



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

**Repositorio de estándares de diseño de interfaz para el
Sistema de Información para la Salud.**

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

AUTORES:

Víctor Vázquez Estrada.

Leonardo Miguel González Arnosó.

TUTORES:

Ing. Alfredo Sánchez Rodríguez.

Ing. Yosvanys Sánchez Corrales.

CIUDAD DE LA HABANA, JULIO DE 2007.

DECLARACIÓN DE AUTORÍA.

Declaramos ser autores de la presente tesis y reconocemos 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 6 días del mes de Julio del año 2007.

Leonardo Miguel
González Arnoso
Autor

Víctor Vazquez
Estada
Autor

Ing. Alfredo Sánchez
Rodríguez
Tutor

Ing. Yosvanys
Sánchez Corrales
Tutor

DATOS DE CONTACTO.

Ing. Alfredo Sánchez Rodríguez: Graduado en el año 2000, de Ingeniería Informática en el "Instituto Superior Politécnico José Antonio Echeverría" CUJAE, Ciudad Habana. Comenzó a trabajar en la Empresa de Servicios Informáticos de Santiago de Cuba. Se traslada a trabajar en Softel en Octubre del año 2003, como prestación de servicio, y meses más tarde es contratado por esta Empresa por tiempo indeterminado como especialista y ocupando el cargo de Arquitecto de Sistemas. Es profesor Instructor categorizado, impartiendo la asignatura de Ingeniería de SW en la Universidad de las Ciencias Informáticas. Ha presentado ponencias en diferentes eventos científicos durante su vida laboral y ha terminado diversos cursos de postgrados relacionados con su actividad. Correo Electrónico: alfredo@softel.cu.

Ing. Yosvanys Sánchez Corales: Profesor graduado de Ingeniero en Informática en el año 2005 en la CUJAE. Ha impartido las asignaturas Programación 3 e Inteligencia Artificial, Forma parte del proyecto de Atención Primaria de Salud de la facultad 7 y actualmente cursa diplomados para la matrícula en la maestría GPI (Gestión de Proyectos Informáticos). Correo Electrónico: yscorales@uci.cu.

El hombre debe transformarse al mismo tiempo que la producción progresa; no realizaríamos una tarea adecuada si fuéramos tan sólo productores de artículos, de materias primas y no fuéramos al mismo tiempo productores de hombres.

Ernesto Che Guevara

AGRADECIMIENTOS.

A la Revolución, a Fidel y a la Universidad de las Ciencias Informáticas, por dejarnos formar parte de este gran proyecto.

A nuestros tutores Alfredo Sánchez Rodríguez y Yosvanys Sánchez Corrales por su confianza.

A María Lilia Rdguez., Lisandra Pérez, Leonardo Glez. Glez., Johander León, Yoiler Frómeta, Yusniel Brocard, Carlos Acosta, Alexander Paneque, Liusvani Suárez, Danieskis Rodríguez, Leinier Lubian y Luis Laguna, muchas gracias por haberse unido al equipo incondicionalmente y habernos ayudado tanto.

A los compañeros de Softel en especial a Mirna Cabrera por confiar en nosotros y contribuir a nuestra formación como profesionales.

A mi hijo, por ser lo que más quiero en esta vida.

A mis padres, por quererme tanto, darme esa confianza en todo y enseñarme a caminar por la vida.

A mi esposa por entregarme siempre su amor y cariño, estar siempre a mi lado, en los buenos y en los malos momentos.

A mis queridos abuelos Orquídea, María Cristina, Fernando y Regla.

A mis tías lindas Lourdes, Rebeca, por ser tan buenas y preocupadas.

A Eliserio por ser tío y amigo a las vez.

A mis primos Helwin, Javier, Delvis, Gressi, Yadira y a todos mis familiares que de una forma u otra han influido en mi vida. A mi suegro Juan por su confianza y por ser mucho más que mi suegro.

A todos, muchas gracias.

A mis padres por su amor, apoyo, confianza, respeto a mis opiniones, por lo todo lo que han hecho por mí en la vida y sobre todas las cosas por ser tan buenos padres.

A mi tía Nidia Arnosó, Alberto Juan y Albertico por darme la posibilidad de contar con ellos para todo y darme la posibilidad de tener otra familia.

A Yaimí Márquez por haberme guiado durante tanto tiempo.

A todos mis amigos y compañeros, que de una forma u otra siempre me apoyaron.

Leonardo Miguel

Víctor

DEDICATORIA.

... A mi hijo, a quien esta dedicada toda mi vida.

... A mis padres, ¿Qué hiciera sin ustedes?

...A mi esposa, te amo linda.

...A mis Abuelos, en especial a Fernando para que se sienta orgulloso siempre de su nieto que lo adora.

...A un gran amigo, que aunque no se encuentra entre nosotros, siempre quiso ver este día, gracias Galindo.

...A Yordany, Ariel, Alnair, Owen, por ser amigos y hermanos a las vez, a todos mis amistades UCI y de Villa Clara.

Víctor

...A mi madre Silvia Arnosó, la persona más linda y pura del mundo. Para ella es esta tesis y todo lo que haga en mi vida.

...A mi padre Abel González, mi ejemplo y para hacerle sentir por mí el mismo orgullo que yo siento por él.

...A mi hermanita Diané Chávez y a mis sobrinos Toño y Angelito.

...A mi abuela Roselina Gómez (Ñaña) a quien adoro.

...A mi abuelo Leonardo Alfredo quien me enseñó hacer tantas cosas desde niño.

...A mis abuelos Elba Rodríguez y Miguel González.

Leonardo Miguel

RESUMEN.

La presente investigación tiene como objetivo general, implementar una aplicación Web que contenga un repositorio de estándares de diseño de interfaz para la capa de presentación generalizables para las aplicaciones del SISalud. Esto ha sido necesario, debido a la falta de definición de estándares; la capa de presentación que se le ofrecía al usuario final, no poseía un diseño robusto y homogéneo por lo que existía una gran inconsistencia entre los distintos entornos de las aplicaciones desarrolladas.

Hasta el momento no se había pensado en una solución que permitiera el acceso de todo el equipo de desarrollo a un repositorio que facilitara de forma inmediata la reutilización de soluciones, evitando de esta forma la inconsistencia entre entornos y las funcionalidades asociadas a estos. Para la implementación de esta solución automatizada se utilizaron tendencias y tecnologías actuales, como por ejemplo PHP, XML, XSLT, XML Web Services, MySQL, XHTML, JavaScript y herramientas como Rational Rose Enterprise Edition 2003, NuSphere, Stylus Studio, Dreamweaver 8, entre otras.

Con la implementación de esta aplicación web, se pone a disposición de todo el equipo de desarrollo del proyecto APS, definiciones únicas para los diferentes entornos, que tratan el aspecto y las funcionalidades de la interfaz de forma compacta, presentando dicha información en forma de ejemplos prácticos y compartiendo su código fuente a los desarrolladores, para de esta forma acortar el tiempo de desarrollo de los componentes, estandarizar la capa de presentación y las funcionalidades asociadas a esta.

ÍNDICE.

AGRADECIMIENTOS.....	I
DEDICATORIA.	II
RESUMEN.....	III
ÍNDICE.	IV
INTRODUCCIÓN.	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.....	4
1.1 Introducción.	4
1.2 Sistema de Información para la Salud (SISalud).....	4
1.3 Antecedentes.....	5
1.4 Situación problemática y problema a resolver.....	5
1.5 Objeto de Automatización e información manipulada.	6
1.6 Tendencias y tecnologías a utilizar.....	7
1.6.1 Aplicaciones web.	7
1.6.2 XML/WebServices. Arquitectura Orienta a Servicios.....	7
1.6.3 Entornos Distribuidos. Modelo Cliente Servidor.....	8
1.6.3.1 Modelo Cliente-Servidor de tres Capas (Three Tier).....	9
1.6.3.2 Servidor Web Apache.....	10
1.6.3.3 Arquitectura Basada en Componentes (CBA).	10
1.6.4 Lenguajes de Programación Web.	10
1.6.4.1 PHP.....	10
1.6.4.2 XSLT.....	11
1.6.4.3 JavaScript.	11
1.6.4.4 XHTML.....	12
1.6.5 Hojas de Estilo en Cascada (Cascading Style Sheets, CSS).....	12
1.6.6 Sistemas de Gestión de Bases de Datos (SGBD).....	13
1.6.7 Metodología de desarrollo de software.	13
1.6.7.1 UML.	14
1.6.7.2 Rational Rose.....	15
1.6.8 Plataforma de Servicios (PLASER).....	15

1.7	Conclusiones	16
CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA.		17
2.1	Introducción.	17
2.2	Modelo del Dominio.	17
2.2.1	Conceptos fundamentales.	17
2.2.2	Modelo del Dominio.	19
2.3	Propuesta del sistema.	19
2.3.1	Especificación de los requisitos del software.	21
2.3.1.1	Requisitos funcionales.	21
2.3.1.2	Requisitos no funcionales.	21
2.3.1.2.1	Usabilidad.	21
2.3.1.2.2	Apariencia o interfaz externa.	22
2.3.1.2.3	Rendimiento.	22
2.3.1.2.4	Software.	22
2.3.1.2.5	Hardware.	22
2.3.1.2.6	Portabilidad.	22
2.3.1.2.7	Seguridad.	22
2.3.1.2.8	Confiabilidad.	22
2.3.1.2.9	Ayuda y documentación en línea.	23
2.4	Modelo de Casos de Uso del Sistema.	23
2.4.1	Definición de los actores.	23
2.4.2	Definición de Casos de Uso.	24
2.4.3	Diagrama de Casos de Uso.	28
2.4.4	Descripción textual de los Casos de Uso.	29
2.5	Conclusiones.	53
CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL SISTEMA.		54
3.1	Introducción.	54
3.2	Modelo de Análisis.	54
3.2.1	Diagramas de Clases del Análisis.	54
3.3	Modelo de Diseño.	59
3.3.1	Justificación del uso de patrones.	59

3.3.2	Definición de estructura del diseño.	60
3.3.3	Diagramas de clases del diseño.....	61
3.3.4	Diagramas de interacción.	66
3.3.5	Descripción de las clases y atributos.	75
3.4	Diseño de Base de Datos.....	80
3.4.1	Diagrama de clases persistentes.	80
3.4.1	Modelo de datos.	81
3.4.2	Descripción de tablas y atributos.....	82
3.5	Conclusiones.	84
CAPÍTULO 4. IMPLEMENTACIÓN.....		85
4.1	Introducción.	85
4.2	Justificación de integración con otros sistemas.....	85
4.3	Modelo de Implementación.	86
4.3.1	Diagrama de Componentes.....	86
4.3.2	Diagrama de Despliegue.	87
4.4	Descripción de los métodos (agentes o métodos más complejos).	87
4.5	Estándares de diseño de interfaz.....	88
4.6	Estándares de Codificación.....	91
4.7	Tratamiento de excepciones.	94
4.8	Conclusiones.	94
CONCLUSIONES.		95
RECOMENDACIONES.....		96
REFERENCIAS BIBLIOGRÁFICAS.		97
BIBLIOGRAFÍA.		98
ANEXOS.		100
GLOSARIO DE TÉRMINOS.....		122

INTRODUCCIÓN.

El país enfrenta el reto de automatizar la sociedad haciendo uso de las Tecnologías de Informática y las Comunicaciones. Al igual que otras ramas el Ministerio de Salud Pública desarrolla múltiples tareas para lograr esta informatización, surgiendo a raíz de ello el Sistema de Información para la Salud (SISalud). Este consta de varios módulos que se le irán incorporando al Sistema en la medida en que sean implementados por el grupo de trabajo del Proyecto APS, conjuntamente con la empresa SOFTEL, empresa encargada de llevar a cabo el diseño de estos módulos y que utiliza una estrategia nunca antes concebida en un proceso de desarrollo de software en el país.

En el proceso de desarrollo de los componentes del Sistema de Información para la Salud, se realiza el diseño de interfaz y sus funcionalidades asociadas en la capa de presentación según definiciones individuales del grupo de trabajo en cada módulo. Esto ocurre debido a la falta de definición de estándares aplicables a los distintos entornos y a que no existe una aplicación, única y actualizada, que contenga un repositorio de estos estándares, que faciliten su reutilización, agilizando y estandarizando el desarrollo de los componentes. Esto trae consigo una situación de inconsistencia entre entornos y que la capa de presentación que se le ofrece al usuario final no posea un diseño robusto y homogéneo.

Con relación al tema de estandarización del diseño de interfaz para la capa de presentación del Sistema de Información para la Salud (SISalud), se han hecho ya intentos para lograr dicho fin, un ejemplo de esto es la construcción de Guías de Estilo, demostrándose que no tienen utilidad real y que la documentación caduca por desuso con lo que se vuelve a una situación similar a la de partida.

Por ello es necesario implementar una aplicación web que sirva de repositorio para garantizar parte del proceso relacionado con el diseño de interfaz de usuario y sus funcionalidades asociadas en la capa de presentación de los módulos del Sistema de Información para la Salud.

Con el desarrollo de esta aplicación se espera identificar estándares que puedan derivar en componentes de interfaz reutilizables para su uso por los diferentes equipos de desarrollo del proyecto APS. Estos permitirán abstraer al programador de algunas tareas de diseño de interfaces, sin dejar de atrás los aspectos visuales y estéticos. Estos recursos que contienen los estándares predefinidos por el proyecto

deben ser puestos a disposición de los equipos de programación a través de un repositorio único y actualizado.

Dada la situación anterior el **problema** radica en: ¿Cómo estandarizar el diseño de interfaz para la capa de presentación del Sistema de Información para la Salud?

El **objeto de estudio** se centra en el proceso de informatización del Sistema de Información para la Salud.

El **campo de acción** apunta al proceso de informatización que define los estándares de diseño de interfaz y sus funcionalidades asociadas en la capa de presentación en el Sistema de Información para la Salud.

Para dar solución a este problema se propone como **objetivo general**: Implementar una aplicación Web que contenga un repositorio de estándares de diseño de interfaz para la capa de presentación generalizables para las aplicaciones del Sistema de Información para la Salud.

Definiendo las siguientes **tareas**:

Estudiar el entorno de presentación actual utilizado en los componentes implementados en el proyecto APS del Sistema de Información para la Salud.

Realizar un estudio de las tendencias y tecnologías actuales para realizar el proceso de implementación de dicha solución automatizada como por ejemplo PHP, XML, XSLT, XML Web Services, MySQL, XHTML, JavaScript.

Asimilar la Plataforma de Servicio (PlaSer), framework de la empresa SOFTEL, sobre la cual desarrollará el proceso de implementación.

Definir estándares para el diseño de capa de presentación de la aplicación a implementar.

Obtener demos de estándares de diseño de interfaz de capa de presentación en el Sistema de Información para la Salud.

Implementar el repositorio de estándares de diseño de interfaz para la capa de presentación de las aplicaciones del SISalud.

El presente trabajo está formado por cuatro capítulos, cuyos contenidos serán mencionados a continuación.

En el primer capítulo se expone la fundamentación teórica del tema a tratar. Muestra un estado del arte del tema tratado y un análisis detallado de los principales conceptos y tecnologías que son consideradas para el desarrollo del sistema.

El segundo capítulo se hace énfasis la descripción de las características del sistema. Se definen conceptos que se agrupan en un Modelo de Dominio. Se enumeran además los requisitos funcionales y no funcionales, los casos de uso del sistema, el Diagrama de Casos de Uso y la descripción de cada uno de los casos de usos definidos.

En el tercer capítulo se modelan los artefactos relacionados con el flujo de trabajo Análisis y Diseño. Se muestran los diagramas de clases del análisis, luego en la sección correspondiente al diseño se exponen los detalles relacionados con el diseño del sistema propuesto, utilizándose para su modelado los diagramas de clases de los casos de uso del sistema. Además se presenta el diagrama de clases persistentes y el modelo de datos que es la base para construir la base de datos que soportará toda la información del sistema conjuntamente con la descripción de sus tablas.

En cuarto capítulo y final se trazan las estrategias de integración con otros sistemas. Se generan los diagramas de Componentes y de Despliegue. Se describen algunos de los métodos del negocio y se definen los estándares de diseño de interfaz, codificación y tratamiento de excepciones que serán utilizados.

Al finalizar el documento se exponen las conclusiones del trabajo, recomendaciones propuestas, bibliografía utilizada y anexos, los cuales aportan una mayor ilustración del trabajo realizado.

Cada capítulo es iniciado por una breve introducción donde se dan a conocer los temas que se abordan en el mismo y finaliza con las conclusiones, que describen los resultados obtenidos en el mismo.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.

1.1 Introducción.

El presente capítulo muestra un estudio del estado del arte del tema tratado y un análisis detallado de los principales conceptos y tecnologías que se consideran para el desarrollo del sistema. Se describen los sistemas integrados y distribuidos, el uso de las arquitecturas existentes y la metodología a utilizar para el análisis y diseño del sistema teniendo en cuenta las facilidades que puede aportar al trabajo. Se hace un estudio de algunos de los diferentes lenguajes de programación y de los Sistemas de Gestión de Bases de datos (SGBD) más usados; definiéndose los más adecuados para el sistema.

1.2 Sistema de Información para la Salud (SISalud).

El Sistema de Información para la Salud es una aplicación web que posibilita la integración de sistemas o componentes, cada uno de ellos desarrolla sus procesos y al estar integrados pueden ser accedidos por el resto y facilitar el acceso e intercambio de información. Su infraestructura esta formada por un Registro Informatizado de Salud (RIS), el Sistema Informatizado de Atención Primaria (SIAP) y el Sistema Informatizado de Gestión Hospitalaria (SIGH).

El RIS lo integran el Registro no Médico Informatizado de Salud (RNMIS) y el Registro Médico Informatizado de Salud (RMIS).

El SIAP contempla los módulos de la Atención Primaria de Salud e integra diversos subsistemas, como el Registro de Actividades Diarias del Equipo Básico de Salud (EBS) y el Registro de Población. Ambos constituirán una nueva herramienta para la transformación de servicios que se brindan este nivel.

El SIGH congrega los módulos que pertenecen al nivel de atención secundaria.

1.3 Antecedentes.

Se han hecho varios intentos con el objetivo de lograr la estandarización del diseño de interfaz para la capa de presentación del Sistema de Información para la Salud (SISalud), un ejemplo de esto es la construcción de guías de estilo, las cuales no son más que documentos que recogen normativas, estándares y patrones básicos relacionados con el aspecto de una interfaz para su aplicación en el desarrollo de nuevas pantallas dentro de un entorno concreto; son creadas inicialmente como documentos voluminosos que se basan en la apariencia de la interfaz de un sistema. Su problema más grave es su falta de usabilidad, puesto que están pensadas desde una perspectiva de diseño solamente y no tienen en cuenta las necesidades de sus verdaderos destinatarios: “Diseñadores y Programadores de Capa de Presentación”. Otra de las limitantes que presentan estas guías son:

- Falta de adecuación a los métodos de desarrollo: El desarrollador no tiene tiempo para leer ni asimilar una documentación que además de ser voluminosa, le resulta ajena.
- Falta de mantenimiento consistente: No existe una política de mantenimiento del manual con una visión integradora de todo el proceso de desarrollo.
- Falta de apoyo: La Guía se publica sin acciones de promoción, formación y apoyo. La documentación caduca por desuso con lo que se vuelve a una situación similar a la de partida.
- No tienen utilidad real: No se promueve la reutilización de soluciones.[1]

1.4 Situación problemática y problema a resolver.

En el proceso de desarrollo de los componentes del Sistema de Información para la Salud, se realiza el diseño de interfaz y sus funcionalidades asociadas en la capa de presentación según definiciones individuales del grupo de trabajo en cada módulo, ocurriendo esto en consecuencia de la falta de definición de estándares aplicables a los distintos entornos y que no existe una aplicación, única, centralizada y actualizada, que contenga un repositorio de estos estándares facilitando su reutilización, agilizando y estandarizando el desarrollo de los componentes. Esto trae consigo una situación de inconsistencia entre subsistemas, ya que la capa de presentación que se le ofrece al usuario final no posee un diseño robusto y homogéneo tanto en apariencia como en funcionamiento.

Por ello es necesario implementar una aplicación web que servirá de repositorio para garantizar parte del proceso relacionado con el diseño de interfaz de usuario y sus funcionalidades asociadas, en la capa de presentación de los módulos del Sistema de Información para la Salud.

Con el desarrollo de esta aplicación se espera definir estándares que puedan derivar en componentes de interfaz reutilizables para su uso por los diferentes equipos de desarrollo del proyecto APS, permitiendo al desarrollador abstraerse de tareas relacionadas con el diseño de interfaz, teniendo en cuenta puntos visuales y estéticos.

Dada la situación anterior el problema radica en ¿cómo estandarizar el diseño de interfaz para la capa de presentación del Sistema de Información para la Salud?

1.5 Objeto de Automatización e información manipulada.

Con el desarrollo de esta aplicación se desea automatizar y estandarizar lo referente al diseño de interfaz de usuario de los componentes del SISalud. Para ello además de desarrollar una aplicación web que sirva de repositorio se deben definir y construir estándares para su posterior publicación y reutilización.

Los estándares se mostrarán en forma de ejemplos prácticos, es decir demostraciones (demos) con sus respectivas funcionalidades incluidas, para su asimilación de forma rápida. (*Anexo1*).

Además de los estándares el repositorio brindará un servicio de descarga de recursos (herramientas, tutoriales, etc.) que son necesarios en el proceso de diseño e implementación de la capa de presentación.

Este sistema debe permitir que sólo acceda a él el personal autorizado del proyecto y tener disponible en todo momento de la información que contiene. Todo esto orientado a garantizar el correcto funcionamiento del diseño de interfaz de usuario.

1.6 Tendencias y tecnologías a utilizar.

1.6.1 Aplicaciones web.

Una aplicación web es un sistema web que permite a los usuarios ejecutar lógica de negocio a través de un navegador, o lo que es lo mismo modificar el estado del negocio.

Las aplicaciones web utilizan las tecnologías existentes para generar contenidos dinámicos y permitir a los usuarios del sistema modificar la lógica del negocio en el servidor. Si no existe lógica de negocio en el servidor, el sistema no puede ser considerado una aplicación Web.

1.6.2 XML/WebServices. Arquitectura Orienta a Servicios.

XML (Extensible Markup Lenguaje) está diseñado para representar datos estructurados y se ha convertido en un formato estándar en Internet; es un metalenguaje para definir otros lenguajes de marcados adecuados a un uso en específico, es además la base de los servicios Web. Se emplea principalmente para separar el contenido de la presentación de forma total.

Los servicios Web XML actúan de forma independiente y además permiten que las aplicaciones compartan información e invoquen funciones de otras aplicaciones independientemente del sistema operativo o la plataforma en que se ejecutan y los dispositivos utilizados para obtener acceso a ellos, o sea, son rutinas en Internet actuando como catalizadores de transacciones vía Web. Los Servicios Web usan SOAP (Simple Object Access Protocol) como protocolo de transporte estándar por su simplicidad, se puede identificar un mensaje SOAP como un documento XML conformado por una envoltura “envelope” obligatoria, un encabezamiento “header” opcional y un cuerpo “body”, también obligatorio [SOAP-W3C], y la forma de acceder a ellos es a través del WSDL (Web Services Description Languages). Estos servicios deben publicar una interfaz que funja como un contrato de servicio y donde se describan cada una de las funciones que provee además de las funciones que estos ofrecen, como realizar el intercambio de mensajes, especificar el contenido de una petición y el aspecto de la respuesta en una notación inequívoca. Además de describir el contenido de los mensajes, WSDL define dónde está disponible el servicio y qué protocolo de comunicaciones utilizar para hablar con el servicio. Esto significa que el

archivo WSDL define todo lo necesario para escribir un programa que interactúe con un Servicio Web. [2]
[3] [4]

Cada vez más las empresas exigen aplicaciones más complejas, con menos tiempo y presupuesto que antes. Crear estas aplicaciones, requiere en muchos casos de funcionalidades ya antes implementadas como parte de otros sistema. SOA (Service Oriented Architecture) nace como una estrategia de integración, expone servicios con funcionalidad bien definida a la aplicación que la requiera. De esta manera, una aplicación final simplemente orquesta la ejecución de un conjunto de estos servicios, añade su lógica particular y le presenta una interfaz al usuario final.

Exponer procesos de negocio como servicios es la clave a la flexibilidad de la arquitectura. Esto permite que otras piezas de funcionalidad (incluso también implementadas como servicios) hagan uso de otros servicios de manera natural, sin importar su ubicación física. Así un sistema evoluciona con la adición de nuevos servicios y su mejoramiento. Donde cada servicio evoluciona de una manera independiente. La Arquitectura Orientada a Servicios resultante, define los servicios de los cuales estará compuesto el sistema, sus interacciones, y con qué tecnologías serán implementados. Las interfaces que utiliza cada servicio para exponer su funcionalidad son gobernadas por contratos, que definen claramente el conjunto de mensajes soportados, su contenido y las políticas aplicables.

1.6.3 Entornos Distribuidos. Modelo Cliente Servidor.

La arquitectura Cliente-Servidor, es una forma de dividir y especializar programas y equipos de cómputo a fin de que la tarea que cada uno de ellos realiza se efectúe con la mayor eficiencia, y permita simplificar las actualizaciones y mantenimiento del sistema, en una arquitectura monolítica no hay distribución; los tres niveles tienen lugar en el mismo equipo, en el modelo cliente-servidor, en cambio, el trabajo se reparte entre dos ordenadores. [5]

Su principal ventaja es que reduce el tráfico de red considerablemente, puesto que el cliente se conecta al servidor cuando es estrictamente necesario, obtiene los datos que necesita y cierra la conexión dejando la red libre.

1.6.3.1 Modelo Cliente-Servidor de tres Capas (Three Tier).

El modelo cliente-servidor propone un ambiente para el desarrollo de aplicaciones de avanzada. Usando un modelo de tres capas, el sistema es dividido en datos, negocio y presentación. La idea de esta arquitectura está basada principalmente en la capacidad de estabilidad que nos ofrece.

Datos: La capa de datos tiene como misión la administración de la información que maneja el sistema. Esto incluye el almacenamiento, la actualización y la consulta de todos los datos contenidos en el sistema, es por esto que contiene a la base de datos y las clases de acceso a la misma.

Negocio: El comportamiento de la aplicación es definido por los componentes que modelan la lógica de negocio. Estos componentes reciben las acciones a realizar a través de la capa de presentación, y llevan a cabo las tareas necesarias utilizando la capa de datos para manipular la información del sistema. Los servicios de esta capa son encapsulados en 2 paquetes, las Entidades del Negocio, que representan objetos que van a ser manejados o consumidos por toda la aplicación, y Lógica del Negocio que contienen las clases principales relacionadas con dicho negocio.

Presentación: La capa de presentación representa la parte del sistema con la que interactúa el usuario. En una aplicación Web, generalmente la capa de presentación se divide en dos: el lado servidor y el lado cliente. En el lado servidor ocurre toda la interacción con la lógica de negocio. En el lado cliente se presenta la interfaz generada en el servidor al usuario, de forma tal que éste pueda trabajar con ella.

Como se puede observar esta arquitectura permite que tanto la interfaz de usuario, las reglas de negocios y el motor de datos se conviertan en entidades separadas unas de otras, manteniendo bien definidas las interfaces que cada una de estas exponen para comunicarse con la otra. Los componentes y servicios creados según este modelo se pueden compartir y reutilizar, por lo que las aplicaciones que lo usan, alcanzan una mayor capacidad de crecimiento. [6]

1.6.3.2 Servidor Web Apache.

Su coste gratuito, gran fiabilidad y extensibilidad le convierten en una herramienta potente y muy configurable.

Es un software libre, porque sus desarrolladores defienden la teoría de que las transmisiones usando la red deben estar en las manos de todos, y que las compañías de software deben hacer el dinero ofertando servicios con valor añadido tales como módulos especializados, soportes, entre otros, y no siendo dueñas de un protocolo. Así, el proyecto de crear una implementación robusta con referencia absolutamente libre para quien lo quiera usar es un buen paso para evitar la propiedad sobre los protocolos.

1.6.3.3 Arquitectura Basada en Componentes (CBA).

Su objetivo es construir aplicaciones complejas mediante el ensamblado de módulos (componentes), que han sido previamente diseñados por otras personas a fin de ser reusados en múltiples aplicaciones. Cada componente describe de forma completa las interfaces que ofrece y las interfaces que requiere para su operación. Y debe operar correctamente con independencia de los mecanismos internos que utilice para soportar la funcionalidad de la interfaz. Es una de las mejores técnicas para incrementar la calidad del software.

1.6.4 Lenguajes de Programación Web.

Los lenguajes de Programación Web se clasifican en lenguajes del lado del cliente y lenguajes del lado del servidor. Los que trabajan del lado del servidor desarrollan la lógica de negocio dentro del servidor, además se encargan del acceso a los distintos Sistemas de Gestión de Bases de Datos. Los lenguajes del lado del cliente transforman y visualizan la respuesta de negocio.

1.6.4.1 PHP.

PHP, cuyas siglas responden a un acrónimo recursivo (PHP: Hypertext Preprocessor) es un lenguaje de script usado principalmente para scripts a ejecutar en servidores web, sobre todo Apache. Es utilizado para la generación de páginas Web dinámicas, similar al ASP de Microsoft o el JSP de Sun, embebido en

páginas HTML y ejecutado en el servidor. Es una de las mejores herramientas para la creación de aplicaciones. PHP es un lenguaje sencillo, de sintaxis cómoda y similar a la de otros lenguajes como C o C++, es rápido a pesar de ser interpretado, multiplataforma. Dispone de una gran cantidad de librerías que facilitan muchísimo el desarrollo de las aplicaciones de código amigable, como es un lenguaje basado en herramientas con licencia de software libre dispone de una gran cantidad de características que lo convierten en la herramienta ideal para la creación de páginas web dinámicas.

1.6.4.2 XSLT.

XSLT (Extensible Stylesheet Language Transformation) es una tecnología que define un lenguaje para modificar archivos XML. Precisa definir que transformaciones se quieren hacer y a que partes del documento, es un lenguaje que se usa para convertir documentos XML en otros documentos XML.

XSLT es muy usado en la edición web, generando páginas HTML o XHTML. La unión de XML y XSLT permite separar contenido y presentación, aumentando así la productividad.

1.6.4.3 JavaScript.

Lenguaje de programación del lado del cliente utilizado en la mayoría de los sitios web por ser el navegador el que soporta la carga de procesamiento ya que no requiere de compilación y además por ser compatible con la mayoría de los navegadores modernos. Es un lenguaje con muchas posibilidades, pues permite la programación de pequeños scripts y programas más grandes, orientados a objetos, con funciones y estructuras de datos complejas. Pone a disposición del programador todos los elementos que forman la página Web, para que éste pueda acceder a ellos y modificarlos dinámicamente.

Con este lenguaje se pueden realizar efectos especiales sobre páginas Web, para crear contenidos dinámicos y elementos de la página que tengan movimiento, cambien de color o cualquier otro dinamismo.

1.6.4.4 XHTML.

XHTML, acrónimo inglés de (eXtensible Hypertext Markup Language) lenguaje extensible de marcado de hipertexto. Es la versión XML de HTML, por lo que tiene básicamente las mismas funcionalidades, pero cumple las especificaciones más estrictas de XML. Sirve para transmitir la información que contiene un documento, dejando para hojas de estilo y JavaScript su aspecto y diseño. Es el sucesor de HTML y por eso muchos lo consideran la versión actual del HTML, pero es una recomendación aparte y a la vez paralela de este lenguaje.

Las principales ventajas del XHTML son:

- Facilidad de edición directa del código y de mantenimiento.
- Formato abierto, compatible con los nuevos estándares que actualmente está desarrollando el W3C 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.

La familia XHTML es el siguiente paso en la evolución de Internet. Al migrar en este momento hacia XHTML, los desarrolladores de contenidos web entran en el mundo de XML con todos los beneficios que se esperan de él a la vez que se aseguran la compatibilidad con aplicaciones de usuario pasadas y futuras. [7]

1.6.5 Hojas de Estilo en Cascada (Cascading Style Sheets, CSS).

Lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML y por extensión en XHTML. Su finalidad es crear estilos físicos, separados de las etiquetas HTML, y aplicarlos en los bloques que se desee. Estos estilos podrán ser modificados en algunas ocasiones desde JavaScript, lo que brinda mayor interactividad. La sintaxis CSS permite aplicar al documento un formato de modo mucho más exacto.

Posibilitan una colocación flexible de la información de estilo puesto que al colocar las hojas de estilo en ficheros separados es más fácil reutilizarlas. Algunas veces es útil incluir instrucciones de representación

dentro del documento al que se aplican, ya sean agrupadas al comienzo del documento, o en atributos de los elementos a lo largo del cuerpo del documento. Para facilitar la gestión de estilos a nivel de sitio, esta especificación describe cómo utilizar los encabezados HTTP para especificar qué hojas de estilo se aplican a un documento. [8]

1.6.6 Sistemas de Gestión de Bases de Datos (SGBD).

Los Sistemas Gestores de Bases de Datos están dedicados a servir de interfaz entre las bases de datos y las aplicaciones que la utilizan. El objetivo fundamental de un SGBD consiste en suministrar al usuario las herramientas que le permitan manipular los datos, o sea, de forma que no le sea necesario conocer el modo de almacenamiento de los datos en la computadora, ni el método de acceso empleado. [9]

MySQL.

Es un SGBD basado en Open Source (Código abierto) diseñado para los sistemas Unix formando parte de la tecnología LAMP (Linux, Apache, MySQL y PHP), aunque existen versiones para Windows. Actualmente está en su versión 5.0.6-beta incluyendo procedimientos almacenados (stored procedures), disparadores (triggers), vistas (views) y muchas otras características. [10]

Su popularidad de uso para aplicaciones web está muy ligada a su buena integración con PHP, que a menudo aparece en combinación con MySQL. Diseñado con el objetivo de aumentar la velocidad. Es válido citar que consume muy pocos recursos de CPU y memoria por lo que tiene buen rendimiento, así como también un buen control de acceso usuarios-tablas-permisos.

