

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



**Sistema de Autenticación, Autorización y Auditoría para  
los productos desarrollados en el Área Temática Sistemas  
de Apoyo a la Salud (SAAA-SAS) v1.1**

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

**Autores:** Yudmila Ferrera Ramírez  
Aramis Souлары Pelegrín

**Tutores:** Ing. Danisbel Rojas Ríos  
Ing. Hector Manuel Solis Mulet

**Ciudad de La Habana, Junio de 2009**  
**“Año del 50 aniversario del triunfo de la Revolución”**

## DECLARACIÓN DE AUTORÍA

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

Para que así conste firmamos la presente a los 22 días del mes de junio del año 2009.

---

Yudmila Ferrera Ramírez

**Autora**

---

Aramis Soulary Pelegrín

**Autor**

---

Ing. Danisbel Rojas Ríos

**Tutor**

---

Ing. Hector Manuel Solis Mulet

**Tutor**

## DATOS DE CONTACTO

**Ing. Danisbel Rojas Ríos** ([drojas@uci.cu](mailto:drojas@uci.cu)): Recién graduado con Título de Oro de Ingeniero en Ciencias Informáticas en la UCI. Posee Categoría Docente de Profesor Adiestrado. Ha participado en proyectos de desarrollo de Sistemas Informáticos para la Salud desde el año 2005. Imparte la asignatura de Arquitectura para Sistemas de Salud. Actualmente se desempeña como miembro del Centro de Consultoría TI de la UCI.

**Ing. Hector Manuel Solis Mulet** ([hmsolis@uci.cu](mailto:hmsolis@uci.cu)): Recién graduado con Título de Oro de Ingeniero en Ciencias Informáticas en la UCI. Posee Categoría Docente de Profesor Adiestrado. Ha participado en proyectos de desarrollo de Sistemas Informáticos para la Salud desde el año 2005. Imparte la asignatura de Bases de Datos. Actualmente se desempeña como miembro del grupo de Bases de Datos del Área Temática Sistemas de Apoyo a la Salud.

## **AGRADECIMIENTOS**

*Agradecemos a nuestras familias por apoyarnos y confiar en nosotros, a los tutores por ayudarnos y guiar nuestro trabajo, a la profe Kina y a Javier por su ayuda y preocupación, al oponente y al resto del tribunal, por las recomendaciones realizadas para que el trabajo tuviera la calidad requerida.*

### **De Yudmila:**

*A mis padres, por todo el amor, por la confianza y el apoyo que me han brindado siempre, gracias porque hicieron posible alcanzar mi sueño.*

*A Tita, por ser mi segunda madre, por apoyarme siempre y por ser mi ángel guardián.*

*A mis hermanitas, por ser mi inspiración y mi motivación.*

*A mi Tito, por tu amor, dedicación, cariño y tu ayuda incondicional. Gracias mi chiquito, por estar siempre a mi lado y por ser tú. A tu familia por su apoyo y preocupación.*

*A mi compañero de tesis Aramis, por todo el esfuerzo y dedicación para que este trabajo se realizara.*

*A mis tíos y a mis primas por el apoyo y cariño.*

*A todos mis amigos, especialmente a Ariagne (Cuquita), Yunelis, Lisseidy, Elaine, Dayrelis, Amarilis, Ayali, Clenda, Elizabeth, Arianna R, Idalmys y Lisaime, porque de cada una aprendí algo, por la gran familia que formamos, por todo el apoyo y cariño. A todos muchas graciasssss.*

### **De Aramis:**

*A mi compañera de tesis Yudmila por su disposición, por haber compartido conmigo todo este tiempo, por ser una amiga, verdaderamente fue un placer realizar la tesis contigo.*

*A Walfrido Serrano Pérez por su disposición, y ayuda durante el desarrollo de la tesis. Mil gracias hermano.*

*A Yusi (mi cosita) por siempre darme las fuerzas para seguir adelante, y estar siempre conmigo.*

*A tus padres por todo el apoyo y cariño que me brindaron.*

*A mi tíos Carlos, Laritza y a mi cuñada Betsy por su apoyo todos estos años, y estar junto a mí cuando más los necesité, les agradeceré eternamente.*

*A mis amigos de siempre: Claritza, Yoangel, Anelis, Adamis y Daylenis, gracias por estar ahí cuando los necesité, y apoyarme en todo momento.*

*A mis amigos: Mario, Dannier, Yordanger, David y Karel por todos los buenos momentos de esta amistad.*

*A lo más bello que puede existir, mi madre y mi hermano. Sin ustedes jamás esto sería posible.*

## DEDICATORIA

### *De Yudmíla:*

*Dedicado a las personas más importantes de mi vida, los que han hecho posible que yo haya llegado hasta aquí. A ustedes con todo mi amor, a los pedacitos de mi corazón:*

*Papí y mamá: por la vida, porque ustedes son lo más grande que tengo y por guiarme siempre.*

*Tita, por luchar todos estos años conmigo, por hacer tanto por mí y por ser mi sol.*

*Mis hermanas Wendy y Yuli: por ser lo más bello que me ha regalado la vida.*

*Mi Tito por ser el amor de mi vida, por estar siempre conmigo y por ocupar una parte tan importante en mi corazón.*

### *De Aramis:*

*Especialmente a mi madre y a mi hermano por ser todo lo que más quiero en esta vida y por guiarme a alcanzar el sueño de mi vida.*

*A Yusi por ser lo más bello que he tenido y representar todo en mi vida.*

*A mi querida abuela Marianita que no logró verme hecho ingeniero...*

*Al amigo Buchí, al amigo de todos nosotros, que no pudo alcanzar este sueño.*

### RESUMEN

La investigación del presente Trabajo de Diploma está centrada en obtener una segunda versión del Sistema de Autenticación, Autorización y Auditoría para los productos desarrollados en el Área Temática Sistemas de Apoyo a la Salud, que permita el mejoramiento sustancial del rendimiento de la versión precedente. Además que proporcione mayores prestaciones en la Autenticación, Autorización y Auditoría para que pueda ajustarse con mayor facilidad a los requerimientos de seguridad de las soluciones informáticas de esta área temática.

Su desarrollo está basado sobre una arquitectura en capas, utiliza PHP 5 como lenguaje de programación, el patrón de arquitectura Modelo Vista Controlador. Así como, PostgreSQL 8.3 como Sistema Gestor de Bases de Datos, tecnología AJAX y la librería YUI para obtener una interfaz visual moderna. Utiliza estándares abiertos como XML lo que permite la interoperabilidad entre aplicaciones desarrolladas sobre diferentes plataformas.

El sistema cuenta con un servicio web que utiliza el patrón proxy, este sirve de interfaz a las peticiones de los sistemas clientes y garantiza el acceso seguro a los mismos. Propone un mecanismo automatizado que permite balancear la información de las trazas generadas por los usuarios. Presenta una mejor apariencia visual de los reportes relacionados con la Auditoría, lo que permite al usuario seleccionar los elementos que le resulten de mayor interés y muestra la información de forma más intuitiva a través de gráficos. Además brinda un módulo de instalación que facilita este proceso para aquellos usuarios con pocos conocimientos informáticos.

**Palabras Claves:** Auditoría, Autenticación, Autorización, Proxy, Seguridad, Servicios.

ÍNDICE

INTRODUCCIÓN ..... 1

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA ..... 7

    1.1 Objeto de automatización ..... 7

    1.2 Seguridad en aplicaciones web ..... 8

    1.3 Ataques sobre aplicaciones Web ..... 8

    1.4 Protocolo LDAP ..... 9

    1.5 Tendencias y tecnologías actuales ..... 10

        1.5.1 Sistemas distribuidos. Modelo Cliente – Servidor ..... 10

        1.5.2 Patrones de arquitectura y diseño ..... 12

        1.5.3 Lenguajes utilizados para el proceso de desarrollo ..... 15

        1.5.4 Tecnologías utilizadas en el proceso de desarrollo ..... 20

        1.5.5 Protocolo HTTPS ..... 23

        1.5.6 Sistema Gestor de Base de Datos ..... 23

        1.5.7 Servidor Web Apache ..... 26

        1.5.8 Metodología Proceso Unificado de Desarrollo de Software (RUP) ..... 26

        1.5.9 Herramientas a utilizar ..... 27

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA ..... 31

    2.1 Modelo de Dominio ..... 31

        2.2.1 Conceptos fundamentales ..... 31

        2.2.2 Diagrama del Modelo de Dominio ..... 33

    2.3 Propuesta de Sistema ..... 33

        2.3.1 Especificación de los Requerimientos de Software ..... 34

        2.3.2 Modelo de Casos de Uso del Sistema ..... 39

CAPÍTULO 3: DISEÑO DEL SISTEMA ..... 55

    3.1 Modelo de Diseño ..... 55

        3.1.1 Estructura de Diseño ..... 55

        3.1.2 Definición de Elementos de Diseño ..... 56

        3.1.3 Diagramas del Diseño ..... 59

        3.1.4 Descripción de las clases del diseño ..... 64

    3.2 Modelo de Datos ..... 65

        3.2.1 Descripción de las tablas de la base de datos ..... 67

CAPÍTULO 4: IMPLEMENTACIÓN ..... 69

    4.2 Modelo de Implementación ..... 69

        4.2.1 Diagramas de Componentes ..... 69

        4.2.2 Diagrama de Despliegue ..... 72

    4.3 Descripción de la solución propuesta ..... 74

4.3.1	Servicio web Proxy .....	74
4.3.2	Pruebas de rendimiento.....	76
4.3.3	Mecanismo de balanceo de trazas en la base de datos.....	79
4.3.4	Diseño de los Reportes de Auditoría.....	80
4.3.5	Módulo de Instalación .....	81
	CONCLUSIONES .....	83
	RECOMENDACIONES .....	84
	REFERENCIAS BIBLIOGRÁFICAS.....	85
	BIBLIOGRAFÍA.....	89
	ANEXOS.....	91
	GLOSARIO DE TÉRMINOS .....	108



## INTRODUCCIÓN

En la actualidad, en diferentes sectores de la sociedad cubana se utilizan sistemas informáticos para la gestión de la información, como parte del programa de informatización que se lleva a cabo en el país. Por lo que es de vital importancia aplicar políticas o medidas que permitan llevar un control de la seguridad de dichos sistemas. La base de la seguridad informática está centrada en garantizar que los recursos de software de una organización, empresa o institución se usen únicamente para los propósitos que fueron creados y dentro del marco previsto y que el acceso y modificación de la información manejada por estas entidades, solo sea posible a las personas que se encuentren acreditadas y dentro de los límites de su autorización.

Existen varias instituciones cuya misión es el desarrollo de software para la industria nacional cubana. Algunos de los más destacados exponentes de estos son: la Empresa Nacional de Software (DESOFT), la Empresa de Soluciones Informáticas (SOFTTEL) y la Universidad de las Ciencias Informáticas (UCI), que cuenta con varias Áreas Temáticas que dirigen su producción hacia diferentes líneas de desarrollo, una de ellas es Sistema de Apoyo a la Salud (SAS) de la facultad 7, que se dedica a la producción y soporte de aquellos productos de software que informatizan los procesos no asistenciales del Ministerio de Salud Pública (MINSAP).

La seguridad de los productos que se desarrollan en dicha Área Temática es gestionada por un sistema externo, no teniendo control sobre todas las acciones que se realizan sobre el mismo, pues este es no configurable para los componentes que lo utilizan, en cuanto a los privilegios de acceso, el control de las acciones que se realizan por cada componente, y el monitoreo de las trazas. Debido a la gran cantidad y naturaleza de la información manipulada por los sistemas desarrollados por esta área, se debe tener en cuenta mecanismos que permitan la gestión de requerimientos de seguridad que cada sistema necesita. Por lo que es necesario restringir el acceso de los usuarios para evitar la pérdida o modificación de los datos.

Se han desarrollado varios sistemas para el control de la seguridad en productos de software cubano entre ellos se encuentra el Sistema de Autenticación, Autorización y Auditoría (SAAA), como componente

de seguridad para las aplicaciones desarrolladas con el fin de informatizar el Sistema Nacional de Salud (SNS), el mismo en su versión actual, no gestiona los requerimientos de seguridad para proyectos como los desarrollados en el Área Temática SAS que está integrada por componentes heterogéneos en cuanto a: negocio, información manipulada y distribuciones físicas.

El SAAA presenta un conjunto de limitaciones: en primer lugar, limitado nivel para la administración de usuarios, puesto que los tipos de usuario están predefinidos (administrador, editor y visualizador); y no pueden ser creados nuevas clasificaciones. En segundo lugar, los roles actuales representan el privilegio que puede ser asignado al usuario sobre las funcionalidades del sistema y no la ocupación que desempeña en el SNS. Por último, varios usuarios pueden acceder a la aplicación simultáneamente, usando la misma credencial de acceso y se otorga la prioridad al último autenticado, trayendo como consecuencia, la privación del acceso al primer usuario. (1)

Ante las limitaciones señaladas, en el año 2007 se creó el Centro de Control para el Sistema de Información para la Salud. Donde se desarrolló una versión renovada del componente de seguridad SAAA, este heredó los requerimientos de su antecesor y soluciona algunas de las dificultades ya identificadas, además de agregar nuevas funcionalidades, algunas de ellas son:

- Homologación de datos de los usuarios con el Registro de Ciudadanos debido a necesidades de los nuevos módulos en desarrollo.
- El perfeccionamiento de los procesos de gestión de usuarios y asignación de privilegios.
- La eliminación de los usuarios y sus respectivas trazas por el sistema, es lógica y no física, fortaleciéndose el proceso de auditoría.
- Generación automática de ficheros de configuración y despliegue.

A pesar de que este nuevo componente consigue mejorar las prestaciones del SAAA, aún existen algunas limitaciones:

- Un usuario no puede jugar diferentes roles de gestión a diferentes niveles jerárquicos con un mismo conjunto de credenciales de acceso.

- Un usuario no puede jugar diferentes roles de gestión en un mismo nivel para un módulo determinado.
- Las trazas tienden a crecer con mucha rapidez y actualmente no se tiene en cuenta el tiempo por el cual serán almacenadas las mismas físicamente o una cantidad máxima de almacenamiento, lo que provoca esto mantener un balanceo manualmente de la información en la base de datos.
- No se pueden hacer reportes sobre las trazas históricas de la aplicación.
- No existe un codificador de roles de usuarios.
- No permite organizar los módulos en grupos de acuerdo al objetivo por el cual fueron creados.

En el año 2008 la dirección del Área Temática SAS decidió elaborar un nuevo SAAA-SAS que unificara las fortalezas de sus sistemas predecesores y permitiera gestionar los requerimientos de seguridad de los productos que allí se desarrollan. Este sin dudas logró solventar las dificultades antes mencionadas y se logró ajustar en gran medida a las características deseadas pero aún así posee algunas debilidades tales como:

- Los datos referentes a las trazas generadas por los usuarios tienden a crecer con mucha rapidez, y actualmente el sistema no brinda un mecanismo automatizado que controle la acumulación de este tipo de información en la base de datos.
- La información mostrada en los reportes generados en los servicios de auditoría, no muestra todos los elementos relacionados con las trazas, presenta solo un conjunto fijo de datos de forma poco intuitiva impidiéndole al usuario elegir la información que realmente necesita.

Actualmente el Área Temática SAS no utiliza la versión existente de su SAAA-SAS, por lo que la situación en cuanto a la gestión de los requerimientos de seguridad de sus proyectos, siguen siendo las mismas que antes de la realización del sistema.

En los proyectos del Área Temática SAS, la gestión de la seguridad resulta de vital importancia para su correcto funcionamiento y para el manejo de sus negocios individuales. El control de la seguridad se

realiza de forma independiente en cada proyecto, no existe una estandarización en cuanto a la implementación y utilización de los procesos de seguridad. Los privilegios de los distintos niveles de visibilidad, se asignan individualmente a cada usuario. Los roles son estáticos debido a la no existencia de un codificador de roles y además no se pueden crear grupos de usuarios. (2)

El registro de las trazas históricas consta de una implementación muy básica. No es lo suficientemente profundo, como para llegar a la raíz de todas las acciones que realizan los usuarios dentro del sistema, debido a que no identifica individualmente los accesos y peticiones establecidos de los que pudieran ser no autorizados. No admite que se realicen búsquedas exactas o por parámetros definidos que permitan identificar rápidamente alguna violación o problema que se haya presentado. No permite hacer reportes personalizados de acuerdo a criterios previamente seleccionados. (3)

La problemática antes expuesta permite el planteamiento del siguiente **problema a resolver**: ¿Cómo lograr que el SAAA-SAS brinde los requerimientos de seguridad identificados para los productos desarrollados en el Área Temática SAS?

Con vista a dar solución al problema planteado queda definido como **objeto de estudio**: Proceso de desarrollo de sistemas informáticos en el Área Temática SAS. Como **campo de acción**: Proceso de gestión de requerimientos de seguridad para los productos del Área Temática SAS.

Teniendo como **objetivo general**: Desarrollar una segunda versión del SAAA-SAS que brinde los requerimientos de seguridad identificados para los productos desarrollados en el Área Temática SAS.

Para lograr el cumplimiento del objetivo se proponen las siguientes **tareas de la investigación**:

1. Asimilar la arquitectura definida para el desarrollo del SAAA-SAS, Orientada a Servicios y Basada en Componentes (SOA-CBA).
2. Asimilar las tendencias y tecnologías actuales para realizar el proceso de implementación de dicha solución: Ejemplos: PHP, XML, XML Web Services, AJAX, Smarty, PostGreSQL, XHTML y JavaScript.

3. Analizar los procesos de negocio actuales de Autenticación, Autorización y Auditoría en los componentes desarrollados en el Área Temática SAS.
4. Realizar la implementación del patrón proxy, un servicio web que sirva de interfaz a las peticiones de los clientes y garantice el acceso seguro a los mismos.
5. Analizar el rendimiento del sistema cuando aumente la cantidad de trazas almacenadas e implementar una política de balanceo para las tablas asociada a las mismas.
6. Mejorar el diseño de los reportes asociados a la Auditoría.
7. Implementar un módulo de instalación personalizado para este sistema.

De forma general, la realización del SAAA-SAS aporta beneficios tales como:

- Mayor gestión de los requerimientos de seguridad para los productos desarrollados en el Área Temática SAS, mediante un servicio web desarrollado que utiliza el patrón proxy que sirve de interfaz a las peticiones de los sistemas clientes y garantiza el acceso seguro a los mismos.
- Mejoramiento del rendimiento del sistema aún cuando el volumen de datos relacionados con la información auditable crezca, debido a la implementación de una política de balanceo de trazas para las tablas donde se almacena la información de esta naturaleza.
- Mejor apariencia visual de los reportes relacionados con la Auditoría, que permite al usuario seleccionar los elementos que le resulten de mayor interés y mostrar la información de forma más intuitiva.
- 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 seguridad.

Este trabajo está estructurado en cuatro capítulos, los que se describen a continuación:

En el Capítulo 1 **Fundamentación teórica**: se abordan los diferentes aspectos para la realización de este trabajo. Se realiza una caracterización de las tecnologías y metodologías sobre las cuales se hará la aplicación, así como las herramientas que se utilizarán para el desarrollo de la misma.

En el Capítulo 2 **Características del sistema**: se tiene en cuenta la realización del marco conceptual para un mejor entendimiento del sistema, definiendo además los requisitos funcionales que debe cumplir la aplicación y los casos de uso del sistema.

El Capítulo 3 **Diseño del sistema**: profundiza en el diseño del sistema, así como en la elaboración de los diagramas de clases y de interacción, de los casos de uso detallados en el capítulo anterior.

En el Capítulo 4 **Implementación**: se describe cómo se va a implementar los elementos del diseño y se realiza el diagrama de componentes general y el diagrama de despliegue.

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Este capítulo tiene como objetivo abordar los diferentes aspectos teóricos que son la base conceptual para el desarrollo de esta aplicación. Para un mejor entendimiento se exponen conceptos fundamentales, y se caracteriza las diferentes tecnologías, metodologías y herramientas de software sobre las cuales se desarrolla el sistema.

### 1.1 Objeto de automatización

Un sistema con el objetivo de gestionar los requerimientos de seguridad de aplicaciones Web, apoyándose en algunos elementos esenciales con el fin de lograr el propósito de su implementación. Se tiene en cuenta el control de acceso que constituye una poderosa herramienta para proteger la entrada a una aplicación completa o solo a ciertos directorios concretos. Este control consta generalmente de dos pasos: (4)

- En primer lugar, la **autenticación**, es el proceso de intento de verificar la identidad digital del remitente de una comunicación como una petición para conectarse. El remitente siendo autenticado puede ser una persona que usa un ordenador, un ordenador por sí mismo o un programa del ordenador. Es el acto de establecimiento o confirmación de algo (o alguien) como auténtico, es decir, es verificar que el usuario sea quien dice ser.
- En segundo lugar, procede la cesión de derechos, es decir, la **autorización**, que dota al usuario de privilegios para poder efectuar ciertas operaciones con los datos protegidos, tales como leerlos, modificarlos, crearlos, etc.

Implementar una fuerte política de **Auditoría** gracias a la cual quedan registrados todos los accesos y peticiones realizadas por los usuarios, donde siempre quedan almacenados un conjunto de datos como: usuario, servicio que consume, componente y dirección IP desde el cual accede, fecha, hora, tipo de traza que genera. Además, una descripción que permite aumentar el nivel de detalles acerca de las acciones de los usuarios, lo que contribuye a facilitar el proceso de análisis de las trazas.

## 1.2 Seguridad en aplicaciones web

La Seguridad en Aplicaciones Web, se encuentra relacionada pura y exclusivamente con: la lógica, la escritura de código y el contenido de una aplicación web. Toda aplicación web, requiere de un entorno conformado por elementos externos, tales como sistemas operativos, servidor web y servicios. (5)

La seguridad en la Web está formada por tres etapas fundamentales: (6)

- **Seguridad de la computadora del usuario:** Los usuarios deben contar con navegadores y plataformas seguras, libres de virus y vulnerabilidades. También debe garantizarse la privacidad de los datos del usuario.
- **Seguridad del servidor Web y de los datos almacenados:** Se debe garantizar la operación continua del servidor que los datos no sean modificados sin autorización (integridad) y que la información solo sea distribuida a las personas autorizadas (control de acceso).
- **Seguridad de la información que viaja entre el servidor Web y el usuario:** Garantizar que la información en tránsito no sea leída (confidencialidad), modificada o destruida por terceros. También es importante asegurar que el enlace entre cliente y servidor no pueda interrumpirse fácilmente (disponibilidad).

## 1.3 Ataques sobre aplicaciones Web

Las cinco acciones más usuales para explotar aplicaciones Web son: (7)

1. **Ejecución de código remota:** Como su nombre lo indica, esta vulnerabilidad permite al atacante ejecutar código en el servidor vulnerable y obtener información almacenada en él.
2. **Inyección SQL:** Es una vulnerabilidad de las Web, que afectan directamente a las bases de datos de una aplicación, El problema radica al filtrar erróneamente las variables utilizadas en parte de la página con código SQL, consiste en insertar o inyectar código SQL malicioso dentro de código SQL, para alterar el funcionamiento normal y hacer que se ejecute el código “invasor” dentro del sistema.



3. **Vulnerabilidades asociadas a cadenas:** Estos alteran el flujo de una aplicación utilizando las capacidades proporcionadas por las librerías de formato de cadenas para acceder a otro espacio de memoria.
4. **Cross-Site Scripting:** Es una técnica de ataque que fuerza a un sitio Web a repetir un código ejecutable facilitado por el atacante, y que se cargará en el navegador del usuario.
5. **Enumeración de usuarios:** Tipo de ataque donde los mensajes de validaciones de autenticación le dicen al atacante si el nombre de usuario proporcionado es correcto o no. Explotando esta vulnerabilidad el atacante puede experimentar con diferentes nombres de usuarios y determinar cuales son válidos.

## 1.4 Protocolo LDAP

LDAP (Protocolo compacto de acceso a directorios) es un protocolo estándar que permite administrar directorios, esto es, acceder a bases de información de usuarios de una red mediante protocolos TCP/IP. El objetivo del protocolo LDAP, fue reemplazar al protocolo DAP (utilizado para acceder a los servicios de directorio X.500 por OSI) integrándolo al TCP/IP. Desde 1995, DAP se convirtió en LDAP independiente, con lo cual se dejó de utilizar solo para acceder a los directorios tipo X.500. LDAP es una versión más simple del protocolo DAP, de allí deriva su nombre Protocolo compacto de acceso a directorios. (8)

Un directorio LDAP se destaca entre los demás tipos de bases de datos por las siguientes características:

- Logra mayor rendimiento en la lectura de registros.
- Permite replicar el servidor de forma muy sencilla y económica.
- Se puede integrar fácilmente a diferentes tipos de aplicaciones.
- Dispone de un modelo de nombres globales que asegura que todas las entradas son únicas.
- Usa un sistema jerárquico de almacenamiento de información.
- Permite múltiples directorios independientes.

- Funciona sobre TCP/IP necesario para el acceso a Internet y SSL (Secure Socket Layer).
- La mayoría de servidores LDAP son fáciles de instalar, mantener y optimizar.

Dadas las características de LDAP sus usos más comunes son: (9)

- Servidores de certificados públicos y llaves de seguridad.
- Autenticación única o Single Sign On (SSO) para la personalización de aplicaciones.
- Perfiles de usuarios centralizados.
- Libretas de direcciones compartidas.
- Directorios de información.
- Sistemas de autenticación/autorización centralizada.
- Sistemas de correo electrónico.
- Sistemas de alojamiento de páginas Web y FTP.

### 1.5 Tendencias y tecnologías actuales

En este punto se trata los conceptos fundamentales relacionados con la metodología, tecnologías, y herramientas que se utilizarán para el desarrollo de la aplicación. Para esto se utilizan los patrones de arquitectura y de diseño siguientes: Modelo-Vista-Controlador (MVC), Arquitectura en tres capas, Arquitectura Orientada a Servicios y basada en Componentes y el patrón Proxy.

#### 1.5.1 Sistemas distribuidos. Modelo Cliente – Servidor

Un sistema de cómputo distribuido, es una colección de sistemas de cómputo autónomos. Estos están interconectados a través de una red y de software de comunicaciones, capaces de cooperar para la realización de una tarea común; mientras que la computación distribuida es el conjunto de modelos,

técnicas y herramientas computacionales. Las mismas ayudan a resolver los problemas planteados por los sistemas distribuidos en cuanto a la generación, el procesamiento y la utilización de la información requerida para su funcionamiento. (10)

Entre las principales ventajas de los sistemas distribuidos se encuentran: (11)

- **Rapidez de respuesta y rendimiento:** Los recursos se encuentran compartidos, de manera que una petición determinada pueda ser procesada con mayor velocidad.
- **Fiabilidad:** En el caso de los sistemas no distribuidos, si el servidor principal falla, el sistema colapsa en su totalidad, no ofreciéndose ningún servicio hasta que se repare. Sin embargo si se realiza una distribución del sistema, la salida de servicio de un servidor no implica que se dejen de brindar los servicios que no hayan sido afectados.
- **Escalabilidad:** Este tipo de sistema puede crecer más fácilmente que uno centralizado. Pues si se decide mejorar su productividad es necesario adquirir un nuevo servidor principal, mientras que si desea mejorar uno distribuido basta con incrementar el número de servidores que conforman el sistema.
- **Modularidad y reutilización en el desarrollo:** Esto permite la agilización de los procesos de desarrollo pues se tendrá en cuenta para implementar un nuevo componente las funcionalidades que podrá obtener de otro ya desplegado.

