

Universidad de las Ciencias Informáticas
Facultad 7



**Título: Herramienta para la administración de
repositorios Subversion (PhpSvnAdmin)**

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

Autoras

Clenda Pérez Bisset
Kenia María Mojena Fis

Tutor

Ing. Yovannys Sánchez Corales

Ciudad de La Habana, Junio del 2009

“Año del 50 aniversario del triunfo de la Revolución”

DECLARACIÓN DE AUTORÍA

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

Para que así conste firmamos la presente a los 18 días del mes de junio del año 2009.

Clenda Pérez Bisset

Kenia María Mojena Fis

Ing. Yovannys Sánchez Corales

DATOS DE CONTACTO

Ing. Yovannys Sánchez Corales: Profesor graduado de Ingeniero en Informática en el año 2005 en la CUJAE. Ha impartido las asignaturas Programación 3, Segundo Perfil e Inteligencia Artificial. Forma parte del proyecto de Atención Primaria de Salud de la facultad 7, ejerciendo el rol de Arquitecto. Pasó cursos de diplomado para la maestría GPI (Gestión de Proyectos Informáticos), actualmente cursa la maestría. Ha participado en eventos científicos nacionales e internacionales. Su dirección de correo es yscorales@uci.cu.

AGRADECIMIENTOS

Agradecemos:

A todos nuestros profesores, por haber contribuido a nuestra formación durante toda la carrera. A nuestro tutor Yovannys Sánchez Corales, por ser también nuestro amigo y apoyo. A Maykell por siempre estar ahí cuando lo necesitamos.

A nuestro Planí (Alejandro Planos) por ser tan especial con nosotras. A nuestro Píngui (Luis Javier), Yorch (Yordanger) y Lui (Luis José). A Tara (Alejandro Tarafa) por ser incondicional sin importar la hora ni el día. A todos nuestros amigos que nos han acompañado en estos cinco años en las malas y en las buenas. A la Revolución y a Fidel por darnos la oportunidad de convertirnos en ingenieras.

Kenia:

Quiero agradecer en primer lugar a mi madre. Junto a ella agradezco a mi hermana, a mi tía, a mi abuela a mi tío, a mi papá, a Marino, y a mi otra hermana Neyled.

Agradezco a mi compañera de tesis, a mi Clendita, pues hacer la tesis con ella es lo mejor que me pudo pasar y la alegría más grande e inimaginable.

Agradezco a mi Juan Pablo, por cuidarme, ayudarme, apoyarme y quererme.

Agradezco a otras personas que fuera conociendo durante la carrera y que vivieran conmigo salud y enfermedad: Yisle, Cosí, Mary, Lianys, Elis, Irina (Michi Mía), Yusdelis, Yudi, Jolie...

Clenda:

A mamá por ser lo más grande en mi mundo, eres especial, por eso te amo.

A mi papí por ser como es, me ha enseñado mucho y gran parte de lo que soy se lo debo a él.

A mi hermanita, que no dudó nunca que alcanzaría esta meta.

A mis tías, tíos y primos, por su preocupación constante.

A mi AMIGO Alejandro Planos...

A mis amigas La morita, La osita, Ima, Darma, Baby, por estar siempre ahí.

A mi sobrinito Ernestico, por regalarme una sonrisa cada fin de semana.

A mis amigas uci Yudita la Linda, Elizabeth, Ana, Lichí, Lisi, El buche, Naza, Mary, Michi, Lore, Azúcar, Jolie, por su incondicionalidad y apoyo.

A mi Tato, por su amor y comprensión.

A mis amigos de la pizzería Eduardo y Enrique, por quererme y cuidarme.

A las chicas del 51106, 80106, 85103, y las chinas del 83201, por soportarme. A los chicos de la cancha y A los chicos del Merengón.

DEDICATORIA

Kenia:

Dedico esta tesis a mi madre querida que siempre ha estado junto a mí dándome fuerzas y que es la verdadera causa de que me convierta en ingeniera.

A mi hermana por ser el mejor apoyo del mundo.

A mi tía por confiar en mí siempre y darme el amor de una madre.

A mi abuelo porque no está presente, pero sé que estaría muy orgulloso de mí.

A mi padre, por haberme aconsejado y guiado.

A mi abuela y a mi tío por ser más que eso en mi vida.

Al resto de mi familia por haber estado siempre que lo necesité.

A mi CT Clendita, por ser mi sostén cuando pensaba que me caía, mi hombro para llorar y mi boca para reír.

Clenda:

En especial a mi Abu, por guiarme y confiar en que este sueño se haría realidad.

A mi títí, que su cariño no deja de acompañarme.

A mi ma' y mi pa', por ser los mejores padres del mundo.

Y a Kenita mi CT y AMIGA, este no es el final, es el inicio de una amistad que ni el destino podrá separar.

RESUMEN

En la actualidad para la producción de software es indispensable el uso de Sistemas de Control de Versiones (SCV). Uno de los SCV más usados a nivel mundial es Subversion y para interactuar con este se ha generalizado el uso de numerosos clientes. El presente trabajo tiene como objetivo desarrollar una herramienta que facilite la administración de los repositorios contenidos en un sistema de control de versiones Subversion.

La aplicación está desarrollada sobre estándares libres, y se basa en las pautas establecidas por el Documento de Arquitectura de la facultad 7 para el desarrollo de aplicaciones informáticas. Se utilizó como metodología RUP (Proceso de Desarrollo Unificado), acompañada del lenguaje UML (Lenguaje Unificado de Modelado) y Visual Paradigm como herramienta para el modelado visual. Para la creación del prototipo de interfaz de usuario se empleó el editor visual Macromedia Dreamweaver 8.0. Además se empleó una arquitectura en tres capas.

La herramienta propuesta permitirá automatizar la gestión de la administración de un repositorio Subversion, lo que garantizará la agilidad y rapidez de este proceso en los proyectos productivos de la Universidad de las Ciencias Informáticas, debido a que actualmente este proceso, que se torna complejo en proyectos de grandes dimensiones, se realiza en su totalidad manualmente.

ÍNDICE

DECLARACIÓN DE AUTORÍAI

DATOS DE CONTACTO.....II

AGRADECIMIENTOSIII

DEDICATORIA IV

RESUMEN..... V

INTRODUCCIÓN 1

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA..... 6

 1.1 Sistemas de Control de Versiones..... 6

 1.2 Herramientas para la gestión de sistemas de control de versiones.....7

 1.2.1 CVS7

 1.2.2 Subversion.....8

 1.3 Gestión de la configuración 10

 1.4 Análisis de las soluciones existentes 11

 1.4.1 Aplicaciones clientes para interactuar con un sistema de control de versiones Subversion 11

 1.4.1.1 TortoiseSVN..... 12

 1.4.1.2 RapidSVN..... 13

 1.4.2 Herramienta para administración de repositorios 13

 1.4.2.1 VisualSVN 13

 1.4.3 Clientes web para interactuar con un sistema de control de versiones Subversion 15

 1.4.3.1 WebSVN: Navegador de repositorios Subversion en línea 15

1.4.3.2 WebClient para SVN: Herramienta Libre de Subversion	15
1.5 Aplicaciones Web	16
1.6 Servidor Web Apache	17
1.7 Navegadores	18
1.8 Modelo cliente-servidor	18
1.9 Selección de tecnologías, herramientas y metodologías a utilizar	20
1.9.1 UML: lenguaje de modelado	20
1.9.2 Metodologías de Desarrollo	20
1.9.2.1 RUP	20
1.9.3 Lenguajes de Programación Web	21
1.9.3.1 PHP	21
1.9.3.2 JavaScript	23
1.9.3.3 Bash	24
1.9.4 Herramientas	24
1.9.4.1 Visual Paradigm	24
1.9.4.2 Rational Rose	25
1.9.4.3 Macromedia Dreamweaver 8	26
1.9.4.4 Zend Studio	27
Conclusiones	28
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA	29
2.1 Modelo de Dominio	29
2.1.1 Conceptos fundamentales	29
2.1.2 Modelo del Diagrama del Dominio	30
2.2 Propuesta de Sistema	31

2.2.1 Especificación de Requerimientos de Software.....	31
2.2.1.1 Requerimientos Funcionales	31
2.2.1.2 Requerimientos No Funcionales	33
2.3 Modelo del Sistema	40
2.3.1 Actores del sistema.....	40
2.3.2 Diagrama de Casos de Uso del Sistema.....	41
2.3.3 Descripción textual de los CU del Sistema.....	41
Conclusiones.....	56
Capítulo 3: ANÁLISIS Y DISEÑO DEL SISTEMA	57
3.1 Análisis.....	57
3.1.1 Modelo de análisis	57
3.1.1.2 Diagramas de clases del análisis.....	57
3.1.1.3 Diagramas de interacción	62
3.2 Diseño	75
3.2.1 Modelo de Diseño	75
3.2.1.1 Diagrama de clases de diseño	75
Conclusiones.....	84
Capítulo 4: IMPLEMENTACIÓN Y PRUEBA.....	85
4.1 Componente de Seguridad (SAAA)	85
4.2 Implementación	85
4.2.1 Modelo de Implementación	86
4.2.2 Diagrama de Componentes	87
4.2.3 Diagrama de Despliegue.....	90
4.3 Estándares de Codificación	91



Índice

4.4 Tratamiento de Errores.....	93
4.5 Prueba.....	94
Conclusiones.....	96
CONCLUSIONES	97
RECOMENDACIONES	98
REFERENCIA BIBLIOGRÁFICA.....	99
BIBLIOGRAFÍA.....	101
ANEXOS.....	104
GLOSARIO DE TÉRMINOS.....	108

INTRODUCCIÓN

Las últimas décadas del siglo XX y los inicios del XXI han sido testigos del desarrollo acelerado que han experimentado, y en el que aún se encuentran, las Tecnologías de la Información y las Comunicaciones (TIC). Estas tecnologías se convirtieron en las protagonistas de los grandes cambios y transformaciones ocurridas en todos los campos de la sociedad. Han influenciado poderosamente los procesos económicos, sociales y culturales; así como las formas de educación, interacción y comunicación entre las personas y los organismos de la sociedad.

Cuba no se encuentra al margen de este proceso que supone la incorporación masiva de estas herramientas en la sociedad. Mediante su uso racional y adecuado en todos los sectores, y con el objetivo de lograr una mejor eficacia y eficiencia; se pretende alcanzar una mayor generación de riquezas y el aumento de la calidad de vida de los ciudadanos cubanos. La nación cubana ha diseñado y comenzado la aplicación de numerosas estrategias para convertir los conocimientos y las tecnologías de la información y las comunicaciones en herramientas disponibles que permitan el avance y las profundas transformaciones revolucionarias.

En este sentido se han alcanzado numerosos logros, tanto que se puede asegurar que las TIC se han insertado en casi todas las ramas de la sociedad cubana. Se planifican metas a la altura de los países del primer mundo y hoy, algunas de ellas no están distantes de alcanzarlas. Dentro de estas, se encuentra el surgimiento de una Industria Cubana del Software, que con el protagonismo de la Universidad de las Ciencias Informáticas (UCI) y otras empresas cubanas productoras de software, ha alcanzado numerosos logros y continúa ampliando y perfeccionando su trabajo.

La producción de software en la Universidad de las Ciencias Informáticas, al igual que en cualquier otra entidad productora de software, se realiza en equipos o grupos de trabajo. Cuando se produce en grupo, surge la necesidad de coordinar las acciones, es imprescindible la buena comunicación y entendimiento entre sus miembros. Esta forma de trabajo acarrea ciertas incomodidades técnicas: al haber más de una persona modificando el código fuente de forma simultánea, aumenta la complejidad; al tener que sincronizarlo y mantenerlo coherente entre todos los miembros del grupo. También ocurren relaciones

asimétricas respecto del código, cada grupo tiene una forma y un flujo de trabajo particular. Por ejemplo, se pueden dar relaciones jerárquicas, revisión de código entre pares, subgrupos, etc.

Lo que va a reflejarse en el código fuente con el surgimiento de nuevas necesidades: la integración de múltiples trabajos individuales y la distribución del mismo en distintas máquinas. Así como la coordinación para que todos puedan trabajar sobre el mismo código fuente. Para resolver estas dificultades, la mayoría de las entidades desarrolladoras de software usan una herramienta conocida como sistema de control de versiones. La que se utiliza para permitir el desarrollo concurrente, mantener la historia del código fuente y parte de la documentación producida en el proyecto.

Al tratarse de proyectos informáticos lo normal es que se trabaje con código fuente y con documentos que van evolucionando a lo largo del desarrollo. Estos deben ser modificados por múltiples personas, lo que hace casi imprescindible disponer de un sistema de control de versiones que permita mantener la historia de los ficheros generados y que más de una persona trabaje concurrentemente sobre el mismo código.

Aunque un sistema de control de versiones puede realizarse de forma manual, es muy aconsejable disponer de herramientas que faciliten esta gestión, algunas de estas pueden ser: CVS, Subversion, SourceSafe, ClearCase, Darcs, Plastic SCM, Git, Mercurial, etc.

En la Universidad de las Ciencias Informáticas se usa como sistema de control de versiones el Subversion (SVN) en la mayoría de los proyectos productivos. Como cliente para interactuar con éste, se ha generalizado el uso de TortoiseSVN en los proyectos desarrollados bajo el sistema operativo Windows y RapidSVN en los que utilizan el sistema operativo Linux, debido a sus características, que los convierten en los candidatos más usados a escala mundial para esta aplicación. Una característica común para ambas herramientas es que poseen una interfaz gráfica amigable, lo que las hace favoritas ante otras aplicaciones de su tipo.

TortoiseSVN es un cliente gratuito de código abierto para el sistema de control de versiones Subversion. Esta herramienta maneja ficheros y directorios a lo largo del tiempo, los que se almacenan en un repositorio central. Esto permite que pueda recuperar versiones antiguas de sus ficheros y examinar la historia de cuándo y cómo cambiaron sus datos, y quién hizo el cambio. RapidSvn es fácil de usar y permite acceder a direcciones SVN, subir y descargar contenido y sincronizarlo con el servidor original, comprobar su estado, crear y fusionar direcciones, etc. Además tiene la ventaja de funcionar en varias plataformas y de disponer de un completo manual en línea.

A pesar de poseer características ventajosas las herramientas TortoiseSVN y RapidSVN no brindan la posibilidad de administrar los repositorios, es decir, no permiten: la creación, eliminación y actualización de usuarios; el acceso de estos a las acciones sobre un repositorio; la asignación de permisos y además son aplicaciones de escritorio, por lo que para su uso, deben estar instaladas en cada PC vinculada al proyecto productivo.

Al no brindar estas posibilidades el proceso de administración del repositorio Subversion se torna lento, pues la persona encargada de realizar estas acciones, debe emplear tiempo en verificar si existe un repositorio contenido en un sistema de control de versiones Subversion. Luego debe comprobar que dentro de este exista el proyecto al cual se desean gestionar la administración. Finalmente debe buscar los archivos de configuración que permiten el control de este proceso, teniendo que acceder a numerosas carpetas. La ejecución de este proceso se realiza manualmente, por lo que ante la tarea de crear usuarios y otorgar permisos para grandes proyectos, esta actividad se torna tediosa y poco eficiente.

Existe una herramienta VisualSVN, que no es más que un complemento que se integra perfectamente con Visual Studio y que brinda la posibilidad de gestionar la administración del repositorio. Es decir, con esta herramienta se pueden administrar usuarios y grupos de usuarios, así como sus permisos. Además, permite realizar acciones sobre los archivos, modificarlos, revertir cambios, actualizarlos y aplicarlos, además de administrar los cambios para todo el proyecto.

A pesar de todas las ventajas que brinda, el componente VisualSVN presenta varios inconvenientes: es un software propietario, o sea, no gratuito; funciona solamente para el sistema operativo Windows y es una aplicación de escritorio; además, para su funcionamiento es necesario que esté instalada la herramienta TortoiseSVN y su acceso es sólo a través del entorno de Visual Studio.

Teniendo en cuenta estas dificultades y después de concluir que no existe otra herramienta que pueda solucionarlas, se identifica como **Problema científico**: Inexistencia de una herramienta que permita administrar eficientemente los repositorios contenidos en un sistema de control de versiones Subversion.

Se define como **Objeto de estudio**: Proceso de control de repositorios contenidos en un sistema de control de versiones. Se delimita como **Campo de Acción**: proceso de administración de los repositorios contenidos en un sistema de control de versiones Subversion.

Para la solución del problema planteado se define como **Objetivo general**: Desarrollar una herramienta que facilite la administración de los repositorios contenidos en un sistema de control de versiones Subversion.

Para dar cumplimiento al objetivo general, se trazan las siguientes **Tareas de la investigación**:

- ✓ Describir las herramientas y tecnologías informáticas de interacción con el Subversion.
- ✓ Asimilar las herramientas CVS (Concurrent Version System) y SVN (Subversion).
- ✓ Documentar los Flujos de Trabajo “Modelamiento del Negocio”, “Gestión de Requerimientos”, “Análisis”, “Implementación” y “Prueba”.
- ✓ Documentar los Flujos de Trabajo: “Diseño” e “Implementación”.
- ✓ Implementar la herramienta PhpSvnAdmin, utilizando estándares de codificación, tratamiento de errores o excepciones.

El contenido del trabajo está estructurado en cuatro capítulos, distribuidos de la siguiente manera:

Capítulo 1. Fundamentación teórica: se mencionan algunos aspectos fundamentales del sistema de control de versiones Subversion, haciendo referencia a las herramientas existentes para interactuar con la mencionada aplicación. Se profundizará, además, en los principales conceptos asociados al dominio del problema, así como en las herramientas y tecnologías a utilizar en el desarrollo de la herramienta propuesta.

Capítulo 2. Características del sistema: se determinan los requerimientos funcionales y no funcionales, los actores del sistema, casos de uso del sistema. Se realizan además, los diagramas de casos de uso del sistema, la descripción de los casos de uso del sistema, y además se construye la Vista de Casos de Uso.

Capítulo 3. Análisis y diseño del sistema: se representan los diagramas de las clases para aplicaciones Web con sus relaciones, los diagramas de interacción, se construyen los prototipos no funcionales, se hace referencia a la arquitectura del sistema a través de la Vista Lógica y de Despliegue, y se realiza la validación del Diseño propuesto.

Capítulo 4. Implementación y Prueba: aborda los aspectos correspondientes a la construcción de la herramienta propuesta, se modelan los diagramas de componentes y despliegue, aborda la descripción de

los estándares de diseño, codificación y además del tratamiento de errores en la solución del sistema. Además se aborda el tema relacionado con las pruebas que se le realizan a la herramienta propuesta.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Este capítulo tiene como objetivo fundamental explicar los conceptos fundamentales que sirven de soporte teórico para el desarrollo de la aplicación PhpSvnAdmin. Además se realiza el análisis de las herramientas, tecnologías y metodologías existentes para el desarrollo de sistemas que implementen funcionalidades similares a las que debe cumplir la herramienta PhpSvnAdmin.

1.1 Sistemas de Control de Versiones

Un Sistema de Control de Versiones es un sistema de gestión de archivos y directorios, cuya principal característica es que mantiene la historia de los cambios y modificaciones que se han realizado sobre ellos a lo largo del tiempo. [1]

Estas aplicaciones permiten identificar, revertir, comparar y almacenar los cambios que ocurren en la información, así como el manejo del desarrollo paralelo de un proyecto. De esta forma el sistema de control de versiones tiene la capacidad de “recordar” versiones antiguas de los datos almacenados; esto brinda la posibilidad de valorar y examinar los cambios y de recuperar versiones anteriores de determinado fichero, aún cuando éste haya dejado de existir. (Ver **Anexo I**)

Los sistemas de control de versiones son ampliamente utilizados en la industria informática para mantener el control sobre las versiones del código fuente. Sin embargo su aplicación se extiende a otros ambientes, como la gestión de documentos, imágenes, ficheros de todo tipo, sitios web, etc.

Un sistema de control de versiones debe proporcionar:

- ✓ Un mecanismo de almacenaje para los elementos a gestionar (archivos de texto, imágenes, documentación, etc.).
- ✓ La posibilidad de efectuar transformaciones sobre los elementos almacenados (modificaciones parciales, añadir, borrar, renombrar o mover elementos).
- ✓ El mantenimiento de un registro histórico de las acciones realizadas con cada elemento o conjunto de elementos (normalmente pudiendo volver o extraer un estado anterior del producto). [2]

Garantizar la actualización de los ficheros modificados y las copias de seguridad centralizadas, es otra de las funciones de un sistema de control de versiones, pues sólo el administrador es el encargado de realizar copias de seguridad en el repositorio.

En la actualidad desarrollar un proyecto de software provoca invertir mucho tiempo y recursos, por lo que sería riesgoso no proteger esta inversión con un sistema de control de versiones. La ausencia de ésta importante herramienta significaría un alto grado de desconocimiento del desarrollo de software y a la vez podría acarrear grandes pérdidas e incluso el fracaso del proyecto.

1.2 Herramientas para la gestión de sistemas de control de versiones

Un sistema de control de versiones puede realizarse de forma manual, aunque actualmente es muy práctico el uso de herramientas para facilitar su gestión. Las dos herramientas que gozan de mayor popularidad y reconocimiento mundial ante cualquier sistema operativo son: CVS y Subversion.

1.2.1 CVS

El Sistema de Control de Versiones más conocido por sus siglas en inglés CVS (Concurrent Versions System) es una herramienta que mantiene el registro de todo el trabajo y los cambios en la implementación de un proyecto informático (de software) y permite que distintos desarrolladores (potencialmente situados a gran distancia) colaboren en el mismo. [3] (Ver **Anexo II**)

CVS es difundido bajo la Licencia Pública General de GNU (licencia GPL) más conocida por sus siglas en inglés: GNU General Public License. El hecho de ser libre, unido a sus características avanzadas, le dieron gran popularidad en la comunidad del software libre. Fue desarrollado por GNU (GNU's Not Unix), cuyo sitio lo distribuye, denominándolo "paquete GNU" con aplicaciones básicas a través de esta página.

Esta herramienta tiene una arquitectura cliente-servidor. El servidor de CVS en Linux puede ser usado tanto por clientes en Linux como en Windows, e incluso en otros sistemas operativos usados a nivel mundial. Es el encargado de almacenar todos los archivos del proyecto, así como su historial de cambios. Mientras que los clientes se conectan al servidor para obtener una copia íntegra de los archivos correspondientes a un módulo determinado.

1.2.2 Subversion

Subversion (SVN) es un software para el control de versiones que ha alcanzado gran popularidad en las comunidades de desarrolladores de software pues es distribuido bajo licencia libre de tipo Apache, es multiplataforma, está escrito en C, preparado para funcionar en red y fue diseñado específicamente con la intención de sustituir y mejorar al Concurrent Versions System. SVN conserva las ideas fundamentales que rigen el funcionamiento de CVS y su filosofía de desarrollo, pero posee la peculiaridad de suplir sus carencias y evitar sus errores. Fue desarrollado en el año 2000 por CollabNet y emplea licencia Apache/BSD.

Subversion soluciona estos problemas del CVS:

- ✓ CVS es incapaz de registrar los cambios en la estructura de directorios, pues no es posible mover, renombrar, ni copiar. Dichas operaciones se consiguen eliminando y añadiendo, pero con estas acciones se pierde el historial de cambios. Este defecto se debe a que CVS usa internamente el sistema de almacenamiento de RCS (Revision Control System), que solo registra cambios de contenido en ficheros individuales.
- ✓ En CVS es necesario interrumpir el acceso al repositorio para crear copias de seguridad.
- ✓ CVS no permite lo que se conoce como "conjuntos de cambios". Si un desarrollador sube un conjunto de cambios, estos van subiendo uno a uno, quizás al mismo tiempo que otro desarrollador realiza la misma operación. Como no es una operación atómica, no se puede asegurar el estado del repositorio tras su commit o transacción, es decir, no se asegura que el conjunto de cambios sea consistente.
- ✓ Concurrent Version System almacena ficheros binarios enteros (no sus diferencias entre versiones). Esto consume espacio en disco y ancho de banda.
- ✓ El sistema de control de versiones CVS no usa la red con eficiencia. Las diferencias entre versiones solo son enviadas desde el servidor al cliente, cuando el cliente sube sus cambios envía ficheros enteros.
- ✓ Para CVS el código fuente es difícil de mantener. CVS comenzó como un conjunto de scripts shell que usaban RCS e implementaban algoritmos desarrollados entre los años 60 y 80. El resultado del desarrollo experimentado es producto de sucesiones de parches, y no tiene un diseño fácil de entender o mejorar, lo que hace difícil su evolución.

Subversion aumenta la funcionalidad:

- ✓ Su objetivo es mejorar y ampliar las prestaciones de CVS.
- ✓ SVN registra cambios en la estructura de directorios (permite mover y renombrar sin perder el historial). El uso de la base de datos Berkeley permite aislamiento, atomicidad, recuperación de datos, integridad, backups en caliente, y concurrencia sin necesidad de usar ficheros de lock. Con Berkeley DB no se pueden editar los ficheros manualmente como con CVS, pero eso tampoco es necesario porque el repositorio no se corrompe. (Ver **Anexo III**)
- ✓ Los commits atómicos se realizan todos o ninguno. Las transacciones atómicas permiten identificar conjuntos de cambios. De esta forma cuando un desarrollador sube un conjunto de ficheros lo hace en una transacción atómica, es así que todos los ficheros se etiquetan con un número de revisión en el repositorio. La atomicidad es una característica que también impide que el repositorio quede en estado no compilable porque la red cae durante la subida de cambios.
- ✓ En Subversion, servidor y cliente intercambian diferencias entre versiones. Al enviar una nueva versión nunca es necesario transmitir ficheros enteros.
- ✓ Otra característica es la interoperabilidad con WebDAV. Es posible acceder al repositorio con cualquier software que soporte dicho protocolo ("Web Folders" de Windows XP, Photoshop, etc.).
- ✓ Apache + SSL puede usarse con firewalls y proxys.
- ✓ MIME types y detección automática de ficheros binarios.
- ✓ Permite operar directamente sobre el repositorio, sin copia local.

Subversion mejora, además, el rendimiento y diseño:

- ✓ Se usa el protocolo WebDAV/DeltaV para el protocolo de red.
- ✓ La arquitectura de red es mucho mejor: Apache 2.0, envío de diffs binarios entre cliente y servidor, datos comprimidos con mod_deflate.
- ✓ Se basa en APIs C bien definidas y documentadas. CVS en cambio, fue construido mediante sucesiones de parches.

- ✓ SVN usa la biblioteca Apache Portable Runtime, la que hace posible portar la capa de red a diferentes sistemas operativos.
- ✓ El cliente es una pequeña aplicación que usa una biblioteca de alto nivel.
- ✓ No es necesario duplicar el código en el repositorio para crear ramas. Subversion usa copia perezosa, solo se crea un nuevo fichero cuando es modificado. Mientras tanto, el fichero de la nueva rama, está implementado como un enlace al fichero original. Al contrario, CVS tarda por ejemplo 40 minutos en crear un tag de release en el servidor de GCC.
- ✓ En Subversion no es necesario la conexión a red para ciertas operaciones como: status, diff, revert. Esto ocurre porque la copia local contiene una copia del fichero original presente en el repositorio. Este comportamiento ahorra ancho de banda a costa de mayor espacio en disco.

1.3 Gestión de la configuración

“El arte de coordinar el desarrollo de software para minimizar...la confusión, se denomina gestión de la configuración. La gestión es el arte de identificar, organizar y controlar las modificaciones que sufre el software...la meta es maximizar la productividad minimizando errores.” Babich. [4]

El proceso de Gestión de la Configuración del Software (Software Configuration Management, SCM), es clave en el desarrollo de software. Esto se debe a que en el proceso de construcción de un software, los cambios generados por modificaciones de requisitos y fallos son inevitables, y a la vez provocan bastante confusión e incertidumbre porque en un gran esfuerzo de desarrollo de software, donde el proceso de cambio no es controlado, se produce inmediatamente el caos.

Generalmente en los procesos de desarrollo de software, el trabajo se realiza en equipos por lo que el control y registro de los cambios o sea el control de versiones es fundamental para elevar la calidad y productividad, minimizar errores y evitar los inconvenientes de una mala sincronización de los ya mencionados cambios, que afectan a otros elementos del sistema o a las actividades realizadas por miembros del equipo de trabajo.

La Gestión de la Configuración no constituye un proceso de mantenimiento del software, sino que es realizada a través de todas y cada una de las etapas de desarrollo, de un sistema informático. Comienza

cuando se inicia el desarrollo del software y termina sólo cuando este queda fuera de circulación e incluye además, entre sus actividades, el mantenimiento y control de cambios. Su objetivo principal es mantener la integridad de los productos alcanzados a través del desarrollo de los sistemas de información. De esta forma garantiza que no se produzcan cambios incontrolados y que los desarrolladores del sistema tengan acceso a la versión adecuada de los productos usados. Es por ello que, ejecutables y código fuente no son los únicos elementos de configuración de software, se incluyen además especificaciones de requisitos, pruebas, modelos de datos, modelos de procesos, etc.

La Gestión de la Configuración del Software facilita el mantenimiento del sistema, pues brinda información que es necesaria para valorar el impacto de los cambios solicitados, además de minimizar el tiempo de implementación que necesita un cambio, evolutivo o correctivo. A la vez, posibilita el control del sistema como un producto global durante todo su ciclo de desarrollo, obtener informes que brinden información sobre el estado en que se encuentra y reducir los errores de adaptación; lo que conlleva a elevar la calidad del producto, la satisfacción del cliente, y la consecuente mejora de la organización.

La Gestión de la Configuración de Software establece las bases sobre las que se realizará la gestión de toda la infraestructura de red; sin ésta, colapsarían el resto de las funciones (seguridad, optimización del rendimiento, etc.). Evidentemente algunas tareas, como el aislamiento de fallos, no serían posibles si no se tienen en cuenta determinados detalles acerca del estado de la configuración de los dispositivos que experimentan problemas. Otra ventaja evidente radica en que no existe duda de que, al prever problemas, la gestión de la configuración permite mejorar el control de los costes y la disponibilidad. En otro sentido el proceso de Gestión de la Configuración de Software posibilita un aumento en la efectividad de las herramientas de gestión existentes en la organización.

