UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD X

Título:

Arquitectura, Análisis y Diseño, del Sistema de Autentificación Única de RINDE.

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

Autor:

Alexander Reyes Grillo

Tutor:

Ing. Eliurkis Díaz Terrero

Ciudad de La Habana, Junio de 2008

DECLARACIÓN DE AUTORÍA.

Por este medio se declara que soy él único autor de este trabajo y se autoriza a la Universidad de las Ciencias Informáticas para que haga el uso que estime pertinente del mismo.

Para que así conste firmo la presente a los___días del mes___del año 2008.

_____ _____

Firma del Autor Firma del Tutor

AGRADECIMIENTOS.

*A mis **padres** por ser las personas que más quiero y a quienes les debo lo que soy, por haber creído en mí y brindarme todo su apoyo.*

*A mi **abuela "nena"** por todos estos años de crianza y ternura.*

*A mi **novia**, Yenisleydis Guillén Estupiñán, por todo el amor, apoyo e inspiración que me brinda.*

*A toda mi **familia, amigos y compañeros** que de una forma u otra han contribuido a mí formación como profesional.*

"… lo que da al hombre el poder no es el mero conocimiento que viene del uso de los sentidos, sino, ese otro conocimiento más profundo que se llama Ciencia."

José Martí.

DEDICATORIA.

*A la memoria de mi **abuelo**.*

*A mi **familia**, por el apoyo constante.*

*A mi **madre** que me enseñó a luchar siempre por lo que se quiere.*

*A mi **padre**, mi fuente de inspiración constante.*

*A mi **novia**, Yeny, por todo el amor y apoyo que me brinda.*

*A mis **primos-hermanos** Giselle y Tavy.*

*A mi **abuela** "Lili" y tía "Reina" de su pichoncito.*

*A mi **tío** por el apoyo brindado.*

*A mis **amigos** de los buenos y malos momentos.*

*A **todos** los que en algún momento contribuyeron a formarme como profesional y persona en este mundo.*

RESUMEN.

El proyecto llamado *Red de Integración y Desarrollo Nacional de Software Libre de la República Bolivariana de Venezuela* (RINDE) surge para la unificación del movimiento de Software Libre en la República de Venezuela. Este trabajo expone el modelo realizado para el mismo como la solución más óptima, a través de la creación de un *Single Sign-On* (SSO) que logra la integración de los tres subsistemas Web que lo conforman, eliminando redundancia de datos y garantizando un mayor grado de seguridad.

Para ello se realizó un estudio de las principales arquitecturas de SSO usadas a nivel mundial, se analizaron las características y exigencias del cliente y se conformó la arquitectura, análisis y diseño, del modelo acorde a los resultados del estudio preliminar realizado.

El producto obtenido ha sido bien aceptado por los clientes, avalado por el alto grado de aceptación que ha tenido el proyecto de forma general hasta el momento.

ÍNDICE.

INTRODUCCIÓN.

El gran desarrollo de la industria del software ha traído consigo el avance de la informática a nivel mundial, convirtiéndose esta en una de las más importantes ciencias de las últimas décadas.

Con el uso de Internet numerosas empresas obtienen extraordinarias ganancias. Muchas de ellas poseen sitios Web donde ofertan sus servicios y productos y, a la vez, hacen referencia a otros sitios donde se promocionan o venden otros productos.
En no pocas ocasiones, estos grandes sistemas de ofertas de productos, etc. son conformados de forma netamente independiente.
Con el desarrollo de sistemas Web hoy en día se integran múltiples servicios.

En el caso del sistema creado por RINDE para el Desarrollo Colaborativo de la Red Nacional de Software Libre de la Republica Bolivariana de Venezuela fue necesario integrar varios subsistemas en uno mismo, con el objetivo de evitar información redundante y tener una mayor seguridad y control de la información por parte de los administradores.
La necesidad de integrar algunos servicios, como la autentificación del usuario en el sistema, para poseer un único lugar donde almacenar y gestionar los datos de los usuarios inscritos en el mismo, constituye un **problema a resolver.**

De lo anteriormente expuesto surge la necesidad de dar respuesta al siguiente **problema científico**: ¿Cómo mantener de forma centralizada el proceso de autentificación en un gran sistema Web mediante un Single Sign-On[I] (SSO)?

La integración de servicios en sistemas Web constituye el **objeto de estudio** de este trabajo, limitando así el **campo de acción** del mismo a la integración del proceso de autentificación en todos los subsistemas creados en RINDE[II].
Para que la realización del trabajo tenga la calidad requerida se definieron muy precisamente sus objetivos de investigación, declarándose como **objetivo general** el estudio y exposición del modelo de arquitectura, análisis y diseño, utilizado por el proyecto RINDE en la creación del SSO.

[I] Sistema de Autentificación Única.
[II] Red de Integración y Desarrollo Nacional de Software Libre de la República Bolivariana de Venezuela.

Surgen entonces las siguientes **preguntas científicas**:

- ¿Qué es un sistema SSO?
- ¿Cuáles son las principales arquitecturas básicas usadas para la implementación de un SSO en la actualidad?
- ¿Cuáles son los principales servicios que brinda el SSO de RINDE?
- ¿Cuál es el modelo de arquitectura, análisis y diseño, usado en la confección del SSO de RINDE?

Para satisfacer eficazmente el objetivo general del trabajo, se declararon como **objetivos específicos** los que a continuación se relacionan:

- Definir que es un SSO.
- Investigar la forma de trabajo de un SSO.
- Definir las arquitecturas básicas usadas para la implementación de un SSO.
- Diseñar un modelo de arquitectura para el SSO.

Las **Tareas** generales desarrolladas para satisfacer los objetivos expuestos anteriormente son las siguientes:

- Realizar a través de búsquedas bibliográficas un estudio de las definiciones actuales de SSO y redactar la definición que mejor se adecua a nuestra investigación.
- Exponer las principales arquitecturas básicas utilizadas hoy en día para la implementación de un sistema SSO, destacando cuál de ellas fue la usada en RINDE y el por qué de su elección.
- Describir en detalle suficiente el funcionamiento de nuestro SSO, haciendo hincapié en sus características distintivas.
- Utilizando herramientas CASE (Visual Paradigm) lograr un modelo de arquitectura eficiente para nuestro sistema SSO.

Para dar cumplimiento a las distintas tareas, se pusieron en práctica los siguientes métodos:

Métodos Teóricos:

Analítico-Sintético: sirvió para realizar el procesamiento de toda la información, sintetizándola y diferenciándola para de esta forma enfocarla hacia el desarrollo.

Histórico-Lógico: posibilitó conocer los antecedentes y tendencias actuales de la implementación de sistemas de SSO.

Sistémico: facilitó la implementación de cada uno de los elementos desarrollados, en la conformación del sistema SSO.

Métodos Empíricos:

Medición: posibilitó establecer las comparaciones entre las arquitecturas básicas para implementar sistemas de SSO en cuanto a características, ventajas y desventajas que presentan.

Experimento: este método posibilitó ir probando las distintas soluciones a los problemas que se iban presentando.

La memoria de la investigación contiene esta Introducción, tres capítulos, conclusiones, recomendaciones, referencias bibliográficas, bibliografía, glosario de términos y cuatro anexos, en un volumen de ciento veintidós páginas.

En el **Capítulo 1, *Fundamentación Teórica,*** se aborda el estado del arte de los Sistemas de Autentificación Única (SSO), su organización con respecto a los sistemas descentralizados y arquitecturas básicas existentes, haciendo una explicación detallada en cuanto a características, ventajas y desventajas que presentan. Al mismo tiempo se realiza un análisis de la arquitectura a usar para la implementación de nuestro SSO, definiendo el por qué de su elección.

En el ***Capítulo 2, Características del sistema***, se realiza un estudio y análisis de las causas por las cuales se decide adoptar la realización del SSO, mostrando así los beneficios que esto reportaría para el sistema existente en RINDE. Se muestran las especificaciones de los requerimientos de software, requerimientos funcionales y no funcionales, el modelo de dominio y de casos de uso, junto con la descripción detallada de cada caso de uso existente en la creación del sistema SSO.

En el **Capítulo 3, *Análisis y Diseño del sistema,*** en correspondencia con los aspectos tratados en el capítulo anterior (requerimientos, casos de uso, etc.) se elabora la concepción del sistema y se representan los diagramas de interacción, de clases del análisis y clases (Web) del diseño con todos sus elementos.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

Los sistemas de información corporativos incluyen un número creciente de aplicaciones heterogéneas (correo, Web, bases de datos, CRM) y recursos alojados en diversos plataformas (Windows, GNU/Linux, Mac-OS, Solaris, etc.). Mientras esta variedad y heterogeneidad facilita los procesos de negocios, también constituyen un problema desde el punto de vista de la gestión de la seguridad. Se plantea la dificultad de definir e implantar una política de seguridad única, que sea aplicable a todos esos recursos y aplicaciones.
Esta solución es la gestión centralizada y segura de usuarios y contraseñas para los distintos servicios de la organización mediante la implantación de un SSO.

1.1 - Generalidades:

Single Sign-On (SSO) es un procedimiento de autentificación que habilita al usuario para acceder a varios sistemas con una sola instancia de identificación; para ello supone efectuar, en lugar del usuario, la operación de identificarse mediante sus credenciales (identificador y contraseña) frente a las aplicaciones y recursos corporativos. Los usuarios se autentifican una única vez, contra el sistema de SSO, y después este se encarga, de forma transparente, de las autentificaciones subsiguientes en su lugar, según se van produciendo los accesos correspondientes.
El concepto de Single Sign-On (SSO) se refiere al acceso a múltiples recursos por medio de un único proceso de ingreso. Gran cantidad de las arquitecturas implementadas en diferentes organizaciones han sido diseñadas con el objetivo de dar acceso a los usuarios a múltiples servicios Web y/o aplicaciones. En la mayoría de los casos se encuentra que cada uno de los servicios o aplicaciones cuenta con su propio componente de seguridad, lo cual generalmente compromete la seguridad de todo el sistema, dado que el nivel de seguridad de todo un sistema es igual al nivel de seguridad del componente más inseguro (eslabón más débil) que lo compone. [1]

A menudo se ha optado por pensar que la implementación de un sistema SSO trae múltiples inconvenientes, como son: afrontar problemas de seguridad y tener que asumir altos costos de implementación y mantenimiento. Sin embargo las nuevas tecnologías desarrolladas, las

infraestructuras de las aplicaciones que son cada vez más robustas y la maduración de las diferentes técnicas de autentificación y seguridad, han permitido que se reconsidere la posibilidad de implementar el SSO como una estrategia para fortalecer la seguridad informática de las organizaciones.

Hay cinco tipos principales de SSO, también se les llama “Reduced Sign On Systems” (en español, Sistemas de Autentificación Reducida).

- ***Enterprise Single Sign-On (E-SSO):*** también llamado”Legacy Single Sign-On”, funciona luego de una autentificación primaria, interceptando los requerimientos de login presentados por las aplicaciones secundarias para completar los mismos con el usuario y contraseña. Los sistemas E-SSO permiten interactuar con sistemas que pueden deshabilitar la presentación de la pantalla de login.

- ***Web Single Sign-On (Web-SSO):*** también llamado “Web Access Management (Web-AM)” trabaja sólo con aplicaciones y recursos accedidos vía Web. Los accesos son interceptados con la ayuda de un servidor Proxy o de un componente instalado en el servidor Web destino. Los usuarios no autentificados que tratan de acceder son redirigidos a un servidor de autentificación y regresan solo después de haber logrado un acceso exitoso. Se utilizan cookies, para reconocer aquellos usuarios que acceden y su estado de autentificación.

- ***Kerberos:*** es un método popular de externalizar la autentificación de los usuarios. Los usuarios se registran en el servidor Kerberos y reciben un ticket, luego los clientes de acceso lo presentan para obtener acceso.

- ***Federation:*** es una nueva manera de concebir este tema, también para aplicaciones Web. Utiliza protocolos basados en estándares para habilitar que las aplicaciones puedan identificar los clientes sin necesidad de autentificación redundante.
- ***OpenID:*** es un proceso de SSO distribuido y descentralizado donde la identidad se compila en una URL que cualquier aplicación o servidor puede verificar. [2]

Los SSO, de manera general, presentan características distintivas, para su buen funcionamiento, como son:

- ***Multiplataforma:*** facilita las tareas de inicio de sesión y de acceso a recursos de red desde distintas plataformas (Windows, GNU/Linux, Solaris, etc.).

- ***Transparencia:*** el acceso a los recursos de sistemas se efectúa de forma transparente al usuario debido a la automatización del inicio de sesión.

- ***Facilidad de uso:*** el usuario se autentifica una única vez y el sistema le permite acceder a los recursos para los cuales esta autorizado. Así se evita las interrupciones producidas por la solicitud de usuario y contraseña para el acceso a diferentes recursos.

- ***Gestión sencilla:*** el uso de SSO aconseja la sincronización de contraseñas e información de los usuarios. Esto implica la simplificación de la gestión de los recursos por parte de los administradores.

- ***Control de acceso:*** no se ve afectado por el uso de este sistema, SSO implica cambiar los mecanismos de autentificación del cliente y/o servidor, pero no modifica los permisos de los recursos.

- ***Seguridad:*** en todos los casos la información viaja cifrada por la red (SSL, certificados digitales, etc.).[3]

Existen diferentes tipos de arquitecturas que permiten implementar un SSO. Cada una de ellas posee características que la hace más apropiada para algún tipo de organización. La decisión de adoptar una u otra arquitectura básicamente depende de los recursos computacionales y/o económicos disponibles, y las decisiones de diseño establecidas por el equipo del proyecto.
El principal objetivo de una arquitectura que implemente Single Sign-On es transferir la funcionalidad y complejidad de todos los componentes de seguridad a un solo servicio. En dicha arquitectura, todos los mecanismos de protección se encuentran concentrados en el SSO, siendo este el único punto de autentificación y registro en el sistema.

Las diferentes arquitecturas SSO están conformadas, y descritas en este trabajo, en función a tres componentes básicos:

- ***Interfaz*:** El modo en que el SSO interactúa con una determinada aplicación. Usualmente reside en el cliente, y es conocido como Agente SSO.

- ***Administración*:** El mecanismo que permite configurar, mantener y monitorear el proceso de SSO.

- ***Credenciales*:** Cada aplicación a la que se accede requiere información confidencial (nombre de usuario, contraseña, etc.), que agrupada recibe el nombre de credenciales. Las credenciales deben almacenarse de manera protegida para que sea únicamente el agente SSO quien pueda acceder a ellas.[4]

Dentro de las arquitecturas ya existentes para la implementación de un SSO se encuentran cinco casos típicos, cuatro básicos y uno especial donde se fusionan aspectos de dos arquitecturas diferentes, que se diferencian entre ellos en cuanto a características, ventajas y desventajas que ofrecen a la organización que decida adoptarlos. Estos casos son, a continuación, descritos en detalle para una mayor comprensión del funcionamiento de los sistemas SSO.

1.2 – Arquitectura de Password Vault:

Esta es la arquitectura (*Fig. 1*) que se define con menos nivel de complejidad para su implantación. En este caso los tres elementos de la arquitectura se encuentran ubicados en el cliente y, por lo tanto, es justamente allí desde donde se accede a las aplicaciones. Por ello se deben previamente almacenar las credenciales correspondientes, para que puedan ser suministradas a las aplicaciones cuando sea necesario.

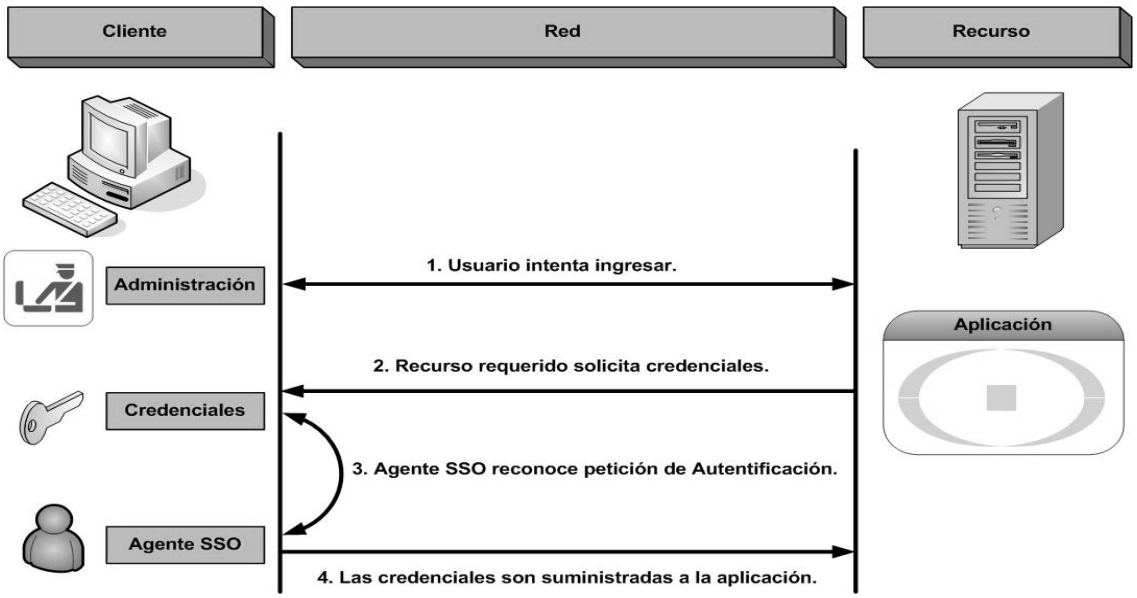

Fig.1: Arquitectura para SSO de Password Vault.