Todos estos elementos, no descartan la existencia de desventajas en los sistemas distribuidos, entre las más significativas están: el uso ineficiente de los recursos distribuidos, la capacidad reducida para administrar apropiadamente grupos de procesadores y memoria localizada en distintos sitios. Además, se complejiza el proceso de administración y mantenimiento y existe una gran dependencia del desempeño de la red y de la confiabilidad de la misma.

El modelo de arquitectura Cliente-Servidor de un sistema distribuido, es el más conocido y difundido en la actualidad. Hay un conjunto de procesos servidores, cada uno actuando como un administrador para una colección de recursos de un tipo; y una colección de procesos clientes, cada uno llevando a cabo una

tarea que requiere acceso a algunos recursos hardware y software compartidos. Es decir, el proceso cliente es el encargado de permitir al usuario formular requerimientos y enviarlos al servidor, quien es el autorizado de atender a múltiples clientes que hacen peticiones de algún recurso administrado por él. El servidor gestiona las funciones relacionadas con las reglas del negocio y los accesos a datos.

## 1.5.2 Patrones de arquitectura y diseño

Un patrón en sentido general es una solución a un problema de diseño. Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser reusable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias. Existen varios tipos de patrones, dependiendo del contexto particular en el cual se aplican o de la etapa en el proceso de desarrollo, algunos de estos tipos son: Patrones de Diseño, de Arquitectura, para Ambientes Distribuidos, entre otros.

### 1.5.2.1 Patrón Proxy

El patrón Proxy, es un patrón de diseño estructural que proporciona un sustituto o representante de otro objeto para controlar el acceso a este. (12)

Existen varios tipos de proxy:

- Proxy remoto: es un representante local para un objeto de otro proceso, posiblemente en otra máquina. También se le denomina embajador. Debe codificar las peticiones y enviárselas al sujeto real remoto.
- Proxy virtual: crea objetos costosos bajo demanda. Puede tener un caché con información sobre el sujeto real para evitar accesos innecesarios.
- Proxy de protección: controla el acceso al objeto original cuando hay distintos niveles de acceso. Comprueba que el cliente tenga permisos de acceso para realizar una petición.

Ventajas y desventajas del patrón Proxy:

- El proxy remoto oculta espacios de direcciones diferentes.
- El proxy virtual evita consumir recursos hasta que no es necesario.
- El proxy de protección y los punteros inteligentes permiten realizar tareas internas (recolección de basura, controlar accesos).

### 1.5.2.2 Arquitectura Orientada a Servicios y Basada en Componentes (SOA-CBA)

La Arquitectura Orientada a Servicios (SOA, Service Oriented Architecture) es una filosofía de diseño que permite un mejor alineamiento de las Tecnologías de Información (IT) con las necesidades de negocio, permitiendo a empleados, clientes y socios comerciales responder de forma más rápida y adaptarse adecuadamente a las presiones del mercado. SOA supone una estrategia general de organización de los elementos de IT, de forma que una colección abigarrada de sistemas distribuidos y aplicaciones complejas se pueda transformar en una red de recursos integrados, simplificada y sumamente flexible.

La Arquitectura SOA establece un marco de diseño para la integración de aplicaciones independientes de manera que desde la red pueda accederse a sus funcionalidades, las cuales se ofrecen como servicios. La forma más habitual de implementarla es mediante Servicios Web. (13)

Entre los beneficios de SOA se destacan: (14)

- La reducción de costos y tiempo en el desarrollo de aplicaciones ya que permite reutilizar los módulos de aplicaciones existentes y el código nuevo para generar nuevas aplicaciones. Como consecuencia también se reducen los costos de mantenimiento.
- Dado que todas las aplicaciones pretenden satisfacer un servicio, se incrementa así la calidad del mismo y la productividad de la empresa.
- Las metodologías que aterrizan el concepto de SOA facilitan la integración entre aplicaciones nuevas así como con los sistemas existentes.

- Desarrollo de aplicaciones más productivas, flexibles, más seguras y manejables para gestionar procesos de negocio críticos a medida que evolucionan o cambian las necesidades del negocio.

### 1.5.2.3 Arquitectura en tres capas

La arquitectura en tres capas tiene como objetivo primordial separar la lógica de negocios de la lógica de diseño. La carga se divide en tres partes: una capa para la presentación (también conocida como interfaz de usuario), otra para el modelado de las reglas del negocio y otra para la persistencia de los datos. La ventaja principal de este estilo es que el desarrollo se puede llevar a cabo en varios niveles y, en caso de que sobrevenga algún cambio, solo se ataca al nivel requerido sin tener que revisar entre código mezclado.

1.- Capa de presentación: es la que ve el usuario, presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de procesos. Esta capa se comunica únicamente con la capa de negocio. También es conocida como interfaz gráfica y debe tener la característica de ser "amigable" (entendible y fácil de usar) para el usuario.

2.- 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) porque 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.

3.- Capa de datos: es donde residen los datos y es la encargada de acceder a los mismos. 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.

### 1.5.2.4 Modelo Vista Controlador

Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. (15)

**Modelo:** Esta es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos.

**Vista:** Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.

**Controlador:** Este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista. En una aplicación web el controlador recibe la petición del usuario, interactúa con el modelo para procesar los datos y hace disponible esos datos a la vista.

### 1.5.2.5 Alta cohesión y bajo acoplamiento

Los patrones de Alta cohesión y bajo acoplamiento pertenecen a la familia GRASP (General Responsibility Assignment Software Patterns, son patrones generales de software para la asignación de responsabilidades). Una clase tiene una alta cohesión cuando el objeto tiene bien delimitadas sus responsabilidades, en la programación orientada a objetos, cada objeto tiene una (o varias, aunque eso sería la mayor parte de las veces un mal diseño) responsabilidad que ha de cumplir dentro de un programa. El patrón de bajo acoplamiento plantea que cada objeto tenga las mínimas dependencias posibles entre las clases, para poder realizar modificaciones en alguna de ellas sin necesidad de hacer muchos cambios en las demás.

## 1.5.3 Lenguajes utilizados para el proceso de desarrollo

### 1.5.3.1 PHP5

PHP (Hypertext Pre-processor) es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es un lenguaje de propósito general ampliamente usado y que está diseñado especialmente para desarrollo web y puede ser embebido dentro de código HTML. Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno.

Algunas ventajas de este lenguaje:

- Es un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos (POO).
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables.
- Tiene manejo de excepciones (desde PHP5).

Este potente lenguaje de programación también posee algunas desventajas entre ellas se pueden citar que todo el trabajo se realiza del lado del servidor, no delegando responsabilidades al cliente, puede que en un determinado momento la capacidad de respuesta sea ineficiente en la medida en que las peticiones al servidor aumenten considerablemente. Por sus características, promueve la creación de código desordenado y complejo de mantener.

PHP 5 fue lanzado 13 de julio de 2004, utiliza el motor Zend Engine 2.0. Actualmente se encuentra en su versión 5 aunque se espera próximamente el lanzamiento de su versión 6, la versión más reciente de PHP es la 5.2.6, que incluye todas las ventajas que provee el nuevo Zend Engine 2 como:

- Mejor soporte para la POO.
- Mejoras de rendimiento.
- Mejor soporte para MySQL con extensión completamente reescrita.
- Mejor soporte a XML (XPath, DOM).



- Soporte integrado para SOAP.
- Iteradores de datos.
- Manejo de excepciones.

### 1.5.3.2 JavaScript

JavaScript es un lenguaje de programación interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C. El lenguaje fue inventado por Brendan Eich. Al igual que Java, JavaScript es un lenguaje orientado a objetos ya que dispone de Herencia. Todos los navegadores modernos interpretan el código JavaScript integrado dentro de las páginas web.

### 1.5.3.3 Lenguajes de marcas

Se crea con la intención de estandarizar el formato de los diversos documentos que se encuentran en las computadoras, debido a que cuando se envía algún texto no se puede ver porque está hecho en un programa diferente al que se tiene instalado en la computadora, es decir, tiene un formato distinto al que el autor quiere darle.

Suele diferenciarse entre tres clases de lenguajes:

- **Marcado de presentación** es aquel que indica el formato del texto. Este tipo de marcado es útil para maquetar la presentación de un documento para su lectura, pero resulta insuficiente para el procesamiento automático de la información. El marcado de presentación resulta más fácil de elaborar, sobre todo para cantidades pequeñas de información. Sin embargo resulta complicado de mantener o modificar, por lo que su uso se ha ido reduciendo en proyectos grandes en favor de otros tipos de marcado más estructurados.
- **Marcado procedimental** está enfocado hacia la presentación del texto, sin embargo, también es visible para el usuario que edita el texto. El programa que representa el documento debe interpretar el código en el mismo orden en que aparece. Por ejemplo, para formatear un título,

debe haber una serie de directivas inmediatamente antes del texto en cuestión, indicándole al software instrucciones tales como centrar, aumentar el tamaño de la fuente, o cambiar a negrita. Inmediatamente después del título deberá haber etiquetas inversas que reviertan estos efectos. En sistemas más avanzados se utilizan macros o pilas que facilitan el trabajo.

- **Marcado descriptivo o semántico** utiliza etiquetas para describir los fragmentos de texto, pero sin especificar cómo deben ser representados o en qué orden, lo cual le da al mismo una gran flexibilidad. El marcado descriptivo también simplifica la tarea de reformatear un texto, debido a que la información del formato está separada del propio contenido.

### 1.5.3.3.1 HTML

HTML, siglas de HyperText Markup Language (Lenguaje de Marcas de Hipertexto), 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. Fue creado en 1991 por Tim Berners-Lee que tomó el SGML y lo combinó con la idea de que en un documento se pudiese crear un "enlace" para acceder a otro (idea que ya existía desde los ochentas y se llamaba "hipertexto"). HTML es hoy día el tipo de documento más empleado en el mundo por su sencillez. (16)

### 1.5.3.3.2 XML

XML, siglas en inglés de Extensible Markup Language (lenguaje de marcas ampliable), es un metalenguaje extensible de etiquetas. Es una simplificación y adaptación del Lenguaje de Marcado Generalizado (Standard Generalized Markup Language, SGML) y permite definir la gramática de lenguajes específicos. Tiene como ventajas: (17)

- Es extensible: Después de diseñado y puesto en producción, es posible extender XML con la adición de nuevas etiquetas, de modo que se pueda continuar utilizando sin complicación alguna.
- El analizador es un componente estándar, no es necesario crear un analizador específico para cada versión de lenguaje XML. Esto posibilita el empleo de cualquiera de los analizadores disponibles. De esta manera se evitan bugs y se acelera el desarrollo de aplicaciones.

- Si un tercero decide usar un documento creado en XML, es sencillo entender su estructura y procesarla. Mejora la compatibilidad entre aplicaciones.

### 1.5.3.3 XHTML

XHTML, Extensible Hypertext Markup Language (lenguaje extensible de marcado de hipertexto) es una familia de tipos de documentos que reproducen, y extienden, el lenguaje HTML. Es una versión más estricta y limpia de HTML, que nace precisamente con el objetivo de reemplazarlo, ante su limitación de uso con las cada vez más abundantes herramientas basadas en XML. Extiende HTML 4.0 que combina la sintaxis de HTML, diseñado para mostrar datos, con la de XML y diseñado para describir los datos. (18)

El lenguaje XHTML presenta un conjunto de ventajas sobre otros formatos, algunas de ellas son: (19)

- Compatibilidad parcial con navegadores antiguos: la información se visualiza, aunque sin formato. XHTML 1.0 fue diseñado expresamente para ser mostrado en navegadores que soportan HTML de base.
- Un mismo documento puede adoptar diseños radicalmente distintos en diferentes dispositivos, pudiendo incluso escogerse entre varios diseños para un mismo medio.
- Facilidad de edición directa del código y de mantenimiento.
- Formato abierto, compatible con los nuevos estándares que actualmente está desarrollando el W3C (World Wide Web Consortium) como recomendación para futuros agentes de usuario o navegadores.
- Los documentos escritos conforme a XHTML 1.0 pueden potencialmente presentar mejor rendimiento en las actuales herramientas Web que aquellos escritos conforme a HTML.

### 1.5.3.4 UML

UML es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Es utilizado además para entender, diseñar, configurar, mantener y

controlar la información sobre los sistemas a construir. Es un lenguaje de propósito general para el modelado orientado a objetos y es también un lenguaje de modelado visual que permite una abstracción del sistema y sus componentes. Ofrece un estándar para describir un modelo del sistema, incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables. En UML 2.0 hay 13 tipos diferentes de diagramas. Para comprenderlos de manera concreta, son categorizados de la siguiente forma: (20)

- Los diagramas de estructura estática:
  - Diagrama de clases.
  - Diagrama de objetos.
  - Diagrama de casos de uso.
- Los diagramas de comportamiento:
  - Diagramas de interacción (secuencia y colaboración).
  - Diagrama de estados.
  - Diagrama de actividad.
- Los diagramas de implementación:
  - Diagrama de componentes.
  - Diagrama de despliegue.

### 1.5.4 Tecnologías utilizadas en el proceso de desarrollo

Para el desarrollo de esta aplicación se utilizan diferentes tecnologías tales como: servicios web XML, Smarty y Ajax.

### 1.5.4.1 Tecnología servicios web XML

Un servicio Web es una colección de protocolos y estándares empleados para intercambiar datos entre aplicaciones y sistemas. Las aplicaciones, escritas en diversos lenguajes de programación y ejecutándose en distintas plataformas pueden utilizar los servicios Web para intercambiar datos sobre una red de ordenadores como Internet de una forma similar a la comunicación entre procesos en un solo ordenador. En los servicios Web, todos los datos se intercambian se formatean con etiquetas XML. (21)

Los Servicios Web usan SOAP (Simple Object Access Protocol) como protocolo para invocar llamadas remotas debido a su simplicidad, se puede identificar un mensaje SOAP como un documento XML conformado por una envoltura obligatoria, un encabezamiento opcional y un cuerpo también obligatorio. Este permite la comunicación entre aplicaciones heterogéneas, de modo que clientes de diferentes plataformas o lenguajes de programación pueden comunicarse entre sí de manera satisfactoria.

### 1.5.4.2 Smarty

Smarty es un motor de plantillas para PHP. Esta herramienta facilita la manera de separar la aplicación lógica y el contenido en la presentación. Smarty tiene como objetivo principal darle solución a los problemas que trae al cambiar dentro del mismo código alguna parte del diseño, esto sucede cuando en la programación utilizando PHP el rol de diseñador gráfico y el programador son cubiertos por una persona. (22)

Algunas de las características de Smarty:

- Es extremadamente rápido.
- Es eficiente ya que puede interpretar el trabajo más desorganizado.
- No analiza gramaticalmente desde arriba el template, solo compila una vez.
- Es posible incrustar directamente código PHP en los archivos de plantilla, aunque esto puede no ser necesario (no recomendado) dado que la herramienta se puede ajustar.

- Soporte de caching incrustado (Uso de la caché).
- Fuentes de Plantilla absoluto.
- Funciones habituales de manipulación de cache.
- Arquitectura de Plugin.

### 1.5.4.3 Ajax

AJAX, acrónimo de Asynchronous JavaScript and XML (JavaScript asíncrono y XML), en realidad no es una tecnología, sino la unión de varias tecnologías, es utilizada para el desarrollo de aplicaciones web. Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones. Los navegadores que permiten Ajax son Mozilla Firefox versión 7.1 y Microsoft Internet Explorer para Windows versión 5.0. (23)

Como Ajax es la unión de diferentes tecnologías incorpora:

- Presentación basada en estándares que usa XHTML y hojas de estilos en cascada (CSS).
- Exhibición e interacción dinámicas usando el Document Object Model (DOM).
- Intercambio y manipulación de datos usando XML y XSLT (Extensible Stylesheet Language Transformations).
- Recuperación de datos asíncrona usando XMLHttpRequest.
- JavaScript poniendo todo junto.

### 1.5.4.4 YUI

Yahoo User Interface (YUI), una serie de bibliotecas escritas en JavaScript, para la construcción de aplicaciones. Dichas bibliotecas son utilizadas para el desarrollo web específicamente para ser usadas como la programación de aplicaciones de escritorio, con componentes vistosos y personalizables y con una amplia implementación con AJAX.

### 1.5.5 Protocolo HTTPS

HTTPS, Hypertext Transfer Protocol Secure (en español: Protocolo seguro de transferencia de hipertexto), es un protocolo de red basado en el protocolo HTTP, destinado a la transferencia segura de datos de hipertexto, es decir, es la versión segura de HTTP.

El sistema HTTPS utiliza un cifrado basado en las Secure Socket Layers (SSL) para crear un canal cifrado cuyo nivel de cifrado depende del servidor remoto y del navegador utilizado por el cliente más apropiado para el tráfico de información sensible que el protocolo HTTP. Es válido destacar que el uso del protocolo HTTPS no impide que se pueda utilizar HTTP. (24)

Los protocolos HTTPS son utilizados por navegadores como:

- Safari.
- Internet Explorer.
- Mozilla Firefox.
- Opera.

### 1.5.6 Sistema Gestor de Base de Datos

Un Sistema Gestor de Bases de Datos (SGBD) es una colección de datos interrelacionados y un conjunto de programas para acceder a esos datos. Tiene como objetivo primordial proporcionar un entorno que sea a la vez conveniente y eficiente para ser utilizado al extraer y almacenar información de la base de datos.

SGBD permite a los usuarios definir, crear y mantener la base de datos, y proporciona acceso controlado a la misma.

Existen diversos tipos de SGBD algunos ejemplos son: Oracle, Microsoft SQL Server, Postgres SQL, MySQL pero en sentido general todos convergen en un conjunto de objetivos que les permitirá manejar de una forma clara, sencilla y ordenada una colección de datos que posteriormente se convertirá en información, los cuales se comentan a continuación: (25)

- **Abstracción de la información.** Los SGBD ahorran a los usuarios detalles acerca del almacenamiento físico de los datos. El hecho de que una base de datos ocupe uno o cientos de archivos se hace transparente al usuario.
- **Redundancia mínima.** Un buen diseño de una base de datos logrará evitar la aparición de información repetida o redundante.
- **Consistencia.** En aquellos casos en los que no se ha logrado una redundancia nula, será necesario vigilar que aquella información que aparece repetida se actualice de forma coherente, es decir, que todos los datos repetidos se actualicen de forma simultánea.
- **Seguridad.** La información almacenada en una base de datos puede llegar a tener un gran valor. Los SGBD deben garantizar que esta información se encuentra segura frente a usuarios malintencionados, que intenten leer información privilegiada; frente a ataques que deseen manipular o destruir la información; o simplemente ante las torpezas de algún usuario autorizado pero despistado.
- **Integridad.** Se trata de adoptar las medidas necesarias para garantizar la validez de los datos almacenados. Es decir, se trata de proteger los datos ante fallos de hardware, datos introducidos por usuarios descuidados, o cualquier otra circunstancia capaz de corromper la información almacenada.



- **Respaldo y recuperación.** Los SGBD deben proporcionar una forma eficiente de realizar copias de respaldo de la información almacenada en ellos, y de restaurar a partir de estas copias los datos que se hayan podido perder.
- **Control de la concurrencia.** En la mayoría de entornos es habitual que sean muchas las personas que acceden a una base de datos de forma simultánea, tanto para recuperar o almacenar información. El SGBD debe ser capaz de controlar estos accesos concurrentes a la información, que podría derivar en inconsistencias.

### 1.5.6.1 PostgreSQL

PostgreSQL (llamado también Postgres95) es un SGBD derivado del paquete Postgres escrito en Berkeley. Es el gestor de bases de datos de código abierto más avanzado hoy en día, ofreciendo control de concurrencia multi-versión, soporta casi toda la sintaxis SQL (incluyendo sub consultas, transacciones, y tipos y funciones definidas por el usuario), cuenta también con un amplio conjunto de enlaces con lenguajes de programación como por ejemplo C, C++, Java, Python, PHP y muchos más. Es un sistema objeto-relacional, ya que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. Presenta un conjunto de características tales como: (26)

- Incorpora una estructura de datos array.
- Incorpora 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.

- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.

### 1.5.7 Servidor Web Apache

Es el servidor HTTP más usado, su desarrollo está basado en el de código abierto para plataformas Unix (GNU/Linux), Windows, Macintosh y otras e implementa el protocolo HTTP/1.1. Presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido.

Algunas razones por las que el servidor es grandemente reconocido en el ámbito tecnológico son: (27)

- Corre en una multitud de Sistemas Operativos, lo que lo hace prácticamente universal.
- Apache es una tecnología gratuita de código fuente abierta.
- Apache es un servidor altamente configurable de diseño modular. Es muy sencillo ampliar las capacidades del servidor Web Apache.
- Apache trabaja con gran cantidad de Perl, PHP y otros lenguajes de script.
- Apache te permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurar Apache para que ejecute un determinado script cuando ocurra un error en concreto.
- Tiene una alta configurabilidad en la creación y gestión de logs. Apache permite la creación de ficheros de log a medida del administrador, de este modo puedes tener un mayor control sobre lo que sucede en tu servidor.

### 1.5.8 Metodología Proceso Unificado de Desarrollo de Software (RUP)

El Proceso Unificado de Rational (Rational Unified Process en inglés, RUP) es un proceso de desarrollo de software con un conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software, es orientado a objetos y utiliza UML como lenguaje. RUP agrupa las actividades

definiendo ciclos de desarrollo y flujos de trabajo, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

El ciclo de vida de RUP se caracteriza por ser:

- **Dirigido por casos de uso:** Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso (cómo se llevan a cabo).
- **Centrado en la arquitectura:** La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente.
- **Iterativo e Incremental:** Es práctico dividir el trabajo en partes más pequeñas o mini proyectos. Cada mini proyecto es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en los flujos de trabajo, y los incrementos, al crecimiento del producto. Cada iteración se realiza de forma planificada es por eso que se dice que son mini proyectos.

## 1.5.9 Herramientas a utilizar

Con las descripciones dadas anteriormente, se pueden definir las herramientas que se utilizan para el desarrollo de este sistema. Se utilizará el Enterprise Architect 7.0 como herramienta CASE (Computer-Aided Software Engineering) basada en la última especificación UML 2.1. Para la creación de las páginas Web, Dreamweaver 8.0 debido a que soporta el lenguaje de marcas XHTML 1.0, para la edición de código PHP, se utiliza el Zend Studio 6.1 y para administrar la base de datos el EMS SQL Manager 2007 para PostgreSQL, por ser un ambiente gratuito de sencilla utilización que posee una interfaz muy intuitiva.

### 1.5.9.1 Enterprise Architect

Combina el poder de la última especificación UML 2.1 con alto rendimiento, interfaz intuitiva, para traer modelado avanzado al escritorio, y para el equipo completo de desarrollo e implementación. Presenta un gran conjunto de características como: (28)

- **Alta capacidad:** es una herramienta comprensible de diseño y análisis UML, cubriendo el desarrollo de software desde el paso de los requerimientos a través de las etapas del análisis, modelos de diseño, pruebas y mantenimiento. EA es una herramienta multi-usuario, basada en Windows, diseñada para ayudar a construir software robusto y fácil de mantener. Ofrece salida de documentación flexible y de alta calidad.
- **Velocidad, estabilidad y buen rendimiento:** el Lenguaje Unificado de Modelado provee beneficios significativos para ayudar a construir modelos de sistemas de software rigurosos y donde es posible mantener la trazabilidad de manera consistente. Enterprise Architect soporta este proceso en un ambiente fácil de usar, rápido y flexible.
- **Trazabilidad de extremo a extremo:** EA provee trazabilidad completa desde el análisis de requerimientos hasta los artefactos de análisis y diseño, a través de la implementación y el despliegue.

### 1.5.9.2 Dreamweaver

Dreamweaver es la herramienta de diseño de páginas web más avanzada. Cumple perfectamente el objetivo de diseñar páginas con aspecto profesional, y soporta gran cantidad de tecnologías, además muy fáciles de usar:

- Hojas de estilo y capas.
- JavaScript para crear efectos e interactividades.
- Inserción de archivos multimedia.

### 1.5.9.3 Zend Studio

Es un programa para desarrollar aplicaciones web en el lenguaje PHP. Sirve como editor de texto para páginas PHP y proporciona una serie de ayudas que pasan desde la creación y gestión de proyectos hasta la depuración de código. El programa entero está escrito en Java, esto ha permitido a Zend lanzar con relativa facilidad y rapidez versiones del producto para Windows, Linux y MacOS.

### 1.5.9.4 Jmeter

JMeter es una herramienta de carga para llevar acabo simulaciones sobre cualquier recurso de Software. Inicialmente diseñada para pruebas de estrés en aplicaciones web, hoy en día, su arquitectura ha evolucionado no solo para llevar acabo pruebas en componentes habilitados en Internet (HTTP), sino además en Bases de Datos, programas en Perl, requisiciones FTP y prácticamente cualquier otro medio. Además, posee la capacidad de realizar desde una solicitud sencilla hasta secuencias de peticiones que permiten diagnosticar el comportamiento de una aplicación en condiciones de producción. Simula todas las funcionalidades de un Navegador ("Browser"), o de cualquier otro cliente, siendo capaz de manipular resultados en una determinada petición y reutilizarlos para ser empleados en una nueva secuencia. (29)

### 1.5.9.5 EMS SQL Manager 2007 para PostgreSQL

EMS SQL Manager para PostgreSQL es una herramienta gráfica sencilla, liviana y gratuita para la administración de PostgreSQL. Funciona con cualquier versión de PostgreSQL, hasta la 8.1, y soporta todas sus nuevas características. Cuenta con una nueva y avanzada interfaz gráfica de usuario con un sistema asistente bastante descriptivo. (30)

Características principales:

- Soporte completo para las versiones de PostgreSQL hasta la 8.1.
- Poderosa y sencilla interfaz gráfica de usuario.
- Ágil navegación y administración de la base de datos.

- Administración sencilla de todos los objetos PostgreSQL.
- Herramientas avanzadas de manipulación de datos.
- Administración efectiva de seguridad.
- Excelentes herramientas para la elaboración de consultas.

En este capítulo, se profundizó en los conceptos y definiciones necesarias para comprender el proceso de desarrollo del sistema. Se hizo un análisis de las tecnologías, lenguajes y herramientas para el mejoramiento del mismo. Lo que permitirá que cumplan con las políticas definidas por el Área Temática SAS y la Facultad.

### CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

En este capítulo se definen todos los requerimientos funcionales y no funcionales. Se realiza además el modelo de dominio debido a que los procesos no están visibles y no se puede describir el negocio por casos de uso, el cual incluye una serie de conceptos que se relacionan entre sí para un mejor entendimiento del funcionamiento del sistema.

#### 2.1 Modelo de Dominio

El Modelo de Dominio o Modelo Conceptual es una representación visual de los conceptos u objetos del mundo real significativos para el problema o área de interés. Representa clases conceptuales del dominio del problema. Este modelo se encarga de capturar los tipos más importantes de objetos y eventos que suceden en el entorno. Se realiza el Modelo de Dominio debido a que no pueden ser identificados los procesos de negocio asociados al contexto del sistema, como consecuencia de la poca estructuración y fronteras de los mismos.