1.4 Análisis de las soluciones existentes

1.4.1 Aplicaciones clientes para interactuar con un sistema de control de versiones Subversion

Subversion (SVN) es un sistema de control de versiones donde los miembros del grupo de desarrollo, obtienen una copia para su trabajo, lo que se conoce como clientes del repositorio, éstos pueden ser sincronizados a través de líneas de comando o usando una interfaz gráfica, en dependencia del sistema operativo que se esté utilizando. Para el sistema operativo Windows, TortoiseSVN es la interfaz que ha

alcanzado mayor popularidad en la comunidad de desarrolladores de software, al proporcionar entre otras características ventajosas, la propiedad de integrarse con el explorador de ésta plataforma. También el cliente RapidSVN, obedeciendo a todas las facilidades que brinda, se ha convertido en la herramienta más usada para interactuar con Subversion en el sistema operativo Linux.

1.4.1.1 TortoiseSVN

TortoiseSVN es un cliente para Subversion, implementado de forma muy práctica, que se integra perfectamente al entorno de Windows. Es un software libre, distribuido según la licencia GNU/GPL; fácil de usar y que brinda un acceso más rápido y visual a la generalidad de las funciones de Subversion. Este cliente es ejecutable sobre Windows 2000 SP2, Windows XP o superiores. La última versión es TortoiseSVN 1.4.7, que vio la luz el 5 de enero de 2008.

El uso de esta herramienta se ha popularizado en la comunidad mundial de desarrolladores de software debido a sus características, por lo que se ha convertido en el cliente perfecto para el Subversion. TortoiseSVN se integra perfectamente con el shell de Windows, es decir, puede seguir trabajando con las herramientas conocidas, no tiene que cambiar a una aplicación diferente cada vez que necesite las funciones del control de versiones y tampoco está obligado a usar el Explorador de Windows. Este cliente proporciona fácil acceso a los comandos de Subversion, pues añade su propio menú en el menú contextual del explorador donde están disponibles los comandos de este sistema de control de versiones.

Otra característica de TortoiseSVN es que posee imágenes que decoran los íconos de los archivos mostrando qué archivos o directorios necesitan ser enviados al repositorio. Además puede ser usado aún cuando no exista un entorno de desarrollo y está disponible en 28 idiomas diferentes (multididioma). Posee, además, otras características que lo hacen atractivo para la generalidad de usuarios, por ejemplo, posee corrector ortográfico integrado para mensajes de registro, terminación automática de las trayectorias y las palabras claves de los archivos modificados y formato de texto con caracteres especiales. TortoiseSVN proporciona un mecanismo flexible para integrar a cualquier web de seguimiento de fallos de sistema. Es estable y posee una herramienta de informes personalizados de choque, lo que ayuda a corregir los errores mucho más rápido. Es una herramienta rápida y descriptiva, cuya documentación está disponible en varios idiomas y formatos; además de poseer una nutrida guía de preguntas frecuentes y los problemas con todas las respuestas disponibles.

Un sistema de control de versiones soluciona el problema que significa para el sistema compartir información entre usuarios, evitando que estos se pisoteen unos a otros, incluso de manera accidental. El cliente TortoiseSVN presenta la solución copiar-modificar-fusionar (Ver **Anexo IV**). El mencionado modelo consiste en que el cliente para cada usuario realiza la lectura del repositorio y produce una copia personal de trabajo, ya sea del fichero o del proyecto. De ésta forma el trabajo de los usuarios ocurre en paralelo modificando sus copias privadas y al finalizar, TortoiseSVN, permite la fusión de las copias privadas en una única versión final.

1.4.1.2 RapidSVN

Una de las herramientas más populares para el control de versiones bajo el Sistema Operativo Linux es RapidSVN, el cual es un cliente gráfico para Subversion. Permite el acceso a direcciones SVN, subir y descargar contenido y sincronizarlo con el servidor original, comprobar su estado, crear y fusionar direcciones, y otras funcionalidades, además de disponer de un manual en línea bastante completo.

Constituye la alternativa del TortoiseSVN, pues aunque su manejo es un poco más complicado que el de este cliente de Windows, posee sus mismas funcionalidades. Fue escrito en C++ utilizando el framework wxWidgets.

Es una herramienta sencilla y eficiente, que proporciona una interfaz fácil de acuerdo a las características de Subversion. Su eficiencia radica en su sencillez de uso para los principiantes, pero lo suficientemente flexible como para aumentar la productividad para los usuarios experimentados de Subversion. RapidSVN se caracteriza además por ser portable, pues funciona en cualquier plataforma en la que Subversion y wxWidgets se ejecuten: Linux, Windows, Mac OS 9 / X, Solaris, etc. La última versión de ésta herramienta es RapidSVN 1.5.5.

1.4.2 Herramienta para administración de repositorios

1.4.2.1 VisualSVN

VisualSVN un complemento de Visual Studio que ofrece una excelente integración entre Visual Studio y Subversion. Esta herramienta elimina prácticamente la gestión de los archivos de código fuente el

proyecto dentro de Subversion. Al usarlo se mostrarán marcadores fáciles de comprender junto a cada uno de los archivos del Explorador de soluciones que ofrece Visual Studio. Estos indicarán cuando un archivo se modifica, no se modifica o tiene conflicto; además de permitir mostrar los cambios, revertirlos, actualizarlos o aplicarlos.

Mediante el menú de VisualSVN, se pueden administrar los cambios para todo el proyecto. Este menú también contiene accesos directos a funciones comunes de TortoiseSVN como, por ejemplo, el explorador repo, la creación y aplicación de parches, la visualización de un registro de Subversion, así como ramificación, combinación e intercambio de repositorios. Al realizar una integración lo más rápido posible, descubrirá los problemas potenciales de combinación, y si implementa la integración continua, se generará todo el código y se ejecutarán las pruebas. VisualSVN facilita la protección temprana y frecuente gracias a la estrecha integración con Visual Studio: siempre recibirá recordatorios que tiene cambios modificados que aún no ha protegido. No hay motivo por aplazar la aplicación ya que puede proteger esos cambios directamente desde Visual Studio. [5]

Ventajas:

- ✓ Facilidad de uso. VisualSVN hace mucho más fácil de usar Subversion dentro de Visual Studio, que utilizar TortoiseSVN a solas con Subversion.
- ✓ Profunda integración de proyectos y soluciones de Visual Studio con la gestión de archivos Subversion.
- ✓ Ventanilla única de control de código fuente que permite realizar todas las funciones de nivel de usuario de Subversion: gestionar los archivos de proyecto, checkout, commit, y otras operaciones, dentro de Visual Studio.
- ✓ Compatible con las últimas versiones de Subversion y TortoiseSVN. Es actualizado continuamente.
- ✓ Fácil de configurar.
- ✓ La barra de herramientas de VisualSVN proporciona el acceso directo a los comandos y características más importantes de Subversion. Proporciona incluso la URL de la rama de Subversion en que se trabaja y puede usarse para cambiar rápidamente entre ramas.

1.4.3 Clientes web para interactuar con un sistema de control de versiones Subversion

1.4.3.1 WebSVN: Navegador de repositorios Subversion en línea

WebSVN ofrece una vista de sus repositorios Subversion, ha sido diseñado para reflejar la metodología de SVN. Puede ver el registro de cualquier archivo o directorio y ver una lista de todos los archivos cambiados, añadidos o eliminados en cualquier revisión dada. También puede ver las diferencias entre dos versiones de un archivo para ver exactamente qué se ha cambiado en una revisión.

Desde su escrito usando PHP, WebSVN es también muy portable y fácil de instalar. Ésta herramienta web es liberada bajo la licencia GNU (General Public License).

WebSVN ofrece las siguientes características: [6]

- ✓ Interfaz fácil de usar.
- ✓ Sistema de plantillas personalizables.
- ✓ Archivo coloreado de anuncios.
- ✓ Autoría de vista.
- ✓ Mensaje de registro de la búsqueda.
- ✓ RSS de apoyo.

1.4.3.2 WebClient para SVN: Herramienta Libre de Subversion

Si se utiliza Subversion (o se planea usarlo), puede que desee ofrecer a sus usuarios la opción de trabajar con Subversion en la web. WebClient para SVN, es una de las herramientas Subversion libres de Polarion Software, su uso se basa en la web, este cliente permite a los usuarios trabajar con repositorios SVN utilizando un navegador web.

Cuando es instalado en el servidor de Subversion, WebClient proporciona una manera conveniente para navegar por un repositorio de Subversion (su contenido e historia). Además permite realizar operaciones de escritura simple. WebClient se puede conectar a cualquier repositorio de Subversion que proporcione el acceso HTTP (es soportado por Subversion 1.1 y superior).

WebClient para SVN está basado en la web, proporcionando un gran rendimiento y usabilidad. Sus desarrolladores han desarrollado a WebClient para SVN con el objetivo de brindar un mayor rendimiento,

estabilidad y usabilidad, ya que es el mismo cliente que se utiliza para la gestión en el repositorio de Subversion basado en soluciones ALM (Application lifecycle management).

Entre las características de WebClient se incluyen: [7]

- ✓ Navegador para las carpetas y archivos del repositorio Subversion.
- ✓ Navegar y comparar revisiones de carpetas y archivos SVN.
- ✓ Ver detalles de la revisión y comparar revisiones SVN.
- ✓ Permite las operaciones crear/ borrar/ modificar archivos en un repositorio Subversion.
- ✓ Permite Crear/ Suprimir las carpetas del repositorio SVN.
- ✓ Ofrece fácil navegación por ramas/ etiquetas SVN.
- ✓ 100% puro Java, la implementación de la capa de acceso a SVN (utilizando la biblioteca JavaSVN).

WebClient para SVN es una plataforma independiente de solución basada en Java, implementada en JSP/ Java. Para usarlo, se necesita:

- ✓ Del lado del servidor: Java J2RE/J2SDK 1.4, Tomcat 4.
- ✓ Del lado del cliente: Cualquier navegador Web.

1.5 Aplicaciones Web

Una aplicación web no es más que una aplicación informática que los usuarios utilizan para acceder a un servidor web a través de la Internet o de una intranet usando un navegador web que la ejecuta y soporta el lenguaje en que está codificada (HTML, JavaScript, Java, y otros). (Ver **Anexo V**) Estas han alcanzado gran popularidad, pues el navegador web resulta muy práctico, además de la facilidad que brinda para actualizar y mantener aplicaciones web sin la necesidad de instalar y distribuir software a numerosos usuarios.

Las aplicaciones web ofrecen numerosas ventajas:

- ✓ Compatibilidad multiplataforma. Ofrecen un camino mucho más sencillo para la compatibilidad multiplataforma que las aplicaciones de software descargables.
- ✓ Actualización. Las aplicaciones basadas en web están siempre actualizadas con el último lanzamiento sin requerir que el usuario tome acciones pro-activas.

- ✓ Inmediatez de acceso. No necesitan ser descargadas, instaladas y configuradas.
- ✓ Facilidad de prueba. Con ellas no existen obstáculos para permitir pruebas sencillas y efectivas de herramientas y aplicaciones
- ✓ Menos requerimientos de memoria. Tienen muchas más razonables demandas de memoria RAM de parte del usuario final que los programas instalados localmente.
- ✓ Precio. Las aplicaciones basadas en web no requieren la infraestructura de distribución, soporte técnico y marketing requerido por el software descargable tradicional.
- ✓ Múltiples usuarios concurrentes. Las aplicaciones basadas en web pueden realmente ser utilizadas por múltiples usuarios al mismo tiempo.
- ✓ Se facilita el trabajo a distancia.
- ✓ Las aplicaciones Web son fáciles de usar (no requieren conocimientos avanzados de computación).
- ✓ Otorga la flexibilidad de determinar niveles de acceso según la confidencialidad de los datos así como la posibilidad de realizar transacciones on-line.

1.6 Servidor Web Apache

El Servidor Web Apache es un software de código abierto (libre) que funciona sobre cualquier plataforma (Unix, Windows, Macintosh y otras). Apache es el servidor web hecho por excelencia, su configurabilidad, robustez y estabilidad hacen que cada vez millones de servidores reiteren su confianza en este programa. Presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido.

Apache es un servidor altamente configurable de diseño modular. Es muy sencillo ampliar las capacidades de este servidor. Trabaja con gran cantidad de Perl, PHP y otros lenguajes de script. Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor, pues puede ser configurado para que ejecute un determinado script cuando ocurra un error en concreto. Además, tiene una alta configurabilidad en la creación y gestión de logs. Apache permite la creación de ficheros de log a medida del administrador, de este modo puedes tener un mayor control sobre lo que sucede en tu servidor. Servidor HTTP Apache 2.2.11 es la mejor versión disponible y la más reciente.

1.7 Navegadores

Un navegador o navegador Web es el programa que permite visualizar los contenidos de las páginas Web en Internet, o sea, es una aplicación software que posibilita recuperar y visualizar documentos de hipertexto desde servidores web de todo el mundo (Ver **Anexo VI**). Se conoce como World Wide Web (WWW) o con el nombre de browser. Ejemplos de estos son: Internet Explorer, Netscape Navigator, Opera, Mozilla Firefox y otros.

El funcionamiento de un navegador web se basa en permitir la visualización de documentos de texto. Estos documentos pueden encontrarse en la computadora usada por el usuario o en otro dispositivo conectado a dicha computadora o a través de Internet, además debe tener los recursos necesarios para la transmisión de los documentos usando un software que sirva como servidor web.

Ventajas del uso de navegadores web o browsers:

- ✓ En la actualidad son un tipo de aplicación de uso masivo que brinda a los usuarios una interfaz conocida y fácil de usar.
- ✓ El lenguaje HTML utilizado para el desarrollo de páginas web proporciona aspectos adecuados para el desarrollo de libros como por ejemplo hipervínculos, imágenes, estilos de texto y otros. Además, este lenguaje cuenta con una gran cantidad de aplicaciones orientadas al desarrollo de páginas HTML.

1.8 Modelo cliente-servidor

En el desarrollo de la herramienta se hace uso de una arquitectura en capas, sustentada en el modelo cliente-servidor. El esquema cliente-servidor "es un modelo de computación en el que el procesamiento requerido para ejecutar una aplicación o conjunto de aplicaciones relacionadas se divide entre dos o más procesos que cooperan entre sí". [8]

Sustentados en este modelo se desarrollan sistemas de información donde las transacciones son divididas en procesos independientes que intercambian información, servicios o recursos. El cliente no es más que el proceso que inicia el diálogo o solicita los recursos y el servidor es el proceso que responde a

dichas solicitudes. De esta manera el servidor es el contenedor de la parte que debe ser compartida por el conjunto de usuarios, y el cliente sólo posee lo particular de cada usuario. (Ver **Anexo VII**)

Características principales de la arquitectura cliente/servidor: [9]

- ✓ El servidor presenta a todos sus clientes una interfaz única y bien definida.
- ✓ El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- ✓ El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- ✓ Los cambios en el servidor implican pocos o ningún cambio en el cliente.

Ventajas del esquema Cliente/Servidor:

- ✓ Brinda la posibilidad de utilizar máquinas considerablemente más baratas que las requeridas por una solución centralizada, basada en sistemas grandes.
- ✓ Con este modelo se pueden utilizar componentes, tanto de hardware como de software, de varios fabricantes, lo cual contribuye considerablemente a la reducción de costos y favorece la flexibilidad en la implantación y actualización de soluciones.
- ✓ Al favorecer el uso de interfaces gráficas interactivas, los sistemas construidos basados en este esquema poseen una interacción más intuitiva con el usuario.
- ✓ Al utilizar interfaces gráficas para interactuar con el usuario, el esquema Cliente/ Servidor, permite que no sea siempre necesario transmitir información gráfica por la red pues esta puede residir en el cliente, lo cual permite aprovechar mejor el ancho de banda de la red.
- ✓ El esquema Cliente/ Servidor es más rápido el mantenimiento y el desarrollo de aplicaciones pues se pueden emplear las herramientas existentes.

1.9 Selección de tecnologías, herramientas y metodologías a utilizar

1.9.1 UML: lenguaje de modelado

UML o Lenguaje Unificado de Modelado (Unified Modeling Language, por sus siglas en inglés), es un lenguaje gráfico para visualizar, especificar, construir y documentar artefactos de un sistema que involucra gran cantidad de software. Es el lenguaje de modelado para sistemas de software que mayor popularidad ha alcanzado en la actualidad; y que se usa para especificar y no para describir métodos o procesos. Se emplea en el desarrollo de software con tecnología orientada a objetos.

Puede ser aplicada en el desarrollo de software a través de gran variedad de formas para dar soporte a una metodología de desarrollo de software pero no especifica una metodología o proceso específico para usar. UML cuenta con varios tipos de diagramas usados para definir sistemas de software y hardware, detallar artefactos en los sistemas y documentar la programación.

Este lenguaje garantiza la calidad de la ingeniería de software, con la participación de usuarios satisfacción de requerimientos, adaptabilidad a cambios, reutilización, etc. Además es independiente del lenguaje o del proceso de implementación o desarrollo. UML es un lenguaje conocido, estándar y fácil de aprender. Ya se encuentra lista la versión UML 2.1.1 que consiste en infraestructuras y superestructuras.

1.9.2 Metodologías de Desarrollo

Las metodologías de desarrollo de software constituyen un conjunto de procedimientos, técnicas, herramientas y documentación empleados en el desarrollo de productos software. Si no se usa una metodología de desarrollo, el proceso de desarrollo de un software se torna difícil de controlar.

El uso de las metodologías, impone un proceso disciplinado en el desarrollo de software, con el objetivo o finalidad de hacerlo eficiente y predecible. Para ello desarrollan un proceso detallado y específico, haciendo énfasis en la planificación basada en otras disciplinas ingenieriles.

1.9.2.1 RUP

RUP: Proceso Unificado de Rational (Rational Unified Process por sus siglas en inglés) es un proceso de desarrollo de software, que constituye la metodología más usada para el análisis, implementación y

documentación de sistemas con tecnología orientada a objetos. No es una metodología que sigue un proceso estático, sino que es flexible, adaptable al contexto y necesidades de cada organización. Además proporciona prácticas probadas y una arquitectura configurable.

Se caracteriza por ser iterativo e incremental, centrado en la arquitectura y guiado por casos de uso. El desarrollo iterativo e incremental es una de las mejores prácticas que posee RUP. Éste organiza los proyectos en términos de disciplinas y fases, compuesta cada una en varias iteraciones.

Esta metodología establece una forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo). Facilita la administración de requisitos, el control de cambios y la verificación de la calidad de software. Además incluye modelado visual del software.

1.9.3 Lenguajes de Programación Web

Actualmente la mayor parte de las páginas web que se encuentran en Internet se sustentan sobre dos pilares importantes: las bases de datos y diversos lenguajes de desarrollo web. Los lenguajes de desarrollo web permiten realizar sitios web dinámicos y generan las páginas según las peticiones de los usuarios. A partir del uso de estos lenguajes de programación web, los creadores pueden actualizar con facilidad los contenidos del sitio y automatizar tareas, que antes se hacían eternas, usando sólo el lenguaje HTML. Estos lenguajes de programación funcionan en el servidor, por lo que los usuarios los usan inconscientemente, pues sólo ven el resultado final: la página con el código HTML.

Entre los lenguajes de programación web figuran algunos como PHP, ASP.NET, Java, JSP, PERL y otros.

1.9.3.1 PHP

PHP (PHP Hypertext Pre-processor) es un lenguaje de programación potentísimo, que está diseñado esencialmente para desarrollo web. Es un lenguaje script que ofrece una solución simple, universal y de fácil programación para el desarrollo de páginas web dinámicas del lado del servidor. Puede estar embebido en el código HTML y es de Open Source (código abierto), las que constituyen algunas de las causas por las que su uso se ha popularizado y extendido en la web.

Ventajas del lenguaje PHP:

- ✓ Es un lenguaje multiplataforma, es decir, puede ser ejecutado sobre la mayoría de los sistemas operativos: Windows, Mac OS, Linux, Unix, y otros.
- ✓ Permite la conexión a varios tipos de servidores de bases de datos, entre las que se encuentran MySQL, PostgreSQL, Oracle, MS SQL Server, Sybase mSQL, Informix, ODBC, etc.
- ✓ Es independiente del navegador, pues como se ejecuta en el servidor no necesita que éste lo soporte.
- ✓ Se caracteriza por su rapidez. El hecho de ser usado, generalmente, como módulo de Apache, lo hace extremadamente veloz.
- ✓ Está completamente escrito en C, así que se ejecuta rápidamente utilizando poca memoria.
- ✓ Su intérprete, puede ser configurado como módulo en servidor Web o ejecutado como un binario CGI, permite el acceso a ficheros, la ejecución de comandos y abrir comunicaciones de red en el servidor.
- ✓ Logra la integración con varias bibliotecas externas, lo que permite desde generar documentos en PDF (Acrobat Reader) hasta analizar código XML.

Además, es capaz de realizar acciones determinadas, fácil y eficazmente, debido a que ofrece un amplio conjunto de funciones usadas en la explotación de bases de datos, sin complicaciones; por esto es instalado en el servidor y funciona con versiones de Apache, Microsoft IIS, Netscape Enterprise Server y otros. Es un lenguaje que guarda gran similitud con otros como C y Perl, lo que constituye una ventaja, pues permite a la generalidad de los programadores crear aplicaciones complejas, empleando un tiempo de aprendizaje corto.

El diseño de PHP fue realizado específicamente para lograr un lenguaje seguro para escribir programas, que combina la correcta selección de opciones de configuración de tiempo de compilación y ejecución, con lo que se obtiene la combinación perfecta de libertad y seguridad necesaria. Al ser un lenguaje libre, es soportado por una gran comunidad de desarrolladores, por lo que goza de la ayuda de muchos programadores, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente. Esta es una de las razones por las que el código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar sus capacidades. La última versión es PHP 5.2.8 publicada el 8 de diciembre de 2008.

1.9.3.2 JavaScript

JavaScript es un lenguaje de programación cuyo uso se ha generalizado a nivel mundial en el desarrollo web, debido a su versatilidad y potencia en la realización de tareas pequeñas así como en la gestión de complejas aplicaciones. JavaScript es un lenguaje de programación interpretado, con una sintaxis semejante a lenguajes como Java y C, es Open Source (código abierto) y su uso principal radica en el desarrollo de páginas web. La versión más reciente es JavaScript 1.8 emitida en el 2008.

Todos los navegadores modernos son capaces de interpretar código JavaScript embebido en las páginas web. Puede ser incluido en cualquier documento HTML o todo aquel que termine traducándose en HTML en el navegador del cliente, ya sea PHP, ASP, JSP, SVG, etc., usándose para realizar tareas y operaciones en el marco de la aplicación cliente.

Este lenguaje posee una característica que lo hace especialmente ideal para el trabajo en la Web, pues son los navegadores, que se usan para viajar en ella, los que interpretan y ejecutan los programas escritos usando JavaScript. Es así que se pueden enviar documentos, a través de la Web, que contienen código fuente de programas, por lo que dejan de ser simples fuentes de información estática, para convertirse en documentos dinámicos.

En JavaScript se identifican dos características principales: es un lenguaje basado en objetos; programación dirigida a objetos, pero con menos restricciones, y es además un lenguaje orientado a eventos, debido al tipo de entornos en los que es utilizado: Windows y sistemas X-Windows. Los programas escritos en este lenguaje son independientes de la plataforma hardware o sistema operativo, siempre y cuando exista un navegador que lo soporte para la plataforma de trabajo.

Los scripts se pueden desarrollar en un período de tiempo relativamente pequeño, por lo que el lenguaje no requiere un tiempo de compilación. Otras características son las relativas a la interfaz: cuadro de diálogo, formularios y otros elementos GUI (Interfaz Gráfico de Usuario), son gestionados por el navegador y por el código HTML. Es por ello que los programadores que usan el lenguaje JavaScript no deben preocuparse por la creación o control de los mencionados elementos en sus aplicaciones.

1.9.3.3 Bash

Bash es un lenguaje interpretado de programación, cuyo fin es ayudar al administrador a ejecutar la mayor cantidad de las tareas necesarias en la automatización y arranque del sistema. Es usado especialmente en sistemas Unix, aunque no es exclusivo de éste. En síntesis es un shell de Unix, o sea, un intérprete de órdenes de Unix, escrito para el proyecto GNU. Es el intérprete predeterminado para la mayoría de los sistemas GNU/Linux, incluyendo Mac OS X Tigre, y se ejecuta sobre la mayoría de los sistemas operativos tipo Unix, e inclusive en la actualidad se ha llevado a Microsoft Windows por el proyecto Cygwin. Bash 3.2.48, emitida el 18 de noviembre del 2008, es la última versión de éste lenguaje.

Las aplicaciones principales del lenguaje Bash son: [10]

- ✓ El prompt se puede modificar al gusto del usuario.
- ✓ Crear scripts que faciliten el uso diario de la máquina Linux.
- ✓ Ser una magnífica manera de acercarse a la programación y entender los fundamentos de esta.

Las mejoras ofrecidas por BASH incluyen: [11]

- ✓ Línea de comandos de edición.
- ✓ Tamaño ilimitado historial de comandos.
- ✓ Control de trabajos.
- ✓ Funciones y Shell Alias.
- ✓ Indexados arreglos de tamaño ilimitado.
- ✓ De la aritmética entera en cualquier base de dos a sesenta y cuatro.

1.9.4 Herramientas

1.9.4.1 Visual Paradigm

Visual Paradigm es una herramienta CASE (Computer-Aided Software Engineering), que se destaca por su robustez, usabilidad y portabilidad. Es una herramienta UML profesional, capaz de soportar el ciclo de vida completo del desarrollo de un software (análisis y diseño orientados a objetos, construcción, pruebas

y despliegue). Usar este software en el modelado UML contribuye a la construcción de aplicaciones de calidad, mejores, con un menor coste y en menos tiempo.

Ésta herramienta permite realizar tanto ingeniería directa, como ingeniería inversa, ya que partiendo de un modelo relacional en SQL Server, MySql u otros, es capaz de desplegar todas las clases asociadas a las tablas. Para la gestión y mapeo de estas clases con la base de datos, usa Hibernate para Java y NHibernate en caso de los proyectos .Net. Es una herramienta colaborativa, pues soporta múltiples usuarios trabajando sobre el mismo proyecto. Además permite el control de versiones y es capaz de generar la documentación del proyecto automáticamente en diferentes formatos: Web, Pdf, y otros. Visual Paradigm, permite además, dibujar todos los tipos de diagramas de clases, código inverso y generar código desde diagramas.

Visual Paradigm para UML actualmente soporta una interfaz en cuatro idiomas. Inglés, Chino Simplificado, Chino Tradicional y Japonés. Además soporta la importación desde el Rational Rose, permite la exportación/ importación XML, y la integración IDE con Visual Studio, IntelliJ IDEA, Eclipse, NetBeans y otros. Las últimas versiones conocidas son Visual Paradigm for UML (SE) [Windows] 6.3 y Visual Paradigm for UML (VP-UML) 6.0 Service Pack 1

Algunas de las principales ventajas que ofrece Visual Paradigm son:

- ✓ Entorno de creación de diagramas para UML 2.1.
- ✓ Diseño centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad.
- ✓ Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- ✓ Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- ✓ Disponibilidad de múltiples versiones, para cada necesidad.
- ✓ Disponibilidad en múltiples plataformas.

1.9.4.2 Rational Rose

Rational Rose Enterprise es el producto más completo de la familia Rational Rose. Proporciona un lenguaje común de modelado para el equipo que facilita la creación de software de calidad más rápidamente. Está orientado a UML, y es la herramienta de diseño de software destinada al modelado visual y el componente de construcción de la empresa a nivel de aplicaciones de software.

Dos características esenciales de Rational Rose son su capacidad para proporcionar el desarrollo iterativo y la ingeniería directa e inversa. Permite a los diseñadores tomar ventaja del desarrollo iterativo, ya que la nueva aplicación se puede crear por etapas con la salida de una iteración de convertirse en la entrada a la siguiente. La última versión conocida es Rational Rose 7.0.0 Enterprise Edition.

Algunas características de de esta herramienta son:

- ✓ La generación de código Ada, ANSI C ++, C++, CORBA, Java y Visual Basic, con capacidad de sincronización modelo- código configurables.
- ✓ Característica de control por separado de componentes, modelo que permite una administración más granular y el uso de modelos.
- ✓ Capacidad de análisis de calidad de código.
- ✓ Modelado UML para trabajar en diseños de base de datos, con capacidad de representar la integración de los datos y los requerimientos de aplicación a través de diseños lógicos y físicos.
- ✓ Integración con otras herramientas de desarrollo de Rational.
- ✓ Capacidad para integrarse con cualquier sistema de control de versiones SCC-compliant, incluyendo a Rational ClearCase.

1.9.4.3 Macromedia Dreamweaver 8

Macromedia Dreamweaver 8 es la herramienta por excelencia para el desarrollo de páginas web. Actualmente ha alcanzado un nivel de popularidad que la convierte en el número uno entre las herramientas de su tipo en el mercado, debido a sus funcionalidades e integración con otras herramientas como Adobe Flash. Permite diseñar, crear o mantener de forma eficaz sitios y aplicaciones Web. Cuenta con un sistema de ayuda que garantiza su uso fácil y eficiente.

Constituye la opción profesional para crear sitios Web y aplicaciones. Macromedia Dreamweaver 8 combina herramientas visuales de diseño, funciones de desarrollo de aplicaciones y soporte para la edición del código, facilidades que permiten a los desarrolladores y diseñadores profesionales e incluso a los menos expertos crear rápidamente sitios Web y aplicaciones basados en estándares. Presenta soporte

de diseño en CSS y funciones de codificación manual, proporcionando las herramientas profesionales para el desarrollo en un entorno integrado y agilizado. Incluye, además, un editor de gráficos incorporado, soporte único para el desarrollo con ColdFusion, ASP.NET, JSP, PHP, XML y servicios web. Macromedia Dreamweaver MX 2004 es la última versión conocida de ésta herramienta.