1.6.7 Metodología de desarrollo de software.

Desarrollo basado en RUP bajo la herramienta Rational Rose.

Para lograr la productividad del software se necesita un proceso que integre las múltiples facetas del desarrollo del mismo.

El Proceso Unificado es un proceso de desarrollo de software (conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software). Es un marco de trabajo genérico que

puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyectos.

El Proceso Unificado está basado en componentes. Utiliza el lenguaje unificado de modelado (UML) para preparar todos los esquemas de un sistema de software. De hecho, UML es una parte esencial de RUP, sus desarrollos fueron paralelos. No obstante los verdaderos aspectos definitorios del proceso unificado se resumen en tres fases claves: dirigido por casos de uso, centrado en la arquitectura, e iterativo e incremental. [11]

1.6.7.1 UML.

UML (Unified Modeling Language) o Lenguaje de Modelación Unificado es un lenguaje gráfico para especificar, construir, visualizar y documentar las partes o artefactos (información que se utiliza o produce mediante un proceso de software). Pueden ser artefactos: un modelo, una descripción que comprende el desarrollo de software que se basen en el enfoque Orientado a Objetos, utilizándose también en el diseño Web. UML usa procesos de otras metodologías, aprovechando la experiencia de sus creadores, eliminó los componentes que resultaban de poca utilidad práctica y añadió nuevos elementos. [UML]

Es un lenguaje más expresivo, claro y uniforme que los anteriores definidos para el diseño Orientado a Objetos, que no garantiza el éxito de los proyectos pero si mejora sustancialmente el desarrollo de los mismos, al permitir una nueva y fuerte integración entre las herramientas, los procesos y los dominios.

UML es desde finales de 1997, un lenguaje de modelado orientado a objetos estándar, de acuerdo con el Object Management Group, siendo utilizado diariamente por grandes organizaciones como: Microsoft, Oracle, Rational. [12]

1.6.7.2 Rational Rose.

Rational Rose es la herramienta de modelación visual que provee el modelado basado en UML y es la que se utiliza en la modelación de este proyecto.

La Corporación Rational ofrece un Proceso Unificado (RUP) para el desarrollo de los proyectos de software, desde la etapa de Ingeniería de Requerimientos hasta la de pruebas. Para cada una de estas etapas existe una herramienta de ayuda en la administración de los proyectos, Rose es la herramienta del Rational para la etapa de análisis y diseño de sistemas.

Rose es una herramienta con plataforma independiente que ayuda a la comunicación entre los miembros de equipo, a monitorear el tiempo de desarrollo y a entender el entorno de los sistemas. Una de las grandes ventajas de Rose es que utiliza la notación estándar en la arquitectura de software(UML), la cual permite a los arquitectos de software y desarrolladores visualizar el sistema completo utilizando un lenguaje común, además los diseñadores pueden modelar sus componentes e interfaces en forma individual y luego unirlos con otros componentes del proyecto.[13]

Para desarrollar el proyecto se usó Rational Unified Process (RUP). Este es un proceso que garantiza la elaboración de todas las fases de un producto de software orientado a objeto. RUP utiliza UML, que es un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos.

1.6.8 Plataforma de Servicios (PLASER).

La plataforma de Servicios (PLASER), está conformada fundamentalmente por varias clases en PHP, una librería, que puede o no ser usada para que un componente se integre al Registro Informatizado de la Salud (RIS), pero de no ser usada, la seguridad corre por parte del programador. En esta versión PLASER sólo soporta como llamada RPC el protocolo SOAP, pero en futuras versiones se piensa en otros protocolos de transportes o incluso el acceso local a código a nivel de File System, de forma tal que para el programador es totalmente transparente si la invocación del proceso es remoto, local, por SOAP, directamente a código, etc.

Este sistema está concebido completamente sobre Arquitectura Basada en Componentes y Orientada a Servicios, usando el paradigma de XML Web Services específicamente SOAP. En su concepción se han utilizado estándares actuales y normas abiertas. PLASER constituye una plataforma sobre la que se pueden desplegar aplicaciones XML – Web Services, además facilita la programación y homogeneidad de los componentes. PLASER desde el punto de vista estructural, permite trabajar con cualquier base de datos que cumpla con la norma SQL ANSI 92; pero desde el punto de vista de implementación solo trabaja con las bases de datos soportadas por el componente DBX, ya que encapsula a dicho componente y lo utiliza para el acceso a bases de datos. [14]

1.7 Conclusiones

En este capítulo, se realizó un estudio del estado del arte acerca del Estándares de Diseño, para desarrollar el repositorio que será implementado. Se profundizó en el conocimiento de algunos conceptos necesarios para la comprensión de este trabajo como Estándares, Demos, Repositorio. Además, se llevó a cabo un análisis de las tecnologías utilizadas: lenguajes de programación como PHP, JavaScript; XHTML, el sistema gestor de bases de datos manipulado y la metodología que será utilizada a lo largo del desarrollo del sistema propuesto. Manteniendo la política adoptada por el MINSAP para el desarrollo de las aplicaciones de salud.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA.

2.1 Introducción.

En este capítulo se realiza la descripción de las características del sistema. Para poder entender mejor el contexto en que se ubica, se definen conceptos que se agrupan en un Modelo de Dominio. Se enumeran además los requisitos funcionales y no funcionales que debe tener la idea que se propone, lo que permite hacer una concepción general del trabajo, e identificar mediante un Diagrama de Casos de Uso, las relaciones de los actores que interactúan y las secuencias de acciones que se realizan. Se realiza además la descripción de cada uno de los casos de usos definidos, facilitando una mayor interpretación de los mismos.

2.2 Modelo del Dominio.

Permite de manera visual mostrar al usuario los principales conceptos que se manejan en el dominio del sistema en desarrollo, puesto que los procesos del negocio no son visibles. Esto ayuda a los usuarios, clientes y desarrolladores e interesados, a utilizar un vocabulario común para poder entender el contexto en que se emplaza el sistema. Para capturar correctamente los requisitos y poder construir un sistema correcto se necesita tener un firme conocimiento del funcionamiento del objeto de estudio. Este modelo va a contribuir posteriormente a identificar algunas clases que se utilizarán en el sistema.

2.2.1 Conceptos fundamentales.

A continuación se identifican todos los conceptos que se utilizarán en el diagrama, mediante un glosario de términos:

Repositorio: Sitio centralizado donde se almacena y mantiene información, habitualmente bases de datos o archivos informáticos. Puede ser de acceso público o estar protegido y necesitar de previa autenticación.

Estándar: Es un requisito, regla, especificación o recomendación basada en principios probados y en la práctica. Se aplican generalmente a características básicas de la interfaz de usuario. Con el hecho de

desarrollar estándares para la interfaz se intenta conseguir un software más fácil y seguro, estableciendo unos requisitos mínimos de fabricación y eliminando inconsistencias y variaciones innecesarias en las interfaces.

Demo: Abreviatura de la palabra demostración. Ejemplo práctico que ejemplifica un estándar tanto en apariencia como en funcionamiento.

Usuario: Persona que accede a la aplicación.

Recurso: Se le denomina recurso a aquel elemento que proporcione algún tipo de información o aplicación como es el caso de archivos, documentos y herramientas que se publican en la aplicación, y que están relacionados con el diseño de interfaz y la programación.

Administrador: Usuario con privilegios de Administrador, tiene a su cargo la gestión de la aplicación.

Visualizador: Usuario con privilegios de Visualizador, accede al repositorio para consultar o descargar los estándares de diseño de interfaz y los recursos que se brindan en este.

2.2.2 Modelo del Dominio.

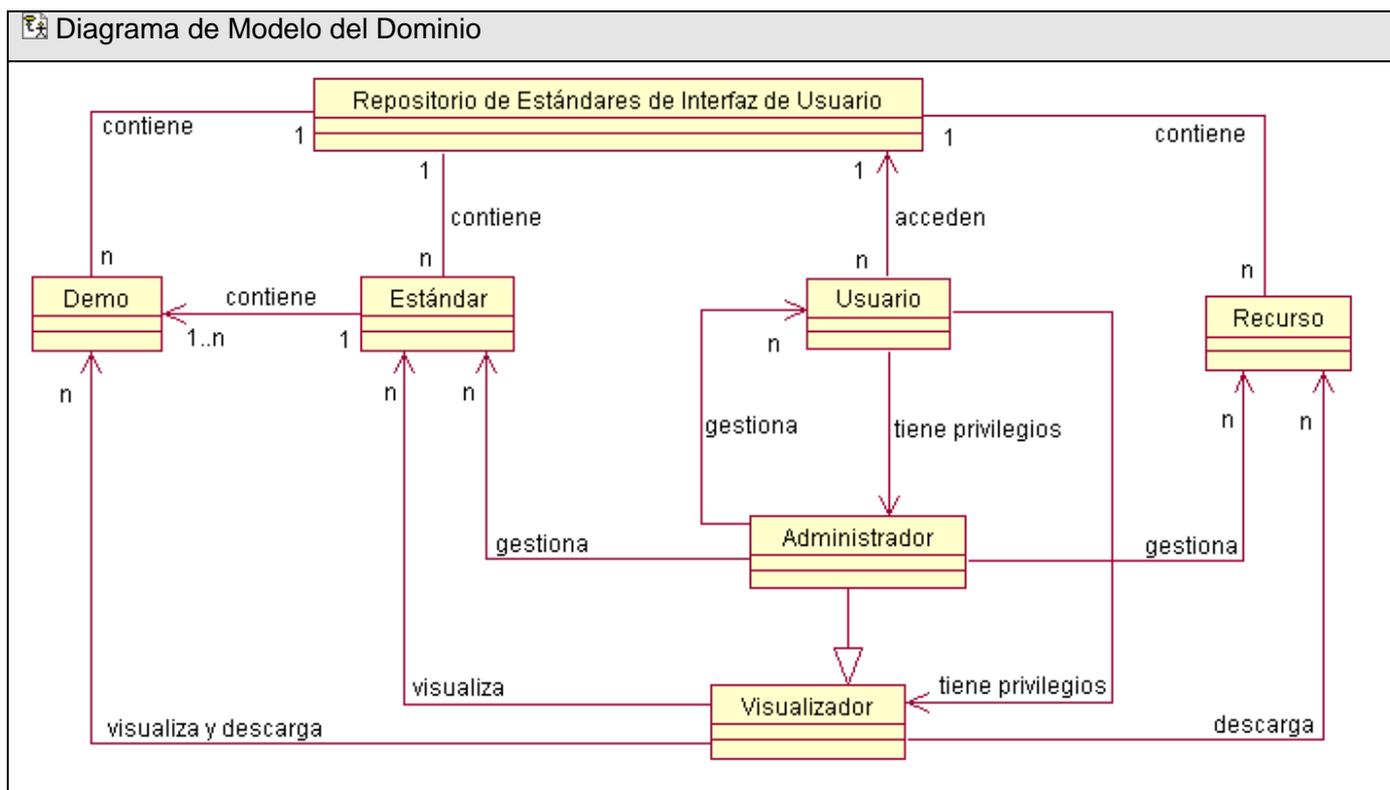


Figura 2.1 – Diagrama de Modelo del Dominio.

2.3 Propuesta del sistema.

Dando cumplimiento a los objetivos propuestos al inicio de este trabajo, se arriba a la conclusión de que es necesario implementar una aplicación web que servirá de repositorio para garantizar el proceso relacionado con el diseño de interfaz y sus funcionalidades asociadas, en la capa de presentación de los módulos del Sistema de Información para la Salud. Que cuente con las tecnologías actuales usadas en el mundo y que además posibilite a los Diseñadores y Programadores de Capa de Presentación consultar y utilizar la información registrada en dicho sistema, para agilizar y estandarizar el desarrollo de los módulos a implementar.

Es de gran importancia la estandarización del diseño de interfaz puesto permitirá que la capa de presentación que se le ofrece al usuario final posea un diseño robusto y homogéneo tanto en:

- Significado: Es la base del interfaz. Recoge el contenido o información de la pantalla. Textos, campos de formularios, botones, menús...
- Comportamiento: Trata el funcionamiento del interfaz. Cómo se comporta cuando un usuario envía un formulario (validaciones), hace clic en un enlace...
- Aspecto: Apariencia final de un sistema: colores, tipografía, disposición de los elementos en pantalla...

Para utilizar todas las funcionalidades, el sistema constará 2 tipos de usuarios (administrador y visualizador); o sea, un usuario con privilegios de administrador que puede realizar todo tipo de acciones sobre el sistema, tendrá acceso a toda la información, llevará el control de los demás usuarios y además podrá crear, modificar, eliminar y hacer los cambios y actualizaciones pertinentes. Por otro lado el usuario con privilegios de visualizador podrá consultar y descargar la información disponible en la aplicación.

Con el desarrollo de esta aplicación se espera identificar estándares que puedan derivar en componentes de interfaz reutilizables para su uso por los diferentes equipos de desarrollo del proyecto APS. Para abstraer al programador de tareas de diseño, sin dejar de recordar que estos objetos deben llevar aspectos visuales y estéticos puesto que están dedicados además a Diseñadores. Deben ser puestos a disposición de los equipos de programación a través de un repositorio único y actualizado con las siguientes características:

- Usable: Invitar al uso. Debe integrarse de forma cómoda en el proceso de trabajo de un desarrollador dándole respuestas a situaciones propias dentro de la construcción del interfaz de la aplicación.
- Visual y educativa: Crear documentación de carácter visual, compuesta de literatura esencial, ejemplos razonados que permitan desarrollar criterios mínimos de usabilidad y estética al personal técnico del proyecto.
- Actualizada: Debe contener ejemplos útiles, actuales y materiales para su aplicación directa.

2.3.1 Especificación de los requisitos del software.

2.3.1.1 Requisitos funcionales.

Con su definición se busca establecer un común entendimiento con el cliente sobre los objetivos del negocio propuesto y reflejar todo lo que el sistema debe hacer. La especificación de los requisitos debe ser precisa, completa y clara.

Definición de los Requisitos Funcionales.		
RF1 Autenticar Usuario.	RF10 Editar Estándar.	RF19 Visualizar Demos.
RF2 Agregar Usuario.	RF11 Editar Demo.	RF20 Descargar Demos.
RF3 Editar Usuario.	RF12 Eliminar Estándar.	RF21 Agregar Recurso.
RF4 Eliminar Usuario.	RF13 Eliminar Demo.	RF22 Editar Recurso.
RF5 Editar Perfil.	RF14 Buscar Estándar.	RF23 Eliminar Recurso.
RF6 Buscar Usuario.	RF15 Buscar Demo.	RF24 Buscar Recursos.
RF7 Listar Usuarios.	RF16 Listar Estándares.	RF25 Listar Recursos.
RF8 Agregar Estándar.	RF17 Listar Demo.	RF26 Descargar Recurso.
RF9 Agregar Demo.	RF18 Visualizar Estándar.	RF27 Cerrar Sesión.

Tabla 2.1 – Definición de los requisitos funcionales.

2.3.1.2 Requisitos no funcionales.

Los requisitos no funcionales se refieren a las cualidades del Sistema.

2.3.1.2.1 Usabilidad.

El sistema será desarrollado para uso del personal del proyecto, es decir personas con un nivel medio y avanzado en el manejo de las computadoras y de ambientes web. Garantizar un acceso fácil y rápido a los usuarios.

2.3.1.2.2 Apariencia o interfaz externa.

La interfaz será amigable para el usuario y capaz de adaptarse a cualquier resolución de pantalla (800x600 píxeles o 1024x768 píxeles) sin perder la estructura.

2.3.1.2.3 Rendimiento.

Garantizar una respuesta rápida del sistema.

2.3.1.2.4 Software.

Los usuarios podrán acceder a la aplicación a través de cualquier navegador web. Recomendados: Mozilla Firefox 1.5, Internet Explorer 4.0 o superior.

2.3.1.2.5 Hardware.

Hardware. Procesador 486DX / 66 MHz o superior. 16 MB de memoria, se recomienda más memoria para obtener un mayor rendimiento. Monitor VGA o superior. Ratón Microsoft o compatible.

2.3.1.2.6 Portabilidad.

Permitir la ejecución del sistema tanto en el Sistema Operativo GNU/Linux, como en Microsoft Windows 98 o superior.

2.3.1.2.7 Seguridad.

Otorgar privilegios a los usuarios. Según el tipo de usuario registrado (Administrador o Visualizador) accederá sólo a la información a la que tiene derecho. Disponer de un mecanismo de seguridad basado en el modelo de Autenticación, Autorización y Auditoría (AAA).

2.3.1.2.8 Confiabilidad.

La información manejada por el sistema está protegida de acceso no autorizado.

2.3.1.2.9 Ayuda y documentación en línea.

Todos los campos y botones mostrarán al poner el cursor del ratón encima una breve descripción de lo que hace o el usuario debe entrar.

2.4 Modelo de Casos de Uso del Sistema.

En este epígrafe se definen los actores y los casos de uso del sistema, así como también se describen los casos de uso más relevantes.

2.4.1 Definición de los actores.

Actores	Justificación
Administrador.	Tiene a su cargo la gestión de la información de la aplicación, interactúa con esta para introducir, actualizar y eliminar los datos de los usuarios, recursos, estándares y demos. Puede realizar además todas las operaciones de un Visualizador.
Visualizador.	Interactúa con el sistema para obtener (visualizar o descargar) información relacionada con los estándares, demos y recursos; puede buscar y listar dicha información.

Tabla 2.2 – Definición de los actores del sistema.

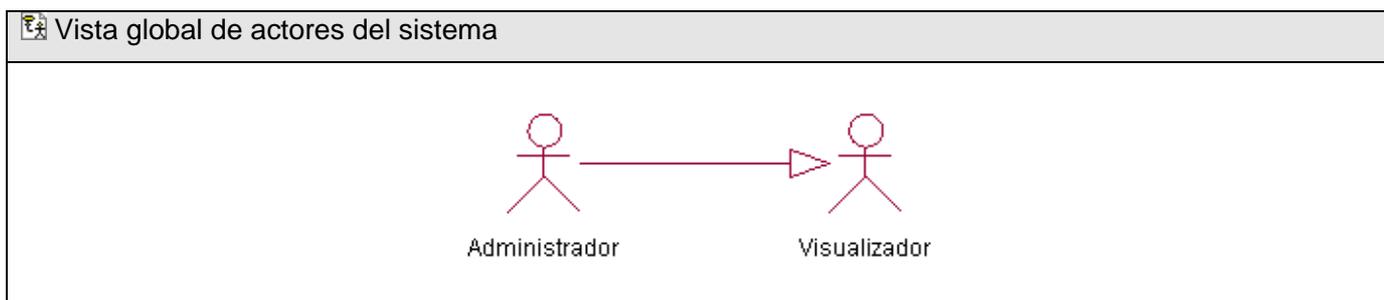


Figura 2.2 – Vista global de actores del sistema.

2.4.2 Definición de Casos de Uso.

Seguidamente se presentan los casos de uso definidos para satisfacer los requerimientos funcionales de sistema:

CU1	Autenticar Usuario.
Actor	Administrador y Visualizador.
Descripción	Los usuarios se identifican en el sistema para acceder a las opciones disponibles del mismo en correspondencia de sus privilegios.
Referencia	RF1.

Tabla 2.3 – Definición Caso de Uso Autenticar Usuario.

CU2	Gestionar Usuario.
Actor	Administrador.
Descripción	Posibilita a los administradores agregar, editar y eliminar la información relacionada con los usuarios y además que una vez creado el usuario se le envíe una notificación con su perfil. Permite además hacer una búsqueda de usuarios por (nombre, apellidos, usuario y tipo de usuario) obteniendo un listado de los usuarios que correspondan con los parámetros de búsqueda introducidos, en caso de no introducir ningún parámetro se muestra un listado de todos los usuarios existentes en la aplicación.
Referencia	RF2, RF3, RF4, RF6, RF7, RF27.

Tabla 2.4 – Definición Caso de Uso Gestionar Usuario.

CU3	Editar Perfil.
Actor	Administrador y Visualizador.
Descripción	Permite a los Administradores modificar los datos referentes a su usuario y a los Visualizadores cambiar su contraseña.
Referencia	RF5, RF27.

Tabla 2.5 – Definición Caso de Uso Editar Perfil.

CU4	Gestionar Estándar.
Actor	Administrador.
Descripción	Permite a los administradores agregar, editar y eliminar la información relacionada con los Estándares y sus respectivos Demos.
Referencia	RF8, RF9, RF10, RF11, RF12, RF13, RF27.

Tabla 2.6 – Definición Caso de Uso Gestionar Estándar.

CU5	Buscar Estándar.
Actor	Administrador y Visualizador.
Descripción	Posibilita hacer una búsqueda de estándares por (nombre y(o) descripción) obteniendo un listado de los estándares que correspondan con los criterios de búsqueda entrados en caso de no introducir ningún parámetro se muestra un listado de todos los estándares existentes en la aplicación.
Referencia	RF14, RF16, RF27.

Tabla 2.7 – Definición Caso de Uso Buscar Estándar.

CU6	Buscar Demo.
Actor	Administrador y Visualizador
Descripción	Posibilita hacer una búsqueda de demos por (nombre y(o) descripción) obteniendo un listado de los demos que correspondan con los criterios de búsqueda entrados en caso de no introducir ningún parámetro se muestra un listado de todos los demos existentes en la aplicación.
Referencia	RF15, RF17, RF27.

Tabla 2.8 – Definición Caso de Uso Buscar Demo.

CU7	Visualizar Estándar.
Actor	Administrador y Visualizador.
Descripción	Muestra los datos de un estándar seleccionado (nombre y descripción) y el listado de los Demos asociados a dicho estándar.
Referencia	RF18, RF27.

Tabla 2.9 – Definición Caso de Uso Visualizar Estándar.

CU8	Visualizar Demo.
Actor	Administrador y Visualizador.
Descripción	Posibilita visualizar un demo específico para observar su funcionamiento.
Referencia	RF19.

Tabla 2.10 – Definición Caso de Uso Visualizar Demo.

CU9	Descargar Demo.
Actor	Administrador y Visualizador.
Descripción	Permite descargar un ejemplo práctico del estándar descrito.
Referencia	RF20.

Tabla 2.11 – Definición Caso de Uso Descargar Demo.

CU10	Gestionar Recurso.
Actor	Administrador
Descripción	Posibilita a los administradores agregar, editar y eliminar recursos de la aplicación.
Referencia	RF21, RF22, RF23, RF27.

Tabla 2.12 – Definición Caso de Uso Gestionar Recurso.

CU11	Buscar Recurso.
Actor	Administrador y Visualizador.
Descripción	Posibilita hacer una búsqueda de los recursos por (nombre, y tipo de recurso) obteniendo un listado de los recursos que corresponden con los parámetros de búsqueda introducidos, en caso de no introducir ningún parámetro se muestra un listado de todos los recursos existentes en la aplicación.
Referencia	RF24, RF25, RF27.

Tabla 2.13 – Definición CU Buscar Recurso.

CU12	Descargar Recurso.
Actor	Administrador y Visualizador.
Descripción	Permite descargar un recurso.
Referencia	RF26.

Tabla 2.14 – Definición Caso de Uso Descargar Recurso.

2.4.3 Diagrama de Casos de Uso.

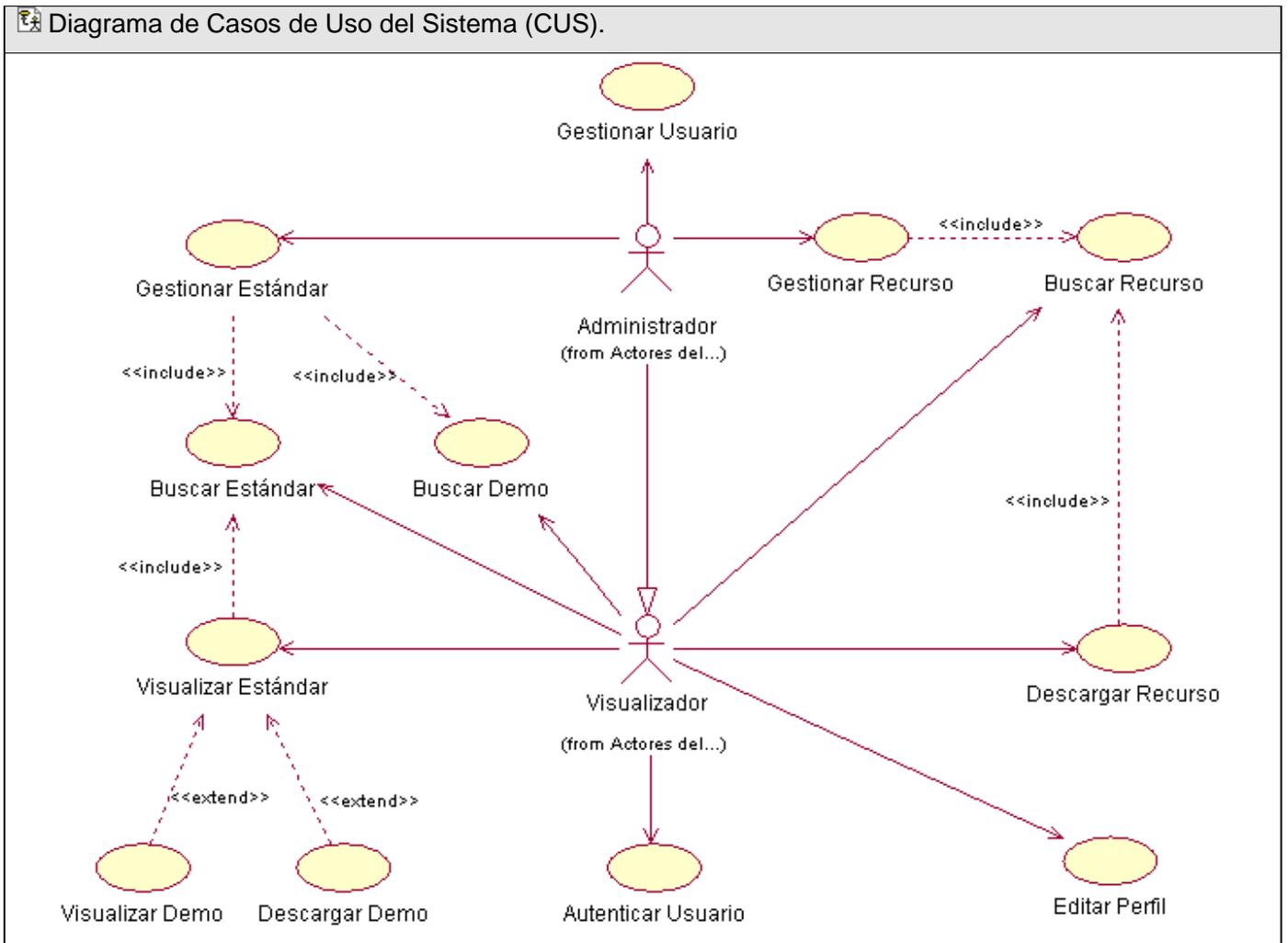


Figura 2.3 – Diagrama de CUS.

2.4.4 Descripción textual de los Casos de Uso.

CU1	Autenticar Usuario	
Propósito	Acceder a la aplicación según privilegios.	
Actores	Administrador, Visualizador (Inicia).	
Resumen	El caso de uso se inicia cuando el actor decide entrar en la aplicación. Posibilitando a los administradores y visualizadores identificarse en el sistema para acceder a las opciones en correspondencia de sus privilegios.	
Precondiciones	Tener un usuario válido en el sistema.	
Poscondiciones	Acceder a la aplicación según los privilegios.	
Referencias	RF1.	
Prioridad	Crítico.	
Flujo Normal de eventos		
Acción del actor	Respuesta del sistema	
1. Accede a la aplicación.	2. Muestra la interfaz entrada al sistema.	
3. Introduce su usuario y contraseña.	4. Valida los datos entrados. 4.1. En caso de ser correctos muestra la interfaz de bienvenida al sistema y las opciones a las que tiene acceso según sus privilegios, terminando así el CU. 4.2. En caso de que los datos no sean correctos, ver Flujo alternativo 1.	
Flujo alternativo 1		
Acción del actor	Respuesta del sistema	
	1. Muestra mensaje de error "Usuario o Contraseña no válidos. Por favor intente de nuevo".	
Interfaz: (Anexo 2).		

Tabla 2.15 – Descripción del Caso de Uso Autenticar Usuario.

CU2	Gestionar Usuario	
Propósito	Efectuar las operaciones agregar, buscar, editar y eliminar usuarios para mantener actualizada la información referente a los mismos.	
Actores	Administrador. (Inicia).	
Resumen	El Caso de Uso comienza cuando el actor escoge la opción “Búsqueda/Listado” de Usuarios o decide agregar un usuario, modificar la información asociada a los datos de un usuario o eliminarlo, el sistema en cada caso muestra la interfaz correspondiente con la opción seleccionada. Posibilita hacer una búsqueda de usuarios para su posterior gestión, dicha búsqueda se hará por (nombre, apellidos, usuario y privilegios) obteniendo un listado de los usuarios que correspondan con los parámetros de búsqueda entrados, se le permitirá además al actor obtener un listado completo de todos los usuarios de la aplicación. Viabiliza a los administradores gestionar la información relacionada con los datos de los usuarios.	
Precondiciones	El usuario debe estar autenticado en el Sistema con privilegios de Administrador.	
Poscondiciones	Se registran los datos del nuevo usuario o los cambios producidos en los datos del mismo en el sistema.	
Referencias	RF2, RF3, RF4, RF6, RF7, RF27.	
Prioridad	Crítico.	
Flujo Normal de eventos		
Acción del actor	Respuesta del sistema	
1. Selecciona la opción “Búsqueda/Listado” de Usuarios.	2. Muestra la interfaz correspondiente para realizar la búsqueda de usuarios por nombre, apellidos, usuario y privilegios.	
3. Introduce los criterios de búsqueda que desee y presiona buscar. En caso contrario ver Flujo alternativo1.	4. Muestra un listado con los resultados encontrados para esos criterios de búsqueda, ordenados por nombre en forma ascendente. Si son más de 10 los resultados, activa un paginado que permitirá al usuario moverse por todos los resultados encontrados, puesto que estos se muestran de 10 en 10. El paginado tiene la siguiente	

	<p>estructura:</p> <p><i>“Página #”</i> Combo box para escoger de forma directa la página que desee el usuario visualizar.</p> <p><i>“Primero”</i> Botón para escoger de forma directa el primer listado (los primeros 10 resultados).</p> <p><i>“Anterior”</i> Botón para escoger el listado anterior (los 10 resultados anteriores).</p> <p><i>“Próximo”</i> Botón para escoger el próximo listado (los 10 resultados siguientes).</p> <p><i>“Último”</i> Botón para escoger de forma directa el último listado de resultados.</p>
5. El actor selecciona el usuario que desea, marcando el radio button.	6. Ilumina la fila en la que se encuentra el usuario seleccionado.
7. Si presiona el botón <i>“Editar”</i> ver Sección <i>“Editar Usuario”</i> . 7.1 Si presiona el botón <i>“Eliminar”</i> ver Sección <i>“Eliminar Usuario”</i> .	
Flujo alternativo 1	
Acción del actor	Respuesta del sistema

<p>1. Selecciona el botón “Mostrar Todos”.</p>	<p>2. Muestra un listado con todos los usuarios de la aplicación, ordenados por nombre en forma ascendente.</p> <p>Si son más de 10 los resultados, activa un paginado que permitirá al usuario moverse por todos los resultados encontrados, puesto que estos se muestran de 10 en 10. El paginado tiene la siguiente estructura:</p> <p>“Página #” Combo box para escoger de forma directa la página que desee el usuario visualizar.</p> <p>“Primero” Botón para escoger de forma directa el primer listado (los primeros 10 resultados).</p> <p>“Anterior” Botón para escoger el listado anterior (los 10 resultados anteriores).</p> <p>“Próximo” Botón para escoger el próximo listado (los 10 resultados siguientes).</p> <p>“Último” Botón para escoger de forma directa el último listado de resultados.</p>
<p>3. El actor selecciona el usuario que desea, marcando el botón de opción.</p>	<p>4. Ilumina la fila en la que se encuentra el usuario seleccionado, indicando que el usuario está seleccionado.</p>
<p>5. Si presiona el botón “<i>Editar</i>” ver Sección “Editar Usuario”.</p> <p>5.1 Si presiona el botón “Eliminar” ver Sección “Eliminar Usuario”.</p>	
<p>Sección “Agregar Usuario”</p>	
<p>Acción del actor</p>	<p>Respuesta del sistema</p>
<p>1. Selecciona la opción “Agregar” del menú.</p>	<p>2. Muestra la interfaz Agregar Usuario.</p>
<p>3. Introduce nombre, apellidos, correo electrónico, usuario, contraseña, confirmación de</p>	<p>4. Activa el botón Aceptar y Cancelar.</p>

contraseña y privilegios (Administrador o Visualizador).	
5. Selecciona el botón "Aceptar". En caso contrario ver Flujo alternativo1.	6. Valida los datos entrados. 6.1. En caso de que los datos no sean correctos, ver Flujo alternativo 2. 6.2. En caso de que los datos sean correctos, se verifica si existe el usuario en la aplicación, si existe ver Flujo alternativo 3.
	7. Crea el nuevo usuario terminando así el CU.
Flujo alternativo 1	
Acción del actor	Respuesta del sistema
1. Selecciona el botón "Cancelar".	2. Muestra el mensaje de confirmación "¿Desea borrar los datos entrados hasta el momento?".
3. Acepta el mensaje de confirmación.	4. Borra los datos de los campos.
5. Ejecuta la acción 3 del Flujo Normal de Eventos del escenario agregar usuario o realiza otra acción terminando así el CU.	
Flujo alternativo 2	
Acción del actor	Respuesta del sistema
	1. Muestra un mensaje de error en correspondencia con el dato incorrecto.
Flujo alternativo 3	
Acción del actor	Respuesta del sistema
	1. Muestra un mensaje de error informando que ya existe el usuario.
Sección "Editar Usuario"	
Acción del actor	Respuesta del sistema
	1. Muestra la interfaz Editar Usuario.
2. Introduce los datos que desea modificar.	3. Activa el botón Aceptar.