##### 2.2.1 Conceptos fundamentales

Para un mejor entendimiento del Diagrama del Modelo de Dominio se tiene en cuenta un marco conceptual, donde se identifica las definiciones necesarias para el desarrollo de este sistema, basándose también en la versión anterior del SAAA-SAS. (31)

**Componente:** Es una unidad ejecutable que representa el núcleo de la aplicación, la cual puede ser implantada independientemente y ser a la vez sujeto de composición de terceras partes, es decir, se puede tomar el componente y agregarlo a otro componente en desarrollo o simplemente consumir algunos de los servicios que brinda.

**Certificado:** Conjunto de datos y privilegios de un usuario determinado que se crea de forma automática durante los procesos de autenticación y autorización. El mismo posee un identificador único de treinta y dos caracteres que se genera de manera aleatoria, contiene el nivel de acceso del usuario, el identificador

del nivel de acceso y un listado de los componentes a los que tiene derecho de acceso, así como los privilegios de ejecución correspondientes en estos componentes.

**Nivel de Actividad:** Clasificación del usuario que permite diferenciar entre un usuario activo (que está autorizado a utilizar el sistema dentro de un período de tiempo determinado) y uno de tipo inactivo (contrarresta la definición anterior).

**Traza:** Historial donde se almacenan todos los eventos realizados por el usuario al interactuar con la información perteneciente al sistema.

**Rol:** Papel que cumple un usuario dentro de un componente y que limita el conjunto de funcionalidades que puede desempeñar en ese ámbito dentro del sistema.

**Servicios:** Operación proporcionada por un componente determinado.

**Organismo:** Conjunto de dependencias, oficinas o empleos que cumplen con determinadas leyes, usos y costumbres y que a la vez forman una institución social.

**Nivel:** Concepto asociado a las diferentes instancias de dirección administrativa de un organismo determinado.

**Ubicación:** Recoge la ubicación exacta del usuario en cada uno de los niveles en que se desempeña.

**Grupo de Nivel:** Especifica los diferentes niveles que tiene cada uno de los organismos.

**Grupo de Rol:** Se recogen todos los roles que pertenecen a cada organismo.

**Período de Actividad:** Es el tiempo durante el cual el usuario va a estar en estado activo dentro de la aplicación.

**Historial de Traza:** Se especifica los tipos de trazas definidos que pueden dejar los usuarios.



### 2.2.2 Diagrama del Modelo de Dominio

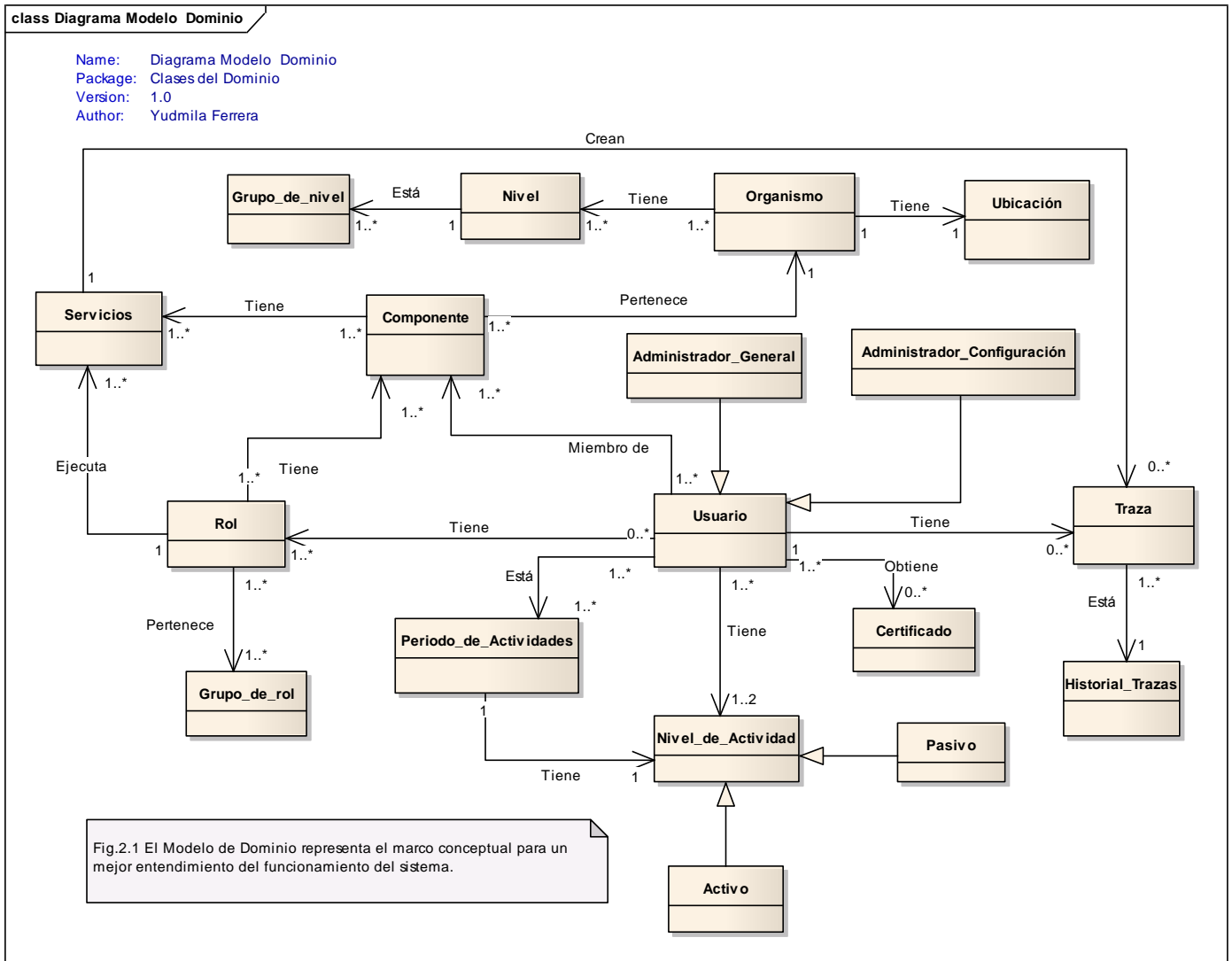


Fig. 2.1 Diagrama del Modelo de Dominio

### 2.3 Propuesta de Sistema

Se propone realizar una nueva versión del SAAA-SAS que tenga implementado, un servicio que utilice el patrón proxy, que sirva de interfaz a las peticiones de los clientes para consumir los servicios web, un mecanismo que permita controlar la acumulación de la información almacenada en las tablas de la base

de datos relacionadas con la auditoría, es decir, una política que permita el balanceo de las trazas de forma automática y brindarle la posibilidad al usuario de configurar los reportes, seleccionando los elementos que quiera mostrar a la hora de generar los mismos. Además el sistema debe contar con un módulo de instalación personalizado, que brinde comodidad y facilidad al usuario para instalar la aplicación.

### 2.3.1 Especificación de los Requerimientos de Software

Especificación de Requerimientos de Software (ERS): define de forma precisa el producto de software que se va a construir. Tiene como objetivo documentar todos los requerimientos del sistema, este describe las funciones del sistema, los requerimientos no funcionales, características del diseño, y otros elementos necesarios para proporcionar una descripción completa y comprensiva de los requerimientos para el software a desarrollar, es decir, define los requerimientos de software como condiciones o capacidades que deben estar presentes en un sistema o componentes de este, para satisfacer un contrato, estándar, especificación u otro documento formal. (32)

#### 2.3.1.1 Requerimientos Funcionales

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir. Definen 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. Para el desarrollo de la nueva versión del SAAA-SAS se tiene en cuenta los siguientes requerimientos funcionales que el sistema debe cumplir:

LISTADO DE REQUERIMIENTOS FUNCIONALES	
<b>RF1-</b> Adicionar Servicio Web	<b>RF2-</b> Modificar Servicio Web
<b>RF3-</b> Eliminar Servicio Web	<b>RF4-</b> Buscar Servicios Web
<b>RF5-</b> Verificar Componente	<b>RF6-</b> Verificar Servicio Web
<b>RF7-</b> Verificar Petición	<b>RF8-</b> Autorizar
<b>RF9-</b> Verificar Disponibilidad Servicio Web	<b>RF10-</b> Buscar Direcciones Opcionales

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

<b>RF11-</b> Consumir Servicio Web	<b>RF12-</b> Realizar Petición
<b>RF13-</b> Registrar Traza	<b>RF14-</b> Balancear Cantidad de Trazas
<b>RF15-</b> Verificar Cantidad de Trazas	<b>RF16-</b> Buscar Trazas
<b>RF17-</b> Generar Reporte	<b>RF18-</b> Adicionar Configuración
<b>RF19-</b> Eliminar Configuración	<b>RF20-</b> Graficar

**Tabla 2.1** Listado de Requerimientos Funcionales.

En la siguiente tabla se muestran los requerimientos funcionales agrupados por casos de uso, que serán las funcionalidades que el sistema va a ofrecer, los casos de uso son:

DEFINICIÓN DE CASOS DE USO	
<b>CUS1: Adicionar_Servicio_Web</b> RF1- Adicionar Servicio Web	<b>CUS2: Modificar_Servicio_Web</b> RF2- Modificar Servicio Web
<b>CUS3: Eliminar_Servicio_Web</b> RF3- Eliminar Servicio Web	<b>CUS4: Listar_Servicio_Web</b> RF4- Listar Servicios Web
<b>CUS5: Verificar_Componente</b> RF5- Verificar Componente	<b>CUS6: Verificar_Servicio_Web</b> RF6- Verificar Servicio Web
<b>CUS7: Verificar_Peticion</b> RF7- Verificar Petición	<b>CUS8: Autorizar</b> RF8- Autorizar
<b>CUS9: Verificar Disponibilidad Servicio Web</b> RF9- Verificar Disponibilidad Servicio Web	<b>CUS10: Buscar_Direcciones_Opcionales</b> RF10- Buscar Direcciones Opcionales
<b>CUS11: Consumir_Servicio_Web</b> RF11- Consumir Servicio Web	<b>CUS12: Realizar_Peticion</b> RF12- Realizar Petición
<b>CUS13: Registrar_Trazas</b> RF13- Registrar Traza	<b>CUS14: Verificar_Cantidad_Trazas</b> RF14- Verificar Cantidad de Trazas

<b>CUS15: Balancear_Trazas</b> RF15- Balancear Cantidad Trazas	<b>CUS16: Buscar_Trazas</b> RF16- Buscar Trazas
<b>CUS17: Generar_Reporte</b> RF17- Generar Reporte	<b>CUS18: Adicionar_Configuracion</b> RF18- Adicionar Configuración
<b>CUS19: Eliminar_Configuracion</b> RF19- Eliminar Configuración	<b>CUS20: Graficar</b> RF20- Graficar

**Tabla 2.2** Definición de Casos de Uso del Sistema.

### 2.3.1.2 Requerimientos No Funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener, como restricciones de ambiente y desarrollo, performance, dependencias de plataformas y confiabilidad. Los requerimientos no funcionales se clasifican de la siguiente manera: (33)

#### 2.3.1.2.1 Usabilidad

**RNF1:** El sistema debe garantizar un acceso fácil y rápido. Podrá ser usado por cualquier usuario que posea pocos conocimientos informáticos y de un ambiente Web en sentido general.

**RNF2:** El sistema debe permitir una fácil configuración para su posterior uso. Teniendo en cuenta un módulo de instalación que sea fácil de manejar por los usuarios.

#### 2.3.1.2.2 Rendimiento

**RNF3:** El sistema debe tener una similitud entre sus páginas y estar poco cargado con imágenes, lo que posibilita que se devuelvan respuestas de una manera eficiente en un tiempo mínimo, siendo más sencillo de entender y utilizar por el usuario.

### 2.3.1.2.3 Soporte

**RNF4:** Una vez terminado el desarrollo del sistema se llevarán a cabo los procesos de despliegue, capacitación y mantenimiento de software con motivo de asistir a los clientes del producto, así como lograr su mejoramiento progresivo, evolución y sostenibilidad en el tiempo. Además se contará con un grupo de trabajo designado para el mantenimiento y actualización continuo del sistema.

### 2.3.1.2.4 Portabilidad

**RNF5:** Permitir que el sistema se ejecute sobre el Sistema Operativo Linux, Windows 98 o superior.

### 2.3.1.2.5 Seguridad

**RNF5:** Disponer de un mecanismo de seguridad basado en el modelo de Autenticación, Autorización y Auditoría (AAA).

- **Confiabilidad:** La información manejada por el sistema está protegida contra acceso no autorizado.
- **Integridad:** Que la información sea modificada incluyendo su creación y eliminación solo por personal autorizado. Se implementarán políticas de resguardo de información, así como la realización de copias periódicas de seguridad. Se permitirá la creación de copias de respaldo que puedan restaurar el sistema en caso de fallo crítico o pérdida total de la información.
- **Disponibilidad:** Los usuarios autorizados tendrán acceso a la información en todo momento.

**RNF5.1:** La autenticación será la primera acción del usuario en el sistema y consiste en suministrar un nombre único de usuario y una contraseña que debe ser de conocimiento exclusivo de la persona que se está involucrada en esta operación.

**RNF5.2:** Cada petición de usuario autorizada o no será registrada, almacenándose día, mes, año, hora, minuto y segundo en que fue solicitada dicha petición.

**RNF5.3:** El sistema controlará la cantidad de caracteres mínimo y máximo para el nombre de usuario y contraseña, el registro de las cuentas creadas por el administrador y no permitirá conexiones concurrentes con un mismo conjunto de credenciales.

**RNF5.4:** Se utilizará el protocolo HTTPS para el transporte de información a través de la Web.

**RNF5.5:** Aplicar políticas de seguridad y acceso a los servidores donde sea desplegado el SAAA-SAS.

### **2.3.1.2.6 Apariencia o Interfaz Externa**

**RNF6:** La interfaz debe ser sencilla y amigable ya que el usuario no es experto en el uso de las aplicaciones Web. Diseño sencillo, con pocas entradas, permitiendo que no sea necesario mucho entrenamiento para utilizar el sistema, además que exista paginación de reportes de búsqueda y listados.

**RNF6.1:** El sistema debe presentar un diseño que permita una clara distinción entre los elementos de las ventanas, formularios a través del uso de colores, tamaño de las fuentes, iconografía, así como otros recursos.

### **2.3.1.2.7 Ayuda y documentación en línea**

**RNF7:** Se contará con una Ayuda en línea así como un Manual de Usuario en formato PDF que indicará al usuario cómo interactuar con las funcionalidades del sistema.

### **2.3.1.2.8 Software**

**RNF8.1:** Para clientes: Se tendrá acceso al sistema a través del navegador Web. Firefox Mozilla 1.5 o superior que soporte DHTML y CSS2.

**RNF8.2:** Para Capa de Presentación: El servidor debe tener PHP Versión 5.2.5 o superior. Servidor HTTP preferiblemente Apache. Sistema Operativo GNU/LINUX, distribución Debian 4 Etch.

**RNF8.3:** Para Capa de Datos: Servidor de Base de Datos PostgreSQL Versión 8.3. Sistema Operativo GNU/LINUX, distribución Debian 4 Etch.

### 2.3.1.2.9 Hardware

**RNF9.1:** Requerimientos mínimos para clientes:

- Ordenador Pentium o superior.
- 64 MB de Memoria RAM.
- Monitor VGA o superior.
- Teclado y Mouse.
- Procesador 486DX / 66 MHZ o superior.
- Disco duro de 20 GB.
- Impresora de puntos.
- Insumos. (Disquetes, CD- RW, Papel continuo y cintas de impresora).
- La PC de trabajo debe estar conectada a una Red de Área Local (LAN).

### 2.3.2 Modelo de Casos de Uso del Sistema

Este artefacto se basa en la descripción de elementos o usuarios externos al sistema (actores) y de la funcionalidad del sistema (casos de uso). Un Modelo de Casos de Uso describe los requerimientos funcionales de un actor en términos de las interacciones que este ejecuta con el sistema. 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. Presenta el sistema desde la perspectiva de su uso y esquematiza como proporcionará valor a sus usuarios. (34)

#### 2.3.2.1 Definición de los actores

Los actores del sistema pueden representar el rol que juega una o varias personas, un equipo o un sistema automatizado. Pueden intercambiar información con él o ser recipientes pasivos de información. Los actores que va a tener el sistema son los siguientes: (35)

ACTORES	JUSTIFICACIÓN
Usuario	Actor que interactúa directamente con la aplicación una vez que haya superado sus fases de desarrollo correspondientes, con el objetivo de consultar, modificar o eliminar la información gestionada por el mismo.
Administrador_General	Actor que en dependencia de su nivel de acceso sobre un organismo cuyos requerimientos de seguridad son gestionados por el SAAA-SAS, tiene los permisos necesarios para gestionar la información de los usuarios que en la estructura jerárquica de niveles defina por su organismo, posean un nivel igual o inferior al suyo. Esta gestión incluye crear usuarios, modificar sus datos y privilegios, eliminarlos y realizar búsquedas a partir de diferentes parámetros. Por otra parte puede realizar una auditoría estricta a través de las trazas almacenadas por el sistema, conociendo qué usuario ha participado en cada transacción.
Administrador_Configuración	Actor que tiene permiso total sobre la configuración del sistema. Único encargado de la gestión de organismos, niveles, ubicaciones, componentes, servicios y roles.
Proxy	Actor que representa un componente, el cual tiene la responsabilidad de autorizar y consumir un servicio web.
Componente	Actor que representa un sistema externo, una aplicación que va a utilizar los requerimientos de seguridad que brinda el SAAA-SAS. Hace una petición al Proxy, este consume un servicio web y le devuelve el resultado.
Sistema_Servicio_Web	Actor que representa un sistema externo, una aplicación que va a brindar los servicios web siendo el Proxy el que los consume.

**Tabla 2.3** Justificación de Actores del Sistema.



### 2.3.2.1.1 Vista global de los actores del sistema

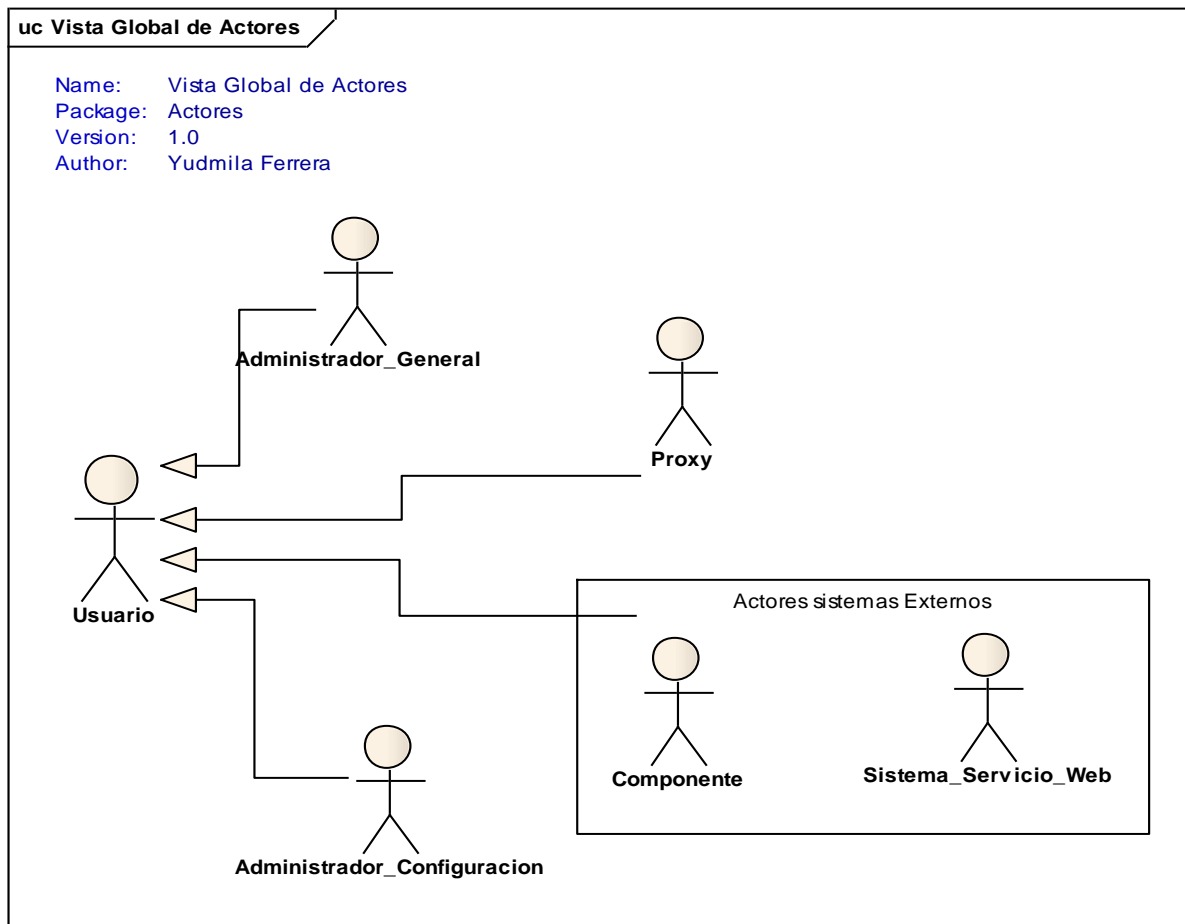


Fig. 2.2 Vista Global Actores del Sistema.

### 2.3.2.2 Diagrama de Casos de Uso

Un diagrama de casos de uso del sistema representa gráficamente a los procesos y su interacción con los actores.

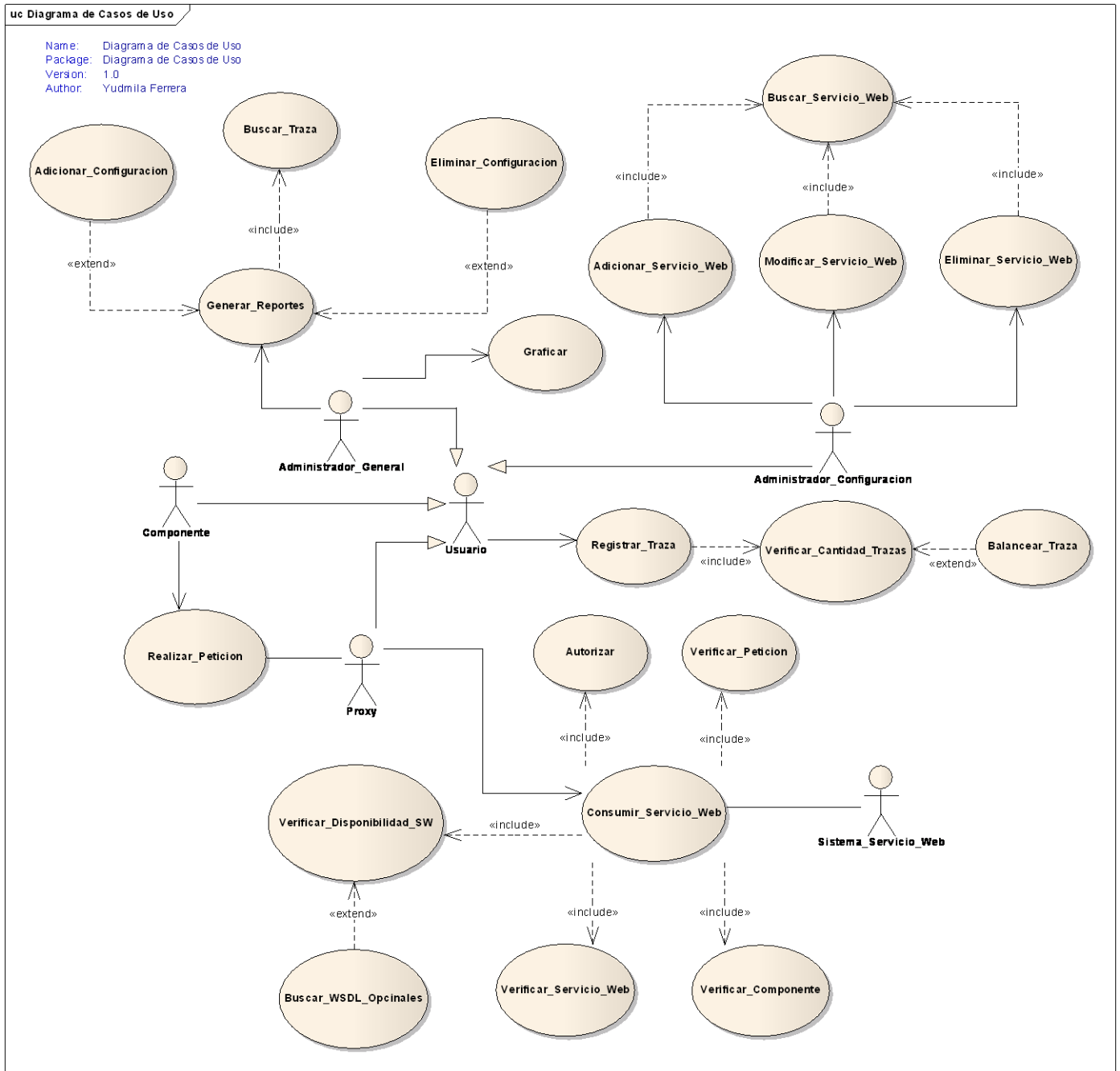


Fig. 2.3 Diagrama de Casos de Uso.

### 2.3.2.3 Descripción Textual de los Casos de Uso

Para dar cumplimiento a los requerimientos funcionales del sistema se describen los casos de uso identificados de la siguiente forma:

<b>Caso de Uso:</b>	<b>CUS1: Adicionar_Servicio_Web</b>
<b>Actores:</b>	Administrador Configuración (Inicia)
<b>Resumen:</b>	El caso de uso inicia cuando el Administrador_Configuración accede a la opción Adicionar, el sistema brinda la posibilidad de introducir datos como el organismo, el componente y la dirección donde se encuentra ubicado dicho servicio.
<b>Precondiciones:</b>	El usuario debe haber sido autenticado en el sistema.
<b>Referencias</b>	RF1, RF4, Buscar_Servicio_Web (Caso de Uso incluido)
<b>Prioridad</b>	Alta
<b>Poscondiciones</b>	Se adiciona un nuevo servicio web por el Administrador de Configuración.

**Tabla 2.4** Descripción textual CUS1\_Adicionar\_Servicios\_Web.

<b>Caso de Uso:</b>	<b>CUS2: Modificar_Servicio_Web</b>
<b>Actores:</b>	Administrador Configuración (Inicia)
<b>Resumen:</b>	El caso de uso inicia cuando el Administrador de Configuración busca un servicio web de un determinado componente, lo selecciona y accede a la opción Editar, el sistema brinda la posibilidad de modificar datos como el organismo, el componente y la dirección donde se encuentra ubicado dicho servicio.

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

---

<b>Precondiciones:</b>	El usuario debe haber sido autenticado en el sistema.
<b>Referencias</b>	RF2, RF4, Buscar_Servicio_Web (Caso de Uso incluido)
<b>Prioridad</b>	Alta
<b>Poscondiciones</b>	Se modifica un servicio web por el Administrador de configuración.