1.9.4.4 Zend Studio

Zend Studio constituye el entorno integrado de desarrollo (IDE) que abarca todos los componentes necesarios para desarrollo mediante el lenguaje de programación PHP. Está escrito en Java. Es multiplataforma, pues está disponible para Microsoft Windows, Mac OS X y GNU/Linux.

Una de las características de Zend Studio es que no requiere la instalación previa del lenguaje PHP ni del entorno de ejecución de Java. Permite además el resaltado de sintaxis, autocompletado y ayuda de código y lista de parámetros de funciones y métodos de clase. Ofrece inserción automática de paréntesis y corchetes de cierre y detección de errores en tiempo real. Zend Studio permite, además, la instalación de barras de herramientas para Internet Explorer y Mozilla Firefox. Brinda soporte para gestión de grandes proyectos de desarrollo y para el control de versiones usando CVS o Subversion. Tiene integrado un Manual de PHP y un Cliente FTP. Soporta navegación en base de datos y ejecución de consulta SQL. Aunque Zend Studio fue diseñado para usarse con el lenguaje PHP; ofrece soporte básico para lenguajes Web como JavaScript, XML y HTML. La última versión conocida es Zend Studio 6.0.1.

Conclusiones

El desarrollo de este capítulo permitió realizar el estado del arte de algunas de las herramientas que existen a nivel mundial con características y funcionalidades similares a la que se pretende implementar con el desarrollo de éste trabajo. La investigación realizada permite concluir que las herramientas web estudiadas no poseen las características de ser web, multiplataforma, libres y que posibiliten administrar un repositorio que utilice un sistema de control de versiones Subversion. Por ésta razón se hace necesario diseñar e implementar una herramienta que cumpla con estas características. Además, en éste capítulo se presentaron las principales tecnologías, metodologías y herramientas a utilizar para el desarrollo de la herramienta PhpSvnAdmin.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

En este capítulo se realiza una primera propuesta de solución a la herramienta para la administración de repositorio Subversion (PhpSvnAdmin). Debido a que no se evidencia una visualización concernida de los procesos del negocio, se desarrolla un Modelo de Dominio.

2.1 Modelo de Dominio

El Modelo de Dominio es la representación de conceptos de importancia, constituido por reglas de modelado, no por componentes propios de un sistema de software. Puede ser tomado como punto de partida para el desarrollo de la herramienta para la administración de repositorio Subversion (PhpSvnAdmin), considerándose un diccionario visual de conceptos del dominio.

Es posible realizar un elevado grado de detalles en un Modelo de Dominio, ya que tiene como objetivo fundamental capturar lo necesario para comprender el funcionamiento del sistema, representado a través de clases conceptuales en forma de un Diagrama de clases.

2.1.1 Conceptos fundamentales

Para lograr una buena asimilación del Diagrama del Modelo de Dominio se proporciona un mapa conceptual con las definiciones indicadas en el proceso de desarrollo de la herramienta para la administración de repositorio Subversion (PhpSvnAdmin):

- ✓ **Usuarios:** Actores que interactúan directamente con la herramienta PhpSvnAdmin, con el objetivo de consultar, modificar o eliminar la información gestionada por el esta.
- ✓ **Fichero:** Archivo para almacenar datos persistentes en el tiempo.
- ✓ **Bitácora:** Registro de movimientos realizados por el usuario al interactuar con el sistema, también conocido como trazas.
- ✓ **Historial de bitácora:** Almacena detallada y ordenadamente las bitácoras dejadas por los usuarios.

- ✓ **Permisos:** Autoriza realizar la operación proporcionada por un fichero determinado.
- ✓ **Administrador:** Actor que tiene los permisos necesarios para adicionar y eliminar usuarios, así como gestionar los privilegios de estos. Además tiene permiso total de las acciones sobre el sistema.
- ✓ **Rol:** Es la función que desempeña un usuario y que delimita el conjunto de funcionalidades que puede ejecutar dentro del sistema.
- ✓ **Visualizador:** Actor que posee sólo permisos de lectura.
- ✓ **Identificador:** Conjunto de datos y privilegios que caracterizan a un usuario determinado que se crea de forma automática durante los procesos de autenticación y autorización, usando MD5. Este incluye un identificador único generado de manera aleatoria y un listado de los componentes a los que tiene derecho de acceso, así como los privilegios de ejecución correspondientes a esos componentes.

2.1.2 Modelo del Diagrama del Dominio

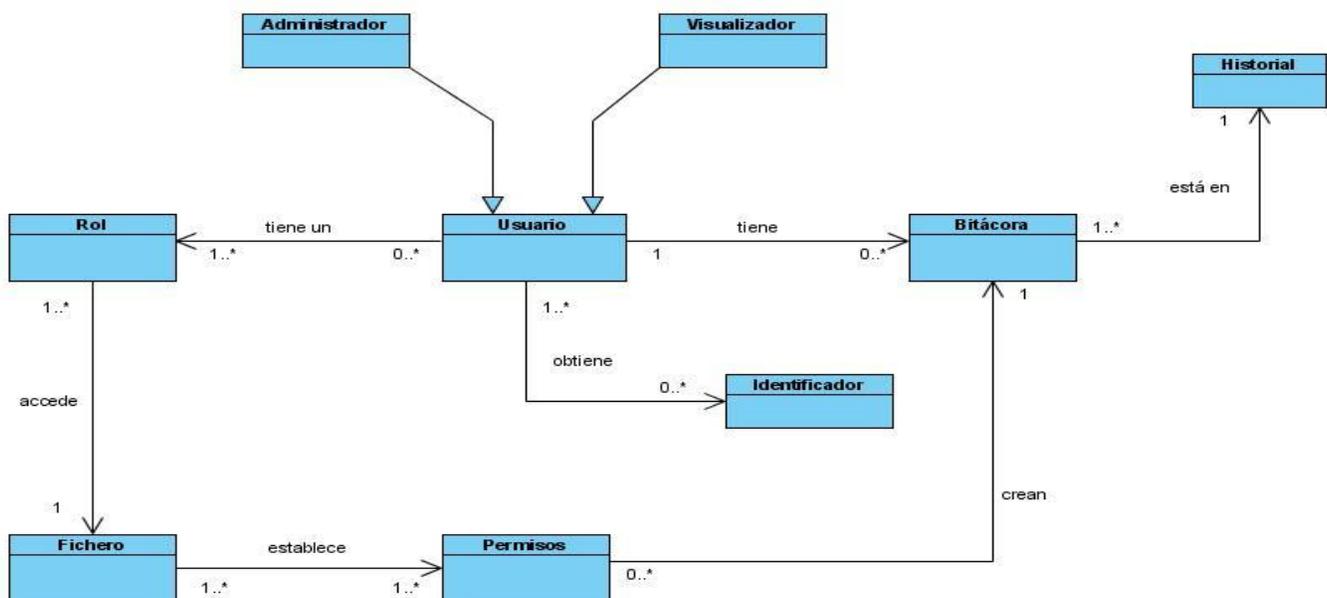


Figura 2.1 Diagrama del Modelo de Dominio.

2.2 Propuesta de Sistema

La creación de un sistema informático combina elementos como hardware, software y soporte humano. Inicialmente se comienza a construir el sistema teniendo en cuenta las características del entorno donde este será implantado, o sea, se crea una visión general que se refleja en el Modelo de Dominio, lo que permite analizar los conceptos definidos y sus relaciones, para finalmente determinar todos los requerimientos que debe cumplir el sistema. De esta forma queda conformado el Modelo del Sistema.

2.2.1 Especificación de Requerimientos de Software

La Standard Glossary of Software Engineering Terminology (IEEE) define un requerimiento como la condición o capacidad que debe poseer un usuario para resolver un problema o alcanzar un objetivo. Esta condición o capacidad tiene que evidenciarse en un sistema o en el componente de un sistema de forma que satisfaga un contrato, estándar, u otro documento establecido formalmente. De manera general todas las ideas que sean manejadas por los usuarios, miembros del equipo de desarrollo del proyecto y cliente constituyen candidatas a convertirse en requerimientos, estos se pueden clasificar en funcionales y no funcionales.

Realizar una correcta especificación de requerimientos de software garantizará la elaboración de un software de alta calidad, por lo que es de singular importancia, realizar este proceso de manera precisa y eficiente. El éxito de la ejecución de esta tarea garantiza que se minimicen los problemas que se derivan del desarrollo del sistema, pues permite la definición exacta y clara del comportamiento del mismo, así como de lo que se desea producir.

2.2.1.1 Requerimientos Funcionales

Los requerimientos funcionales constituyen condiciones o capacidades que el sistema debe cumplir, y que definen el comportamiento interno de un software. Estos requerimientos no alteran la funcionalidad del producto, o sea los requerimientos funcionales mantienen un comportamiento estable y constante sin importarle con que propiedades o cualidades se relacionen.

Requerimientos funcionales (RF) determinados

El sistema debe:

RF 1 Autenticar usuario.

RF 2 Buscar repositorios.

RF 3 Buscar archivos de configuración.

RF 4 Mostrar perfiles.

RF 5 Gestionar usuario.

RF 5.1 Agregar un nuevo usuario al repositorio.

RF 5.2 Modificar usuarios del repositorio.

RF 5.3 Eliminar usuario del repositorio.

RF 6 Gestionar grupos de usuarios.

RF 6.1 Agregar grupos de usuarios.

RF 6.2 Modificar grupo de usuarios.

RF 6.3 Eliminar grupos de usuarios.

RF 7 Gestionar permisos a usuarios.

RF 7.1 Agregar permisos a usuarios.

RF 7.2 Modificar permisos a usuarios.

RF 7.3 Eliminar permisos a usuarios.

RF 8 Gestionar permisos a usuarios.

RF 8.1 Agregar permisos a grupo de usuarios.

RF 8.2 Modificar permisos a grupo de usuarios.

RF 8.3 Eliminar permisos a grupo de usuarios.

2.2.1.2 Requerimientos No Funcionales

Los requerimientos no funcionales son cualidades o propiedades que el producto debe tener. No son más que restricciones impuestas al producto que está siendo desarrollado. Estos requerimientos no se dedican a describir la funcionalidad del software, sino como la ejecutará, de ahí su importancia, ya que permite al cliente evaluar y valorar las características no funcionales del producto como usabilidad, seguridad, rapidez, confiabilidad y ser atractivo, funcionalidades que pueden establecer la diferencia entre un producto bien aceptado y uno con poca aceptación. La correcta definición de los requisitos no funcionales es de gran importancia, un error en este proceso es difícil y caro de resolver y en ocasiones estos requerimientos son fundamentales para alcanzar un producto exitoso.

Requerimientos no funcionales (RNF) determinados

Requerimientos de Software

Especifican el software del cual el sistema debe disponer para un funcionamiento correcto. Debe indicarse además, si es necesaria la integración del producto de software con otros o si necesita que sean asociados otros programas al mismo.

RNF 1

Los clientes (usuarios) tendrán acceso a la herramienta web para la administración de repositorios Subversion **PhpSvnAdmin** a través de navegadores web como: Internet Explorer 5.0 o superior, Mozilla Firefox 1.5 y Opera.

RNF 2

PhpSvnAdmin se ejecuta sin ningún tipo de problema en cualquier ordenador que disponga de Windows (2000, XP, o Vista) o Linux.

RNF 3

El hardware recomendado deberá incluir un Pentium o una unidad CPU equivalente, 256 MB de RAM, 1GB mínimo de espacio de disco duro.

RNF 4

Se necesita de un servidor web instalado, se recomienda el Apache, configurado con los módulos svn y ldap.

RNF 5

Se recomienda tener instalado Subversion en su versión 1.3.3 para Windows y 1.3.0 Linux.

Requerimientos de Hardware

Especifican los elementos de hardware de los cuales se debe disponer, o sea, la infraestructura tecnológica de hardware necesaria (Servers, terminales, etc.).

RNF 6

En las terminales se necesita de CPU: (Pentium Recomendado) aunque bien se puede usar 486 con suficiente memoria. Equipo PC con CPU de 1 GHz, 20 Gbytes de disco duro, 256 Mbytes de RAM y tarjeta gráfica SVGA con resolución 1024x768 píxeles.

RNF 7

La capacidad de memoria debe ser de 128 Mb, (16 recomendado)

Requerimientos de apariencia o interfaz externa

Este requerimiento se refiere a la apariencia del producto, especificando cómo se desea que sea la interfaz externa del producto. También se refieren a necesidades de cumplir con normas estándares (por ejemplo las interfaces de sistemas tipo Windows/Apple/Motif)

RNF 8

La mayoría de la funciones serán realizadas desde la página principal y luego de completadas las tareas se regresa a ella.

RNF 9

Realizar el diseño de una interfaz donde los usuarios o clientes de la herramienta, puedan seleccionar y organizar la información que desean visualizar a través de pocos pasos.

RNF 10

La interfaz externa de la herramienta debe ser legible, simple de usar, interactiva y debe hacer uso de colores adecuados.

RNF 11

Se desea que la página muestre pestañas que brinden información como: "Información General", "Datos de autores" y "Ayuda".

RNF 12

Los usuarios serán previamente creados por el administrador del dominio.

RNF 13

Brinda al usuario un enlace para hacer login y logout desde donde se encuentre.

RNF 14

Los usuarios registrados disponen de un historial.

Requerimientos de Seguridad

Es un tipo de requerimiento no funcional que puede provocar los mayores riesgos si no se maneja de manera correcta. Para garantizar la seguridad de un sistema se debe tener en cuenta, además de su propia seguridad, el ambiente en que se ejecutará dicho sistema; o sea, la seguridad física del lugar donde la aplicación es usada, los controles administrativos de acceso al sistema establecidos y las regulaciones legales que rigen el uso del sistema en cuestión. Los requerimientos de seguridad manejan tres aspectos fundamentales sobre la información:

Confidencialidad (información protegida de accesos no autorizados y divulgación)

RNF 15

Se debe disponer de un mecanismo de seguridad que tenga su base en el modelo de Autenticación, Autorización y Auditoría (AAA).

RNF 16

El acceso al sistema será controlado con usuarios y contraseñas.

RNF 17

Ante un usuario autenticado que no se encuentre registrado, el sistema debe reportar un error de acceso.

RNF 18

A un usuario autenticado que se encuentre registrado se le autoriza su acceso y se crea un certificado.

RNF 19

El certificado debe constar de:

usuario:rol:clave

RNF 20

Serán registradas todas las acciones de usuario, ya sean autorizadas o no, guardando además el día, mes, año, hora, minuto, segundo en que este se registra y si le fue otorgada la autorización.

RNF 21

Solamente los usuarios con derechos de administrador podrán acceder a las funcionalidades administrativas, estos permisos le serán negados a usuarios normales.

RNF 22

Las contraseñas deben tener una longitud no menor de 6 caracteres.

Integridad (información protegida contra la corrupción y estados inconsistentes)

RNF 23

Será posible la creación de copias de respaldo, que garanticen que el sistema pueda restaurarse ante fallos críticos o casos de pérdida total de la información.

Disponibilidad (garantizar el acceso a la información de usuarios autorizados)

RNF 24

A los usuarios que se registren con éxito se les garantizará el acceso a las funcionalidades de la herramienta para la administración de repositorios Subversion.

RNF 25

El establecimiento de la seguridad en el sistema no provocará retraso o lentitud en la respuesta del sistema, es por ello que se debe optimizar el código, además de minimizar y reducir el tiempo de respuesta.

Requerimientos de Usabilidad

Describen los niveles apropiados de usabilidad en dependencia de los usuarios finales del producto. Para la especificación de estos requerimientos es necesario chequear las especificaciones en los perfiles de usuarios, teniendo en cuenta sus niveles de experiencia.

RNF 26

La herramienta para administración de repositorios Subversion **PhpSvnAdmin** será usada por estudiantes, profesores y trabajadores de cualquier facultad de la Universidad de las Ciencias Informáticas (UCI), así como por estudiantes de nuevo ingreso. Podrá ser usada por cualquier otra institución autorizada por la UCI.

RNF 27

Para el uso de la herramienta para administración de repositorios Subversion **PhpSvnAdmin** es necesario tener conocimientos básicos de informática y dominio en el trabajo con administración de Subversion.

RNF 28

A cada usuario autenticado solo se le otorgará acceso a la información que le corresponde según el rol: Administrador y Visualizador.

RNF 29

La interfaz del usuario deberá ser tan familiar como sea posible para los usuarios que han usado otras aplicaciones o herramientas web.

Requerimientos de Rendimiento

Se especifican los requerimientos relacionados con la carga que el sistema debe soportar; por ejemplo, número de usuarios simultáneamente conectados, cantidad de transacciones que se espera que el sistema soporte, etc.

RNF 30

El sistema debe ser capaz de soportar un tiempo de respuesta no mayor de 30 segundos.

Requerimientos de Soporte

Definen la totalidad de las acciones que deben ser tomadas tras el desarrollo del software y que están encaminadas a asistir a los clientes, además de garantizar el mejoramiento progresivo del sistema y su evolución en el tiempo. En la definición de estos requerimientos pueden incluirse: Extensibilidad, Adaptabilidad, Configuración, Instalación, Pruebas, Internacionalización y otras.

RNF 31

Los usuarios que trabajen con la herramienta deben tener la preparación mínima requerida, que incluya conocimientos en el manejo de administración de repositorios.

Requerimientos de Portabilidad

Estos requerimientos se especifican cuando existe la posibilidad de que el software pueda ser trasladado a otras plataformas u entornos de desarrollo. Con esto se garantiza un software más duradero.

RNF 32

Posibilitar que el sistema se ejecute sobre el sistema Operativo Windows 98 o superior o Linux.

Requerimientos de Ayuda y documentación en línea

Describen los requerimientos que permiten a los usuarios y clientes apropiarse de la documentación necesaria para la manipulación, uso y entendimiento del sistema.

RNF 33

La herramienta debe disponer de una ayuda en formato HTML que brinde instrucciones al usuario.

RNF 34

Debe existir una documentación del sistema, que esté documentada usando alguna metodología de desarrollo.

Requerimientos de restricciones de diseño e implementación

Especifican o restringen la codificación o construcción de una aplicación o sistema; y que deben ser cumplidas estrictamente. Por ejemplo: lenguajes de programación usados en la implementación, uso obligatorio de determinadas herramientas de desarrollo, restricciones en la arquitectura o el diseño y otras.

RNF 35

El proceso de captación de los datos debe ser validado con el objetivo de evitar entradas inadecuadas o incorrectas.

RNF 36

Serán usadas herramientas de desarrollo que aseguren que todo el ciclo de desarrollo del producto se realice con la calidad requerida.

RNF 37

La lógica de presentación está unida a la lógica de negocio, centrando su función en la interfaz de usuario, validaciones simples de los datos de entrada y lógica de negocio, y estas dos separadas de la capa de persistencia.

RNF 38

Para llevar a cabo el desarrollo de la aplicación se seguirá el estándar de codificación oficializado en el documento de arquitectura de la facultad 7.

2.3 Modelo del Sistema

2.3.1 Actores del sistema

Los actores del sistema se corresponden usualmente con los trabajadores del negocio y en ocasiones son también actores del mismo. Estos no son parte de él, pero sí intercambian información con el mismo y son recipientes pasivos de información. Pueden representar el rol que juega una o varias personas, un equipo e incluso un sistema automatizado. La definición de los actores del sistema constituye una actividad fundamental, pues un actor desempeña un papel por cada caso de uso con el que colabora, representando terceros fuera del sistema, de esta forma ya identificados los actores del sistema se identifica y define el entorno externo al mismo.

Actores del sistema	Descripción
Administrador	Tiene permiso total sobre las acciones del sistema: agregar y eliminar usuarios o grupos de usuarios, así como gestionar sus permisos.
Visualizador	Sus privilegios se limitan a visualizar información que brinda el sistema: usuarios y grupos de usuarios, así como permisos otorgados.

Tabla 2.1 Actores del Sistema.

2.3.2 Diagrama de Casos de Uso del Sistema

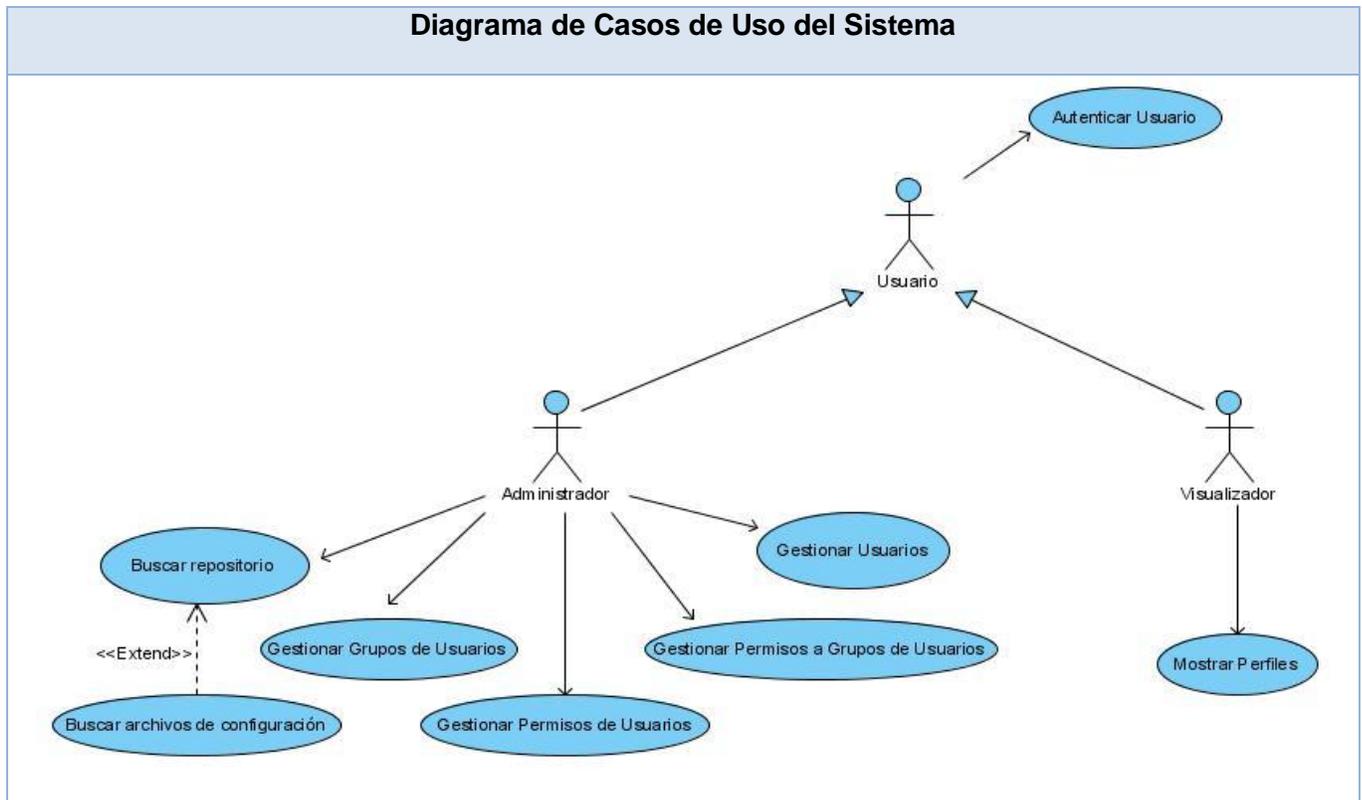


Tabla 2.2 Diagrama de Casos de Uso del Sistema.

2.3.3 Descripción textual de los CU del Sistema

CU 1	Autenticar Usuario
Actores:	Administrador (inicia), Visualizador (inicia).
Propósito:	Confirmar la validez de los usuarios y permitirle el acceso a las funcionalidades que le corresponden.
Resumen:	El caso de uso confirma la validez de los usuarios, redireccionándolos a las funcionalidades a que tengan acceso según su rol.

Referencias:	RF 1
Precondiciones:	El usuario debe estar registrado en el dominio UCI.
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
	1.1 El sistema muestra la página de autenticación.
2. Introduce usuario y contraseña y pulsa el botón "Aceptar".	2.1. El sistema verifica los permisos y muestra la página principal. <ul style="list-style-type: none"> • Si no tiene permisos ver flujo alterno 1.
Flujo alterno 1	
Acción del actor	Respuesta del sistema
	2.1 El sistema informa que no tiene permisos suficientes.
Puntos de extensión	
Poscondiciones:	El usuario se autenticó correctamente.
Prioridad:	Crítica
Interfaz:	

Tabla 2.3 Descripción del Caso de Uso Autenticar Usuario

CU 2	Buscar repositorios
Actores:	Administrador (inicia)
Propósito:	Buscar si existen repositorios en la PC donde trabaja.

Resumen:	El caso de uso inicia cuando el Administrador desea buscar repositorios.
Referencias:	RF 2
Precondiciones:	El usuario debe estar debidamente autenticado.
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
	<p>1.1 El sistema muestra un listado de los repositorios existentes.</p> <p>✓ Si no muestra ningún repositorio ver flujo alternativo 1.</p>
2. El Administrador selecciona el repositorio donde se encuentra el proyecto del que desea gestionar la administración.	2.1 El sistema muestra un listado de los proyectos existentes en el repositorio seleccionado.
3. El Administrador selecciona el proyecto deseado.	<p>3.1 El sistema realiza la búsqueda interna de los archivos de configuración.</p> <p>✓ Si no encuentra los archivos de configuración ver caso de uso extendido Buscar archivos de configuración.</p>
	3.2 El sistema muestra la interfaz gestionar usuario y activas las opciones del menú para gestionar la administración.
Flujo alternativo 1	
Acción del actor	Respuesta del sistema
	1.1 El sistema brinda la opción de realizar la búsqueda del repositorio Subversion manualmente.

2. El Administrador le suministra al sistema la dirección URL donde se encuentra el repositorio y el proyecto. Ejemplo: C://svnrepo/repo1/proy1.	2.1 El sistema realiza la búsqueda del repositorio indicado.
	2.2 El sistema realiza la búsqueda interna de los archivos de configuración en la dirección indicada. ✓ Si no encuentra los archivos de configuración ver caso de uso extendido Buscar archivos de configuración.
	2.3 El sistema muestra la interfaz gestionar usuario y activas las opciones del menú para gestionar la administración.
Puntos de extensión	
Poscondiciones:	El sistema muestra activas las opciones del menú para gestionar la administración.
Prioridad:	Crítico
Interfaz:	

Tabla 2.4 Descripción del Caso de Uso Buscar repositorios.

CU 3	Buscar archivos de configuración
Actores:	Administrador (inicia), Visualizador (inicia).
Propósito:	Buscar manualmente los archivos de configuración para gestionar la administración del repositorio.
Resumen:	El caso de uso permite a los usuarios buscar los archivos de configuración manualmente en caso de que el sistema no los haya encontrado.
Referencias:	RF 3

Precondiciones:	Debe haberse encontrado satisfactoriamente el repositorio deseado.	
Flujo normal de eventos		
Acción del actor	Respuesta del sistema	
	1.1 El sistema muestra un formulario con las opciones para buscar los archivos de configuración authz y path.	
2. E usuario realiza la búsqueda de los archivos de configuración proporcionándole al sistema la dirección URL donde se encuentra cada archivo.		
3. Presiona el botón "Enviar".	3.1 El sistema guarda la URL de cada archivo.	
	3.2 El sistema muestra la interfaz gestionar usuario y activas las opciones del menú para gestionar la administración.	
Flujo alterno 1		
Acción del actor	Respuesta del sistema	
Puntos de extensión		
Poscondiciones:	El sistema encuentra los archivos de configuración, muestra la información satisfactoriamente y activas las opciones del menú para gestionar la administración.	
Prioridad:	Crítica	
Interfaz:		

Tabla 2.5 Descripción del Caso de Uso Buscar archivos de configuración.

CU 4	Mostrar perfiles
-------------	------------------

Actores:	Visualizador(inicia)
Propósito:	Mostrar listado de usuarios, grupos de usuario y permisos existentes por proyecto.
Resumen:	El Caso de Uso inicia cuando el Visualizador desea mostrar el listado de usuarios, grupos de usuario y permisos existentes por proyecto.
Referencias:	RF 4
Precondiciones:	El Visualizador debe estar autenticado.
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
	1.1. El sistema muestra un listado de usuarios, grupo de usuarios a que pertenece, permisos y nivel de acceso en el repositorio brindando la posibilidad de buscar según el usuario.
2. El usuario busca información específica según los criterios de selección.	2.1 El sistema muestra la información deseada.
Flujo alterno 1	
Acción del actor	Respuesta del sistema
Puntos de extensión	
Poscondiciones:	El sistema muestra la información seleccionada por el usuario satisfactoriamente.
Prioridad:	Secundario
Interfaz:	

Tabla 2.6 Descripción del Caso de Uso Mostrar perfiles.

CU 5	Gestionar usuario	
Actores:	Administrador (Inicia)	
Propósito:	Gestionar los usuarios que tendrán acceso al repositorio de un proyecto dado.	
Resumen:	El caso de uso permite al administrador agregar, eliminar y modificar usuarios.	
Referencias:	RF 5, RF 5.1, RF 5.2, RF 5.3.	
Precondiciones:	Deben haberse encontrado satisfactoriamente los archivos de configuración.	
Flujo normal de eventos		
Acción del actor	Respuesta del sistema	
	<p>1.1 El sistema muestra un listado de los usuarios existentes y las opciones para Agregar, Eliminar y Modificar.</p> <p>a) Si desea agregar usuario ver sección "Agregar usuario".</p> <p>b) Si desea eliminar usuario ver sección "Eliminar usuario"</p> <p>c) Si desea modificar usuario ver sección "Modificar usuario".</p>	
Sección "Agregar usuario"		
Acción del actor	Respuesta del sistema	
	1.1 El sistema muestra un formulario con las opciones para agregar un usuario.	
2. El Administrador introduce los datos y presiona el botón "Aceptar".	2.1 El sistema agrega el nuevo usuario.	
	2.2 Muestra el usuario agregado en el listado.	
Sección "Eliminar usuario"		
Acción del actor	Respuesta del sistema	
1. El Administrador selecciona la opción de	1.1 El sistema muestra un mensaje: "Está seguro	

eliminar el usuario deseado.	que desea eliminar este usuario".
2. El Administrador acepta.	2.1 Elimina el usuario del listado de usuarios que muestra el sistema.
Sección "Modificar usuario"	
Acción del actor	Respuesta del sistema
1. El Administrador selecciona la opción de modificar el usuario deseado.	1.1 El sistema brinda la posibilidad de modificar los datos del usuario deseado.
2. El Administrador modifica los datos deseados y presiona el botón "Modif".	2.1 El sistema guarda los cambios efectuados.
	2.2 Muestra los cambios efectuados en el listado de usuarios.
Flujo alterno	
Acción del actor	Respuesta del sistema
Puntos de extensión	
Poscondiciones:	El usuario fue gestionado (agregado, eliminado y modificado).
Prioridad:	Crítica
Interfaz:	

Tabla 2.7 Descripción del Caso de Uso Gestionar usuario.

