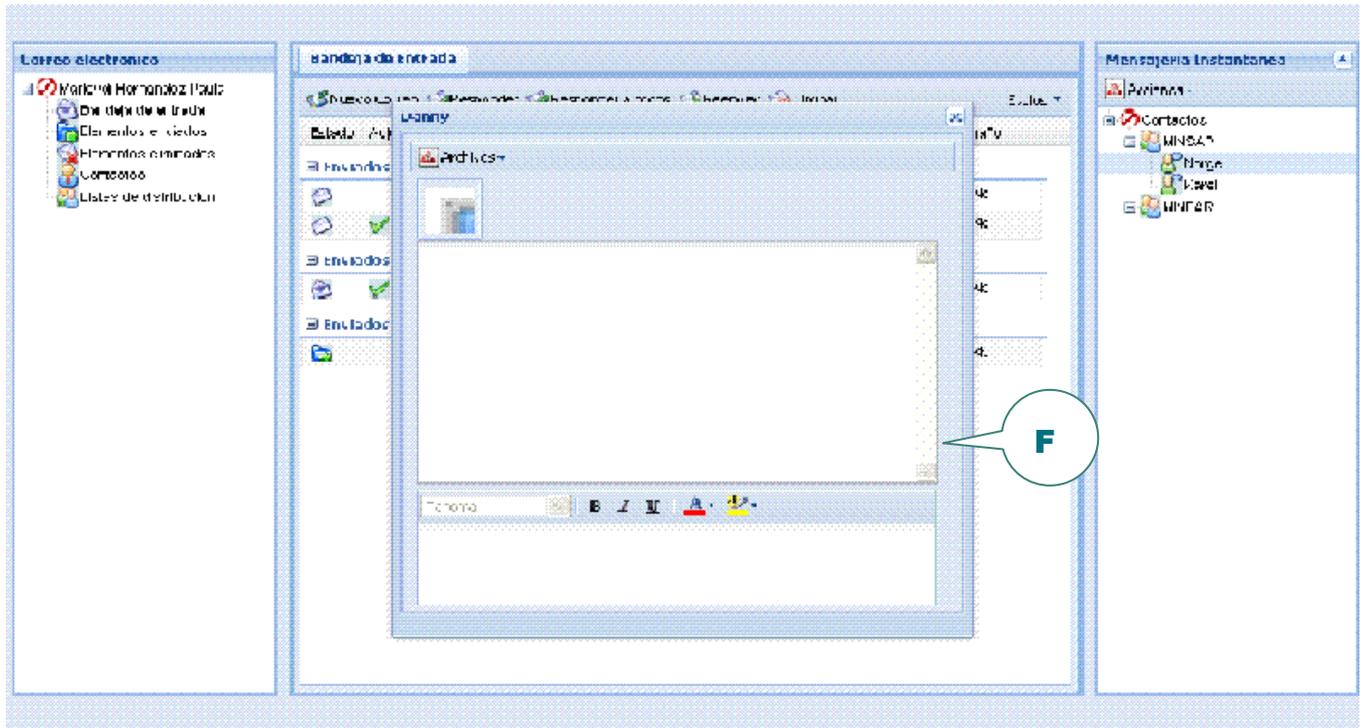


Figura 3.21 Página mensaje síncrono

**F:** Región a través de la cual el usuario persona envía mensajes síncronos.

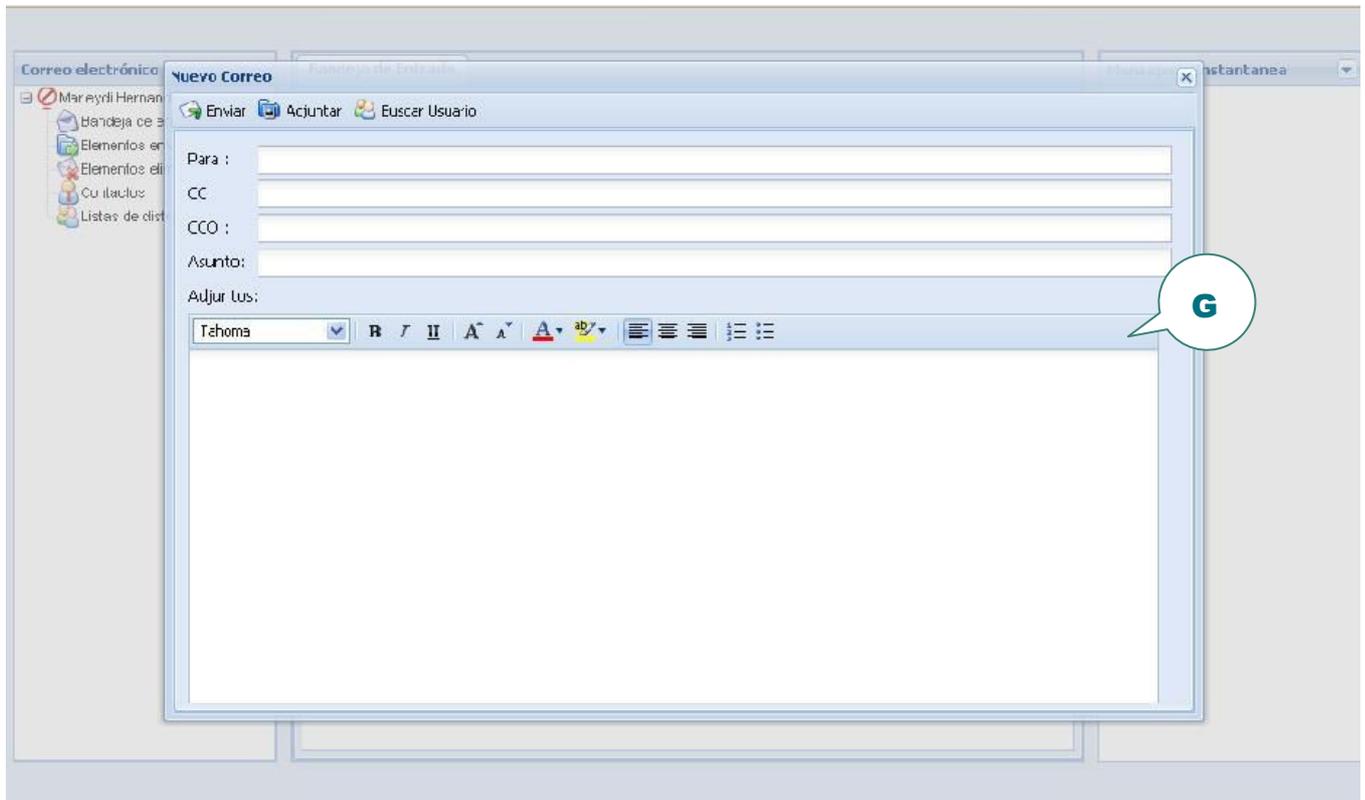


Figura 3.22 Página mensaje asíncrono.

**G:** Región a través de la cual el usuario persona envía mensajes asíncronos.

En este capítulo se documentaron los artefactos para el desarrollo de Componente de Notificación y Mensajería como los diagramas de clases de análisis y diseño, el diagrama de clases persistentes y el modelo de datos, siendo este último el paso fundamental para diseñar la base de datos del sistema. Se describió los diferentes estereotipos generados por estos diagramas para un mejor entendimiento de la información de cada uno de ellos y se le realizó un diagrama de despliegue para comprender la distribución arquitectónica del sistema.

### CONCLUSIONES

Al concluir el presente Trabajo de Diploma se le dio cumplimiento a todas las Tareas de la Investigación identificadas así como el objetivo propuesto. Finalmente se arriban a las siguientes conclusiones:

- El ambiente de desarrollo propuesto sigue los lineamientos establecidos por el MINSAP para el desarrollo de sus aplicaciones.
- Los sistemas de mensajería analizados no cumplen las cualidades como para resolver las necesidades de integración de los sistemas del MINSAP.