**Tabla 2.5** Descripción textual CUS2\_Modificar\_Servicios\_Web.

<b>Caso de Uso:</b>	<b>CUS3: Eliminar_Servicio_Web</b>
<b>Actores:</b>	Administrador Configuración (Inicia)
<b>Resumen:</b>	El caso de uso inicia cuando el Administrador _Configuración busca un servicio web de un determinado componente, lo selecciona y accede a la opción Eliminar, el sistema brinda la posibilidad de eliminar el servicio web
<b>Precondiciones:</b>	El usuario debe haber sido autenticado en el sistema.
<b>Referencias</b>	RF3, RF4, Buscar_Servicio_Web (Caso de Uso incluido)
<b>Prioridad</b>	Alta
<b>Poscondiciones</b>	Se elimina un servicio web por el Administrador de configuración.

**Tabla 2.6** Descripción textual CUS3\_Eliminar\_Servicio\_Web.

<b>Caso de Uso:</b>	<b>CUS4: Buscar_Servicio_Web</b>
<b>Actores:</b>	Administrador Configuración (Inicia)

<b>Resumen:</b>	El caso de uso inicia cuando el administrador de configuración realiza una búsqueda por diferentes criterios, el sistema muestra un listado con los servicios web que cumplan con los criterios especificados.
<b>Precondiciones:</b>	El usuario debe haber sido autenticado en el sistema.
<b>Referencias</b>	RF4
<b>Prioridad</b>	Alta
<b>Poscondiciones</b>	Se muestra un listado de los servicios web encontrados para un determinado componente.

**Tabla 2.7** Descripción textual CUS4\_Buscar\_Servicio\_Web.

<b>Caso de Uso:</b>	<b>CUS5: Verificar_Componente</b>
<b>Actores:</b>	Proxy(Inicia)
<b>Resumen:</b>	El caso de uso inicia cuando hay una petición de un servicio por parte de un Sistema_Externo, para ello el Proxy verifica que ese componente esté registrado en el sistema.
<b>Precondiciones:</b>	Se debe haber realizado la petición de un servicio web, a partir de ahí el sistema verifica que el componente esté registrado en el sistema.
<b>Referencias</b>	RF5, RF11, RF12.
<b>Prioridad</b>	Crítica
<b>Poscondiciones</b>	Se verifica la existencia del componente.

**Tabla 2.8** Descripción textual CUS5: Verificar\_Componente.

<b>Caso de Uso:</b>	<b>CUS6: Verificar_Servicio_Web</b>
<b>Actores:</b>	Proxy(Inicia)
<b>Resumen:</b>	El caso de uso inicia cuando hay una petición de servicio, donde el Proxy verifica que el servicio web solicitado este asignado al componente.
<b>Precondiciones:</b>	Se debe haber solicitado un servicio web y el componente debe estar registrado en el sistema.
<b>Referencias</b>	RF5, RF6, RF11, RF12, Verificar_Componente.
<b>Prioridad</b>	Crítica
<b>Poscondiciones</b>	Se verifica que el servicio web solicitado está asignado al componente.

**Tabla 2.9** Descripción textual CUS6: Verificar\_Servicio\_Web.

<b>Caso de Uso:</b>	<b>CUS7: Verificar_Peticion</b>
<b>Actores:</b>	Proxy(Inicia)
<b>Resumen:</b>	El caso de uso inicia cuando hay una petición de servicio, donde el Proxy verifica la cantidad de peticiones que ha realizado el usuario en un determinado tiempo.
<b>Precondiciones:</b>	Se debe haber solicitado un servicio web y este debe estar asignado a un componente.
<b>Referencias</b>	RF6, RF7, RF11, RF12, Verificar_Servicio_Web.
<b>Prioridad</b>	Crítica

<b>Poscondiciones</b>	Queda verificado que los usuarios no hayan excedido el número máximo de peticiones.
-----------------------	---

**Tabla 2.10** Descripción textual CUS7: Verificar\_Peticion.

<b>Caso de Uso:</b>	<b>CUS8: Autorizar</b>
<b>Actores:</b>	Proxy(Inicia)
<b>Resumen:</b>	El caso de uso inicia cuando hay una petición de servicio, donde el Proxy verifica que el usuario que realiza la solicitud tenga permiso para consumir el mismo.
<b>Precondiciones:</b>	Se debe haber solicitado un servicio web y el usuario no debe haber excedido el número máximo de peticiones.
<b>Referencias</b>	RF7, RF8, RF11, RF12, Verificar_Peticion.
<b>Prioridad</b>	Crítica
<b>Poscondiciones</b>	Se verifica que el usuario tenga permiso para acceder al servicio solicitado.

**Tabla 2.11** Descripción textual CUS8: Autorizar.

<b>Caso de Uso:</b>	<b>CUS9: Verificar_Disponibilidad_Servicio_Web</b>
<b>Actores:</b>	Proxy(Inicia)
<b>Resumen:</b>	El caso de uso inicia cuando hay una petición de un servicio, donde el Proxy verifica que el servicio solicitado esté disponible, en caso de no estarlo se verifica su disponibilidad en dos direcciones alternativas que son proporcionadas

	de forma opcional por el administrador de configuración en el momento de registrar un componente.
<b>Precondiciones:</b>	Se debe haber solicitado un servicio web y el usuario debe tener permiso para consumir el servicio web.
<b>Referencias</b>	RF8, RF9, RF10 RF11, RF12, Autorizar, Buscar_WSDL_Opcionales (Caso de Uso extendido).
<b>Prioridad</b>	Crítica
<b>Poscondiciones</b>	Retorna si el servicio web está disponible o no.

**Tabla 2.12** Descripción textual CUS9: Verificar\_Disponibilidad\_Servicio\_Web.

<b>Caso de Uso:</b>	<b>CUS10: Buscar_Direcciones_Opcionales</b>
<b>Actores:</b>	Proxy(Inicia)
<b>Resumen:</b>	El Proxy, en caso de que el servicio web solicitado no esté disponible en la primera dirección verifica su disponibilidad en dos direcciones alternativas que son proporcionadas de forma opcional por el administrador de configuración
<b>Precondiciones:</b>	El servicio web solicitado no debe estar disponible en la primera dirección.
<b>Referencias</b>	RF9, RF10, RF11, Verificar_Disponibilidad_Servicio_Web
<b>Prioridad</b>	Crítica
<b>Poscondiciones</b>	Se buscan las direcciones opcionales del servicio web solicitado.

**Tabla 2.13** Descripción textual CUS10\_Buscar\_Direcciones\_Opcionales.



<b>Caso de Uso:</b>	<b>CUS11: Consumir_Servicio_Web</b>
<b>Actores:</b>	Proxy(Inicia), Sistema_Servicio_Web
<b>Resumen:</b>	El caso de uso inicia cuando hay una petición de un servicio por parte de un Componente, para poder consumir el mismo el Proxy realiza una serie de verificaciones como: verificación del componente, la asignación del servicio web al componente, la cantidad de peticiones, el permiso del usuario y la disponibilidad del servicio web.
<b>Precondiciones:</b>	El Sistema_Externo debe haber realizado una petición y tener los permisos para que el Proxy consuma el servicio web.
<b>Referencias</b>	RF5, RF6, RF7, RF8, RF9, RF10, RF11, RF12, Verificar_Componente (Caso de Uso incluido), Verificar_Servicio_Web (Caso de Uso incluido), Verificar_Peticiones (Caso de Uso incluido), Autorizar (Caso de Uso incluido), Verificar_Disponibilidad_Servicio_Web (Caso de Uso incluido).
<b>Prioridad</b>	Crítica
<b>Poscondiciones</b>	El Proxy consume el servicio web solicitado por el componente

**Tabla 2.14** Descripción textual CUS11\_Consumir\_Servicios\_Web.

<b>Caso de Uso:</b>	<b>CUS12: Realizar_Peticion</b>
<b>Actores:</b>	Componente(Inicia), Proxy
<b>Resumen:</b>	El caso de uso inicia cuando el Componente realiza la petición de un servicio, el Proxy verifica que el Componente pueda realizar esta petición, consume el

	servicio web y le devuelve el resultado.
<b>Precondiciones:</b>	El Proxy debe haber consumido el servicio web solicitado.
<b>Referencias</b>	RF11, RF12, Consumir_Servicio_Web
<b>Prioridad</b>	Crítica
<b>Poscondiciones</b>	El Componente obtiene el resultado del servicio web consumido.

**Tabla 2.15** Descripción textual CUS12: Realizar\_Peticion.

<b>Caso de Uso:</b>	<b>CUS14: Verificar_Cantidad_Trazas</b>
<b>Actores:</b>	Usuario
<b>Resumen:</b>	El caso de uso inicia cuando el actor registra cualquier acción sobre el sistema, a partir de ahí se decide registrar una traza, para ello se verifica que la cantidad de trazas en la base de datos es mayor que 25 000 pasándose a balancear las mismas en caso de que exista sobrecarga, termina el caso de uso.
<b>Precondiciones:</b>	El usuario debe haber realizado una acción sobre el sistema.
<b>Referencias</b>	RF13, RF14, RF15, Balancear_Trazas (extendido)
<b>Prioridad</b>	Alta
<b>Poscondiciones</b>	El sistema verifica que se puede registrar una traza.

**Tabla 2.16** Descripción textual CUS14\_Verificar\_Carga\_Trazas.

<b>Caso de Uso:</b>	<b>CUS15: Balancear_Trazas</b>
<b>Actores:</b>	Usuario
<b>Resumen:</b>	El caso de uso inicia cuando el actor registra cualquier acción sobre el sistema, a partir de ahí se decide registrar una traza y para ello se debe verificar la cantidad de trazas, donde se determinó que la cantidad de trazas existentes en la base de datos excedían las 25 000 trazas, por lo que se procede balancear la cantidad de trazas que hay en la misma, comprobando el numero de trazas que hay en las Trazas Históricas, si estás tienen una sobrecarga se pasan a un archivo de texto, sino se guardan en las Trazas Históricas, permitiendo que se registre la traza producto de la acción del usuario, terminado el caso de uso.
<b>Precondiciones:</b>	El usuario debe haber realizado una acción sobre el sistema.
<b>Referencias</b>	RF13, RF14, RF15, Verificar_Cantidad_Trazas
<b>Prioridad</b>	Alta
<b>Poscondiciones</b>	Son balanceadas las trazas

**Tabla 2.17** Descripción textual CUS15\_Balancear\_Trazas.

<b>Caso de Uso:</b>	<b>CUS16: Buscar_Trazas</b>
<b>Actores:</b>	Administrador General(Inicia)
<b>Resumen:</b>	El caso de uso inicia cuando el usuario hace una búsqueda en la opción Auditoría. La misma se puede realizar por diferentes criterios, el sistema mostrara un listado de las trazas según los datos especificados.

<b>Precondiciones:</b>	El usuario debe tener permiso para realizar una búsqueda de trazas.
<b>Referencias</b>	RF16
<b>Prioridad</b>	Alta
<b>Poscondiciones</b>	El sistema devuelve un listado con el resultado de la búsqueda.

**Tabla 2.18** Descripción textual CUS16\_Buscar\_Trazas.

<b>Caso de Uso:</b>	<b>CUS17: Generar_Reporte</b>
<b>Actores:</b>	Administrador General(Inicia)
<b>Resumen:</b>	Inicia cuando el Administrador_General selecciona la opción Generar Reporte, para esto el mismo debe haber realizado la búsqueda de las trazas y debe tener al menos una configuración adicionada.
<b>Precondiciones:</b>	Se debe buscar las trazas.
<b>Referencias</b>	RF17, RF18, RF19, Adicionar_Configuracion (Caso de Uso extendido), Eliminar_Configuracion (Caso de Uso extendido), Buscar_Trazas (Caso de Uso incluido).
<b>Prioridad</b>	Alta
<b>Poscondiciones</b>	Se realiza un reporte determinado.

**Tabla 2.19** Descripción textual CUS17\_Generar\_Reporte

<b>Caso de Uso:</b>	<b>CUS18: Adicionar_Configuracion</b>
<b>Actores:</b>	Administrador General(Inicia)
<b>Resumen:</b>	Inicia cuando el Administrador_General selecciona la opción Configurar Reporte, el sistema le brinda la posibilidad de seleccionar los elementos que se desean mostrar en el reporte y luego guarda la configuración realizada.
<b>Precondiciones:</b>	El usuario debe tener permiso para realizar una configuración.
<b>Referencias</b>	RF18
<b>Prioridad</b>	Alta
<b>Poscondiciones</b>	Se adiciona una configuración.

**Tabla 2.20** Descripción textual CUS18\_Adicionar\_Configuracion

<b>Caso de Uso:</b>	<b>CUS19: Eliminar_Configuracion</b>
<b>Actores:</b>	Administrador General(Inicia)
<b>Resumen:</b>	Inicia cuando el Administrador_General selecciona la opción Configurar Reporte, selecciona la configuración que desea eliminar y elimina la configuración.
<b>Precondiciones:</b>	El usuario debe tener permiso para realizar una configuración.
<b>Referencias</b>	RF19
<b>Prioridad</b>	Alta

<b>Poscondiciones</b>	Se elimina una configuración.
-----------------------	-------------------------------

**Tabla 2.21** Descripción textual CUS19\_Eliminar\_Configuracion

<b>Caso de Uso:</b>	<b>CUS20: Graficar</b>
<b>Actores:</b>	Administrador General(Inicia)
<b>Resumen:</b>	Inicia cuando el Administrador_General selecciona en la opción Graficar un gráfico determinado. El sistema brinda la posibilidad de mostrar los gráficos: Estado de la auditoría, porcentaje de AD/Componentes, Ranking usuarios con AD y Ranking usuarios con OF.
<b>Precondiciones:</b>	El usuario debe tener permiso para realizar esta operación.
<b>Referencias</b>	RF20
<b>Prioridad</b>	Alta
<b>Poscondiciones</b>	Se muestra un gráfico específico.

**Tabla 2.22** Descripción textual CUS20\_Graficar

Este capítulo ha permitido alcanzar un mejor entendimiento del sistema. Se especificaron los requerimientos funcionales y no funcionales, los actores que intervienen y los casos de uso. Se han detallado nuevas funcionalidades para esta aplicación, lo que da paso a la realización del diseño, que es donde se requiere más esfuerzo para el desarrollo de la nueva versión del SAAA-SAS.

### CAPÍTULO 3: DISEÑO DEL SISTEMA

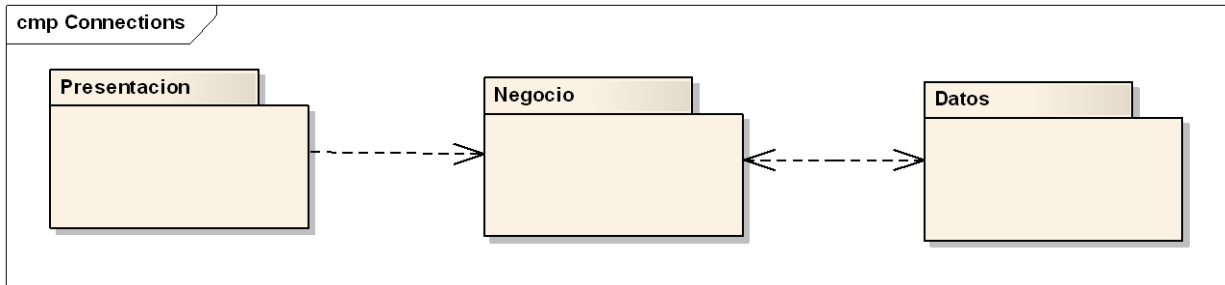
En este capítulo se modela el diseño del sistema para darle cumplimiento a los requerimientos levantados en el capítulo anterior. Se realizan los diagramas de interacción y de clases de los casos de uso más significativos para el desarrollo de la nueva versión del SAAA-SAS. El objetivo de este capítulo es obtener el modelo de diseño, que es donde se va a estructurar el sistema como tal para su posterior implementación, así como la descripción de sus clases para un mejor entendimiento.

#### 3.1 Modelo de Diseño

El Modelo de Diseño describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar. Es una abstracción del Modelo de Implementación y su código fuente, el cual fundamentalmente se emplea para representar y documentar su diseño. Es usado como entrada esencial en las actividades relacionadas a implementación. Representa a los casos de uso en el dominio de la solución.

##### 3.1.1 Estructura de Diseño

Para realizar el Modelo de Diseño se tiene en cuenta la descomposición en subsistemas, lo que representaría la estructura fundamental de la aplicación, donde se utiliza la arquitectura en tres capas para separar la lógica de negocio de la lógica de diseño. El siguiente diagrama muestra la estructura establecida para el desarrollo de la aplicación, este representa un conjunto de clases agrupadas en las diferentes capas:



**Fig.3.1** Diagrama que representa la organización de las clases del diseño en paquetes.

1.- Capa de presentación: es la que ve el usuario, presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de procesos. Esta capa se comunica únicamente con 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 y el motor de plantilla Smarty utilizado para la transformación de los ficheros HTML.

2.- 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 porque 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.

3.- Capa de datos: es donde residen los datos y es la encargada de acceder a los mismos. 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.

### 3.1.2 Definición de Elementos de Diseño

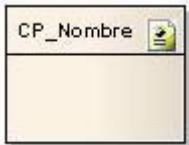
UML tiene varios mecanismos de extensión que fueron definidos para extender su semántica en aquellas situaciones en las que no es posible capturar todas las características de una determinada arquitectura o dominio en particular. Entre los estereotipos más usados se encuentra:



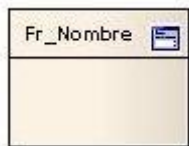
- **Estereotipo de las clases:**



<<Server Page>> Representa la página Web que tiene el código que se ejecuta en el servidor. Este código interactúa con recursos en el servidor. Las operaciones representan las funciones del código y los atributos las variables visibles dentro del alcance de la página.



<<Client Page>> Una instancia de Página Cliente es una página Web, con formato HTML. Mezcla de datos, presentación y lógica. Son interpretadas por el navegador. Sus atributos son las variables declaradas dentro del script.



<<Form>> Colección de elementos de entrada que son parte de una página cliente. Se relaciona directamente con la etiqueta de igual nombre del HTML. Sus atributos son los elementos de entrada del formulario (input boxes, text areas, radio buttons, check boxes y hidden fields).

- **Estereotipo de las relaciones:**

<<Build>>




<<Build>> Representa una asociación especial que relaciona las páginas cliente con las páginas servidoras, de forma general se expresa cómo las páginas que se encuentran en el servidor construyen las páginas en el cliente. Debe ser una relación direccional, donde una página servidora puede construir una o más páginas cliente.

<<Link>>



<<Link>> Expresa las asociaciones más comunes entre las páginas, en este caso la del hipervínculo; esta asociación siempre se origina desde una página cliente y apunta hacia otra página cliente o una página de servidor.

 **<<Submit>>** Es la relación que se crea siempre entre una página servidora y un formulario, a través de esta relación el formulario manda los valores de sus campos al servidor, para ser procesados por la página servidora. Cada formulario se relaciona con al menos, una página servidora a través de esta relación.

La página servidora además de construir una página cliente puede redireccionar el procesamiento a otra página. Esto se representa con una asociación con el estereotipo **<<Redirect>>**. Por otro lado una página servidora puede incluir a otra página servidora. Esta relación se representa con una asociación con el estereotipo **<<Include>>**.

<b>Hasta:</b>	<b>Página Cliente</b>	<b>Página Servidora</b>	<b>Formulario</b>
<b>Desde:</b>			
<b>Página Cliente</b>	<<Link>> <<Redirect>>	<<Link>> <<Redirect>>	Aggregation
<b>Página Servidora</b>	<<Build>> <<Redirect>>	<<Redirect>> <<Include>>	-----
<b>Formulario</b>	Aggregated By	<<Submit>>	-----

**Tabla.3.1** Relaciones entre las clases principales que conforman la extensión de UML para Web.

3.1.3 Diagramas del Diseño

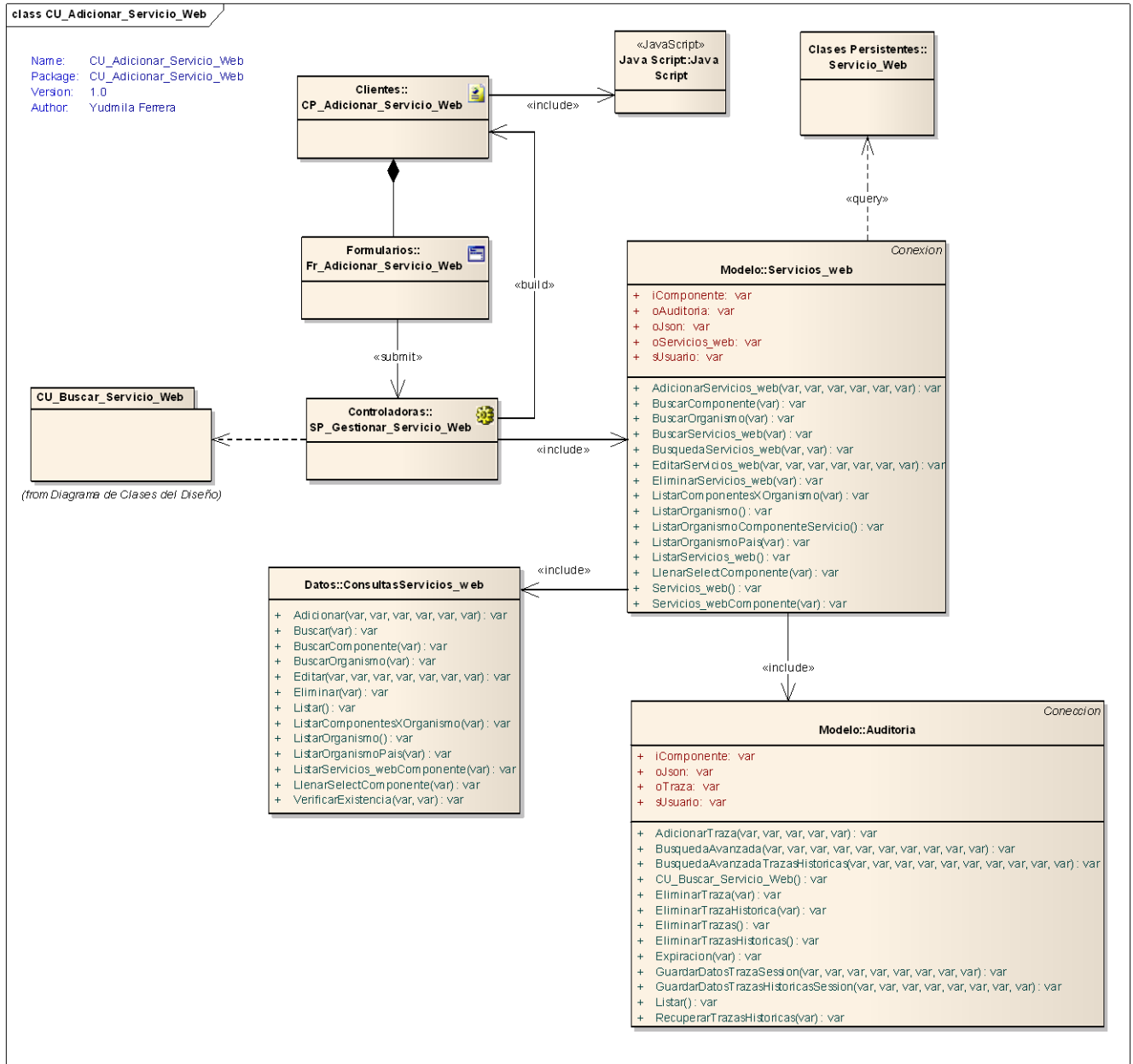
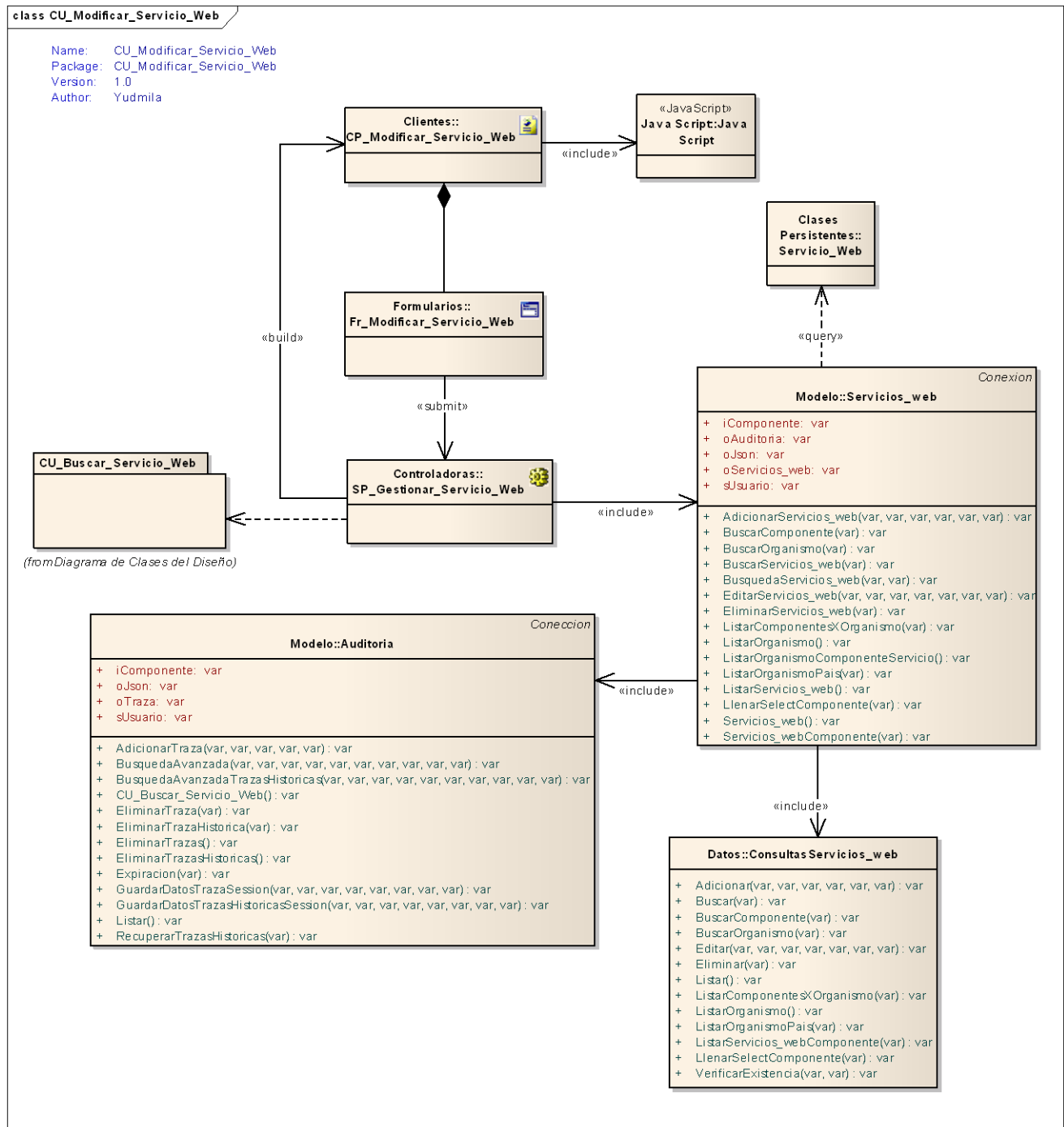
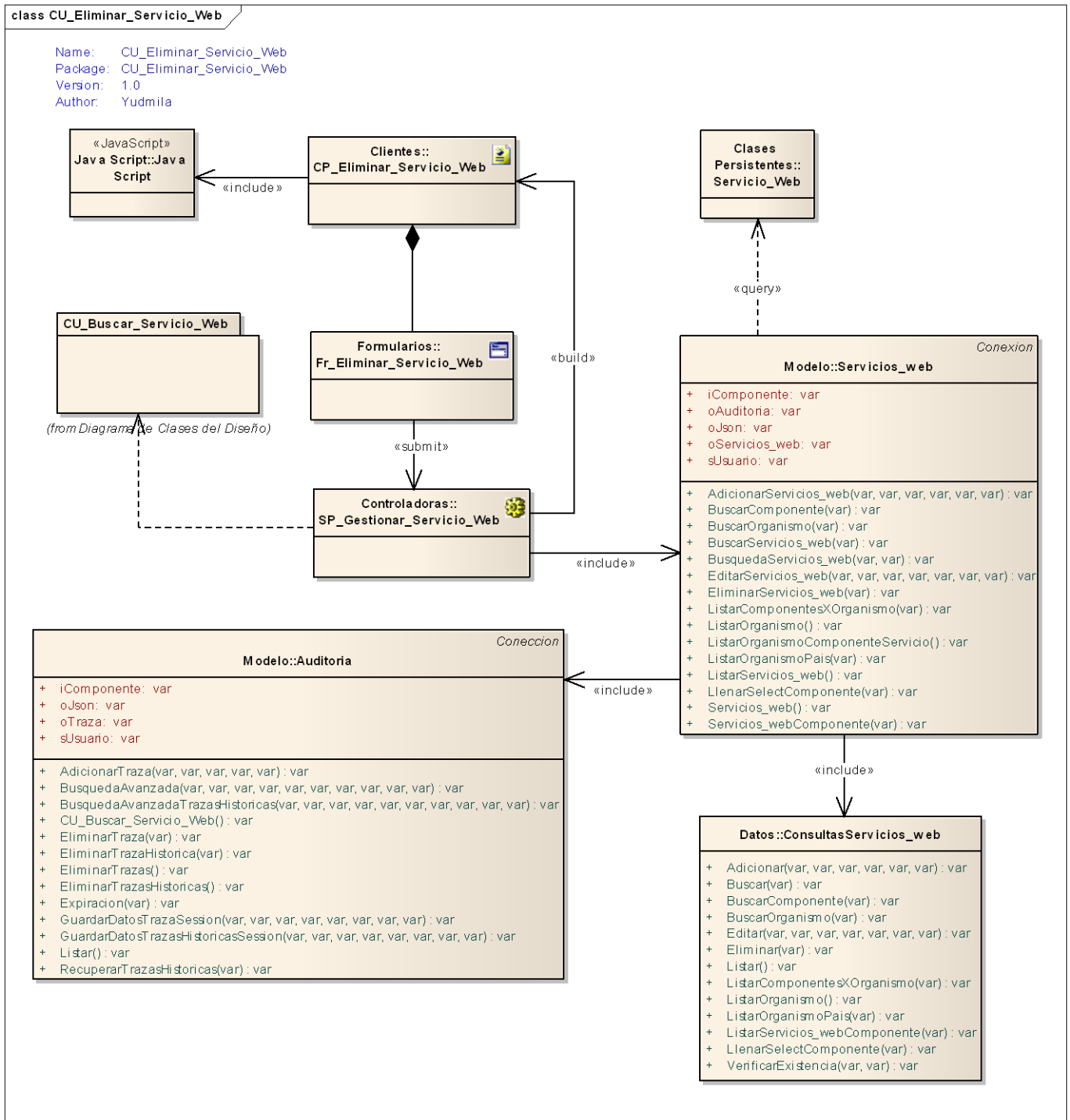