CU 6	Gestionar grupos de usuarios
Actores:	Administrador (Inicia)
Propósito:	Gestionar los grupos de usuarios que tendrán acceso al repositorio de un proyecto dado.

Resumen:	El caso de uso permite al administrador agregar, eliminar y modificar grupos de usuarios.
Referencias:	RF 6, RF 6.1, RF 6.2, RF 6.3.
Precondiciones:	Deben haberse encontrado satisfactoriamente los archivos de configuración.
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. El Administrador selecciona la opción gestionar grupos de usuarios.	<p>1.1 El sistema muestra un listado de los grupos de usuarios existentes y las opciones para Agregar, Eliminar y Modificar.</p> <p>a) Si desea agregar grupos de usuarios ver sección "Agregar grupos de usuarios".</p> <p>b) Si desea eliminar grupos de usuarios ver sección "Eliminar grupos de usuarios".</p> <p>c) Si desea modificar grupos de usuarios ver sección "Modificar grupos de usuarios".</p>
Sección "Agregar grupo de usuarios"	
Acción del actor	Respuesta del sistema
	1.1 El sistema muestra un formulario con las opciones para agregar un grupo de usuarios.
2. El Administrador agrega el nombre del grupo y presiona el botón "Aceptar".	2.1 Crea un nuevo grupo, lo muestra y brinda la opción de agregarle usuarios al mismo.
3. Selecciona los nuevos miembros del grupo y presiona el botón aceptar.	3.1 Agrega los usuarios al grupo creado.
<p>✓ Si no existe un usuario deseado ver flujo alternativo 1.</p>	
	3.2 El sistema muestra el listado de grupos de usuarios actualizado.
Sección "Eliminar grupo de usuarios"	

Acción del actor	Respuesta del sistema
1. El Administrador selecciona la opción de eliminar el grupo deseado.	1.1 El sistema muestra un mensaje: "Está seguro que desea eliminar este grupo".
2. El Administrador acepta.	2.1 Elimina el grupo del listado de grupos de usuarios que muestra el sistema.
Sección "Modificar grupo de usuarios"	
Acción del actor	Respuesta del sistema
1. El Administrador selecciona la opción de modificar el grupo deseado.	1.1 El sistema muestra un formulario que brinda la posibilidad de modificar los datos del grupo deseado.
2. El Administrador modifica los datos deseados y presiona el botón "Guardar".	2.1 El sistema guarda los cambios efectuados.
	2.2 Muestra en el listado de grupos de usuarios los cambios efectuados.
Flujo alternativo 1	
Acción del actor	Respuesta del sistema
3. Selecciona la opción Agregar usuario.	3.1 Muestra la interfaz para agregar un nuevo usuario. Ver caso de uso Gestionar usuario Escenario Agregar usuario.
	3.2 Muestra el(los) usuario(s) agregados en el listado.
4. Selecciona los nuevos miembros del grupo y presiona el botón aceptar.	4.1 Agrega los usuarios al grupo creado.
	4.2 El sistema muestra el listado de grupos de usuarios actualizado.
Puntos de extensión	
Poscondiciones:	El grupo de usuarios fue gestionado (agregado, eliminado y modificado).
Prioridad:	Crítica

Interfaz:	
------------------	--

Tabla 2.8 Descripción del Caso de Uso Gestionar grupos de usuarios.

CU 7	Gestionar permisos a usuarios	
Actores:	Administrador (Inicia)	
Propósito:	Gestionar los permisos de usuarios de un proyecto dado.	
Resumen:	El caso de uso permite al administrador agregar, eliminar y modificar permisos a usuarios.	
Referencias:	RF 7, RF 7.1, RF 7.2, RF 7.3	
Precondiciones:	Deben haberse encontrado satisfactoriamente los archivos de configuración.	
Flujo normal de eventos		
Acción del actor	Respuesta del sistema	
1. El Administrador selecciona la opción gestionar permisos a usuarios.	<p>2. El sistema muestra un listado de los usuarios con la información concerniente a los permisos que posee y su nivel de acceso así como las opciones para Agregar, Eliminar y Modificar.</p> <p>a) Si desea agregar permiso ver sección "Agregar permiso".</p> <p>b) Si desea eliminar permiso ver sección "Eliminar permiso".</p> <p>c) Si desea modificar permiso ver sección "Modificar permiso".</p>	
Sección "Agregar permisos"		
Acción del actor	Respuesta del sistema	
1. El Administrador selecciona el usuario.		
2. Selecciona la opción agregar los permisos.		

3. Agrega permisos y nivel de acceso presiona el botón "Aceptar"	3.1 El sistema guarda los cambios.
	3.2 Muestra los cambios en el listado.
Sección "Eliminar permisos"	
Acción del actor	Respuesta del sistema
1. El Administrador selecciona el usuario.	
2. Selecciona la opción eliminar permiso.	2.1 El sistema muestra un mensaje de confirmación.
3. El Administrador acepta.	3.1 El sistema elimina los permisos del listado de permisos.
Sección "Modificar permisos"	
Acción del actor	Respuesta del sistema
1. El Administrador selecciona el usuario.	
2. Selecciona la opción modificar permiso.	2.1 El sistema muestra un formulario que brinda la posibilidad de modificar los datos deseados.
3. El Administrador modifica los datos deseados y presiona el botón "Aceptar".	3.1 El sistema guarda los cambios efectuados.
	3.2 Muestra los cambios efectuados en el listado.
Flujo alterno	
Acción del actor	Respuesta del sistema
Puntos de extensión	
Poscondiciones:	Los permisos sobre usuarios fueron gestionados (eliminado, agregado y modificado).
Prioridad:	Crítica
Interfaz:	

Tabla 2.9 Descripción del Caso de Uso Gestionar permisos a usuarios.

CU 8	Gestionar permisos a grupos de usuarios	
Actores:	Administrador (Inicia)	
Propósito:	Gestionar los permisos de grupos de usuarios y de un proyecto dado.	
Resumen:	El caso de uso permite al administrador agregar, eliminar y modificar permisos a grupos de usuarios.	
Referencias:	RF 8, RF 8.1, RF 8.2, RF 8.3	
Precondiciones:	Deben haberse encontrado satisfactoriamente los archivos de configuración.	
Flujo normal de eventos		
Acción del actor	Respuesta del sistema	
1. El Administrador selecciona la opción gestionar permisos a grupos de usuarios.	<p>2. El sistema muestra un listado de los grupos de usuarios con la información concerniente a los permisos que posee y su nivel de acceso así como las opciones para Agregar, Eliminar y Modificar.</p> <p>a) Si desea buscar un usuario determinado sección "Buscar grupo de usuario".</p> <p>b) Si desea agregar permiso ver sección "Agregar permiso".</p> <p>c) Si desea eliminar permiso ver sección "Eliminar permiso".</p> <p>d) Si desea modificar permiso ver sección "Modificar permiso".</p>	
Sección "Agregar permisos"		
Acción del actor	Respuesta del sistema	

1. El Administrador selecciona el grupo de usuarios.	
2. Selecciona la opción agregar los permisos.	
3. Agrega permisos y nivel de acceso presiona el botón "Aceptar"	3.1 El sistema guarda los cambios.
	3.2 Muestra los cambios en el listado.
Sección "Eliminar permisos"	
Acción del actor	Respuesta del sistema
1. El Administrador selecciona el grupo de usuarios.	
2. Selecciona la opción eliminar permiso.	2.1 El sistema muestra un mensaje de confirmación.
3. El Administrador acepta.	3.1 El sistema elimina los permisos del listado de permisos.
Sección "Modificar permisos"	
Acción del actor	Respuesta del sistema
1. El Administrador selecciona el grupo de usuarios.	
2. Selecciona la opción modificar permiso.	2.1 El sistema muestra un formulario que brinda la posibilidad de modificar los datos deseados.
3. El Administrador modifica los datos deseados y presiona el botón "Aceptar".	3.1 El sistema guarda los cambios efectuados.
	3.2 Muestra los cambios efectuados en el listado.
Flujo alterno	
Acción del actor	Respuesta del sistema
Puntos de extensión	

Poscondiciones:	Los permisos sobre grupos de usuarios fueron gestionados (eliminado, agregado y modificado).
Prioridad:	Crítica
Interfaz:	

Tabla 2.10 Descripción del Caso de Uso Gestionar permisos a grupo de usuarios.

Conclusiones

El desarrollo de este capítulo permitió la elaboración de un conjunto de artefactos que contribuyeron a establecer una mejor comprensión del sistema así como las condiciones que este debe cumplir. Se precisaron detalles específicos: requerimientos funcionales y no funcionales del sistema, los actores que intervienen y los casos de uso del sistema fueron detallados con exactitud. De esta forma se está en condiciones de iniciar el flujo de trabajo Análisis y Diseño que debe estar guiado por los requisitos y los casos de uso descritos.

Capítulo 3: ANÁLISIS Y DISEÑO DEL SISTEMA

Este capítulo muestra el desarrollo del flujo Análisis y Diseño, donde se describe cómo implementar el sistema. Se construyen diagramas de clases del análisis para cada caso de uso, así como diagramas de colaboración para cada escenario en los casos de uso. Además se crean diagramas de clases del diseño para cada uno de los casos de uso descritos en el capítulo anterior.

3.1 Análisis

El objetivo más importante del Análisis es obtener una visión del sistema, se enfoca en ver qué hace este, por lo que toma en cuenta los requerimientos funcionales. Se genera además, la realización de los casos de uso del análisis, lo que constituye el principal artefacto que se obtiene en este flujo de trabajo, el cual describe como se ejecutan los casos de uso en términos de clases del análisis y su interacción.

3.1.1 Modelo de análisis

El Modelo de análisis incluye las clases que describen la realización de los casos de usos, atributos y relaciones que se establecen entre ellas. De esta forma se obtiene el diagrama de clases del análisis. En este están representados, en resumen, los conceptos asociados al dominio del problema. El Modelo de análisis se utiliza como entrada en las actividades de Diseño e Implementación.

3.1.1.2 Diagramas de clases del análisis

Las clases de análisis enfocan los requisitos funcionales y son fácilmente identificables en el dominio del problema porque representan conceptos y relaciones del dominio. Se caracterizan por poseer atributos y entre ellas se establecen relaciones de asociación, agregación/ composición, generalización/ especialización y otros tipos asociativos. RUP propone clasificar estas clases en:

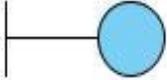
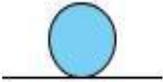
Nombre	Características	Representación
Interfaz	Son las clases encargadas de modelar la interacción Actor – Sistema. Son representativas de elementos como ventanas, formularios, comunicación con otros sistemas o dispositivos, interfaces de impresoras, y otros.	
Entidad	Las clases entidad son las que modelan la información persistente del sistema y el comportamiento asociado a una información. Estas muestran generalmente una estructura de datos lógica y permiten entender de qué información depende el sistema.	
Control	Son las encargadas de coordinar el trabajo de las clases. Su característica principal es que encapsulan el comportamiento de un CU. De esta manera los aspectos dinámicos del sistema son modelados con clases de control, pues estas manejan y coordinan las acciones y los flujos de control principales, delegando responsabilidades a otros objetos, es decir objetos de interfaz y de entidad.	

Tabla 3.1 Clases del análisis.

Diagrama de clases del análisis CUS Autenticar usuario

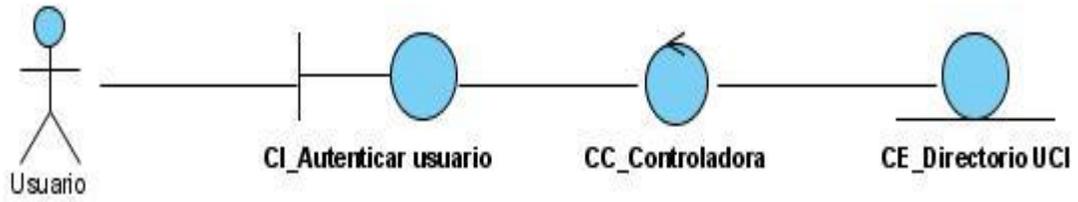


Tabla 3.2 Diagrama de Clases del Análisis CUS Autenticar usuario.

Diagrama de clases del análisis CUS Buscar repositorios

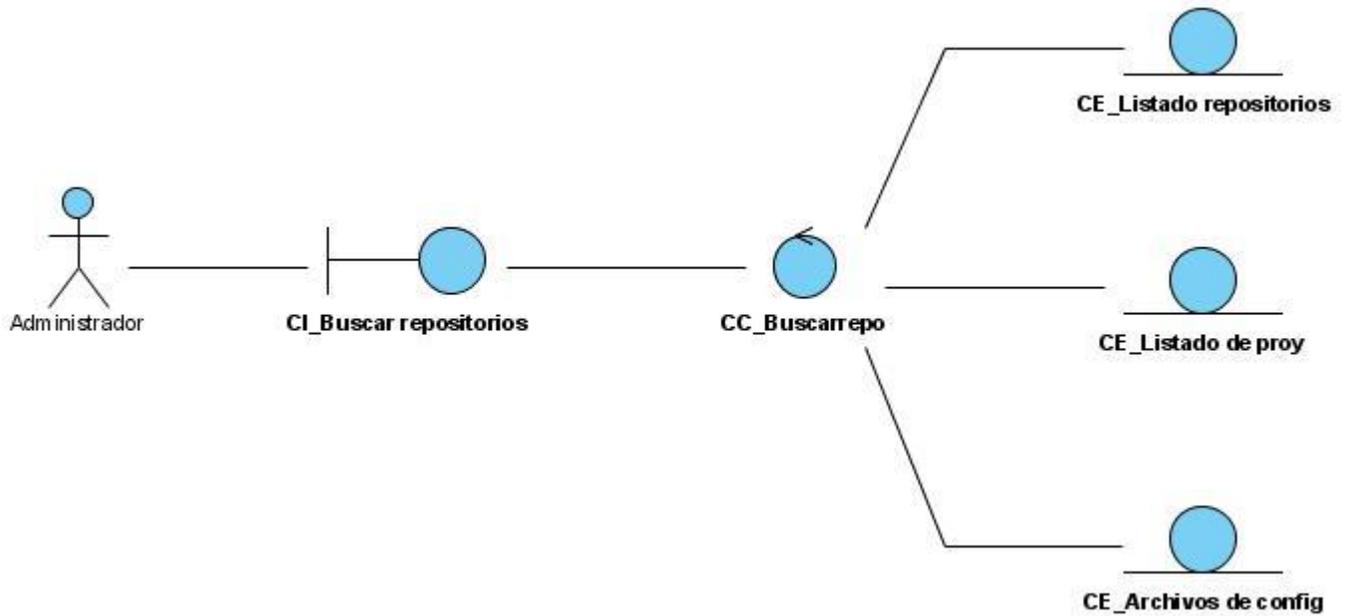


Tabla 3.3 Diagrama de Clases del Análisis CUS Buscar repositorios.

Diagrama de clases del análisis CUS Buscar archivos de configuración

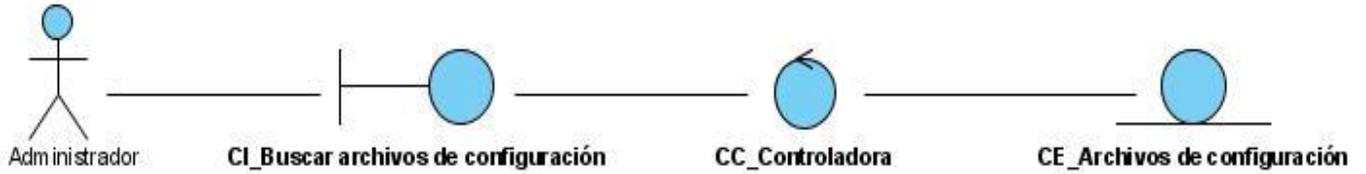


Tabla 3.4 Diagrama de Clases del Análisis CUS Buscar archivos de configuración.

Diagrama de clases del análisis CUS Mostrar perfiles

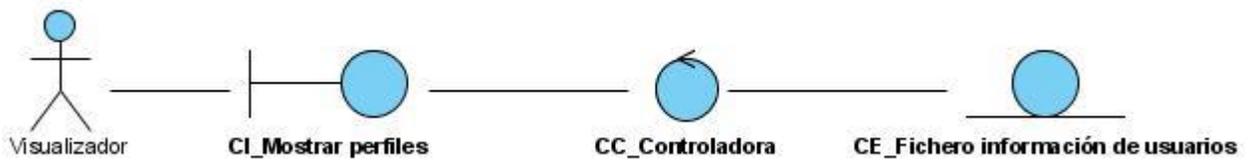


Tabla 3.5 Diagrama de Clases del Análisis CUS Mostrar perfiles.

Diagrama de clases del análisis CUS Gestionar usuarios



Tabla 3.6 Diagrama de Clases del Análisis CUS Gestionar usuarios.

Diagrama de clases del análisis CUS Gestionar grupos de usuarios

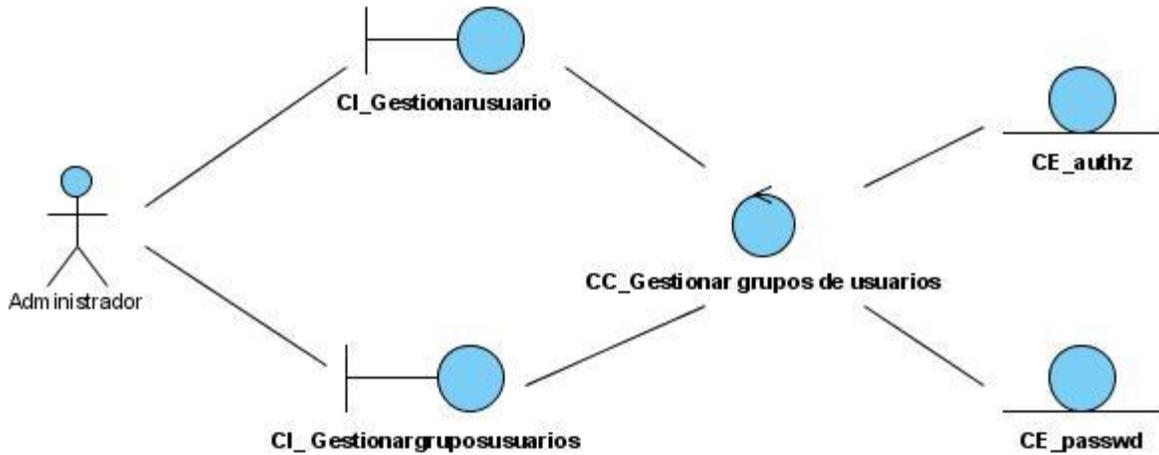


Tabla 3.7 Diagrama de Clases del Análisis CUS Gestionar grupos de usuarios.

Diagrama de clases del análisis CUS Gestionar permisos a usuarios

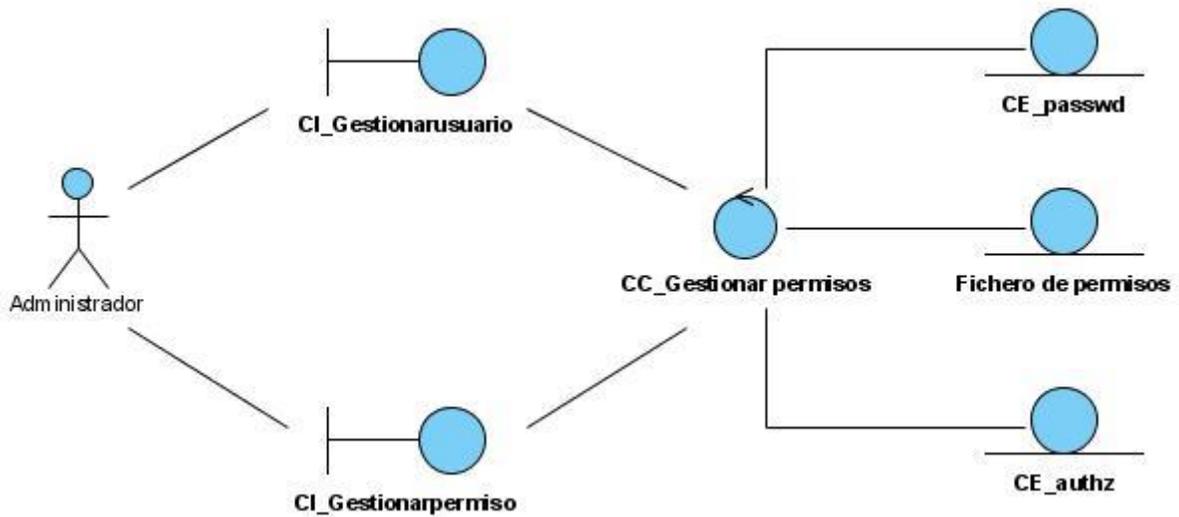


Tabla 3.8 Diagrama de Clases del Análisis CUS Gestionar permisos a usuarios.

Diagrama de clases del análisis CUS Gestionar permisos a grupos de usuarios

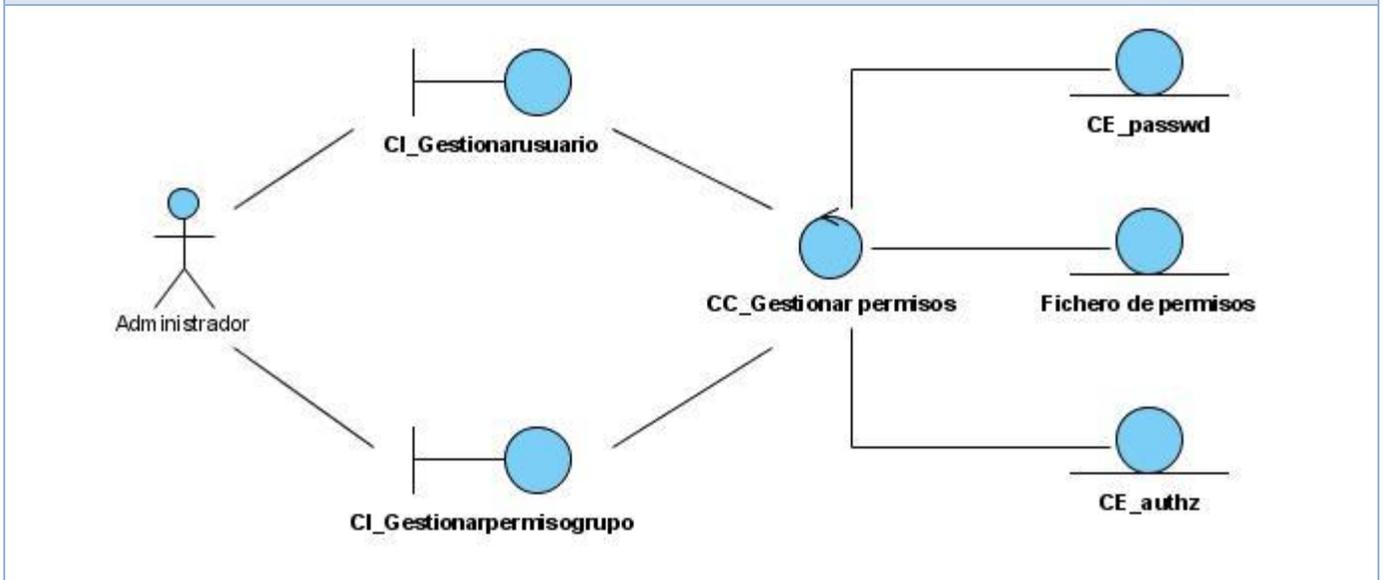


Tabla 3.9 Diagrama de Clases del Análisis CUS Gestionar permisos a grupos de usuarios.

3.1.1.3 Diagramas de interacción

Un Diagrama de Interacción muestra una interacción entre un conjunto de objetos, sus relaciones, e incluye los mensajes que se pueden enviar entre ellos o entre subsistemas. Se utilizan para modelar los aspectos dinámicos de un sistema y se dividen en Diagramas de Colaboración y Diagramas de Interacción.

Un Diagrama de Colaboración muestra las interacciones entre objetos creando enlaces entre estos, además de añadir mensajes a los enlaces creados. Este tipo de diagrama de interacción destaca la organización estructural de los objetos que envían y reciben mensajes.

A continuación se muestran los diagramas de colaboración para la herramienta PhpSvnAdmin.

Diagrama de Colaboración del Análisis CU Autenticar usuario

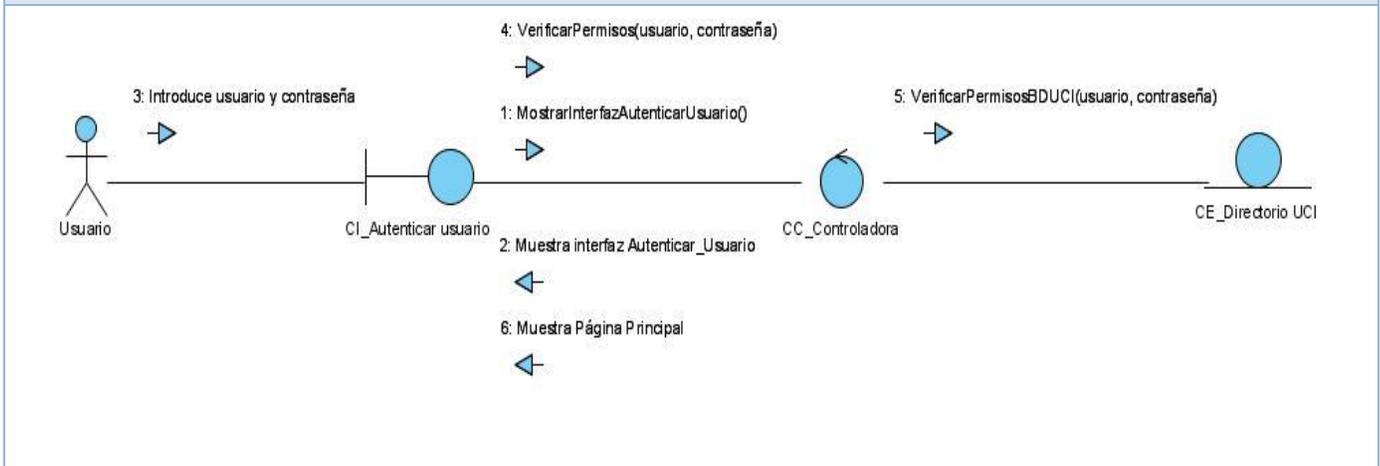


Tabla 3.10 Diagrama de Colaboración del Análisis CU Autenticar usuario.

Diagrama de Colaboración del Análisis CU Buscar repositorios

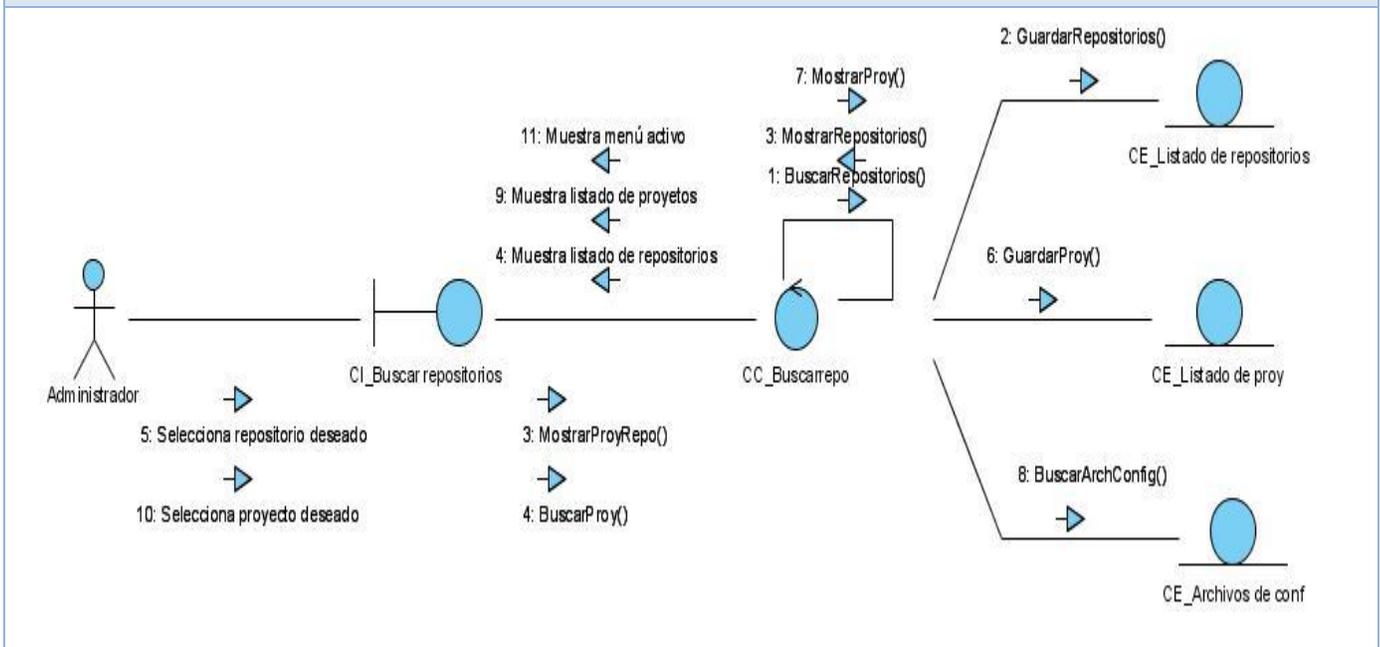


Tabla 3.11 Diagrama de Colaboración del Análisis CU Buscar repositorios.