4. Selecciona el botón "Aceptar". En caso contrario ver Flujo alternativo1.	5. Valida los datos entrados. 5.1 Si los datos son correctos, se modifica la información introducida terminando así el CU. 5.2 En caso de que los datos no sean correctos, ver Flujo alternativo 2.
	6. Actualiza los cambios terminando así el CU.
Flujo alternativo 1	
Acción del actor	Respuesta del sistema
1. Selecciona el botón "Cancelar"	2. Se cierra la interfaz de editar usuario y muestra la pantalla de búsqueda de donde partió.
Flujo alternativo 2	
Acción del actor	Respuesta del sistema
	1. Muestra un mensaje de error en correspondencia con el dato incorrecto.
Sección "Eliminar Usuario"	
Acción del actor	Respuesta del sistema
	1. Muestra el mensaje de confirmación "Está seguro que desea eliminar el usuario".
2. Selecciona el botón "Aceptar". En caso contrario ver Flujo alternativo1.	3. Elimina toda la información perteneciente al usuario seleccionado terminando así el CU.
Flujo alternativo 1	
Acción del actor	Respuesta del sistema
1. Selecciona el botón "Cancelar".	1. Cierra la ventana del mensaje de confirmación y no se elimina el usuario.
Interfaz: (Anexo 3).	

Tabla 2.16 – Descripción del Caso de Uso Gestionar Usuario.

CU3	Editar Perfil	
Propósito	Modificar datos en el perfil de usuario.	
Actores	Administrador, Visualizador (Inicia).	
Resumen	El caso de uso se inicia cuando el actor desean hacer cambios en su perfil de usuario, permite a los Administradores modificar los datos referentes a su usuario y a los Visualizadores cambiar su contraseña.	
Precondiciones		
Poscondiciones	El sistema deja actualizada la información del usuario.	
Referencias	RF5, RF27.	
Prioridad	Auxiliar.	
Flujo Normal de eventos		
Acción del actor	Respuesta del sistema	
1. Selecciona la opción Editar Perfil del menú.	2. Muestra la interfaz Editar Perfil y los campos que se podrán modificar en correspondencia con el tipo de usuario.	
2. Introduce los datos que desea modificar.	3. Activa el botón Aceptar.	
4. Selecciona el botón "Aceptar". En caso contrario ver Flujo alternativo1.	6. Valida los datos entrados. 6.1 Si los datos son correctos, se modifica la información introducida terminando así el CU. 6.2 En caso de que los datos no sean correctos, ver Flujo alternativo 2.	
	7. Actualiza los cambios terminando así el CU.	
Flujo alternativo 1		
Acción del actor	Respuesta del sistema	
1. Selecciona el botón "Cancelar".	2. Muestra el mensaje de confirmación "¿Desea cancelar la solicitud de "Editar Perfil?".	
Flujo alternativo 2		
Acción del actor	Respuesta del sistema	

	1. Muestra un mensaje de error en correspondencia con el dato incorrecto.
Interfaz: (Anexo 4).	

Tabla 2.17 – Descripción del Caso de Uso Editar Perfil.

CU4	Gestionar Estándar.	
Propósito	Agregar, editar y eliminar los datos de los Estándares y sus respectivos Demos.	
Actores	Administrador (Inicia).	
Resumen	El Caso de Uso comienza cuando el actor decide agregar un estándar, modificar la información asociada a los estándares o eliminarlo. Permite gestionar la información relacionada con los Estándares y sus respectivos Demos.	
Precondiciones	El usuario debe estar autenticado en el Sistema con privilegios de Administrador.	
Poscondiciones	Se registra(n) lo(s) datos(s) del nuevo estándar o los cambios producidos en los datos del mismo en el sistema.	
Referencias	RF8, RF9, RF10, RF11, RF12, RF13, RF27.	
Prioridad	Crítico.	
Flujo Normal de eventos		
Sección Agregar Estándar		
Acción del actor	Respuesta del sistema	
1. Selecciona la opción Agregar Estándar.	2. Muestra la interfaz Agregar Estándar.	
3. Introduce los datos del estándar (nombre y descripción).		
4. Selecciona la opción de Agregar Demo para adjuntar todos los demos perteneciente a dicho estándar.	5. Muestra la interfaz Agregar Demo.	
6. Introduce los datos del demo (nombre, descripción y la url para subir el archivo).	7. Activa el botón “Aceptar” en la ventana de Agregar Demo.	
8. Selecciona el botón “Aceptar”. En caso contrario ver Flujo alternativo1.	9. Valida los datos entrados.	
	9.1. En caso de que los datos no sean correctos, ver	

	Flujo alternativo 2.
	10. Agrega el nuevo demo al estándar.
	11. Activa el botón “ <i>Aceptar</i> ” de la interfaz Agregar Estándar.
12. Selecciona el botón aceptar para agregar el estándar con su(s) respectivo(s) demo(s) o repite la acción 4 tantas veces desee.	13. Valida los datos entrados. 13.1. En caso de que los datos no sean correctos, ver Flujo alternativo 2. 13.2. En caso de que los datos sean correctos, se verifica si existe el estándar en la aplicación, si existe ver Flujo alternativo 3.
	14. Registra el nuevo estándar y sus respectivos demos, terminando así el CU.
Flujo alternativo 1	
Acción del actor	Respuesta del sistema
1. Selecciona el botón “ <i>Cancelar</i> ”.	2. Muestra el mensaje de confirmación “¿Desea cancelar la solicitud de “Agregar Demo?””.
Flujo alternativo 2	
Acción del actor	Respuesta del sistema
	1. Muestra un mensaje de error en correspondencia con el dato incorrecto.
Flujo alternativo 3	
Acción del actor	Respuesta del sistema
	1. Muestra un mensaje de error informando que ya existe el estándar.
Sección Editar Estándar	
Acción del actor	Respuesta del sistema
1. Selecciona la opción Editar Estándar.	2. Muestra la interfaz Editar Estándar.

<p>3. Introduce el dato que desea modificar.</p> <p>3.1. Si desea Agregar un Demo ver Sección “Agregar Estándar” acción 4.</p> <p>3.2. Si desea Eliminar un Demo ver Sección “Eliminar Demo”.</p> <p>3.3. Si desea Editar un Demo ver Sección “Editar Demo”.</p>	<p>3. Activa el botón Aceptar de la interfaz Editar Estándar.</p>
<p>4. Selecciona el botón “Aceptar”. En caso contrario ver Flujo alternativo 1.</p>	<p>5. Valida los datos entrados.</p> <p>5.1. Si los datos son correctos, actualiza los cambios terminando así el CU.</p> <p>5.2. En caso de que los datos no sean correctos, ver Flujo alternativo 2.</p>
Flujo alternativo 1	
Acción del actor	Respuesta del sistema
1. Selecciona el botón “ <i>Cancelar</i> ”.	2. Muestra el mensaje de confirmación “¿Desea cancelar la solicitud de “Editar Estándar?””
Flujo alternativo 2	
Acción del actor	Respuesta del sistema
	1. Muestra un mensaje de error en correspondencia con el dato incorrecto.
Sección Editar Demo	
Acción del actor	Respuesta del sistema
1. Selecciona el demo que desea editar, marcando el botón de opción.	2. Ilumina la fila en la que se encuentra el demo elegido, indicando que el demo está seleccionado.
3. Presiona el botón “Editar”.	4. Muestra la interfaz Editar Demo.
6. Introduce los datos del demo que desea modificar (nombre, descripción y la url para subir el archivo).	7. Activa el botón “ <i>Aceptar</i> ” en la ventana de Editar Demo.

8. Selecciona el botón "Aceptar". En caso contrario ver Flujo alternativo1.	9. Valida los datos entrados. 9.1. En caso de que los datos no sean correctos, ver Flujo alternativo 2.
	10. Agrega el nuevo demo al estándar.
	11. Modifica el demo y activa el botón "Aceptar" de la interfaz Editar Estándar.
Flujo alternativo 1	
Acción del actor	Respuesta del sistema
1. Selecciona el botón "Cancelar".	2. Muestra el mensaje de confirmación "¿Desea cancelar la solicitud de "Editar Demo?".
Flujo alternativo 2	
Acción del actor	Respuesta del sistema
	1. Muestra un mensaje de error en correspondencia con el dato incorrecto.
Sección Eliminar Estándar	
Acción del actor	Respuesta del sistema
1. Selecciona el botón "Eliminar".	2. Muestra el mensaje de confirmación "Está seguro que desea eliminar el estándar seleccionado".
3. Selecciona el botón "Aceptar". En caso contrario ver Flujo alternativo1.	4. Elimina toda la información perteneciente al estándar y sus respectivos demos terminando así el CU.
Flujo alternativo 1	
Acción del actor	Respuesta del sistema
1. Selecciona el botón "Cancelar".	2. Cierra la ventana del mensaje de confirmación y no elimina el estándar.
Sección Eliminar Demo	
Acción del actor	Respuesta del sistema
1. Selecciona el demo que desea eliminar, marcando el botón de opción.	2. Ilumina la fila en la que se encuentra el demo elegido, indicando que el demo está seleccionado.

3. Presiona el botón "Eliminar".	4. Muestra el mensaje de confirmación "Está seguro que desea eliminar el demo seleccionado".
5. Selecciona el botón "Aceptar". En caso contrario ver Flujo alternativo1.	6. Elimina el demo terminando así el CU.
Flujo alternativo 1	
Acción del actor	Respuesta del sistema
1. Selecciona el botón "Cancelar".	2. Cierra la ventana del mensaje de confirmación y no elimina el estándar.
Interfaz: (Anexo 5).	

Tabla 2.18 – Descripción del Caso de Uso Gestionar Estándar.

CU5	Buscar Estándar
Propósito	Hacer una búsqueda o mostrar el listado de todos los estándares.
Actores	Administrador, Visualizador (Inicia).
Resumen	El Caso de Uso comienza cuando el actor escoge la opción "Búsqueda/Listado" de estándares, el sistema muestra la interfaz con las opciones de búsqueda correspondientes, posibilitando hacer una búsqueda de estándares por (nombre y descripción) obteniendo un listado de los estándares que correspondan con los parámetros de búsqueda introducidos, en caso de no querer hacer una búsqueda se le da posibilidad al actor de mostrar un listado con todos los estándares existentes en la aplicación.
Precondiciones	
Poscondiciones	
Referencias	RF14, RF16, RF27.
Prioridad	Crítico.
Flujo Normal de eventos	
Acción del actor	Respuesta del sistema
1. Selecciona la opción "Búsqueda/Listado" de Estándares.	2. Muestra la interfaz correspondiente para realizar la búsqueda de estándares por (nombre y(o) descripción).

<p>3. Introduce los criterios de búsqueda que desee y presiona el botón “Buscar”. En caso contrario ver Flujo alternativo1.</p>	<p>4. Muestra un listado con los resultados encontrados, ordenados por nombre en forma ascendente.</p> <p>Si son más de 10 los resultados, activa un paginado que permitirá al usuario moverse por todos los resultados encontrados, puesto que estos se muestran de 10 en 10. El paginado tiene la siguiente estructura:</p> <p>“Página #” Combo box para escoger de forma directa la página que desee el usuario visualizar.</p> <p>“Primero” Botón para escoger de forma directa el primer listado (los primeros 10 resultados).</p> <p>“Anterior” Botón para escoger el listado anterior (los 10 resultados anteriores).</p> <p>“Próximo” Botón para escoger el próximo listado (los 10 resultados siguientes).</p> <p>“Último” Botón para escoger de forma directa el último listado de resultados.</p>
<p>5. El actor selecciona el estándar que desea, marcando el radio button.</p>	<p>6. Ilumina la fila en la que se encuentra el estándar seleccionado.</p>
<p>7. Si desea visualizar el estándar ver CU7 “Visualizar Estándar”.</p> <p>7.1. Si desea editar el Estándar ver CU4 “Gestionar Estándar” – Sesión “Editar Estándar”.</p> <p>7.1 Si desea eliminar el estándar ver CU4 “Gestionar Estándar” – Sesión “Eliminar Estándar”.</p>	
<p>Flujo alternativo 1</p>	
<p>Acción del actor</p>	<p>Respuesta del sistema</p>

<p>1. Presiona el botón <i>“Mostrar Todos”</i>.</p>	<p>2. Muestra un listado con todos los estándares existentes en la aplicación, ordenados por nombre en forma ascendente.</p> <p>Si son más de 10 los resultados, activa un paginado que permitirá al usuario moverse por todos los resultados encontrados, puesto que estos se muestran de 10 en 10. El paginado tiene la siguiente estructura:</p> <p><i>“Página #”</i> Combo box para escoger de forma directa la página que desee el usuario visualizar.</p> <p><i>“Primero”</i> Botón para escoger de forma directa el primer listado (los primeros 10 resultados).</p> <p><i>“Anterior”</i> Botón para escoger el listado anterior (los 10 resultados anteriores).</p> <p><i>“Próximo”</i> Botón para escoger el próximo listado (los 10 resultados siguientes).</p> <p><i>“Último”</i> Botón para escoger de forma directa el último listado de resultados.</p>
<p>Interfaz: (Anexo 6).</p>	

Tabla 2.19 – Descripción del Caso de Uso Buscar Estándar.

CU6	Buscar Demo	
Propósito	Hacer una búsqueda o mostrar el listado de todos los demos.	
Actores	Administrador, Visualizador (Inicia).	
Resumen	El Caso de Uso comienza cuando el actor escoge la opción “Búsqueda/Listado” de demos, el sistema muestra la interfaz con las opciones de búsqueda correspondientes, posibilitando hacer una búsqueda de demos por (nombre y(o) descripción) obteniendo un listado con los demos que correspondan con los parámetros de búsqueda introducidos, en caso de no querer hacer una búsqueda se le da posibilidad al actor de mostrar un listado con todos los demos existentes en la aplicación, informando a que estándar pertenece cada uno.	
Precondiciones		
Poscondiciones		
Referencias	RF15, RF17, RF27.	
Prioridad	Crítico.	
Flujo Normal de eventos		
Acción del actor	Respuesta del sistema	
1. Selecciona la opción “Búsqueda/Listado” de Demos.	2. Muestra la interfaz correspondiente para realizar la búsqueda de demos por (nombre y(o) descripción).	
3. Introduce los criterios de búsqueda que desee y presiona el botón “Buscar”. En caso contrario ver Flujo alternativo1.	4. Muestra un listado con los resultados encontrados, ordenados por nombre en forma ascendente. Si son más de 10 los resultados, activa un paginado que permitirá al usuario moverse por todos los resultados encontrados, puesto que estos se muestran de 10 en 10. El paginado tiene la siguiente estructura: “Página #” Combo box para escoger de forma directa la página que desee el usuario visualizar. “Primero” Botón para escoger de forma directa el primer listado (los primeros 10 resultados). “Anterior” Botón para escoger el listado anterior (los	

	<p>10 resultados anteriores).</p> <p><i>“Próximo”</i> Botón para escoger el próximo listado (los 10 resultados siguientes).</p> <p><i>“Último”</i> Botón para escoger de forma directa el último listado de resultados.</p>
5. El actor selecciona el demo que desea, marcando el radio button.	6. Ilumina la fila en la que se encuentra el demo seleccionado.
<p>7. Si desea visualizar un demo ver CU8 “Visualizar Demo”.</p> <p>7.1. Si desea descargar un demo ver CU9 “Descargar Demo”.</p> <p>7.2. Si desea editar un demo ver CU4 “Gestionar Estándar” – Sesión “Editar Demo”.</p> <p>7.3. Si desea eliminar un demo ver CU4 “Gestionar Estándar” – Sesión “Eliminar Demo”.</p>	
Flujo alternativo 1	
Acción del actor	Respuesta del sistema

<p>1. Presiona el botón “<i>Mostrar Todos</i>”.</p>	<p>2. Muestra un listado con todos los demos existentes en la aplicación, ordenados por nombre en forma ascendente.</p> <p>Si son más de 10 los resultados, activa un paginado que permitirá al usuario moverse por todos los resultados encontrados, puesto que estos se muestran de 10 en 10. El paginado tiene la siguiente estructura:</p> <p>“<i>Página #</i>” Combo box para escoger de forma directa la página que desee el usuario visualizar.</p> <p>“<i>Primero</i>” Botón para escoger de forma directa el primer listado (los primeros 10 resultados).</p> <p>“<i>Anterior</i>” Botón para escoger el listado anterior (los 10 resultados anteriores).</p> <p>“<i>Próximo</i>” Botón para escoger el próximo listado (los 10 resultados siguientes).</p> <p>“<i>Último</i>” Botón para escoger de forma directa el último listado de resultados.</p>
<p>Interfaz: (Anexo 7).</p>	

Tabla 2.20 – Descripción del Caso de Uso Buscar Demo.

CU7	Visualizar Estándar
Propósito	Visualizar un estándar específico para observar con mayor claridad sus datos.
Actores	Administrador, Visualizador (Inicia).
Resumen	El Caso de Uso comienza cuando el actor pincha el vínculo para visualizar el estándar y se muestra la interfaz correspondiente, una vez visualizado el estándar tendrá acceso a descargar y visualizar todos los demos pertenecientes al estándar.
Precondiciones	El estándar debe estar seleccionado.
Poscondiciones	
Referencias	RF18, RF27.

Prioridad	Secundario.	
Flujo Normal de eventos		
Acción del actor	Respuesta del sistema	
1. Selecciona la opción visualizar estándar mediante el vínculo en la página de búsqueda.	2. Muestra la interfaz con los datos del estándar seleccionado.	
5. Si desea visualizar un demo ver CU8 “Visualizar Demo”. 5.1. Si desea descargar un demo ver CU9 “Descargar Demo”. 5.2. Si desea editar el estándar visualizado ver CU4 “Gestionar Estándar” – Sesión “Editar Estándar”. 5.3. Si desea eliminar el estándar visualizado ver flujo alternativo 1.		
Flujo alternativo 1		
Acción del actor	Respuesta del sistema	
1. Selecciona el estándar que desea eliminar, marcando el botón de opción.	2. Ilumina la fila en la que se encuentra el estándar elegido, indicando que el estándar está seleccionado.	
3. Presiona el botón “ <i>Eliminar</i> ”.	4. Muestra el mensaje de confirmación “Está seguro que desea eliminar el Estándar”.	
5. Selecciona el botón “Aceptar”.	6. Elimina el estándar y muestra la pantalla de búsqueda de donde había partido, terminando así el CU.	
Interfaz: (Anexo 8).		

Tabla 2.21 – Descripción del Caso de Uso Visualizar Estándar.

CU8	Visualizar Demo
Propósito	Visualizar un demo específico para observar su funcionamiento.
Actores	Administrador, Visualizador (Inicia).
Resumen	El Caso de Uso comienza cuando el actor selecciona el vínculo para visualizar el demo ya sea a partir de la interfaz de búsqueda o de la del estándar y se muestra una ventana con el demo.
Precondiciones	El demo debe estar seleccionado.
Poscondiciones	
Referencias	RF19.
Prioridad	Secundario.
Flujo Normal de eventos	
Acción del actor	Respuesta del sistema
1. Selecciona la opción visualizar demo mediante el vínculo.	2. Muestra una nueva ventana con el demo seleccionado.
Interfaz: (Anexo 9).	

Tabla 2.22 – Descripción del Caso de Uso Visualizar Demo.

CU9	Descargar Demo
Propósito	Descargar de la aplicación un demo específico para su posterior utilización.
Actores	Administrador, Visualizador (Inicia).
Resumen	El Caso de Uso comienza cuando el actor decide descargar el demo y termina cuando se completa la operación de descarga.
Precondiciones	El demo debe estar seleccionado.
Poscondiciones	
Referencias	RF20.
Prioridad	Secundario.
Flujo Normal de eventos	
Acción del actor	Respuesta del sistema
1. Selecciona la opción descargar demo	2. Ejecuta la acción de descarga.

mediante el botón “ <i>Descargar</i> ”.	
Interfaz: (<i>Anexo 10</i>).	

Tabla 2.23 – Descripción del Caso de Uso Descargar Demo.

CU10	Gestionar Recurso	
Propósito	Agregar, editar y eliminar los recursos.	
Actores	Administrador (Inicia).	
Resumen	El Caso de Uso comienza cuando el actor decide agregar un recurso, modificar la información asociada a un recurso o eliminarlo. Permite gestionar la información relacionada con los Estándares y sus respectivos Demos.	
Precondiciones	El usuario debe estar autenticado en el Sistema con privilegios de Administrador.	
Poscondiciones	Se registra(n) lo(s) datos(s) del nuevo recurso o los cambios producidos en los datos del mismo en el sistema.	
Referencias	RF21, RF22, RF23, RF27.	
Prioridad	Crítico.	
Flujo Normal de eventos		
Sección Agregar Recurso		
Acción del actor	Respuesta del sistema	
1. Selecciona la opción Agregar Recurso.	2. Muestra la interfaz Agregar Recurso.	
3. Introduce los datos del recurso (nombre, tipo, url y descripción).	4. Al entrar al menos un dato activa el botón “ <i>Cancelar</i> ” y cuando haya entrado todos los campos de entrada obligatoria activa el “ <i>Aceptar</i> ”.	
4. Selecciona el botón “ <i>Aceptar</i> ” para agregar el recurso. En caso contrario ver Flujo alternativo 1.	13. Valida los datos entrados. 13.1. En caso de que los datos no sean correctos, ver Flujo alternativo 2. 13.2. En caso de que los datos sean correctos, se verifica si existe el recurso en la aplicación, si existe ver Flujo alternativo 3.	
	14. Registra el nuevo recurso y sus respectivos demos, terminando así el CU.	

Flujo alternativo 1	
Acción del actor	Respuesta del sistema
1. Selecciona el botón “ <i>Cancelar</i> ”.	2. Muestra el mensaje de confirmación “¿Desea cancelar la solicitud de “Agregar Recurso?””.
Flujo alternativo 2	
Acción del actor	Respuesta del sistema
	1. Muestra un mensaje de error en correspondencia con el dato incorrecto.
Flujo alternativo 3	
Acción del actor	Respuesta del sistema
	1. Muestra un mensaje de error informando que ya existe el recurso.
Sección Editar Recurso	
Acción del actor	Respuesta del sistema
1. Selecciona la opción Editar Recurso mediante el botón “Editar”.	2. Muestra la interfaz Editar Recurso.
3. Introduce el(los) dato(s) que desea modificar. 3.1. Si desea Agregar un nuevo recurso ver Sección “Agregar Recurso”. 3.2. Si desea Eliminar el Recurso ver Sección “Eliminar Recurso”. 3.3. Si desea cancelar solicitud de agregar un nuevo recurso ver Flujo alternativo 1.	3. Activa el botón Aceptar de la interfaz si al menos se efectuó un cambio.
4. Selecciona el botón “Aceptar”. En caso contrario ver Flujo alternativo1.	5. Valida los datos entrados. 5.1. Si los datos son correctos, actualiza los cambios terminando así el CU. 5.2. En caso de que los datos no sean correctos, ver Flujo alternativo 2.
Flujo alternativo 1	

Acción del actor	Respuesta del sistema
1. Selecciona el botón “Cancelar”.	2. Muestra el mensaje de confirmación “¿Desea cancelar la solicitud de “Editar Recurso?””.
Flujo alternativo 2	
Acción del actor	Respuesta del sistema
	1. Muestra un mensaje de error en correspondencia con el dato incorrecto.
Sección Eliminar Recurso	
Acción del actor	Respuesta del sistema
1. Selecciona el botón “Eliminar”.	2. Muestra el mensaje de confirmación “Está seguro que desea eliminar el recurso seleccionado”.
3. Selecciona el botón “Aceptar”. En caso contrario ver Flujo alternativo1.	4. Elimina toda la información perteneciente al recurso terminando así el CU.
Flujo alternativo 1	
Acción del actor	Respuesta del sistema
1. Selecciona el botón “Cancelar”.	2. Cierra la ventana del mensaje de confirmación y no elimina el recurso.
Interfaz: (Anexo 11).	

Tabla 2.24 – Descripción del Caso de Uso Gestionar Recurso.

CU11	Buscar Recurso
Propósito	Hacer una búsqueda o mostrar el listado de todos los recursos existentes en la aplicación.
Actores	Administrador, Visualizador (Inicia).
Resumen	El Caso de Uso comienza cuando el actor escoge la opción “Búsqueda/Listado” de recursos, el sistema muestra la interfaz con las opciones de búsqueda correspondientes, posibilitando hacer una búsqueda de recursos por (nombre y(o) tipo) obteniendo un listado con los recursos que correspondan con los parámetros de búsqueda introducidos, en caso de no querer hacer una búsqueda se le da posibilidad al actor de mostrar un listado con todos los recursos existentes en la aplicación.

Precondiciones	
Poscondiciones	
Referencias	RF24, RF25, RF27.
Prioridad	Crítico.
Flujo Normal de eventos	
Acción del actor	Respuesta del sistema
1. Selecciona la opción "Búsqueda/Listado" de Recursos.	2. Muestra la interfaz correspondiente para realizar la búsqueda de recursos por (nombre y(o) tipo).
3. Introduce los criterios de búsqueda que desee y presiona el botón "Buscar". En caso contrario ver Flujo alternativo1.	4. Muestra un listado con los resultados encontrados, ordenados por nombre en forma ascendente. Si son más de 10 los resultados, activa un paginado que permitirá al usuario moverse por todos los resultados encontrados, puesto que estos se muestran de 10 en 10. El paginado tiene la siguiente estructura: "Página #" Combo box para escoger de forma directa la página que desee el usuario visualizar. "Primero" Botón para escoger de forma directa el primer listado (los primeros 10 resultados). "Anterior" Botón para escoger el listado anterior (los 10 resultados anteriores). "Próximo" Botón para escoger el próximo listado (los 10 resultados siguientes). "Último" Botón para escoger de forma directa el último listado de resultados.
5. El actor selecciona el recursos que desea, marcando el radio button.	6. Ilumina la fila en la que se encuentra el recurso seleccionado.
7. Si desea descargar un recurso ver CU12 "Descargar Demo". 7.2. Si desea editar un recurso ver CU11	

<p>“Gestionar Recurso” – Sesión “Editar Recurso”.</p> <p>7.3. Si desea eliminar un recurso ver CU11</p> <p>“Gestionar Recurso” – Sesión “Eliminar Recurso”.</p>	
<p>Flujo alternativo 1</p>	
<p>Acción del actor</p>	<p>Respuesta del sistema</p>
<p>1. Presiona el botón “Mostrar Todos”.</p>	<p>2. Muestra un listado con todos los recursos existentes en la aplicación, ordenados por nombre en forma ascendente.</p> <p>Si son más de 10 los resultados, activa un paginado que permitirá al usuario moverse por todos los resultados encontrados, puesto que estos se muestran de 10 en 10. El paginado tiene la siguiente estructura:</p> <p>“Página #” Combo box para escoger de forma directa la página que desee el usuario visualizar.</p> <p>“Primero” Botón para escoger de forma directa el primer listado (los primeros 10 resultados).</p> <p>“Anterior” Botón para escoger el listado anterior (los 10 resultados anteriores).</p> <p>“Próximo” Botón para escoger el próximo listado (los 10 resultados siguientes).</p> <p>“Último” Botón para escoger de forma directa el último listado de resultados.</p>
<p>Interfaz: (Anexo 12).</p>	

Tabla 2.25 – Descripción del Caso de Uso Buscar Recurso.

CU12	Descargar Recurso	
Propósito	Descargar de la aplicación un recurso específico.	
Actores	Administrador, Visualizador (Inicia).	
Resumen	El Caso de Uso comienza cuando el actor decide descargar un recurso y termina cuando se completa la operación de descarga.	
Precondiciones		
Poscondiciones		
Referencias	RF26.	
Prioridad	Secundario.	
Flujo Normal de eventos		
Acción del actor	Respuesta del sistema	
1. Selecciona la opción descargar demo mediante el vínculo en el nombre del recurso.	2. Ejecuta la acción de descarga.	
Interfaz: (Anexo 13).		

Tabla 2.26 – Descripción del Caso de Uso Descargar Recurso.

2.5 Conclusiones.

En el presente capítulo se realizó la descripción de las características del sistema. Se obtuvo un modelo de dominio partiendo de la definición de conceptos. Se precisaron los requerimientos del sistema, tanto funcionales como no funcionales, tomando como punto de partida para ellos la modelación del dominio realizado.

Se definieron los Casos de Uso y mediante un Diagrama de Casos de Uso del Sistema, se ilustraron las relaciones de los actores que interactúan con el sistema, y las secuencias de acciones que se realizan. Por último se realizó la descripción de cada uno de los casos de usos definidos.

CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL SISTEMA.

3.1 Introducción.

En este capítulo se modelan los artefactos relacionados con el flujo de trabajo Análisis y Diseño. Para ello los componentes de la aplicación se tratan como clases, y utilizando las extensiones del UML, se presentan a través de diagramas de clases Web. Además se presenta el diagrama de clases persistentes y el modelo de datos que es la base para construir la base de datos que soportará toda la información del sistema, y se termina con la descripción de sus tablas.

3.2 Modelo de Análisis.

En este flujo se refinan y estructuran los requisitos, profundizándose en el dominio de la aplicación. Permitiendo una mayor comprensión del problema para modelar la solución y ofreciendo un mayor poder expresivo y una mayor formalización. El análisis consiste en obtener una visión del sistema que se preocupa de ver qué hace, de modo que sólo se interesa por los requisitos funcionales.

3.2.1 Diagramas de Clases del Análisis.

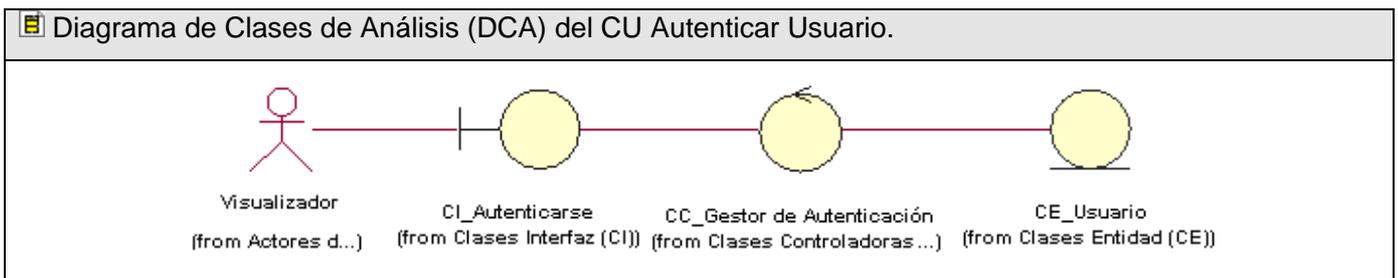


Figura 3.1 – DCA del CU Autenticar Usuario.

DCA del CU Gestionar Usuario.

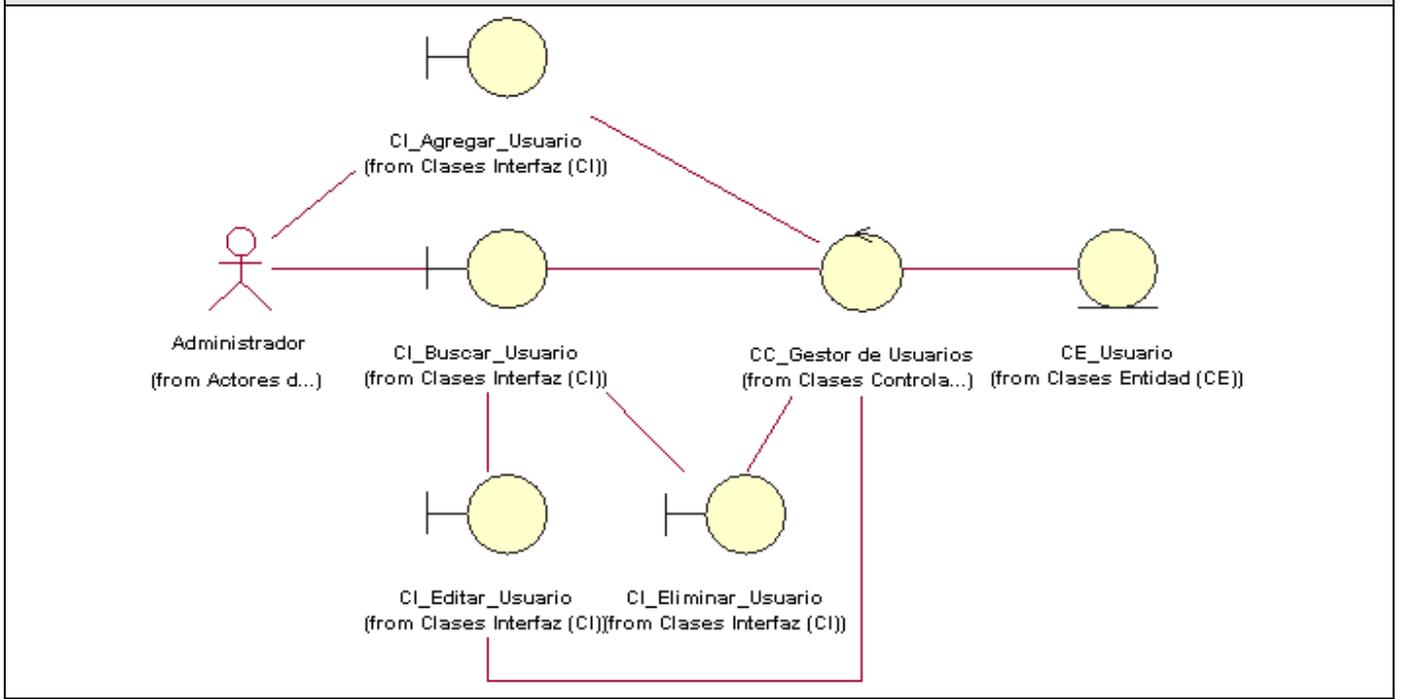


Figura 3.2 – DCA del CU Gestionar Usuario.