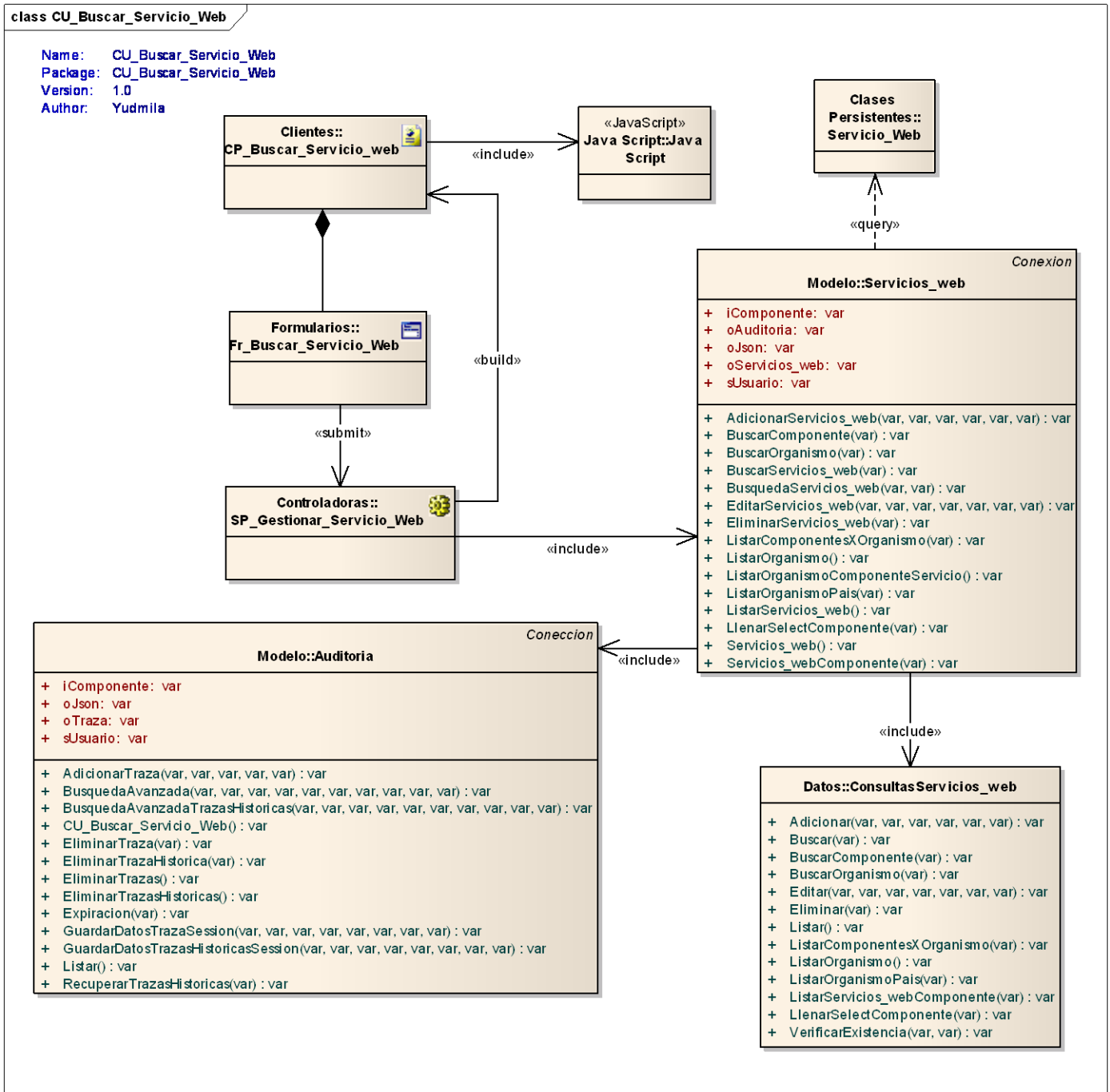
Fig.3.2 Diagrama de Clases del Diseño CUS1\_Adicionar\_Servicio\_Web.



**Fig.3.3** Diagrama de Clases del Diseño CUS2\_Modificar\_Servicio\_Web.



**Fig.3.4** Diagrama de Clases del Diseño CUS3\_Eliminar\_Servicio\_Web.



**Fig.3.5** Diagrama de Clases del Diseño CUS4\_Buscar\_Servicio\_Web.

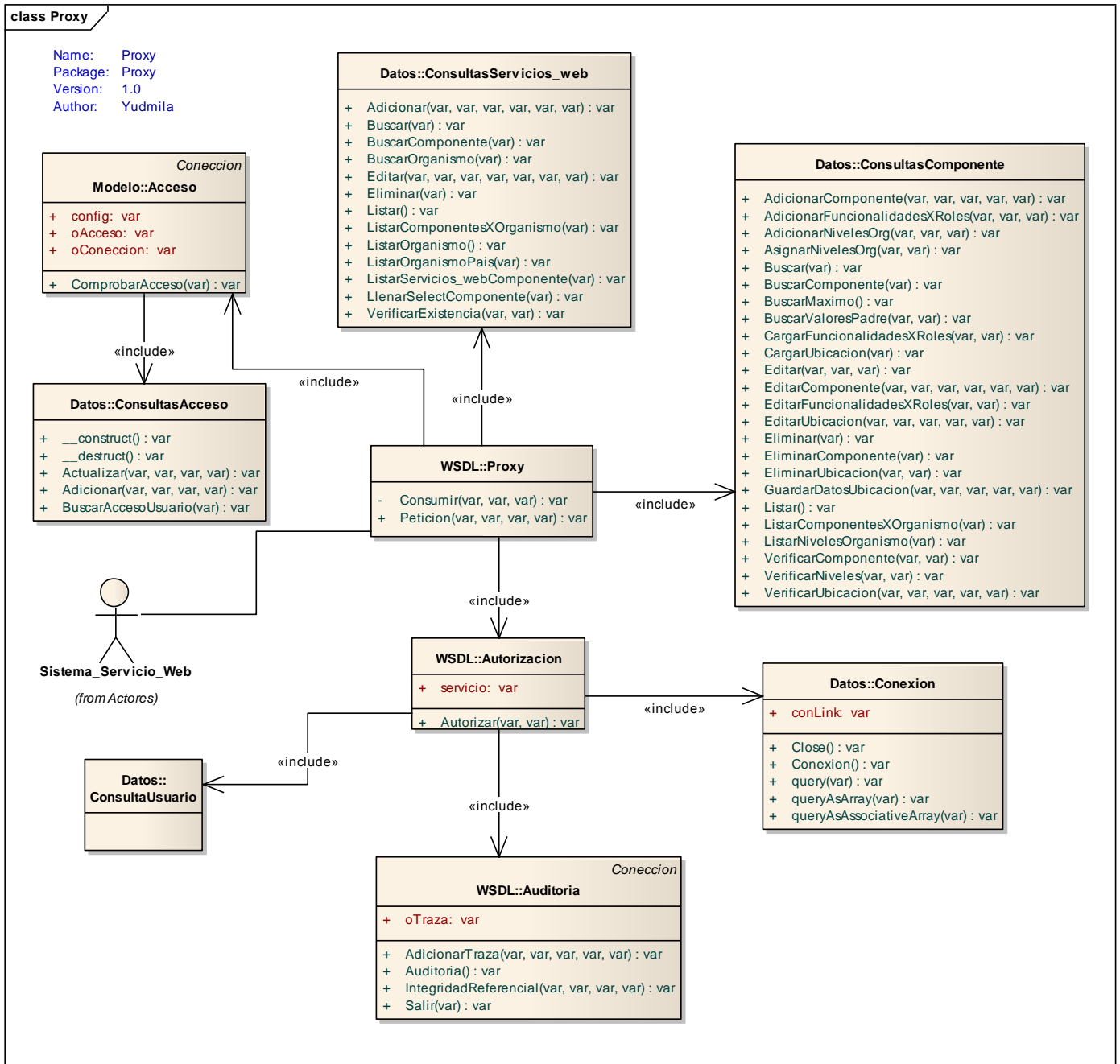


Fig.3.6 Diagrama de Clases del Diseño CUS11\_Consumir\_Servicio\_Web.

### 3.1.4 Descripción de las clases del diseño

<b>Nombre: CP_Adicionar_Servicio_Web</b>
<b>Tipo de Clase:</b> Client Page
<b>Descripción General:</b> Esta página le brinda la posibilidad al usuario de adicionar un servicio web, capturando los datos pertenecientes al servicio para luego ser almacenados en la base de datos del sistema.

**Tabla.3.2** Descripción página cliente CP\_Adicionar\_Servicio\_Web.

<b>Nombre: CP_Modificar_Servicio_Web</b>
<b>Tipo de Clase:</b> Client Page
<b>Descripción General:</b> Esta clase es una página Web que se ejecuta del lado del cliente. Muestra los datos del elemento que ha sido previamente seleccionado para modificar desde la página cliente CP_Listar_Servicio_Web. Permite modificar la información que se visualiza en la misma.

**Tabla.3.3** Descripción página servidora CP\_Modificar\_Servicio\_Web.

<b>Nombre: CP_Buscar_Servicio_Web</b>
<b>Tipo de Clase:</b> Client Page
<b>Descripción General:</b> Permite la búsqueda de los servicios web y luego mostrarlos en un listado a través de esta página se puede adicionar, modificar o eliminar la información almacenada o simplemente visualizar información.

**Tabla.3.4** Descripción página servidora CP\_Buscar\_Servicio\_Web



<b>Nombre: CP_Eliminar_Servicio_Web</b>
<b>Tipo de Clase:</b> Client Page
<b>Descripción General:</b> Esta página le brinda la posibilidad al usuario de eliminar un servicio web, y luego muestra el listado actualizado.

**Tabla.3.5** Descripción página servidora CP\_Eliminar\_Servicio\_Web

<b>Nombre: SP_Gestionar_Servicio_Web</b>
<b>Tipo de Clase:</b> Server Page
<b>Descripción General:</b> Esta clase servidora tiene como tarea fundamental construir las páginas clientes siguientes: CP_Adicionar_Servicio_Web CP_Modificar_Servicio_Web CP_Listar_Servicio_Web CP_Eliminar_Servicio_Web

**Tabla.3.6** Descripción página servidora SP\_Gestionar\_Servicio\_Web.

### 3.2 Modelo de Datos

Un modelo de datos es un lenguaje orientado a describir una base de datos. Permite describir los elementos de la realidad que intervienen en un problema dado y la forma en que se relacionan esos elementos entre sí.



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

Este punto corresponde a la descripción de las tablas de la base de datos del sistema, para el desarrollo de esta versión se agregan tres tablas nuevas: `tb_servicio_web`, `tb_acceso`, `tb_conf_reporte`.

<b>Nombre: <code>tb_servicio_web</code></b>			
Descripción: En esta tabla se almacena los datos referentes a los servicios web, permitiendo la gestión de los mismos.			
<b>ATRIBUTOS</b>	<b>Tipo</b>	<b>Nulo</b>	<b>Descripción</b>
<code>id_codificador_funcionalidad</code>	Integer	no	Identificador único para la funcionalidad en la tabla <b><code>tb_codificador_funcionalidad</code></b> (FK/PK).
<code>wsdl</code>	varchar	no	Dirección URL Principal del WSDL.
<code>wsdl_1</code>	varchar	si	Dirección URL Opcional 1 del WSDL.
<code>wsdl_2</code>	varchar	si	Dirección URL Opcional 2 del WSDL.

**Tabla.3.7** Descripción de la tabla de la Base de Datos: `tb_servicio_web`.

<b>Nombre: <code>tb_acceso</code></b>			
Descripción: En esta tabla se almacena los datos referentes a las peticiones hechas por los usuarios de los componentes al proxy, permitiendo una mejor gestión del proxy.			
<b>Atributos</b>	<b>Tipo</b>	<b>Nulo</b>	<b>Descripción</b>
<code>id_usuario</code>	varchar	no	Identificador único de usuario. <b>(PK)</b>
<code>cantidad_acceso</code>	Integer	no	Cantidad de peticiones hecha por el usuario.
<code>tiempo_inicial</code>	time	no	Tiempo en que se hace la primera petición, y de las próximas posibles.

**Tabla.3.8** Descripción de la tabla de la Base de Datos: `tb_acceso`.

<b>Nombre: tb_conf_reporte</b>			
<b>Descripción:</b> En esta tabla se almacena las configuraciones de los usuarios referentes a los datos relacionados con la auditoría, permitiendo seleccionar los campos que le son de interés y cuál configuración usarán para generación de los reportes relacionados con la misma.			
<b>Atributos</b>	<b>Tipo</b>	<b>Nulo</b>	<b>Descripción</b>
id_conf_reporte	Integer	no	Identificador único <b>(PK)</b> .
id_usuario	varchar	no	Identificador único para el usuario en la tabla <b>tb_usuario (FK/PK)</b> .
usuario	boolean	si	Especifica si el campo usuario está activo.
componente	boolean	si	Especifica si el campo componente está activo.
funcionalidad	boolean	si	Especifica si el campo funcionalidad está activa.
ip	boolean	si	Especifica si el campo ip está activo.
fecha	boolean	si	Especifica si el campo fecha, está activo.
nombre	boolean	si	Especifica si el campo nombre, está activo.
descripción	boolean	si	Especifica si el campo descripción está activo.
activo	boolean	si	Especifica cual configuración es la que está activa para generar el reporte.

**Tabla.3.9** Descripción de la tabla de la Base de Datos: tb\_conf\_reporte.

En este capítulo se realizó el Modelo de Diseño, lo que permitió alcanzar un mejor entendimiento de la estructura y diseño del sistema. También se realizaron los diagramas de clases correspondientes a las nuevas funcionalidades de la versión desarrollada. Esto constituye una abstracción para la implementación y facilita el trabajo a la hora de implementar ya que sirve como base para realizar el nuevo flujo de trabajo.

### CAPÍTULO 4: IMPLEMENTACIÓN

Una vez realizado el modelo del diseño, este capítulo tiene como objetivo tomar los elementos obtenidos para implementar el sistema, siguiendo una secuencia lógica de pasos. Se realiza además el Modelo de Implementación, lo permite obtener los diagramas de Componentes y de Despliegue para una mejor organización en la estructura de la aplicación. En este son mostradas todas las dependencias y relaciones entre los nodos físicos.

#### 4.2 Modelo de Implementación

El Modelo de Implementación es comprendido por un conjunto de componentes y subsistemas que constituyen la composición física de la implementación del sistema. Entre los componentes se pueden encontrar datos, archivos, ejecutables, código fuente y los directorios. Fundamentalmente, se describe la relación que existe desde los paquetes y clases del modelo de diseño a subsistemas y componentes físicos. (36)

##### 4.2.1 Diagramas de Componentes

Es un diagrama que muestra un conjunto de elementos del modelo, tales como componentes, subsistemas de implementación y sus relaciones. Se utiliza para modelar la vista estática de un sistema. Muestra la organización y las dependencias lógicas entre un conjunto de componentes software, algunos de sus estereotipos son: ejecutable, biblioteca, tabla, archivo y documento. Para la realización de la segunda versión del SAAA-SAS se obtiene los siguientes diagramas de componentes:

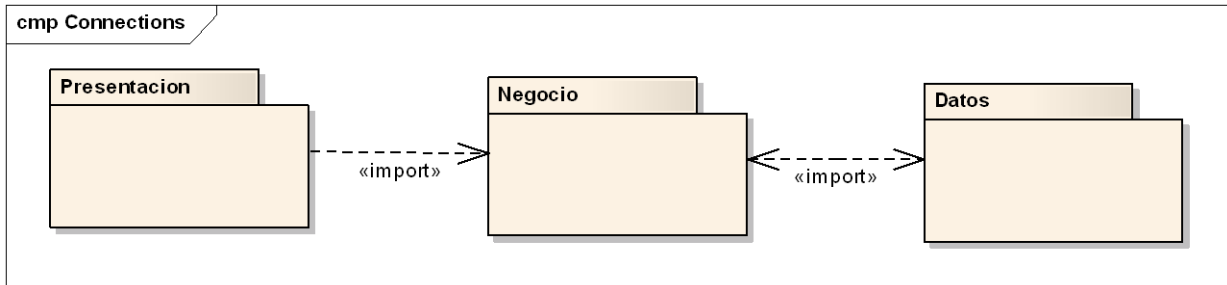


Fig.4.1 Estructura en subsistemas de Implementación.

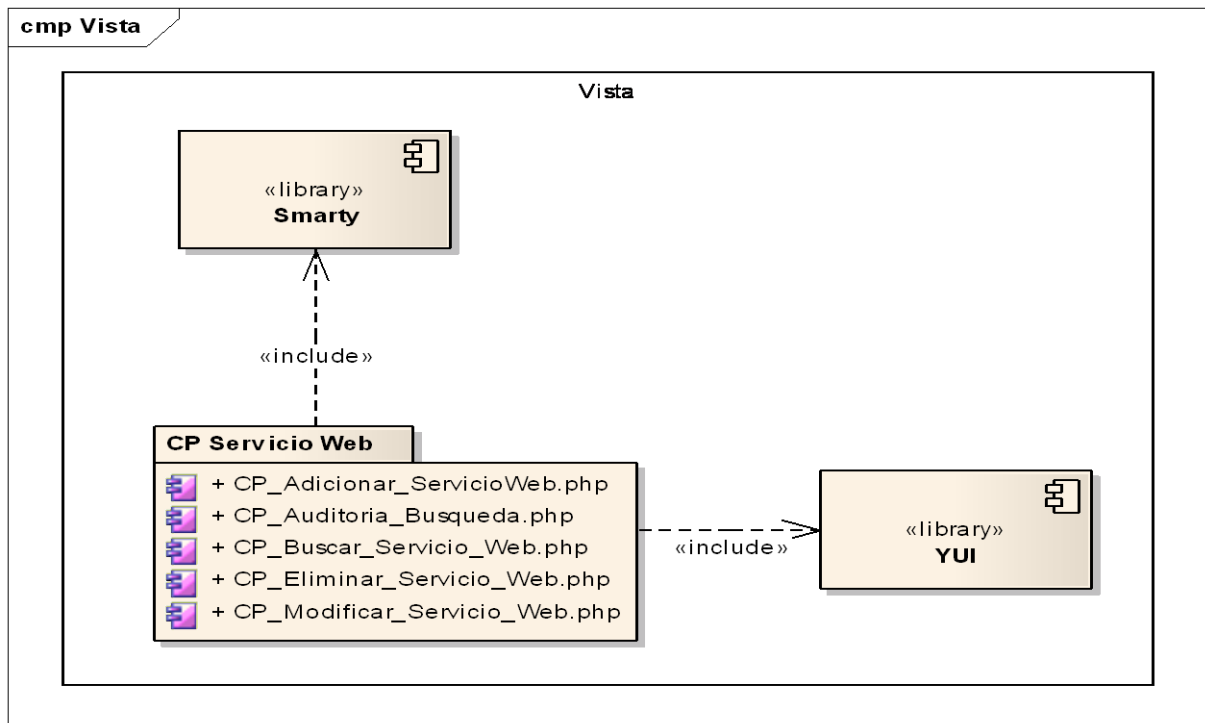


Fig.4.2 Estructura en subsistemas de Implementación: Capa de presentación.

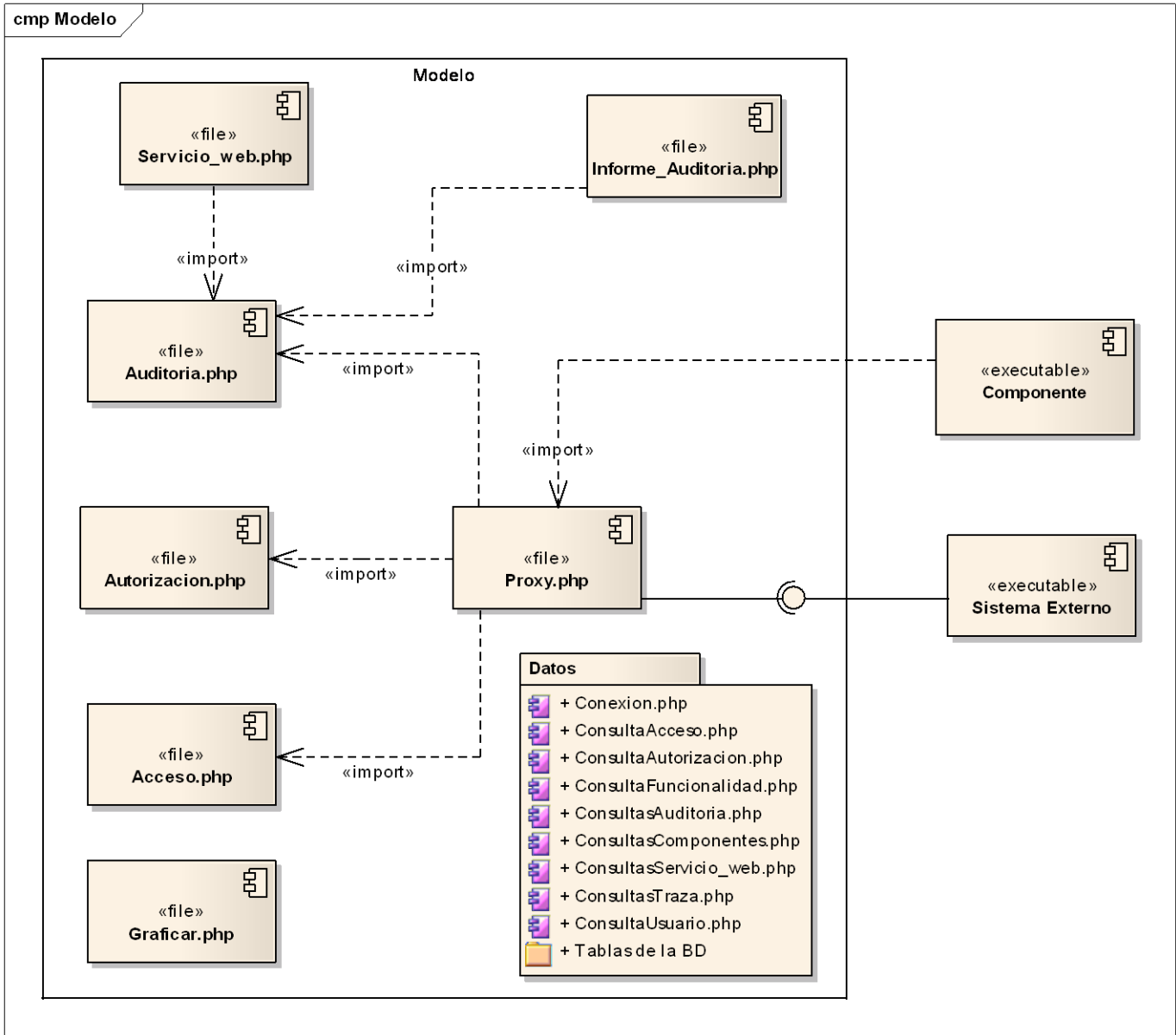


Fig.4.3 Estructura en subsistemas de Implementación: Capa de negocio.