Diagrama de Colaboración del Análisis CU Buscar archivos de configuración

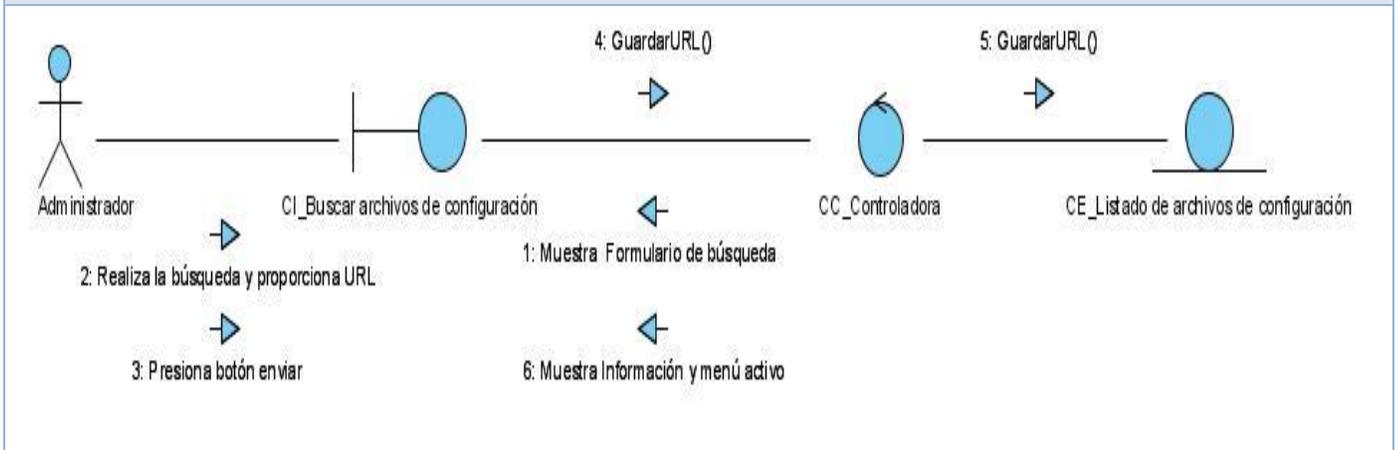


Tabla 3.12 Diagrama de Colaboración del Análisis CU Buscar archivos de configuración.

Diagrama de Colaboración del Análisis CU Mostrar perfiles

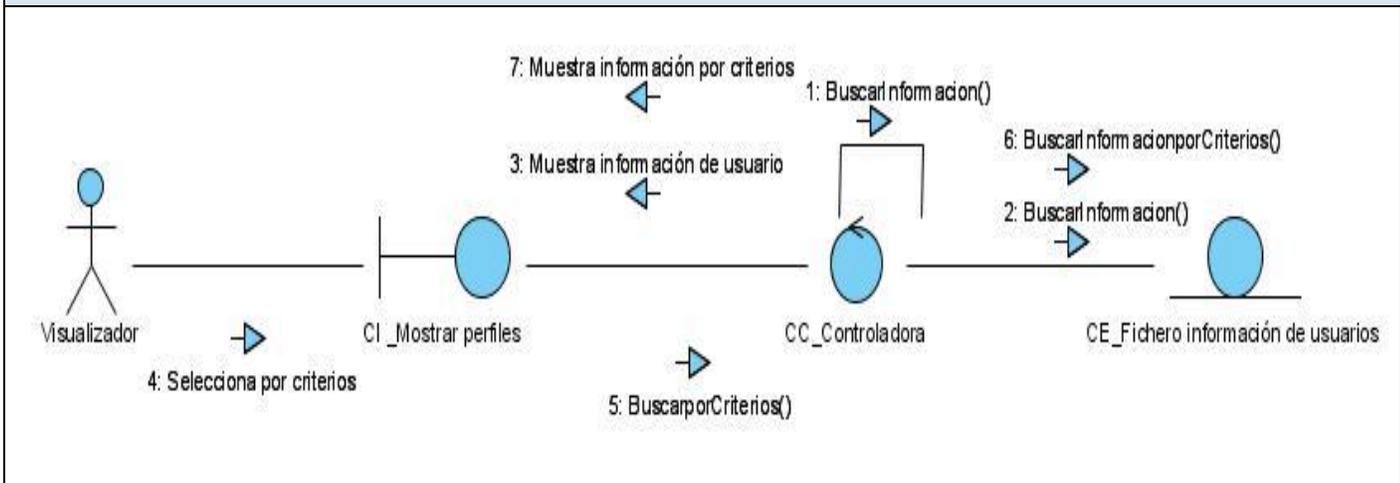


Tabla 3.13 Diagrama de Colaboración del Análisis CU Mostrar perfiles.

Diagrama de Colaboración CU Gestionar usuarios Escenario Agregar usuario

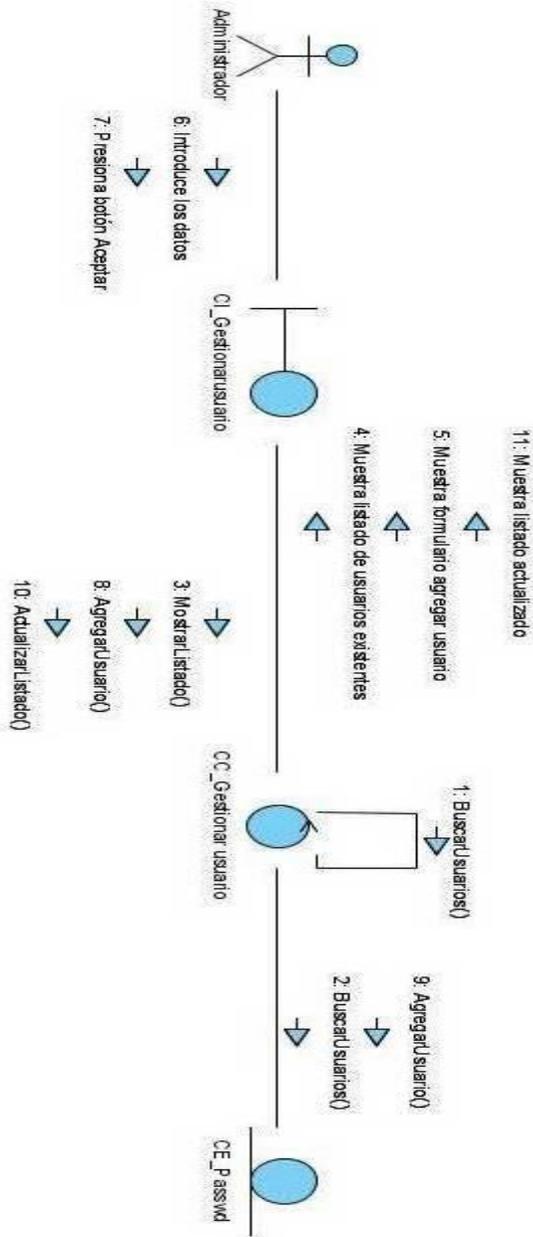


Tabla 3.14 Diagrama de Colaboración del Análisis CU Gestionar usuarios Escenario Agregar usuario.

Diagrama de Colaboración CU Gestionar usuarios Escenario Modificar usuario

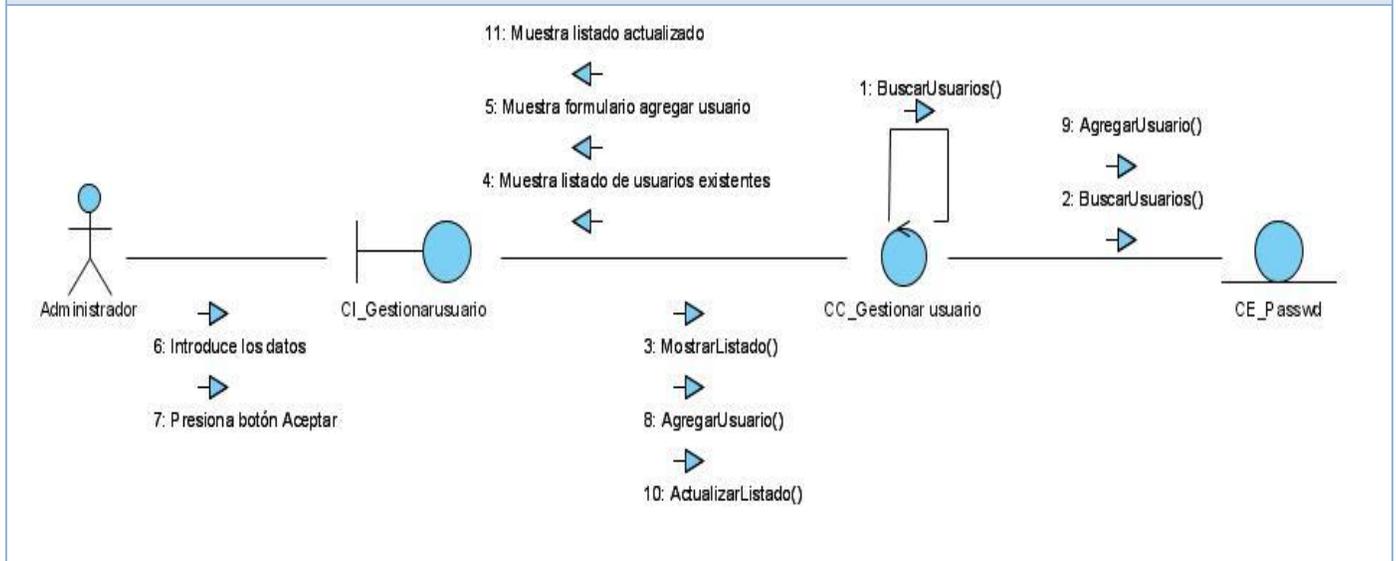


Tabla 3.15 Diagrama de Colaboración del Análisis CU Gestionar usuarios Escenario Modificar usuario.

Diagrama de Colaboración CU Gestionar usuarios Escenario Eliminar usuario

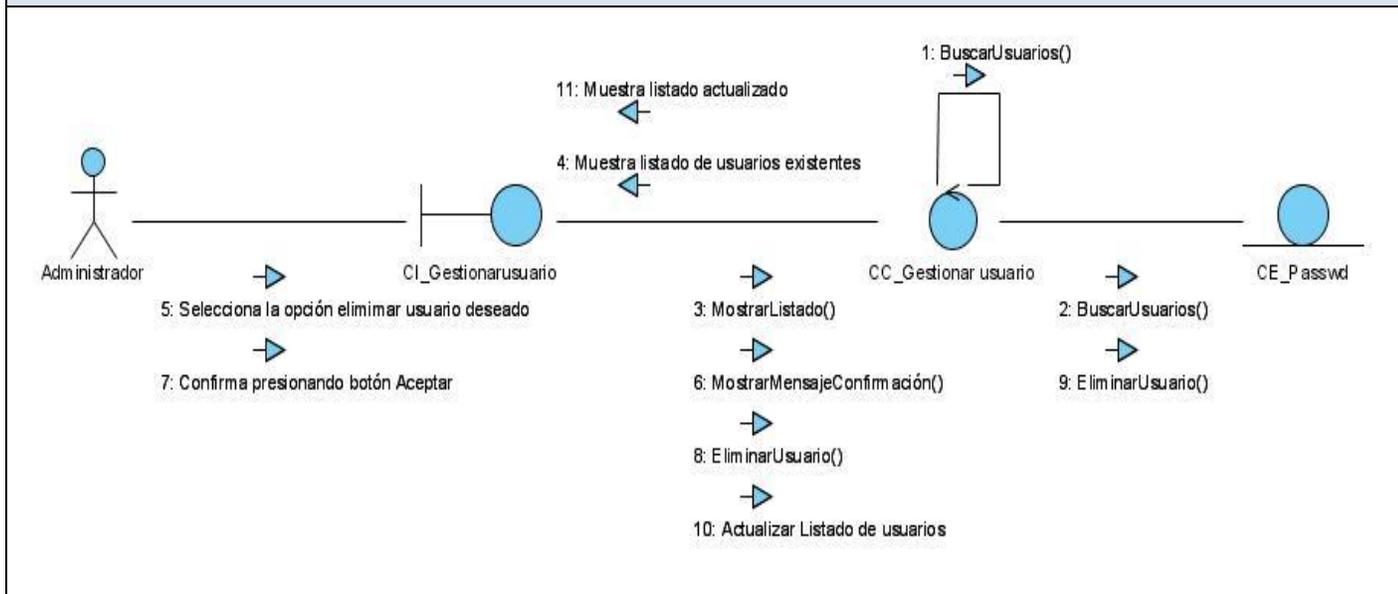


Tabla 3.16 Diagrama de Colaboración del Análisis CU Gestionar usuarios Escenario Eliminar usuario.

Diagrama de Colaboración CU Gestionar grupos de usuarios Escenario Agregar grupos de usuarios

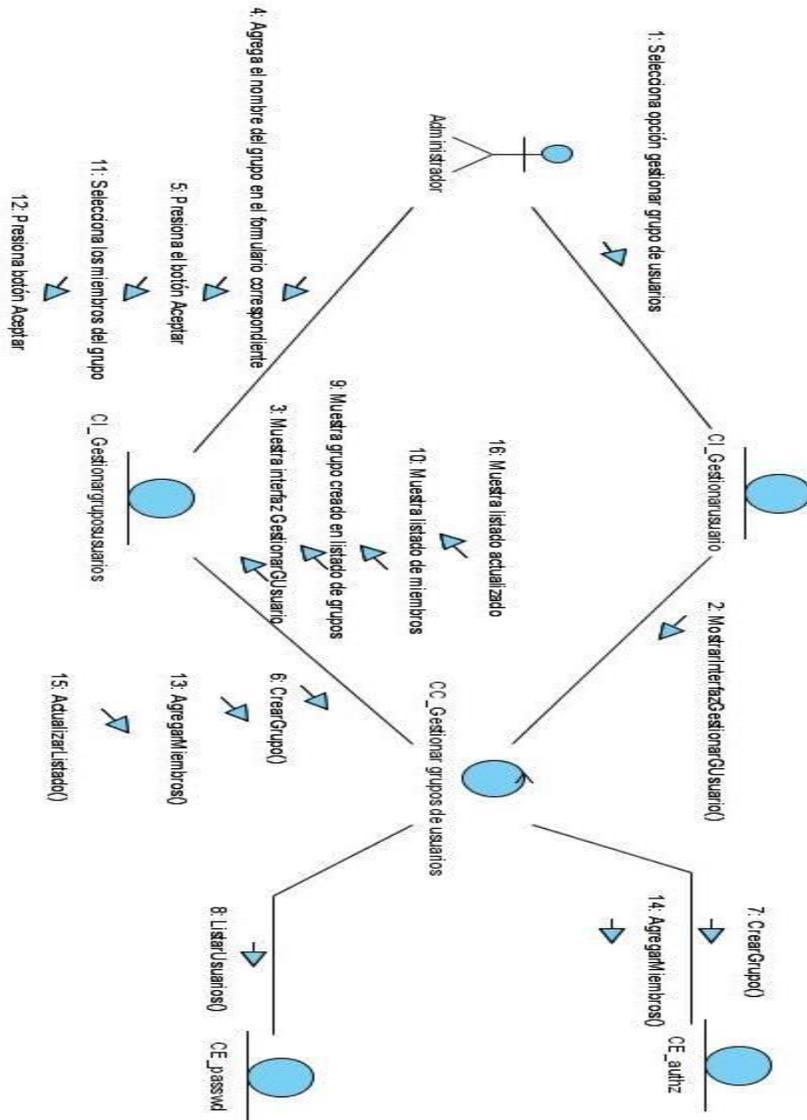


Tabla 3.17 Diagrama de Colaboración del Análisis CU Gestionar grupos de usuarios Escenario Agregar grupos de usuarios.

Diagrama de Colaboración CU Gestionar grupos de usuarios Escenario Modificar grupos de usuarios.

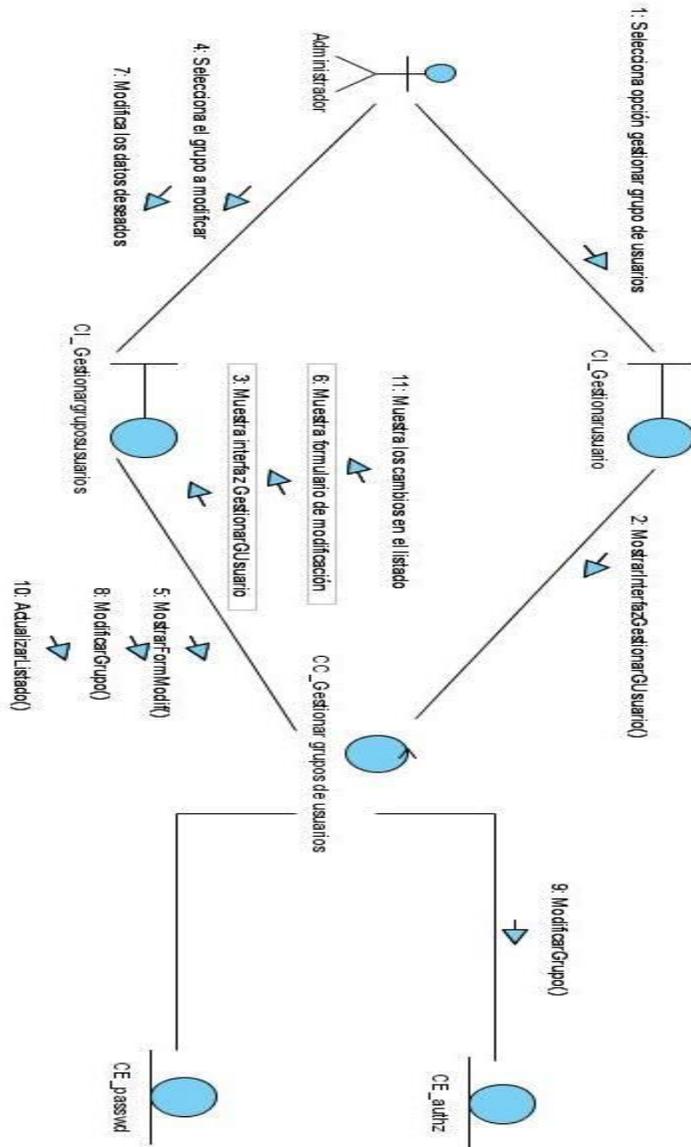


Tabla 3.18 Diagrama de Colaboración del Análisis CU Gestionar grupos de usuarios Escenario Modificar grupos de usuarios.

Diagrama de Colaboración CU Gestionar grupos de usuarios Escenario Eliminar grupos de usuarios.

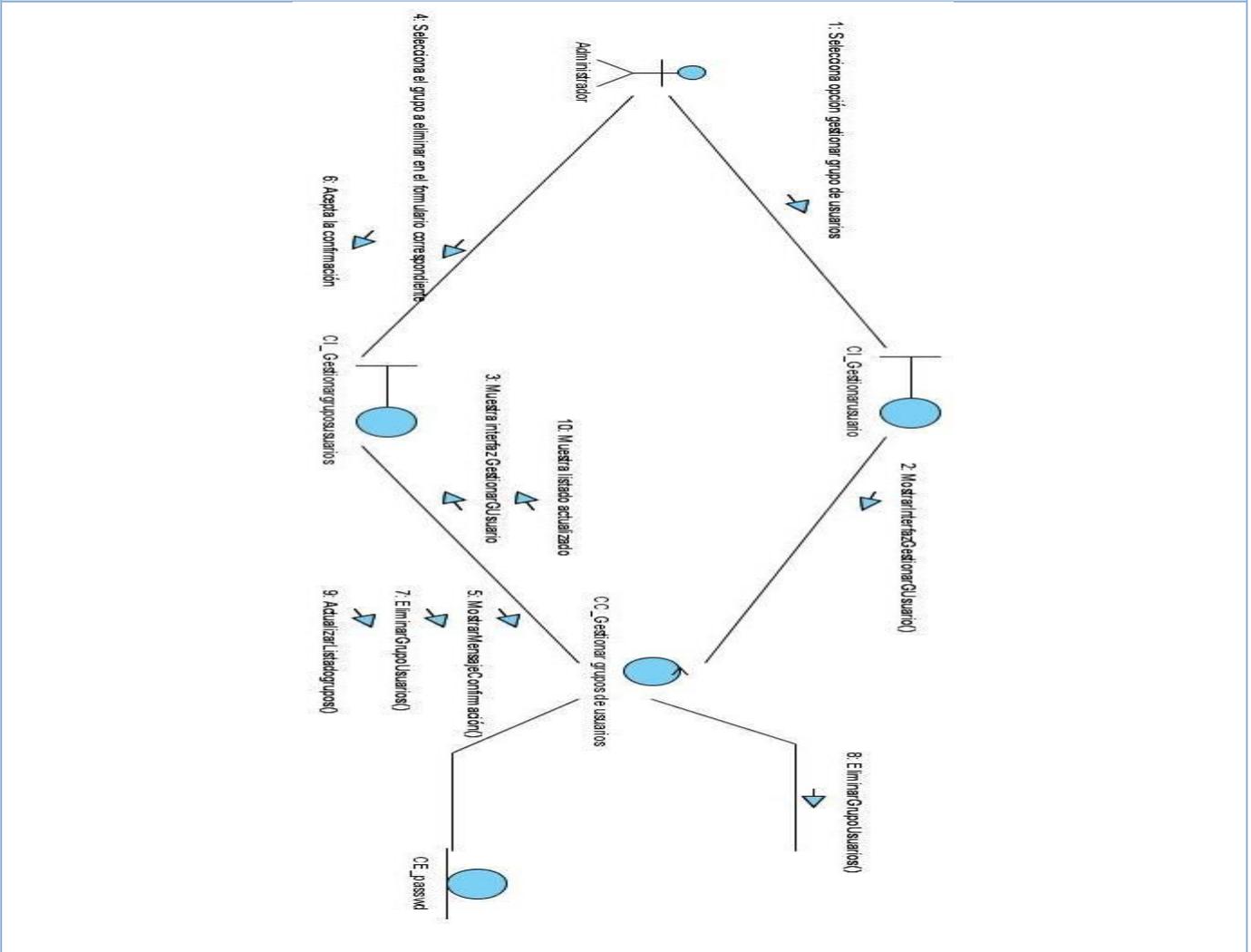


Tabla 3.19 Diagrama de Colaboración del Análisis CU Gestionar grupos de usuarios Escenario Eliminar grupos de usuarios.

Diagrama de Colaboración CU Gestionar permisos a usuarios Escenario Agregar permisos

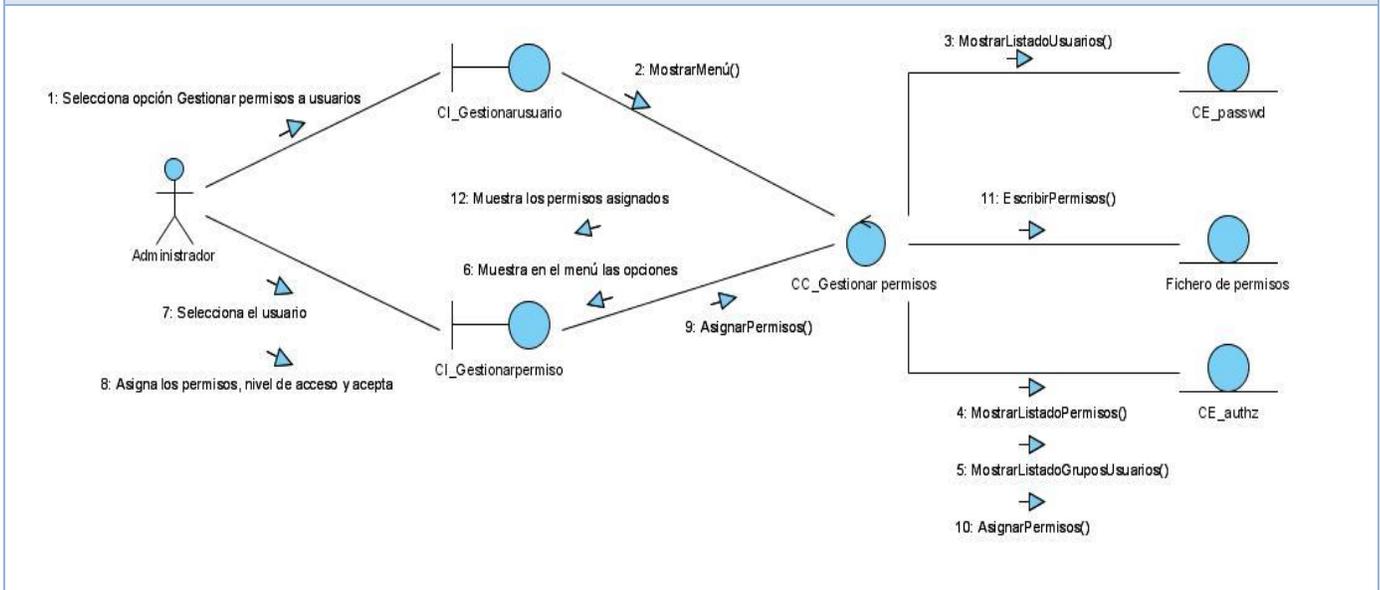


Tabla 3.20 Diagrama de Colaboración del Análisis CU Gestionar permisos a usuarios Escenario Agregar permisos.

Diagrama de Colaboración CU Gestionar permisos a usuarios Escenario Modificar permisos

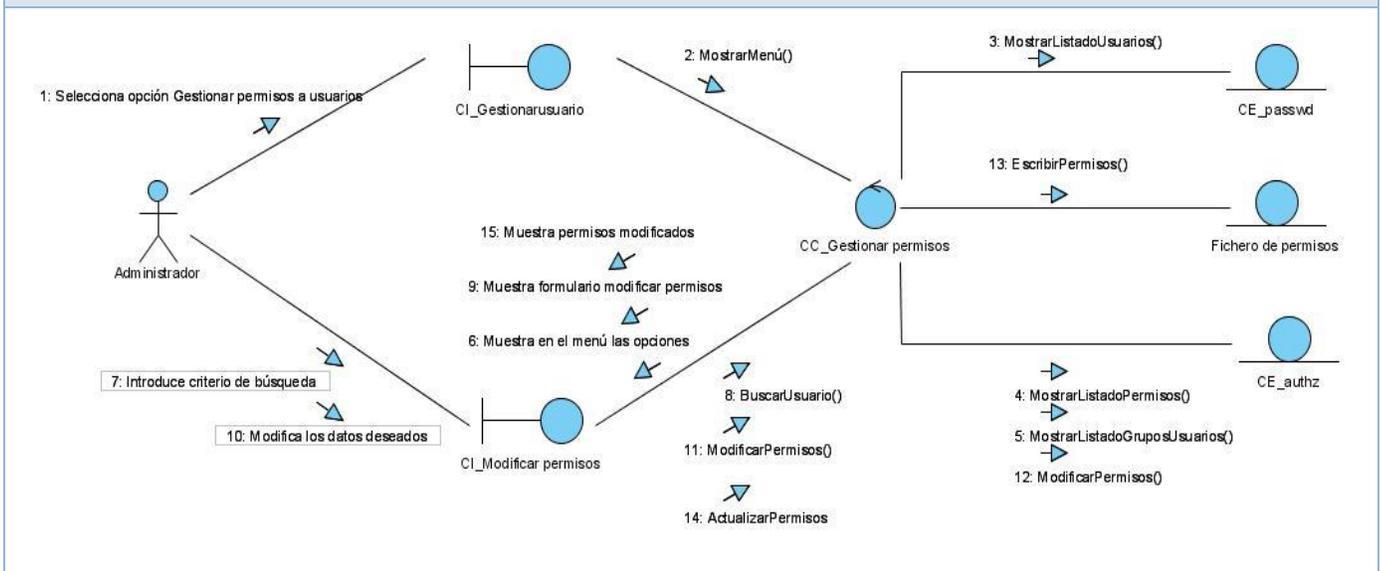


Tabla 3.21 Diagrama de Colaboración del Análisis CU Gestionar permisos Escenario Modificar permisos.

Diagrama de Colaboración CU Gestionar permisos a usuarios Escenario Eliminar permisos

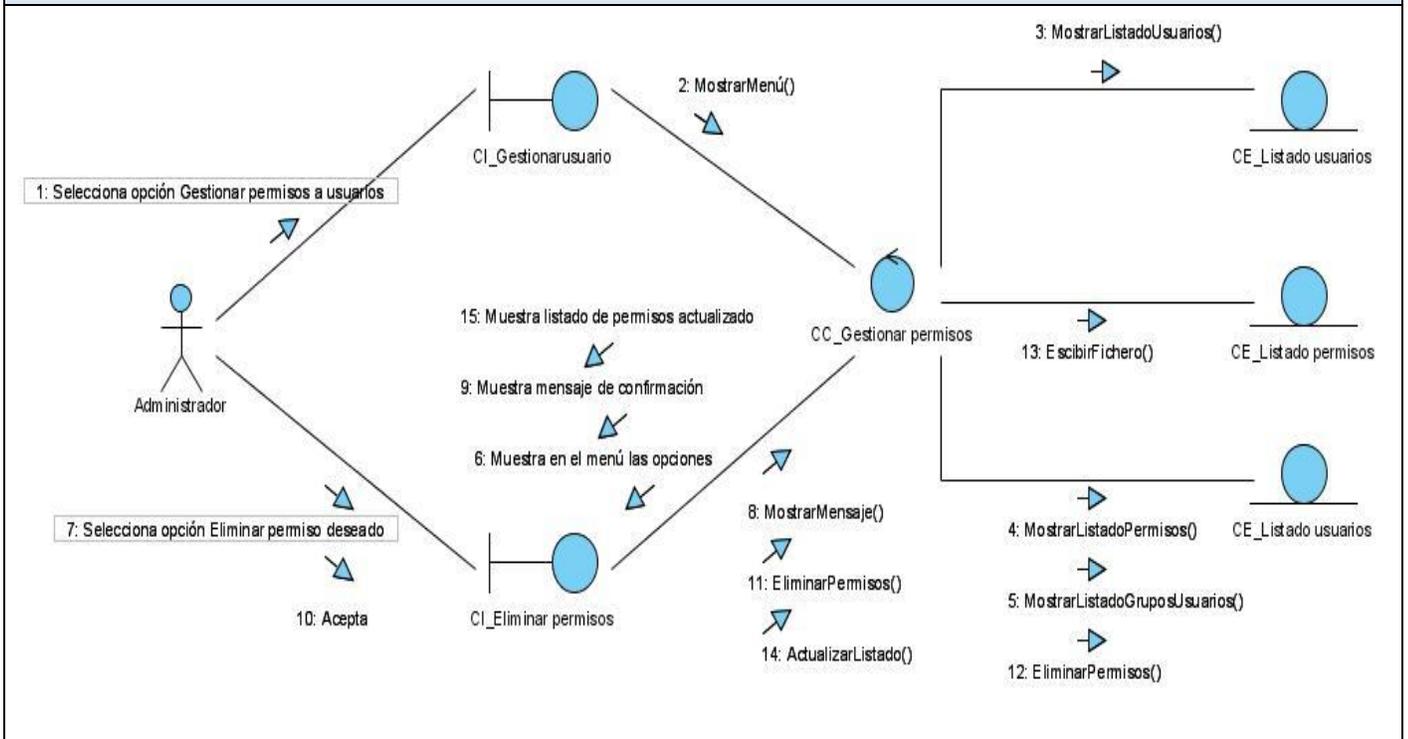


Tabla 3.22 Diagrama de Colaboración del Análisis CU Gestionar permisos a usuarios Escenario Eliminar permisos.