DCA del CU Editar Perfil.

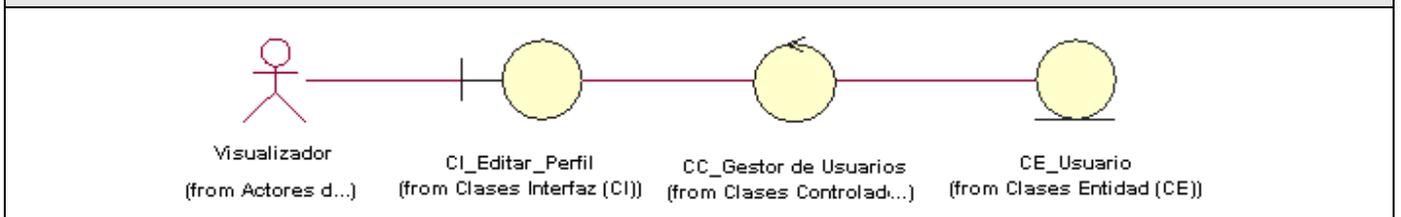


Figura 3.3 – DCA del CU Editar Perfil.

DCA del CU Gestionar Estándar.

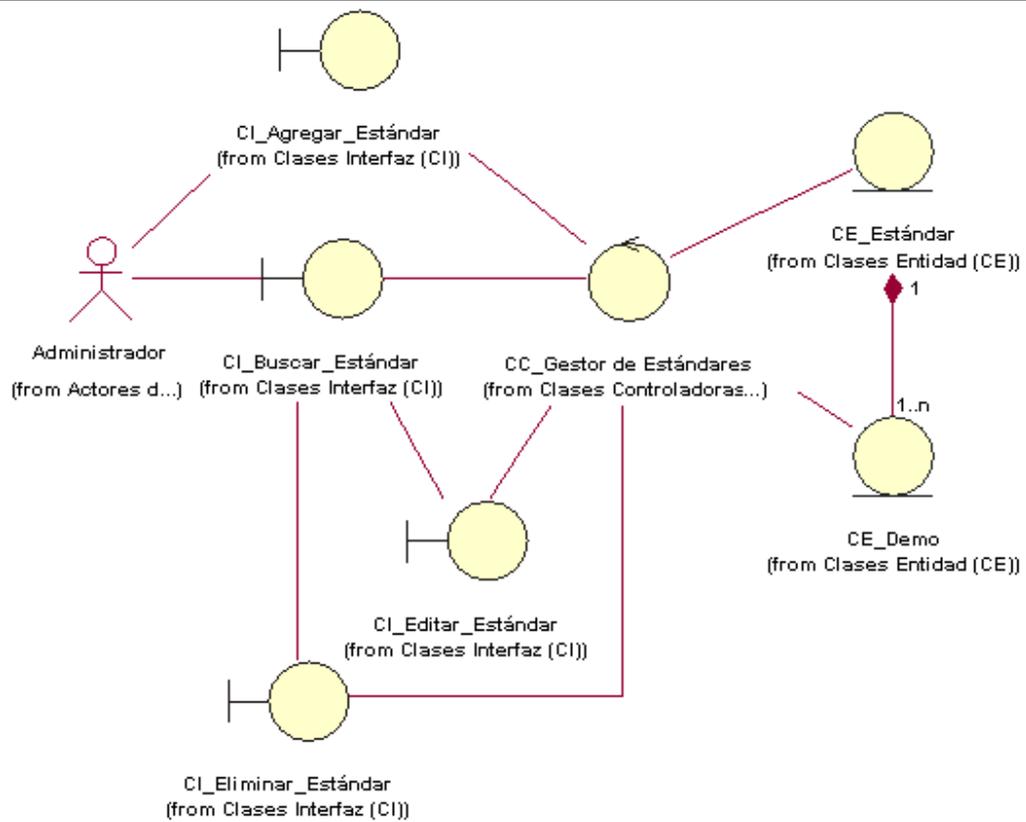


Figura 3.4 – DCA del CU Gestionar Estándar.

DCA del CU Buscar Estándar, Visualizar Estándar, Visualizar Demo y Descargar Demo.

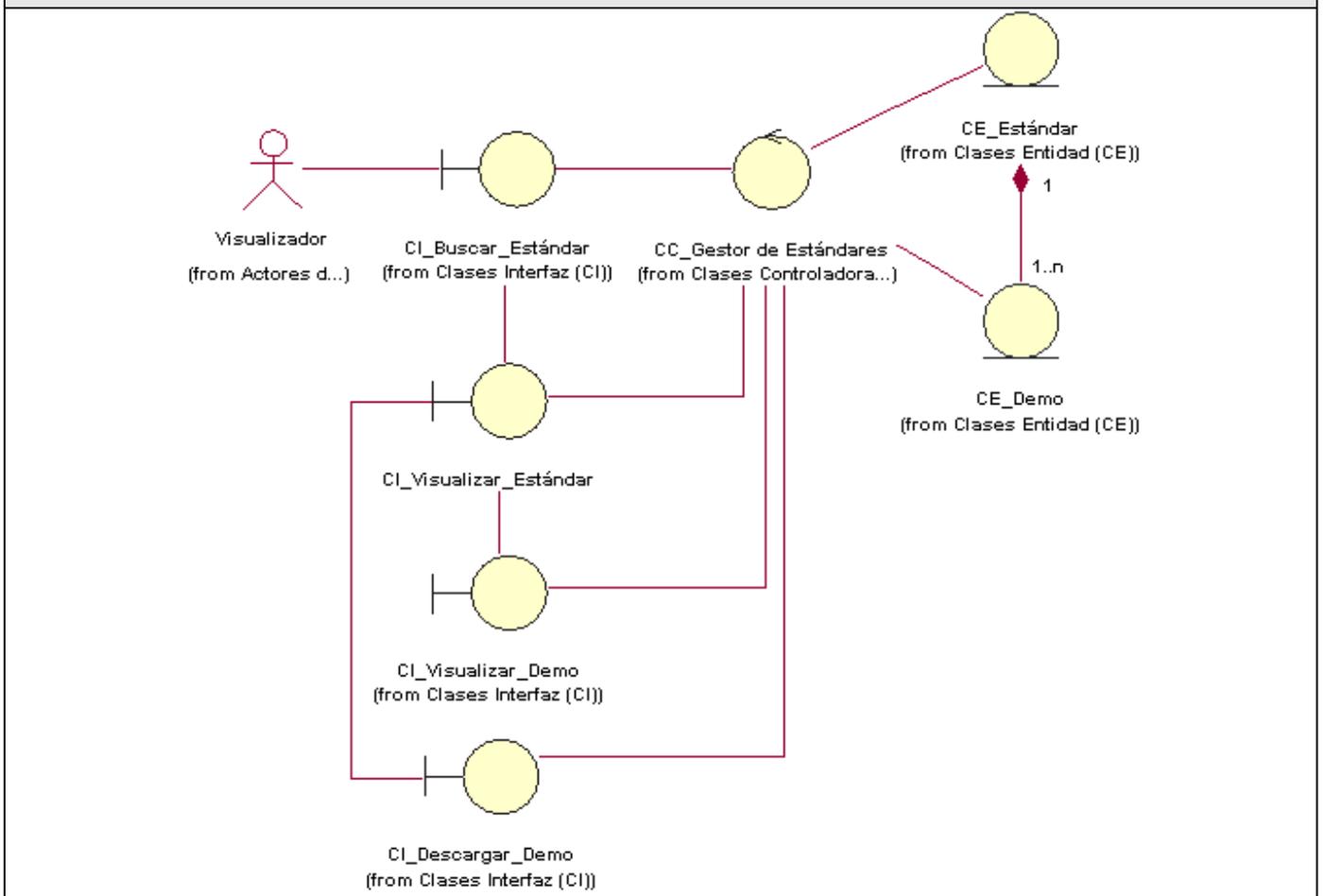


Figura 3.5 – DCA del CU Buscar Estándar, Visualizar Estándar, Visualizar Demo y Descargar Demo.

DCA del CU Buscar Demo.

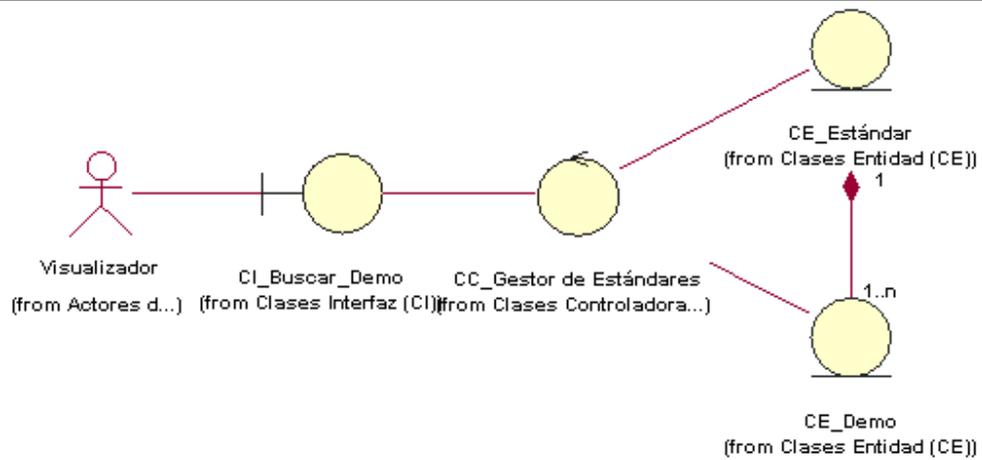


Figura 3.6 – DCA del CU Buscar Demo.

DCA del CU Gestionar Recurso.

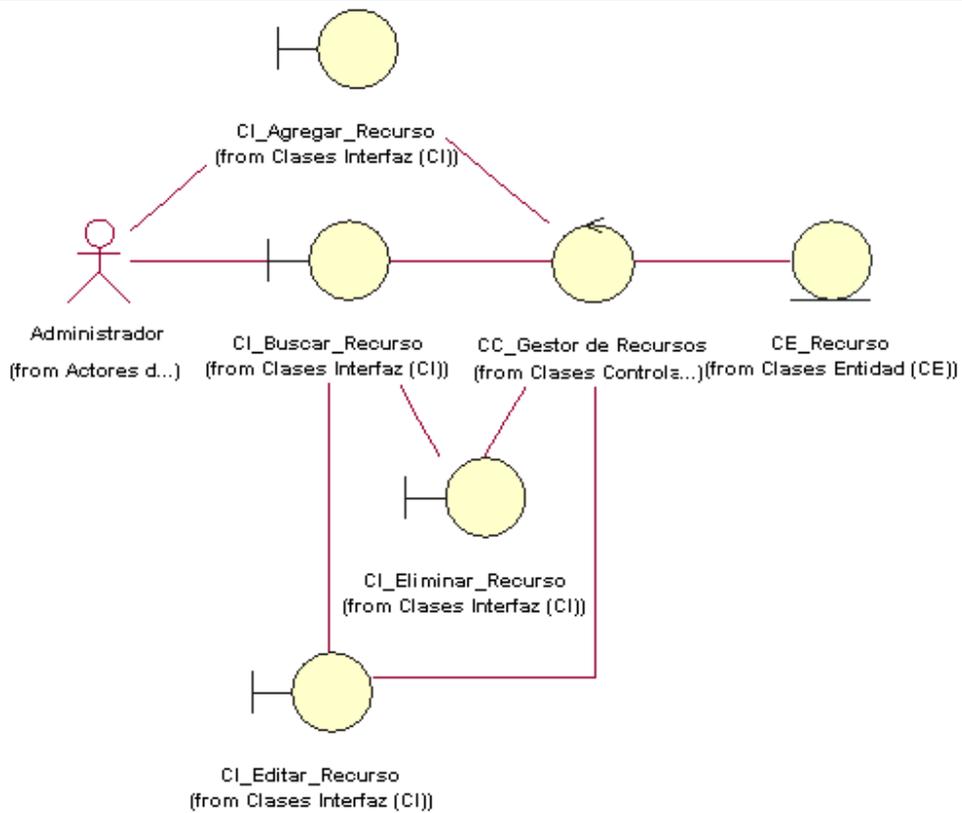


Figura 3.7 – DCA del CU Gestionar Recurso.

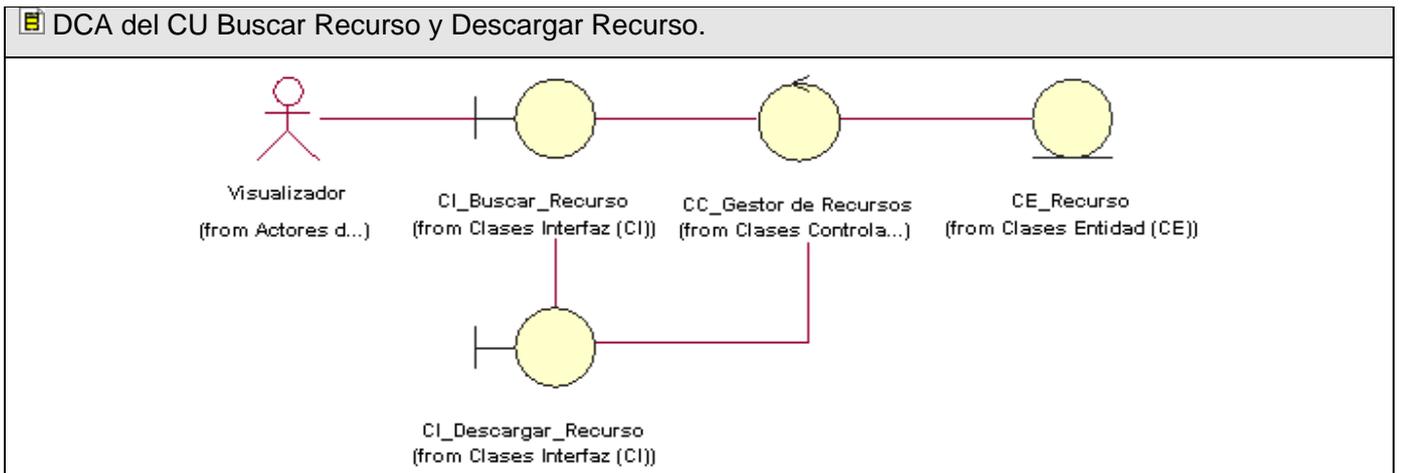


Figura 3.8 – DCA del CU Buscar Recurso y Descargar Recurso.

3.3 Modelo de Diseño.

El modelo de diseño es un refinamiento del análisis que tiene en cuenta los requisitos no funcionales, es decir cómo cumple el sistema sus objetivos.

3.3.1 Justificación del uso de patrones.

El uso correcto de los patrones de diseño es una buena práctica para el desarrollo de sistemas, estos facilitan la reusabilidad, extensibilidad y mantenimiento del mismo. Una arquitectura sólida, sustentada con un buen lenguaje de patrones, es crucial para dar cumplimiento los procesos de negocios.

Por lo citado anteriormente se utiliza en el desarrollo de la solución propuesta el Patrón de Capas, es un patrón arquitectónico y su aplicación en el sistema se centra en descomponer la aplicación en tres capas o servicios independientes, la capa de usuario, la de negocio y la de datos, facilitando la implementación de los distintos componentes en distintas máquinas y la posible reutilización de los mismos.

Se tuvieron en cuenta del mismo modo los siguientes patrones de diseño: Modelo-Vista-Controlador, Patrón Experto, Patrón Creador, Patrón Alta Cohesión y Patrón Bajo Acoplamiento.

3.3.2 Definición de estructura del diseño.

La definición de la estructura de diseño es primordial para la arquitectura de la aplicación. Para mayor comprensión se define un modelo de subsistemas de servicios el cuál se presenta a continuación así como la descripción de los paquetes que lo componen.

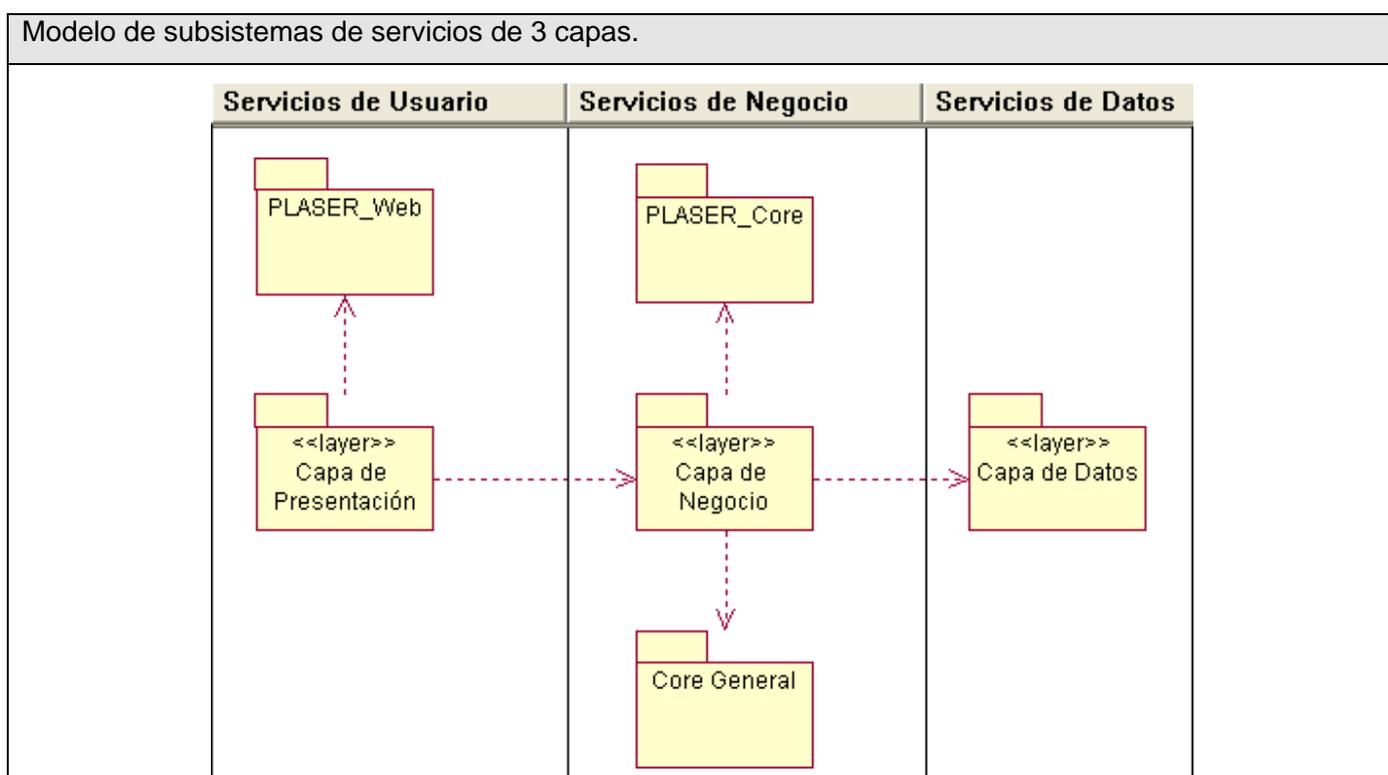


Figura 3.9 – Modelo de subsistemas de servicios de 3 capas.

Capa de Presentación: Representa la parte del sistema con la que interactúa el usuario. Contiene los XSL utilizados para las transformaciones de los documentos XML y los ficheros JavaScript usados en las validaciones y funcionalidades de los formularios de las páginas XHTML a las que accede el usuario.

PLASER_Web: Agrupa un conjunto de clases de la plataforma de servicios PLASER que son utilizadas en la implementación de la capa de presentación, entre ellas Plaser_XML, Plaser_XSLT.

Capa de Negocio: Establece la comunicación entre la Capa de Negocio y la Capa de Datos. Contiene los métodos del negocio de la aplicación.

PLASER_Core: Contiene conjunto de clases de la plataforma de servicios PLASER que son consumidas en la implementación de la capa de negocio, entre ellas dbz_class, confplaser_server.

Core General: Contiene los ficheros de validación utilizados en la Capa de Negocio.

Capa de Datos: Este paquete representa a la capa de datos, administra la información que maneja el sistema. Esto incluye el almacenamiento, la actualización y la consulta de todos los datos contenidos en el sistema, es por esto que contiene a la base de datos y las clases de acceso a la misma.

3.3.3 Diagramas de clases del diseño.

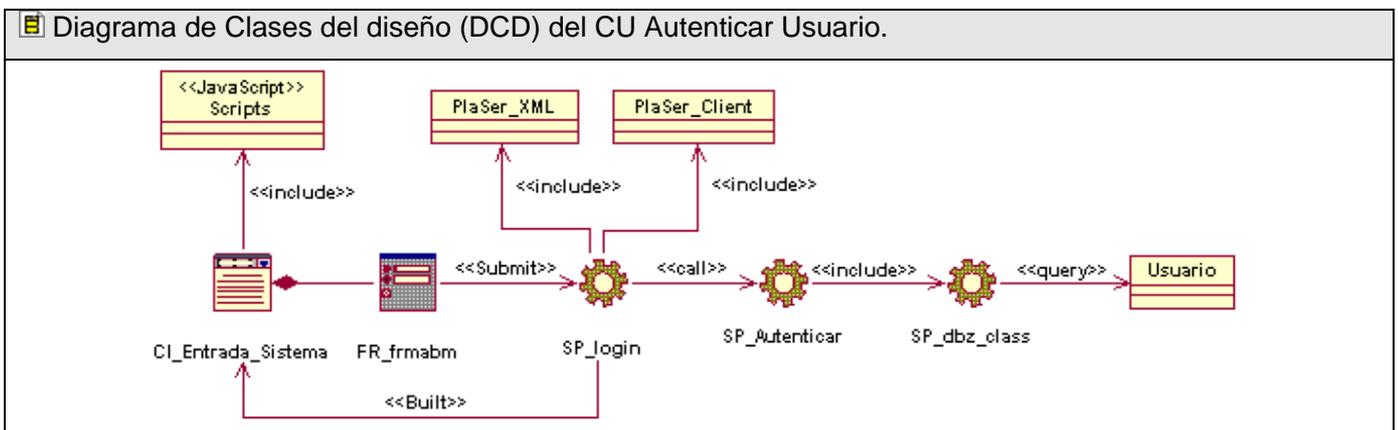


Figura 3.10 – DCD del CU Autenticar Usuario.

DCD del CU Gestionar Usuario.

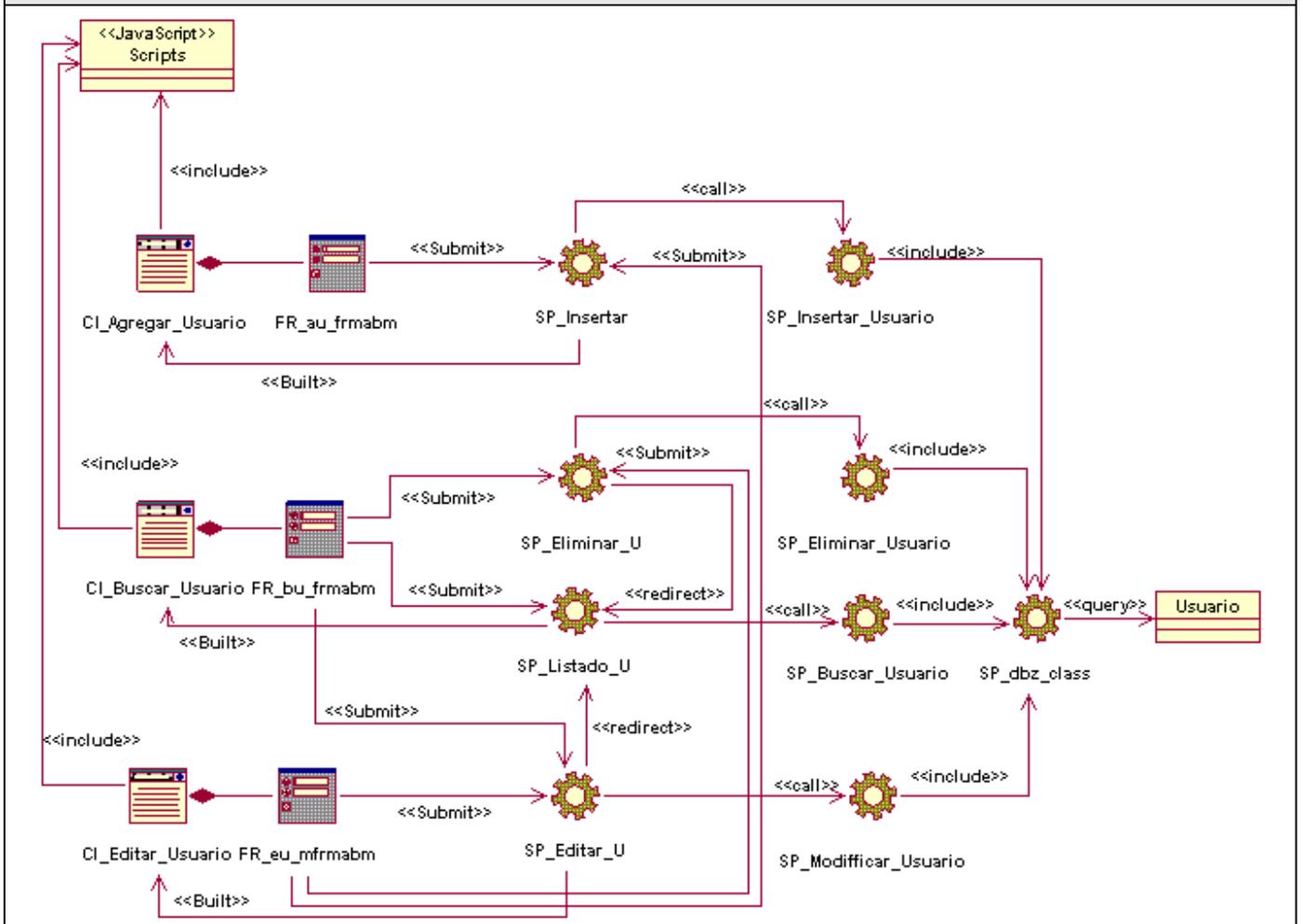


Figura 3.11 – DCD del CU Gestionar Usuario.

DCD del CU Gestionar Estándar.

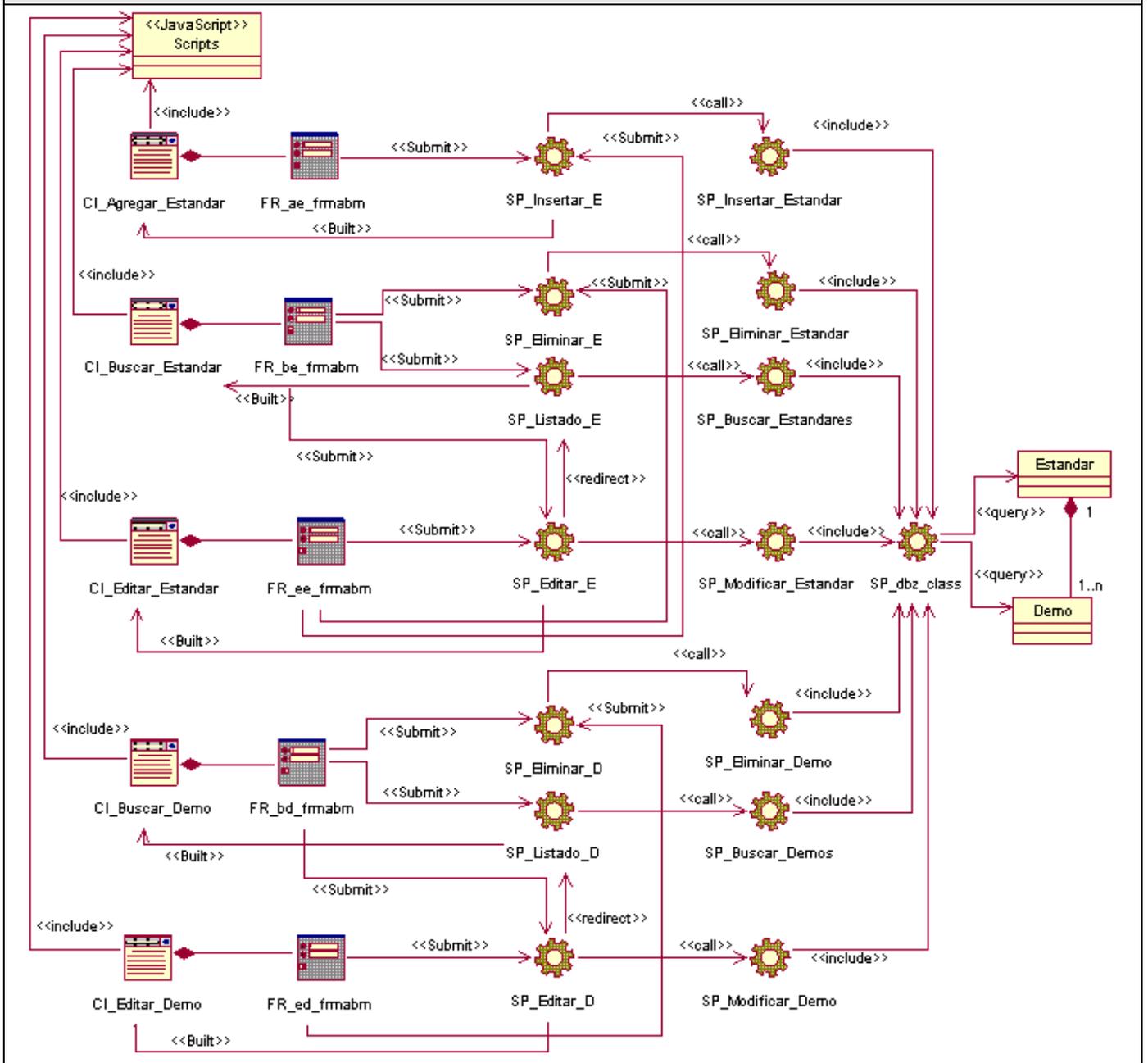


Figura 3.12 – DCD del CU Gestionar Estándar.

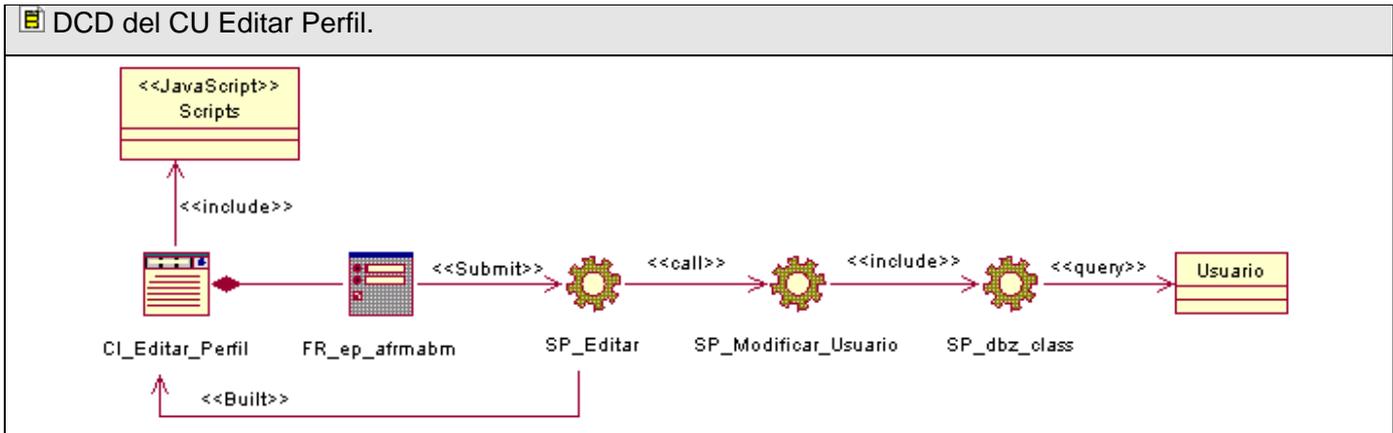


Figura 3.13 – DCD del CU Gestionar Editar Perfil.