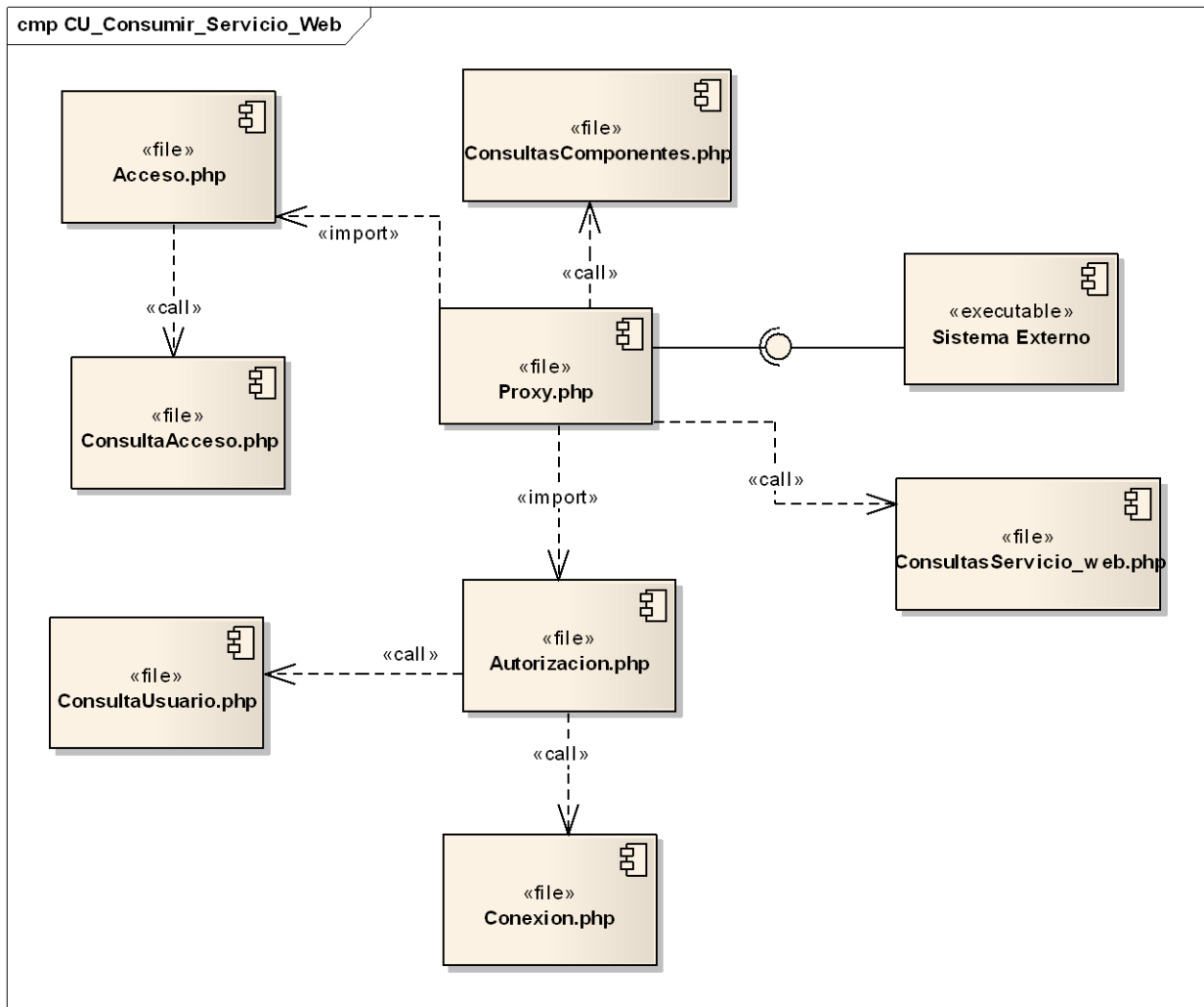


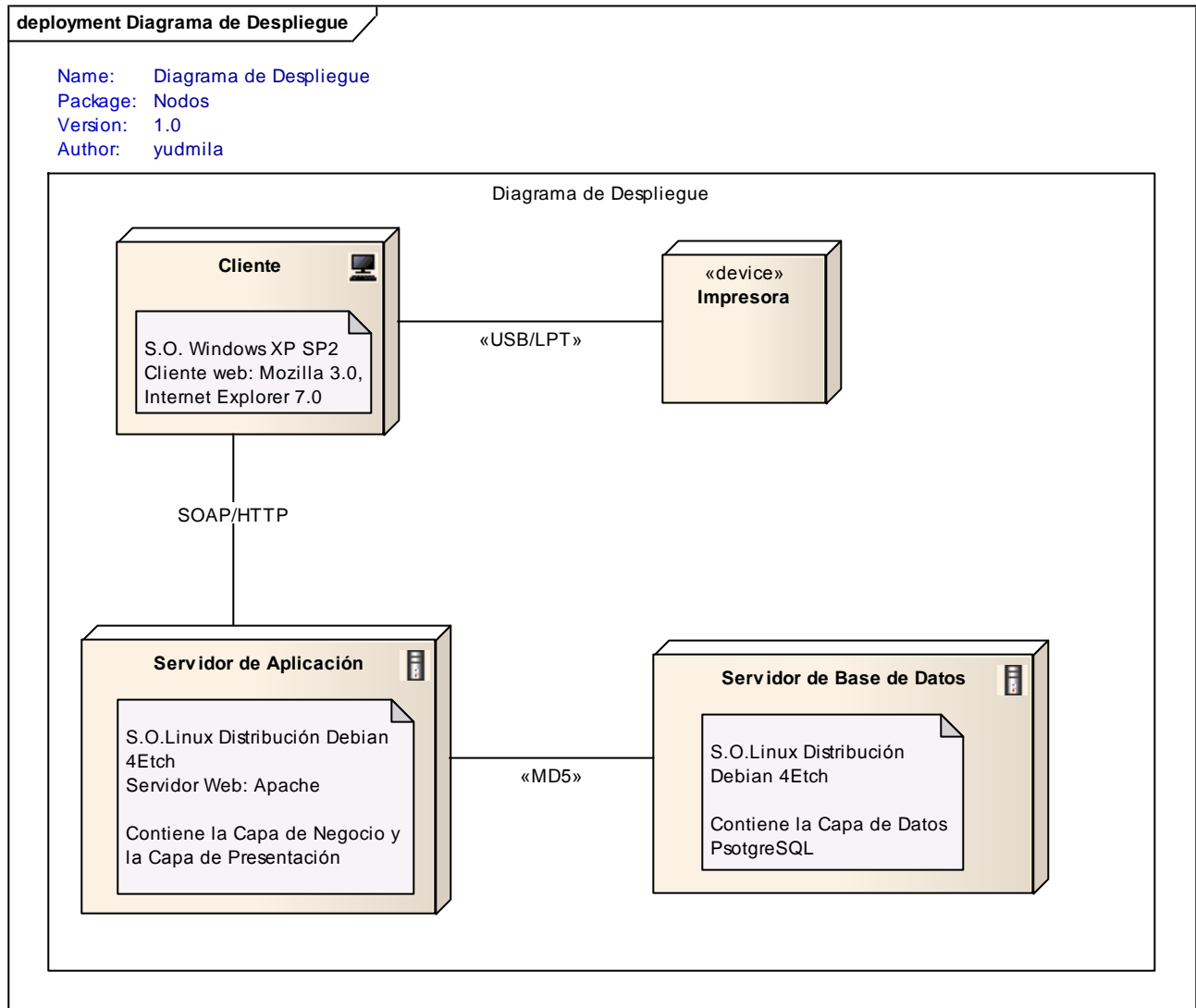
Fig.4.4 Diagrama de Componentes: CU\_Consumir\_Servicio\_Web.

### 4.2.2 Diagrama de Despliegue

Un diagrama de despliegue muestra las relaciones físicas entre los componentes hardware y software en el sistema final. Permite representar la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes software (procesos y objetos que se ejecutan en ellos).



El SAAA-SAS debe estar distribuido en dos nodos, uno representa el servidor de la aplicación, que contiene la capa de presentación y la lógica del negocio. Mientras que el otro, representa el servidor la base de datos conteniendo la capa de datos. Ambos funcionan sobre el sistema operativo LINUX en su distribución Debian y una computadora cliente sobre el sistema operativo Windows.



**Fig.4.7** Diagrama de Despliegue.

### 4.3 Descripción de la solución propuesta

Para el desarrollo de la segunda versión del SAAA-SAS se implementaron nuevas funcionalidades con el objetivo de mejorar el rendimiento de la aplicación así como el control de las peticiones que realizan los sistemas cliente a los servicios web.

#### 4.3.1 Servicio web Proxy

El servicio web Proxy sirve de mediador o intermediario entre las peticiones de otros sistemas clientes o usuarios. Brinda un mayor nivel de seguridad a los componentes cuyos requerimientos de seguridad son gestionados por el SAAA-SAS, debido a que mediante este servicio se garantiza que todos los accesos a los servicios web sean autorizados permitiendo que solo el sistema de seguridad conozca la ubicación real de los servicios web brindados, lo que posibilita que si hay cambios en la ubicación de algún servicio solo es necesario actualizarla en él, siendo un proceso completamente transparente para los usuarios que consumen los mismos.

El servicio web Proxy es capaz de buscar el servicio solicitado por un componente determinado en direcciones alternativas (previamente registradas, para el caso de sistemas que cuenten con servidores de respaldo) lo que garantiza una alta disponibilidad de los mismos. Cada vez que se realiza una petición se registra una serie de datos que permiten posteriormente analizar el comportamiento de los usuarios o sistemas que consumen los servicios, lo que posibilita evaluar las acciones de los mismos y contar con un historial que podría ser de utilidad para identificar tendencias y hacer predicciones de sus comportamientos futuros utilizando técnicas de minería de datos.

##### 4.3.1.1 Funcionamiento del servicio Proxy

El servicio Proxy permite un mayor control de todas las peticiones que realizan los sistemas externos, teniendo la responsabilidad de asignar los permisos para acceder a los servicios web solicitados. Para la utilización del mismo, en la gestión de los diferentes componentes (sistemas clientes), el administrador de configuración debe registrar los servicios web que brindan cada uno para su funcionamiento, teniendo en cuenta los aspectos como: el nombre, dirección en que se encuentran y en caso de estar instalado en

servidores de respaldo, el sistema brinda la posibilidad de poner dos direcciones alternativas. Este servicio funciona siguiendo una serie de pasos:

- Cuando el cliente realiza una petición del servicio web que quiere consumir, envía los parámetros asociados a dicho servicio web, así como el componente al que pertenece y el certificado digital que tiene asociado a él.
- Después de que el proxy recibe todos los parámetros, este comprueba que el componente enviado esté registrado en el sistema.
- Luego verifica que el servicio web solicitado, que debe haber sido gestionado anteriormente en el sistema, esté asignado al componente.
- Comprueba el acceso, teniendo en cuenta que el sistema cliente no haya excedido el número de peticiones máximas en un intervalo de tiempo determinado.
- Verifica si el usuario tiene permiso para acceder al servicio web que solicita y si este está disponible.
- Una vez que se comprueba que el servicio web está disponible, el proxy lo consume para brindárselo al sistema que lo solicita y de esta forma registra dicho acceso (registra traza).

En caso de que el servicio web solicitado no se encuentre disponible, se verifica su disponibilidad en dos direcciones alternativas que son proporcionadas de forma opcional por el administrador de configuración en el momento de registrar un componente. Seguidamente se muestran datos del servicio Proxy incluyendo: una breve descripción, parámetros de entrada y salida y un ejemplo sobre cómo consumirlo.

<b>Nombre del Servicio Web: Proxy</b>	
<b>Descripción General:</b> Este servicio es el encargado de consumir los servicios web solicitados para brindarlos a los sistemas clientes.	
<b>PARÁMETROS DE ENTRADA</b>	
<b>Descripción</b>	<b>Tipo</b>
Componente	Integer
Servicio	String
Certificado	String
Lista de parámetros	Array
<b>PARÁMETROS DE SALIDA</b>	
<b>Descripción</b>	<b>Tipo</b>
Petición	Array

**Tabla 4.1** Descripción del Servicio Web: Proxy.

<b>EJEMPLO DE IMPLEMENTACIÓN PARA CONSUMIR EL SERVICIO WEB PROXY</b>
<pre> &lt;?php try {     \$cliente = new SoapClient("http://10.36.7.34:5800/Aramis/VersionFinal/modelo/wsdl/servicios_proxy.php?wsdl");     \$params = array('parameters' =&gt; array('CI' =&gt; '86102726342'));     \$sCert = 'ee20bf0263cb7f4588fe0dc2b262ce82';     \$arg = array('idcomponente'=&gt;'11', 'servicio'=&gt;'ObtenerPersonaDadoCI', 'certificado'=&gt;\$sCert,                 'listaparametros'=&gt; \$params );     \$r= \$cliente-&gt;__call('Peticion',\$arg);     print_r(\$r); } catch (SoapFault \$f) {     print_r(\$f); } ?&gt; </pre>

**Figura 4.8** Ejemplo de implementación para consumir el Servicio Web: Proxy.

### 4.3.2 Pruebas de rendimiento

Para darle cumplimiento a uno de los objetivos propuestos para el desarrollo de la nueva versión del SAAA-SAS, se le hicieron pruebas de rendimiento tanto a la aplicación como a las tablas de la base de datos relacionadas con las trazas generadas por los usuarios. Para esto se utilizó la herramienta JMeter y

se obtienen resultados que demostraron la necesidad de implementar un mecanismo que permitiera el balanceo de la información de forma automática. A continuación se muestran los resultados obtenidos de un grupo de pruebas, así como las características del computador en que fueron realizadas:

### **Características del computador:**

- **Sistema Operativo:** Windows XP Professional Service Pack 2.
- **Procesador:** Intel (R) Pentium (R) D CPU 3.00 GHz.
- **Tarjeta Madre:** ASUSTeK Computer INC. P5LD2-VM-CR-SI.
- **Memoria RAM:** 512 MB.

### **Resultados de las pruebas a la base de datos y al sistema:**

Con las pruebas realizadas para ver el comportamiento del rendimiento del sistema se tuvo en cuenta la memoria límite del servidor web, que permite establecer la cantidad de memoria que consume un script, en previsión de scripts que puedan estar mal programados y que podrían llegar a consumir toda la memoria y bloquear el sistema. Teniendo el servidor web de PHP como memoria límite 8 MB por defecto.

Se realizaron varias pruebas con los valores de memoria límite: 64 MB, 128 MB y 512 MB, y se llegó a la conclusión de que haciendo uso de cualquiera de estos valores, los tiempos de respuestas para la búsqueda de todas las trazas almacenadas en la base de datos son aproximadamente iguales, por lo que se decidió tomar 64 MB para la configuración del servidor, para obtener un mejor rendimiento del sistema. A continuación se muestra una tabla que contiene los resultados de las pruebas realizadas, donde se utilizan los valores antes mencionados para diferentes cantidades de trazas almacenadas en la base de datos:

Cantidad Trazas	# Muestra	Memoria Límite (MB)	Tiempo Aplicación (segundos)	Tiempo Base de Datos (segundos)
30 000	1	64	10 y 12	0.7 y 1
		128	10 y 11.91	
		512	10 y 11.9	
	10	64	30 y 32	5.5 y 6
		128	30.89 y 31	
		512	30.23 y 31	
	20	64	34 y 36	10 y 11
		128	34.76 y 35	
		512	34 y 35	
	50	64	42 y 44	16 y 18
		128	42.36 y 43	
		512	42 y 43	
40 000	1	64	16 y 18	1 y 1.3
		128	15.76 y 16	
		512	15.85 y 16.06	
50 000	1	64	20 y 22	1 y 1.8
		128	19 y 21	
		512	19 y 20	

**Tabla.4.1** Rendimiento del sistema y de la base de datos con diferentes cantidades de trazas almacenadas.

- **Cantidad Trazas:** Cantidad de trazas almacenadas en la base de datos.
- **# Muestras:** Cantidad de peticiones a la URL.
- **Memoria Límite:** Memoria límite que consume un script
- **Tiempo Aplicación:** Tiempo de respuesta para las peticiones en la aplicación (en segundos).
- **Tiempo Base de Datos:** Tiempo de respuesta para las peticiones en la base de datos (en segundos).

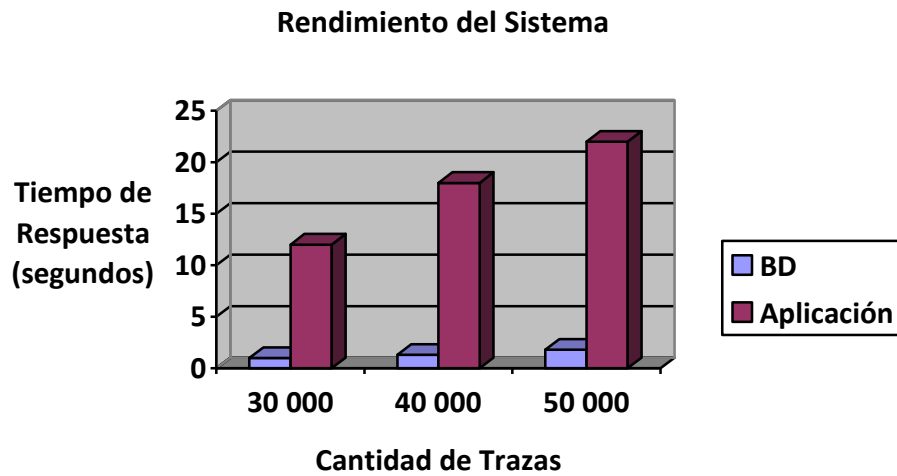
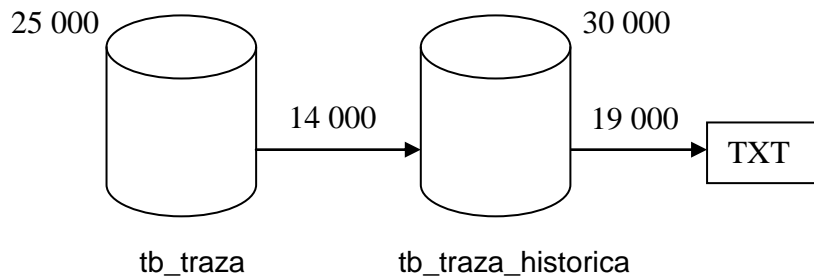


Fig.4.8 Gráfico que representa los tiempos de respuestas a una petición.

Para la memoria límite de 8 MB, el sistema responde para una petición con una cantidad de 5 000 trazas almacenadas en un tiempo de respuesta de 2 segundos aproximadamente. Una vez concluidas las pruebas se define para la implementación de un mecanismo que permita el balanceo de las trazas en la base de datos, que la cantidad máxima de información que va a gestionar la aplicación es de 30 000 trazas con una memoria límite de 64 MB teniendo un tiempo de respuesta entre 10 y 12 segundos.

### 4.3.3 Mecanismo de balanceo de trazas en la base de datos

Para la implementación de un mecanismo automatizado que permite el balanceo de trazas en la base de datos se tuvieron en cuenta los resultados de las pruebas realizadas anteriormente. Esto permite mejorar el rendimiento del sistema en la realización de las búsquedas relacionadas con la Auditoría. (Ver Fig.4.9)



**Fig.4.9** Mecanismo de balanceo de las trazas.

La figura anterior muestra el mecanismo que se tiene en cuenta para el balanceo de la información almacenada en la base de datos. Antes que el sistema registre una traza, verifica si en la tabla ***tb\_traza*** la cantidad de información almacenada sobrepasa los 24 000 registros. En caso de que sea mayor, se comprueba lo mismo en la ***tabla tb\_traza\_historica*** pero para la cantidad de 29 000, si se cumple esta condición se exportan 19 000 trazas a un fichero TXT, se escoge este, por ser un formato liviano y sencillo para el almacenamiento de la información y luego, se realiza el balanceo moviendo de una tabla a otra los 14 000 registros más antiguos de la misma. Las cifras que se presentan en este epígrafe con el objetivo de centrar el proceso del balanceo de datos constituyen una propuesta del SAAA-SAS a raíz de las pruebas expuestas en la sección anterior, pero el sistema permite a los usuarios ajustar estas cifras a las necesidades de sus aplicaciones.

#### 4.3.4 Diseño de los Reportes de Auditoría

Para la implementación de esta segunda versión del SAAA-SAS, uno de los objetivos trazados fue mejorar el diseño de los reportes relacionados con la auditoría, para ello el sistema brinda nuevas funcionalidades que le permiten al usuario, gestionar la configuración de los reportes, lo que posibilita seleccionar los elementos de mayor interés y guardar las configuraciones, dejando activa la última configuración realizada para su posterior uso.



El sistema permite además graficar la información relacionada con la auditoría para un mejor conocimiento del estado de la aplicación. Grafica los datos referentes a: por ciento de trazas almacenadas, los componentes y los usuarios con mayor acceso denegado y los usuarios con mayor cantidad operaciones fallidas, siendo esta información de mucha importancia para el administrador del sistema, ya que le permite tener un mayor control sobre el comportamiento de algunos elementos de interés relacionados con las acciones de los usuarios en el sistema. **(Ver Anexo 11)**

### 4.3.5 Módulo de Instalación

Con el objetivo de lograr una aplicación portable y fácil de instalar se implementó un módulo de instalación personalizado, este cuenta con cinco pasos fundamentales, que le permite a un usuario con pocos conocimientos informáticos la capacidad de instalar el sistema de una manera más sencilla. Para el proceso de instalación el módulo cuenta con los siguientes pasos:

1. En el primer paso se comprueba que esté habilitado en el servidor web las extensiones necesarias para el correcto funcionamiento del sistema, en caso de no estarlo, el usuario no podrá proceder al siguiente paso de la instalación. Además se recomienda activar un grupo de valores de PHP para asegurar una correcta compatibilidad con la aplicación. **(Ver Anexo 13)**
2. Este paso corresponde a la configuración de la base de datos, donde el usuario debe especificar algunos elementos como: el servidor, el puerto de comunicación de la misma, el usuario y la contraseña de PostgreSQL y por último el nombre de la base de datos que utilizará la aplicación. **(Ver Anexo 14)**
3. En este paso el módulo de instalación brinda la información de acceso al sistema (credenciales) y la posibilidad de cambiarlas en caso de que el usuario lo desee. **(Ver Anexo 15)**
4. En la aplicación se definen varias configuraciones recomendadas para optimizar el rendimiento del sistema, no obstante es en este paso donde el usuario puede reconfigurar o ajustar las mismas a sus necesidades. **(Ver Anexo 16)**

5. Este último paso es el encargado notificar al usuario que el proceso de instalación ha transcurrido satisfactoriamente y lo redirecciona al SAAA-SAS una vez que el usuario haya eliminado la carpeta de instalación. **(Ver Anexo 17)**

Como conclusión de este capítulo se obtuvieron los diagramas de Componentes y Despliegue. Estos tienen una distribución de las clases agrupadas en subsistemas y una representación de los nodos físicos que mostrarían las relaciones existentes entre los componentes hardware y software en el sistema final. Además quedó reflejado como se ha realizado la implementación de las principales funcionalidades que debe tener el sistema.

### CONCLUSIONES

En el desarrollo del presente trabajo de diploma se han cumplido el objetivo y las tareas propuestas por lo que se concluye:

- Se generaron todos los artefactos relacionados con el diseño definido por la metodología de desarrollo RUP, lo que constituyó la base para la implementación de la nueva versión del SAAA-SAS.
- Se implementó un servicio web que utiliza el patrón de diseño Proxy, funcionalidad que en términos de seguridad entre servicios web, fortalece las prestaciones del SAAA-SAS.
- Se analizó el rendimiento del sistema cuando existe un aumento de la cantidad de trazas almacenadas en la base de datos, tarea que permitió determinar un grupo concreto de elementos de gran utilidad para mejorar el rendimiento de la aplicación que dio paso además, a la implementación de un mecanismo que permite balancear las trazas de las tablas donde se almacena la información relacionada con los usuarios, lo que aporta al SAAA-SAS mayor eficiencia y confiabilidad en la gestión de la auditoría de los componentes.
- Se obtuvo un mejor diseño de los reportes asociados a la Auditoría, que le permite a los usuarios seleccionar los elementos que les resulten de mayor interés y lograr una mejor estructura de los reportes a generar.
- Se implementó un módulo de instalación del sistema que le permite a un usuario con pocos conocimientos informáticos facilitarle el proceso de instalación del mismo.

### RECOMENDACIONES

Para darle un seguimiento al presente trabajo se propone lo siguiente:

- Utilizar una capa de abstracción de bases de datos con el fin de obtener un producto más portable, permitiéndole a la aplicación funcionar correctamente con otros SGBD como: SQLite, MySQL y Oracle, en dependencia de las necesidades de los usuarios que deseen utilizar el sistema.
- Utilizar técnicas de minería de datos sobre la información relacionada con la auditoría del sistema con el objetivo de identificar tendencias en el comportamiento de los usuarios y poder realizar predicciones de sus comportamientos en el futuro.
- Estudiar el proceso de sindicación de servicios web, elemento que sería de vital importancia debido a que serviría para mostrar en un período determinado el estado de los servicios web brindados por el SAAA-SAA u otros sistemas externos.

### REFERENCIAS BIBLIOGRÁFICAS

1. Solis Mulet, Hector Manuel y Rojas Ríos, Danisbel. Componente de Seguridad. [En línea] Junio de 2008. [Citado el: 5 de Diciembre de 2008.]
2. Idem a la referencia 1
3. Idem a la referencia 1
4. Marañón, Gonzalo Álvarez. CSIC. Autenticación y Autorización. [En línea] 2001. [Citado el: 10 de Enero de 2009.] <http://www.iec.csic.es/CRIPTONOMICON/autenticacion/control.html>.
5. Racciatti, Hernán Marcelo. Seguridad en Aplicaciones Web. [En línea] Noviembre de 2005. [Citado el: 10 de Enero de 2009.] [http://www.hernanracciatti.com.ar/articles/HPP27\\_Seguridad\\_en\\_Aplicaciones\\_Web.pdf](http://www.hernanracciatti.com.ar/articles/HPP27_Seguridad_en_Aplicaciones_Web.pdf).
6. Idem a la referencia 1
7. Mis Algoritmos. Seguridad en aplicaciones Web. [En línea] [Citado el: 10 de Enero de 2009.] <http://www.mis-algoritmos.com/seguridad-en-aplicaciones-web>.
8. Kioskea.net. Protocolo LDAP. [En línea] [Citado el: 13 de Enero de 2009.] <http://es.kioskea.net/contents/internet/ldap.php3>.
9. LDAP.es. Ventajas en el uso de LDAP. [En línea] 19 de Noviembre de 2004. [Citado el: 13 de Enero de 2009.] <http://www.ldap-es.org/contenido/04/11/1.2.-ventajas-en-el-uso-de-ldap>.
10. Rojo, Oscar. AUGCYL. Introducción a los Sistemas Distribuidos. [En línea] [Citado el: 13 de Enero de 2009.] <http://www.augcyl.org/?q=glol-intro-sistemas-distribuidos> .
11. Gómez, Karel y Arencibia, Annia. Junio de 2007. [Citado el: 13 de Enero de 2009.]