Esta arquitectura para SSO, siendo la más simple de implementar, presenta características distintivas, ventajas y desventajas en su modo de empleo.

1.2.1- Características de la arquitectura Password Vault:

- No es posible actualizar los clientes de manera masiva en toda la organización, requiere que se realice máquina por máquina.
- Funciones administrativas limitadas. La administración de cada uno de los clientes debe realizarse desde la estación correspondiente y por lo tanto generalmente termina quedando a cargo del usuario.

1.2.2- Ventajas de la arquitectura Password Vault:

- Pocos recursos computacionales necesarios. Un servidor central donde residen las diferentes aplicaciones y los clientes necesarios.
- Su implementación no es mucho más complicada que instalar un nuevo software en el equipo cliente.

1.2.3- Desventajas de la arquitectura Password Vault:

- El nivel de transparencia para el usuario es bajo ya que este generalmente se encuentra comprometido con la administración del proceso de ingreso.
- La información entre el cliente SSO y el servidor no viaja cifrada.
- El almacenamiento local de credenciales no permite que el usuario acceda a las aplicaciones desde múltiples estaciones.
- La administración local obliga a tomar medidas adicionales de seguridad informática y control de acceso por parte de la empresa.

1.3 - Arquitectura de administración centralizada con almacenamiento local de credenciales:

Con el propósito de solucionar los principales inconvenientes que presenta la arquitectura "*Password Vault"*, surge la de *"Administración centralizada con almacenamiento local de credenciales" (Fig. 2).* La misma ofrece un mecanismo para controlar y supervisar el proceso de ingreso; eliminando, de esta forma, la necesidad de configurar el SSO en cada uno de los clientes.

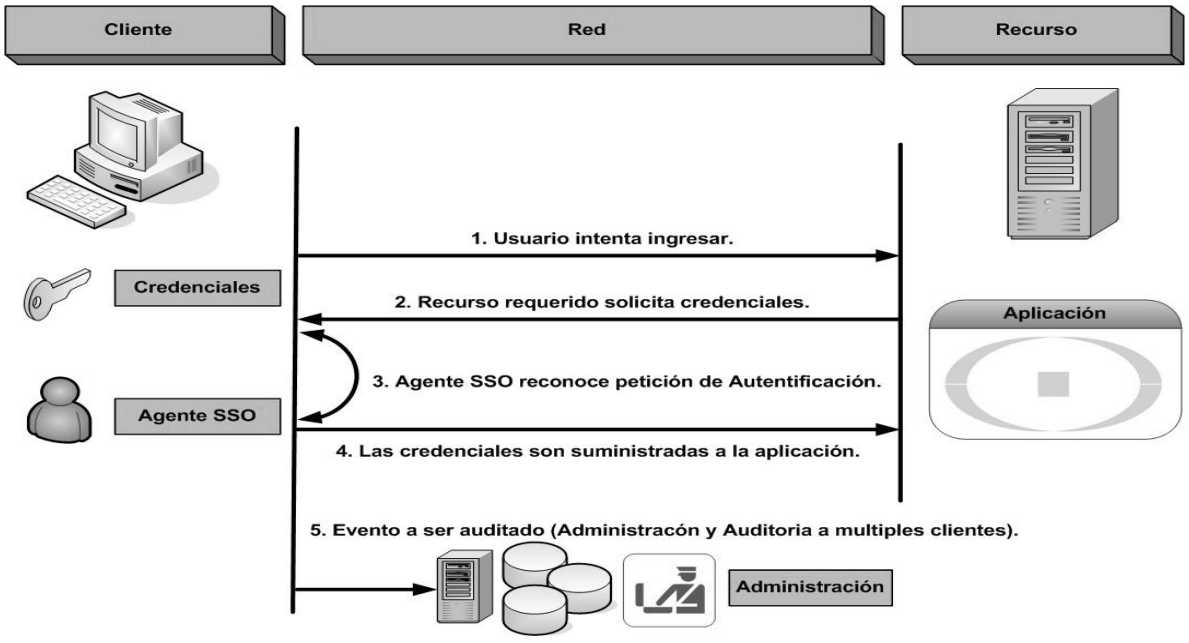

Fig.2: Arquitectura para SSO de administración centralizada con almacenamiento local de credenciales.

1.3.1- Características de la arquitectura de administración centralizada con almacenamiento local de credenciales:

- Las credenciales permanecen en el cliente.
- El software cliente es autónomo durante el proceso de autentificación, la labor de administración se restringe a realizar monitoreo de los clientes.
- Incluye un servidor central que permite realizar labores de administración.

1.3.2- Ventajas de la arquitectura de administración centralizada con almacenamiento local de credenciales:

- Las labores de administración tienen un bajo grado de complejidad.
- Control centralizado de la configuración y monitoreo del software del cliente.

1.3.3- Desventajas de la arquitectura de administración centralizada con almacenamiento local de credenciales:

- La información entre el cliente SSO y el servidor no viaja cifrada.
- Al conectarse el cliente, el administrador del SSO sólo puede monitorear la conexión. No puede efectuar acciones de desconexión o cambio de configuración de la misma.
- Al almacenar las credenciales en el cliente, se deben tomar medidas de control de acceso y confidencialidad de la información.

1.4 - Arquitectura de administración y almacenamiento de credenciales centralizados:

La arquitectura de *"Administración y almacenamiento de credenciales centralizados" (Fig. 3)*, pretende solucionar los principales inconvenientes encontrados en la arquitectura que almacena las credenciales localmente *("Administración centralizada con almacenamiento local de credenciales*"). En ella se puede apreciar un mayor grado de seguridad en cuanto al componente de las credenciales, que se almacenan en un lugar centralizado, facilitando así su protección, administración y manejo.

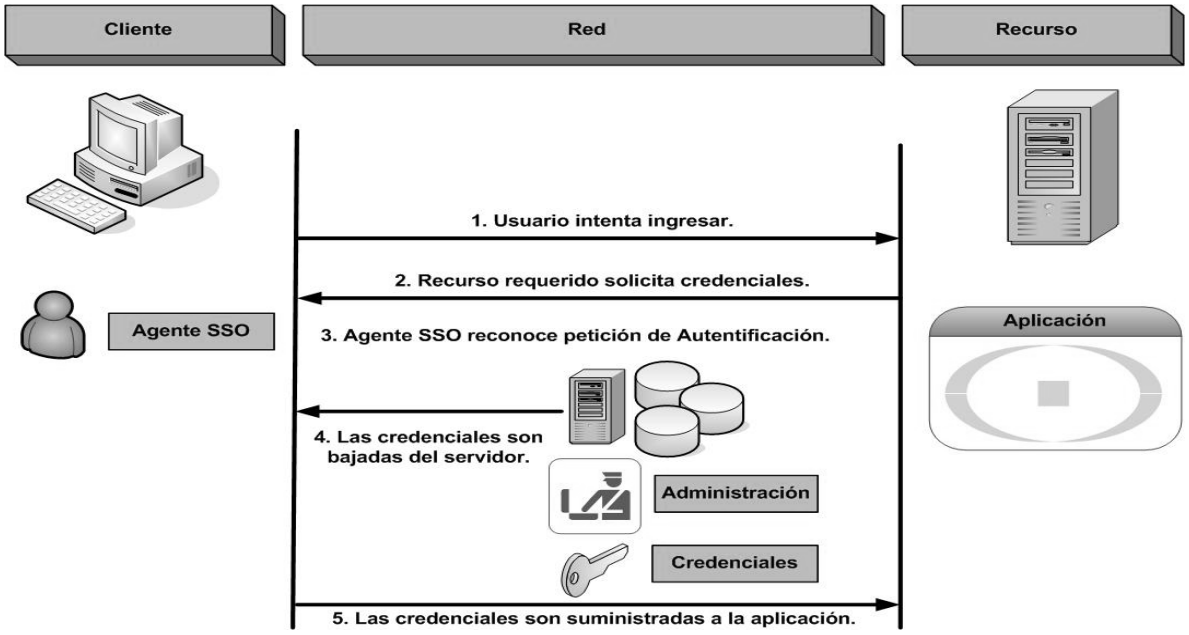

Fig.3: Arquitectura para SSO de administración y almacenamiento de credenciales centralizado.

1.4.1- Características de la arquitectura de administración centralizada con almacenamiento local de credenciales:

- Las credenciales son almacenadas a un servidor central, quien entrega las credenciales al cliente correspondiente en el momento de hacer el ingreso.
- El administrador determina la frecuencia con que se descargan las credenciales del servidor (Por sesión, por autentificación, etc.).

1.4.2- Ventajas de la arquitectura de administración centralizada con almacenamiento local de credenciales:

- Ofrece administración centralizada de credenciales. Esto aumenta la seguridad con respecto al almacenamiento local de las credenciales, por parte del cliente, usado en arquitecturas explicadas previamente en este trabajo.
- Permite a los usuarios el acceso a las aplicaciones desde cualquier estación, con autentificación válida para ello.

1.4.3- Desventajas de la arquitectura de administración centralizada con almacenamiento local de credenciales:

- Se crea un único punto de falla, convirtiendo al SSO en un gateway para todos los recursos de la organización, ya que el servidor debe ser contactado cada vez que se realice un ingreso.
- El acceso a todas las aplicaciones de la organización depende del servidor central.
- La configuración carece de redundancia, recuperación entre fallas y respaldo.
- La información entre el cliente SSO y el servidor no viaja cifrada.

1.5- Arquitectura totalmente distribuida:

Esta arquitectura *(Fig. 4)* es la de mayor complejidad en su implementación y requiere de más inversión en equipos de cómputo. Se caracteriza principalmente por separar el servidor de SSO del servidor de base de datos, lo cual la hace completamente modular.
Soluciona los problemas encontrados en las arquitecturas anteriormente presentadas y adicionalmente ofrece múltiples ventajas.

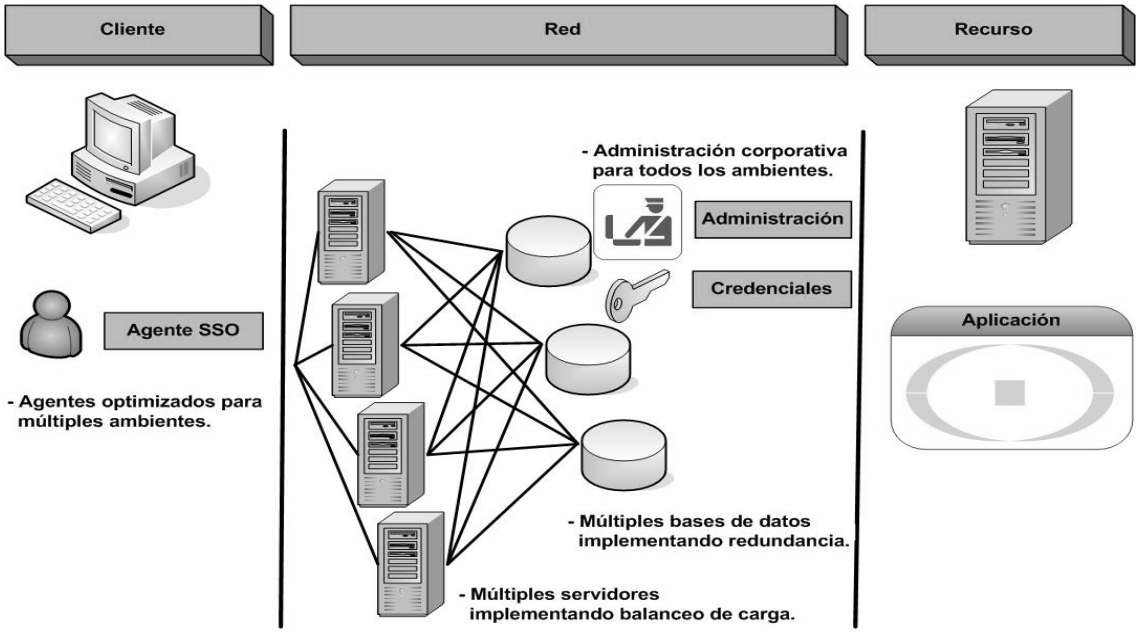

Fig.4: Arquitectura para SSO totalmente distribuida.

1.5.1- Características de la arquitectura totalmente distribuida:

- Cuenta con SSO avanzados que utilizan bases de datos escalables que soportan redundancia (MySQL, PostgreSQL, i.e. SQL Server, Oracle, etc.).
- Las bases de datos se encuentran sincronizadas con el fin de lograr redundancia y respaldo.
- La información se accede en el momento de ingreso.
- El servidor resulta ser una aplicación independiente que cuenta con un administrador diferente.
- El proceso de autentificación se encuentra en un recurso de red. Cuando el agente SSO establezca conexión con un servidor SSO, las credenciales serán solicitadas (almacenadas en memoria caché para realizar offline-logon) y el proceso de autentificación se realizará.
- La información es almacenada en bases de datos o en directorios de manera encriptada.

1.5.2- Ventajas de la arquitectura totalmente distribuida:

- Los agentes SSO se encuentran optimizados para múltiples ambientes (Terminal Server, Win32, GNU/Linux, Mac, etc.).
- Contiene múltiples bases de datos sincronizadas, implementando redundancia.
- Contiene múltiples servidores implementando balanceo de la carga para aumentar la disponibilidad y la atención de los requerimientos de autentificación.
- El hecho de contar con múltiples servidores adicionalmente hace que se disminuya la latencia de la red.
- Permite realizar funciones de administración corporativa para todos los ambientes.

1.5.3- Desventajas de la arquitectura totalmente distribuida:

- Solución altamente costosa por el ambiente distribuido requerido.
- Demorada implementación técnica por interacción entre múltiples sistemas operacionales.
- Soporte y administración complejos por la consideración anterior.

- La información entre el cliente SSO y el servidor no viaja cifrada.

El excesivo costo que representa implementar esta arquitectura se debe no solamente a la adquisición de los servidores, mediante los cuales se logra la característica de balanceo de carga, y las múltiples bases de datos empleadas, para implementar redundancia, sino a los altos costos administrativos (personal especializado, software especial y tiempo requerido) que genera mantener funcionando un sistema de tal dimensión.

1.6- Arquitectura de administración y almacenamiento de credenciales centralizados garantizando alta disponibilidad y redundancia:

La *arquitectura de administración y almacenamiento centralizado de credenciales garantizando alta disponibilidad y redundancia (Fig. 5)* es un caso especial, no pertenece a las arquitecturas básicas de SSO, descritas anteriormente, pues la misma es una adaptación de la *"arquitectura de administración y almacenamiento de credenciales centralizado"* incorporando algunas de las ventajas de la *"arquitectura totalmente distribuida"*.

Se hace conveniente la descripción y exposición de la misma debido a sus características, que la hacen un tipo especial de arquitectura con la cual se adquieren ventajas de alta disponibilidad, donde el hardware y el software se encuentran preparados para presentar situaciones de contingencia, y redundancia (presenta una infraestructura duplicada).