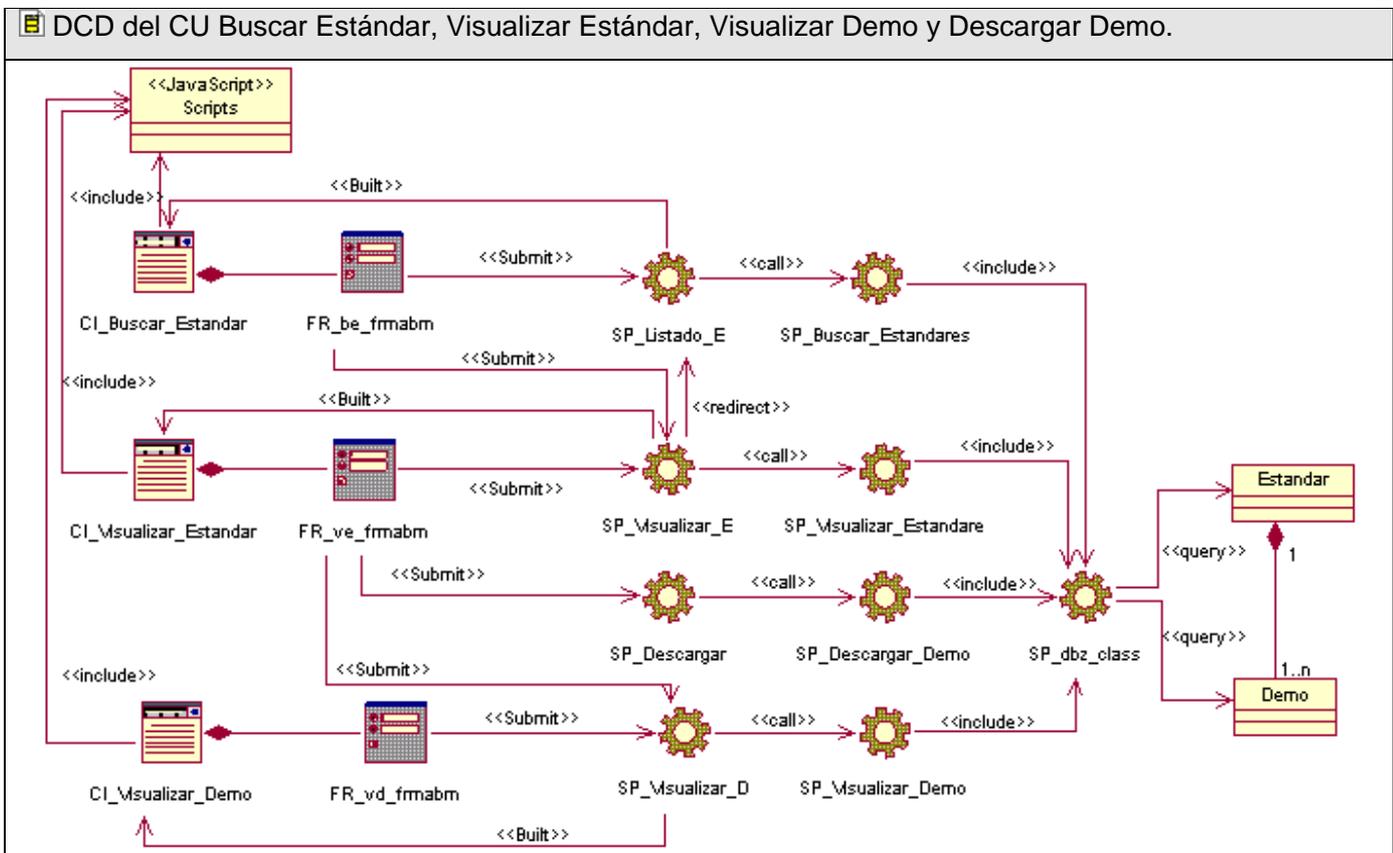


Figura 3.14 – DCD del CU Buscar Estándar, Visualizar Estándar, Visualizar Demo y Descargar Demo.

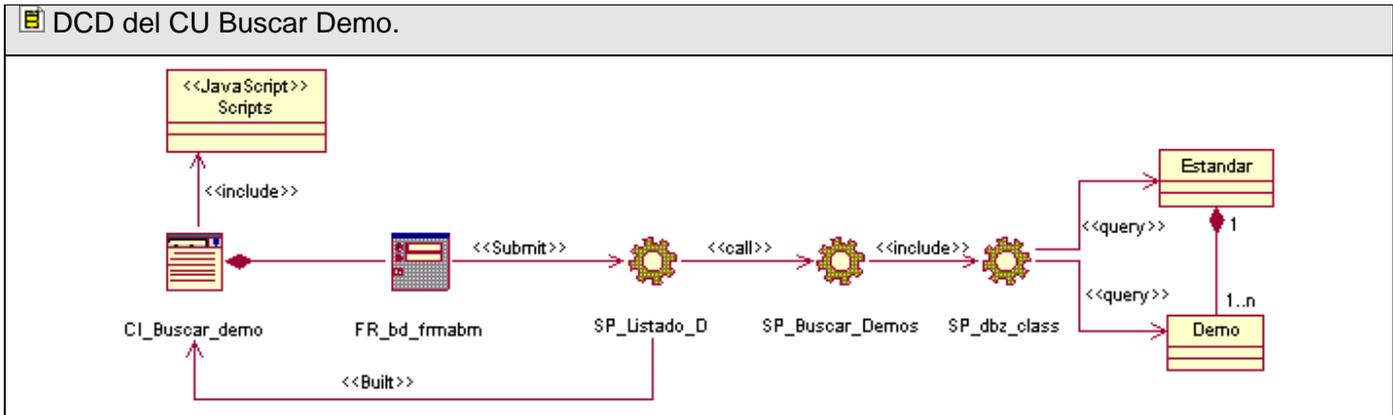


Figura 3.15 – DCD del CU Buscar Demo.

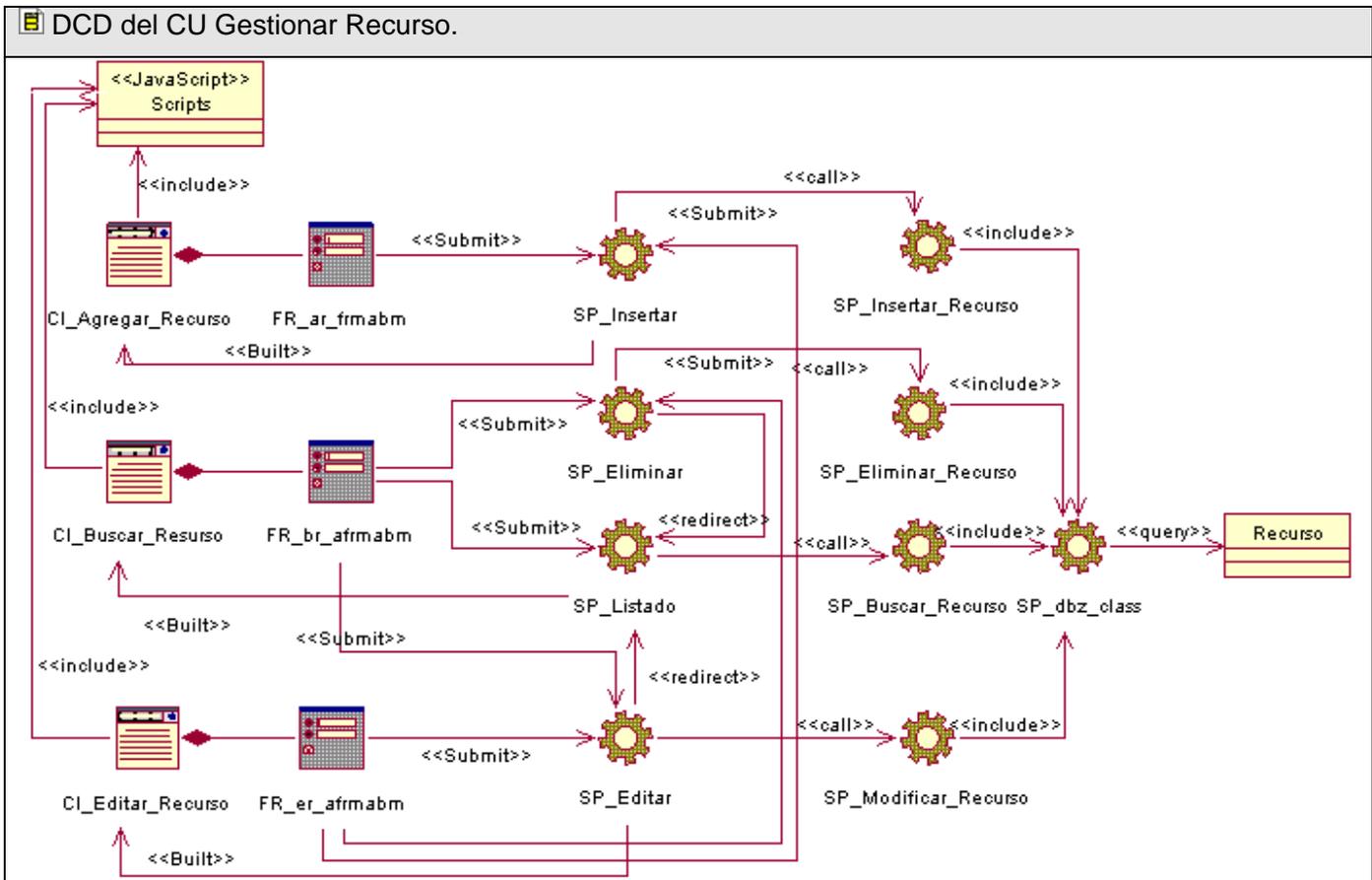


Figura 3.16 – DCD del CU Gestionar Recurso.

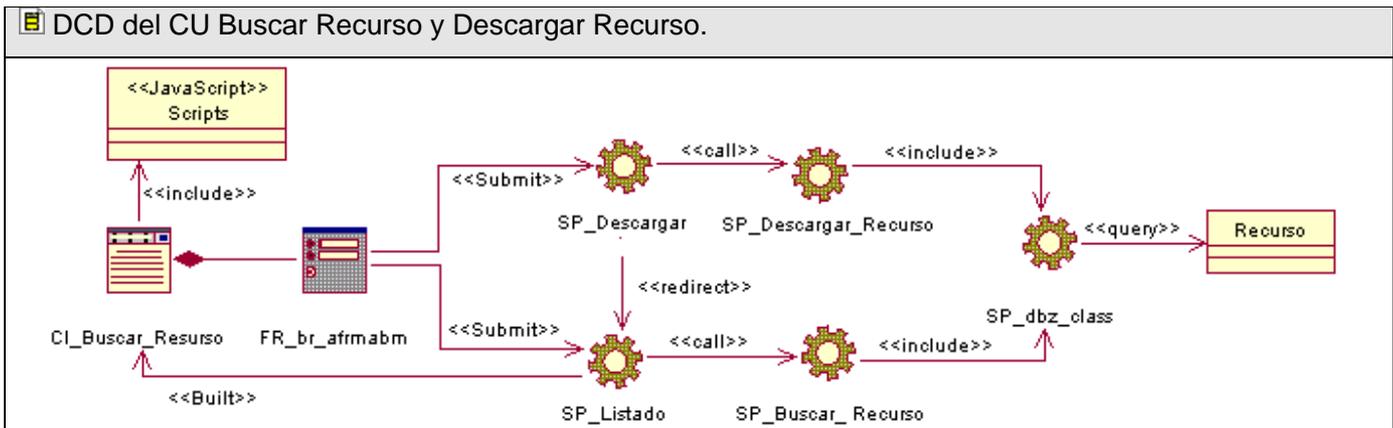


Figura 3.17 – DCD del CU Gestionar Recurso.

3.3.4 Diagramas de interacción.

Los diagramas de interacción se dividen en dos tipos de diagramas de UML, los diagramas de secuencia y los diagramas de colaboración. Para modelar los aspectos dinámicos de este sistema se utilizaron diagramas de secuencia por cada caso de uso.

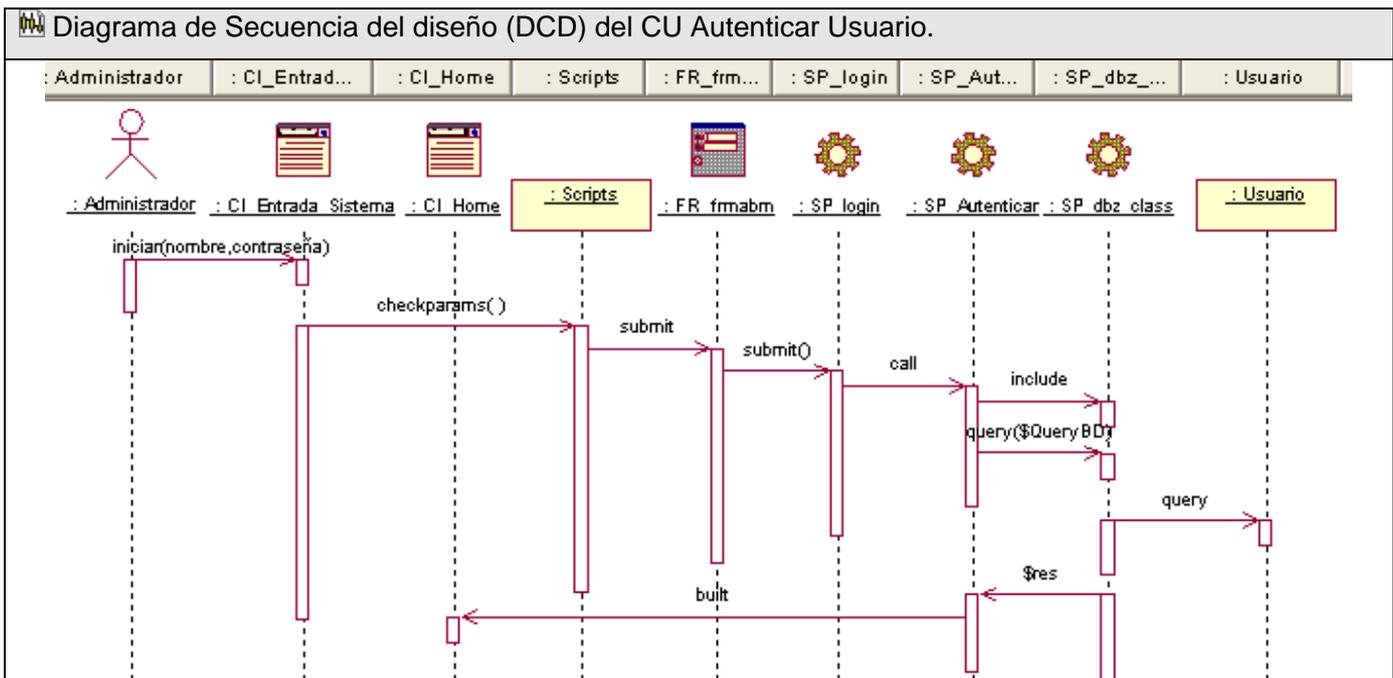


Figura 3.18 – DSD del CU Autenticar Usuario.

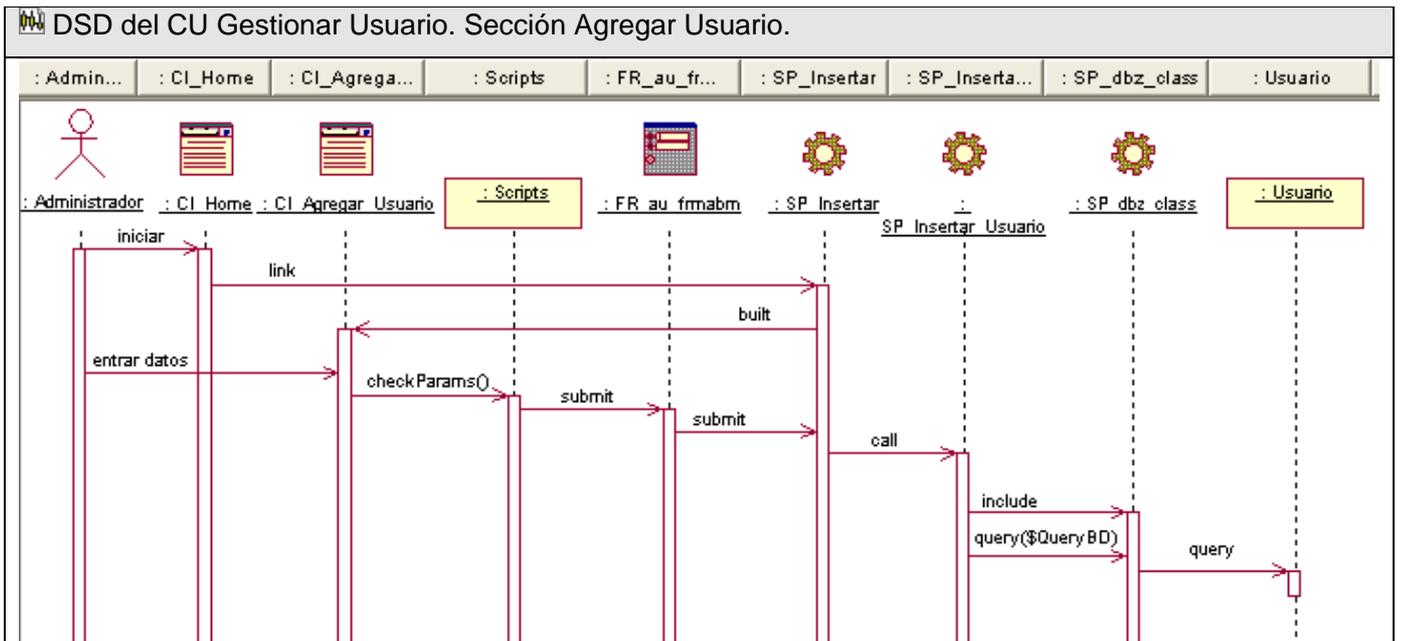


Figura 3.19 – DSD del CU Gestionar Usuario. Sección Agregar Usuario.

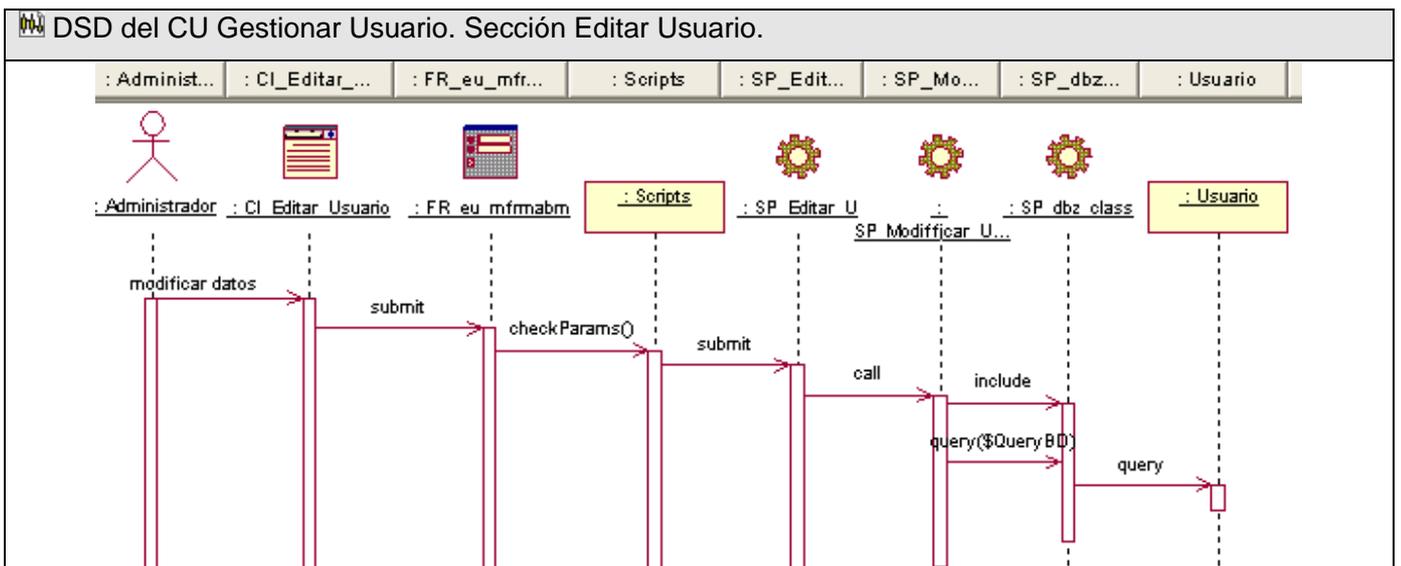


Figura 3.20 – DSD del CU Gestionar Usuario. Sección Editar Usuario.

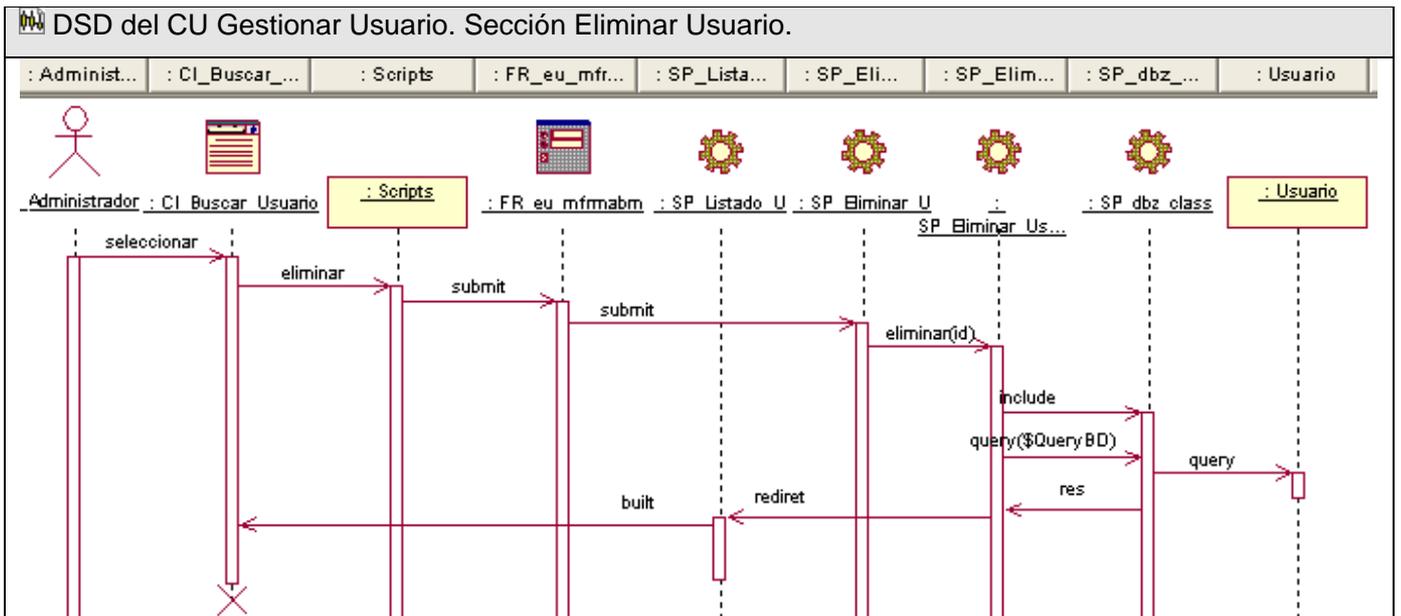


Figura 3.21 – DSD del CU Gestionar Usuario. Sección Eliminar Usuario.

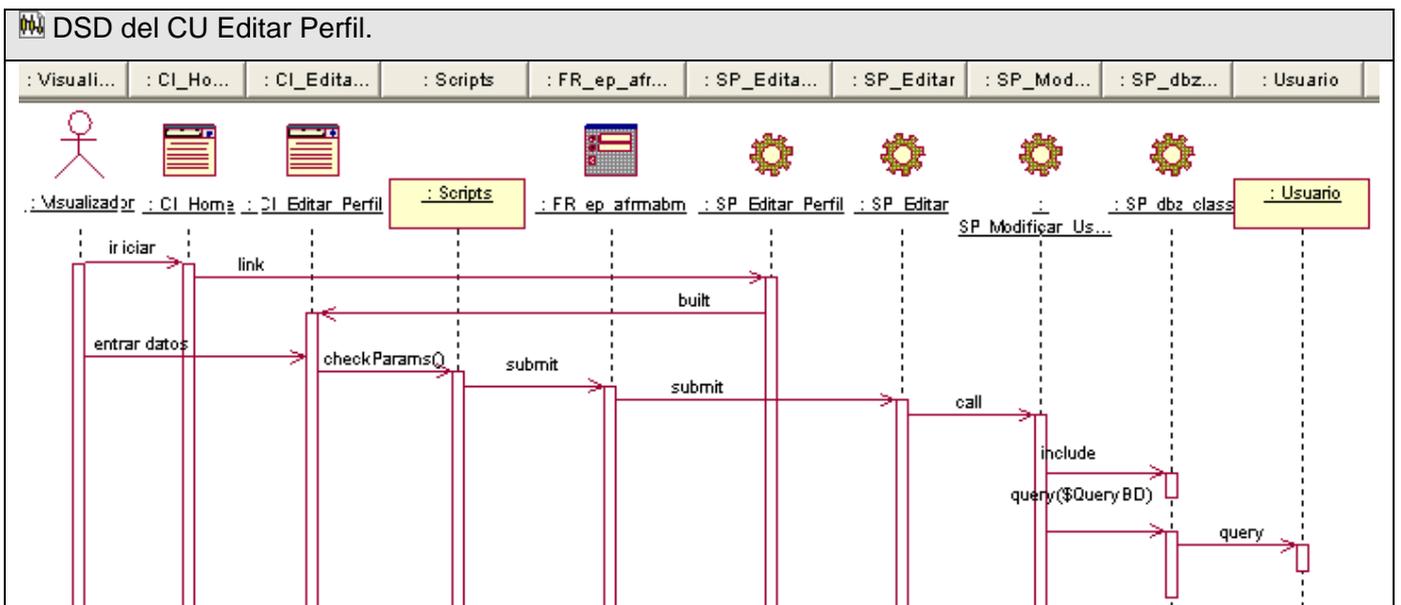


Figura 3.22 – DSD del CU Gestionar Editar Perfil.

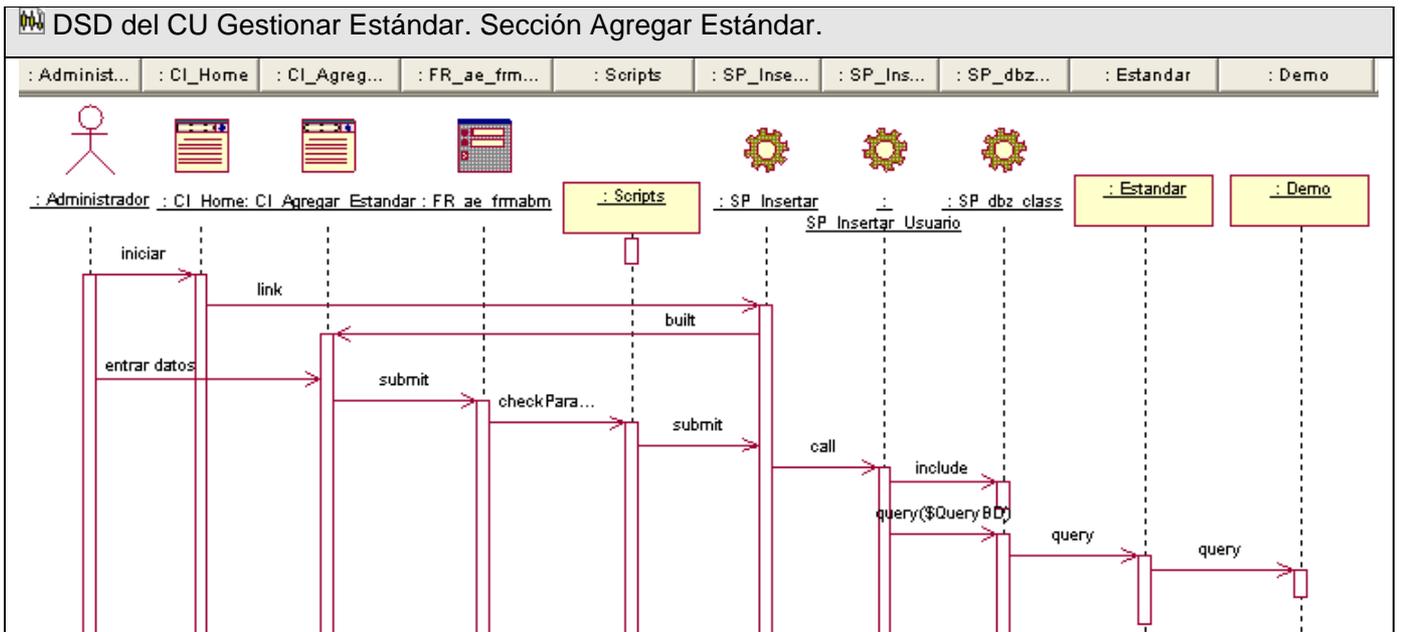


Figura 3.23 – DSD del CU Gestionar Estándar. Sección Agregar Estándar.

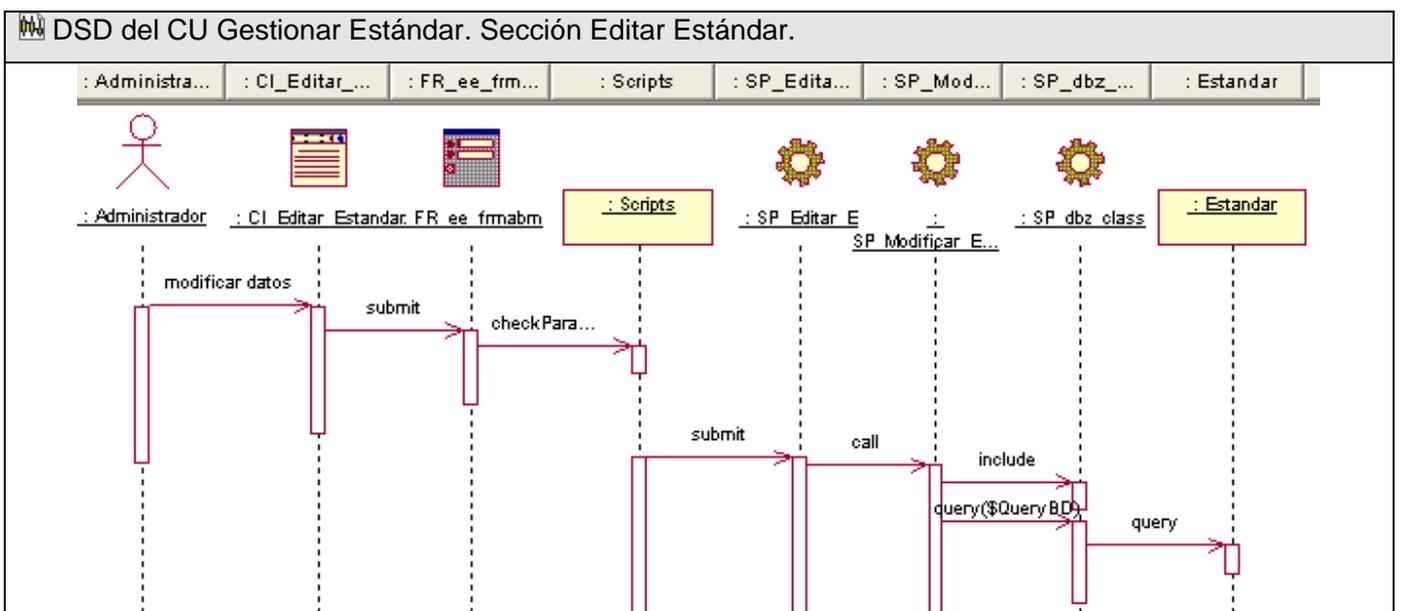


Figura 3.24 – DSD del CU Gestionar Estándar. Sección Editar Estándar.

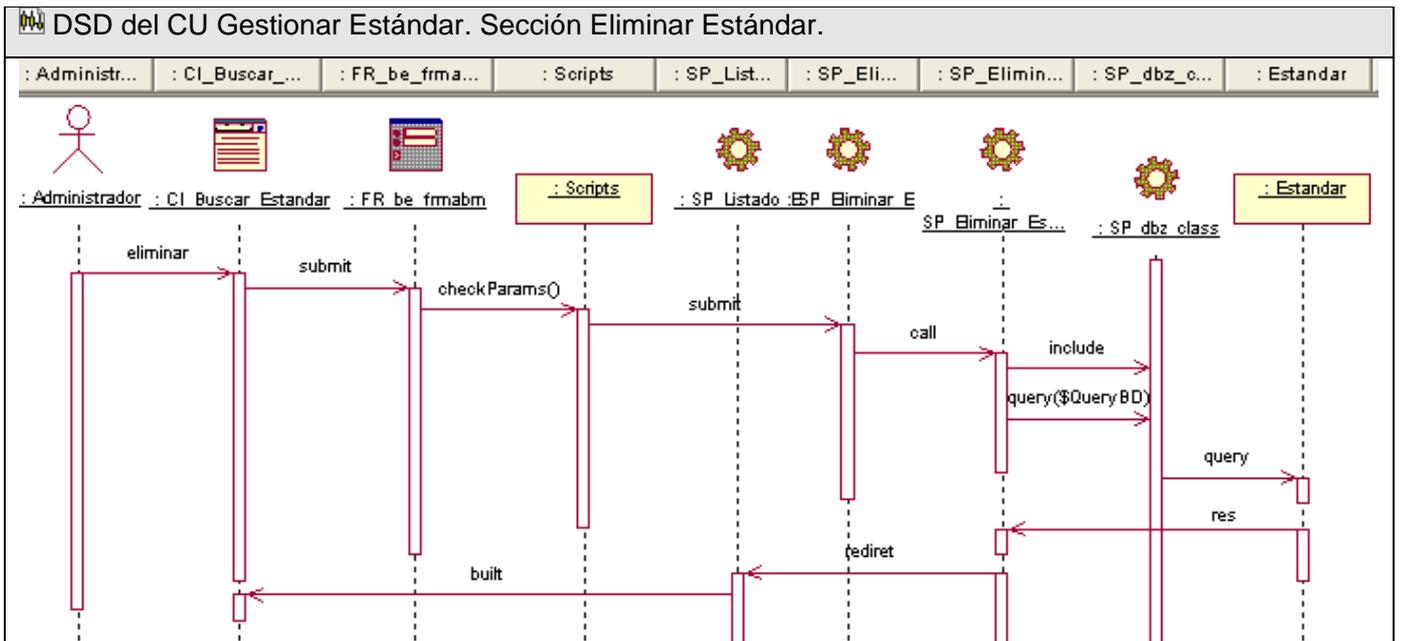


Figura 3.25 – DSD del CU Gestionar Estándar. Sección Eliminar Estándar

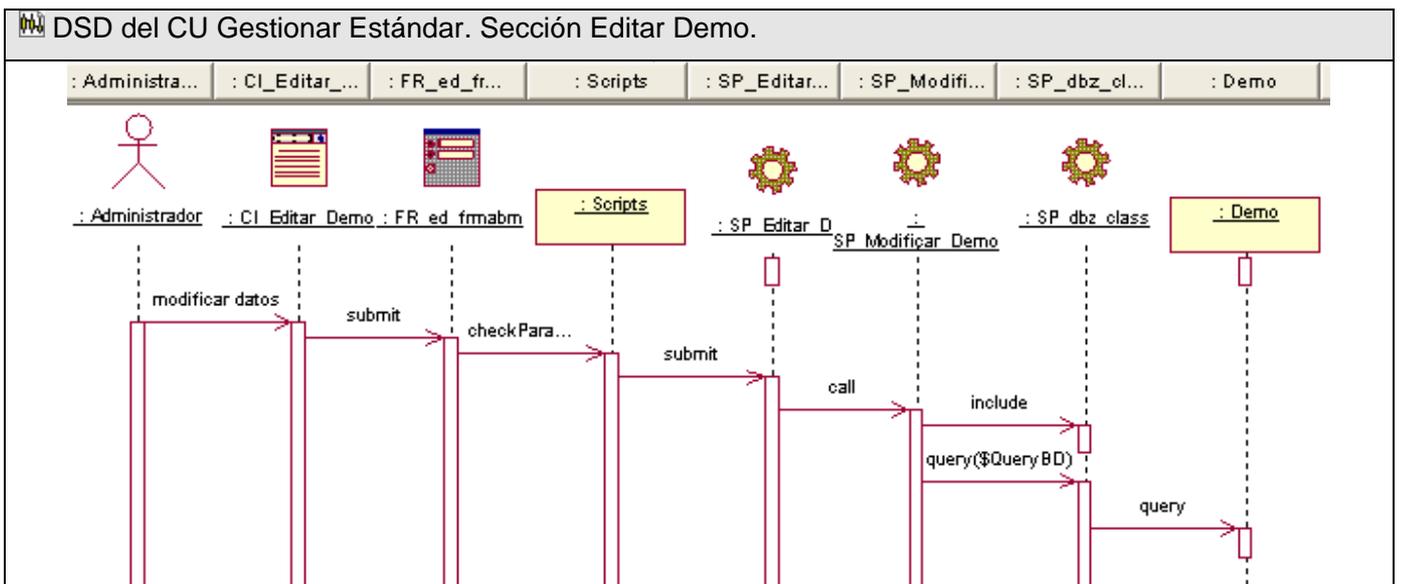


Figura 3.26 – DSD del CU Gestionar Estándar. Sección Editar Demo.