12. Oktaba, Hanna. Ciencia e Ingeniería de la Computación. Introducción a Patrones. [En línea] [Citado el: 15 de Enero de 2009.] <http://www.mcc.unam.mx/~cursos/Algoritmos/javaDC99-2/patrones.html>.
13. SOA. [En línea] [Citado el: 15 de Enero de 2009.] [http://74.125.47.132/search?q=cache:xsXmRpr3FckJ:download.microsoft.com/download/c/2/c/c2ce8a3a-b4df-4a12-ba18-7e050aef3364/070717-Real\\_World\\_SOA.pdf+Arquitectura+Orientada+a+Servicios&hl=es&ct=clnk&cd=7&gl=cu](http://74.125.47.132/search?q=cache:xsXmRpr3FckJ:download.microsoft.com/download/c/2/c/c2ce8a3a-b4df-4a12-ba18-7e050aef3364/070717-Real_World_SOA.pdf+Arquitectura+Orientada+a+Servicios&hl=es&ct=clnk&cd=7&gl=cu).
14. Channel Planet. Beneficios de la Arquitectura Orientada a Servicios. [En línea] 12 de Febrero de 2008. [Citado el: 15 de Enero de 2009.] <http://www.channelplanet.com/?idcategoria=19497>.
15. Ruiz, Iván. Una de código. El paradigma Modelo Vista Controlador. [En línea] 30 de Mayo de 2007. [Citado el: 15 de Enero de 2009.] <http://www.unadecodigo.com/2007/05/30/el-paradigma-modelo-vista-controlador-tutorial-ror-ii/>.
16. HTML. [En línea] [Citado el: 15 de Enero de 2009.] [http://209.204.223.237/albergue/dweb/leng\\_marcas.html](http://209.204.223.237/albergue/dweb/leng_marcas.html).
17. Desarrolloweb.com. Qué es XML. [En línea] [Citado el: 15 de Enero de 2009.] <http://www.desarrolloweb.com/articulos/449.php>.
18. W3C. Guía Breve de XHTML. [En línea] 7 de Febrero de 2008. [Citado el: 15 de Enero de 2009.] <http://www.w3c.es/Divulgacion/Guiasbreves/XHTML>.
19. Ídem a la referencia 1.
20. Qué es UML. [En línea] 7 de Marzo de 2003. [Citado el: 15 de Enero de 2009.] <http://mayi.polanco.googlepages.com/TRABAJODEINGSOFTWAREII.doc>. Fecha: 07/03/2003.
21. Mozilla. Servicios Web XML. [En línea] [Citado el: 15 de Enero de 2009.] [https://developer.mozilla.org/es/Servicios\\_Web\\_XML](https://developer.mozilla.org/es/Servicios_Web_XML).
22. Smarty. Qué es Smarty. [En línea] [Citado el: 15 de Enero de 2009.] Enero <http://www.smarty.net/manual/es/what.is.smarty.php>

23. Garrett, Jesse James. Maestros del web. Ajax. [En línea] 11 de Junio de 2005. [Citado el: 15 de Enero de 2009.] <http://www.maestrosdelweb.com/editorial/ajax/>.
24. Wikipedia. HTTPS. [En línea] [Citado el: 15 de Enero de 2009.] [http://es.wikipedia.org/wiki/Hypertext\\_Transfer\\_Protocol\\_Secure](http://es.wikipedia.org/wiki/Hypertext_Transfer_Protocol_Secure).
25. Ídem a la referencia 1
26. PostgreSQL. [En línea] [Citado el: 15 de Enero de 2009.] [http://www.netpecos.org/docs/mysql\\_postgres/x15.html#AEN30](http://www.netpecos.org/docs/mysql_postgres/x15.html#AEN30).
27. Ciberaula. Introducción al Apache. [En línea] 2006. [Citado el: 15 de Enero de 2009.] [http://linux.ciberaula.com/articulo/linux\\_apache\\_intro/](http://linux.ciberaula.com/articulo/linux_apache_intro/).
28. Sparx Systems. Enterprise Architect. [En línea] 2007. [Citado el: 15 de Enero de 2009.] <http://www.sparxsystems.com.ar/products/ea.html>.
29. Osmosis Latina. Jmeter. [En línea] 20 de Octubre de 2005. [Citado el: 21 de Abril de 2009.] <http://www.osmosislatina.com/jmeter/basico.htm>.
30. SQLManager.net. EMS SQL Manager for PostgreSQL. [En línea] [Citado el: 15 de Enero de 2009.] <http://sqlmanager.net/products/postgresql/manager>.
31. Ídem a la referencia 1
32. Tigris.org. Especificación de Requerimientos de Software . [En línea] [Citado el: 26 de Febrero de 2009.] <http://readysset.tigris.org/nonav/es/templates/srs.html>.
33. Ídem a la referencia 1
34. MeRinde. Modelo de Casos de Uso. [En línea] [Citado el: 2 de Marzo de 2009.] [http://merinde.rinde.gob.ve/index.php?option=com\\_content&task=view&id=89&Itemid=295](http://merinde.rinde.gob.ve/index.php?option=com_content&task=view&id=89&Itemid=295).
35. Ídem a la referencia 1

36. MeRinde. Modelo de Implementación. [En línea] [Citado el: 12 de Abril de 2009.]  
[http://merinde.rinde.gob.ve/index.php?option=com\\_content&task=view&id=96&Itemid=297](http://merinde.rinde.gob.ve/index.php?option=com_content&task=view&id=96&Itemid=297).



**BIBLIOGRAFÍA**

1. AUGCYL. Introducción a los Sistemas Distribuidos. Disponible en: <http://www.augcyl.org/?q=glo1-intro-sistemas-distribuidos>
2. Autenticación y Autorización. 2001. Disponible en: <http://www.iec.csic.es/CRIPTONOMICON/autenticacion/control.html>.
3. Centro de Control. Junio 2007.
4. Ciberaula. Introducción al Apache. 2006. Disponible en: [http://linux.ciberaula.com/articulo/linux\\_apache\\_intro](http://linux.ciberaula.com/articulo/linux_apache_intro)
5. Ciencia e Ingeniería de la Computación. Introducción a Patrones. Disponible en: <http://www.mcc.unam.mx/~cursos/Algoritmos/javaDC99-2/patrones.html>
6. Componente de Seguridad. Junio 2008.
7. Desarrolloweb.com. Qué es XML. Disponible en: <http://www.desarrolloweb.com/articulos/449.php>
8. Dreamweaver. Disponible en: <http://www.desarrolloweb.com/articulos/332.php>
9. Kioskea.net. Protocolo LDAP. Disponible en: <http://es.kioskea.net/contents/internet/ldap.php3>
10. LDAP.es. Ventajas en el uso de LDAP. Noviembre 2004. Disponible en: <http://www.ldap-es.org/contenido/04/11/1.2.-ventajas-en-el-uso-de-ldap>
11. Lenguajes de marcas. Abril 2007. Disponible en: [http://209.204.223.237/albergue/dweb/leng\\_marcas.html](http://209.204.223.237/albergue/dweb/leng_marcas.html)
12. MeRinde. Modelo de Implementación. Disponible en: [http://merinde.rinde.gob.ve/index.php?option=com\\_content&task=view&id=96&Itemid=297](http://merinde.rinde.gob.ve/index.php?option=com_content&task=view&id=96&Itemid=297)

13. Patrones. Octubre 2003. Disponible en: <http://www.code4net.com/archives/000030.html>
14. PostgreSQL. Disponible en: [http://www.netpecos.org/docs/mysql\\_postgres/x15.html#AEN30](http://www.netpecos.org/docs/mysql_postgres/x15.html#AEN30)
15. Seguridad en Aplicaciones Web. Noviembre 2005. Disponible en: [http://www.hernanracciatti.com.ar/articles/HPP27\\_Seguridad\\_en\\_Aplicaciones\\_Web.pdf](http://www.hernanracciatti.com.ar/articles/HPP27_Seguridad_en_Aplicaciones_Web.pdf)
16. Smarty. Qué es Smarty. Disponible en: <http://www.smarty.net/manual/es/what.is.smarty.php>
17. Sparx Systems. Enterprise Architect. 2007. Disponible en: <http://www.sparxsystems.com.ar/products/ea.html>
18. SQLManager.net. EMS SQL Manager for PostgreSQL. Disponible en: <http://sqlmanager.net/products/postgresql/manager>
19. The Apache Jakarta Projec. Disponible en: <http://jakarta.apache.org/jmeter/>
20. Zend Studio. Junio 2003. Disponible en : <http://www.desarrolloweb.com/articulos/1178.php>

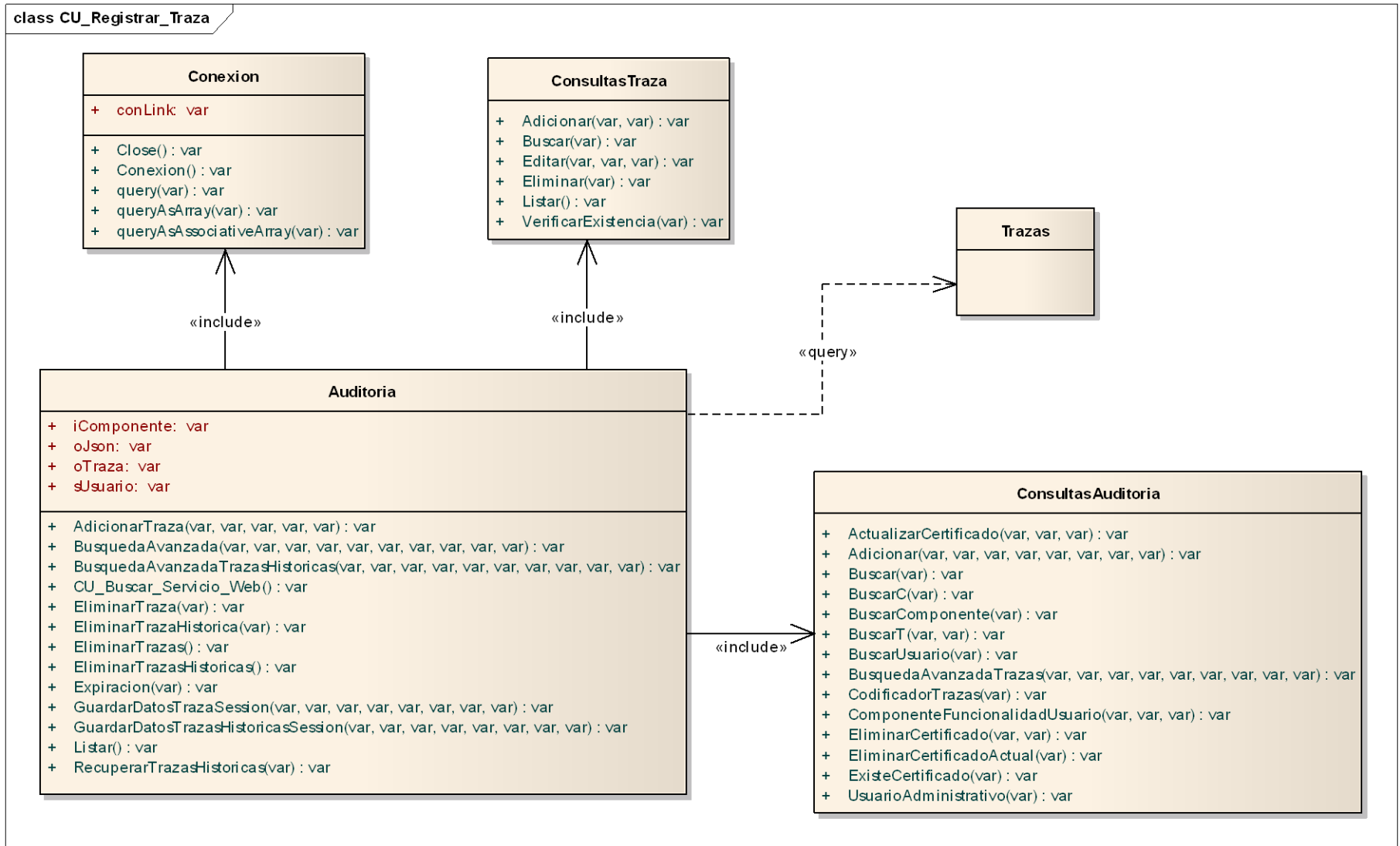
---

## ANEXOS

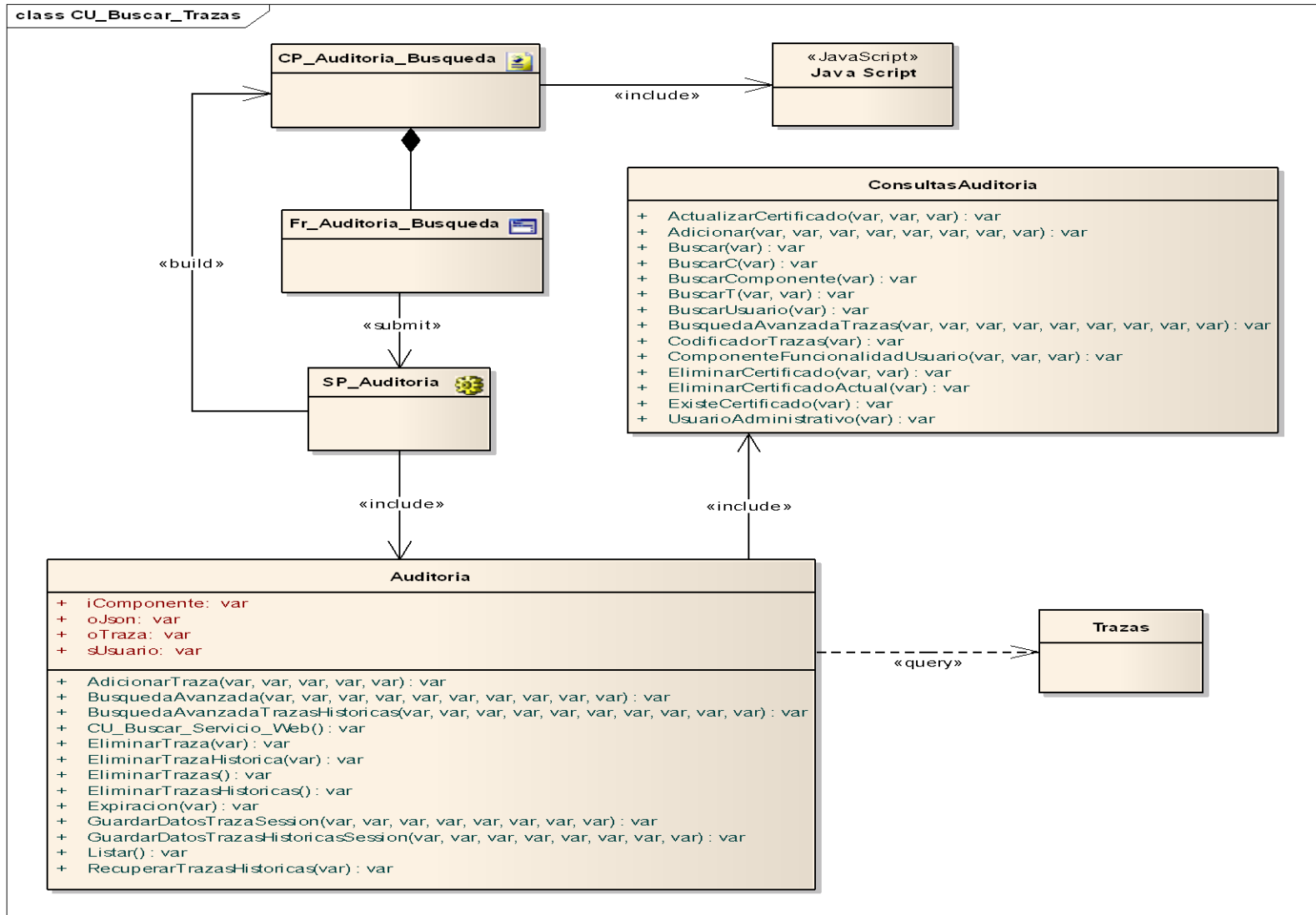
### Anexo 1: Descripción textual CU\_Buscar\_Trazas

<b>Caso de Uso:</b>	<b>CUS16: Buscar_Trazas</b>
<b>Actores:</b>	Administrador General(Inicia)
<b>Resumen:</b>	El caso de uso inicia cuando el usuario hace una búsqueda en la opción Auditoría. La misma se puede realizar por diferentes criterios, el sistema mostrara un listado de las trazas según los datos especificados.
<b>Precondiciones:</b>	El usuario debe tener permiso para realizar una búsqueda de trazas.
<b>Referencias</b>	RF16
<b>Prioridad</b>	Alta
<b>Poscondiciones</b>	El sistema devuelve un listado con el resultado de la búsqueda.

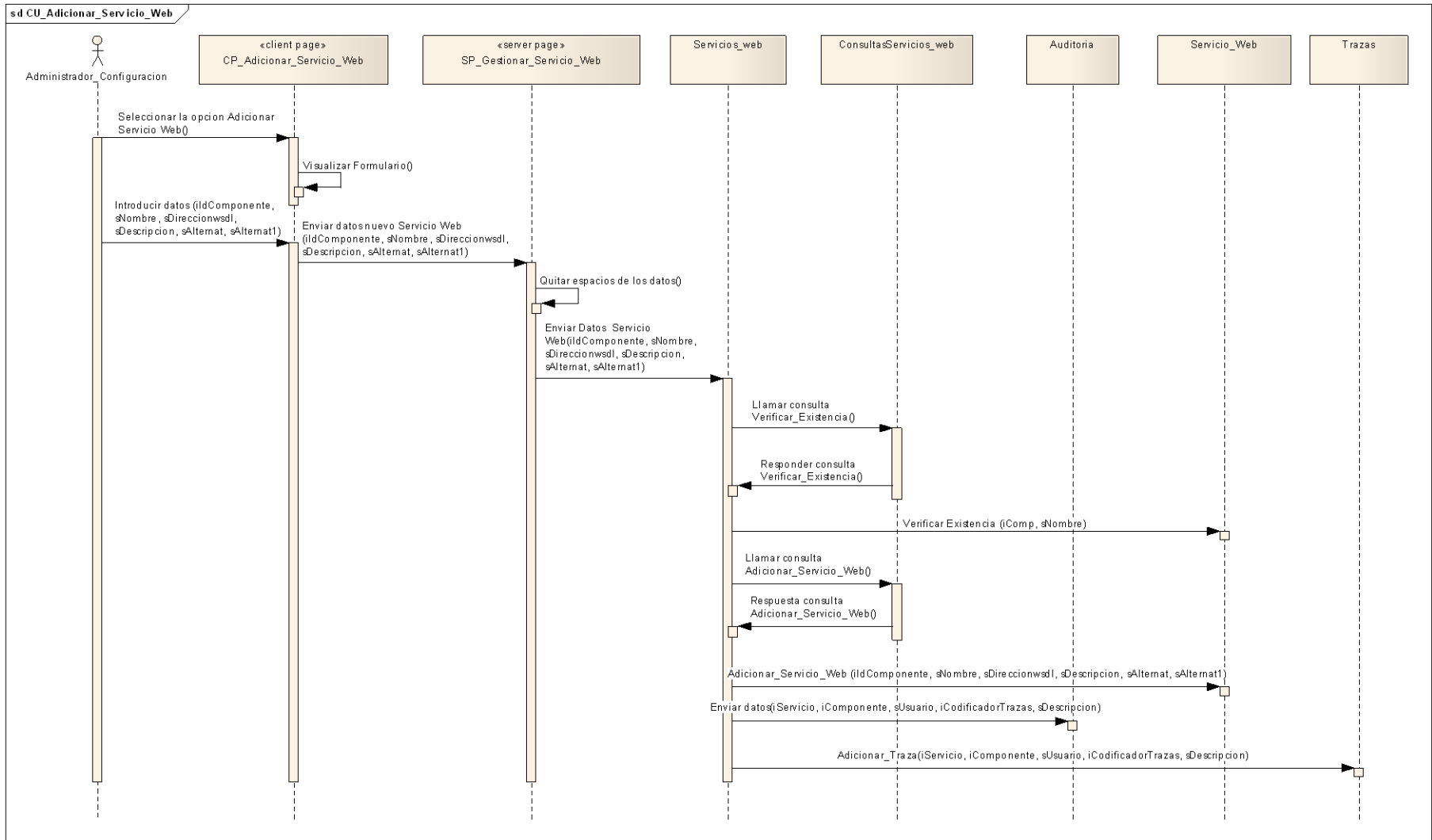
Anexo 2: Diagrama de Clases: CU\_Registrar\_Traza



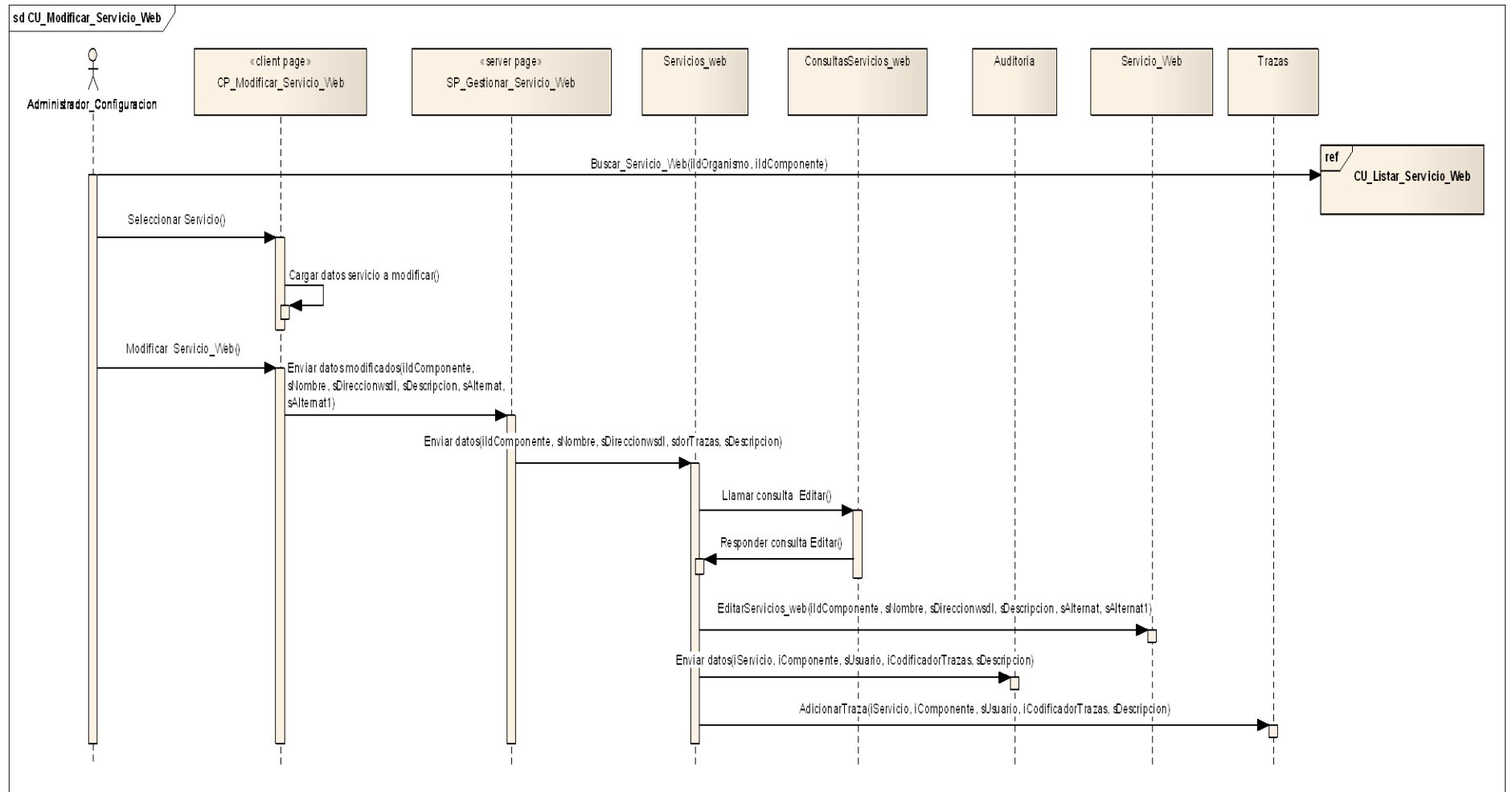
Anexo 3: Diagrama de Clases: CU\_Buscar\_Trazas



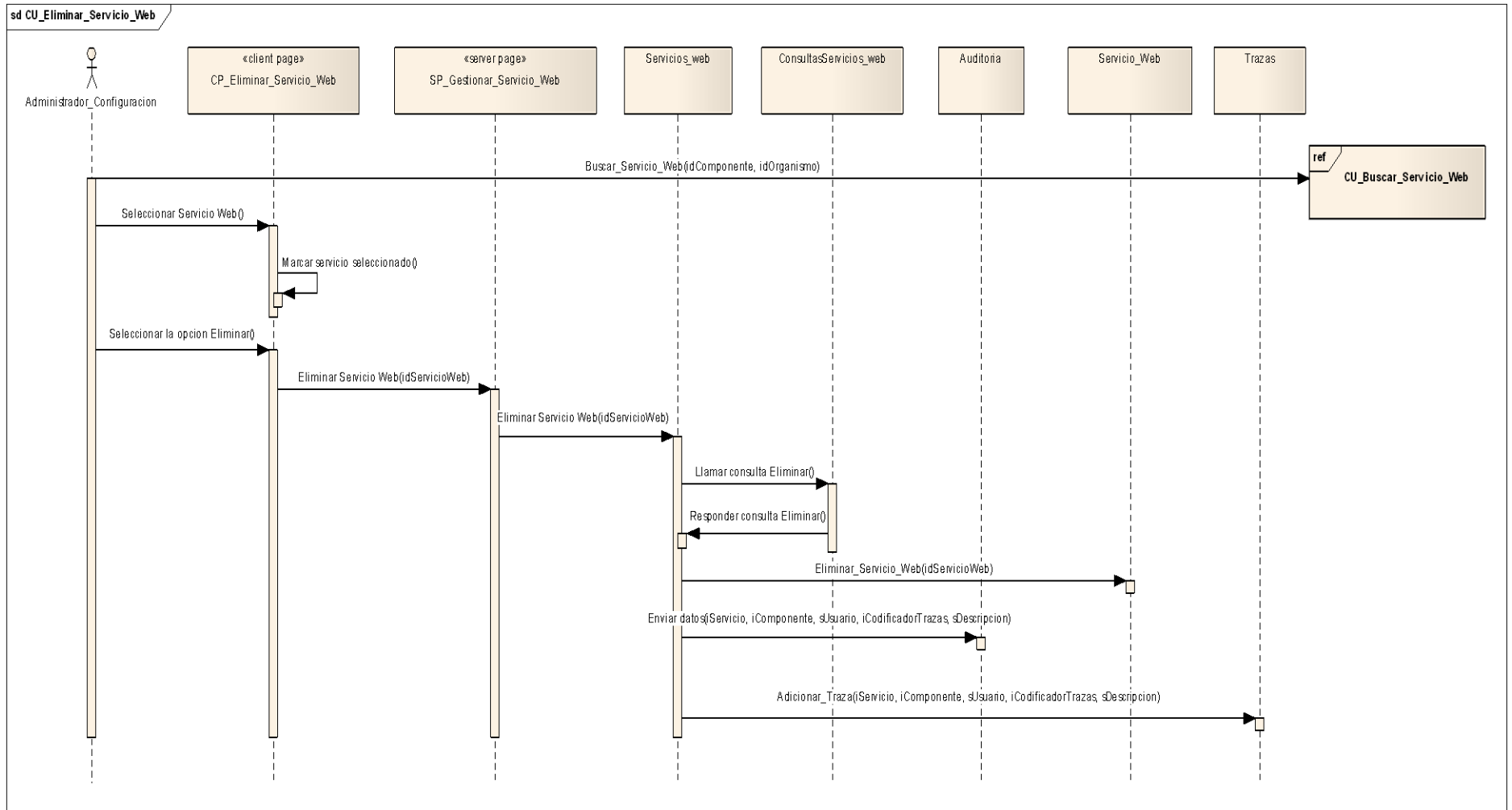
Anexo 4: Diagrama de Secuencia: CU\_Adicionar\_Servicio\_Web