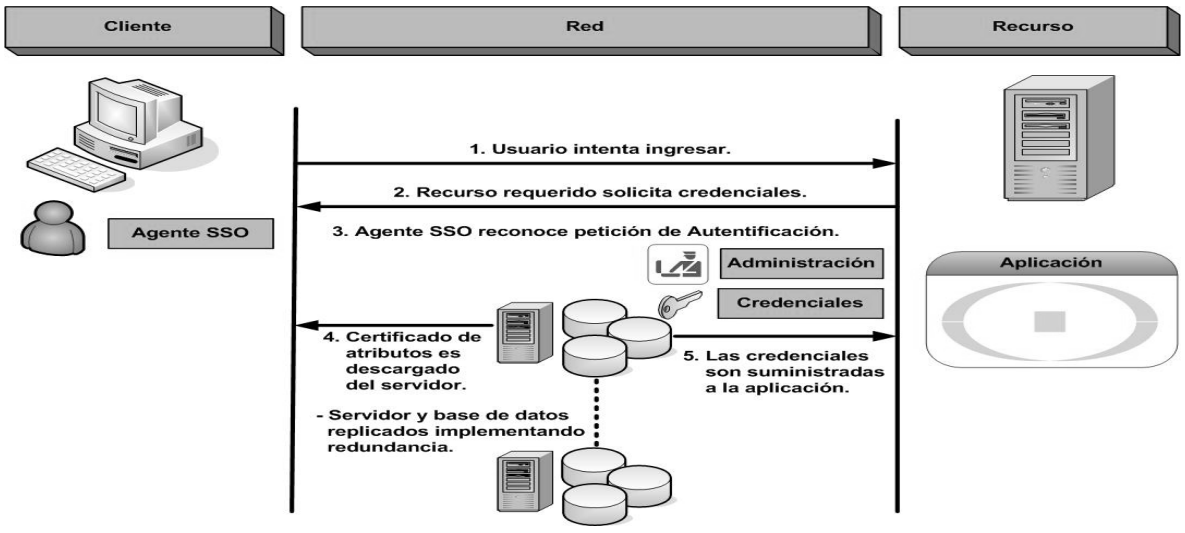

Fig.5: Arquitectura para SSO de administración y almacenamiento de credenciales centralizados garantizando alta disponibilidad y redundancia.

1.6.1- Características de la arquitectura de administración y almacenamiento de credenciales centralizados garantizando alta disponibilidad y redundancia:

- Incorpora infraestructura replicada con el fin de manejar la contingencia y redundancia en tiempo real.
- Las credenciales son almacenadas en un servidor central, quien entrega un certificado al cliente correspondiente, y las credenciales necesarias a la respectiva aplicación, en el momento de realizar el proceso de autentificación.
- Ofrece alta disponibilidad mediante software.

1.6.2- Ventajas de la arquitectura de administración y almacenamiento de credenciales centralizados garantizando alta disponibilidad y redundancia:

- Su infraestructura duplicada permite implementar alta disponibilidad y redundancia.
- Tanto el hardware como el software se encuentran debidamente especificados para enfrentar una situación de contingencia.
- Permite a los usuarios el acceso a las aplicaciones desde cualquier estación.
- Ofrece administración centralizada.

1.6.3- Desventajas de la arquitectura de administración y almacenamiento de credenciales centralizados garantizando alta disponibilidad y redundancia:

- La alta disponibilidad y redundancia que ofrece se basa en su infraestructura replicada, lo cual la hace costosa a nivel de hardware y software, como a nivel de administración y control de la misma.
- La información entre el cliente SSO y el servidor no viaja cifrada.

La implementación de la estrategia de Single Sign-On, para las organizaciones, deviene en beneficios adicionales como:

- Incremento en los niveles de seguridad existentes.
- Control centralizado de autentificación para las aplicaciones corporativas.

- Reducción de costos en la administración de la seguridad.
- Disminución de la operatividad asociada con la administración de contraseñas.
- Mayor comodidad y facilidad de uso de las aplicaciones corporativas para los usuarios finales.

Un sistema SSO correctamente diseñado debe poseer características que le permitan realizar un mismo procedimiento de autentificación bajo todos los ambientes. Esto significa que la organización no debería mantener infraestructuras paralelas para el procedimiento de ingreso y administración de sus usuarios locales y remotos bajo diferentes ambientes. Si esto ocurre, se deben definir con claridad las condiciones de operación en contingencia del SSO que definan las características requeridas para la autentificación y control de acceso a las aplicaciones corporativas.

El tipo de arquitectura SSO a implementar deberá determinarse en base a varios factores: su capacidad de personalización y flexibilidad, la complejidad deseada en la infraestructura, los recursos tecnológicos y económicos disponibles, entre otros. La arquitectura deseable en las grandes organizaciones es la *"Arquitectura Totalmente Distribuida"*, y en la mayoría de los casos la principal razón para no implementarla es la falta de recursos económicos.

1.7- Arquitectura adoptada para la creación del SSO de RINDE:

Teniendo en cuenta todos estos aspectos, estudiados previamente, se establece la arquitectura a implementar en el nuevo sistema SSO. Para ello se hizo un análisis de los servicios a brindar por el SSO frente a recursos económicos y de cómputo con que cuenta la organización. Este análisis descartó una posible línea a seguir de arquitecturas distribuidas *(Arquitectura Totalmente Distribuida* y *Arquitectura de Administración y Almacenamiento de Credenciales Centralizados Garantizando Alta Disponibilidad y Redundancia)* ya que no se hacía del todo necesario mantener sistemas distribuidos para 750 usuarios inicialmente, además de crecer el proyecto en tiempo y costo.

Dado esto se estableció la línea a seguir a partir de arquitecturas centralizadas *(Password Vault, Administración centralizada con almacenamiento local de credenciales* y *Administración y almacenamiento de credenciales centralizados)*, optando entre ellas por la más acorde al trabajo: *Arquitectura de Administración y Almacenamiento Centralizado de Credenciales.*

1.8- Metodología a usar:

El **Proceso Unificado Racional** (*Rational Unified Process* en inglés, habitualmente resumido como **RUP**) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado **UML**, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.
El RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización. La metodología RUP es más apropiada para proyectos grandes, dado que requiere un equipo de trabajo capaz de administrar un proceso complejo en varias etapas. En proyectos pequeños, es posible que no se puedan cubrir los costos de dedicación del equipo de profesionales necesarios.[5]

Existen metodologías como XP, SCRUM, entre otras, que no se ajustan a la aplicación que se desea implementar, ya que se debe desarrollar el sistema en un amplio conjunto de etapas.
XP y SCRUM se asocian más bien con procesos de desarrollo un poco más pequeños, están dentro del grupo de las metodologías ágiles.

1.9- Tecnologías utilizadas.

Se decide trabajar sobre herramientas libres debido al principal propósito de fomentar el desarrollo del Software Libre en la hermana República de Venezuela, además de ser un proyecto nacido en la Facultad Diez (Facultad de Software Libre) de la Universidad de las Ciencias Informáticas.

1.9.1- Lenguaje PHP.

Se escoge PHP como lenguaje de programación por estar bajo licencias de código abierto, por lo que se presenta como una alternativa de fácil acceso para todos. Permite las técnicas de Programación Orientada a Objetos, tiene una biblioteca nativa de funciones sumamente amplia e incluida, no requiere definición de tipos de variables y es capaz de controlar el manejo de excepciones
PHP es un lenguaje de programación interpretado usado normalmente para la creación de páginas Web. PHP es un acrónimo recursivo que significa "PHP Hypertext Pre-processor" (inicialmente PHP

Tools, o, Personal Home Page Tools).
Este lenguaje ofrece muchas ventajas, es un lenguaje multiplataforma, tiene capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL, puede expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones), posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.

1.9.2- Plataforma LAMP.

LAMP es un conjunto de aplicaciones que permite establecer una plataforma Web de desarrollo o producción. En esta palabra (LAMP) las letras tienen un fuerte significado.
La "L" cubre la capa del sistema operativo con Linux, la "A" denota Apache como servidor Web, "M" de MySQL y el último (“P”) componente está para los lenguajes de programación PHP, Perl, y Python.

1.9.2.1- Sistema Operativo Debian GNU/Linux.

Se hace uso del sistema operativo Debian GNU/Linux por ser libre, robusto, estable, rápido y no estar restringido a personas con grandes conocimientos de informática.

1.9.2.2- Servidor Web Apache.

El servidor HTTP Apache es un software (libre) para plataformas UNIX (BSD, GNU/Linux), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1. Presenta entre otras características mensajes de error altamente configurables, bases de datos de autentificación y negociado de contenido, tiene amplia aceptación en la red.

1.9.2.3- Gestor de Bases de Datos MySQL.

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones conocidas en todo el mundo.
No trabajamos con PostgreSQL porque las facilidades que ofrece trabajando con PHP son menores que con MySQL, no es tan ligero y consume más recursos, podríamos decir que PHP y MySQL son la

"pareja ideal" para desarrollo de plataformas Web libres.

1.9.4- CodeIgniter Framework.

Se optó por usar CodeIgniter ya que es un framework de desarrollo Web en PHP, que se encuentra bajo los términos de la licencia GNU/GPL (Software-Libre), con bajo uso de recursos, rendimiento excepcional y es altamente compatible con gran variedad de versiones y configuraciones de PHP. Ayuda a organizar todo el trabajo obligando a desarrollar aplicaciones Web siguiendo el patrón Modelo Vista Controlador (MVC).

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

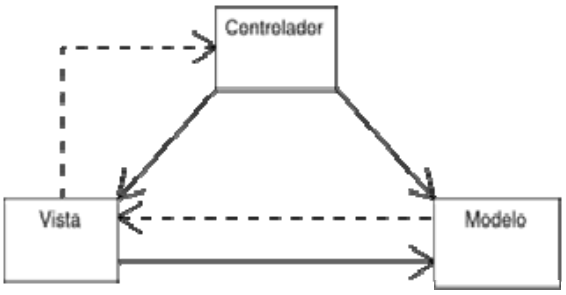

Fig.6: Modelo Vista Controlador.

El patrón MVC se ve frecuentemente en aplicaciones Web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página, el modelo es el Sistema de Gestión de Base de Datos y el controlador representa la lógica de negocio. [6]

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA.

El resultado del proyecto RINDE, en su primera etapa, provee una solución de software para la instalación y mantenimiento de herramientas de desarrollo colaborativo (Drupal, GForge y MediaWiki), personalizadas para el uso del gobierno venezolano.

En este punto del desarrollo de RINDE, existía un gran problema en cuanto a la unificación de los tres subsistemas Web existentes. Es entonces cuando surge la idea de unificar todo el sistema a través de la creación de un SSO.

2.1- Objeto de Automatización:

Este sistema SSO garantizaría que el uso de dichas herramientas se convierta en un proceso fácil y fluido, de forma que el usuario no tenga que hacer ningún tipo de trámite para acceder de uno a otro sitio Web, haciendo la navegación y el trabajo en RINDE más agradable. Todo esto, además, incurre en un mayor grado de seguridad, confidencialidad, coherencia y portabilidad en el gran sistema creado.

2.2- Modelo de Dominio del SSO:

Debido al bajo nivel de estructuración que presenta el negocio que se está estudiando y que está altamente centrado en las tecnologías informáticas, se propone un modelo del dominio ayudando a los usuarios, clientes, desarrolladores y demás interesados, a utilizar un vocabulario común para poder entender el contexto en que se emplaza el sistema.

El proceso de modelar permite obtener una visión de la organización, permitiendo definir los procesos y roles relacionados con la obtención de requerimientos y del análisis-diseño. Sí se determina que no es necesario un modelo completo del negocio se realizará lo que se conoce como un modelo del dominio.
Un modelo del dominio captura los tipos más importantes de objetos que existen o los eventos que suceden en el entorno donde estará el sistema.[7]

El modelo del dominio se describe mediante diagramas UML, específicamente con un diagrama de clases conceptuales significativas en el dominio del problema. Este va a contribuir posteriormente a identificar algunas clases que se utilizarán en el sistema.

2.2.1- Diagrama conceptual, Modelo de Dominio:

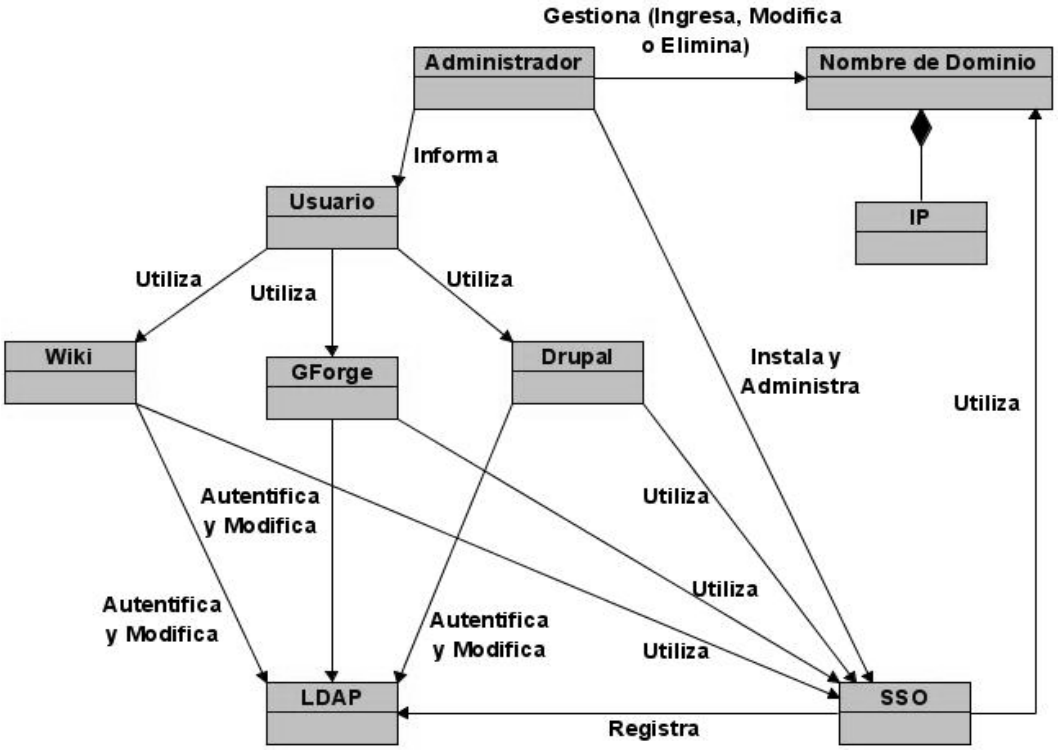

Fig.7: Modelo de Dominio.

Se identifican como conceptos utilizados en el diagrama a los siguientes términos del glosario:

- ***Administrador:*** Representa un usuario especial, con mayor grado de acceso al sistema. Es quien instala el sistema y administra los registros o autentificaciones directas al SSO.
- ***Usuario:*** Generaliza a todos los usuarios del sistema. Realiza las operaciones comunes a ellos.
- ***Drupal:*** Subsistema de RINDE, creado como portal Web informativo principal de comunicación, que trabaja como cliente del SSO de forma directa.
- ***Wiki:*** Subsistema de RINDE, creado como herramienta para publicar y mantener documentación de forma colaborativa, que trabaja como cliente del SSO de forma directa.

- ***GForge:*** Subsistema de RINDE, creado como herramienta para el desarrollo colaborativo de software, que trabaja como cliente del SSO de forma directa.
- ***LDAP:*** Directorio Activo de Usuarios creado para mantener una organización de los mismos, estableciendo para ello roles y niveles de accesos por usuarios, el cual es accedido directamente por el SSO en el instante de crear o autentificar algún usuario.
- ***SSO:*** Sistema de Autentificación Único que permite el registro y autentificación de usuarios de forma centralizada y segura en el sistema creado.

2.3- Especificaciones de los requisitos de software:

El flujo de especificación de los requisitos de software radica su mayor esfuerzo en el reconocimiento del problema como lo ve el usuario. Se define una evaluación del problema y se dan respuesta a las funcionalidades que tendrá el sistema. Con él se pretende entender el comportamiento del software y se establecen las características de la interfaz y el descubrimiento de restricciones adicionales de diseño.[8]

Reúne todas las ideas que los clientes, usuarios y miembros del equipo de proyecto tengan acerca de lo que debe hacer el sistema. Los requisitos se pueden clasificar en: funcionales y no funcionales.

2.3.1- Requerimientos funcionales:

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir, no alteran la funcionalidad del producto y se mantienen invariables sin importarle con que propiedades o cualidades se relacionen. Son el punto de partida para identificar qué debe hacer el sistema.[9]

A continuación se relacionan los que deben cumplir el SSO de RINDE:

- **R1:** Autentificar usuario Drupal.
    1. Introducir nombre de usuario y contraseña.
        - En caso de estar ya autentificado en algún otro sistema RINDE, debe pasar las credenciales de entrada y proveer el acceso según sus privilegios de forma automática.

2. Validar datos introducidos.
3. Mostrar al usuario las opciones a las que tiene acceso según el rol o permisos establecidos.