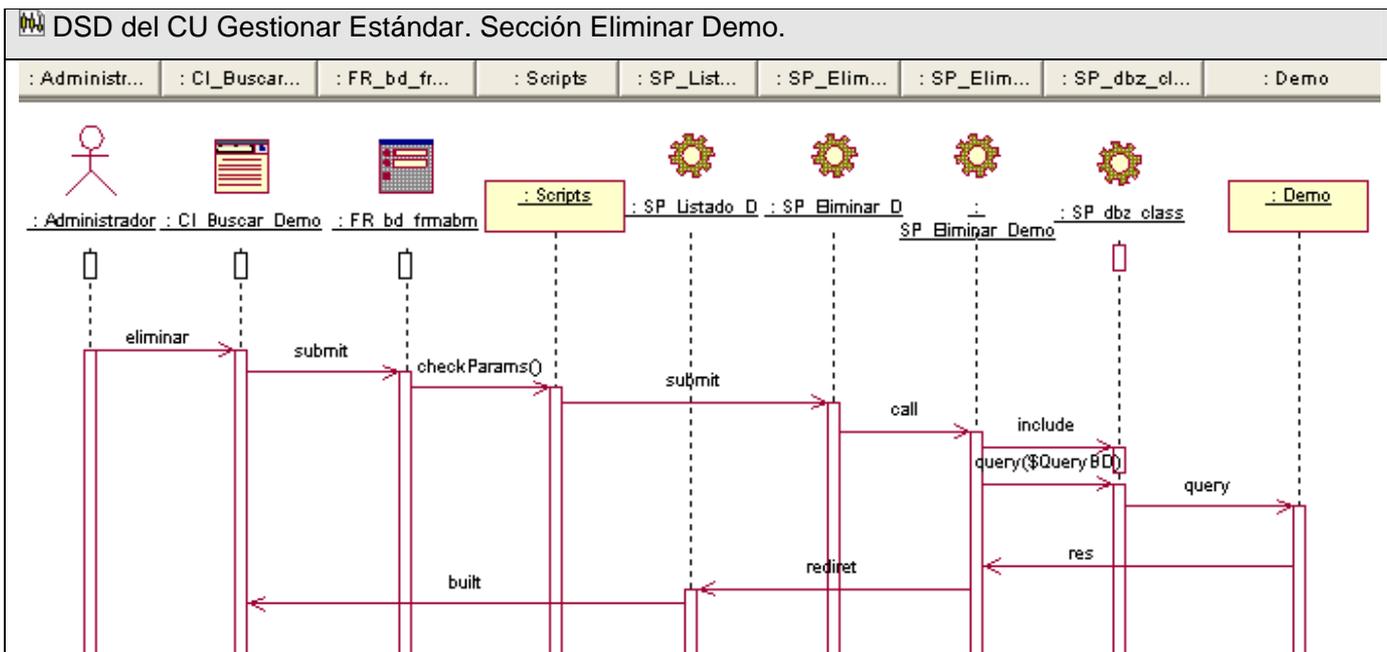


Figura 3.27 – DSD del CU Gestionar Estándar. Sección Eliminar Demo.

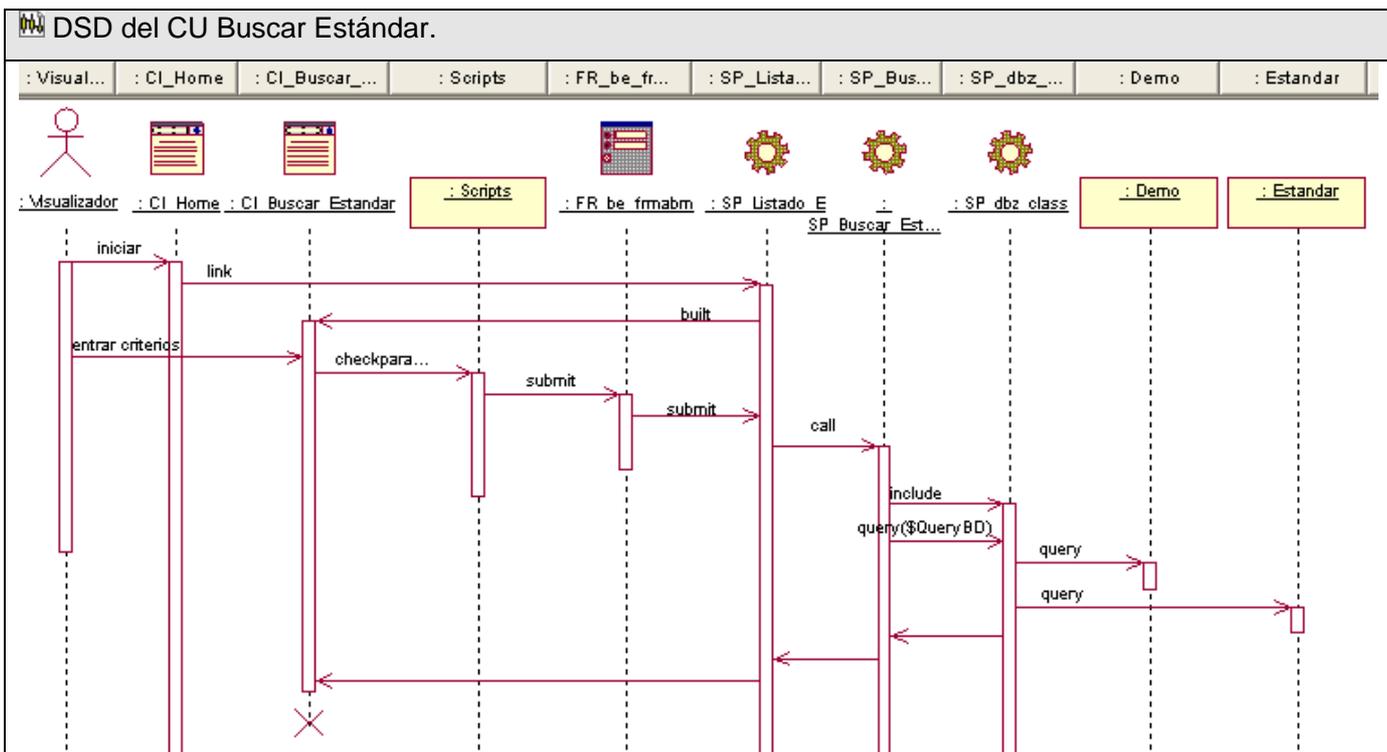


Figura 3.28 – DSD del CU Buscar Estándar.

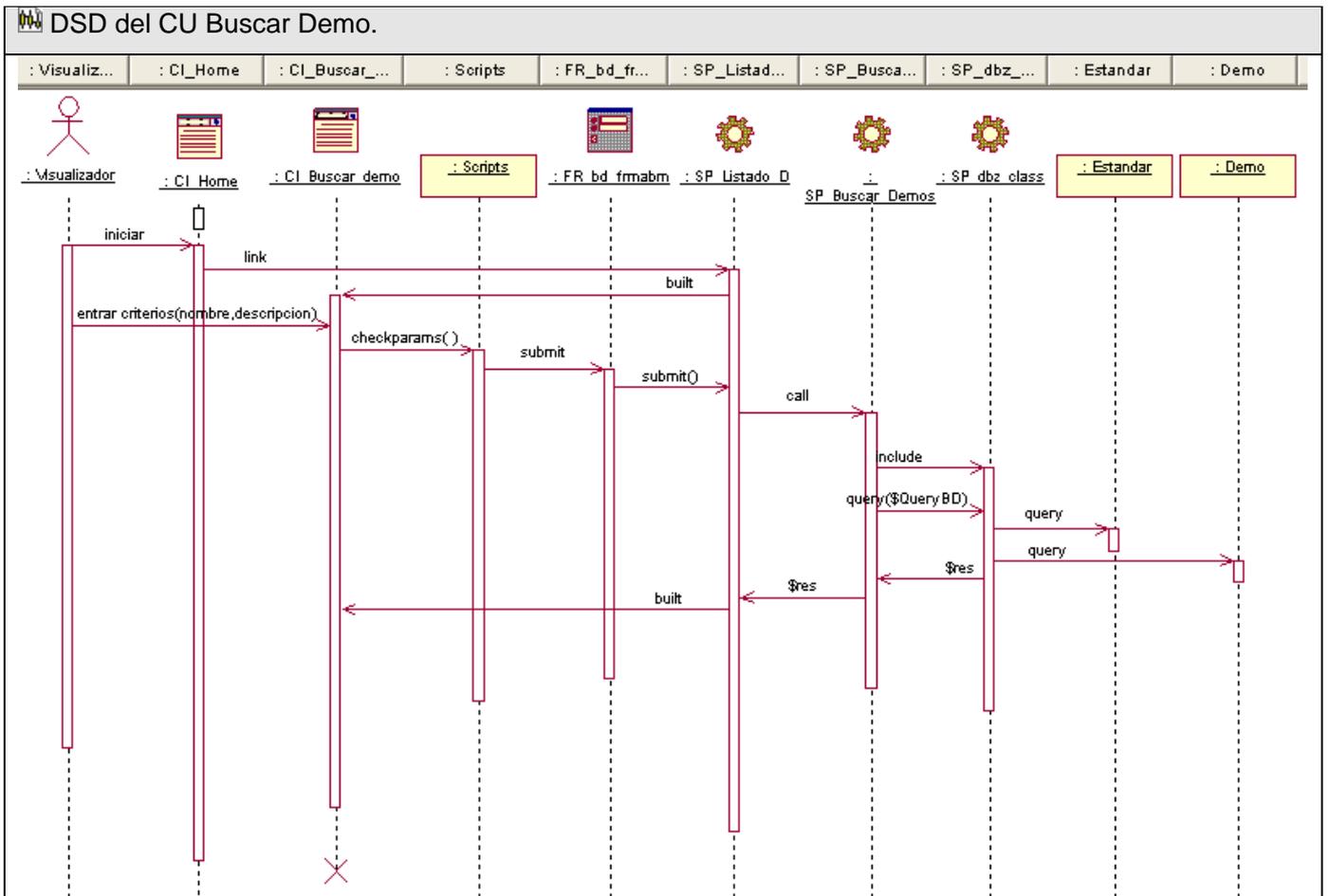


Figura 3.29 – DSD del CU Buscar Demo.

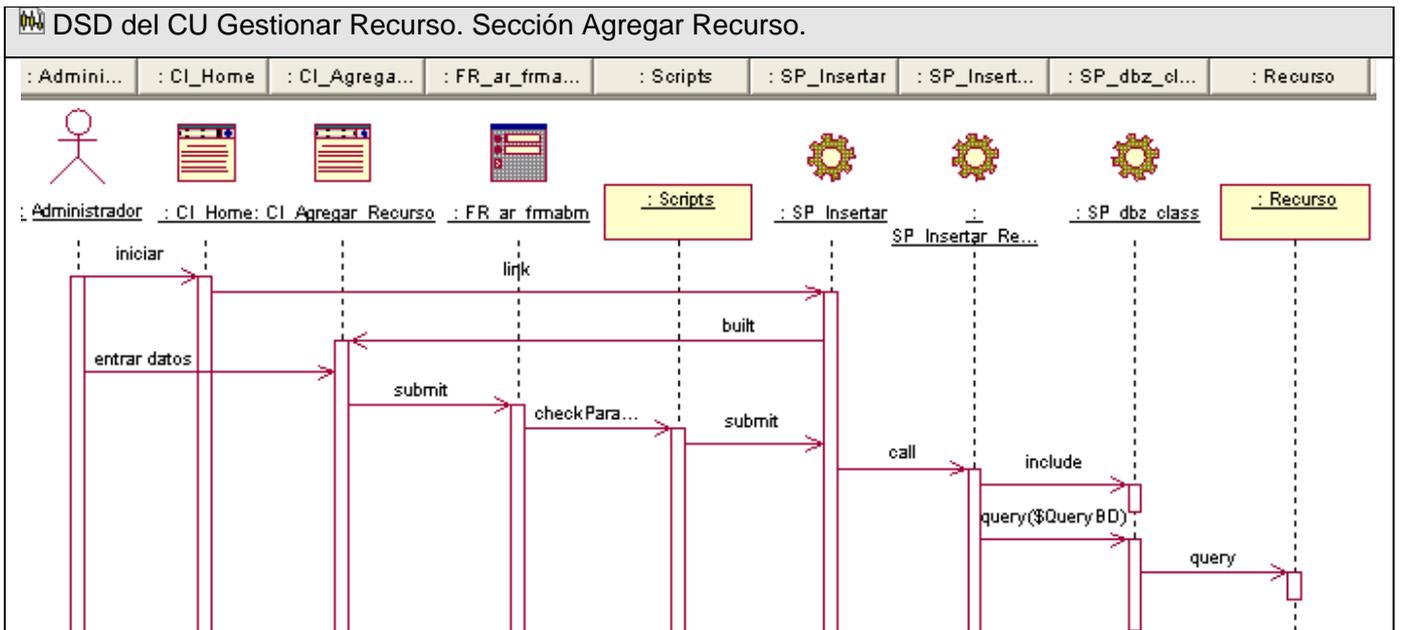


Figura 3.30 – DSD del CU Gestionar Recurso. Sección Agregar Recurso.

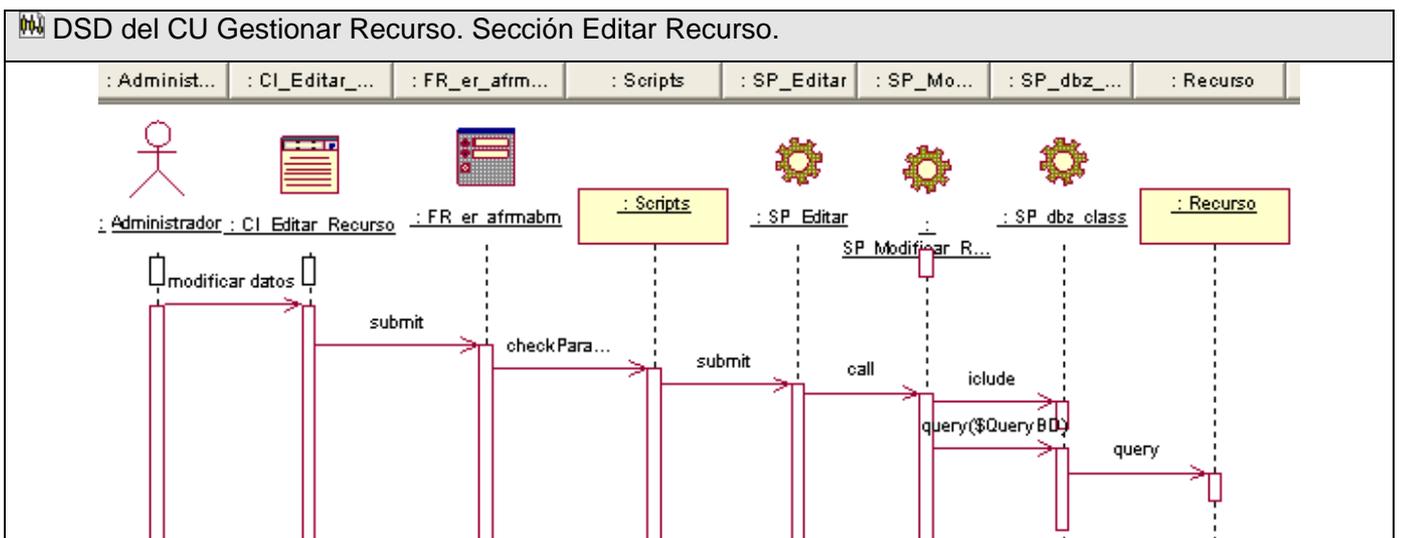


Figura 3.31 – DSD del CU Gestionar Recurso. Sección Editar Recurso.

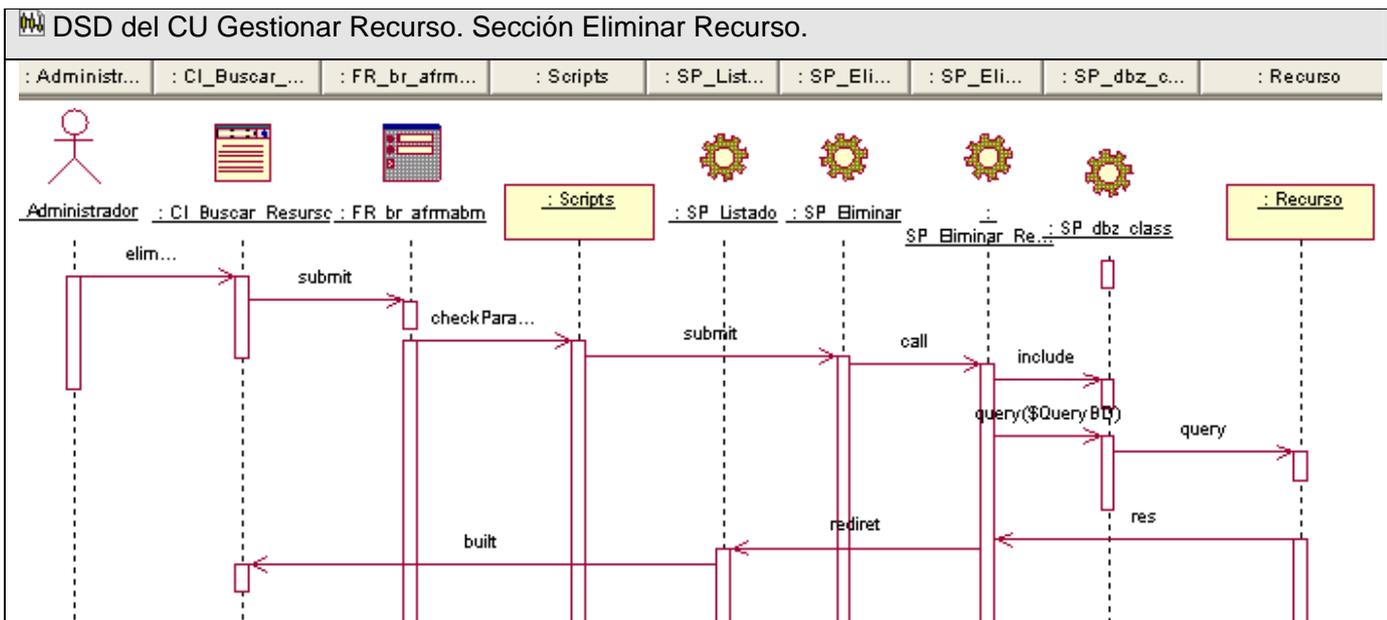


Figura 3.32 – DSD del CU Gestionar Recurso. Sección Eliminar Recurso.

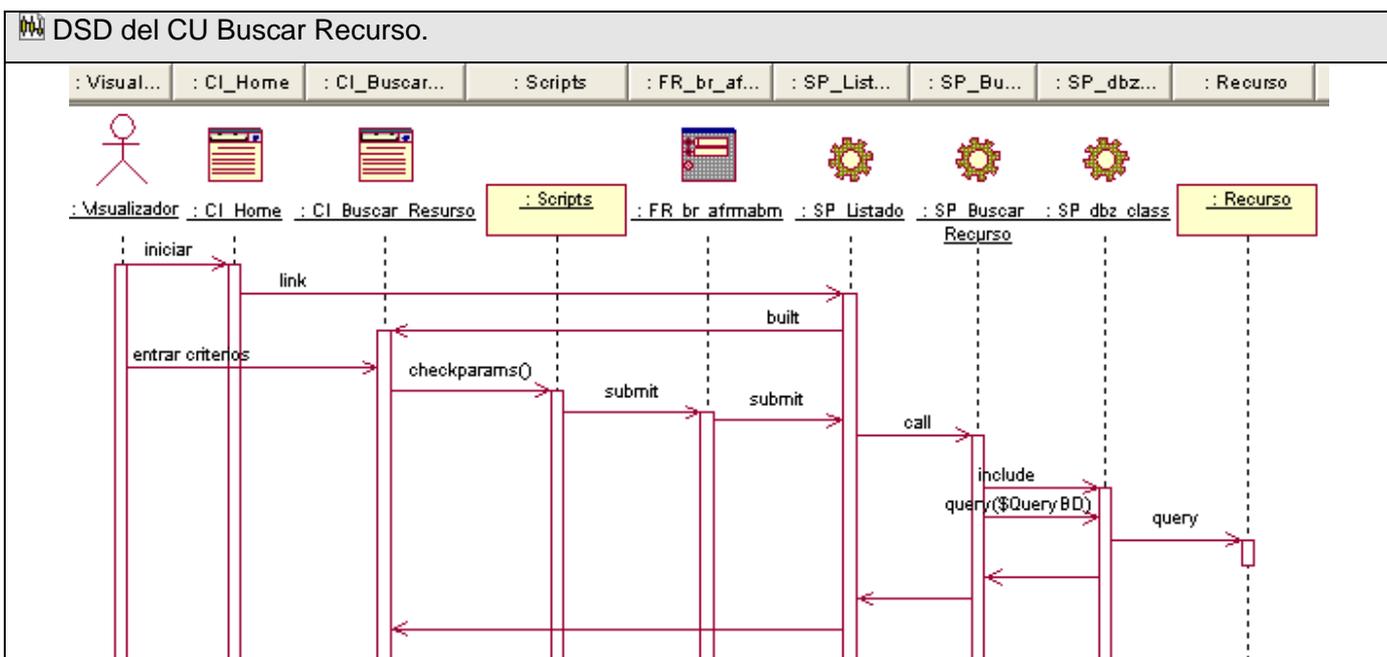


Figura 3.33 – DSD del CU Buscar Recurso.

3.3.5 Descripción de las clases y atributos.

A continuación se describen algunas de las principales clases del diseño:

<i>CI_Entrada_Sistema.</i>	
Tipo de clase:	Client Page.
Descripción:	La clase <i>CI_Entrada_Sistema</i> es una página web que se ejecuta del lado del cliente, es la página de entrada a la aplicación, permite al usuario autenticarse para acceder a la misma identificado según sus privilegios.

Tabla 3.1 – Descripción de la clase del diseño *CI_Entrada_Sistema*.

<i>CI_Agregar_Usuario.</i>	
Tipo de clase:	Client Page.
Descripción:	La clase <i>CI_Agregar_Usuario</i> es una página web que se ejecuta del lado del cliente, permite introducir los datos pertinentes para la inserción en el sistema de un nuevo usuario. Solamente podrán acceder a ella los usuarios autenticados con privilegios de Administrador.

Tabla 3.2 – Descripción de la clase del diseño *CI_Agregar_Usuario*.

<i>CI_Agregar_Estandar.</i>	
Tipo de clase:	Client Page.
Descripción:	La clase <i>CI_Agregar_Estandar</i> es una página web que se ejecuta del lado del cliente, permite introducir los datos pertinentes para la inserción en el sistema de un nuevo estándar y sus demos correspondientes. Sólo podrán acceder a ella los usuarios autenticados con privilegios de Administrador.

Tabla 3.3 – Descripción de la clase del diseño *CI_Agregar_Estandar*.

<i>CI_Agregar_Recurso.</i>	
Tipo de clase:	Client Page.
Descripción:	La clase <i>CI_Agregar_Recurso</i> es una página web que se ejecuta del lado del cliente que permite introducir los datos pertinentes para la inserción en el sistema de un

	nuevo recurso. Sólo podrán acceder a ella los usuarios autenticados con privilegios de Administrador.
--	---

Tabla 3.4 – Descripción de la clase del diseño *CI_Agregar_Recurso*.

<i>CI_Buscar_Usuario</i> .	
Tipo de clase:	Client Page.
Descripción:	La clase <i>CI_Buscar_Usuario</i> es una página web que se ejecuta del lado del cliente que posibilita realizar una búsqueda de usuarios en el sistema para su edición o eliminación, así como también mostrar un listado con todos los usuarios existentes en la aplicación. Sólo podrán acceder a ella los usuarios autenticados con privilegios de Administrador.

Tabla 3.5 – Descripción de la clase del diseño *CI_Buscar_Usuario*.

<i>CI_Buscar_Estandar</i> .	
Tipo de clase:	Client Page.
Descripción:	La clase <i>CI_Buscar_Estandar</i> es una página web que se ejecuta del lado del cliente que posibilita realizar una búsqueda de estándares en el sistema para su visualización, edición o eliminación, así como también mostrar un listado con todos los estándares existentes en la aplicación. Sólo podrán acceder a las opciones de editar y eliminar los usuarios autenticados con privilegios de Administrador.

Tabla 3.6 – Descripción de la clase del diseño *CI_Buscar_Estandar*.

<i>CI_Buscar_Demo</i> .	
Tipo de clase:	Client Page.
Descripción:	La clases <i>CI_Buscar_Demo</i> es una página web que se ejecuta del lado del cliente que posibilita realizar una búsqueda de demos en el sistema para su visualización, descarga, edición o eliminación, así como también mostrar un listado con todos los demos existentes en la aplicación. Sólo podrán acceder a las opciones de editar y eliminar los usuarios autenticados con privilegios de Administrador.

Tabla 3.7 – Descripción de la clase del diseño *CI_Buscar_Demo*.

<i>CI_Buscar_Recurso.</i>	
Tipo de clase:	Client Page.
Descripción:	La clase <i>CI_Buscar_Recurso</i> es una página web que se ejecuta del lado del cliente que posibilita realizar una búsqueda de recursos en el sistema para su descarga, edición o eliminación, así como también mostrar un listado con todos los recursos existentes en la aplicación. Sólo podrán acceder a las opciones de editar y eliminar los usuarios autenticados con privilegios de Administrador.

Tabla 3.8 – Descripción de la clase del diseño *CI_Buscar_Recurso*.

<i>CI_Editar_Perfil.</i>	
Tipo de clase:	Client Page.
Descripción:	La clase <i>CI_Editar_Perfil</i> es una página web que se ejecuta del lado del cliente que posibilita modificar el perfil de usuario en dependencia de los privilegios que se tenga. Los usuarios con privilegios de administrador podrán modificar en su perfil lo que deseen, pero los usuarios con privilegios de visualizador sólo se le permitirán modificar la dirección de correo electrónico, usuario y contraseña.

Tabla 3.9 – Descripción de la clase del diseño *CI_Buscar_Recurso*.

<i>CI_Editar_Usuario.</i>	
Tipo de clase:	Client Page.
Descripción:	La clase <i>CI_Editar_Usuario</i> es una página web que se ejecuta del lado del cliente que posibilita modificar el perfil de un usuario seleccionado y eliminar el mismo. Sólo podrán acceder a esta opción los usuarios autenticados con privilegios de administrador.

Tabla 3.10 – Descripción de la clase del diseño *CI_Editar_Usuario*.

<i>CI_Editar_Estandar.</i>	
Tipo de clase:	Client Page.
Descripción:	La clase <i>CI_Editar_Estandar</i> es una página web que se ejecuta del lado del cliente que posibilita modificar un estándar seleccionado y eliminar el mismo. Sólo podrán

	acceder a esta opción los usuarios autenticados con privilegios de administrador.
--	---

Tabla 3.11 – Descripción de la clase del diseño *CI_Editar_Estandar*.

<i>CI_Editar_Demo.</i>	
Tipo de clase:	Client Page.
Descripción:	La clase <i>CI_Editar_Demo</i> es una página web que se ejecuta del lado del cliente que posibilita un demo seleccionado y eliminar el mismo. Sólo podrán acceder a esta opción los usuarios autenticados con privilegios de administrador.

Tabla 3.12 – Descripción de la clase del diseño *CI_Editar_Demo*.

<i>CI_Editar_Recurso.</i>	
Tipo de clase:	Client Page.
Descripción:	La clase <i>CI_Editar_Recurso</i> es una página web que se ejecuta del lado del cliente que posibilita modificar un recurso seleccionado y eliminar el mismo. Sólo podrán acceder a esta opción los usuarios autenticados con privilegios de administrador.

Tabla 3.13 – Descripción de la clase del diseño *CI_Editar_Recurso*.

<i>SP_login.</i>	
Tipo de clase:	Server Page.
Descripción:	La clase <i>SP_login</i> es una página web que se ejecuta del lado del servidor, construye la página cliente <i>CI_Entrada_Sistema</i> , valida y procesa los datos que recibe de la misma. Se comunica con el método del negocio <i>SP_Autenticar</i> .

Tabla 3.14 – Descripción de la clase del diseño *SP_login*.

<i>SP_Insertar.</i>	
Tipo de clase:	Server Page.
Descripción:	La clase <i>SP_Insertar</i> es una página web que se ejecuta del lado del servidor, valida y procesa los datos que recibe de la página cliente. Se comunica con el método del negocio. Solamente podrán acceder a ella los usuarios autenticados con privilegios de Administrador.

Tabla 3.15 – Descripción de la clase del diseño *SP_Insertar*.

<i>SP_Listado.</i>	
Tipo de clase:	Server Page.
Descripción:	La clase <i>SP_Listado</i> es una página web que se ejecuta del lado del servidor, su propósito es construir la página cliente de búsqueda en cada opción, valida y procesa los datos que recibe de la misma. Se comunica con el método del negocio.

Tabla 3.16 – Descripción de la clase del diseño *SP_Listado*.

<i>SP_Eliminar.</i>	
Tipo de clase:	Server Page.
Descripción:	La clase <i>SP_Eliminar</i> es una página web que se ejecuta del lado del servidor. Recibe el id del elemento seleccionado para eliminar en la página cliente y luego invoca al método del negocio.

Tabla 3.17 – Descripción de la clase del diseño *SP_Eliminar*.

<i>SP_Editar.</i>	
Tipo de clase:	Server Page.
Descripción:	La clase <i>SP_Editar</i> es una página web que se ejecuta del lado del servidor, valida y procesa los datos que recibe de la página cliente. Luego invoca al método del negocio.

Tabla 3.18 – Descripción de la clase del diseño *SP_Editar*.

<i>SP_dbz_class.</i>	
Tipo de clase:	Server Page.
Descripción:	La clase <i>SP_dbz_class</i> es una página web que se ejecuta del lado del servidor, es para la conexión con bases de datos MySQL, usa el modulo dbx de PHP para su funcionalidad. Además crea un objeto conexión que permite hacer consultas, y recuperar los resultados; insertar, eliminar y actualizar datos. Esta clase estará en la capa de Acceso a Datos. Esta clase se encuentra dentro del framework llamado PLASER.

Tabla 3.19 – Descripción de la clase del diseño *SP_dbz_class*.

3.4 Diseño de Base de Datos.

Para diseñar la base de datos del sistema, se utilizó el diagrama de clases persistentes y el modelo de datos, que están basados en la modelación de las clases del epígrafe anterior.