- El prototipo no funcional obtenido, constituye un artefacto primordial para la comprensión de los requerimientos especificados.
- Se documentaron los Modelos de Dominio, Modelo de Sistema, Modelo de Análisis y Modelo de Diseño, siendo estos la principal entrada para el Modelo de Implementación.

### RECOMENDACIONES

Para perfeccionar la presente investigación se proponen las siguientes recomendaciones:

- Incorporar nuevas funcionalidades que permitan definir capacidad máxima de almacenamiento para los historiales de mensajes y facilidades de mantenimiento a la Base de Datos.
- Implementar el Componente de Notificación y Mensajería siguiendo el diseño propuesto.
- Establecer políticas dentro del Área Temática Sistemas de Apoyo a la Salud para la integración de este componente con otros sistemas, en fase de construcción o transición.

### REFERENCIAS BIBLIOGRÁFICAS

[1] Capítulo 18 *Tecnología de la Información y las Comunicaciones (TIC)*

[http://one.cu/aec\\_web/paginas\\_de\\_textos/c\\_xviii.htm](http://one.cu/aec_web/paginas_de_textos/c_xviii.htm)

[2] *La informatización en Cuba*. 2005. Disponible en:

[http://www.cubaminrex.cu/Sociedad\\_Informacion/Cuba\\_SI/Informatizacion.htm](http://www.cubaminrex.cu/Sociedad_Informacion/Cuba_SI/Informatizacion.htm)

[3] Reynoso, Carlos; Kicillof, Nicolás. *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft*.