- **R2:** Autentificar Usuario GForge.
    1. Introducir nombre de usuario y contraseña.
        - En caso de estar ya autentificado en algún otro sistema RINDE, debe pasar las credenciales de entrada y proveer el acceso según sus privilegios de forma automática.
    2. Validar datos introducidos.
    3. Mostrar al usuario las opciones a las que tiene acceso según el rol o permisos establecidos.
- **R3:** Autentificar Usuario Wiki.
    1. Introducir nombre de usuario y contraseña.
        - En caso de estar ya autentificado en algún otro sistema RINDE, debe pasar las credenciales de entrada y proveer el acceso según sus privilegios de forma automática.
    2. Validar datos introducidos.
    3. Mostrar al usuario las opciones a las que tiene acceso según el rol o permisos establecidos.
- **R4:** Autentificar Usuario LDAP.
    1. Chequear los datos introducidos en el servidor LDAP.
    2. Validar datos introducidos.
- **R5:** Registrar Usuario Drupal.
    1. Introducir datos del Usuario.
    2. Validar datos introducidos.
    3. Registrar el usuario
        - Mostrar mensaje de confirmación de registro.

- Mostrar mensaje de error en caso de ocurrir alguna falla en el proceso.

- **R6:** Registrar Usuario GForge.
  1. Introducir datos del Usuario.
  2. Validar datos introducidos.
  3. Registrar el usuario
     - Mostrar mensaje de confirmación de registro
     - Mostrar mensaje de error en caso de ocurrir alguna falla en el proceso.
- **R7:** Registrar Usuario Wiki:
  1. Introducir datos del Usuario.
  2. Validar datos introducidos.
  3. Registrar el usuario
     - Mostrar mensaje de confirmación de registro
     - Mostrar mensaje de error en caso de ocurrir alguna falla en el proceso.
- **R8:** Registrar Usuario LDAP:
  1. Chequear los datos introducidos en el servidor LDAP.
  2. Validar datos introducidos.
  3. Registrar usuario.
- **R9:** Actualizar Usuario Drupal:
  1. Mostrar los datos actuales.
  2. Introducir nuevos datos del Usuario.
  3. Validar datos introducidos.
  4. Registrar el usuario.
     - Mostrar mensaje de confirmación de registro.
     - Mostrar mensaje de error en caso de ocurrir alguna falla en el proceso.

- **R10:** Actualizar Usuario GForge:
    1. Mostrar los datos actuales.
    2. Introducir nuevos datos del Usuario.
    3. Validar datos introducidos.
    4. Registrar el usuario.
        - Mostrar mensaje de confirmación de registro.
        - Mostrar mensaje de error en caso de ocurrir alguna falla en el proceso.
- **R11:** Actualizar Usuario Wiki:
    1. Mostrar los datos actuales.
    2. Introducir nuevos datos del Usuario.
    3. Validar datos introducidos.
    4. Registrar el usuario.
        - Mostrar mensaje de confirmación de registro.
        - Mostrar mensaje de error en caso de ocurrir alguna falla en el proceso.
- **R12:** Actualizar Usuario LDAP:
    1. Chequear los datos introducidos en el servidor LDAP.
    2. Validar datos introducidos.
    3. Actualizar datos.
- **R13:** Crear Usuario de Administración:
    1. Introducir datos del Usuario.
    2. Validar datos introducidos.
    3. Registrar el usuario en el SSO.
        - Mostrar mensaje de confirmación de registro.
        - Mostrar mensaje de error en caso de ocurrir alguna falla en el proceso.

- **R14:** Autentificar Usuario de Administración:
    1. Introducir nombre de usuario y contraseña.
    2. Validar datos introducidos.
    3. Mostrar al usuario la interfaz de administración del SSO.
- **R15:** Actualizar Usuario de Administración:
    1. Mostrar los datos actuales.
    2. Introducir nuevos datos del usuario.
    3. Validar datos introducidos.
    4. Registrar el usuario
        - Mostrar mensaje de confirmación de registro.
        - Mostrar mensaje de error en caso de ocurrir alguna falla en el proceso.
- **R16:** Agregar Nombre de Dominio:
    1. Mostrar los dominios presentes en el sistema.
    2. Introducir nuevos datos de dominio.
    3. Validar datos introducidos.
    4. Agregar el dominio.
        - Mostrar mensaje de confirmación de inserción
        - Mostrar mensaje de error en caso de ocurrir alguna falla en el proceso.
- **R17:** Agregar dirección de IP a un Nombre de Dominio:
    1. Mostrar los dominios presentes y sus direcciones IP asociadas.
    2. Introducir nuevas direcciones de IP al dominio seleccionado.
    3. Validar datos introducidos.
    4. Agregar la dirección de IP.
        - Mostrar mensaje de confirmación de inserción
        - Mostrar mensaje de error en caso de ocurrir alguna falla en el proceso.

- **R18:** Actualizar Nombre de Dominio:
    1. Mostrar los dominios presentes en el sistema.
    2. Introducir nuevos datos de dominio.
    3. Validar datos introducidos.
    4. Agregar el dominio.
        - Mostrar mensaje de confirmación de inserción
        - Mostrar mensaje de error en caso de ocurrir alguna falla en el proceso.
- **R19:** Actualizar dirección de IP de un Dominio:
    1. Mostrar los dominios presentes y sus direcciones IP asociadas.
    2. Introducir nuevas direcciones de IP al dominio seleccionado.
    3. Validar datos introducidos.
    4. Agregar la dirección de IP.
        - Mostrar mensaje de confirmación de inserción
        - Mostrar mensaje de error en caso de ocurrir alguna falla en el proceso.
- **R20:** Eliminar Nombre de Dominio:
    1. Mostrar los nombres de dominio presentes en el sistema.
    2. Seleccionar nombre de dominio a eliminar.
    3. Eliminar nombre de domino seleccionado.
- **R21:** Eliminar dirección de IP de un Dominio:
    1. Mostrar un listado de nombres de dominios existentes en el sistema.
    2. Seleccionar nombre de dominio.
    3. Mostrar un listado de todas las direcciones IP referentes a ese nombre de dominio.
    4. Seleccionar la(s) dirección(es) IP a eliminar.
    5. Eliminar dirección IP seleccionada.

- **R22:** Consultar las direcciones de IP de un Dominio:
    1. Mostrar listado de nombres de dominio.
    2. Seleccionar nombre de dominio.
    3. Mostrar un listado de todas las direcciones IP referentes a ese nombre de dominio.
- **R23:** Consultar el Nombre de Dominio de una dirección IP:
    1. Introducir dirección IP a consultar.
        - Mostrar nombre de dominio al que pertenece.
        - Mostrar mensaje confirmando la no existencia de la dirección IP en dominio alguno.

2.3.2- Requerimientos no funcionales:

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Representan las características que hacen al producto atractivo, usable, rápido o confiable, son fundamentales en el éxito del producto. Normalmente están vinculados a requerimientos funcionales. Los requerimientos no funcionales forman una parte significativa de la especificación. [10]

Entre las diferentes categorías de los requerimientos no funcionales para nuestro SSO se encuentran:

- **Apariencia o interfaz externa:**
    - Diseño sencillo, permitiendo la utilización del sistema sin mucho entrenamiento.
    - Diseñado para la resolución 1024x768, pero preparado para verse en otras resoluciones.
- **Soporte:**
    - Garantía de instalación y prueba del sistema, además de un breve entrenamiento a los futuros usuarios.
- **Implementación:**
    - Usar PHP como Lenguaje de Programación.

- Utilizar MySQL como Sistema Gestor de Bases de Datos.

- **Portabilidad:**
  - Independencia de la Plataforma.
- **Seguridad:**
  - Existencia de distintos roles que establezcan las acciones que pueden realizar los usuarios.
  - Verificación sobre acciones irreversibles (por ejemplo las eliminaciones).
  - Transmisión de datos a la red de forma encriptada.
- **Legales:**
  - La plataforma escogida para el desarrollo e implantación de la aplicación, está bajo los términos de la licencia GNU/GPL v3.
- **Confiabilidad:**
  - Garantía de un tratamiento adecuado de las excepciones y validación de las entradas del usuario.

2.4- Descripción del sistema propuesto:

Teniendo en cuenta todos objetivos de nuestro trabajo y los requerimientos planteados, el sistema SSO queda estructurado en los siguientes paquetes para una mayor organización y comprensión del mismo.

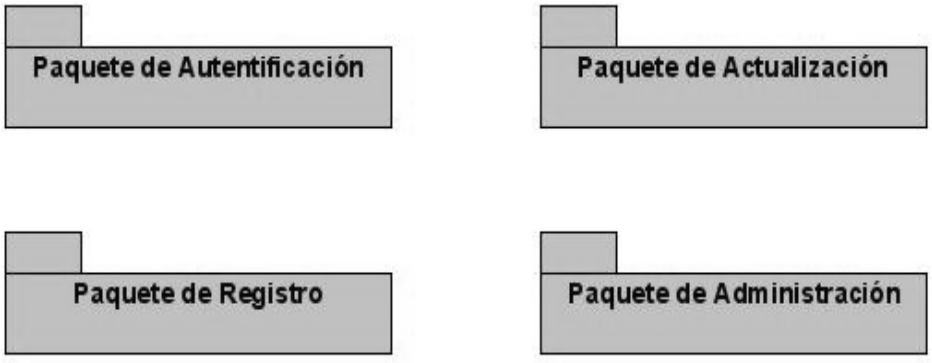

Fig.8 Diagrama de paquetes de casos de uso del sistema.

1. **Paquete de Autentificación.**
    - Autentificación de usuarios de la herramienta GForge.
    - Autentificación de usuarios de la herramienta Drupal.
    - Autentificación de usuarios de la herramienta Wiki.
    - Autentificación de los usuarios (GForge, Drupal, Wiki) contra registros del LDAP.
2. **Paquete de Actualización.**
    - Actualización de datos de lo usuarios de la herramienta GForge.
    - Actualización de datos de lo usuarios de la herramienta Drupal.
    - Actualización de datos de lo usuarios de la herramienta Wiki.
    - Actualización de datos de los usuarios (GForge, Drupal, Wiki) en los registros del LDAP.
3. **Paquete de Registro.**
    - Creación de usuarios en la herramienta GForge.
    - Creación de usuarios en la herramienta Drupal.
    - Creación de usuarios en la herramienta Wiki.
    - Creación de usuarios (GForge, Drupal, Wiki) en los registros del LDAP.
4. **Paquete de Administración.**
    - Autentificación del usuario de administración del sistema SSO.
    - Actualización de datos del usuario de administración del sistema SSO.
    - Creación del usuario de administración del sistema SSO.
    - Agregar dominio de trabajo al SSO.
    - Agregar direcciones de IP a un dominio de trabajo del SSO.
    - Actualizar un dominio de trabajo del SSO.
    - Actualizar direcciones de IP en un dominio de trabajo del SSO.
    - Eliminar un dominio de trabajo del SSO.
    - Eliminar direcciones de IP de un dominio de trabajo del SSO.
    - Consultar un listado de direcciones de IP pertenecientes a un dominio de trabajo del SSO.
    - Consultar el nombre de dominio al que pertenece una dirección IP.

A partir de este momento se realizará todo el modelado de los casos de uso del sistema en base a esta división en paquetes, para de esta forma desglosar en pequeños módulos de trabajo a todo el gran sistema SSO y quede mejor organizado, garantizando así una mayor comprensión por parte del lector.

2.5- Definición de los casos de uso:

Los casos de uso son artefactos narrativos que describen, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario. Por lo tanto, establece un acuerdo entre clientes y desarrolladores sobre las condiciones y posibilidades (requisitos) que debe cumplir el sistema. Entre ellos se encuentran las actividades a automatizar. [11]

2.5.1- Definición de los actores:

Los actores del sistema son los trabajadores del negocio (inclusive si fuera un sistema ya existente) que tiene actividades a automatizar es un candidato a actor del sistema. Si algún actor del negocio va a interactuar con el sistema, entonces también será un actor del sistema.
Los actores del sistema:

- No son parte de él.
- Pueden intercambiar información con él.
- Pueden ser un recipiente pasivo de información.
- Pueden representar el rol que juegan una o varias personas, un equipo o un sistema automatizado. [12]

Tabla 1: Actores del Sistema.

Actores del Sistema	Descripción
Usuario	Generaliza a todos los usuarios del sistema. Realiza las operaciones comunes a ellos.
Administrador	Es el encargado de definir las configuraciones con las que trabaja el sistema así de como mantener el SSO en funcionamiento.

2.5.2- Diagrama de casos de uso del sistema.

Un diagrama de casos de uso del sistema contiene los actores y los casos de uso del sistema, mostrando las diferentes relaciones que existen entre ellos. Estos se pueden estructurar en paquetes y deben ser fáciles de entender por el usuario final.

Los casos de uso son fragmentos de la funcionalidad del software, en ellos se describe la secuencia determinada que sigue un actor en su interacción con el sistema. A continuación se muestran los diagramas de caso de uso del sistema, divididos por paquetes.

En los anexos de este trabajo se encuentra una vista general del diagrama de casos de uso del sistema (*Ver Anexo 1*).

2.5.2.1- Diagrama de Casos de Uso. Paquete de Autentificación.

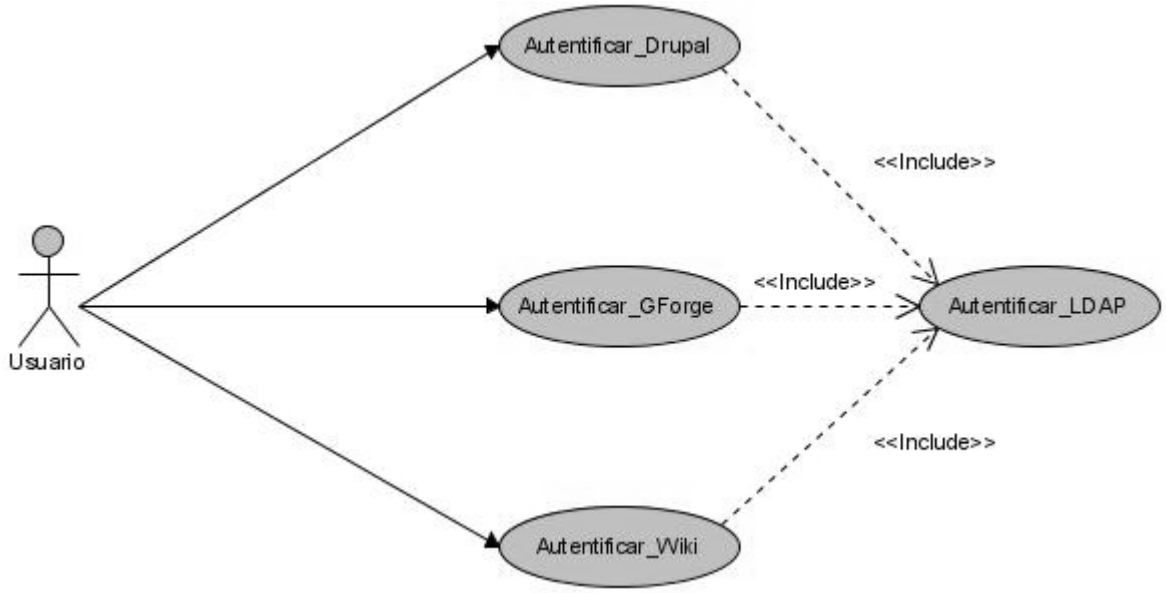

Fig.9: Diagrama de casos de uso. Paquete de autentificación.

- ***Descripción textual de los casos de uso del paquete de autentificación.***

El comportamiento de un caso de uso se especifica describiéndolo de forma textual.
Las especificaciones de los casos de uso se completan, a continuación, con un flujo de eventos detallado para una mejor descripción, comprensión y refinamiento.

A continuación se muestra la descripción detallada de cada uno de los casos de uso del sistema, las acciones a llevar a cabo tanto por el usuario como por el sistema, que intervienen en este paquete de autentificación.

Estos casos de uso del paquete de autentificación del SSO se encuentran relacionados con los requisitos funcionales: *R1, R2, R3, R4.*

Tabla 2: CU Autentificar GForge.