Diagrama de Colaboración CU Gestionar permisos a grupos de usuarios Escenario Agregar permisos

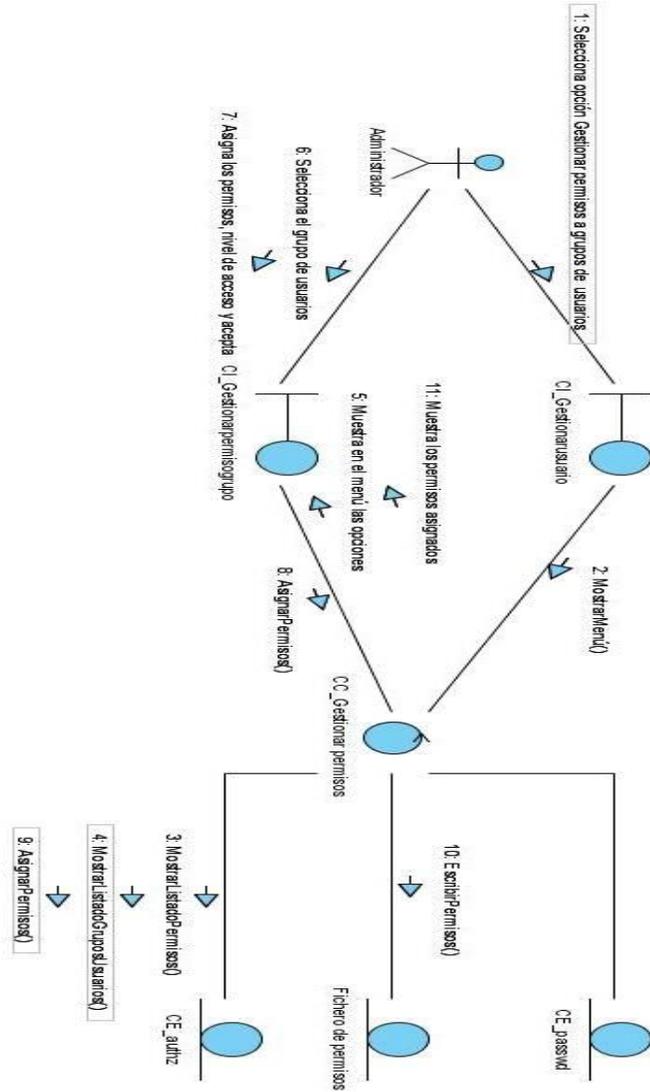


Tabla 3.23 Diagrama de Colaboración del Análisis CU Gestionar permisos a grupos de usuarios Escenario Agregar permisos.

Diagrama de Colaboración CU Gestionar permisos a grupos de usuarios Escenario Modificar permisos

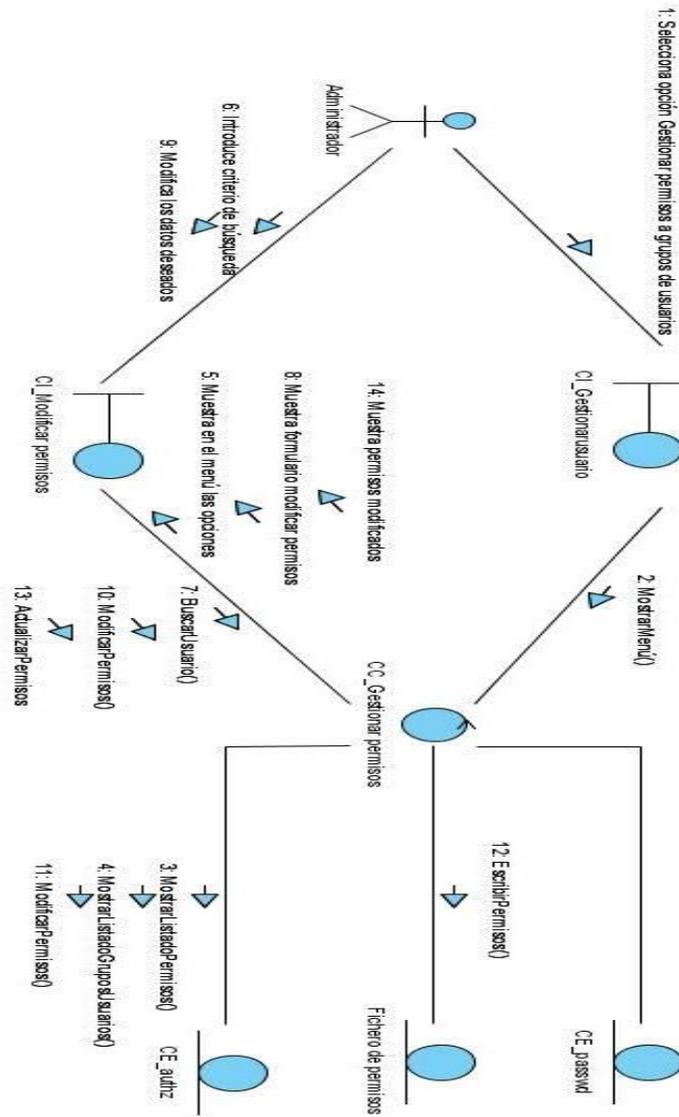


Tabla 3.24 Diagrama de Colaboración del Análisis CU Gestionar permisos a grupos de usuarios Escenario Modificar permisos.

Diagrama de Colaboración CU Gestionar permisos a grupos de usuarios Escenario Modificar permisos

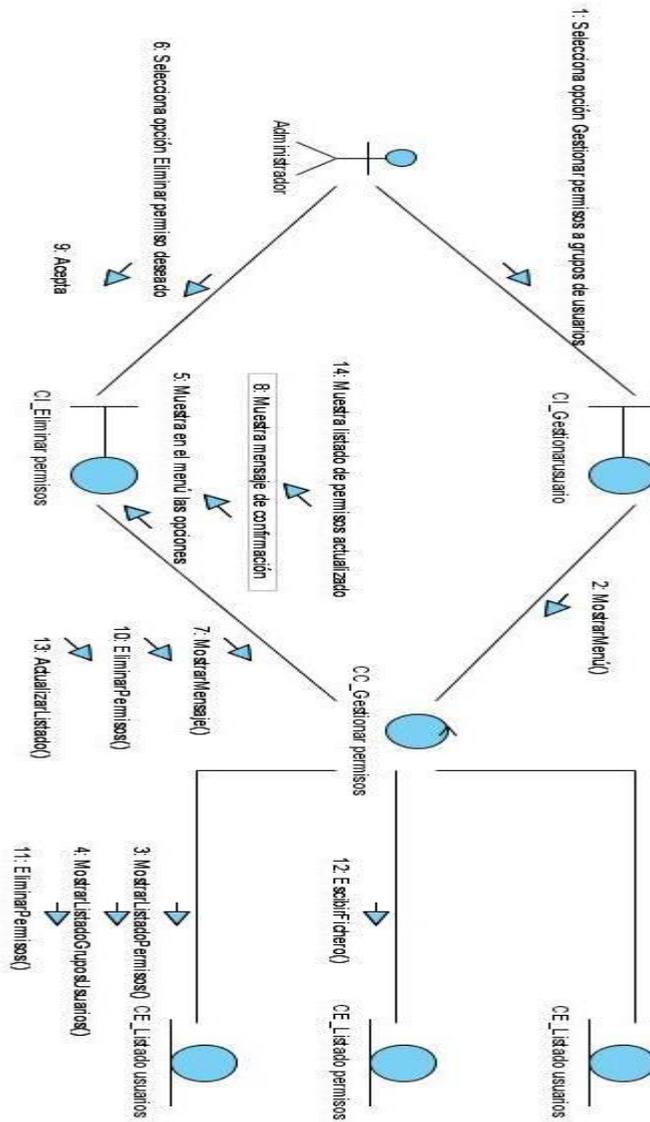


Tabla 3.25 Diagrama de Colaboración del Análisis CU Gestionar permisos a grupos de usuarios Escenario Eliminar permisos.

3.2 Diseño

El diseño constituye un refinamiento del análisis. Se enfoca en los requerimientos no funcionales, o sea, en cómo cumple el sistema sus objetivos. Al finalizar, un buen diseño debe permitir que el sistema sea implementado sin ambigüedades.

3.2.1 Modelo de Diseño

El Modelo de Diseño constituye la realización física de los casos de uso, teniendo en cuenta restricciones de implementación como el lenguaje de programación a utilizar así como componentes, sistemas operativos y tecnologías usadas en la realización del sistema, además de basarse en la comprensión de los aspectos relacionados con los requisitos no funcionales. Constituye además, el punto de partida para el inicio de la fase de Implementación del software.

En esta estructura se desglosan los trabajos de implementación en partes manejables, lo que permite que sobre ellas trabajen diferentes equipos de desarrollo. De esta forma capturan las interfaces entre subsistemas en el ciclo de desarrollo del software. Los artefactos resultantes en el Modelo de Diseño son: Modelo de despliegue, Descripción de la Arquitectura, Realización de Casos de Uso, Clase del Diseño, Subsistema de Diseño e Interfaz de usuario.

3.2.1.1 Diagrama de clases de diseño

El diagrama de clases de diseño constituye la representación de la estructura y organización que poseen las clases que intervienen en un caso de uso determinado pero teniendo en cuenta cuestiones referente a la implementación y especificando los métodos a usar en las clases y las relaciones entre estas.

Clases del diseño

La extensión UML para Web presenta como elementos más significativos tres clases con los siguientes estereotipos: “Server Page”, empleado para el código servidor; “Client Page”, código cliente; y “Form” para los formularios.



Tabla 3.26 Clases del Diseño.

<<Client Page>>: Representa una página Web con formato XHTML. Constituye una mezcla de datos, presentación y lógica, además puede contener scripts que serán interpretados por el navegador o navegador.

<<Server Page>>: Representa una página web cuyos scripts son ejecutados por el servidor, estos interactúan con los recursos que se encuentran con el servidor: Base de datos, sistemas externos, lógica del negocio, etc.

<<Form>>: Representa un formulario, que no es más que una colección de campos de entrada de datos, que parte de una «Client Page».

Estereotipos de asociación:

<<Build>>: Asociación que representa la relación entre las páginas clientes (<<Client Page>>). Expresa cómo las páginas que se encuentran en el servidor construyen las páginas en el cliente: una página servidor construye una o más páginas cliente.

<<Call>>: Es usada para realizar llamadas a páginas servidoras, representativas de métodos en la capa del negocio.

<<Include>>: Representa la asociación donde una página servidor puede incluir a otra página del mismo tipo, de esta forma tiene acceso a la generalidad de las funcionalidades brindadas por la misma.

<<Link>>: Asociación que permite acceder desde una página cliente a otra de su mismo tipo.

<<Redirect>>: Asociación que representa que una página servidora puede redireccionar el procesamiento a otra página, o sea, envía información a otra página para que ejecute la acción.

<<Submit>>: Es la asociación que permite enviar los valores contenidos en un formulario (<<Form>>) a una página servidora (<<Server Page>>).

A continuación se muestran los diagramas de clases del diseño para la herramienta **PhpSsnAdmin**.

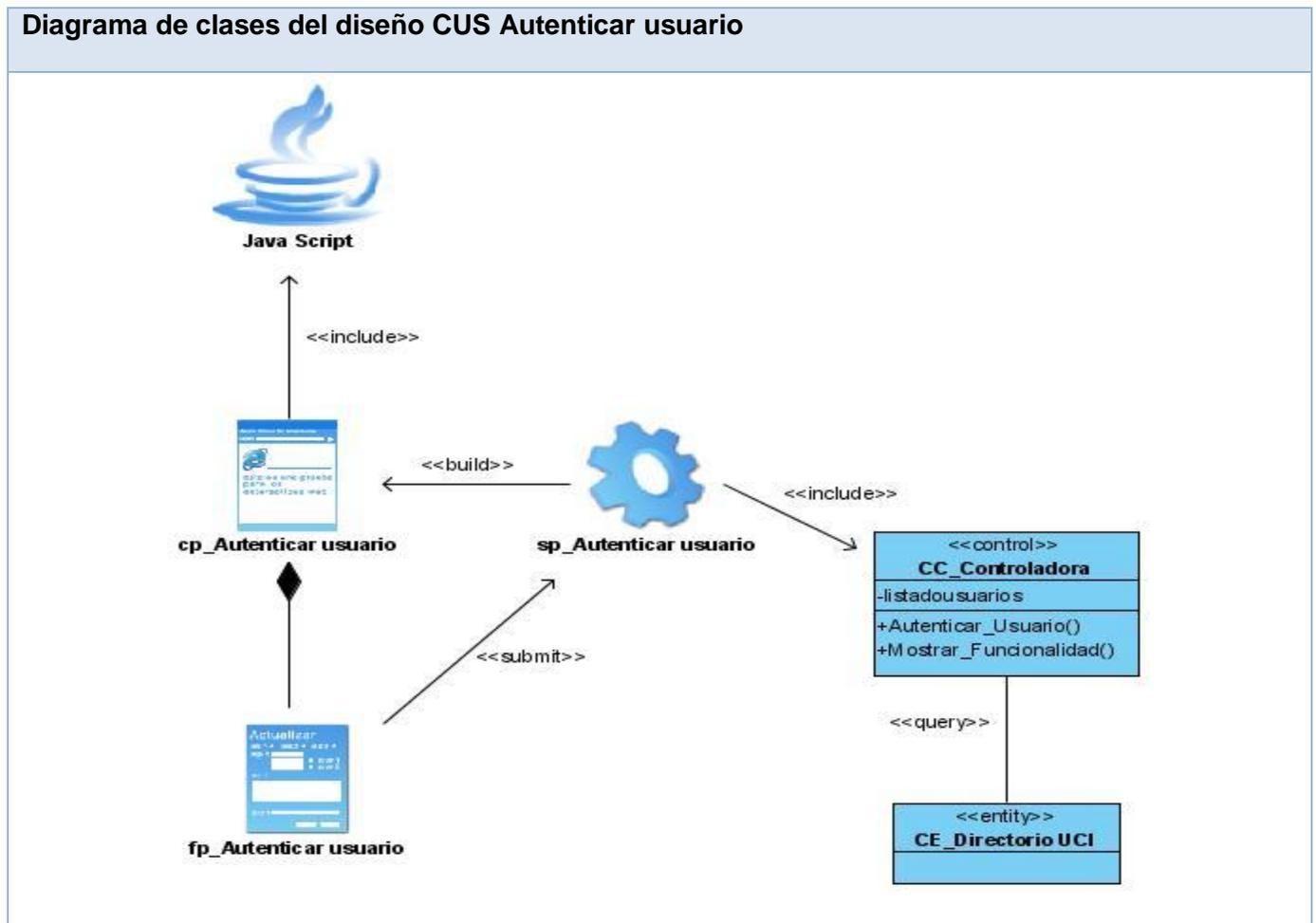


Tabla 3.27 Diagrama de clases del diseño CUS Autenticar usuario.

Diagrama de clases del diseño CUS Buscar repositorios

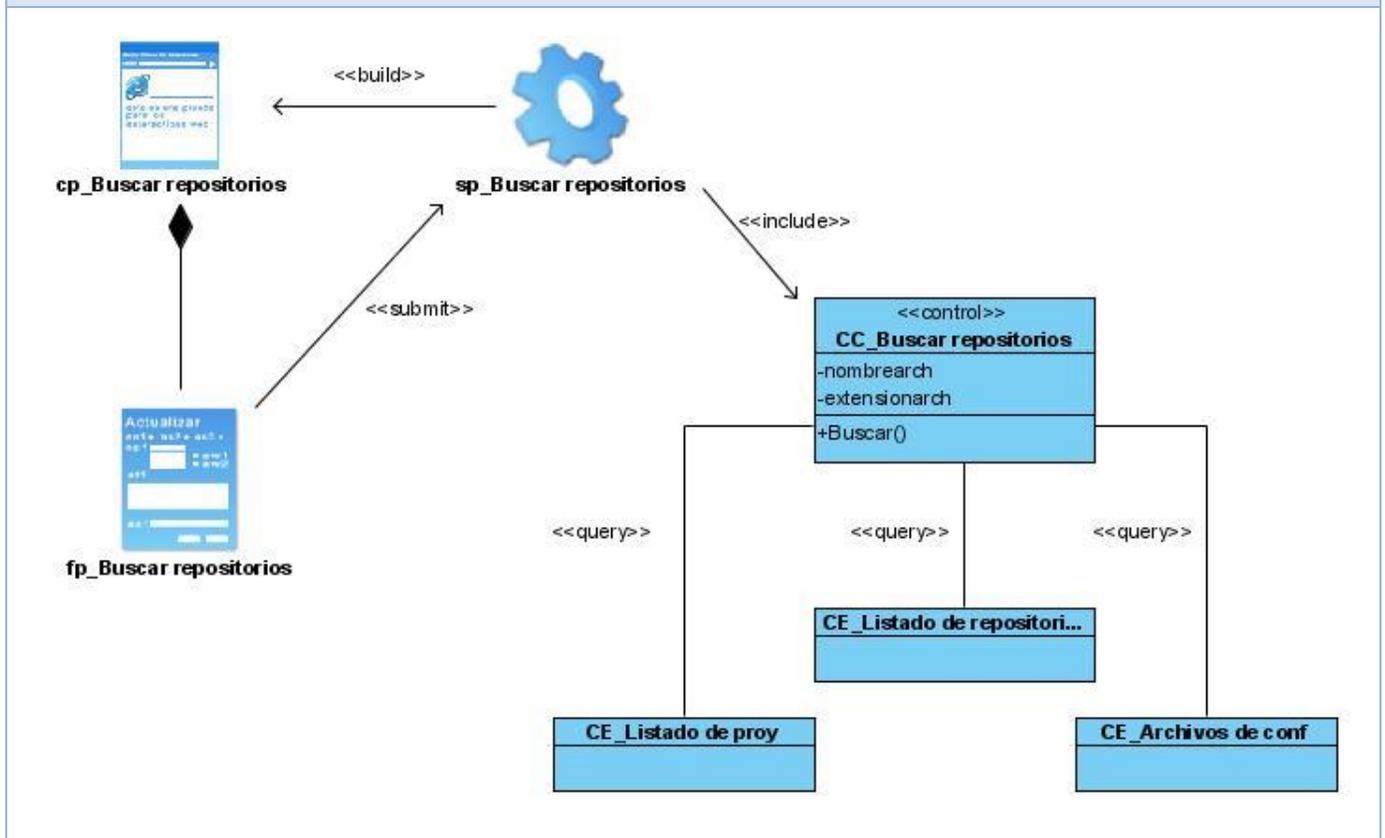


Tabla 3.28 Diagrama de clases del diseño CUS Buscar repositorios.

Diagrama de clases del diseño CUS Buscar archivos de configuración

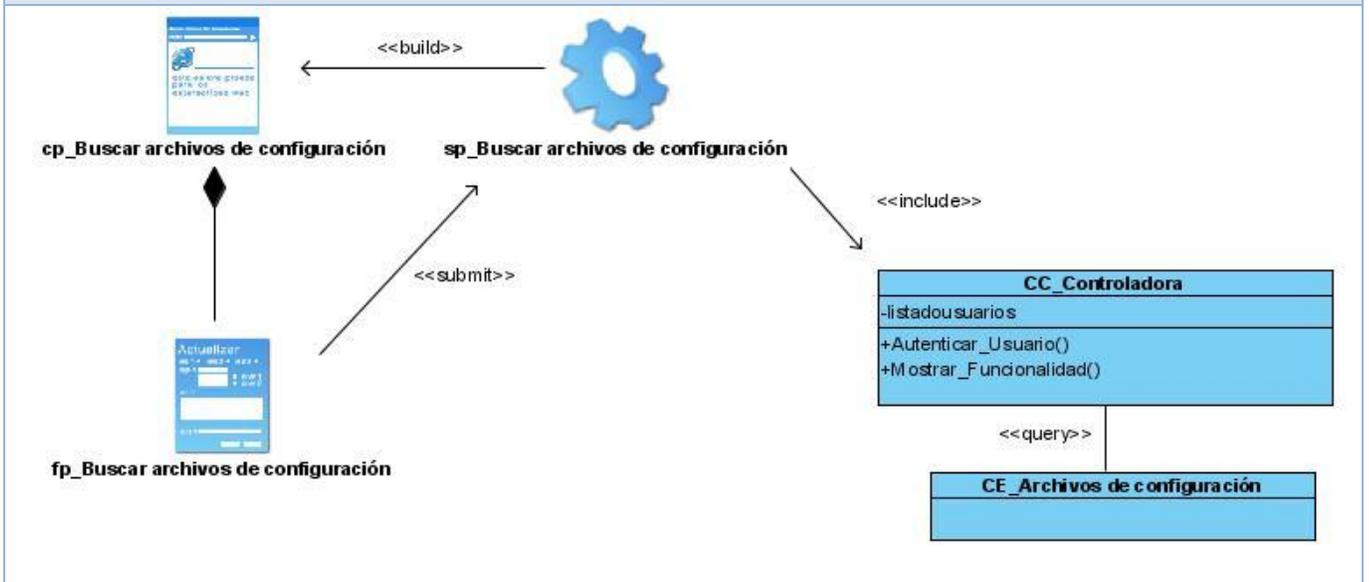


Tabla 3.29 Diagrama de clases del diseño CUS Buscar archivos de configuración.

Diagrama de clases del diseño CUS Mostrar perfiles

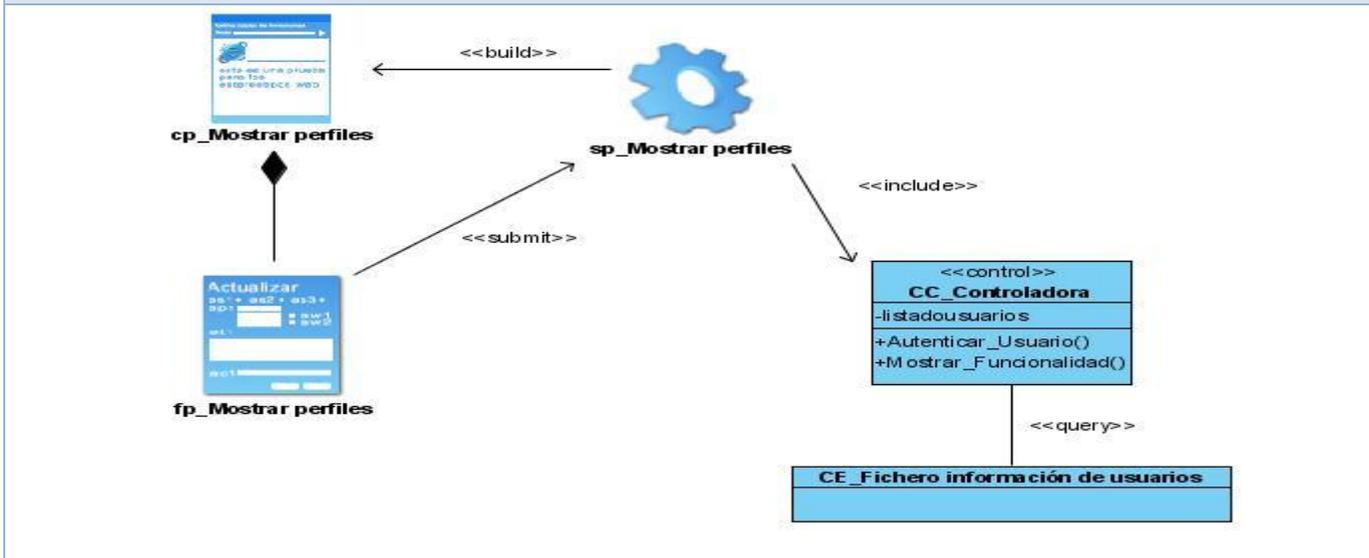


Tabla 3.30 Diagrama de clases del diseño CUS Mostrar perfiles.

Diagrama de clases del diseño CUS Gestionar usuario

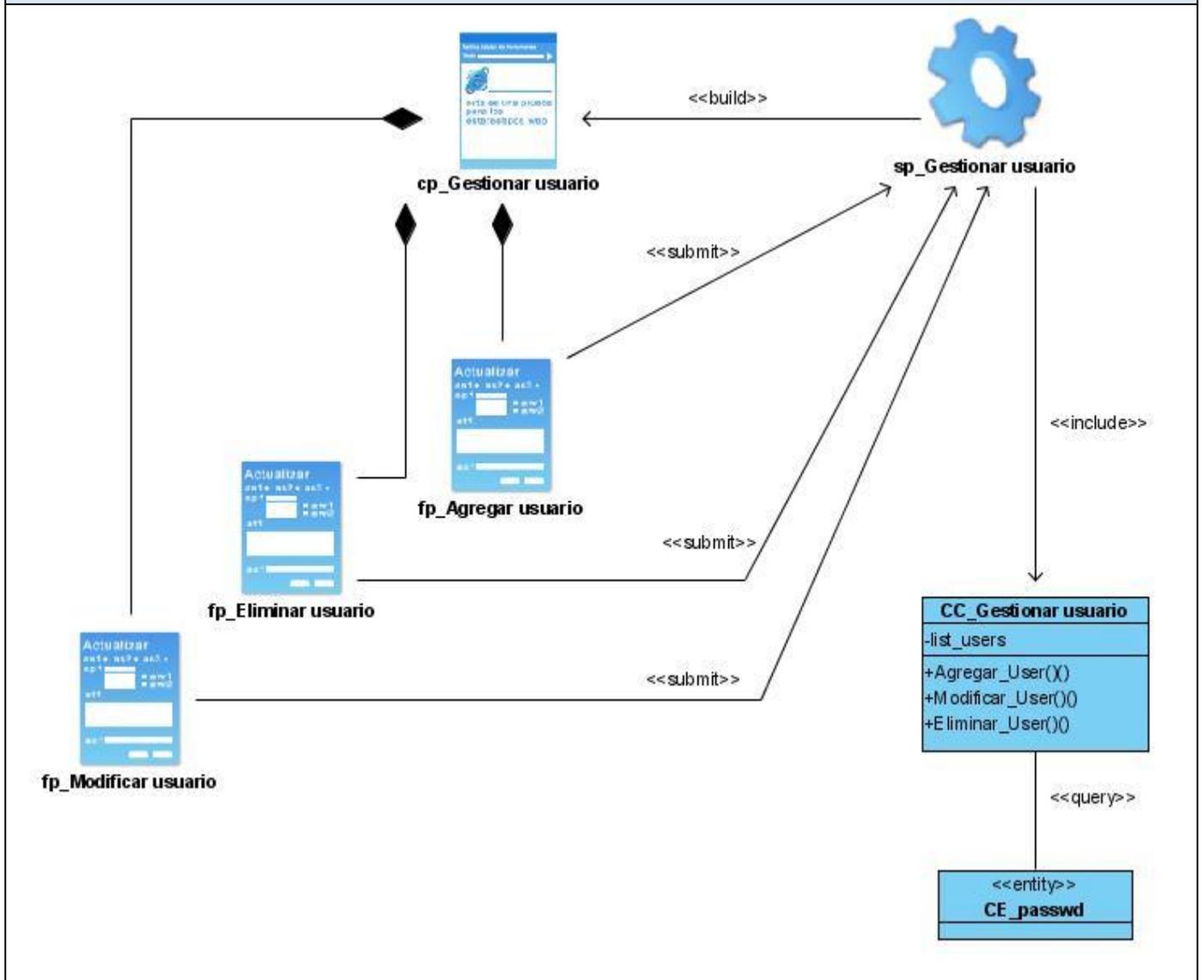


Tabla 3.31 Diagrama de clases del diseño CUS Gestionar usuario.