Universidad de Buenos Aires. 2004. Disponible en:

[http://www.microsoft.com/spanish/msdn/arquitectura/roadmap\\_arg/style.asp#10](http://www.microsoft.com/spanish/msdn/arquitectura/roadmap_arg/style.asp#10)

[4] Maldonado Segura, José Alberto et al. *Tecnologías de la información al servicio de la historia clínica electrónica*. Sociedad Española de Informática para la salud. España. 2002. p 154

[5] Ídem a la referencia [4].

[6]. SOFTEL. *Documento sobre la Arquitectura de Software a emplear en los componentes del Sistema de Información para la Salud*. La Habana. Cuba. 2007

[7] W3C Oficina Española. *Guía Breve de Servicios Web*.

Disponible en: <http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb>

[8] Servicios Web de normalización conceptual con SOAP. Disponible en:

<http://www.w3.org/2001/sw/Europe/events/200406-esp/trabajo-final-extratesauros/node17.html>

[9]. Ídem a la referencia [7].

[10] Reynoso, Carlos; Kicillof, Nicolás. *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft*. Universidad de Buenos Aires 2004.

[11] Javier Eguíluz, *Introducción a AJAX*.

## REFERENCIAS BIBLIOGRÁFICAS

---

- [12] *PostgreSQL vs. MySQL*. Disponible en: [http://www.netpecos.org/docs/mysql\\_postgres/x15.html](http://www.netpecos.org/docs/mysql_postgres/x15.html)
- [13] *HyperText Transfer Protocol*. Disponible en: <http://es.wikipedia.org/wiki/HTTP>
- [14] *Rational Rose Enterprise*. Disponible en:  
<http://www.rational.com.ar/herramientas/roseenterprise.html>
- [15] Almada, Federico. *Zend Studio for Eclipse, desarrollo profesional en PHP*. 2008.  
Disponible en: <http://www.techtear.com/2008/01/22/zend-studio-for-eclipse-desarrollo-profesional-en-php/>
- [16] Franco Navarro, José Angel. *UML en acción. Modelando Aplicaciones Web*. Instituto Superior Politécnico José Antonio Echeverría. La Habana, Cuba. Mayo 2005.

## BIBLIOGRAFÍA

Dirección de Informatización. *Arquitectura para los sistemas que conforman la intranet universitaria*. Universidad de las Ciencias Informáticas. La Habana, Cuba. Mayo 2007.

Ferm, Fredrik. *The what, why and how is a subsystem. The Rational edge, e-zine for the rational community* Junio 2003.

Disponible en: [http://www.therationaledge.com/content/jun\\_03/t\\_subsystem\\_ff.jsp](http://www.therationaledge.com/content/jun_03/t_subsystem_ff.jsp)

Franco Navarro, José Angel. *UML en acción. Modelando Aplicaciones Web*. Instituto Superior Politécnico José Antonio Echeverría. La Habana, Cuba. Mayo 2005.

García, Joaquín. *Patrones de Diseño*. 27 de mayo 2005

Disponible en: <http://www.ingenierossoftware.com/analisisydiseno/patrones-diseno.php>

Jacobson, Ivar; Booch, Grady; Rumbaugh, James. *El Proceso Unificado de Desarrollo de Software*. La Habana, Cuba. Editorial Félix Varela. 2004. p 4, 5, 6, 7.

JACOBSON, Ivar; RUMBAUGH, James; BOOCH, Grady, "El proceso unificado de desarrollo".2000. Addison Wesley.

Larman, Craig. *UML y Patrones: Introducción al análisis y diseño orientado a objetos*. La Habana, Cuba. Editorial Félix Varela. 2004. p 16,17

López ,Ernesto *¿Qué es la Mensajería Instantánea?*

Disponible

en:

[http://www.sitiosargentina.com.ar/webmaster/cursos%20y%20tutoriales/que\\_es\\_la\\_mesajeria\\_instantanea.htm](http://www.sitiosargentina.com.ar/webmaster/cursos%20y%20tutoriales/que_es_la_mesajeria_instantanea.htm)

López Medina, Mosquera Julio E; Luis J. *Arquitectura de integración para el sector financiero*. XXV Salón de Informática "Arquitecturas empresariales de software". Bogotá, Colombia. 28-01-2005.

Alvarez, Miguel Angel, ¿Qué es XML? 13 junio 2001

Disponible en: <http://www.desarrolloweb.com/articulos/449.php>

Pressman, Roger S. *Ingeniería de Software, un enfoque práctico*. Parte 1. La Habana, Cuba. Editorial Félix Varela. 2005.