CU-1	**Autentificar GForge**
Actores	Usuario
Propósito	Autentificar al usuario
Resumen: El caso de uso comienza cuando el usuario se sienta en la máquina con el objetivo de acceder al sistema GForge, escribe su identificador y su contraseña. Termina cuando el sistema le concede el acceso si están correctos los datos entrados por el usuario o envía un mensaje de error en caso de ser datos incorrectos.	
Referencia	R2, R4
Pre-condiciones	El usuario debe haber sido creado
Post-condiciones	El usuario es autentificado
Flujo de Trabajo	
Acción del Actor	**Respuesta del Sistema**
1. El usuario GForge accede a la aplicación. **3.** El usuario GForge introduce sus credenciales y envía los datos. **5.** El usuario GForge accede al sistema.	**2.** El sistema muestra una ventana que brinda la posibilidad al usuario de introducir credenciales. **4.** El sistema procesa los datos entrados por el usuario y le asigna un token de sesión después de verificar que esté registrado en la base de datos de GForge y en el LDAP, concediéndole así el acceso de acuerdo a sus privilegios.
Curso alternativo de los eventos	
	4. Si el usuario no está registrado en la base de datos del sistema GForge, y si en la base de datos central (LDAP), lo registra en la base de datos GForge y le permite acceder al sistema. **4.** Si el usuario no está registrado en la base de datos del sistema GForge, ni en la base de datos central (LDAP), le muestra un mensaje de error diciendo que los datos entrados son incorrectos, y le brinda la posibilidad de volver a introducir sus credenciales (volver a 3).

Tabla 3: CU Autentificar Drupal.

CU-2	**Autentificar Drupal**
Actores	Usuario
Propósito	Autentificar al usuario
Resumen: El caso de uso comienza cuando el usuario se sienta en la máquina con el objetivo de acceder al sistema Drupal, escribe su identificador y su contraseña. Termina cuando el sistema le concede el acceso si están correctos los datos entrados por el usuario o envía un mensaje de error en caso de ser datos incorrectos.	
Referencia	R1, R4
Pre-condiciones	El usuario debe haber sido creado
Post-condiciones	El usuario es autentificado
Flujo de Trabajo	
Acción del Actor	**Respuesta del Sistema**
1. El usuario Drupal accede a la aplicación. **3.** El usuario Drupal introduce sus credenciales y envía los datos. **5.** El usuario Drupal accede al sistema.	**2.** El sistema muestra una ventana que brinda la posibilidad al usuario de introducir credenciales. **4.** El sistema procesa los datos entrados por el usuario y le asigna un token de sesión después de verificar que esté registrado en la base de datos de Drupal y en el LDAP, concediéndole así el acceso de acuerdo a sus privilegios.
Curso alternativo de los eventos	
	4. Si el usuario no está registrado en la base de datos del sistema Drupal, y si en la base de datos central (LDAP), lo registra en la base de datos Drupal y le permite acceder al sistema. **4.** Si el usuario no está registrado en la base de datos del sistema Drupal, ni en la base de datos central (LDAP), le muestra un mensaje de error diciendo que los datos entrados son incorrectos, y le brinda la posibilidad de volver a introducir sus credenciales (volver a 3).

Tabla 4: CU Autentificar Wiki.

CU-3	**Autentificar Wiki**
Actores	Usuario
Propósito	Autentificar al usuario
Resumen: El caso de uso comienza cuando el usuario se sienta en la máquina con el objetivo de acceder al sistema Wiki, escribe su identificador y su contraseña. Termina cuando el sistema le concede el acceso si están correctos los datos entrados por el usuario o envía un mensaje de error en caso de ser datos incorrectos.	
Referencia	R3, R4
Pre-condiciones	El usuario debe haber sido creado
Post-condiciones	El usuario es autentificado
Flujo de Trabajo	
Acción del Actor	**Respuesta del Sistema**
1. El usuario Wiki accede a la aplicación. **3.** El usuario Wiki introduce sus credenciales y envía los datos. **5.** El usuario Wiki accede al sistema.	**2.** El sistema muestra una ventana que brinda la posibilidad al usuario de introducir credenciales. **4.** El sistema procesa los datos entrados por el usuario y le asigna un token de sesión después de verificar que esté registrado en la base de datos de Wiki y en el LDAP, concediéndole así el acceso de acuerdo a sus privilegios.
Curso alternativo de los eventos	
	4. Si el usuario no está registrado en la base de datos del sistema Wiki, y si en la base de datos central (LDAP), lo registra en la base de datos Wiki y le permite acceder al sistema. **4.** Si el usuario no está registrado en la base de datos del sistema Wiki, ni en la base de datos central (LDAP), le muestra un mensaje de error diciendo que los datos entrados son incorrectos, y le brinda la posibilidad de volver a introducir sus credenciales (volver a 3).

2.5.2.2- Diagrama de Casos de Uso. Paquete de Actualización.

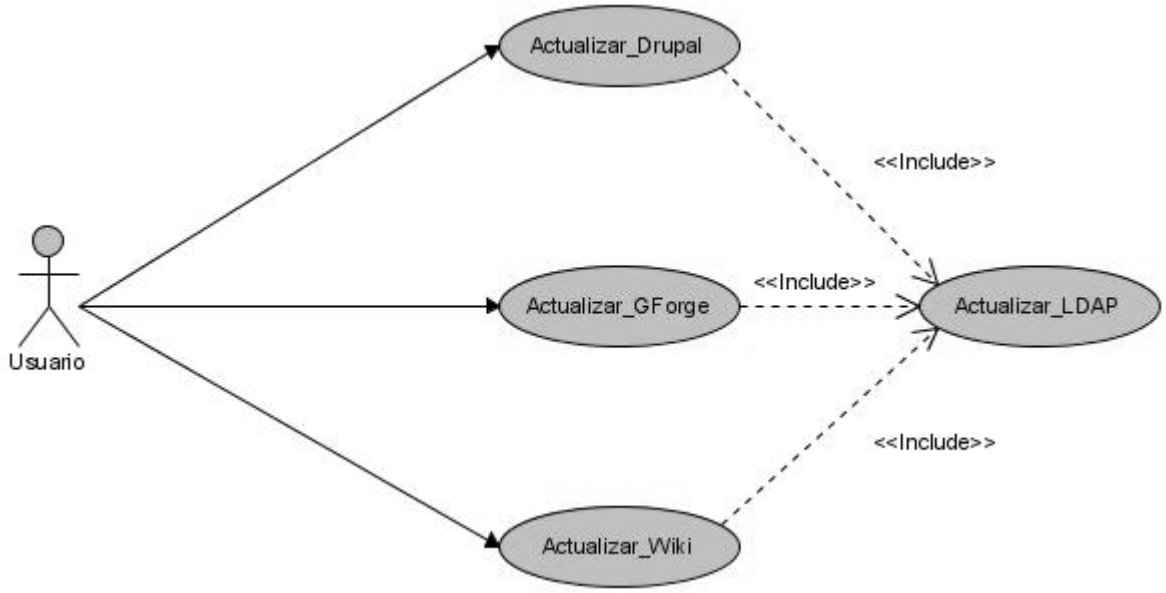

Fig.10: Diagrama de casos de uso. Paquete de actualización.

- ***Descripción textual de los casos de uso del paquete de actualización.***

A continuación se muestra la descripción detallada de cada uno de los casos de uso, las acciones a llevar a cabo tanto por el usuario como por el sistema, que intervienen en este paquete de actualización.

Estos casos de uso del paquete de actualización del SSO se encuentran relacionados con los requisitos funcionales: *R9, R10, R11, R12.*

Tabla 5: CU Actualizar GForge.

CU-4	**Actualizar GForge**
Actores	Usuario
Propósito	Actualizar datos del usuario en GForge.
Resumen: El caso de uso comienza cuando el usuario accede a la aplicación (GForge) y actualiza sus datos. El sistema verifica que estos se correspondan con el formato de entrada y los actualiza en la base de datos. Termina cuando los datos del usuario quedan actualizados en la base de datos de LDAP.	
Referencia	R10, R12
Pre-condiciones	El usuario debe estar autentificado.
Post-condiciones	Datos Actualizados.
Flujo de Trabajo	
Acción del Actor	**Respuesta del Sistema**
1. El usuario accede al subsistema GForge introduciendo sus credenciales. **3.** Se introducen los datos que se desean modificar. **5.** Se accede al sistema GForge con los datos actualizados.	**2.** El sistema muestra una ventana que brinda la posibilidad al usuario de actualizar sus datos. **4.** El sistema procesa los datos entrados por el usuario y actualiza con ellos la base de datos del GForge y de LDAP.
Curso alternativo de los eventos	
	4. Si los datos entrados por el usuario son incorrectos se muestra un mensaje de error y se le brinda la posibilidad de volver a introducir sus datos. (Volver a 3).

Tabla 6: CU Actualizar Drupal.

CU-5	**Actualizar Drupal**
Actores	Usuario
Propósito	Actualizar datos del usuario en Drupal.
Resumen: El caso de uso comienza cuando el usuario accede a la aplicación (Drupal) y actualiza sus datos. El sistema verifica que estos se correspondan con el formato de entrada y los actualiza en la base de datos. Termina cuando los datos del usuario quedan actualizados en la base de datos de LDAP.	
Referencia	R9, R12
Pre-condiciones	El usuario debe estar autentificado.
Post-condiciones	Datos Actualizados.
Flujo de Trabajo	
Acción del Actor	**Respuesta del Sistema**
1. El usuario accede al subsistema Drupal introduciendo sus credenciales. **3.** Se introducen los datos que se desean modificar. **5.** Se accede al sistema Drupal con los datos actualizados.	**2.** El sistema muestra una ventana que brinda la posibilidad al usuario de actualizar sus datos. **4.** El sistema procesa los datos entrados por el usuario y actualiza con ellos la base de datos del LDAP.
Curso alternativo de los eventos	
	4. Si los datos entrados por el usuario son incorrectos se muestra un mensaje de error y se le brinda la posibilidad de volver a introducir sus datos. (Volver a 3).

Tabla 7: CU Actualizar Wiki.

CU-6	**Actualizar Wiki**
Actores	Usuario
Propósito	Actualizar datos del usuario en Wiki.
Resumen: El caso de uso comienza cuando el usuario accede a la aplicación (Wiki) y actualiza sus datos. El sistema verifica que estos se correspondan con el formato de entrada y los actualiza en la base de datos. Termina cuando los datos del usuario quedan actualizados en la base de datos de LDAP.	
Referencia	R11, R12
Pre-condiciones	El usuario debe estar autentificado.
Post-condiciones	Datos Actualizados.
Flujo de Trabajo	
Acción del Actor	**Respuesta del Sistema**
1. El usuario accede al subsistema Wiki introduciendo sus credenciales. **3.** Se introducen los datos que se desean modificar. **5.** Se accede al sistema Wiki con los datos actualizados.	**2.** El sistema muestra una ventana que brinda la posibilidad al usuario de actualizar sus datos. **4.** El sistema procesa los datos entrados por el usuario y actualiza con ellos la base de datos del LDAP.
Curso alternativo de los eventos	
	4. Si los datos entrados por el usuario son incorrectos se muestra un mensaje de error y se le brinda la posibilidad de volver a introducir sus datos. (Volver a 3).

2.5.2.3- Diagrama de Casos de Uso. Paquete de Registro.

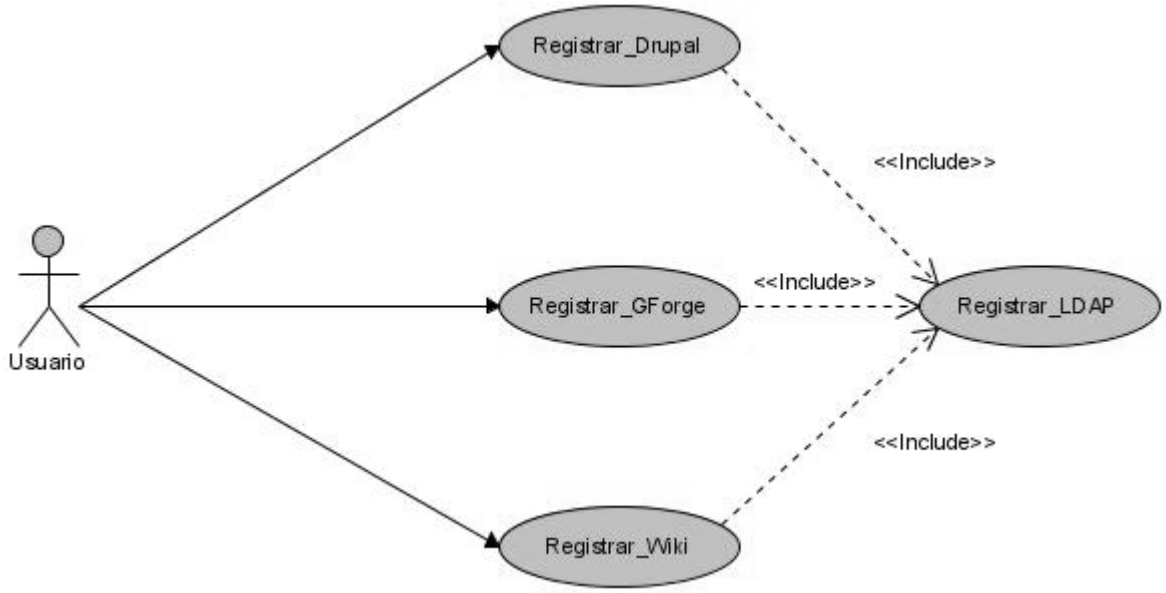

Fig.11: Diagrama de casos de uso. Paquete de registro.

- ***Descripción textual de los casos de uso del paquete de registro.***

A continuación se muestra la descripción detallada de cada uno de los casos de uso del sistema, las acciones a llevar a cabo tanto por el usuario como por el sistema, que intervienen en este paquete de registro.

Estos casos de uso del paquete de registro del SSO se encuentran relacionados con los requisitos funcionales: *R5, R6, R7, R8.*

Tabla 8: CU Registrar GForge.

CU-7	**Registrar GForge**
Actores	Usuario
Propósito	Registrar un usuario GForge a la base de datos.
Resumen: El caso de uso comienza cuando el usuario accede a la aplicación (GForge) y se dispone a registrar sus datos. El sistema verifica que estos se correspondan con el formato de entrada y que no estén registrados previamente en la base de datos de LDAP. De cumplirse esto procede a insertar el nuevo registro en la base de datos. Termina cuando los datos del usuario quedan registrados en el LDAP.	
Referencia	R6, R8
Pre-condiciones	El usuario no debe estar registrado en la base de datos de LDAP.
Post-condiciones	
Flujo de Trabajo	
Acción del Actor	**Respuesta del Sistema**
1. La acción comienza cuando el usuario accede a la aplicación GForge. **3.** El usuario introduce los datos y los envía. **6.** El usuario recibe una notificación y procede a la activación de su cuenta.	**2.** El sistema muestra una opción para registrar un usuario GForge en la base de datos. **4.** El sistema procesa los datos, verifica que el usuario no esté registrado en la base de datos LDAP. **5.** El sistema hace el registro del usuario en la base de datos del sistema GForge y envía una notificación por correo de activación de cuenta de usuario.
Curso alternativo de los eventos	
	4. El sistema procesa los datos y si ya existen en la base de datos de LDAP muestra un mensaje de error.

Tabla 9: CU Registrar Drupal.

CU-8	**Registrar Drupal**
Actores	Usuario
Propósito	Registrar un usuario Drupal a la base de datos.
Resumen: El caso de uso comienza cuando el usuario accede a la aplicación (Drupal) y se dispone a registrar sus datos. El sistema verifica que estos se correspondan con el formato de entrada y que no estén registrados previamente en la base de datos de LDAP. De cumplirse esto procede a insertar el nuevo registro en la base de datos. Termina cuando los datos del usuario quedan registrados en el LDAP.	
Referencia	R5, R8
Pre-condiciones	El usuario no debe estar registrado en la base de datos de LDAP.
Post-condiciones	
Flujo de Trabajo	
Acción del Actor	**Respuesta del Sistema**
1. La acción comienza cuando el usuario accede a la aplicación Drupal. **3.** El usuario introduce los datos y los envía. **6.** El usuario recibe una notificación y procede a la activación de su cuenta.	**2.** El sistema muestra una opción para registrar un usuario Drupal en la base de datos. **4.** El sistema procesa los datos, verifica que el usuario no esté registrado en la base de datos LDAP. **5.** El sistema hace el registro del usuario en la base de datos del sistema Drupal y envía una notificación por correo de activación de cuenta de usuario.
Curso alternativo de los eventos	
	4. El sistema procesa los datos y si ya existen en la base de datos de LDAP muestra un mensaje de error.

Tabla 10: CU Registrar Wiki.