Anexo 5: Diagrama de Secuencia: CU\_Modificar\_Servicio\_Web

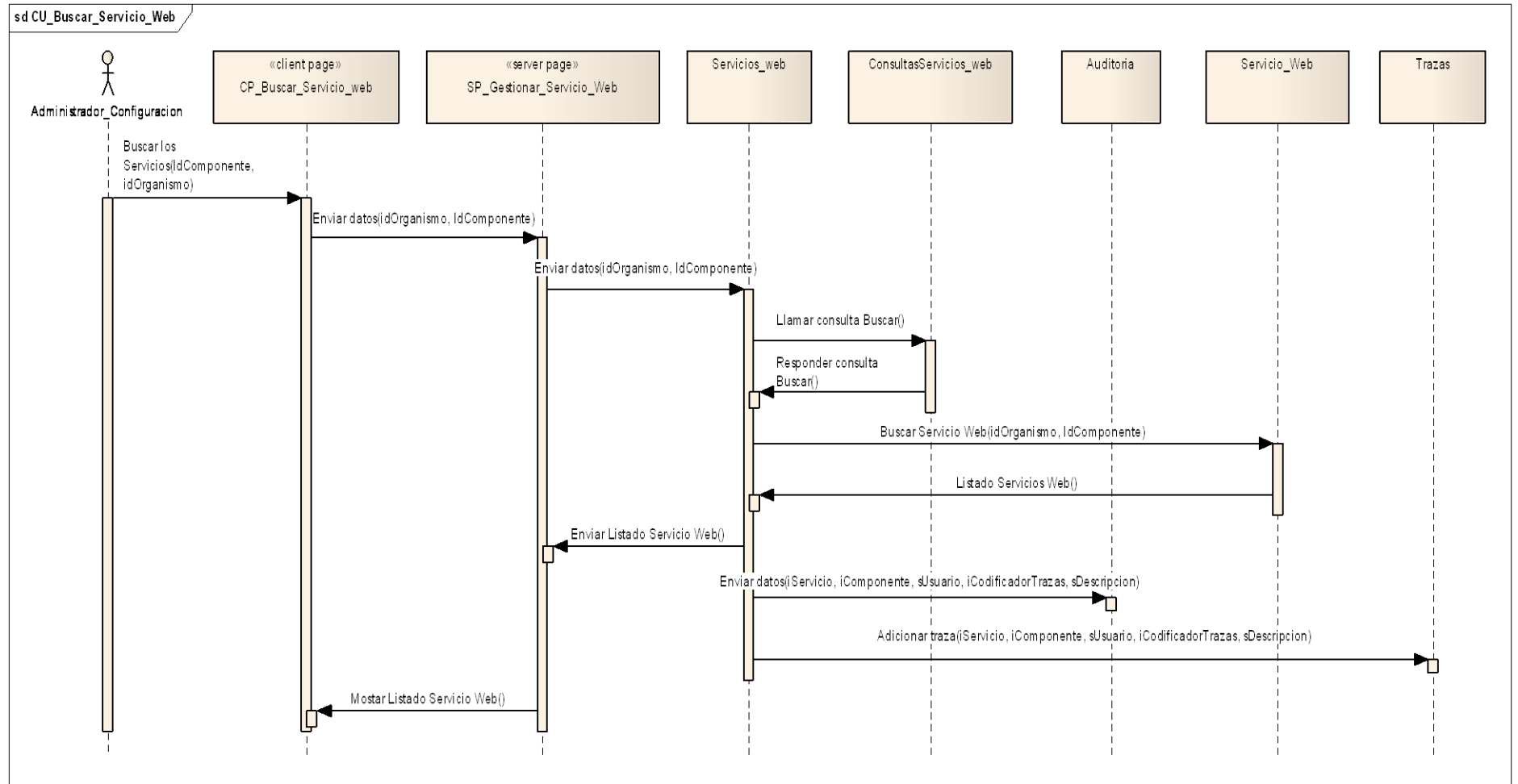


Anexo 6: Diagrama de Secuencia: CU\_Eliminar\_Servicio\_Web

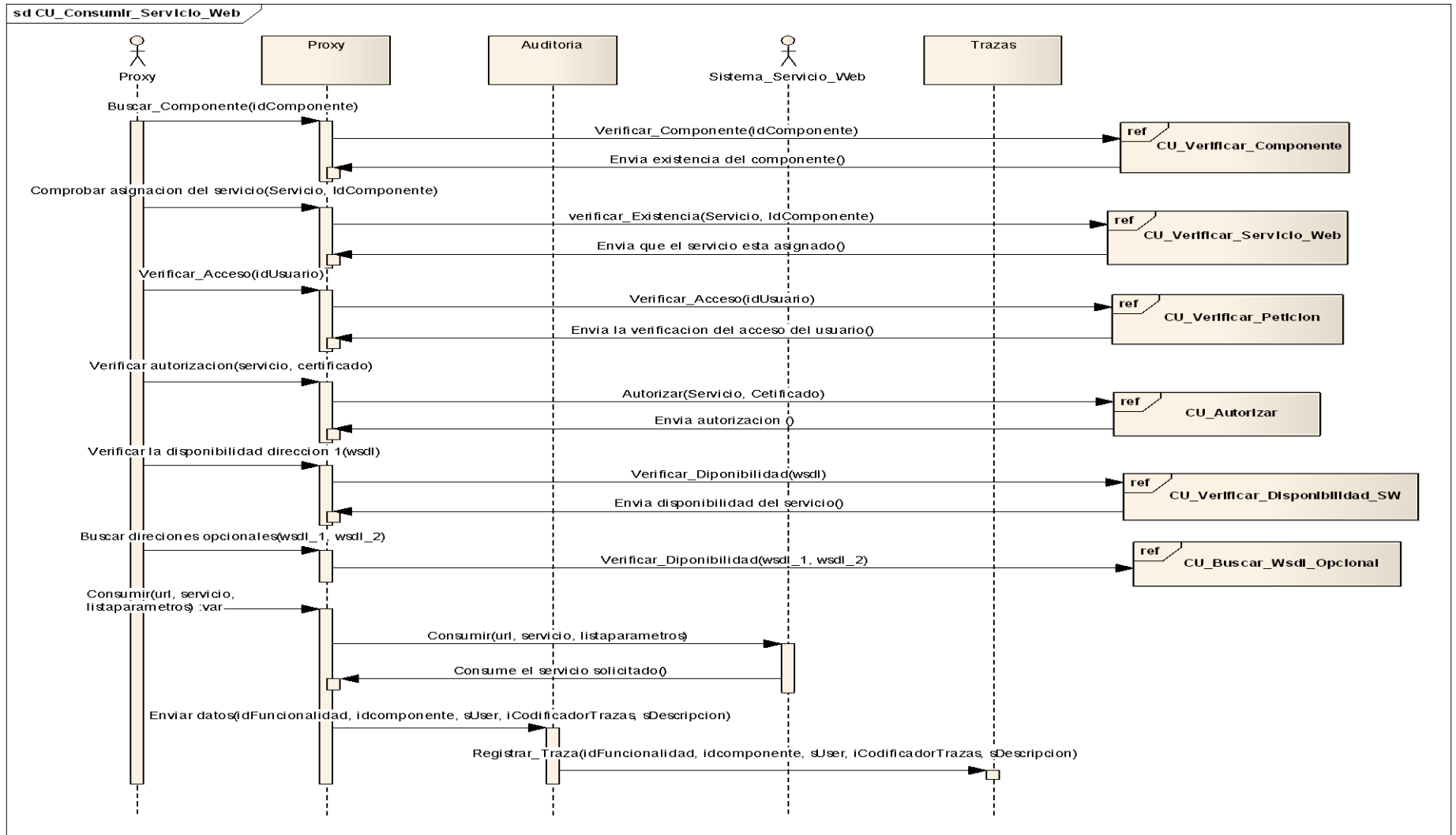




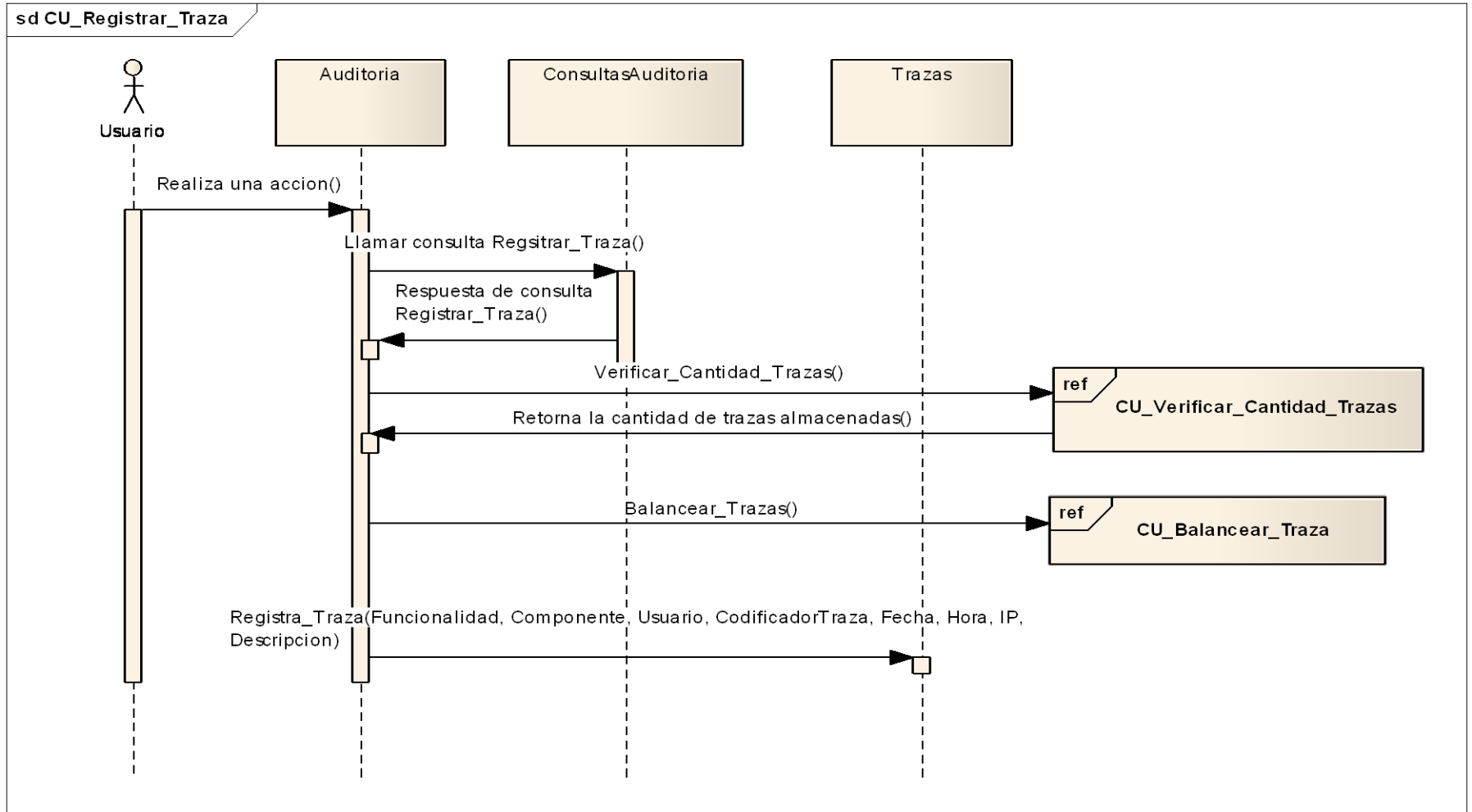
Anexo 7: Diagrama de Secuencia: CU\_Buscar\_Servicio\_Web



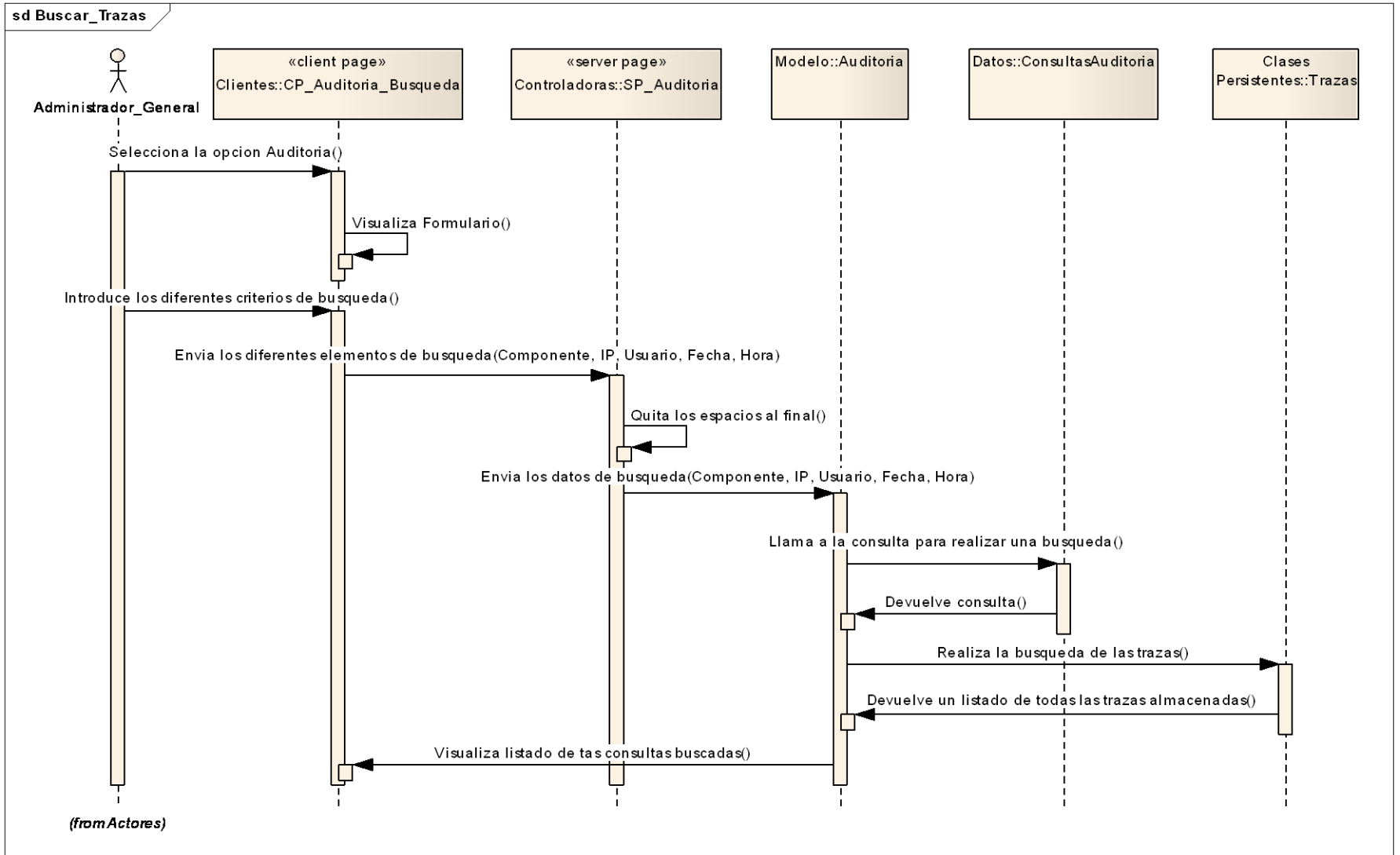
Anexo 8: Diagrama de Secuencia: CU\_Consumir\_Servicio\_Web



Anexo 9: Diagrama de Secuencia: CU\_Registrar\_Traza



Anexo 10: Diagrama de Secuencia: CU\_Buscar\_Trazas



Anexo 11: Ejemplos de Gráficos relacionados con la Auditoría.



Grafico: Estado de la Auditoría

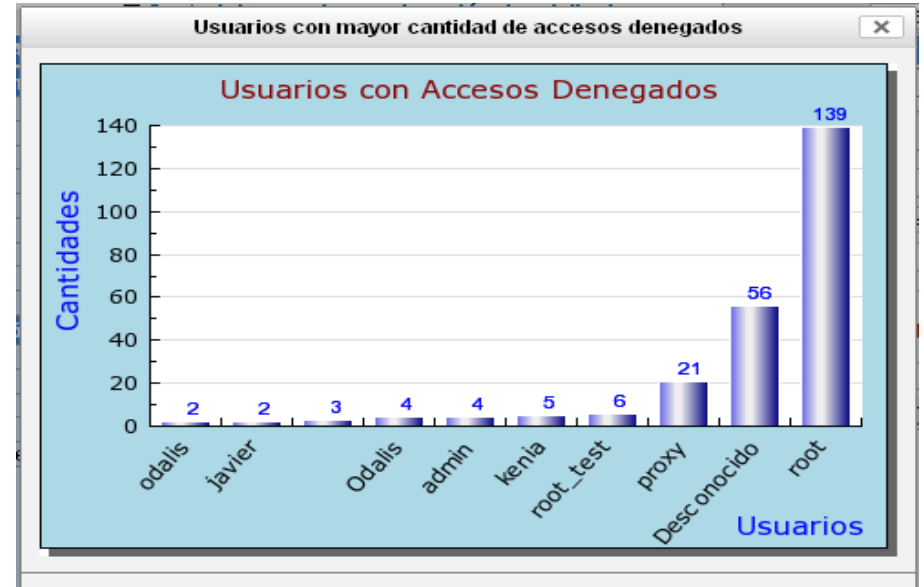



Grafico: Cantidad de Accesos denegados por usuarios

Anexo 12: Página para Gestionar Servicio Web.

**Sistema de Autenticación Autorización y Auditoría.** Lunes, 27 de Abril de 2009 00:13:08



**Componente de Seguridad**  
 Gestión eficiente de requerimientos de seguridad  
 Control de usuarios y asignación de privilegios

Nombre Danisbel Rojas Rios  
 Usuario root  
 Componente <<Seleccione>>

Inicio | Perfil | Ayuda | Salir

**Gestión de Configuración**

- ▶ Países
- ▶ Organismos
- ▶ Niveles
- ▶ Componentes
- ▶ Roles
- ▶ Funcionalidades
- ▶ Tipo de Trazas
- ▶ Servicio Web

**Gestión de Administración**

- ▶ Usuarios
- ▶ Usuarios Históricos
- ▶ Auditoría
- ▶ Auditoría Histórica

>> Inicio >> Componentes >> Adicionar Componentes

Datos Generales | Funcionalidades | Servicios Web | Roles | Roles - Funcionalidades

Gestionar Servicios Web

Nombre  \*

Dirección Oficial Wsld  \*

Wsld Opcionales

Descripción

(\*) Los campos señalados son de entrada obligatoria.

+ Adicionar
✎ Editar
- Eliminar
✖ Cancelar

✔ Terminar

Anexo 13: Interfaz del Módulo de Instalación: Paso 1 de la instalación.

**Sistema de Autenticación Autorización y Auditoría.**
Sábado, 13 de Junio de 2009 11:11:16



**Componente de Seguridad**

- Gestión eficiente de requerimientos de seguridad
- Control de usuarios y asignación de privilegios

**Pasos:**

- Paso 1: ✓

Comprobación Previa
- Paso 2:

Configuración de la Base de Datos
- Paso 3:

Información de Acceso al Sistema
- Paso 4:

Configuraciones
- Paso 5:

Fin de la Instalación

Comprobación Previa: Paso 1 de 5 Siguiente

---

Comprobación Previa para Instalación del Componente de Seguridad:

Para el correcto funcionamiento del Componente de Seguridad es necesario que en su servidor web estén habilitadas estas extensiones, de no estar correctas no se procederá a siguiente paso:

Versión de PHP >= 5.1.4:	✓
Soporte para PostgreSQL:	✓
Soporte para Smarty:	✓
Soporte para LDAP:	✓
Soporte para SOAP:	✓

**Leyenda:**

- ✓ Configuración Correcta.
- ✗ Configuración Incorrecta, debe corregir estas extensiones.

---

Estos son los valores de PHP recomendados para asegurar una completa compatibilidad con el Componente de Seguridad. No obstante el Componente funcionará incluso si estos valores no coinciden con los recomendados.

	<b>Directivas:</b>	<b>Recomendado</b>	<b>Activo</b>
Modo Seguro:		Desactivado	✓
Mostrar Errores:		Desactivado	⚠
Subida de archivos:		Activado	✓
Comillas mágicas en tiempo de ejecución:		Desactivado	✓
Registrar globales:		Desactivado	⚠
Buffer de salida:		Desactivado	⚠
Inicio automático de sesión:		Desactivado	⚠

**Leyenda:**

- ✓ Configuración Recomendada.
- ⚠ Configuración No Recomendada.

Componente de Seguridad. © Copyright 2009 Área Temática SAS

Anexo 14: Interfaz del Módulo de Instalación: Paso 2 de la instalación.

**Sistema de Autenticación Autorización y Auditoría.**
Sábado, 13 de Junio de 2009 11:15:12



**Componente de Seguridad**

- Gestión eficiente de requerimientos de seguridad
- Control de usuarios y asignación de privilegios

---

**Pasos:**

**Paso 1:** ✔  
Comprobación Previa

**Paso 2:** ✔  
Configuración de la Base de Datos

**Paso 3:**  
Información de Acceso al Sistema

**Paso 4:**  
Configuraciones

**Paso 5:**  
Fin de la Instalación

Configuración de la Base de Datos: ↶ Anterior Paso 2 de 5 Siguiente ↷

**Configuración de la Base de Datos del Componente de Seguridad:**

Configurar la Base de Datos del Componente de Seguridad requiere de los siguientes pasos:

1. Escribir el nombre del servidor donde se instalará el Componente de Seguridad.
2. Escribir el nombre del puerto por donde se comunicará con la Base de Datos.
3. Escribir el nombre del usuario PostgreSQL y la contraseña con la que trabajará el Componente de Seguridad.
4. Escribir el nombre de la Base de Datos que usará el Componente de Seguridad.

**Nombre del Servidor:**

**Puerto:**

**Nombre de Usuario:**

**Contraseña:**

**Nombre de la Base de Datos:**

Normalmente es **localhost** ó un nombre de host provisto por su proveedor.

**Puerto** por defecto de postgres.

Un usuario como **root** u otros con **privilegios administrativos**.

Debe indicar la **contraseña** para el usuario entrado.

Escriba el **nombre** de la Base de Datos.

Componente de Seguridad. © Copyright 2009 Área Temática SAS



Anexo 15: Interfaz del Módulo de Instalación: Paso 3 de la instalación.

**Sistema de Autenticación Autorización y Auditoría.**
Sábado, 13 de Junio de 2009 11:17:16



**Componente de Seguridad**

- Gestión eficiente de requerimientos de seguridad
- Control de usuarios y asignación de privilegios

**Pasos:**

**Paso 1:** ✔

Comprobación Previa

**Paso 2:** ✔

Configuración de la Base de Datos

**Paso 3:** ✔

Información de Acceso al Sistema

**Paso 4:**

Configuraciones

**Paso 5:**

Fin de la Instalación

**Configuración de los Datos del Usuario**

Paso 3 de 5 Siguiete 

---

**Cambiar contraseña del Administrador del Componente de Seguridad :**

Para acceder al Componente de Seguridad se necesita de los datos de un usuario administrativo. La Base de Datos creada contiene los datos de este superusuario y sus credenciales por defecto son:

**Usuario: root**  
**Contraseña: 1234567**

Para cambiar estas credenciales entre los datos necesarios en el formulario, de lo contrario continúe al paso Siguiete.

**Cambiar credenciales.**

**Usuario:**


**Contraseña:**

**Confirmar Contraseña:**

Componente de Seguridad. © Copyright 2009 Área Temática SAS

Anexo 16: Interfaz del Módulo de Instalación: Paso 4 de la instalación.

**Sistema de Autenticación Autorización y Auditoría.**
Sábado, 13 de Junio de 2009 11:19:18



**Componente de Seguridad**

- Gestión eficiente de requerimientos de seguridad
- Control de usuarios y asignación de privilegios

**Pasos:**

**Paso 1:** ✔  
Comprobación Previa

---

**Paso 2:** ✔  
Configuración de la Base de Datos

---

**Paso 3:** ✔  
Información de Acceso al Sistema

---

**Paso 4:** ✔  
Configuraciones

---

**Paso 5:**  
Fin de la Instalación

**Configuraciones:** 
↶ Anterior
Paso 4 de 5
Siguiente ↷

---


**Configurar variables globales del sistema:**

Tiempo de expiración del usuario inactivo (segundos):	<input type="text" value="2000"/>	
Cantidad máxima de peticiones:	<input type="text" value="8"/>	
Tiempo máximo de peticiones:	<input type="text" value="10"/>	
Cantidad límite de trazas:	<input type="text" value="24000"/>	
Cantidad límite de trazas históricas:	<input type="text" value="29000"/>	
Cantidad de trazas a balancear la tabla trazas:	<input type="text" value="14000"/>	
Cantidad de trazas a balancear de la trazas historicas:	<input type="text" value="19000"/>	

Componente de Seguridad. © Copyright 2009 Área Temática SAS

## Anexo 17: Interfaz del Módulo de Instalación: Paso 5 de la instalación.

**Sistema de Autenticación Autorización y Auditoría.** Lunes, 15 de Junio de 2009 16:55:04



### Componente de Seguridad

- Gestión eficiente de requerimientos de seguridad
- Control de usuarios y asignación de privilegios

---

**Pasos:**

**Paso 1:** ✔  
Comprobación Previa

**Paso 2:** ✔  
Configuración de la Base de Datos

**Paso 3:** ✔  
Información de Acceso al Sistema

**Paso 4:** ✔  
Configuraciones

**Paso 5:** ✔  
Fin de la Instalación

Fin de la Instalación: ↶ Anterior Paso 5 de 5

---

**La Instalación del Componente de Seguridad ha sido completada:**

El Componente de Seguridad ha sido instalado correctamente, para acceder al mismo puede hacer Clic en el botón "Acceder a Componente".

Para terminar la instalación, por motivos de seguridad **debe borrar el directorio de "instalacion"** que aparece dentro del paquete del Componente de Seguridad.

<< [Acceder al Componente](#) >>

### GLOSARIO DE TÉRMINOS

**DOM:** Siglas de Document Object Model. Es esencialmente un modelo computacional a través de la cual los programas y scripts pueden acceder y modificar dinámicamente el contenido, estructura y estilo de los documentos HTML y XML. Su objetivo es ofrecer un modelo orientado a objetos para el tratamiento y manipulación en tiempo real -o en forma dinámica- a la vez que de manera estática de páginas de internet.

**HTML:** HyperText Markup Language (Lenguaje de Marcas de Hipertexto), es el lenguaje de marcado predominante para la construcción de páginas web.

**MVC:** Modelo Vista Controlador es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

**Plugin:** Es una aplicación informática que interactúa con otra aplicación para aportarle una función o utilidad específica, generalmente muy específica, como por ejemplo servir como controlador en una aplicación, para hacer así funcionar un dispositivo en otro programa. Ésta aplicación adicional es ejecutada por la aplicación principal.

**Proxy:** es un patrón de diseño estructural que proporciona un sustituto o representante de otro objeto para controlar el acceso a este.

**SAAA:** Sistema de Autenticación, Autorización y Auditoría.

**SAS:** Sistema de Apoyo a la Salud.

**XML:** siglas en inglés de Extensible Markup Language (lenguaje de marcas ampliable), es un metalenguaje extensible de etiquetas.

**YUI:** Yahoo User Interface, una serie de bibliotecas escritas en JavaScript, para la construcción de aplicaciones.