Schmuller, Joseph. *Aprendiendo UML en 24 horas*. Macmillan Computer Publishing .Indiana, Estados Unidos. 2000.

Seguridad Informática, Correo Electrónico, Proteja su Información. *¿Qué es el correo electrónico?*.

Disponible

en:

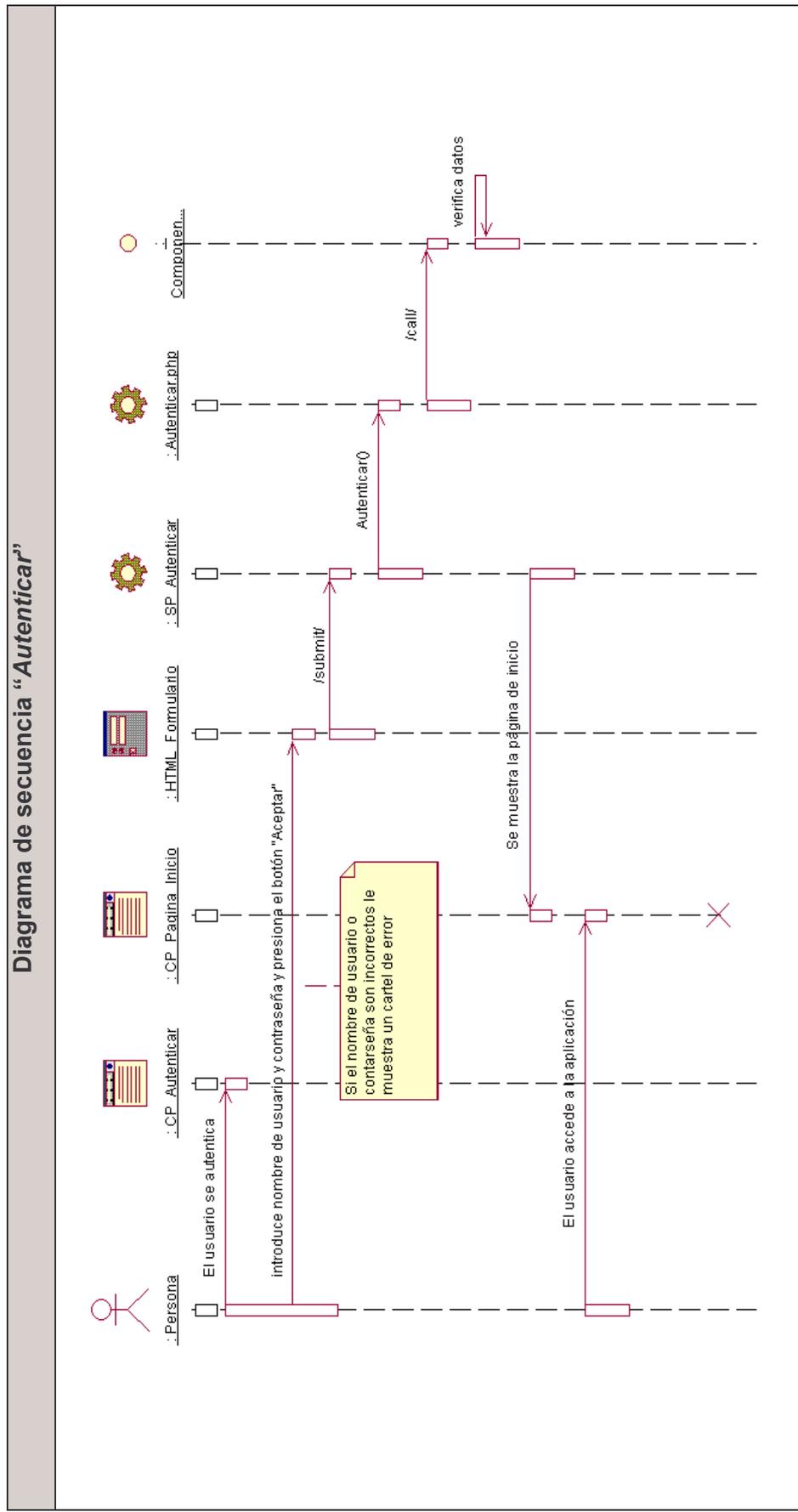
[http://caribdis.unab.edu.co/portal/page?\\_pageid=653.459470.653\\_531907&\\_dad=portal&\\_schema=PORTAL](http://caribdis.unab.edu.co/portal/page?_pageid=653.459470.653_531907&_dad=portal&_schema=PORTAL)