CU-9	**Registrar Wiki**
Actores	Usuario
Propósito	Registrar un usuario Wiki a la base de datos.
Resumen: El caso de uso comienza cuando el usuario accede a la aplicación (Wiki) y se dispone a registrar sus datos. El sistema verifica que estos se correspondan con el formato de entrada y que no estén registrados previamente en la base de datos de LDAP. De cumplirse esto procede a insertar el nuevo registro en la base de datos. Termina cuando los datos del usuario quedan registrados en el LDAP.	
Referencia	R7, R8
Pre-condiciones	El usuario no debe estar registrado en la base de datos de LDAP.
Post-condiciones	
Flujo de Trabajo	
Acción del Actor	**Respuesta del Sistema**
1. La acción comienza cuando el usuario accede a la aplicación Wiki. **3.** El usuario introduce los datos y los envía. **6.** El usuario recibe una notificación y procede a la activación de su cuenta.	**2.** El sistema muestra una opción para registrar un usuario Wiki en la base de datos. **4.** El sistema procesa los datos, verifica que el usuario no esté registrado en la base de datos LDAP. **5.** El sistema hace el registro del usuario en la base de datos del sistema Wiki y envía una notificación por correo de activación de cuenta de usuario.
Curso alternativo de los eventos	
	4. El sistema procesa los datos y si ya existen en la base de datos de LDAP muestra un mensaje de error.

2.5.2.4- Diagrama de Casos de Uso. Paquete de Administración.

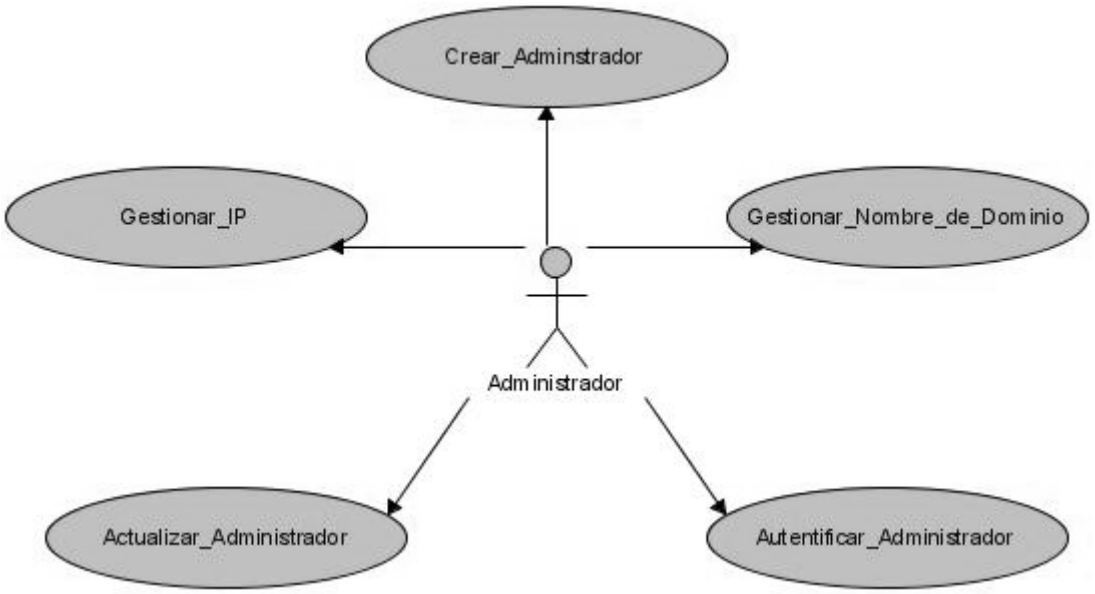

Fig.12: Diagrama de casos de uso. Paquete de administración.

- ***Descripción textual de los casos de uso del paquete de administración.***

A continuación se muestra la descripción detallada de cada uno de los casos de uso del sistema, las acciones a llevar a cabo tanto por el administrador como por el sistema, que intervienen en este paquete de administración.

Estos casos de uso del paquete de administración del SSO se encuentran relacionados con los requisitos funcionales: *R13, R14, R15, R16, R17, R18, R19, R20, R21, R22, R23.*

Tabla 11: CU Autentificar Administrador.

CU-10	**Autentificar Administrador**
Actores	Administrador
Propósito	Autentificar al Administrador.
Resumen: El caso de uso comienza cuando el administrador se sienta en la máquina con el objetivo de acceder a la administración del sistema SSO, escribe su identificador y contraseña. Termina cuando el sistema le concede el acceso, si están correctos los datos entrados por el usuario, o envía un mensaje de error en caso de ser datos incorrectos.	
Referencia	R14
Pre-condiciones	El usuario de administración debe haber sido creado.
Post-condiciones	El usuario de administración es autentificado.
Flujo de Trabajo	
Acción del Actor	**Respuesta del Sistema**
1. Se accede a la interfaz de autentificación del administrador del SSO. **3.** El administrador introduce sus credenciales y envía las envía. **5.** El usuario accede al sistema de administración del SSO.	**2.** La aplicación muestra la interfaz de acceso. **4.** El sistema valida la entrada contra la información de la base de datos, le asigna un token de sesión y provee el acceso.
Curso alternativo de los eventos	
	4. En caso de los datos de entrada sean erróneos, muestra un mensaje de error.

Tabla 12: CU Actualizar Administrador.

CU-11	**Actualizar Administrador**
Actores	Administrador
Propósito	Actualizar los datos del Administrador.
Resumen: El caso de uso comienza cuando el administrador accede al sistema con la intención de modificar sus datos. Termina cuando la información es actualizada en la base de datos.	
Referencia	R15
Pre-condiciones	El usuario de administración debe haber sido creado y autentificado.
Post-condiciones	Datos del administrador actualizados.
Flujo de Trabajo	
Acción del Actor	**Respuesta del Sistema**
1. Se accede a la interfaz de autentificación del administrador del SSO. **3.** El administrador introduce los datos que desea actualizar y los envía. **5.** Se accede al sistema con los datos modificados.	**2.** El sistema muestra una ventana que brinda la posibilidad al administrador de introducir y actualizar sus datos. **4.** El sistema procesa los datos entrados por el administrador, los almacena en la base de datos y le asigna un token de sesión.
Curso alternativo de los eventos	
	4. Si los datos entrados por el administrador son incorrectos, se muestra un mensaje de error y se le brinda la posibilidad de volver a introducirlos.(Volver a 3)

Tabla 13: CU Crear Administrador.

CU-12	**Crear Administrador**
Actores	Administrador
Propósito	Crea el usuario de administración del sistema.
Resumen: Este caso de uso es invocado en el momento de la instalación del sistema. El administrador teclea el nombre y la contraseña del usuario administrativo. Termina dicho caso de uso cuando el sistema finaliza la instalación.	
Referencia	R13
Pre-condiciones	El sistema no se encuentra instalado.
Post-condiciones	El sistema se encuentra instalado.
Flujo de Trabajo	
Acción del Actor	**Respuesta del Sistema**
1. El administrador accede a la aplicación. **3.** Se introducen los datos y se envían. **5.** El administrador procede a activar la nueva cuenta.	**2.** El sistema muestra una ventana que brinda la posibilidad al administrador de introducir sus datos y ser creado como usuario especial del sistema. **4.** El sistema procesa los datos entrados por el administrador, los inserta en la base de datos y envía una notificación de activación de cuenta.
Curso alternativo de los eventos	
	4. El sistema procesa los datos entrados por el administrador y si encuentra un error muestra un mensaje informándole del mismo.

Tabla 14: CU Gestionar Nombre de Dominio.

CU-13	**Gestionar Nombre de Dominio**
Actores	Administrador
Propósito	Poder gestionar los datos de los nombres de dominio. Realizar operaciones de registro, actualización, consulta y eliminación con los nombres de dominio.
Resumen: El caso de uso comienza cuando el administrador realiza cualquier operación, de registro, actualización, consulta o eliminación, con los nombres de dominio. Termina cuando quedan actualizados (o mostrados) los datos del nombre de dominio en el sistema.	
Referencia:	R16, R18, R20, R22
Pre-condiciones:	Administrador del sistema autentificado.
Post-condiciones:	
Flujo de Trabajo.	
Acción de Actor:	**Respuesta del Sistema:**
1. La acción comienza cuando el administrador accede a la aplicación.	**2.** El sistema ejecuta las siguientes acciones: - Si el administrador decide agregar un nombre de dominio, ir a la sección "Agregar Nombre de Dominio". - Si el administrador decide actualizar los datos de algún nombre de dominio existente, ir a la sección "Actualizar Nombre de Dominio". - Si el administrador decide eliminar algún nombre de dominio existente, ir a la sección "Eliminar Nombre de Dominio". - Si el administrador decide consultar las direcciones IP que pertenecen a un nombre de dominio, ir a la sección "Consultar Direcciones IP".
Curso alternativo de los eventos	
Escenario: "Agregar Nombre de Dominio".	
Flujo de Trabajo.	
Acción del Actor:	**Respuesta del Sistema:**
3. El administrador introduce los datos del nuevo nombre de dominio a agregar.	**4.** El sistema verifica que los campos obligatorios hayan sido llenados correctamente.

	5. El sistema almacena los datos del nuevo nombre de dominio registrado. **6.** El sistema envía un mensaje para informándole al administrador que ya se ha efectuado el registro satisfactoriamente.
Curso alternativo de los eventos.	
	4. Se emite un mensaje para que llene los campos obligatorios. **5.** Si el nombre de dominio ya existe se envía un mensaje notificándolo. Se le da la oportunidad de volver a introducir los datos (volver a 3)
Escenario: "Actualizar Nombre de Dominio".	
Flujo de Trabajo.	
Acción del Actor:	**Respuesta del Sistema:**
4. Se escoge el nombre de dominio que se desea actualizar. **6.** El administrador modifica los datos del nombre de dominio y los manda a guardar.	**3.** El sistema muestra un listado de los dominios existentes en la base de datos del SSO. **5.** El sistema muestra los datos del nombre de dominio escogido para actualizar. **7.** El sistema verifica que los campos obligatorios hayan sido llenados correctamente. **8.** El sistema muestra un mensaje informándole al usuario que ya ha sido efectuado satisfactoriamente la modificación.
Curso alternativo de los eventos.	
	7. Se procesan los datos de entrada y en caso de existir algún error se muestra un mensaje notificándolo. Se da la oportunidad de volver a introducir los datos (volver al 6).
Escenario: "Eliminar Nombre de Dominio".	
Flujo de Trabajo.	
Acción del Actor:	**Respuesta del Sistema:**
4. Se escoge la dirección el nombre de dominio que se desea eliminar.	**3.** El sistema muestra un listado de los nombres de dominio existentes en la base de datos del SSO.

	5. El sistema verifica que los campos obligatorios hayan sido llenados correctamente. **6.** El sistema muestra un mensaje informándole al usuario que ya ha sido efectuado satisfactoriamente la eliminación.
Curso alternativo de los eventos.	
Escenario: "Consultar Direcciones de IP de un Dominio".	
Flujo de Trabajo.	
Acción del Actor:	**Respuesta del Sistema:**
4. Se escoge el nombre de dominio que se desea saber sus direcciones de IP.	**3.** El sistema muestra un listado de los nombres de dominio existentes en la base de datos del SSO. **5.** El sistema verifica que los campos obligatorios hayan sido llenados correctamente. **6.** El sistema muestra un listado con las direcciones IP que pertenecen al nombre de dominio consultado.
Curso alternativo de los eventos.	

Tabla 15: CU Gestionar IP.

CU-14	**Gestionar IP**
Actores	Administrador
Propósito	Poder gestionar los datos de las direcciones IP. Realizar operaciones de registro, actualización, consulta y eliminación con las direcciones IP.
Resumen: El caso de uso comienza cuando el administrador realiza cualquier operación, de registro, actualización, consulta o eliminación, con las direcciones IP. Termina cuando quedan actualizados (o mostrados) los datos de las direcciones IP en el sistema.	
Referencia:	R17, R19, R21, R23
Pre-condiciones:	Administrador del sistema autentificado.
Post-condiciones:	
Flujo de Trabajo.	
Acción de Actor:	**Respuesta del Sistema:**
1. La acción comienza cuando el administrador accede a la aplicación.	**2.** El sistema ejecuta las siguientes acciones: - Si el administrador decide agregar direcciones IP ir a la sección "Agregar IP". - Si el administrador decide actualizar los datos de alguna dirección IP existente, ir a la sección "Actualizar IP". - Si el administrador decide eliminar alguna dirección de IP existente, ir a la sección "Eliminar IP". - Si el administrador decide consultar el nombre de dominio al cual pertenece una dirección IP, ir a la sección "Consultar nombre de dominio".
Curso alternativo de los eventos	
Escenario: "Agregar IP".	
Flujo de Trabajo.	
Acción del Actor:	**Respuesta del Sistema:**
3. El administrador introduce los datos de la nueva dirección IP a agregar.	**4.** El sistema verifica que los campos obligatorios hayan sido llenados correctamente. **5.** El sistema almacena los datos de la nueva dirección

	IP registrada **6.** El sistema envía un mensaje para informándole al administrador que ya se ha efectuado el registro satisfactoriamente.
Curso alternativo de los eventos.	
	4. Se emite un mensaje para que llene los campos obligatorios. **5.** Si la dirección IP ya existe se envía un mensaje notificándolo. Se le da la oportunidad de volver a introducir los datos (volver a 3)
Escenario: "Actualizar IP".	
Flujo de Trabajo.	
Acción del Actor:	**Respuesta del Sistema:**
4. Se escoge la dirección IP que se desea actualizar. **6.** El administrador modifica los datos de la dirección IP y los manda a guardar.	**3.** El sistema muestra un listado de las direcciones IP existentes en la base de datos del SSO. **5.** El sistema muestra los datos de la dirección IP escogida para actualizar. **7.** El sistema verifica que los campos obligatorios hayan sido llenados correctamente. **8.** El sistema muestra un mensaje informándole al usuario que ya ha sido efectuado satisfactoriamente la modificación.
Curso alternativo de los eventos.	
	7. Se procesan los datos de entrada y en caso de existir algún error se muestra un mensaje notificándolo. Se da la oportunidad de volver a introducir los datos (volver al 6).
Escenario: "Eliminar IP".	
Flujo de Trabajo.	
Acción del Actor:	**Respuesta del Sistema:**
4. Se escoge la dirección IP que se desea eliminar.	**3.** El sistema muestra un listado de las direcciones IP existentes en la base de datos del SSO. **5.** El sistema verifica que los campos obligatorios

	hayan sido llenados correctamente. **6.** El sistema muestra un mensaje informándole al usuario que ya ha sido efectuado satisfactoriamente la eliminación.
Curso alternativo de los eventos.	
Escenario: "Consulta nombre de dominio".	
Flujo de Trabajo.	
Acción del Actor:	**Respuesta del Sistema:**
4. Se escoge la dirección IP que se desea saber el nombre de dominio al cual pertenece.	**3.** El sistema muestra un listado de las direcciones IP existentes en la base de datos del SSO. **5.** El sistema verifica que los campos obligatorios hayan sido llenados correctamente. **6.** El sistema muestra un mensaje con el nombre de dominio al cual pertenece la dirección IP consultada.
Curso alternativo de los eventos.	
	5. Se emite un mensaje para que llene los campos obligatorios cuando no se escoja dirección IP alguna. Se le da la posibilidad de volver a escoger (volver al 4).

En este capítulo se comenzó a profundizar en el desarrollo de la propuesta de solución, obteniéndose una lista de las funcionalidades que debe tener el software, las mismas fueron representadas mediante un diagrama de casos de uso dividido en paquetes, y finalmente fueron descritas todas las acciones que realizan los actores y el sistema en general. A partir de aquí se puede comenzar la construcción del producto, cumpliendo con todos los requerimientos y las funcionalidades que se consideraron en este capítulo.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA.

Para llevar a cabo la realización del SSO personalizado de RINDE, fue necesario crear y establecer la arquitectura del mismo estableciendo un modelo de análisis y diseño que responda a las necesidades del cliente establecidas en la previa fase de captura de requisitos.
En el Modelo de Análisis del SSO en cuestión, se muestra una visión más específica de los requisitos, estableciéndolos y organizándolos, haciendo así mucho más fácil su comprensión, estructuración y mantenimiento.
En este modelo se tiene una primera aproximación al Modelo de Diseño, aunque no es imprescindible su realización teniendo en cuenta que la aplicación a desarrollar es orientada a la Web, se ha construido con algunas convenciones para facilitar la comprensión del sistema debido a la falta de la extensión de UML creada por Rational para el modelado de aplicaciones Web.
Se ha tomado como estereotipo de clase interfaz aquella página Web cuya función sea solo mostrar código HTML (página cliente), como estereotipo de clase de control aquellas páginas que solo gestionen los datos obtenidos (páginas servidor) y como estereotipo de clase entidad las tablas en las bases de datos.