3.4.1 Diagrama de clases persistentes.

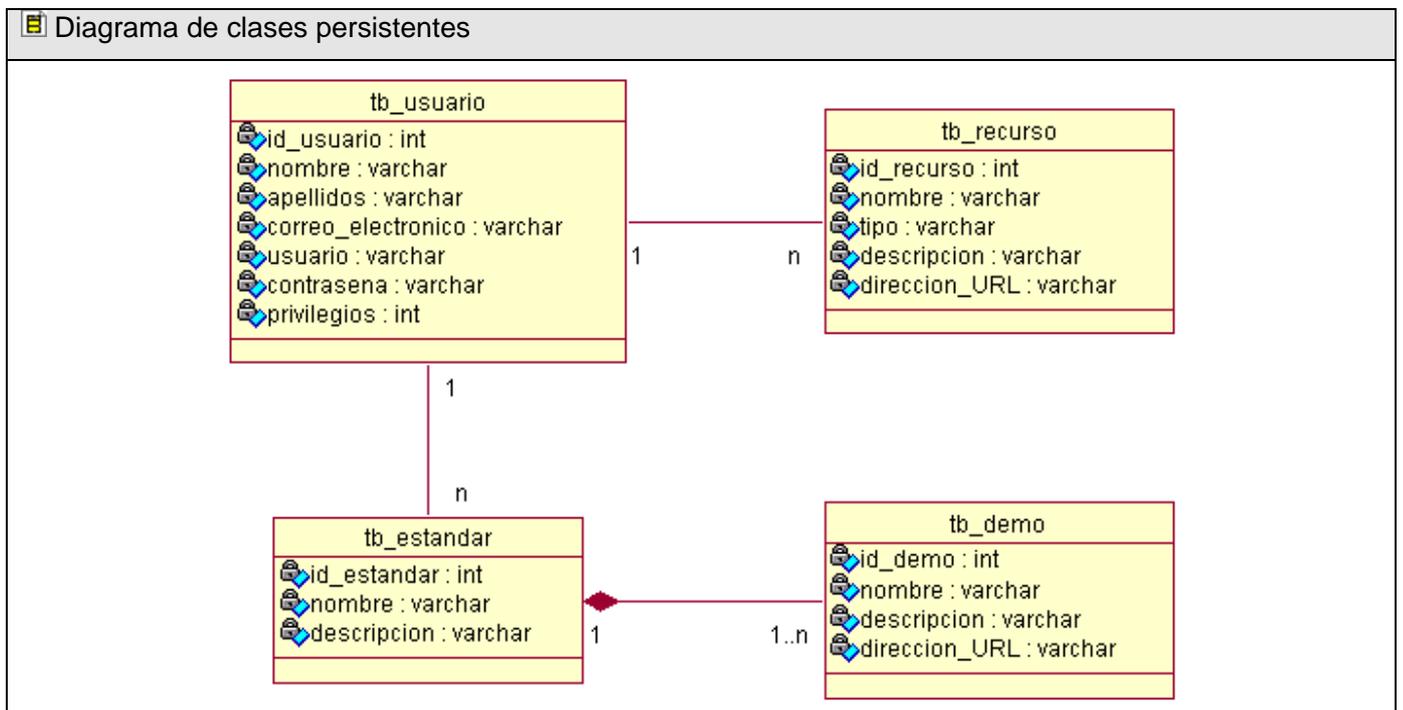


Figura 0.0 – Diagrama de clases persistentes.

3.4.1 Modelo de datos.

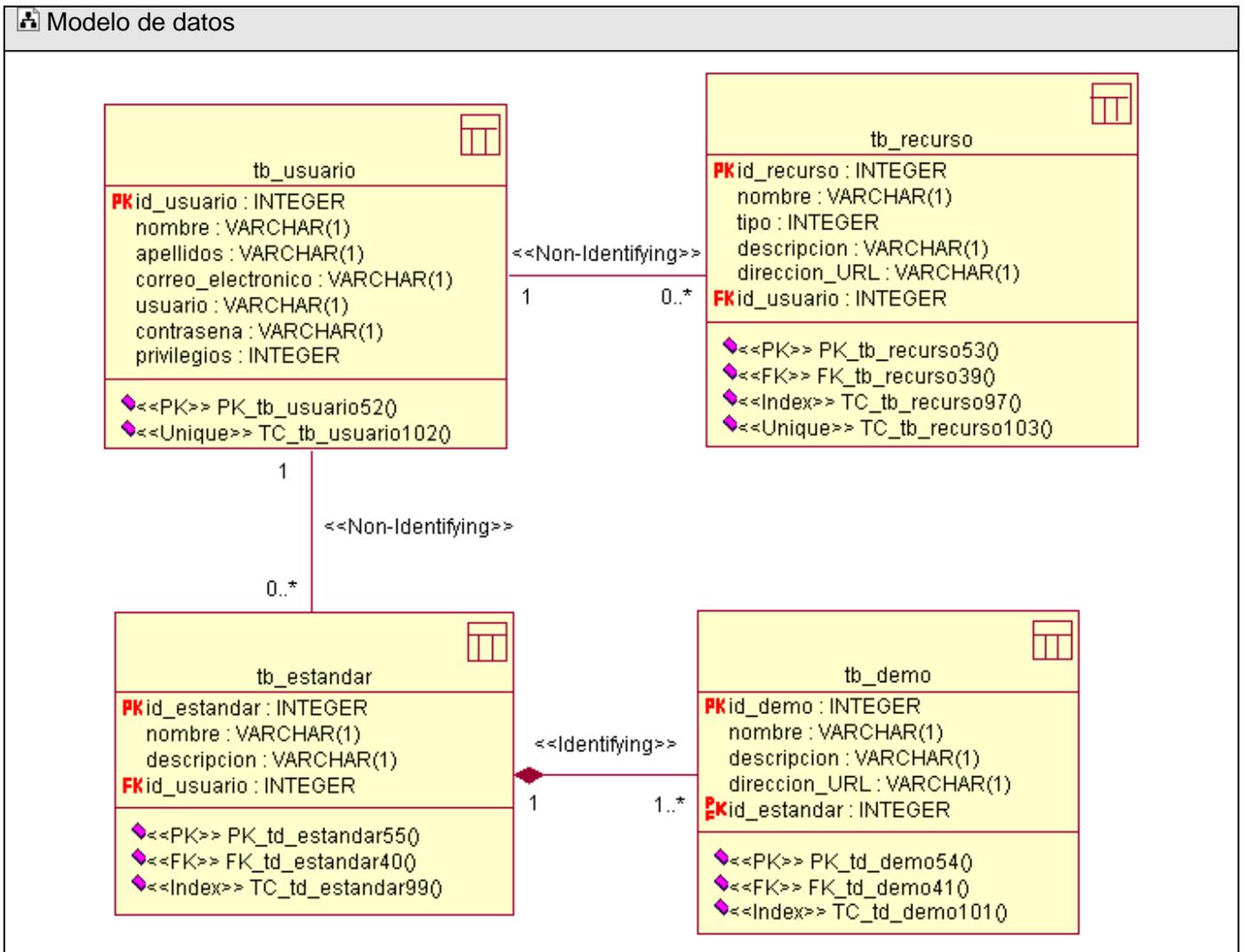


Figura 0.0 – Diagrama del Modelo de datos.

3.4.2 Descripción de tablas y atributos.

Nombre: <i>tb_usuario</i>		
Descripción: En esta tabla se almacenan todos los usuarios que podrán acceder a la aplicación con sus respectivos datos, permitirá la inserción, edición y eliminación de usuarios.		
Atributo	Tipo	Descripción
id_usuario	int(10)	Es el identificador de esta tabla, número autogenerado de forma incremental.
nombre	varchar(50)	El nombre del usuario, es un campo de entrada obligatoria.
apellidos	varchar(50)	Atributo encargado de almacenar los apellidos del usuario, es un campo de entrada obligatoria.
correo_electronico	varchar(50)	
usuario	varchar(50)	El usuario con el que se accederá en la aplicación, es un campo de entrada obligatoria.
contrasena	varchar(50)	La contraseña con la que se accederá en la aplicación, es un campo de entrada obligatoria.
privilegios	int(10)	En dependencia a estos privilegios (1 Administrador y 2 Visualizador) se proporcionarán los permisos a la información que el usuario tendrá derechos, es un campo de entrada obligatoria.

Tabla 3.20 – Descripción de la tabla *tb_usuario*.

Nombre: <i>tb_estandar</i> .		
Descripción: En esta tabla se almacenan los datos de los estándares del repositorio.		
Atributo	Tipo	Descripción
id_estandar	int(10)	Es el identificador de esta tabla, número autogenerado de forma incremental.
nombre	varchar(50)	El nombre del estándar, es un campo de entrada obligatoria y no puede estar repetido en la tabla.
descripcion	varchar(225)	Atributo que recoge una breve descripción del estándar.
id_usuario	int(10)	Llave foránea de esta tabla, permite saber que usuario inserta el estándar.

Tabla 3.21 – Descripción de la tabla *tb_estandar*.

Nombre: <i>tb_demo</i> .		
Descripción: En esta tabla se almacenan los datos de los demos del repositorio.		
Atributo	Tipo	Descripción
id_demo	int(10)	Es el identificador de esta tabla, número autogenerado de forma incremental.
nombre	varchar(50)	Nombre del estándar.
descripcion	varchar(225)	Breve descripción de en qué consiste el demo.
direccion_URL	varchar(255)	Dirección en la que se guardó el demo para permitir el posterior acceso al mismo.
id_estandar	int(10)	Llave foránea de esta tabla, permite saber a que estándar pertenece el demo.

Tabla 3.22 – Descripción de la tabla *tb_demo*.

Nombre: <i>tb_recurso</i> .		
Descripción: En esta tabla se almacenan los datos de los recursos del repositorio.		
Atributo	Tipo	Descripción
id_recurso	int(10)	Es el identificador de esta tabla, número autogenerado de forma incremental.
nombre	varchar(50)	Es el atributo donde se almacena el nombre del recurso, es útil para la búsqueda.
tipo	varchar(20)	El tipo de recurso (Herramienta, Tutorial, Hoja de Estilo (CSS) u Otro).
descripcion	varchar(255)	Atributo que recoge una breve descripción del estándar.
direccion_URL	varchar(255)	Dirección donde queda almacenado el recurso para permitir el posterior acceso al mismo.
id_usuario	int(10)	Llave foránea de esta tabla, permite saber que usuario inserta el recurso.

Tabla 3.23 – Descripción de la tabla *tb_recurso*.

3.5 Conclusiones.

En este capítulo se modelaron los artefactos relacionados con el flujo de trabajo Análisis y Diseño. Los diagramas y especificaciones de diseño que se proponen, son fácilmente entendibles, por cualquier programador; por lo que lo único que resta es implementar la aplicación que se propone desarrollar. Se definieron y presentaron conjuntamente el diagrama de clases persistentes y el modelo de datos que son de vital importancia para la construcción de la base de datos.

CAPÍTULO 4. IMPLEMENTACIÓN.

4.1 Introducción.

En el capítulo anterior se realizó el flujo de trabajo análisis y diseño, en el presente se comienza con el resultado obtenido y se implementa el sistema en términos de componentes. Se trazan las estrategias de integración con otros sistemas. Los artefactos más importantes que serán generados en el modelo de implementación son el Diagrama de Componentes y el de Despliegue. Serán descritos además algunos de los métodos del negocio más complejos, terminando el capítulo con la definición de los estándares de diseño de interfaz, codificación y tratamiento de excepciones que serán utilizados.

4.2 Justificación de integración con otros sistemas.

El repositorio de estándares de diseño de interfaz forma parte del portal de arquitectura de la empresa SOFTEL, este último tiene como objetivo ser el sitio donde se pueden encontrar los componentes reutilizables tanto de la capa de presentación como la de negocio. La aplicación se centra en los elementos reutilizables de capa de presentación, y no sólo posibilitará la reducción del tiempo de desarrollo de los nuevos módulos del SISalud, si no que también logrará que estos posean una armonía visual y funcional.

El sistema completo es el resultado de la integración al repositorio de los componentes de negocio que está siendo desarrollado actualmente por SOFTEL, de forma tal que queden cubiertas las dos capas principales en las que se debe centrar la reutilización.

4.3 Modelo de Implementación.

4.3.1 Diagrama de Componentes.

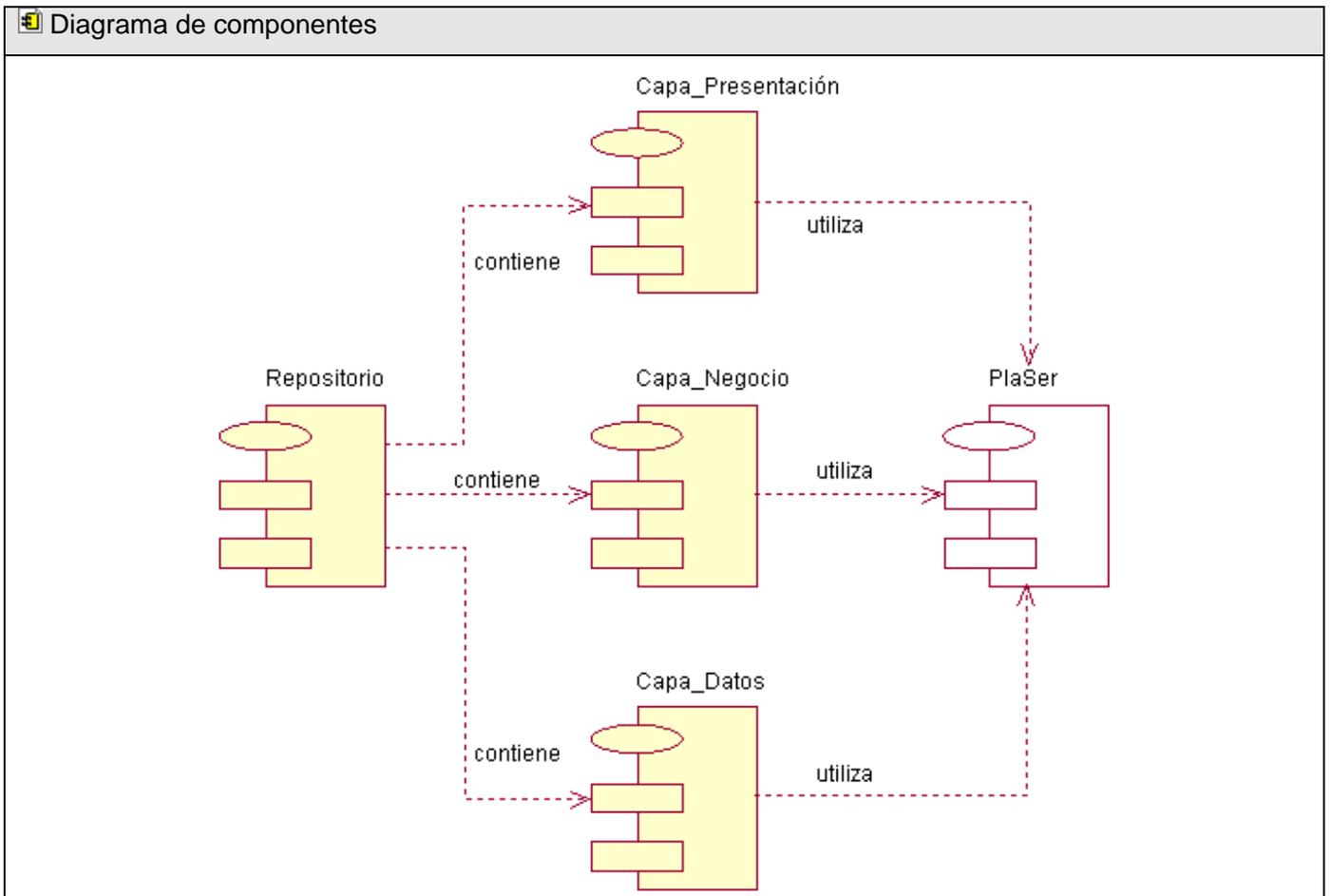


Figura 0.0 – Diagrama de Componentes.

4.3.2 Diagrama de Despliegue.

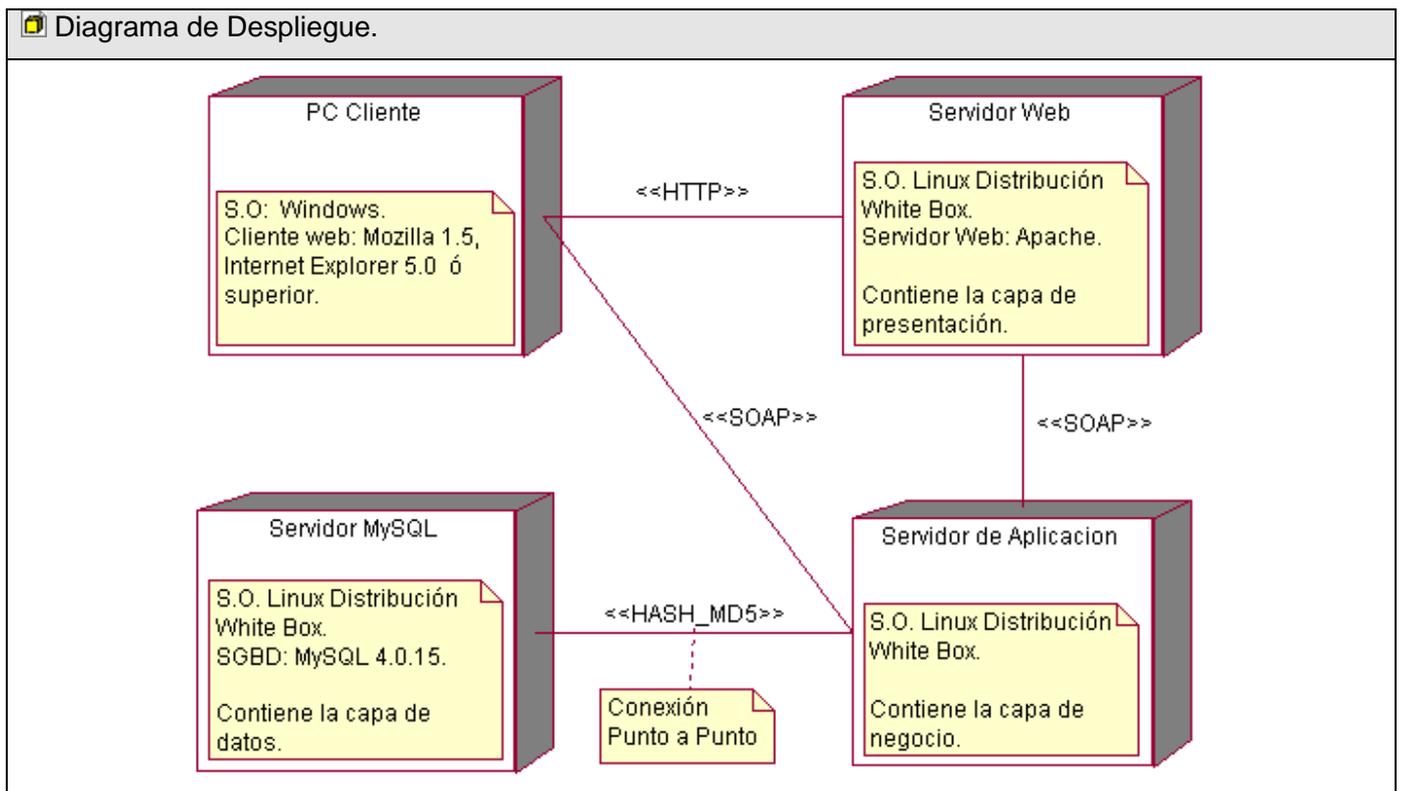


Figura 0.0 – Diagrama de Despliegue.

4.4 Descripción de los métodos (agentes o métodos más complejos).

Buscar Estándar.

Este método es el encargado de buscar todos los estándares que existan en el Repositorio, tiene dos aristas, la primera cuando se obtiene un listado de todos los estándares, para la cual se consulta la base de datos devolviendo todos los campos de la tabla perteneciente a esta opción; la segunda cuando recibe algún parámetro de entrada (nombre, descripción o ambos) se hace una consulta similar a la anterior pero con las restricciones pasadas como parámetros de entrada, es decir que buscará en la tabla estándar todos aquellos elementos que cumplan con las restricciones.

En ambos casos los resultados son devueltos en un listado ordenado ascendente por nombre. El estándar mostrará la cantidad de demos asociados a el, así como también la descripción del mismo.

Buscar Demo.

Este método es el encargado de buscar todos los demos que existan en el Repositorio, tiene dos aristas, la primera cuando se obtiene un listado de todos los demos, para la cual se consulta la base de datos devolviendo todos los campos de la tabla perteneciente a esta opción; la segunda cuando recibe algún parámetro de entrada (nombre, descripción o ambos) se hace una consulta similar a la anterior pero con las restricciones pasadas como parámetros de entrada, es decir que buscará en la tabla demo todos aquellos elementos que cumplan con las restricciones.

En ambos casos los resultados son devueltos en un listado ordenado ascendente por nombre. El demo mostrará el estándar al que pertenece, así como también su descripción.

Agregar Estándar.

Este método es el encargado de insertar un nuevo estándar a la aplicación. Recibe como parámetros: nombre, descripción y un arreglo con los datos de demos asociados al mismo, estos últimos poseen nombre, descripción y serán importados a la aplicación en ficheros compactados.

4.5 Estándares de diseño de interfaz.

Aunque esta aplicación está dirigida a los desarrolladores y diseñadores que tienen un conocimiento medio y en muchos casos avanzado, de lo que es el trabajo en un PC y en aplicaciones web de este tipo, ya que son parte del equipo que las desarrolla, teniendo en cuenta esto los esfuerzos están orientados a lograr una interfaz clara y de fácil uso.

Para lograr lo anteriormente planteado el diseño de la aplicación se basa en los 7 Principios del Diseño Universal o Diseño para Todos:

Principio uso equiparable.

El diseño es útil y vendible a personas con diversas capacidades.

- Que proporcione las mismas maneras de uso para todos los usuarios: idénticas cuando es posible, equivalentes cuando no lo es.
- Que evite segregar o estigmatizar a cualquier usuario.
- Las características de privacidad, garantía y seguridad deben estar igualmente disponibles para todos los usuarios.
- Que el diseño sea atractivo para todos los usuarios.

Principio uso flexible.

El diseño se acomoda a un amplio rango de preferencias y habilidades individuales.

- Que ofrezca posibilidades de elección en los métodos de uso.
- Que pueda accederse y usarse tanto con la mano derecha como con la izquierda.
- Que facilite al usuario la exactitud y precisión.
- Que se adapte al paso o ritmo del usuario.

Principio simple e intuitivo.

El uso del diseño es fácil de entender, atendiendo a la experiencia, conocimientos, habilidades lingüísticas o grado de concentración actual del usuario.

- Que elimine la complejidad innecesaria.
- Que sea consistente con las expectativas e intuición del usuario.
- Que se acomode a un amplio rango de alfabetización y habilidades lingüísticas.
- Que dispense la información de manera consistente con su importancia.
- Que proporcione avisos eficaces y métodos de respuesta durante y tras la finalización de la tarea.

Principio información perceptible.

El diseño comunica de manera eficaz la información necesaria para el usuario, atendiendo a las condiciones ambientales o a las capacidades sensoriales del usuario.

- Que use diferentes modos para presentar de manera redundante la información esencial (gráfica, verbal o táctilmente).
- Que proporcione contraste suficiente entre la información esencial y sus alrededores.
- Que amplíe la legibilidad de la información esencial.
- Que diferencie los elementos en formas que puedan ser descritas (por ejemplo, que haga fácil dar instrucciones o direcciones).
- Que proporcione compatibilidad con varias técnicas o dispositivos usados por personas con limitaciones sensoriales.

Principio con tolerancia al error.

El diseño minimiza los riesgos y las consecuencias adversas de acciones involuntarias o accidentales.

- Que disponga los elementos para minimizar los riesgos y errores: elementos más usados, más accesibles; y los elementos peligrosos eliminados, aislados o tapados.
- Que proporcione advertencias sobre peligros y errores.
- Que proporcione características seguras de interrupción.
- Que desaliente acciones inconscientes en tareas que requieren vigilancia.

Principio que exija poco esfuerzo físico.

El diseño puede ser usado eficaz y confortablemente y con un mínimo de fatiga.

- Que permita que el usuario mantenga una posición corporal neutra.
- Que utilice de manera razonable las fuerzas necesarias para operar.
- Que minimice las acciones repetitivas.
- Que minimice el esfuerzo físico continuado.

Principio tamaño y espacio para el acceso y uso.

Que proporcione un tamaño y espacio apropiados para el acceso, alcance, manipulación y uso, atendiendo al tamaño del cuerpo, la postura o la movilidad del usuario.

- Que proporcione una línea de visión clara hacia los elementos importantes tanto para un usuario sentado como de pie.
- Que el alcance de cualquier componente sea confortable para cualquier usuario sentado o de pie.
- Que se acomode a variaciones de tamaño de la mano o del agarre.
- Que proporcione el espacio necesario para el uso de ayudas técnicas o de asistencia personal.

4.6 Estándares de Codificación.

La definición de estándares de codificación se realiza en el marco del proceso de desarrollo del proyecto. Con el propósito de distribuir los esfuerzos y mejorar los rendimientos de la aplicación, se utilizó la programación orientada a servicios, creando clases genéricas que permiten la definición y distribución de las llamadas a los procedimientos de los módulos distribuidos, la definición de los métodos propios de cada módulo y el manejo de las bases de datos.

Actualmente se encuentran estándares de codificación para la mayoría de los lenguajes existentes. El uso de los mismos, partiendo de las convenciones definidas, permite una mejor comunicación entre los programadores, creando las condiciones para la reusabilidad y el mantenimiento de los sistemas. Para definir el estilo de codificación a seguir en la aplicación se utilizó la notación estándar establecida para aplicaciones desarrolladas en PHP (PHP Coding Standard), que mayormente está basada en el estándar de código para aplicaciones en C++ (C++ Coding Standard).

Las etiquetas de apertura y cierre del lenguaje serán de la forma `<?php ?>`, ya que siempre están disponibles en cualquier configuración.

Se hará uso de los arreglos predefinidos para el manejo de los valores enviados por el usuario `$_GET`, `$_POST`, `$_FILES` evitando el uso de `$_REQUEST`.

Para nombrar las variables se seguirá la regla de escribir los identificadores con letras minúsculas y en lenguaje español, utilizando como separador para las palabras el carácter “_”, tratando de usar nombres sugerentes a la acción de la variable.

Todos los campos identificadores van a comenzar con el identificador (id) seguido del nombre del campo. Ejemplo: id _ enfermedad.

Los arreglos empezarán con el identificador array y las palabras no se separarán con el carácter “_”. Ejemplo: Arrayidtipoenfermedad.

Las estructuras se identificarán poniendo al final del nombre struct. Ejemplo: paginadostruct.

En el caso de las clases se pondrá delante la letra C. Ejemplo: CFachada y en el de los métodos no se usarán abreviaturas y las palabras continuas deben comenzar con mayúsculas. Ejemplo: ListarTotalPersonalSalud.

Para comentar el código se utilizará, en el caso de una línea, al final de la misma el carácter “//” y seguido el comentario y en el caso de un bloque se utilizará los caracteres “/* */”.

Se usará una indentación en el código de cuatro espacios para facilitar la lectura de éste. Las llaves se usarán poniendo la llave inicial en una línea para ella sólo, y en su respectiva columna la llave final también en una línea.

El idioma de las clases auxiliares como sesión y error, será el inglés para garantizar la homogeneidad con las programadas en este ámbito en el mundo, en el caso de los Servicios Web y la interfase de administración se usará el español para esclarecer los objetivos de cada método o script a utilizar.

Para lograr que las comparaciones sean seguras, se colocarán siempre los valores constantes a la izquierda de la comparación "if (6 == \$variable)", con esto garantizará la generación de un error cuando por error escriba '=' y no '=='. Se utilizará el operador “?” para sentencias cortas, preferiblemente que ocupen una sola línea. La sentencia switch siempre tendrá la opción default y se evitará el uso de continue y break, ya que podrían perder la vista lógica del código fuente.

El almacenamiento de la información será en scripts SQL para construir la base de datos e interactuar con ella desde las aplicaciones.

Las palabras correspondientes a las sentencias SQL y sus parámetros deben ir en mayúsculas.

En las consultas de inserción se debe colocar siempre el nombre de los campos en los cuales desea escribir en la tabla.

Los nombres de las tablas deben ir en minúsculas y cada palabra separada por línea abajo "_". (Ejemplo: Id_nombre_tabla)

En el caso de los XSL será con el mismo nombre que el fichero de la capa de presentación.

Los controles seguirán el siguiente tratamiento:

Control	Prefijo	Ejemplo
Botón	Btn	btnAceptar
Etiqueta	Lbl	lblNombre
Lista/Menú	Mn	mnPrincipal
Campo de Texto	Txt	TxtFecha
Botón de Opción	Opt	OptSexo
Casilla de Verificación	Chx	ChxBorrar

Tabla 4.1 Estándares para los controles.

Las páginas HTML se harán sin incluir código y todas las funciones JavaScript que se usarán se escribirán dentro de ficheros ".js".

Para la capa de datos tienen que nombrar la base de datos poniendo el identificador del proyecto "APS" seguido del carácter "_" y del nombre del módulo. Ejemplo: bd_RPOB.

Las tablas se identificarán con el acrónimo tb_Nombre, ejemplo: tb_Usuarios.

Los campos de la base de datos se nombrarán igual que las variables.

4.7 Tratamiento de excepciones.

Para depurar los errores se hará uso del lenguaje JavaScript. Por medio de este serán informados la mayoría de los errores de la página, como apoyo a las validaciones de entrada de datos, garantizando que los datos introducidos por los usuarios sean válidos, o sea posible corregirlos en caso contrario.

Otros errores en la capa de negocio serán tratados devolviendo un SOAP_FAULT, cuyos elementos FaultCode, FaultString, FaultActor describiremos a continuación:

FaultCode:

Código de texto utilizado para indicar la clase de error, codificado de la siguiente manera.

Código del proyecto-código del modulo (:) número del método (.) número del error. Ejemplo: APS-RF: 1.5 que indica error 5 en el método 1 del módulo Registro de Fallecidos perteneciente al Proyecto APS.

FaultString:

Una explicación del error asequible al humano. Debe tenerse en cuenta que este texto puede ser mostrado al operador final del sistema. Ejemplo: Formato de entrada no válido para la fecha de cierre estadístico.

FaultActor:

Un texto que indica quien provocó el error, siempre será el nombre del método que eleva la excepción.

4.8 Conclusiones.

Luego de realizar varias iteraciones, se ha logrado modelar todo el sistema que ha sido objeto de estudio durante el transcurso del presente trabajo investigativo. Se trazaron las estrategias de integración con otros sistemas. Los diagramas y especificaciones que se propusieron en el este capítulo son de fácil entendimiento, tal es el caso de el Diagrama de Componentes y el de Despliegue. Se describieron algunos de los métodos del negocio más complejos y se definieron de los estándares de diseño de interfaz, codificación y tratamiento de excepciones.

CONCLUSIONES.

Luego del estudio realizado y del correspondiente análisis y diseño del sistema:

1. Se logró la implementación de una aplicación web que sirve de repositorio para almacenar la información relacionada con los estándares de diseño de interfaz de los componentes del SISalud.
2. Se estudió el entorno de presentación implementado en el proyecto APS para los componentes del SISalud, además de las tendencias y tecnologías para lograr la implementación.
3. Se utilizó el framework PLASER de la empresa SOFTEL para la disminución de los costes de implementación.
4. Se definieron estándares y demos aplicables a la capa de presentación del SISalud.

Por todo lo anterior se concluye que los objetivos propuestos para el presente proyecto han sido cumplidos satisfactoriamente.

RECOMENDACIONES.

Se recomienda integrar el repositorio de estándares al portal de arquitectura de la empresa Softel, para tener toda la información reutilizable centralizada.

Se debe apoyar la reutilización de soluciones por parte de los líderes del proyecto APS, con vista a impedir que la información caduque por desuso.

Mantener actualizada la aplicación durante todo el proceso de desarrollo con ejemplos útiles y prácticos.

REFERENCIAS BIBLIOGRÁFICAS.

1. VILLA, L. Guías de estilo: diseño, normalización y usabilidad. *[Disponible en: <http://www.desarrolloweb.com/articulos/1513.php>*
2. SKONNARD, A. SOAP: The Simple Object Access Protocol, Enero 2000. *[Disponible en: <http://www.microsoft.com/mind/0100/soap/soap.asp>*
3. GONZÁLEZ, B. SOAP (Simple Object Access Protocol). *[Disponible en: <http://www.desarrolloweb.com/articulos/1557.php>*
4. CHRISTENSEN, E. Lenguaje de descripción de servicios Web (WSDL) 1.0, Enero de 2003. *[Disponible en: <http://www.microsoft.com/spanish/msdn/articulos/archivo/090201/voices/wsdl.asp>*
5. CORALES, Y. S. Registro de Partos y Nacimientos para el Sistema Integral de Salud. Ciudad de La Habana, Instituto Superior Politécnico “JOSE ANTONIO ECHEVERRIA”, 2005. 143. p.
6. RODRÍGUEZ, A. Diseño de Aplicaciones Three Tier, 1997. *[Disponible en: <http://www.fpress.com/revista/Num9711/Nov97.htm>*
7. POZO, J. R. Recomendación del W3C "HTML 4.01 Specification", Junio 2001. *[Disponible en: <http://html.conclase.net/w3c/html401-es/present/styles.html>*
8. LARA, J. L. XHTML 1.0: El Lenguaje de Etiquetado Hipertextual Extensible, enero de 2000. *[Disponible en: <http://www.w3.org/TR/2000/REC-xhtml1-20000126>*
9. Ídem a la referencia 5.
10. Manual de referencia de MySQL. *[Disponible en: <http://www.mysql.com/why-mysql/>*
11. JACOBSON, I.; BOOCH, G., et al. El Proceso Unificado de Desarrollo de Software. Ciudad de La Habana: Félix Varela, 2004. vol. 1, 438 p.
12. CONALLEN, J. Modeling web application architectures with UML. 2000. *Communications of the ACM (volume 42, number 10)*
13. Ídem a la referencia 5.
14. Ídem a la referencia 5.

BIBLIOGRAFÍA.