Saffirio, Mario. *¿Qué son los Web Services?. 5 de Febrero 2006*. Disponible en:

<http://msaffirio.wordpress.com/2006/02/05/%c2%bfque-son-los-web-services/>

ANEXOS

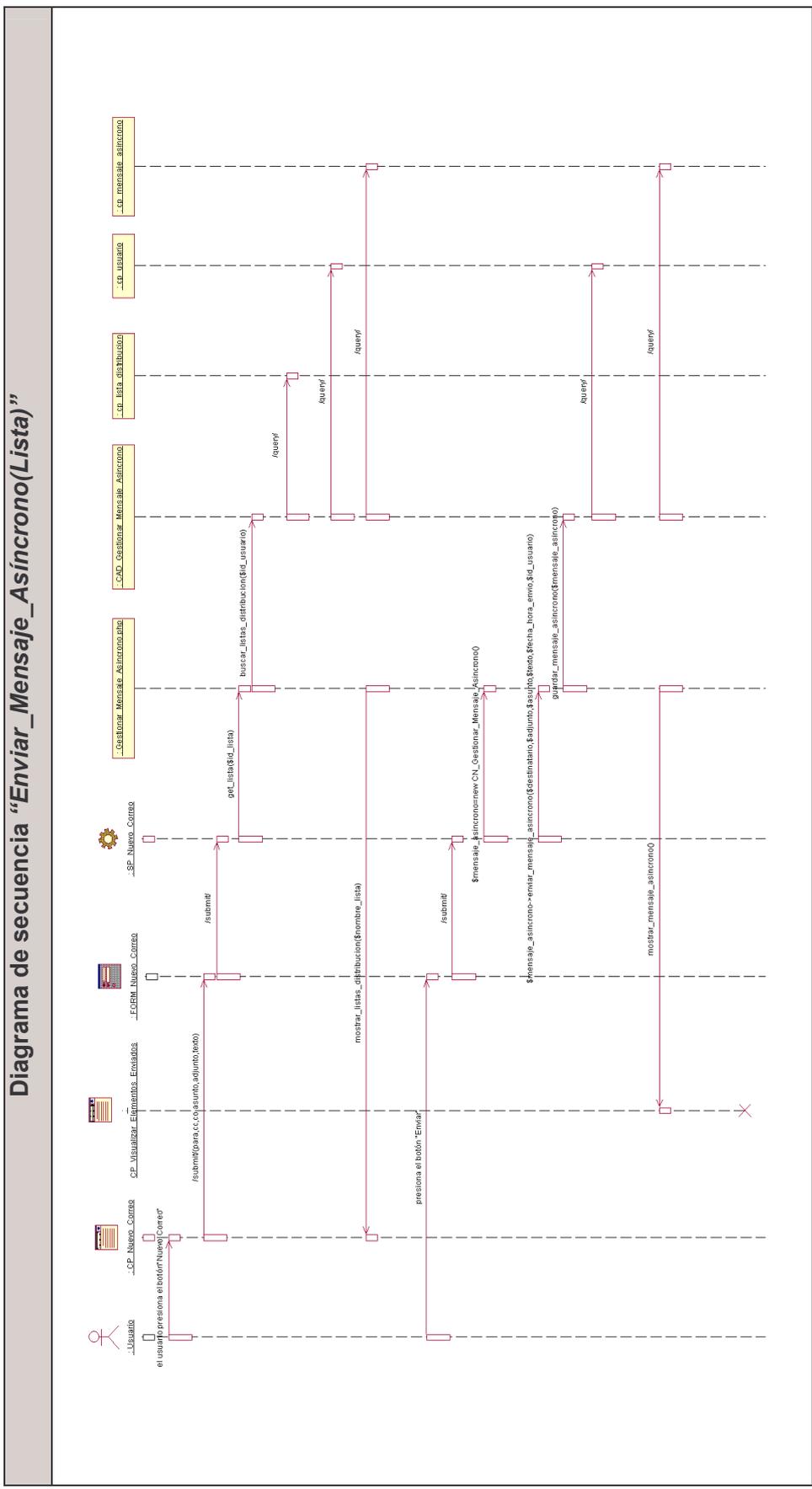
Anexo 1: Diagrama de secuencia "Autenticar"