3.1- Modelo de Análisis del SSO:

El Modelo de análisis contiene clases de análisis y sus objetos organizados en paquetes que colaboran.
Las clases de análisis se centran en los requisitos funcionales y son evidentes en el dominio del problema porque representan conceptos y relaciones del dominio. Tienen atributos y entre ellas se establecen relaciones de asociación, agregación / composición, generalización / especialización y tipos asociativos.
RUP propone clasificar las clases en:

- **Clases de interfaz:** Modelan la interacción entre el sistema y sus actores.
- **Clases entidad:** Modelan información que posee larga vida y que es a menudo persistente.
- **Clases de control:** Coordinan la realización de uno o unos pocos casos de uso coordinando las actividades de los objetos que implementan la funcionalidad del caso de uso.[13]

Los *Diagramas de Clases del Análisis* están organizados en cuatro paquetes, para una mayor claridad, organización y comprensión, puesto que así se concibieron previamente (*Diagrama de casos de uso del paquete de autentificación, Diagrama de casos de uso del paquete de actualización, Diagrama de casos de uso del paquete de registro y Diagrama de Casos de Uso del paquete de administración*) los casos de uso del sistema que intervienen en la creación del SSO:

- Diagramas de clases del análisis de los casos de uso del paquete de autentificación.
- Diagramas de clases del análisis de los casos de uso del paquete de actualización.
- Diagramas de clases del análisis de los casos de uso del paquete de registro.
- Diagramas de clases del análisis de los casos de uso del paquete de administración.

3.1.1- Diagramas de clases del análisis de los casos de uso del paquete de autentificación:

- ***Diagrama de clases del análisis del caso de uso "Autentificar GForge".***

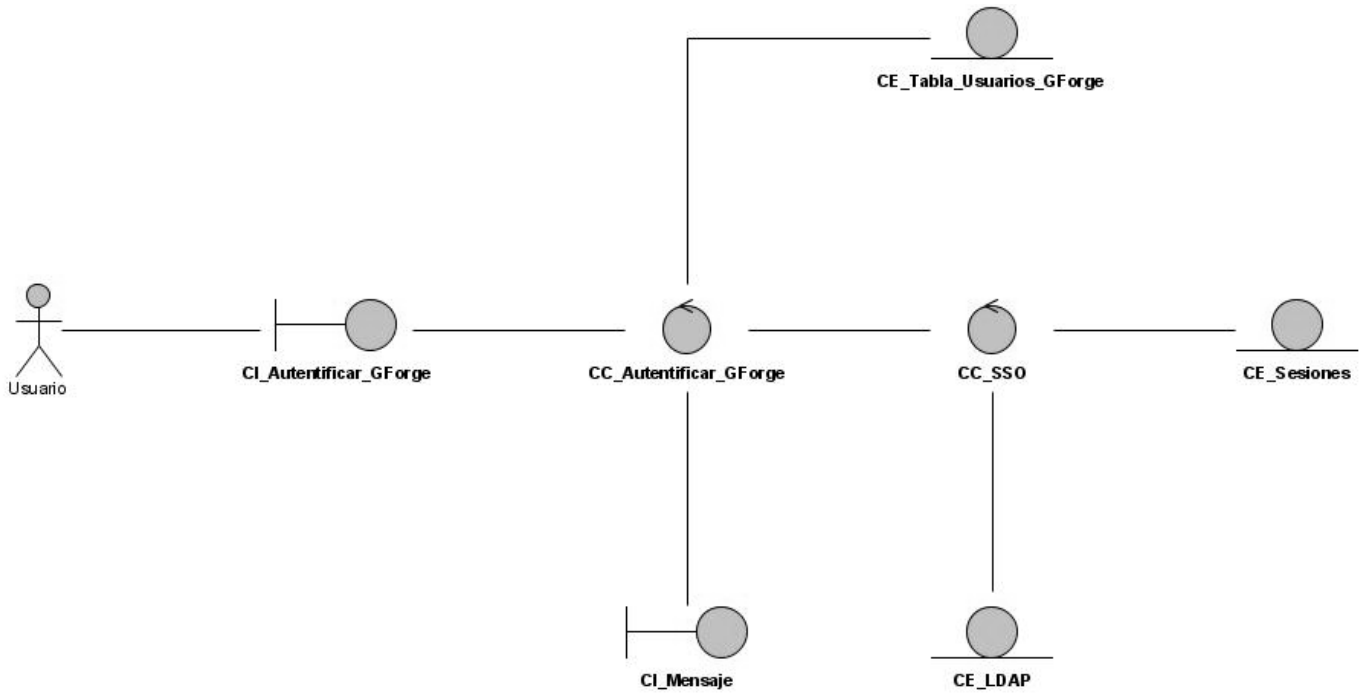

Fig.13: Diagrama de clases del análisis. CU-Autentificar GForge.

- ***Usuario***

Es el actor responsable de iniciar los casos de uso que engloban acciones que se realizan a partir de los subsistemas Web GForge, Drupal y Wiki:

- Autentificar GForge.
- Autentificar Drupal.
- Autentificar Wiki.
- Registrar GForge.
- Registrar Drupal.
- Registrar Wiki.
- Actualizar GForge.
- Actualizar Drupal.
- Actualizar Wiki.

- ***CI_Autentificar_GForge:***

Es la clase de interfaz que representa un sistema complejo independiente (GForge) que interactúan con la aplicación SSO. El usuario se autentifica mediante ella.

- ***CC_Autentificar_GForge:***

Es la clase de control, que se encarga de realizar las operaciones de autentificación en el GForge, para ello accede de forma directa a la tabla de usuarios registrados en dicho subsistema Web y se relaciona con una clase de control (CC_SSO) que verifica la existencia de el usuario en el directorio activo de usuarios del sistema (LDAP).

- ***CI_Mensaje:***

Clase de interfaz que es creada cuando existe algún error en el proceso de autentificación como forma de notificación para el usuario.

- ***CE_Tabla_Usuarios_GForge:***

Clase entidad que es una tabla de la base de datos del subsistema Web GForge, donde se guarda y consulta la información del usuario autentificado.

- ***CC_SSO:***

Clase control que realiza las operaciones del SSO (autentificación, registro, actualización), para ello accede de forma directa al LDAP para verificar los datos de los usuarios.

Es quien establece la relación entre las clases de control de los tres subsistemas Web (GForge, Drupal, Wiki) y el directorio activo de usuarios. Representa la unificación de todo el sistema RINDE.

- ***CE_LDAP:***

Clase entidad que guardará datos de los usuarios de todo el sistema en la base de datos central. Directorio activo de usuarios.

- ***CE_Sesiones:***

Clase entidad que es una tabla de la base de datos del SSO, donde se guardará y consultará la información de las sesiones creadas para cada usuario.

- ***Diagrama de clases del análisis del caso de uso Autentificar Drupal.***[III]

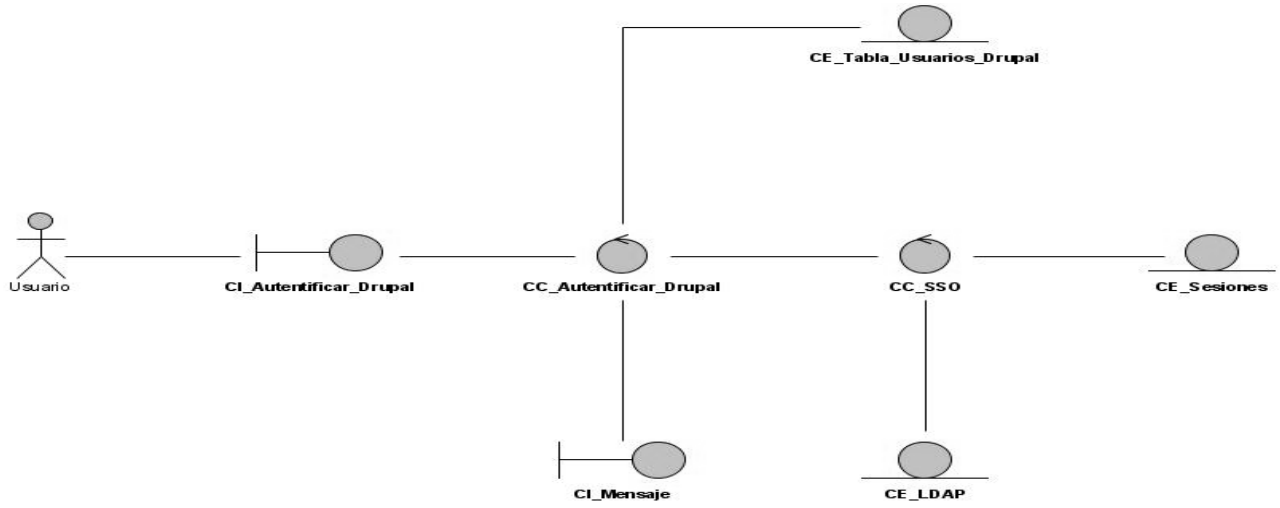

Fig.14: Diagrama de clases del análisis. CU-Autentificar Drupal.

[III] En los diagramas siguientes existen algunos componentes que no aparecen, a continuación, explicados debido a que mantienen la misma simbología y contenido explicativo.

- ***CI_Autentificar_Drupal:***

Es la clase de interfaz que representa un sistema complejo independiente (Drupal) que interactúan con la aplicación SSO. El usuario se autentifica mediante ella.

- ***CC_Autentificar_Drupal:***

Es la clase de control, que se encarga de realizar las operaciones de autentificación en el Drupal, para ello accede de forma directa a la tabla de usuarios registrados en dicho subsistema Web y se relaciona con una clase de control (CC_SSO) que verifica la existencia de el usuario en el directorio activo de usuarios del sistema (LDAP).

- ***CE_Tabla_Usuarios_Drupal:***

Clase entidad que es una tabla de la base de datos del subsistema Web Drupal, donde se guarda y consulta la información del usuario autentificado.

- ***Diagrama de clases del análisis del caso de uso "Autentificar Wiki".***

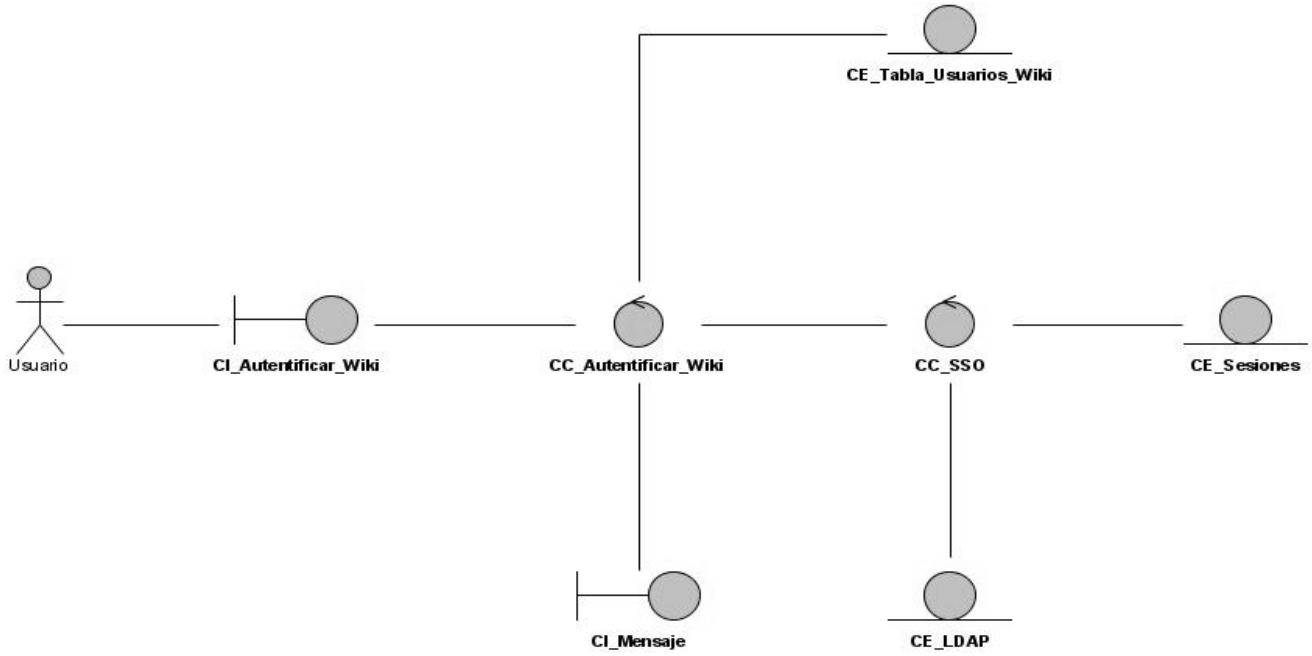

Fig.15: Diagrama de clases del análisis. CU-Autentificar Wiki.

- ***CI_Autentificar_Wiki:***

Es la clase de interfaz que representa un sistema complejo independiente (Wiki) que interactúan con la aplicación SSO. El usuario se autentifica mediante ella.

- ***CC_Autentificar_Wiki:***

Es la clase de control, que se encarga realizar las operaciones de autentificación en el Wiki, para ello accede de forma directa a la tabla de usuarios registrados en dicho subsistema Web y se relaciona con una clase de control (CC_SSO) que verifica la existencia de el usuario en el directorio activo de usuarios del sistema (LDAP).

- ***CE_Tabla_Usuarios_Wiki:***

Clase entidad que es una tabla de la base de datos del subsistema Web Wiki, donde se guarda y consulta la información del usuario autentificado.

3.1.2- Diagramas de clases del análisis de los casos de uso del paquete de actualización:

- ***Diagrama de clases del análisis del caso de uso "Actualizar GForge".***

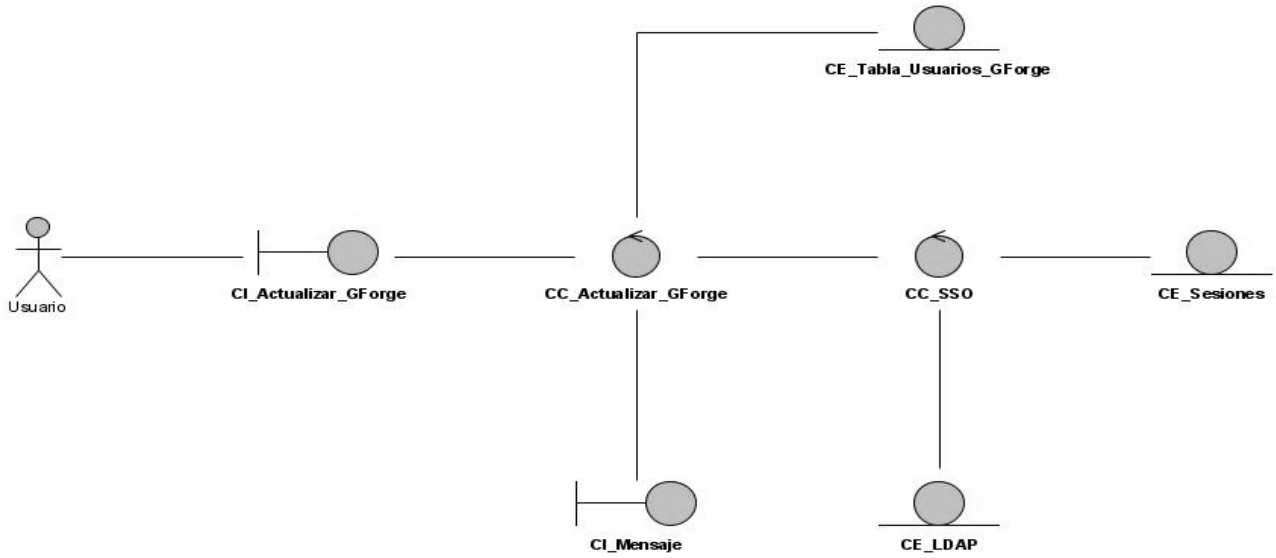

Fig.16: Diagrama de clases del análisis. CU-Actualizar GForge.

- ***CI_Actualizar_GForge:***

Clase que provee la interfaz gráfica necesaria para la actualización de datos del usuario desde el subsistema Web GForge.