1. CHRISTENSEN, E. Lenguaje de descripción de servicios Web (WSDL) 1.0, Enero de 2003. [Disponible en: <http://www.microsoft.com/spanish/msdn/articulos/archivo/090201/voices/wsdl.asp>
2. CONALLEN, J. Modeling web application architectures with UML. 2000. *Communications of the ACM (volume 42, number 10)*
3. CORALES, Y. S. Registro de Partos y Nacimientos para el Sistema Integral de Salud. Ciudad de La Habana, Instituto Superior Politécnico "JOSE ANTONIO ECHEVERRIA", 2005. 143. p.
4. GONZÁLEZ, B. SOAP (Simple Object Access Protocol). [Disponible en: <http://www.desarrolloweb.com/articulos/1557.php>
5. JACOBSON, I.; BOOCH, G., et al. *El Proceso Unificado de Desarrollo de Software*. Ciudad de La Habana: Félix Varela, 2004. vol. 1, 438 p.
6. LARA, J. L. XHTML 1.0: El Lenguaje de Etiquetado Hipertextual Extensible, enero de 2000. [Disponible en: <http://www.w3.org/TR/2000/REC-xhtml1-20000126>
7. RODRÍGUEZ, A. Diseño de Aplicaciones Three Tier, 1997. [Disponible en: <http://www.fpress.com/revista/Num9711/Nov97.htm>
8. Manual de referencia de MySQL. [Disponible en: <http://www.mysql.com/why-mysql/>
9. CABRERA, M. *Propuesta de Esquema para el Sistema Ingral de Salud, 2006. La Habana, 2006.*
10. POZO, J. R. Recomendación del W3C "HTML 4.01 Specification", Junio 2001. [Disponible en: <http://html.conclase.net/w3c/html401-es/present/styles.html>
11. PRESSMAN, R. *Software Engineering. A Practitioner's Approach. Fourth Edition. McGraw – Hill. USA, 1999.*
12. SKONNARD, A. SOAP: The Simple Object Access Protocol, Enero 2000. [Disponible en: <http://www.microsoft.com/mind/0100/soap/soap.asp>
13. SOFTEL. Documento sobre la Arquitectura de Software para los componentes a emplear por el Sistema de Información para la Salud. La Habana, 2006.

14. VILLA, L. Guías de estilo: diseño, normalización y usabilidad. *[Disponible en:*
<http://www.desarrolloweb.com/articulos/1513.php>

ANEXOS.

Anexo1. Algunos ejemplos de los Estándares con sus respectivos Demos.

Estándar Agregar:

Agregar - AAA

Nombre Campo	<input type="text"/>	*
Nombre Campo	<input type="text"/>	*
Nombre Campo	<input type="text"/>	*
Nombre Campo	<input type="text"/>	*
Nombre Campo	« Seleccione » 	*

(*) Los campos señalados son de entrada obligatoria.

Estándar Buscar:

Buscar - AAA

 **Buscar por:**

Nombre Campo	<input type="text"/>	Nombre Campo	<input type="text"/>
Nombre Campo	« Seleccione » 	Nombre Campo	« Seleccione » 

Escoja los criterios de búsqueda y presione "Buscar".

Estándar Búsqueda Avanzada:

Buscar - AAA

 **Buscar por:**

Nombre Campo <input type="text"/>	Nombre Campo <input type="text"/>
Nombre Campo « Seleccione » ▾	Nombre Campo « Seleccione » ▾
<input type="button" value="Buscar"/> <input type="button" value="Búsqueda Avanzada ▾"/>	
Nombre Campo <input type="text"/>	Nombre Campo <input type="text"/>
Nombre Campo « Seleccione » ▾	Nombre Campo « Seleccione » ▾
<input type="button" value="Buscar"/> <input type="button" value="Búsqueda Avanzada ▲"/>	

Escoja los criterios de búsqueda y presione "Buscar".

Estándar Campo Calendario:

Calendario 

file:// - Calendario - Mo... 

Seleccione la Fecha 

Mes ▾ Año ▾

D	L	M	M	J	V	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

Done

Estándar Contenedor de Opciones:

Nombre de Opción	
Opciones no Asignadas	Opciones Asignadas
Opción 1	
Opción 2	
Opción 3	
Opción 4	
Opción 5	
	»
	«
	»»
	««

Estándar Editar:

🔍 Editar - AAA

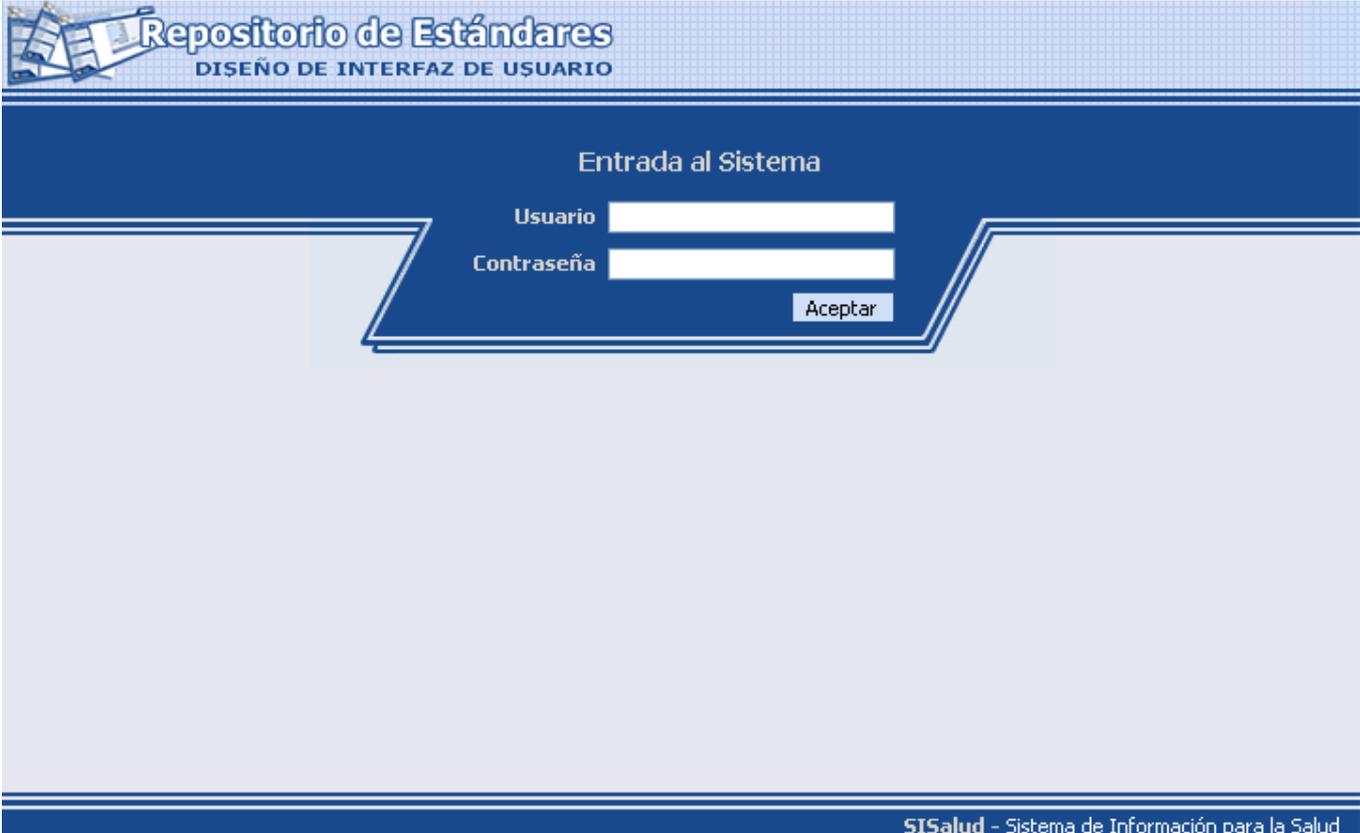
	Agregar	Eliminar
Nombre Campo	<input type="text"/>	*
Nombre Campo	<input type="text"/>	*
Nombre Campo	<input type="text"/>	*
Nombre Campo	<input type="text"/>	*
Nombre Campo	<input type="text"/>	*
(*) Los campos señalados son de entrada obligatoria.		
	Aceptar	Cancelar

Estándar MultiTab:

🔍 Aaa - AAA

Tab 1	Tab 2	Tab 3
Tab 1	<input type="text"/>	*
Nombre Campo	<input type="text"/>	*
(*) Los campos señalados son de entrada obligatoria.		
	Aceptar	Cancelar

Anexo2.



The image shows a user interface for a system login. At the top left, there is a logo with the text "Repositorio de Estándares" and "DISEÑO DE INTERFAZ DE USUARIO". The main content area is a dark blue box with the title "Entrada al Sistema". Inside this box, there are two input fields: "Usuario" and "Contraseña". Below the "Contraseña" field is a button labeled "Aceptar". The background of the page is light blue with a subtle grid pattern. At the bottom right, there is a footer that reads "SISalud - Sistema de Información para la Salud".

Repositorio de Estándares
DISEÑO DE INTERFAZ DE USUARIO

Entrada al Sistema

Usuario

Contraseña

Aceptar

SISalud - Sistema de Información para la Salud

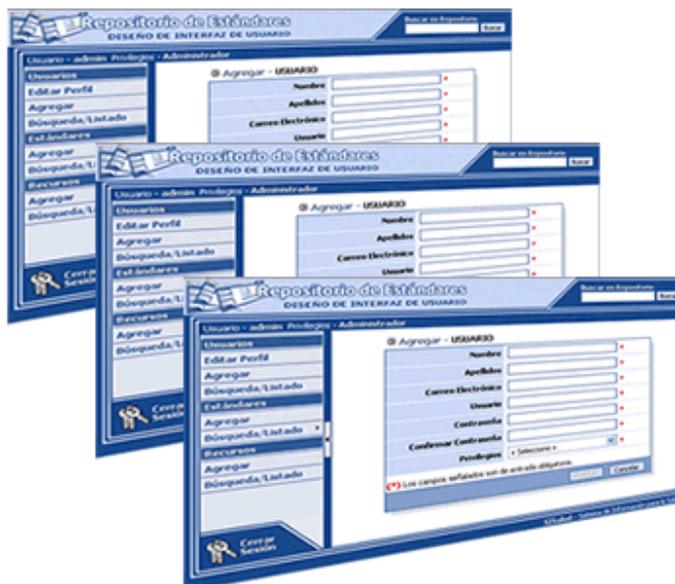
- Usuarios
 - Editar Perfil
 - Agregar
 - Búsqueda/Listado
 - Estándares
 - Agregar
 - Búsqueda/Listado >
 - Recursos
 - Agregar
 - Búsqueda/Listado
-  Cerrar Sesión

> Bienvenido **Leonardo Miguel González Arnosó** al Repositorio de Estándares de Diseño de Interfaz.

Usted tiene Privilegios de Administrador .

Por lo que podrá gestionar la información relacionada con los Usuarios, Estándares, Demos y Recursos que aquí se muestra.

Esta aplicación surge con el objetivo de estandarizar la interfaz de usuario del SISalud - Sistema de Información para la Salud.



Anexo3.

The screenshot displays the 'Repositorio de Estándares' application interface. At the top, the title 'Repositorio de Estándares' and subtitle 'DISEÑO DE INTERFAZ DE USUARIO' are visible. On the right, the user is logged in as 'admin' with 'Administrador' privileges. The left sidebar contains a menu with options: 'Usuarios', 'Editar Perfil', 'Agregar', 'Búsqueda/Listado', 'Estándares', 'Agregar', 'Búsqueda/Listado', 'Recursos', 'Agregar', and 'Búsqueda/Listado'. A mouse cursor is hovering over the 'Agregar' option, which has a tooltip that says 'Agregar Usuario'. The main content area shows a form titled 'Agregar - USUARIO' with the following fields: 'Nombre', 'Apellidos', 'Correo Electrónico', 'Usuario', 'Contraseña', 'Confirmar Contraseña', and 'Privilegios'. Each field has an asterisk indicating it is required. The 'Privilegios' field is a dropdown menu currently set to '« Seleccione »'. Below the form, a note states: '(*) Los campos señalados son de entrada obligatoria.' At the bottom right of the form are 'Aceptar' and 'Cancelar' buttons. In the bottom left corner of the sidebar, there is a 'Cerrar Sesión' button with a key icon. The footer of the page reads 'SISalud - Sistema de Información para la Salud'.

- Usuarios
- Editar Perfil
- Agregar
- Búsqueda/Listado
- Estándares
- Agregar
- Búsqueda/Listado >
- Recursos
- Agregar
- Búsqueda/Listado

Búsqueda/Listado de Usuarios

Buscar - USUARIO

Buscar por:

Nombre Apellidos

Usuario Privilegios « Seleccione »

Buscar Mostrar Todos

Resultados de búsqueda.

Nombre ^	Apellidos	Usuario	Privilegios
<input type="radio"/> Abel	González Rodriguez	abel	Administrador
<input type="radio"/> Ariel	Gómez	haile	Administrador
<input type="radio"/> Carlos	Acosta Montejo	carlos	Visualizador
<input type="radio"/> Leonardo	González González	leogg	Visualizador
<input type="radio"/> Leonardo Miguel	González Arnosó	admin	Administrador
<input type="radio"/> Leonardo Miguel	González Arnosó	visual	Visualizador
<input type="radio"/> Luis	Laguna Oliva	luis	Visualizador
<input type="radio"/> Silvia	Arnosó Gómez	silvia	Administrador
<input type="radio"/> Victor	Vazquez	victor	Administrador
<input type="radio"/> Yendry	Calaña Toledo	yendry	Visualizador

1- 10/ 13

Editar Eliminar Página # 1 Próximo Último

 Cerrar Sesión



Repositorio de Estándares

DISEÑO DE INTERFAZ DE USUARIO

Usuario admin
Privilegios Administrador

- Usuarios
- Editar Perfil
- Agregar
- Búsqueda/Listado
- Estándares
- Agregar
- Búsqueda/Listado >
- Recursos
- Agregar
- Búsqueda/Listado


Cerrar Sesión

✖ Editar -USUARIO

Agregar
Eliminar

Nombre	<input type="text" value="Leonardo Miguel"/>	*
Apellidos	<input type="text" value="González Arnosó"/>	*
Correo Electrónico	<input type="text" value="lgonzaleza@uci.cu"/>	*
Usuario	<input type="text" value="admin"/>	*
Contraseña	<input type="password" value="....."/>	*
Confirmar Contraseña	<input type="password" value="....."/>	*
Privilegios	<input type="text" value="Administrador"/> ▼	*

(*) Los campos señalados son de entrada obligatoria.

Aceptar
Cancelar

SISalud - Sistema de Información para la Salud

Anexo 4.

The screenshot displays the 'Repositorio de Estándares' web application. The header includes the title 'Repositorio de Estándares' and 'DISEÑO DE INTERFAZ DE USUARIO'. On the right, the user is identified as 'admin' with 'Privilegios' of 'Administrador'. The left sidebar contains navigation menus for 'Usuarios', 'Estándares', and 'Recursos', each with 'Agregar' and 'Búsqueda/Listado' options. The 'Usuarios' menu is active, and the 'Editar Perfil' option is selected, with a tooltip showing 'Editar Perfil'. The main content area is titled 'Editar Perfil - USUARIO' and contains a form with the following fields:

Nombre	Leonardo Miguel	*
Apellidos	González Arnosó	*
Correo Electrónico	lgonzaleza@uci.cu	*
Usuario	admin	*
Contraseña	●●●●●●●●	*
Confirmar Contraseña	●●●●●●●●	*

(*) Los campos señalados no pueden quedar vacíos.

Buttons: Aceptar, Cancelar

Footer: SISalud - Sistema de Información para la Salud

Anexo 5.

Repositorio de Estándares
DISEÑO DE INTERFAZ DE USUARIO

Usuario: admin
Privilegios: Administrador

Usuarios
Editar Perfil
Agregar
Búsqueda/Listado
Estándares
Agregar
Búsqueda/Listado
Recursos
Agregar
Búsqueda/Listado

Agregar - ESTÁNDAR

Nombre: *

Descripción:

Listado - DEMOS DEL ESTÁNDAR

Agregar Eliminar

	Nombre	Descripción	URL	
<input type="radio"/>	<input type="text"/> *	<input type="text"/>	<input type="text"/>	Examinar... *
<input type="radio"/>	<input type="text"/> *	<input type="text"/>	<input type="text"/>	Examinar... *
<input type="radio"/>	<input type="text"/> *	<input type="text"/>	<input type="text"/>	Examinar... *

1 - 3

(*) Los campos señalados son de entrada obligatoria.

Aceptar Cancelar

Cerrar Sesión

SISalud - Sistema de Información para la Salud

Repositorio de Estándares
DISEÑO DE INTERFAZ DE USUARIO

Usuario **admin**
Privilegios **Administrador**

Usuarios
Editar Perfil
Agregar
Búsqueda/Listado

Estándares
Agregar
Búsqueda/Listado >

Recursos
Agregar
Búsqueda/Listado

 Cerrar Sesión

✖ Editar - ESTÁNDAR

Agregar Eliminar

Nombre	ListadoTaxonomía *
Descripción	El listado por Taxonomía es un listado usado para separar los elementos por categorías. 

(*) Los campos señalados no pueden quedar vacíos.

Aceptar Cancelar

SISalud - Sistema de Información para la Salud



- Usuarios
- Editar Perfil
- Agregar
- Búsqueda/Listado
- Estándares
- Agregar
- Búsqueda/Listado >
- Recursos
- Agregar
- Búsqueda/Listado



Cerrar Sesión

✖ Editar - DEMO

<input type="button" value="Agregar"/> <input type="button" value="Eliminar"/>	
Nombre	<input type="text" value="Leyenda"/> *
Descripción	<input type="text" value="Demostración de leyenda."/> <input type="button" value="+"/> <input type="button" value="-"/>
Cargar Demo	<input type="text"/> <input type="button" value="Examinar..."/> *

(*) Los campos señalados no pueden quedar vacíos.

Anexo 6.



Repositorio de Estándares

DISEÑO DE INTERFAZ DE USUARIO

Usuario **admin**

Privilegios **Administrador**

- Usuarios
- Editar Perfil
- Agregar
- Búsqueda/Listado
- Estándares
- Agregar
- Búsqueda/Listado >>
- Recursos
- Agregar
- Búsqueda/Listado


Cerrar Sesión

Buscar - ESTÁNDAR

Buscar por:

Nombre <input style="width: 95%;" type="text"/>	Descripción <input style="width: 95%;" type="text"/>
---	--

Resultados de búsqueda.

Estándares	Nº Demos	Descripción
<input type="radio"/> Agregar	5	Este estándar es utilizado en los entornos agregar o nuevo como también se le llama
<input type="radio"/> Buscar	2	Entornos de búsqueda.
<input type="radio"/> Búsqueda Avanzada	1	Entornos de búsqueda avanzada.
<input type="radio"/> Calendario	1	El Calendario se utiliza para que el usuario cargue la fecha.
<input type="radio"/> Campo Descripción	1	Este estándar da solución a los campos de descripciones.
<input type="radio"/> Campo ListaMenú	1	Trata acerca cómo debe ser el ListaMenú o Select.
<input type="radio"/> CampoTexto	1	Trata acerca cómo debe ser el Campo de Texto.
<input type="radio"/> Contenedor de Opciones	1	Entorno del estándar a seguir pa utilizar el contenedor de opciones.
<input type="radio"/> DIY Movil	3	Dan sensación de estar trabajando con una aplicación de desktop.
<input type="radio"/> Editar	1	Entorno editar.

1 - 10 / 14

Página #

SISalud - Sistema de Información para la Salud

Anexo 7.



Repositorio de Estándares
DISEÑO DE INTERFAZ DE USUARIO

Usuario **admin**

Privilegios **Administrador**

- Usuarios
- Editar Perfil
- Agregar
- Búsqueda/Listado
- Estándares**
- Agregar
- Búsqueda/Listado >>
- Recursos
- Agregar
- Búsqueda/Listado



Cerrar Sesión

🔍 Buscar - DEMO

🔍 **Buscar por:**

Nombre Descripción

Entre los criterios de búsqueda y presione "Buscar".

	Pertenece al Estándar	Descripción
<input type="radio"/>	ListaMenú	Lista menú o select como también es llamado.
<input type="radio"/>	CampoTexto	Apariencia de un campo de texto.
<input type="radio"/>	Contenedor de Opciones	Es utilizado como contenedor de opciones.
<input type="radio"/>	DIVMovil1	Contiene 1 DIV Móvil.
<input type="radio"/>	DIVMovil2	Contiene 2 DIV Móviles.
<input type="radio"/>	DIVMovil3	Contiene 3 DIV Móviles.
<input type="radio"/>	Editar	Entorno simple para editar.
<input type="radio"/>	Leyenda	Demostración de leyenda.
<input type="radio"/>	Listado1	Listado sin radiobutton ni checkbox.
<input type="radio"/>	Listado2	Listado con radiobutton.

11 - 20 / 27

Página #

SISalud - Sistema de Información para la Salud

114

Anexo 8.

Repositorio de Estándares
DISEÑO DE INTERFAZ DE USUARIO

Usuario: admin
Privilegios: Administrador

Estándar - Pestañas

Editar Eliminar

Descripción
Trata acerca el Multitab o las pestañas.

Listado - DEMOS DEL ESTÁNDAR

Nombre	Descripción
<input type="radio"/> <u>Pestañas2</u>	Contiene 2 pestañas.
<input type="radio"/> <u>Pestañas3</u>	Contiene 3 pestañas.
<input type="radio"/> <u>Pestañas4</u>	Contiene 4 pestañas.
<input type="radio"/> <u>Pestañas5</u>	Contiene 5 pestañas.

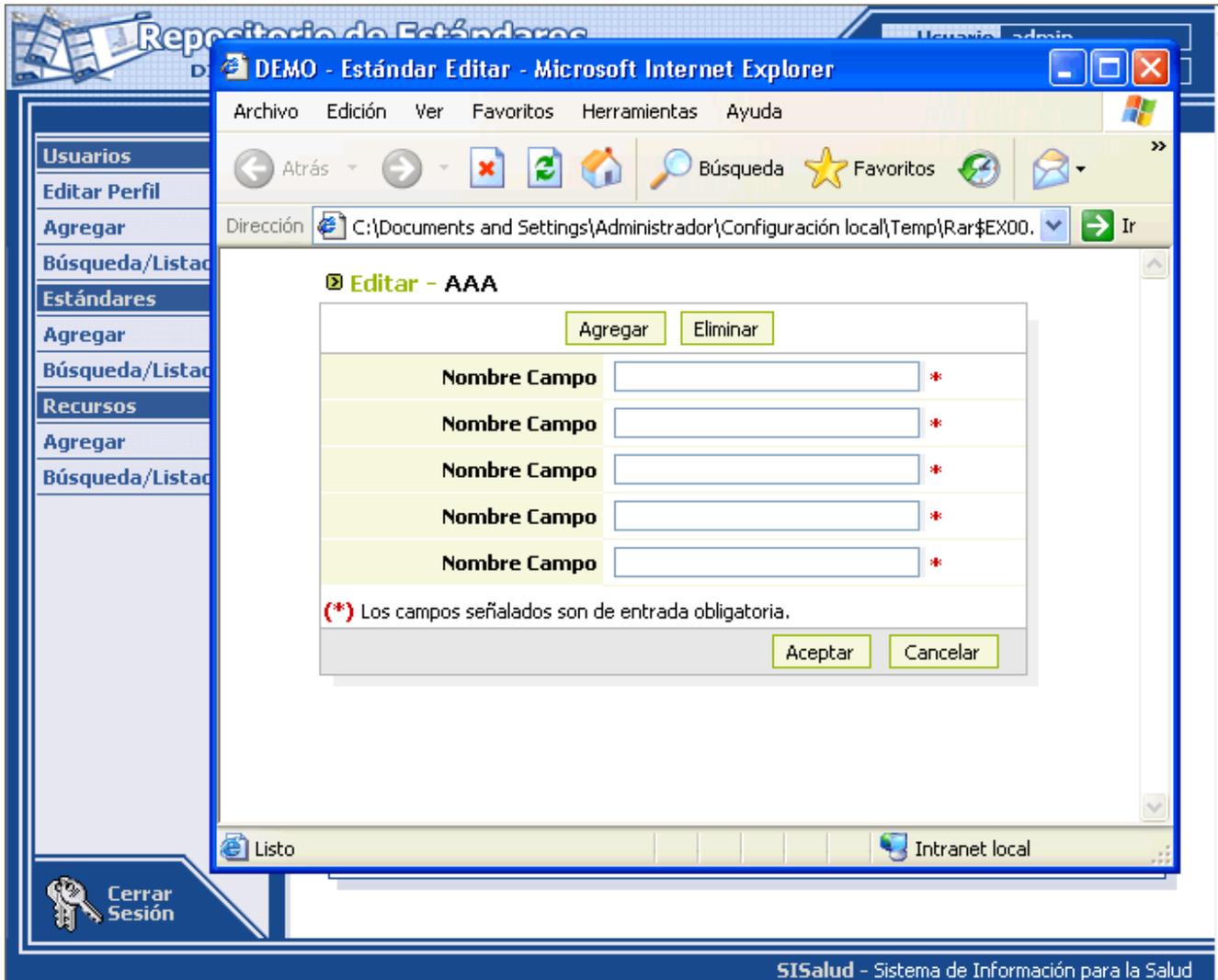
Agregar Editar Eliminar

Listado

Cerrar Sesión

SISalud - Sistema de Información para la Salud

Anexo 9.



Anexo 10.



Repositorio de Estándares
DISEÑO DE INTERFAZ DE USUARIO

Usuario **admin**

Privilegios **Administrador**

- Usuarios
- Editar Perfil
- Agregar
- Búsqueda/Listado
- Estándares
- Agregar
- Búsqueda/Listado >
- Recursos
- Agregar
- Búsqueda/Listado



Cerrar Sesión

🔍 Buscar - DEMO

Buscar por:

Nombre

Entre los criterios:

Nombre	Pe
<input type="radio"/> Campo	Car
<input type="radio"/> ListaMenú	Car
<input type="radio"/> CampoTexto	Car
<input type="radio"/> Contenedor de Opciones	Cor
<input type="radio"/> DIVMovil1	DI
<input type="radio"/> DIVMovil2	DI
<input type="radio"/> DIVMovil3	DI
<input checked="" type="radio"/> Editar	Edi
<input type="radio"/> Leyenda	Leyenda
<input type="radio"/> Listado1	Listado
<input type="radio"/> Listado2	Listado

		Demostración de leyenda.
		Listado sin radiobutton ni checkbox.
		Listado con radiobutton.

11 - 20 / 27

Página #

Descarga de archivo

¿Desea abrir o guardar este archivo?

Nombre: Editar.rar

Tipo: WinRAR archive, 50,6 KB

De: 10.128.6.66

Preguntar siempre antes de abrir este tipo de archivos

Los archivos procedentes de Internet pueden ser útiles, pero algunos archivos pueden dañar potencialmente su equipo. Si no confía en el origen, no abra ni guarde este archivo. [¿Cuál es el riesgo?](#)

Anexo 11.

The screenshot displays the 'Repositorio de Estándares' web application interface. At the top, the title 'Repositorio de Estándares' and subtitle 'DISEÑO DE INTERFAZ DE USUARIO' are visible. The user is logged in as 'admin' with 'Administrador' privileges. A left-hand navigation menu includes options like 'Usuarios', 'Estándares', and 'Recursos'. The 'Recursos' section is expanded, and the 'Agregar' option is selected, with a tooltip 'Agregar Recurso' appearing. The main content area shows a form titled 'Agregar - RECURSO' with the following fields: 'Nombre' (text input, required), 'Tipo' (dropdown menu, required), 'Cargar Recurso' (text input and 'Examinar...' button, required), and 'Descripción' (text area with expand/collapse buttons). A note at the bottom of the form states: '(*) Los campos señalados son de entrada obligatoria.' At the bottom right of the form are 'Aceptar' and 'Cancelar' buttons. A 'Cerrar Sesión' button is located in the bottom left corner of the application frame. The footer of the page reads 'SISalud - Sistema de Información para la Salud'.

- Usuarios
- Editar Perfil
- Agregar
- Búsqueda/Listado
- Estándares
- Agregar
- Búsqueda/Listado >
- Recursos
- Agregar
- Búsqueda/Listado

 Cerrar Sesión

✖ Editar - RECURSO

<input type="button" value="Agregar"/> <input type="button" value="Eliminar"/>	
Nombre	<input type="text" value="Dreamweaver 8"/> *
Tipo	<input type="text" value="Herramienta"/> *
Descripción	<input type="text" value="Para creación de páginas web profesionales, en su versión 8 en español."/> + -
Cargar Recurso	<input type="text"/> <input type="button" value="Examinar..."/> *

(*) Los campos señalados no pueden quedar vacíos.

Anexo 12.



Repositorio de Estándares
DISEÑO DE INTERFAZ DE USUARIO

Usuario admin

Privilegios Administrador

- Usuarios
- Editar Perfil
- Agregar
- Búsqueda/Listado
- Estándares
- Agregar
- Búsqueda/Listado >
- Recursos
- Agregar
- Búsqueda/Listado ✓

Buscar - RECURSO

Buscar por:

Nombre Tipo « Seleccione »

Resultados de búsqueda.

Nombre	Tipo	Descripción
<input type="radio"/> Código Planeta	Herramienta	Ofrece algunas demostraciones en JavaScript y permite ver su código.
<input type="radio"/> Color Impact	Herramienta	Esta es una herramienta profesional para crear los esquemas de colores que armonizan. Versión 3.1
<input type="radio"/> páginas web	Tutorial	Diseño de páginas web, curso de iniciación.
<input type="radio"/> Dreamweaver 8	Herramienta	Para creación de páginas web profesionales, en su versión 8 en español.
<input type="radio"/> Fireworks 8	Herramienta	Es para el diseño de páginas web.
<input type="radio"/> Imágenes y Estilos AZULES	Hoja de Estilo	Contiene las imágenes y Estilos para hacer un entorno azul.
<input type="radio"/> Imágenes y Estilos VERDES	Hoja de Estilo	Contiene las imágenes y estilos para hacer un entorno verde.
<input type="radio"/> keygen Dreamweaver y Fireworks	Otro	keygen para validar el Dreamweaver y Fireworks.
<input type="radio"/> Mozilla Firefox	Otro	Navegador Mozilla Firefox
<input type="radio"/> MySQL Front	Herramienta	MySQL Front en su versión 3.2.10.6

1 - 10/14

Página # 1

 **Cerrar Sesión**

SISalud - Sistema de Información para la Salud

Anexo 13.



Repositorio de Estándares
DISEÑO DE INTERFAZ DE USUARIO

Usuario **admin**

Privilegios **Administrador**

- Usuarios
- Editar Perfil
- Agregar
- Búsqueda/Listado
- Estándares
- Agregar
- Búsqueda/Listado >
- Recursos
- Agregar
- Búsqueda/Listado



Cerrar Sesión

🔍 Buscar - RECURSO

Buscar por:

Nombre

Resultados de búsqueda

Nombre ▲

- Código Planeta
- Color Impact
- Diseño de páginas w
- Dreamweaver 8
- Fireworks 8
- Imágenes y Estilos AZULES
- Imágenes y Estilos VERDES
- keygen Dreamweaver y Fireworks
- Mozilla Firefox
- MySQL Front

Nombre	Categoría	Descripción
Imágenes y Estilos AZULES	Hoja de Estilo	Contiene las imágenes y estilos para hacer un entorno azul.
Imágenes y Estilos VERDES	Hoja de Estilo	Contiene las imágenes y estilos para hacer un entorno verde.
keygen Dreamweaver y Fireworks	Otro	keygen para validar el Dreamweaver y Fireworks.
Mozilla Firefox	Otro	Navegador Mozilla Firefox
MySQL Front	Herramienta	MySQL Front en su versión 3.2.10.6

1 - 10 / 14

Página # 1 ▼

Descarga de archivo - Advertencia de seguridad

¿Desea ejecutar o guardar este archivo?



Nombre: Fireworks8-es.exe

Tipo: Aplicación, 88,5 MB

De: 10.128.6.66



Los archivos procedentes de Internet pueden ser útiles, pero este tipo de archivo puede dañar potencialmente su equipo. Si no confía en el origen, no ejecute ni guarde este software. [¿Cuál es el riesgo?](#)

SISalud - Sistema de Información para la Salud

GLOSARIO DE TÉRMINOS.

Administrador: Usuario con privilegios de Administración, tiene a su cargo la gestión de la aplicación.

Demo: Abreviatura de la palabra demostración. Ejemplo práctico que ejemplifica un estándar tanto en apariencia como en funcionamiento.

Estándar: Es un requisito, regla, especificación o recomendación basada en principios probados y en la práctica. Se aplican generalmente a características básicas de la interfaz de usuario. Con el hecho de desarrollar estándares para la interfaz se intenta conseguir un software más fácil y seguro, estableciendo unos requisitos mínimos de fabricación y eliminando inconsistencias y variaciones innecesarias en las interfaces.

Recurso: Se le denomina recurso a aquel elemento que proporcione algún tipo de información o aplicación como es el caso de archivos, documentos y herramientas que se publican en la aplicación, y que están relacionados con el diseño de interfaz y la programación.

Repositorio: Sitio centralizado donde se almacena y mantiene información, habitualmente bases de datos o archivos informáticos. Puede ser de acceso público o estar protegido y necesitar de previa autenticación.

URL (Uniform Resource Locator): (*Localizador Uniforme de Recursos*): Sistema unificado de identificación de recursos en la red. Las direcciones se componen de protocolo, FQDN y dirección local del documento dentro del servidor. Este tipo de direcciones permite identificar objetos WWW, Gopher, FTP, News,... Ejemplo de URL: <http://www.yahoo.es>

Usuario: Personal que accede a la aplicación, en este caso los usuarios serán los miembros del proyecto

Visualizador: Usuario con privilegios de Visualizador, accede al repositorio para consultar o descargar los estándares de diseño de interfaz y los recursos que se brindan en este.