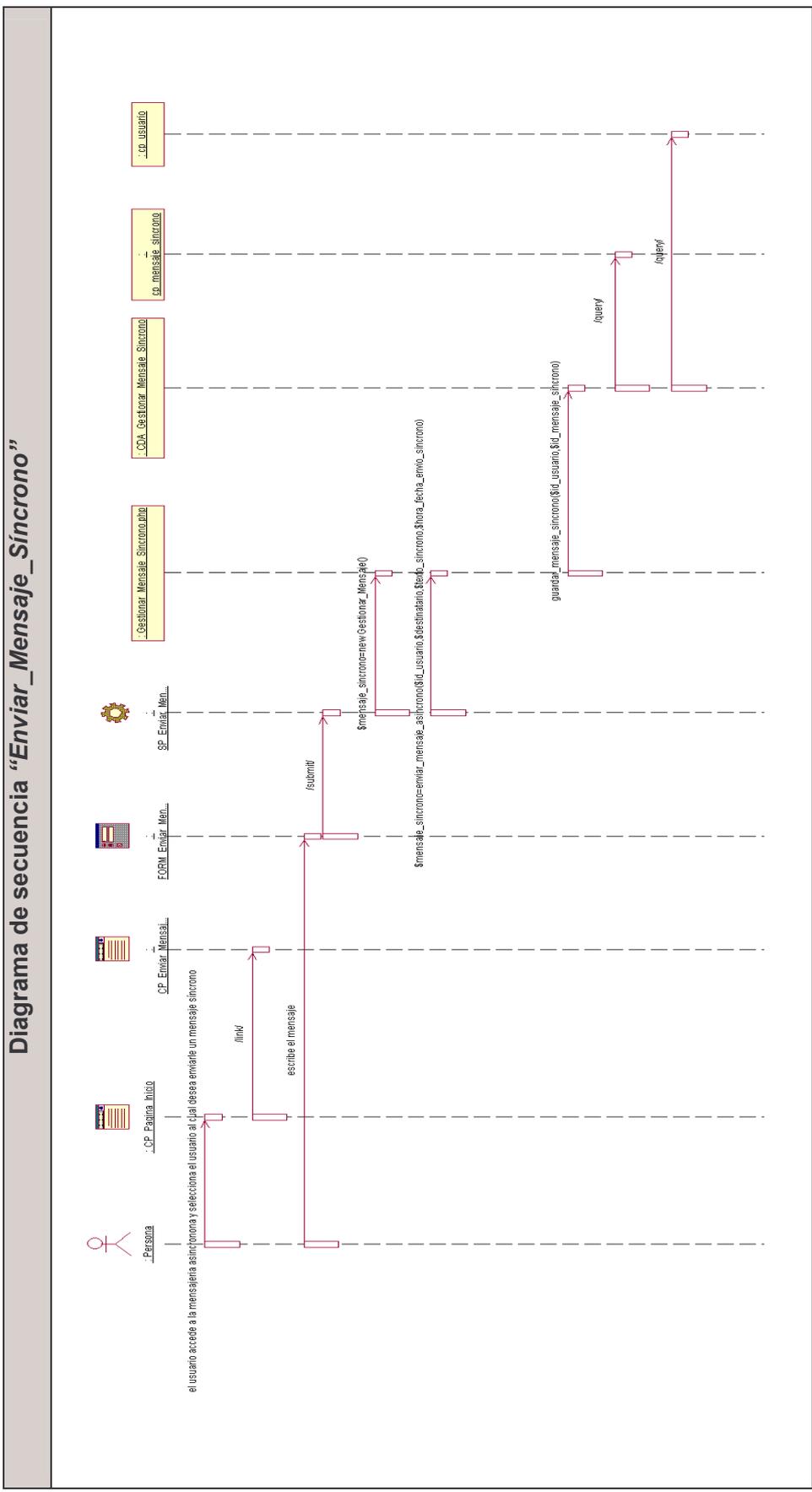




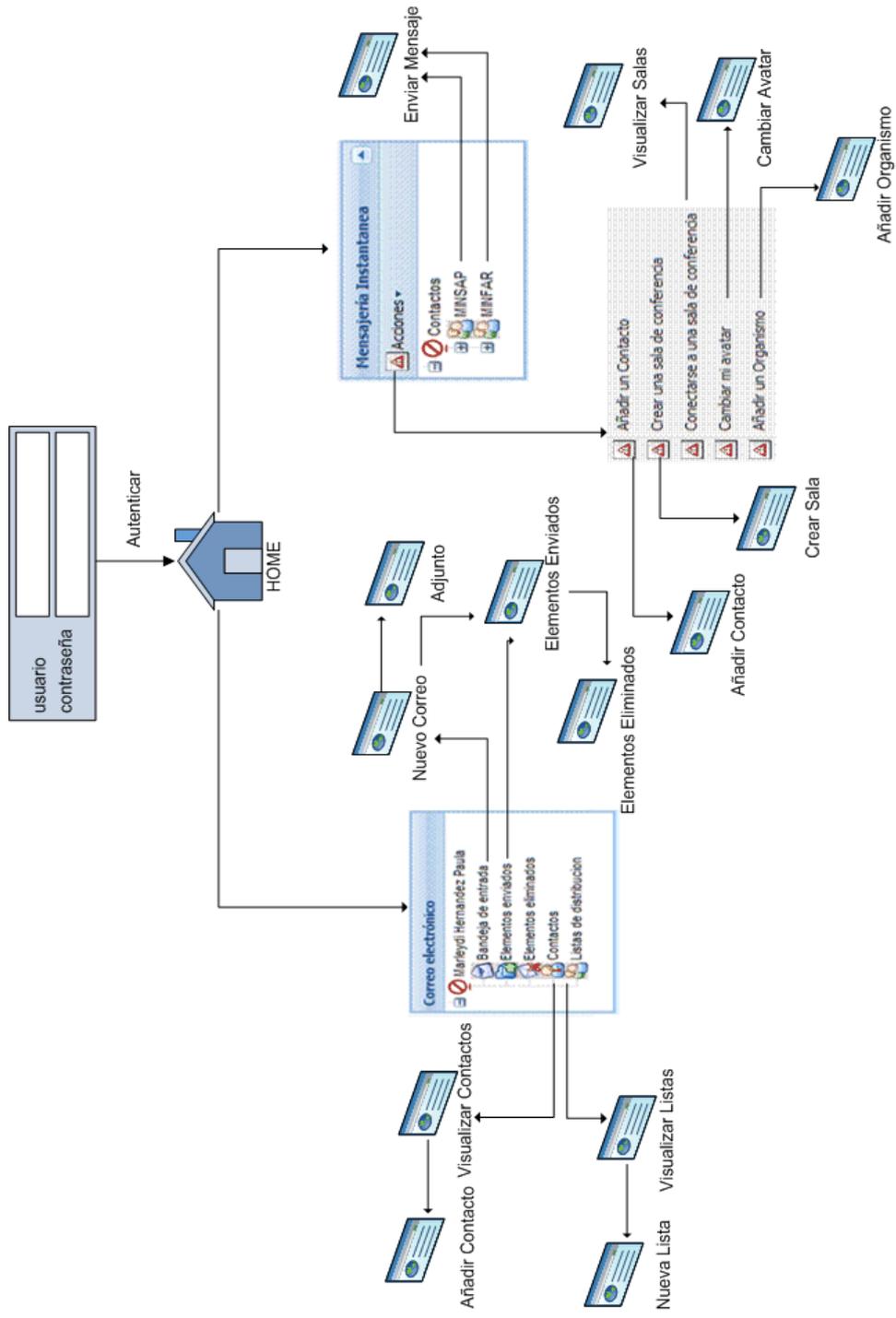
Anexo 4: Diagrama de secuencia “Enviar\_Mensaje\_Asincrono(Lista)”



Anexo 5: Diagrama de secuencia “Enviar\_Mensaje\_Síncrono”



Anexo 5 Mapa de Navegación



### GLOSARIO DE TÉRMINOS

**Array:** Colección ordenada de elementos de un mismo tipo de datos, agrupados de forma consecutiva en memoria. Cada elemento del array tiene asociado un índice, que no es más que un número natural que lo identifica inequívocamente y permite al programador acceder a él.

**Banner:** Gráfico generalmente elaborado en un formato de archivo .GIF o JPG, de aproximadamente 400 x 40 pixeles, el cual se coloca como un encabezado o una representación de una "Página en un Website".

**CCS:** Estilo Cascada. Lenguaje de hojas de estilo más utilizado en páginas web, con él podrá separar el aspecto visual del contenido de las páginas web.

**CGI:** Interfaz de entrada común. Es una importante tecnología de la World Wide Web que permite a un cliente (explorador web) solicitar datos de un programa ejecutado en un servidor web. CGI especifica un estándar para transferir datos entre el cliente y el programa. Es un mecanismo de comunicación entre el servidor web y una aplicación externa cuyo resultado final de la ejecución son objetos MIME. Las aplicaciones que se ejecutan en el servidor reciben el nombre de CGIs.

**DOM:** Modelo de Objetos del Documento (DOM) es una interfaz de programación de aplicaciones (API) para documentos HTML y XML. Define la estructura lógica de los documentos y el modo en que se accede y manipula un documento. En la especificación del DOM, el término "documento" se utiliza en un sentido amplio. Con el Modelo de Objetos del Documento los programadores pueden construir documentos, navegar por su estructura, y añadir, modificar o eliminar elementos y contenido. Se puede acceder a cualquier cosa que se encuentre en un documento HTML o XML, y se puede modificar, eliminar o añadir usando el Modelo de Objetos del Documento.