Diagrama de clases del diseño CUS Gestionar grupos de usuarios

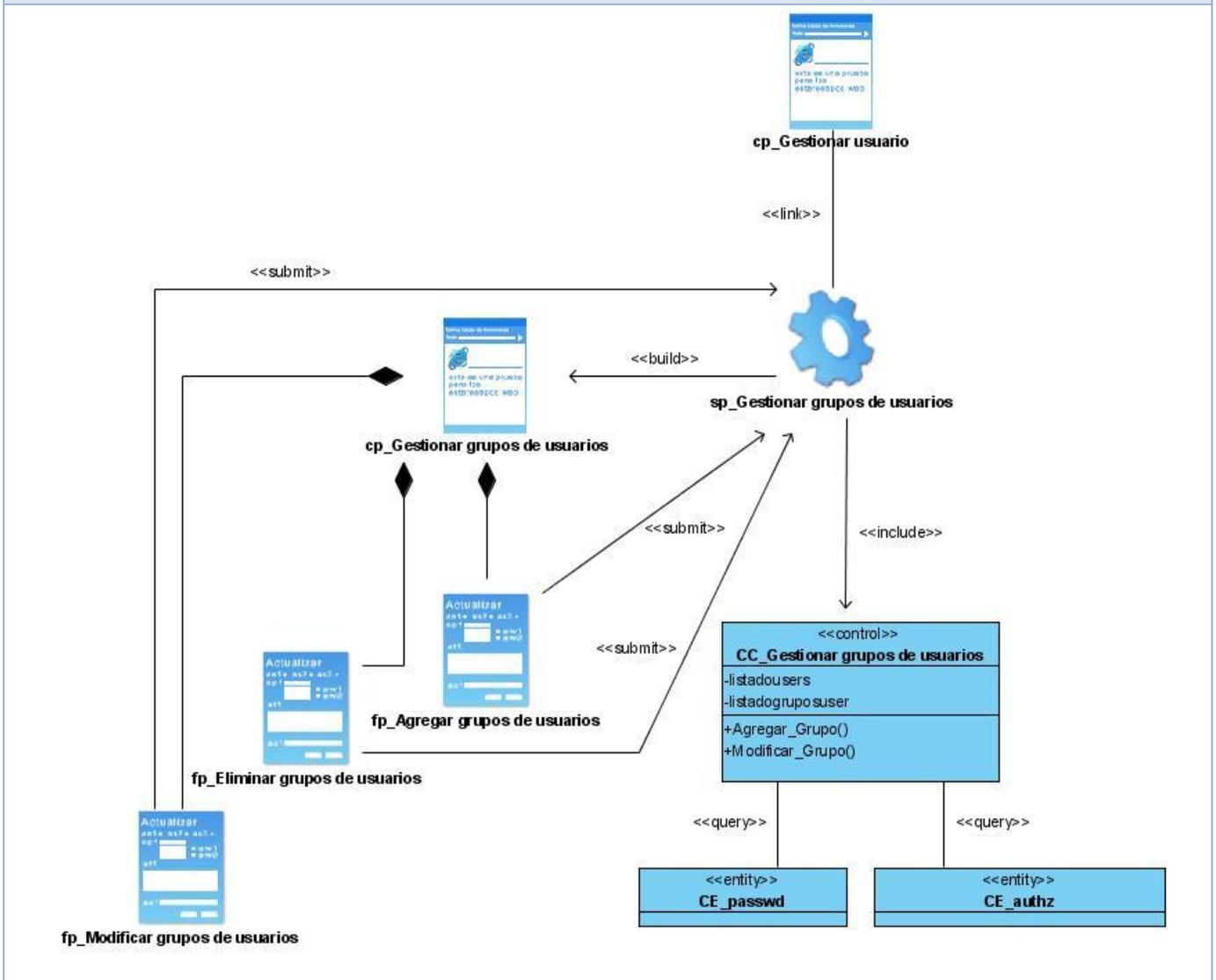


Tabla 3.32 Diagrama de clases del diseño CUS Gestionar grupos de usuarios.

Diagrama de clases del diseño CUS Gestionar permisos a usuarios

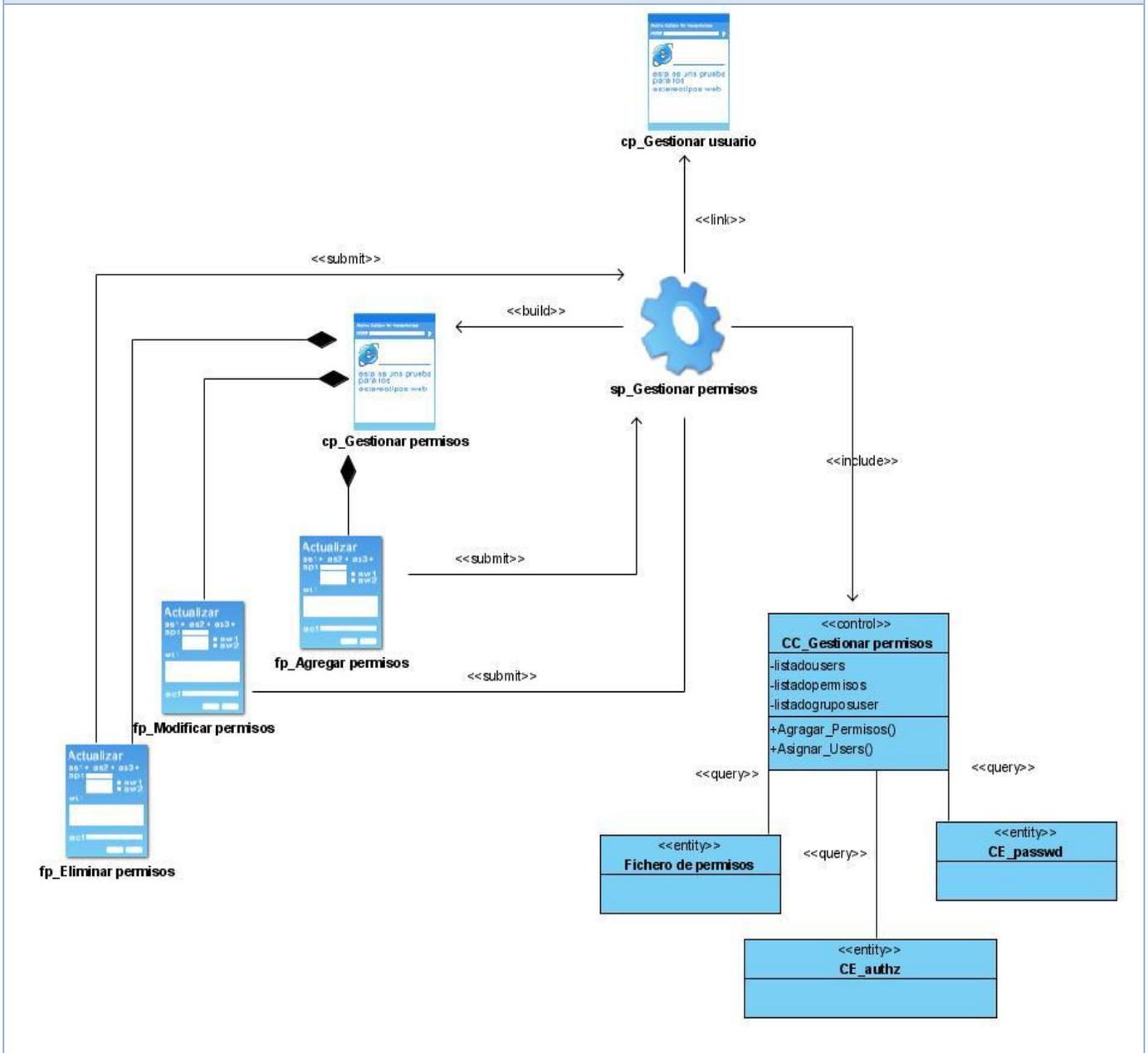


Tabla 3.33 Diagrama de clases del diseño CUS Gestionar permisos a usuarios.

Diagrama de clases del diseño CUS Gestionar permisos a grupos de usuarios

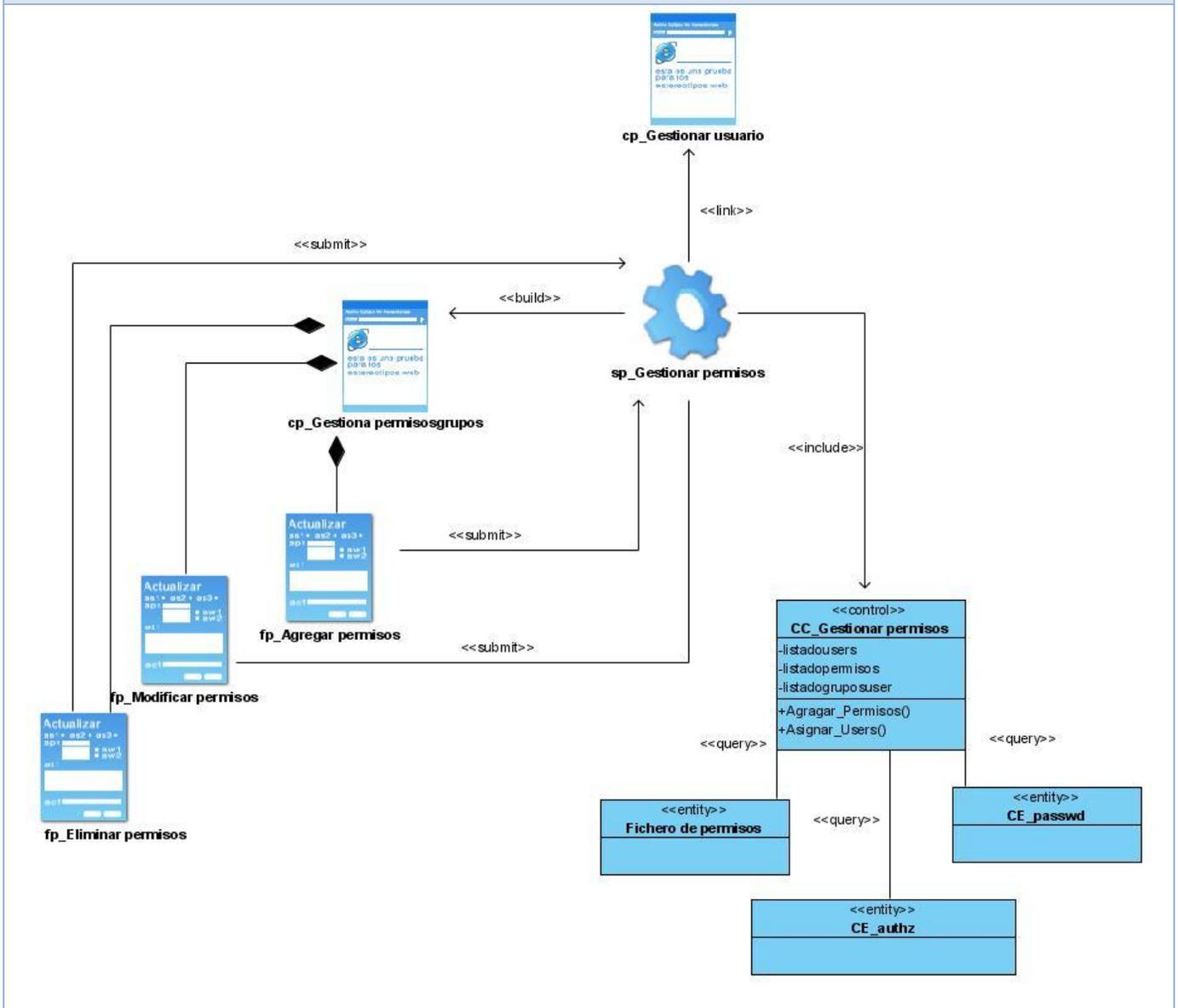


Tabla 3.34 Diagrama de clases del diseño CUS Gestionar permisos a grupos de usuarios.

Conclusiones

Con la realización del flujo Análisis y Diseño, se conforma la infraestructura adecuada para comenzar el flujo de Implementación pues se obtuvieron el Modelo de Análisis y el Modelo de Diseño. Dentro de estos modelos se representaron las relaciones que se establecen entre las clases interfaz, control y entidad, que permiten reflejar cada una de las funcionalidades que tiene el sistema, a través del Diagrama de Clases del Análisis. Además se detallaron los elementos del sistema a través de la descripción de las clases y sus relaciones, haciendo uso del Diagrama de Clases del Diseño. Es válido señalar que de la calidad y éxito del diseño presentado dependerá el éxito de la implementación de la herramienta.

Capítulo 4: IMPLEMENTACIÓN Y PRUEBA

Este capítulo constituye la continuación del proceso descrito en el flujo de trabajo Análisis y Diseño, en el cual se definieron mediante el uso de estereotipos web un grupo de clases y ficheros que serán tratados como componentes en el flujo de Implementación que será abordado en el presente capítulo.

Se presentan además Diagrama de Componentes y el Diagrama de Despliegue y un análisis de los principales estándares de diseño, codificación y tratamiento de excepciones.

4.1 Componente de Seguridad (SAAA)

Single Authentication Authorization and Account (SAAA) es un componente que tiene implementado un mecanismo de seguridad sustentado por el modelo de Autenticación, Autorización y Auditoría. En este modelo la autenticación es la primera acción que el usuario realiza en el sistema, este suministra un nombre de usuario único y una contraseña. Se verifica si el usuario está registrado, en caso negativo se reportará un error de acceso. En caso del que el usuario autenticado se encuentra registrado se autorizará su acceso al sistema, creándose además un certificado digital que consta de: usuario:rol:clave.

4.2 Implementación

El flujo de trabajo de Implementación comienza su desarrollo con el resultado del Diseño, implementando sus elementos en términos de componentes (ficheros de código fuente, scripts, ficheros de código binario, ejecutables y similares) y describiendo cómo estos componentes se organizan teniendo en cuenta los nodos específicos en el modelo de despliegue. Uno de los principales propósitos de este flujo de trabajo es desarrollar la arquitectura e integrar los resultados producidos en un sistema ejecutable.

4.2.1 Modelo de Implementación

El Modelo de Implementación constituye una colección de componentes y subsistemas de implementación que los contienen. Los componentes pueden ser: ficheros ejecutables, ficheros de código fuente y los necesarios que permitan la implantación y despliegue del sistema.

A continuación se muestra una vista global de la estructura y organización de la implementación del sistema desarrollado en el presente trabajo.

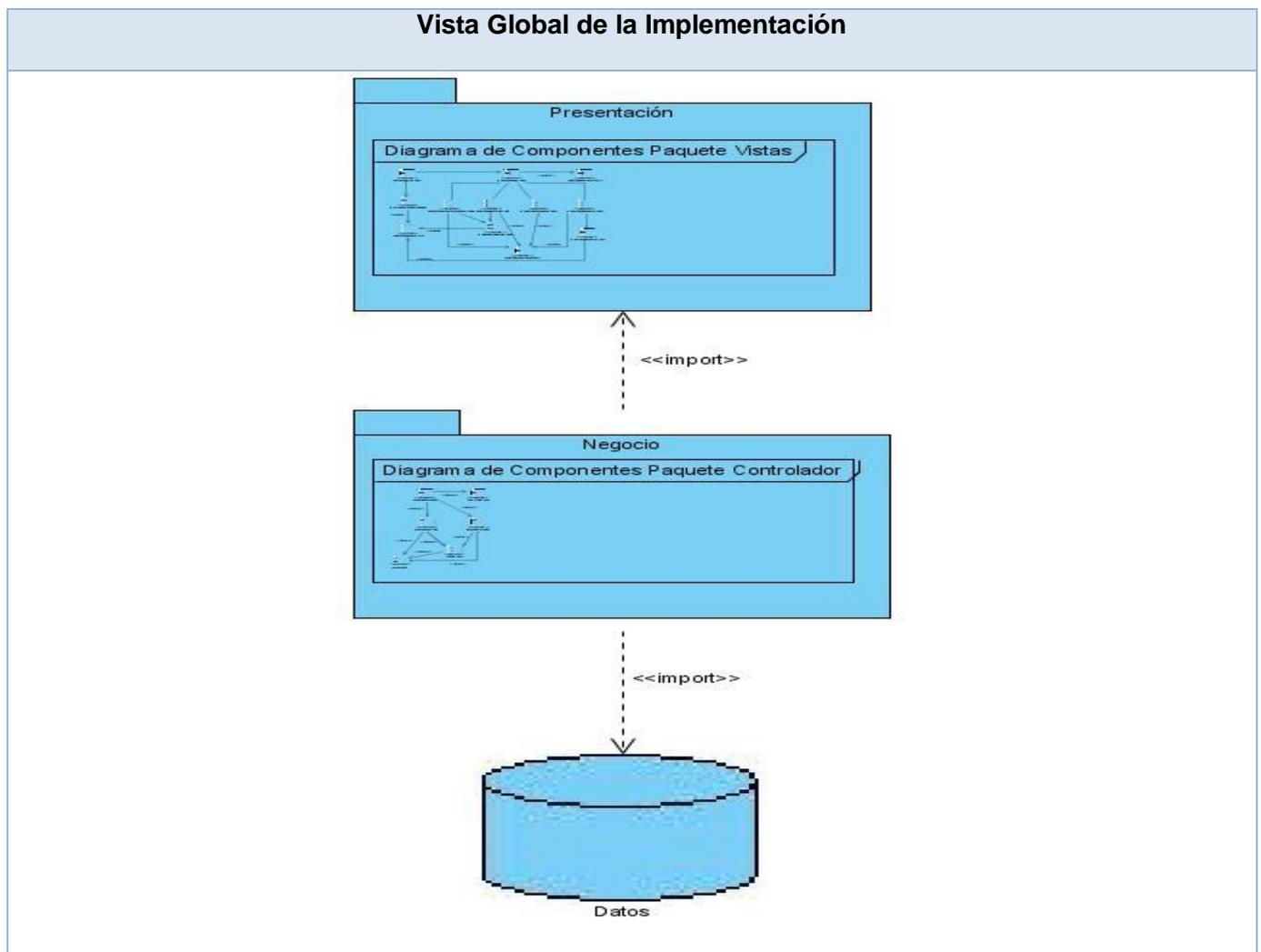


Tabla 4.1 Vista Global de la Implementación.

4.2.2 Diagrama de Componentes

Un **componente** representa una parte modular de un sistema, desplegable y reemplazable que encapsula implementación y un conjunto de interfaces y proporciona la realización de los mismos. Un componente generalmente contiene clases y puede ser implementado por uno o más artefactos (ficheros ejecutables, binarios, etc.).

Un **diagrama de componentes** es usado para estructurar el modelo de implementación en términos de subsistemas de implementación y que posibilita mostrar las relaciones entre los elementos de implementación. Permite modelar la gestión de la configuración de los archivos de código fuente y las versiones, así como las bibliotecas. El uso más importante que se le asigna a estos diagramas es mostrar los subsistemas de implementación y sus dependencias a la hora de importar código, así como organizar los subsistemas de implementación en capas. Estos diagramas son utilizados también para mostrar las dependencias de compilación de los ficheros de código relaciones de derivación entre ficheros de código fuente y ficheros que son resultados de la compilación, dependencias entre elementos de implementación y los correspondientes elementos de diseños que son implementados.

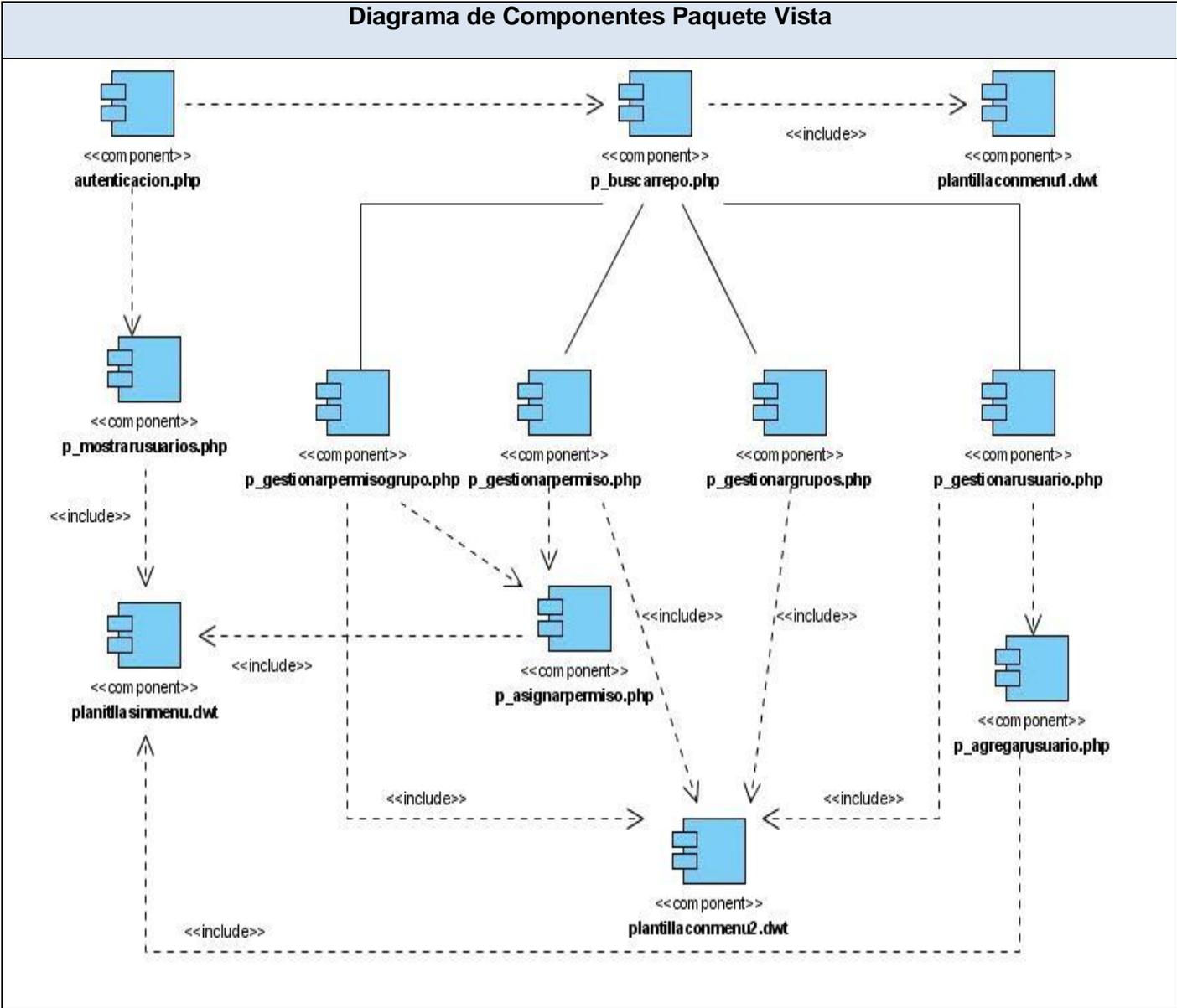


Tabla 4.2 Diagrama de Componentes Paquete Vista.

Diagrama de Componentes Paquete Controlador

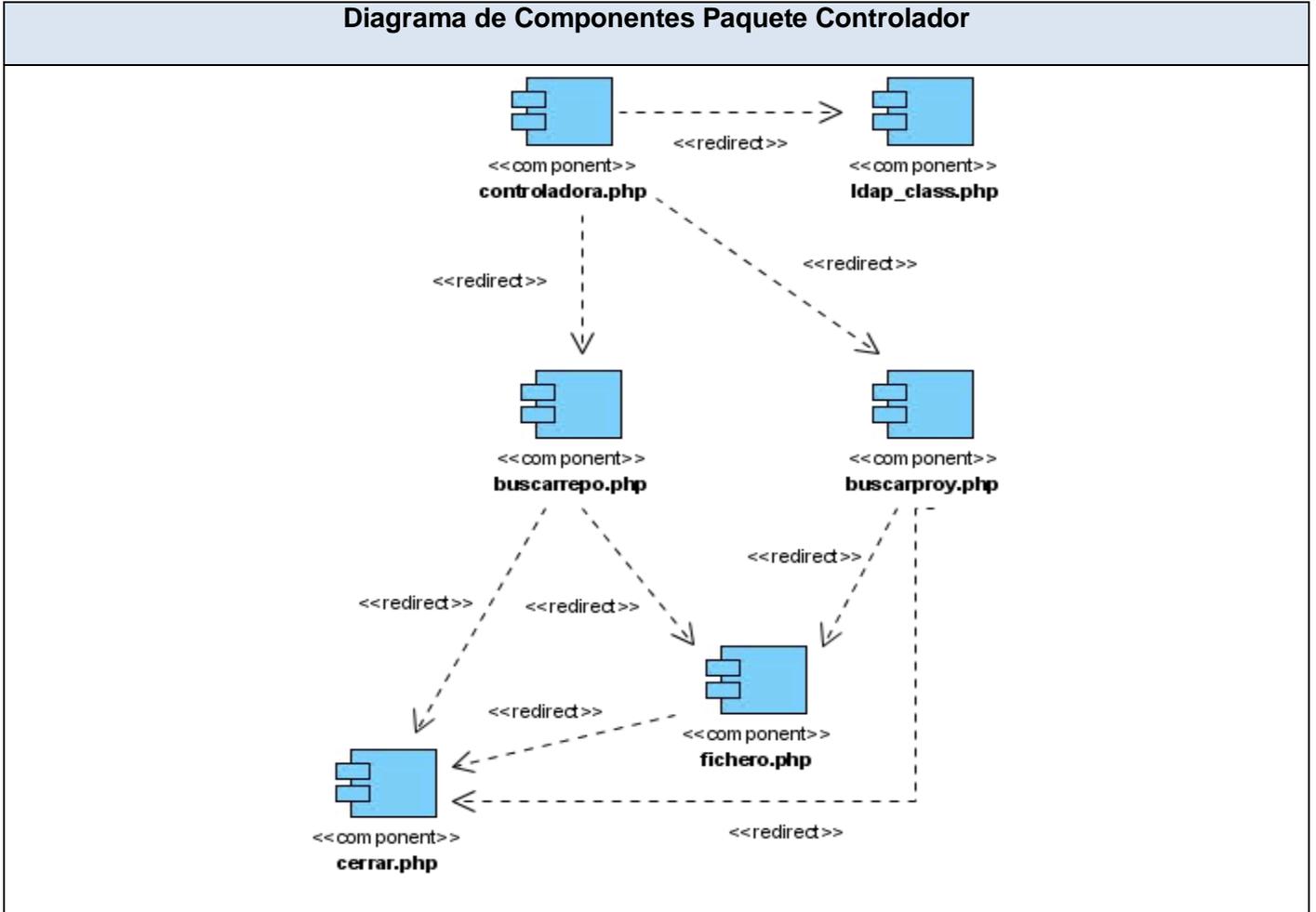


Tabla 4.3 Diagrama de Componentes Paquete Controlador.

Diagrama de Componentes Paquete Acceso a Datos

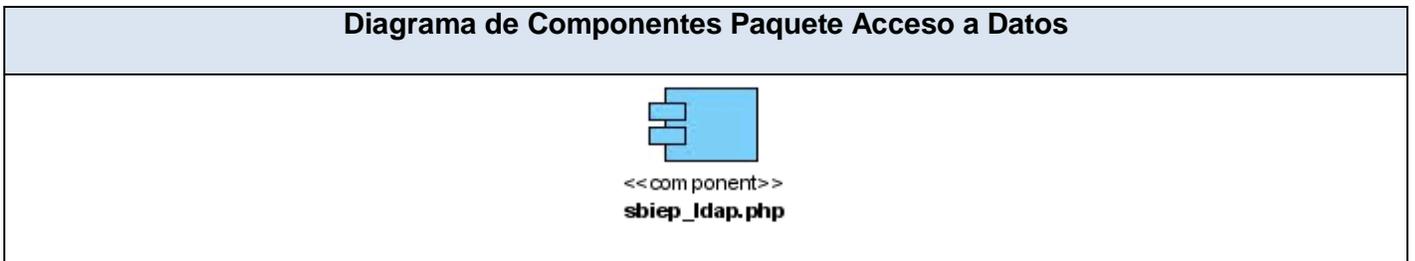


Tabla 4.4 Diagrama de Componentes Paquete Acceso a Datos.

4.2.3 Diagrama de Despliegue

Un diagrama de despliegue muestra las relaciones físicas entre los componentes hardware y software en el sistema final, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes software (procesos y objetos que se ejecutan en ellos). [12]

Este diagrama muestra cómo y dónde se desplegará el sistema. Las máquinas físicas y los procesadores se representan como nodos, y la construcción interna puede ser representada por nodos o artefactos embebidos. Como los artefactos se ubican en los nodos para modelar el despliegue del sistema, la ubicación es guiada por el uso de las especificaciones de despliegue. [13]

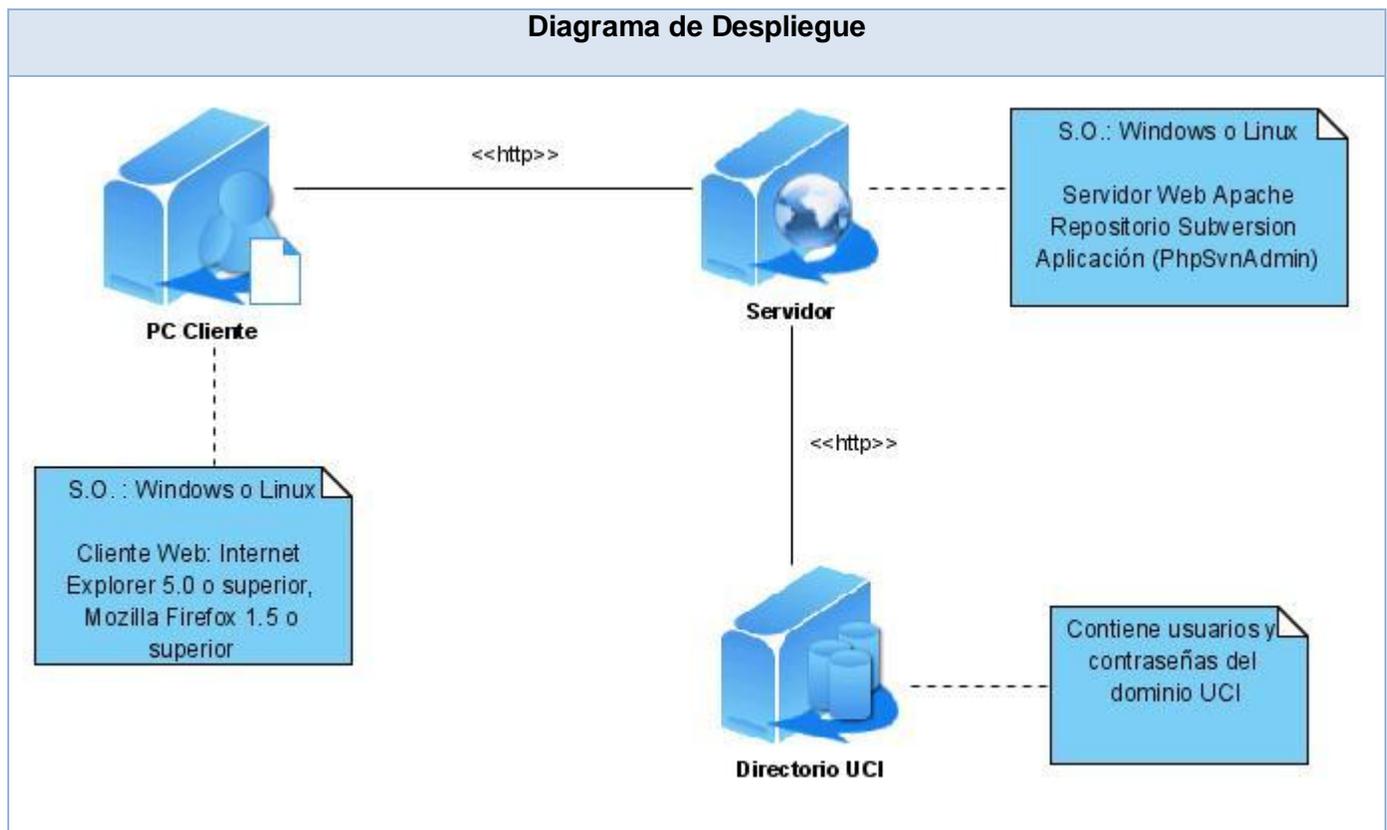


Tabla 4.5 Diagrama de Despliegue.