**EBML:** Extensible Binary Meta Language (Meta Lenguaje Binario Extendible), fue diseñado como una extensión binaria simplificada de XML, con el propósito de almacenar y manipular datos de forma jerárquica con campos de longitud variable.

**GNOME:** Entorno de escritorio para sistemas operativos de tipo Unix bajo tecnología X Windows.

**GRASP:** Acrónimo de General Responsibility Assignment Software Patterns (Patrones de Software para la asignación General de Responsabilidad) describen los principios fundamentales de diseño de objetos para la asignación de responsabilidades

**HTML:** Lenguaje de Marcas de Hipertexto, (HyperText Markup Language) es el lenguaje de marcado predominante para la construcción de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes.

**HTTP:** Lenguaje para marcado de hipertexto". Más claro aún, se trata de un lenguaje para estructurar documentos a partir de texto en World Wide Web. Este lenguaje se basa en tags (instrucciones que le dicen al texto como deben mostrarse) y atributos (parámetros que dan valor al tag).

**Iframe:** Frames flotantes sirve para crear un espacio dentro de la página donde se puede incrustar otra web. Es un cuadrado cuyas dimensiones debe especificar el desarrollador en la propia etiqueta iframe, que tiene asociada una página web que se carga en dicho espacio. Esa página web tendrá sus propios contenidos y estilos, independientes del contexto donde se está mostrando. Además será perfectamente funcional: si tiene enlaces se mostrarán en ese mismo espacio y si tiene scripts o aplicaciones dentro se ejecutarán también de manera autónoma en el espacio reservado al iframe.

**IMAP:** Internet Message Access Protocol es un protocolo de red de acceso a mensajes electrónicos almacenados en un servidor. Mediante IMAP se puede tener acceso al correo electrónico desde cualquier equipo que tenga una conexión a Internet.

**JQUERY:** Query es una librería escrita en JavaScript que te ayuda para trabajar con AJAX, crear animaciones, bonitos efectos y un sin fin de cosas con JavaScript en una página web y de una manera muy sencilla, eso sí la sintaxis, o sea la manera de escribir el código, es muy diferente a la del JavaScript normal.

**LGPL:** La Licencia Pública General Reducida de GNU la usan algunas, pero no todas, las bibliotecas GNU. Esta licencia fue llamada en un principio GPL para bibliotecas pero le cambiamos el nombre debido a que animaba a la gente a emplear esta licencia más de lo debido.

**PHP:** Hypertext Pre-processor es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas.

**POP:** Post Office Protocol (POP3) en clientes locales de correo para obtener los mensajes de correo electrónico almacenados en un servidor remoto. La mayoría de los suscriptores de los proveedores de Internet acceden a sus correos a través de POP3.

**Prototype:** Framework escrito en JavaScript que se orienta al desarrollo sencillo y dinámico de aplicaciones web.

**SHELL:** Es un término utilizado para identificar una interface o la parte más externa de un programa. Algunas aplicaciones y sistemas operativos proveen diferentes *shells* o frentes para que al usuario le sea más sencillo interactuar con un programa.

**SMTP:** Simple Mail Transfer Protocol, o protocolo simple de transferencia de correo. Protocolo de red basado en texto utilizado para el intercambio de mensajes de correo electrónico entre computadoras o distintos dispositivos.

**SOAP:** Siglas de *Simple Object Access Protocol*. Protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. SOAP es uno de los protocolos utilizados en los servicios web.

**S/MIME:** (Secure / Multipurpose Internet Mail Extensions, del inglés, Extensiones de Correo de Internet de Propósitos Múltiples / Seguro) es un estándar para criptografía de clave pública y firmado de correo electrónico.

**UDDI:** Siglas del catálogo de negocios de Internet denominado *Universal Description, Discovery and Integration*. es uno de los estándares básicos de los servicios Web cuyo objetivo es ser accedido por los mensajes SOAP y dar paso a documentos WSDL, en los que se describen los requisitos del protocolo y los formatos del mensaje solicitado para interactuar con los servicios Web del catálogo de registros.

**Webmail:** Correo electrónico *de sitio web, correo basado en web o correo web*, es un servicio que permite acceder a tu cuenta de correo electrónico a través de una página web utilizando un navegador y sin descargar los mensajes al propio ordenador.

**XMLHttpRequest:** *Extensible Markup Language / Hypertext Transfer Protocol*, es una interfaz empleada para realizar peticiones HTTP a servidores WEB.

**YUI:** Yahoo User Interface(YUI), una serie de librerías escritas en JavaScript, para la construcción de aplicaciones interactivas.