4.3 Estándares de Codificación

En la actualidad la mayoría de los lenguajes existentes poseen estándares de codificación. La importancia del uso de estándares, a partir de las convenciones definidas, posibilita un mejor intercambio y comunicación entre los programadores; de esta forma se generan las condiciones que garantizan la reusabilidad y el mantenimiento de los sistemas. Para establecer el estilo de codificación a utilizar en la herramienta se empleó la notación estándar establecida para aplicaciones desarrolladas en PHP (PHP Codig Standard), la cual se basa en gran parte en estándar de código para aplicaciones en C++ (C++ CODIG Standard).

- ✓ Aparecerá `<?php ?>` para identificar las etiquetas de apertura y cierre del lenguaje, pues estas siempre están disponibles en cualquier configuración.
- ✓ Para el manejo de los valores enviados por el usuario se hará uso de arreglos predefinidos: `$_GET`, `$_POST`, `$_FILES`, evitando la utilización de `$_REQUEST`.
- ✓ En la nomenclatura de las variables se escribirán los identificadores con letras minúsculas y en lenguaje español, usando el caracter “_” como separador de palabras. Se usarán nombres que sugieran la funcionalidad de la variable.
- ✓ Se declararán las constantes con todas sus letras en mayúscula.
- ✓ El nombre que se le da a los atributos de las clases debe comenzar con la primera letra en minúscula, la cual estará en correspondencia al tipo de dato al que se refiere.
- ✓ Los campos identificadores estarán estructurados con el identificador (id) y a continuación el nombre del campo. Ejemplo: `id_nombre`.
- ✓ Los arreglos estarán estructurados con el identificador array y sin separar las palabras con el caracter “_”. Ejemplo: `arraylistadonomb`.
- ✓ Las estructuras serán identificadas colocando el nombre struct al final. Ejemplo: `paginadostruct`.
- ✓ Para identificar las clases se colocará la letra C delante. Ejemplo: `CFichero`.
- ✓ Para las clases auxiliares como sesión y error, se utilizará el idioma inglés garantizando la homogeneidad con las programadas en este mismo ambiente a nivel mundial. Para los Servicios Web y

la interface de administración se usará el idioma español para brindar claridad a los objetivos de cada método o script a utilizar.

- ✓ Para los métodos se utilizará letra inicial mayúscula y las palabras continuas deben comenzar con mayúscula. Ejemplo: GestionarGruposUsuarios.
- ✓ Se empleará para comentar el código, en caso de una línea, al final de esta el carácter " // " y a continuación el comentario; para el caso de un bloque se usarán los caracteres " /* */".
- ✓ En las páginas HTML no se incluirá código y las funciones JavaScript usadas serán escritas dentro de ficheros ".js".
- ✓ Entre los bloques se dejarán dos espacios en blanco desde la instrucción anterior para el inicio y fin de bloque {}. Lo mismo sucederá para el caso de las instrucciones if, else, for, while, do while, switch, foreach.
- ✓ Los comentarios se situaran al inicio de la clases o funciones especificando el objetivo de la misma así como los parámetros que usa (especificar tipos de dato, y objetivo del parámetro).
- ✓ Se usarán espacios en blanco entre operadores para lograr una mayor legibilidad en el código. Ejemplo: producto = nomproducto.
- ✓ Para los controles se seguirá la siguiente línea (Estándares para los controles).

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

Grid o rejilla	Grid	grUsuario
Casilla de Selección	cbx	cbxSexo

Tabla 4.6 Estándares para los controles.

4.4 Tratamiento de Errores

Desde el surgimiento de los lenguajes de programación, el tratamiento de errores ha sido un tema difícil de manejar. En ocasiones algunos lenguajes ignoran el tratamiento de errores a causa de su complejidad, pero lo innegable es que este constituye un paso indispensable para lograr el buen funcionamiento de un sistema. Por esta razón desde el inicio del proceso de desarrollo del mismo, se llevan a cabo acciones y se cumplen tareas que permitan evitar la ocurrencia de errores.

Durante la ejecución del programa pueden ocurrir eventos que interrumpen el flujo normal de las sentencias, estos son denominados excepciones. Constituyen una manera clara de controlar los errores, evitando que se confunda el código con un gran número de instrucciones de control de error. De esta forma cuando se verifica un error, se acciona una excepción que permite gestionar un error.

En el transcurso de la ejecución del programa, en las clases pueden aparecer errores de diferente género y que varían en el grado de gravedad. Al invocarse métodos sobre un objeto, pueden aparecer problemas internos de estado (valores que no coinciden), detectarse errores con objetos o datos manipulados (dirección de red o archivo), imposibilidad de acceder a archivos que se encuentran cerrados, u otros problemas. Ante este tipo de situaciones las excepciones proporcionan una manera de verificar estos errores y controlarlos, en inclusive, abortar el código si fuera necesario.

Para alcanzar una programación de excepciones clara y correcta, es necesario que los algoritmos sean diseñados pensando en la forma habitual en que serán ejecutados, manejando las situaciones de carácter extraordinario a parte. De esta manera se obtendrá un diseño con una estructura clara, legible, robusta y mucho más fácil de mantener. Los errores que pueden ser evitados en la capa de presentación, serán tratados mediante funciones del lenguaje JavaScript del lado del cliente, a través de mensajes de alerta.

4.5 Prueba

En el desarrollo del software es común que aparezcan errores desde que comienza el proceso, es por ello que éste debe ir acompañado de una actividad que garantice la calidad. El desarrollo de pruebas constituye una actividad donde se realiza la ejecución, de un sistema o componente, bajo condiciones o requerimientos especificados, los resultados son observados y registrados, y se emite una evaluación de algún aspecto del elemento probado. La prueba de software representa una revisión de las especificaciones del diseño y la codificación, así como se ha establecido como un elemento crítico para garantizar la calidad del producto.

El Flujo de Trabajo de Prueba tiene entre sus principales objetivos encontrar y documentar los defectos que puedan afectar la calidad del software, validar que este trabaje como fue diseñado; así como validar y probar los requisitos que debe cumplir el sistema y que estos fueron implementados correctamente. Los métodos de prueba principales se clasifican en Métodos de Prueba basados en Caja Blanca y Métodos de Prueba basados en Caja Negra.

En la actividad Planificar Pruebas se obtuvo el artefacto Plan de Pruebas, definiéndose las pruebas que iban a ser diseñadas u ejecutadas en cada fase del desarrollo. En la actividad Diseñar Pruebas se realizó el diseño de pruebas de caja negra, teniendo en cuenta los requerimientos funcionales y su descripción plasmada en el Modelo del Sistema, de esta forma se logra la estructura correcta y completa del diseño de casos de prueba.

En el desarrollo del Flujo de Trabajo de Pruebas se verificó además, el resultado de la implementación, probando cada construcción, tanto internas e intermedias, así como las versiones finales del sistema a ser entregadas a terceros.

Resultados de las pruebas

Las pruebas diseñadas se realizaron a lo largo de todo el ciclo de desarrollo de la herramienta. Como resultado, se obtuvo una aplicación con alta funcionalidad y 100% de cumplimiento de los objetivos trazados.

La ejecución de estas pruebas contribuyó a facilitar el desempeño de los desarrolladores, sirviéndoles de apoyo para realizar su trabajo y localizar los errores que a simple vista no son detectados cuando se está

programando. Esto permitió que fueran corregidos mucho antes de dar los toques finales a la herramienta, posibilitando que se hiciera una mejor distribución del tiempo de trabajo.

De esta forma se obtiene una herramienta cuya funcionalidad ha sido comprobada con anterioridad antes de ponerse en manos de los usuarios finales, lo que garantiza la confiabilidad en la aplicación y la obtención de resultados satisfactorios durante su uso.

Conclusiones

En el desarrollo del capítulo se especificaron los resultados de la Fase de Construcción del Sistema. Durante la implementación se cumplieron los principios de diseño establecidos, así como los estándares de interfaz e implementación determinados. Se presentó también el Modelo de Implementación, donde fue descrita la distribución física del sistema y sus componentes. Finalmente se realizaron pruebas al sistema permitiendo su evaluación y valorando su calidad.

CONCLUSIONES

En el desarrollo del presente trabajo de diploma se obtuvo como resultado la creación de una herramienta web capaz de gestionar la administración de un sistema de control de versiones Subversion. Como parte de la presente investigación se concluyó:

- ✓ Tras un estudio de las aplicaciones informáticas en el ámbito nacional e internacional que pudieran resolver el problema planteado, se concluyó que existen algunos sistemas que permiten la administración de un repositorio Subversion, sin embargo ninguno posee la característica de ser una herramienta web, multiplataforma, software libre, flexible y fácil de configurar.
- ✓ El análisis de múltiples herramientas que pudieran utilizarse para dar cumplimiento el objetivo planteado, permitió seleccionar las que cumplieran con la característica de ser libres y multiplataforma, siguiendo la política del país de desarrollar sistemas informáticos con soberanía tecnológica.
- ✓ Fueron generados los artefactos relacionados con el análisis y diseño, según lo establecido por la metodología de desarrollo RUP, los que constituyeron la base para la implementación exitosa del sistema.
- ✓ Tras el desarrollo de la investigación se obtuvo la herramienta web PhpSvnAdmin, que permite gestionar eficientemente la administración de un repositorio Subversion.

RECOMENDACIONES

Luego de concluir el presente trabajo y después de haber cumplido los objetivos propuestos, se recomienda:

- ✓ Desplegar la herramienta web PhpSvnAdmin en los proyectos productivos para que sus servicios puedan ser usados por quien los necesite.
- ✓ Realizar nuevas versiones de la aplicación, donde se agreguen otras funcionalidades para el control de las versiones, además de las existentes que la hagan más práctica y eficiente, así como flexible y fácil de configurar.
- ✓ Realizar estudios de nuevas tecnologías y herramientas que sean adaptables al sistema y con las cuales se le pueda agregar mayor potencialidad.
- ✓ Garantizar que se realice la capacitación adecuada, lo que permita el despliegue del sistema al máximo de sus capacidades.

REFERENCIA BIBLIOGRÁFICA

1. **García, Luis.** Sistema de control de versiones: Subversion. *Educación. Observatorio Tecnológico.* [En línea] 17 de enero de 2008.
<http://observatorio.cnice.mec.es/modules.php?op=modload&name=News&file=article&sid=548>. [1].
2. ANÁLISIS DE DATOS MEDIOAMBIENTALES MEDIANTE. *Ingeniería en Informática – Universidad de Burgos.* [En línea]
<http://74.125.47.132/search?q=cache:l2R3ruIR0gJ:neuroralanalysis.googlecode.com/svn/trunk/Documentacion/Anexo%2520VII-2520Proceso%2520de%2520Investigaci%C3%B3n%2520y%2520Desarrollo.doc+sis+tema+de+control+de+versiones+debe+proporcionar&cd=23&hl=es&ct=cln>. [2].
3. CVS (Concurrent Versions System). *NIS.* [En línea] <http://www.i-nis.com.ar/tutoriales/cvs>. [3].
4. Configuración de software. *El Rincón del Vago.* [En línea] <http://html.rincondelvago.com/configuracion-de-software.html>. [4].
5. **James Avery.** Cuadro de herramientas. Integración de Visual Studio y Subversion. *MSDN Magazine.* [En línea] <http://msdn.microsoft.com/es-es/magazine/cc164246.aspx>. [5].
6. *WebSVN - Online subversion repository browser.* [En línea] <http://www.websvn.info/>. [6].
7. Free Subversion Tool – WebClient for SVN. *Polarion Community for Subversion & Subversive.* [En línea] http://www.polarion.com/products/svn/svn_webclient.php. [7].
8. El Modelo Cliente/Servidor. [En línea] <http://agamenon.uniandes.edu.co/~revista/articulos/cliser.html>. [8].
9. *Arquitectura Cliente/Servidor.* [En línea] <http://www.csi.map.es/csi/silice/Global71.html>. [9].

10. Bases de la programación para el Bash. [En línea] <http://xinfo.sourceforge.net/documentos/bash-scripting/bash-script-2.0.html>. [10].
11. Bash. [En línea] <http://www.gnu.org/software/bash/bash.html>. [11].
12. **Ana Fernandez Vilas**. Diagrama de Despliegue. [En línea] 20 de 03 de 2001. <http://tvdi.det.uvigo.es/~avilas/UML/node50.html>. [12].
13. Diagrama de Despliegue. *Guía de Usuario de Enterprise Architect 6.5*. [En línea] <http://www.sparxsystems.com.ar/EASystemGuide/ea.html?deploymentdiagram.htm>. [13].

BIBLIOGRAFÍA

Ana Fernandez Vilas. Diagrama de Despliegue. [En línea] 20 de 03 de 2001.
<http://tvdi.det.uvigo.es/~avilas/UML/node50.html>.

ANÁLISIS DE DATOS MEDIOAMBIENTALES MEDIANTE. *Ingeniería en Informática – Universidad de Burgos*. [En línea]
<http://74.125.47.132/search?q=cache:l2R3ruIR0gJ:neunoralanalysis.googlecode.com/svn/trunk/Documentacion/Anexo%2520VII-2520Proceso%2520de%2520Investigaci%C3%B3n%2520y%2520Desarrollo.doc+sistema+de+control+de+versiones+debe+proporcionar&cd=23&hl=es&ct=cln>.

Arquitectura Cliente/Servidor. [En línea] <http://www.csi.map.es/csi/silice/Global71.html>.

Bases de la programación para el Bash. [En línea] <http://xinfo.sourceforge.net/documentos/bash-scripting/bash-script-2.0.html>.

Bash. [En línea] <http://www.gnu.org/software/bash/bash.html>.

Cano, Fernando P. Najera. TortoiseSVN. Un cliente de Subversion para Windows. Versión 1.6.1. [En línea] 04 de 04 de 2009. http://tortoisesvn.net/docs/release/TortoiseSVN_es/help-onepage.html#tsvn-preface.

Configuración de software. *El Rincón del Vago*. [En línea] <http://html.rincondelvago.com/configuracion-de-software.html>.

CVS (Concurrent Versions System). *NIS*. [En línea] <http://www.i-nis.com.ar/tutoriales/cvs>.

Diagrama de Despliegue. *Guía de Usuario de Enterprise Architect 6.5*. [En línea]
<http://www.sparxsystems.com.ar/EAUserGuide/ea.html?deploymentdiagram.htm>.

EKKA. Tecnología a su alcance. *Aplicaciones Web*. [En línea] <http://www.ekka.cl/aplicaciones.htm..>

El Modelo Cliente/Servidor. [En línea] <http://agamenon.uniandes.edu.co/~revista/articulos/cliser.html>.

Free Subversion Tool – WebClient for SVN. *Polarion Community for Subversion & Subversive*. [En línea]
http://www.polarion.com/products/svn/svn_webclient.php.

Galiano, Julio Antequera. *Manual Subversion* . 16-7-08.

García, Luis. Sistema de control de versiones: Subversion. *Educación. Observatorio Tecnológico*. [En línea] 17 de enero de 2008.
<http://observatorio.cnice.mec.es/modules.php?op=modload&name=News&file=article&sid=548>.

González, Ing. Guillermo. Sistema de Control de Versiones Caso de estudio: Subversion.

Introducción a los sistemas de control de versiones. [En línea]
<http://www.lug.fi.uba.ar/documentos/scms/index.php#prologo>.

James Avery. Cuadro de herramientas. Integración de Visual Studio y Subversion. *MSDN Magazine*. [En línea] <http://msdn.microsoft.com/es-es/magazine/cc164246.aspx>.

JAMES RUMBAUGH, G. B., IVAR JACOBSONI. *El Proceso Unificado de Desarrollo de Software*. La Habana : Editorial Félix Varela, 2004.

León, Eduardo. Tutorial Visual Paradigm for UML. [En línea] <http://www.slion2000.blogspot.com>.

Mayoral, Antonio Gutiérrez. Front-end y biblioteca de interacción con Subversion para GNOME con tecnologías C# y Mono. Proyecto Fin de Carrera. Febrero de 2005.

Pereira, Federico. Subversion: uso desde el cliente. [En línea]
http://megalinux.com.ar/cliente:subversion:cliente_svn#subversionuso_desde_el_cliente.

PRESMAN, R. S. *Ingeniería del software. Un Enfoque práctico*. La Habana : Editorial Félix Varela, 2005.

RODAS HINOSTROZA, R. *Características de PHP*. [En línea]
<http://www.linuxcentro.net/linux/staticpages/index.php?page=CaracteristicasPHP>.

Sierra, María. INGENIERÍA DEL SOFTWARE I Práctica 1 Trabajando con Visual Paradigm for UML.

Teleformacion.uci.cu. *Conferencia Ingeniería de Software I curso 2008-2009*. [En línea]

WebSVN - Online subversion repository browser. [En línea] <http://www.websvn.info/>.

ANEXOS

Anexo I

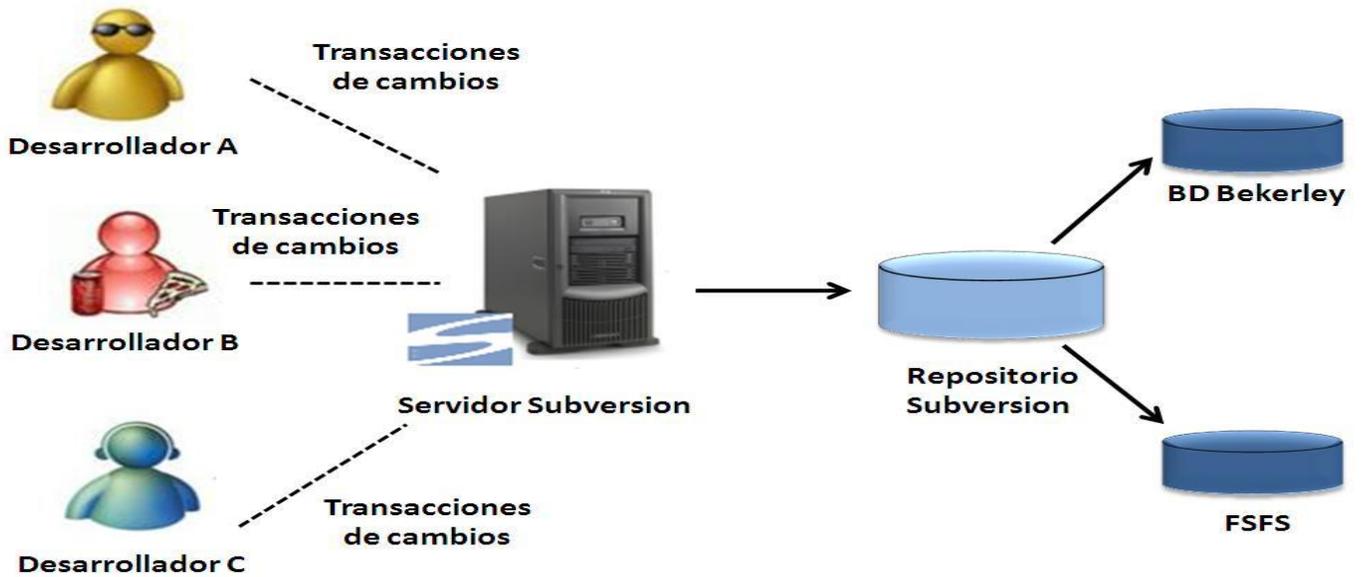


Anexo II

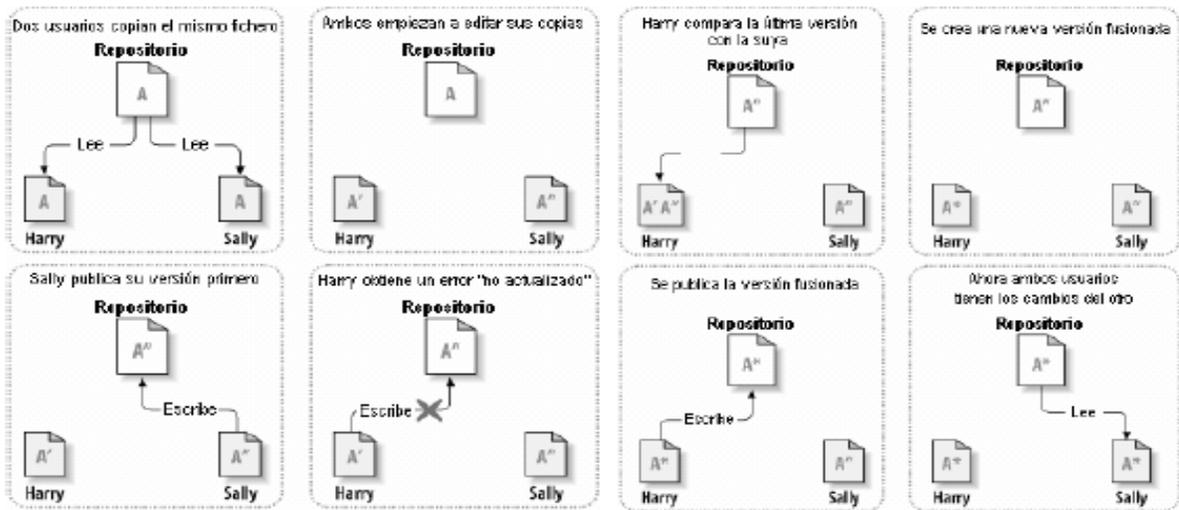


Anexo III

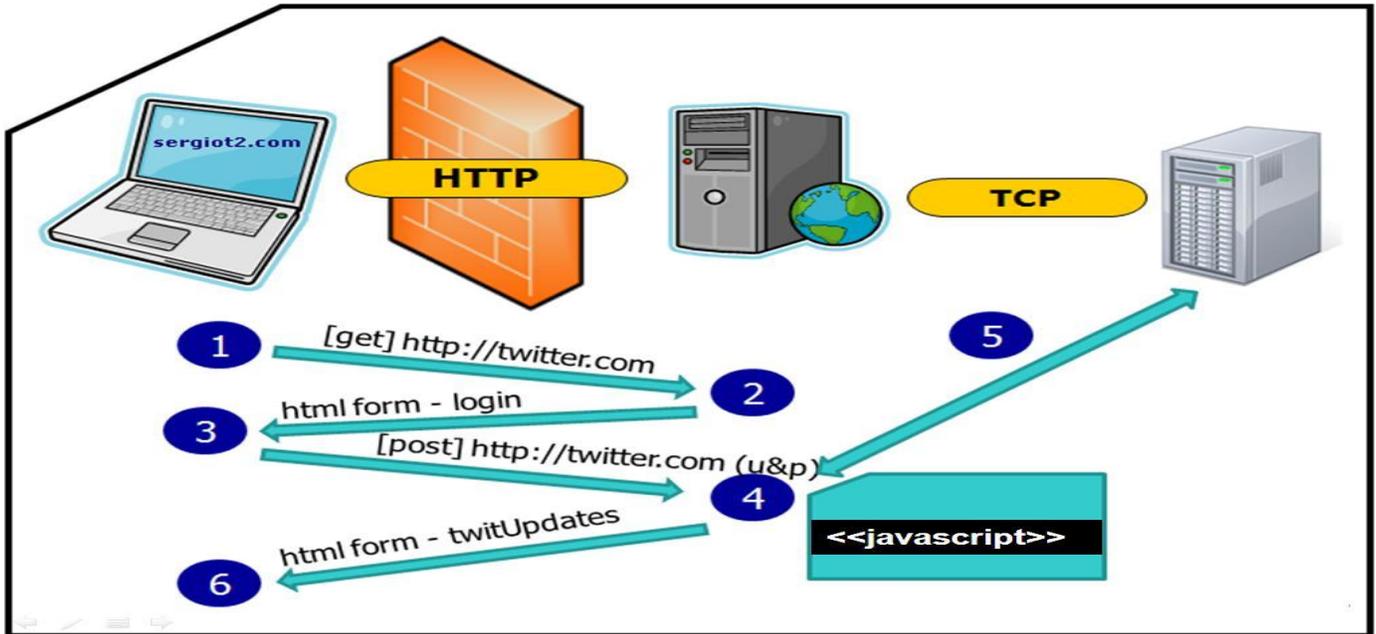
Subversion aumenta la funcionalidad párrafo 2



Anexo IV



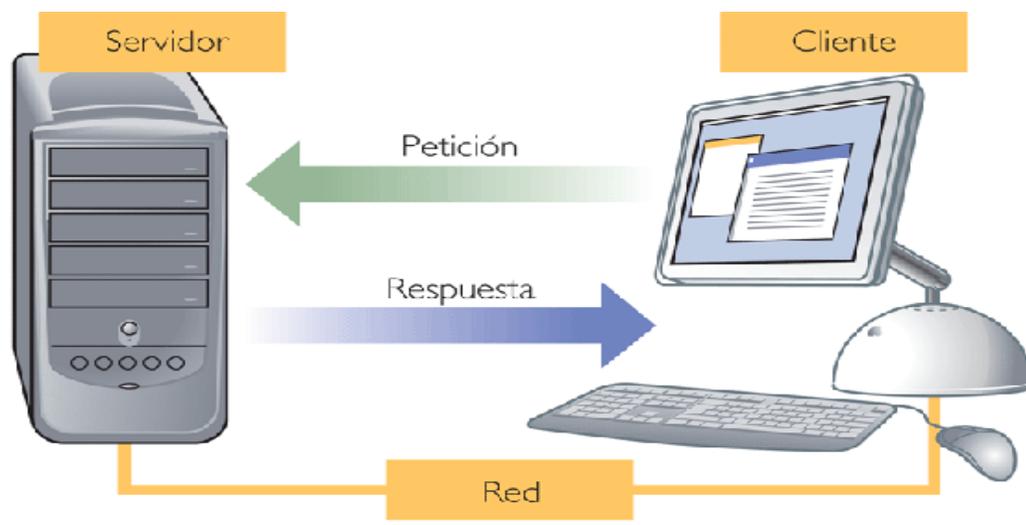
Anexo V



Anexo VI



Anexo VII



GLOSARIO DE TÉRMINOS

API (Interfaz de Programación de Aplicaciones): es un conjunto de funciones residentes en bibliotecas que permiten que una aplicación corra bajo un determinado sistema operativo.

Base de datos Berkeley: es una base de datos embebida con API para C, C++, Java, Perl, Python, Ruby, TCL y muchos otros lenguajes. Permite miles de hilos de control manipulando bases de datos de hasta 256 terabytes en muchos sistemas, incluidos la mayoría de los *tipo-UNIX* y Windows, e incluso sistema operativos de tiempo real. Es usada por Subversion.

CVS (Sistema de Control de Versiones Concurrentes): es una herramienta que mantiene el registro de todo el trabajo y los cambios en la implementación de un proyecto informático (de software) y permite que distintos desarrolladores (potencialmente situados a gran distancia) colaboren en el mismo.

HTML (Lenguaje de Marcas de Hipertexto): es el lenguaje de marcas de texto utilizado normalmente en la *www* (World Wide Web). Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes.

MD5 (Algoritmo de Resumen del Mensaje 5): constituye un algoritmo de reducción criptográfico de 128 bits ampliamente usado ya que es una función hash irreversible, es decir, encripta el password tecleado por el usuario y es imposible que partiendo desde la cadena encriptada se vuelva a la contraseña origen.

Protocolo WebDAV (Protocolo Web Distribuido para la Autorización y Versionado): es una extensión sobre HTTP 1.1. Este protocolo coge el ubicuo protocolo HTTP, núcleo de la World Wide Web, y añade la capacidad de escritura— específicamente el versionado de la misma.

RCS (Sistema de Control de Revisiones): software que automatiza el almacenamiento, recuperación, identificación y fusión de las revisiones. RCS es útil para el texto que se revisa con frecuencia, por ejemplo: programas, documentación, gráficos, documentos y otros.

Repositorio: es un sitio centralizado donde se almacena y mantiene información digital, habitualmente bases de datos o archivos informáticos.

RSS: es una familia de formatos de fuentes web codificados en XML. Se utiliza para suministrar a suscriptores de información actualizada frecuentemente.

RUP (Proceso de Desarrollo Unificado): proceso de desarrollo de software, que constituye la metodología más usada para el análisis, implementación y documentación de sistemas con tecnología orientada a objetos.

SCV (Sistemas de Control de Versiones): es un sistema de gestión de archivos y directorios, que mantiene la historia de los cambios y modificaciones que se han realizado sobre ellos a lo largo del tiempo.

SVN (Subversion): es un programa controlador de versiones empleado en la administración de archivos utilizados en el desarrollo de software o contenido.

Transacción: Una transacción atómica es una operación en la que un conjunto de distintos cambios es aplicado como una operación única. Si todos los cambios son aplicados a la transacción habrán tenido éxito. Si ocurre alguna falla antes de ser completada la operación, se aborta la misma y todos los cambios que han tenido lugar se invierten.

UML (Lenguaje Unificado de Modelado): Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema.

wxWidgets: plataforma especializada en el desarrollo de aplicaciones multiplataforma en lenguaje C++, que soporta varios sistemas operativos, es libre y distribuido bajo licencia LGPL modificada.