- ***CC_Actualizar_GForge:***

Clase de control que manipula la información necesaria para efectuar la actualización del usuario en la base de datos del subsistema Web GForge y establece la relación con la clase de control CC_SSO quien actualiza, a su vez, los nuevos datos en el directorio activo de usuarios (LDAP).

- ***Diagrama de clases del análisis del caso de uso "Actualizar Drupal".***

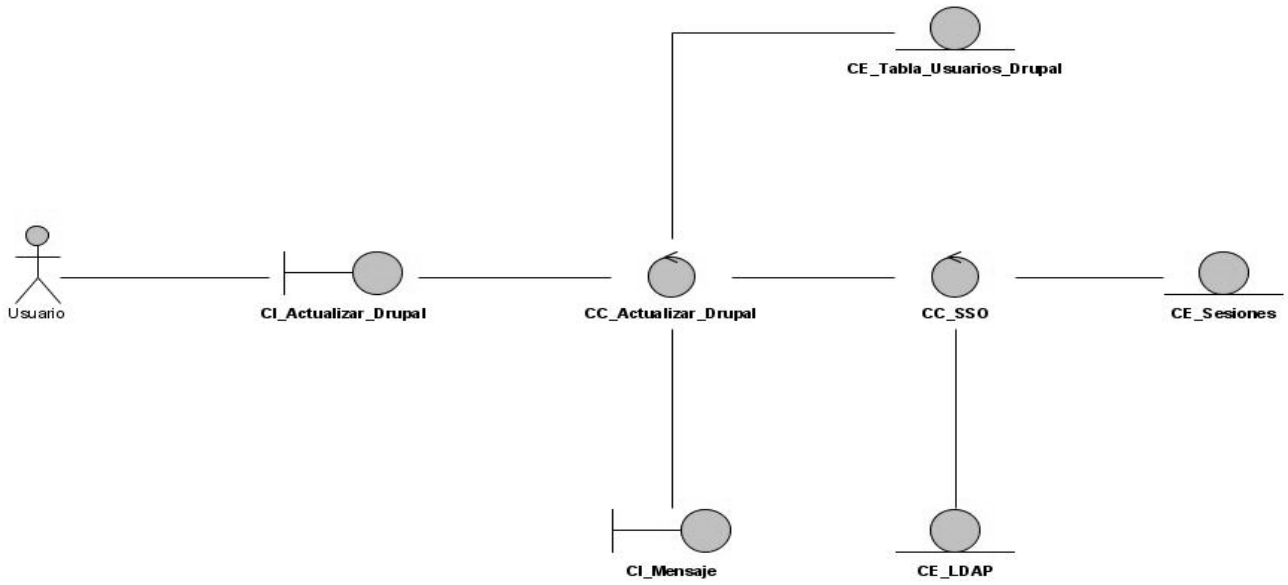

Fig.17: Diagrama de clases del análisis. CU-Actualizar Drupal.

- ***CI_Actualizar_Drupal:***

Clase que provee la interfaz gráfica necesaria para la actualización de datos del usuario desde el subsistema Web Drupal.

- ***CC_Actualizar_Drupal:***

Clase de control que manipula la información necesaria para efectuar la actualización del usuario en la base de datos del subsistema Web Drupal y establece la relación con la clase de control CC_SSO quien actualiza, a su vez, los nuevos datos en el directorio activo de usuarios (LDAP).

❖ ***Diagrama de clases del análisis del caso de uso "Actualizar Wiki".***

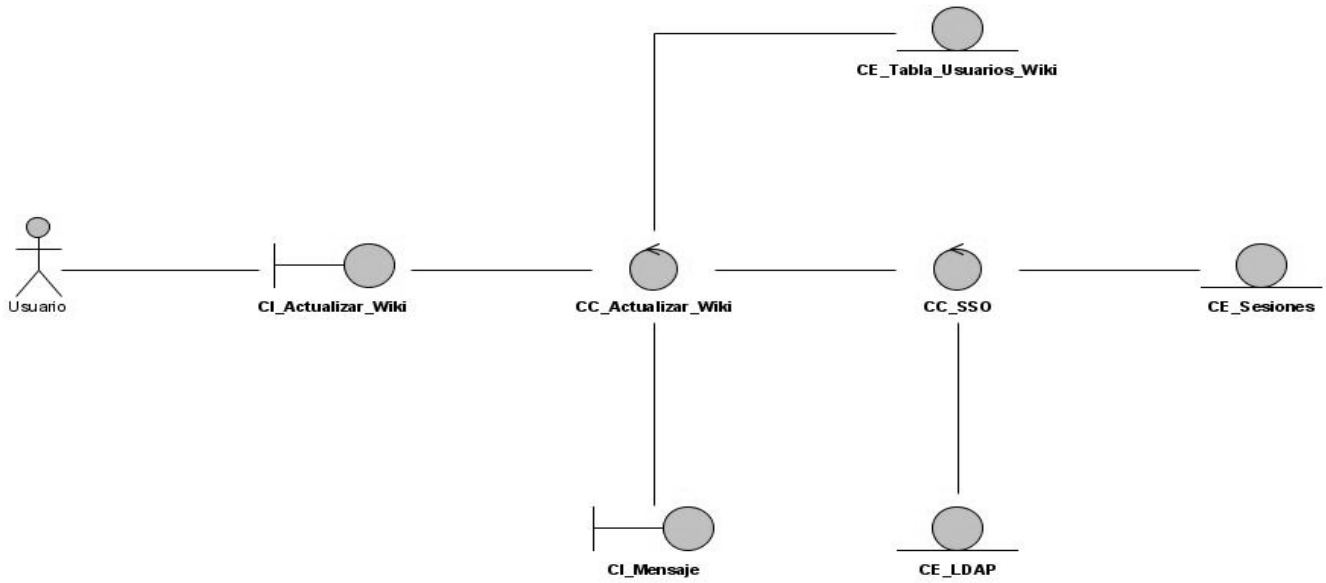

Fig.18: Diagrama de clases del análisis. CU-Actualizar Wiki.

- ***CI_Actualizar_Wiki:***

Clase que provee la interfaz gráfica necesaria para la actualización de datos del usuario desde el subsistema Web Wiki.

- ***CC_Actualizar_Wiki:***

Clase de control que manipula la información necesaria para efectuar la actualización del usuario en la base de datos del subsistema Web Wiki y establece la relación con la clase de control CC_SSO quien actualiza, a su vez, los nuevos datos en el directorio activo de usuarios (LDAP).

3.1.3- Diagramas de clases del análisis de los casos de uso del paquete de registro:

- ***Diagrama de clases del análisis del caso de uso "Registrar GForge".***

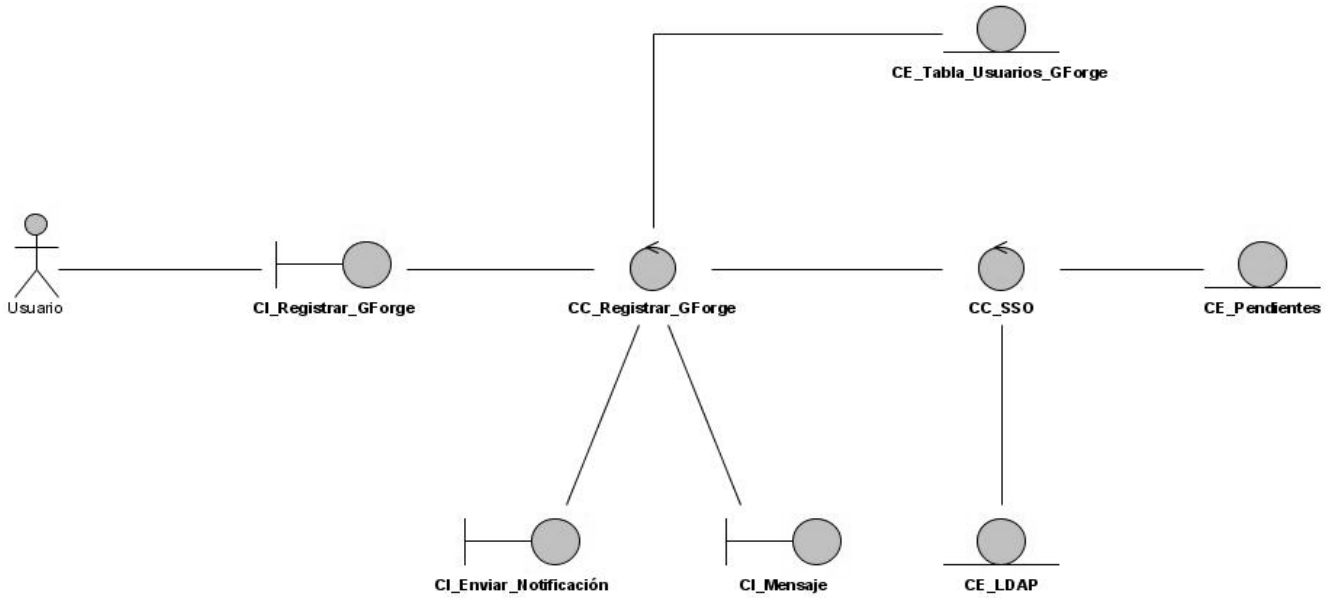

Fig.19: Diagrama de clases del análisis. CU-Registrar GForge.

- ***CI_Registrar_GForge:***

Clase que provee la interfaz gráfica necesaria para el registro de datos del usuario desde el subsistema Web GForge.

- ***CC_Registrar_GForge:***

Clase de control que manipula la información necesaria para efectuar el registro del usuario en la base de datos del subsistema Web GForge y establece la relación con la clase de control CC_SSO quien registra, a su vez, los nuevos datos en el directorio activo de usuarios (LDAP).

- ***CI_Enviar_Notificación:***

Clase interfaz que le brinda al usuario, que de forma exitosa pudo registrarse en el sistema, la notificación del envió de un correo electrónico.

- ***CE_Pendientes:***

Clase entidad que va a ser una tabla de la base de datos del SSO, donde se guardará y consultará la información de las activaciones de cuentas de usuarios.

❖ ***Diagrama de clases del análisis del caso de uso "Registrar Drupal".***

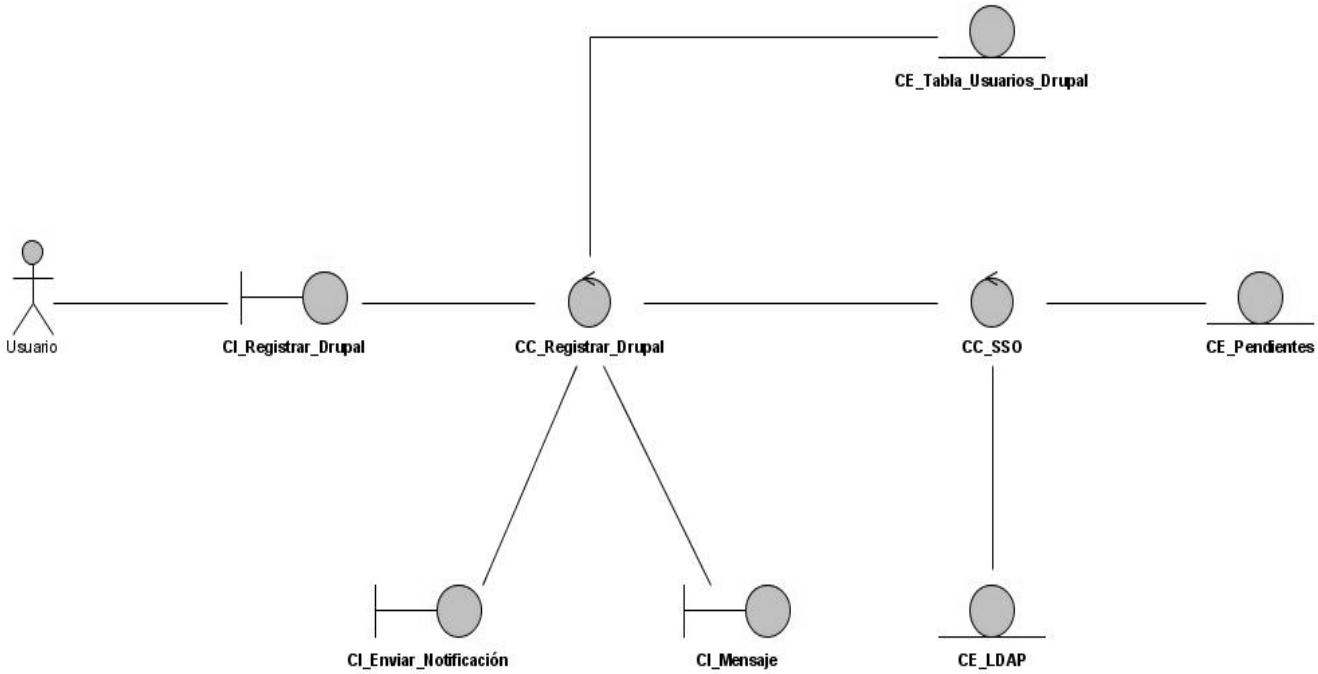

Fig.20: Diagrama de clases del análisis. CU-Registrar Drupal.

- ***CI_Registrar_Drupal:***

Clase que provee la interfaz gráfica necesaria para el registro de datos del usuario desde el subsistema Web Drupal.

- ***CC_Registrar_Drupal:***

Clase de control que manipula la información necesaria para efectuar el registro del usuario en la base de datos del subsistema Web Drupal y establece la relación con la clase de control CC_SSO quien registra, a su vez, los nuevos datos en el directorio activo de usuarios (LDAP).

❖ ***Diagrama de clases del análisis del caso de uso "Registrar Wiki".***

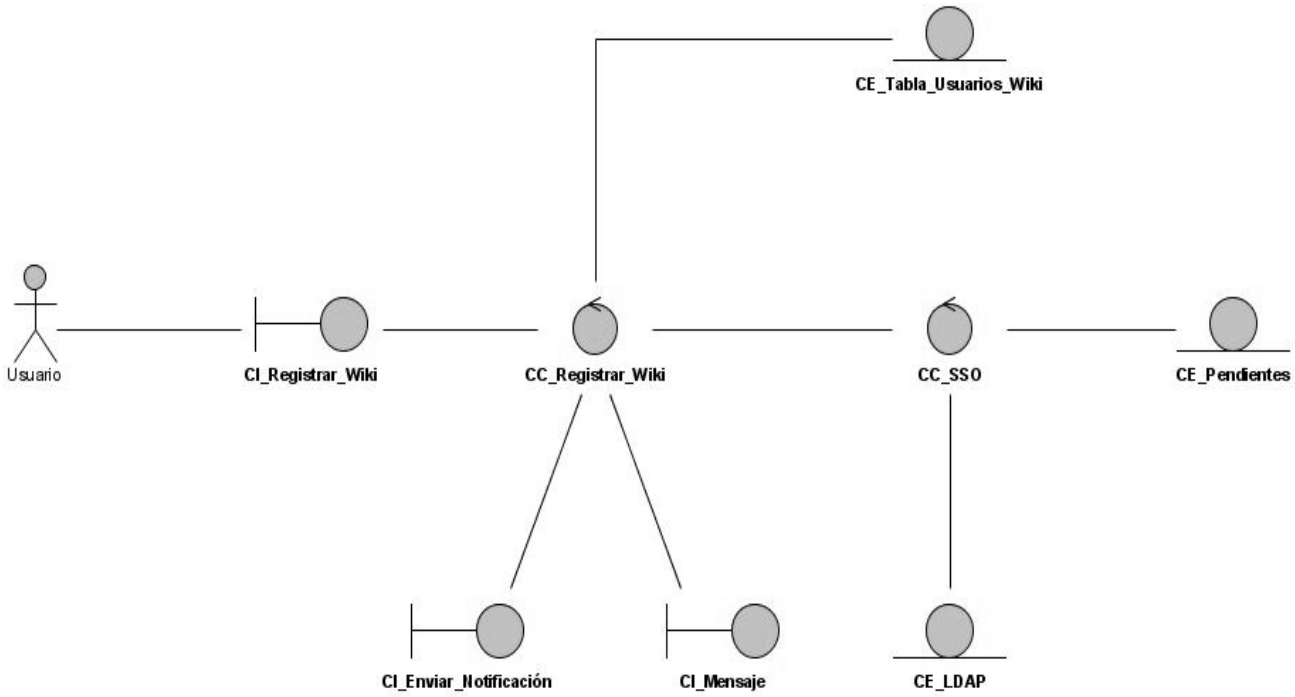

Fig.21: Diagrama de clases del análisis. CU-Registrar Wiki.

- ***CI_Registrar_Wiki:***

Clase que provee la interfaz gráfica necesaria para el registro de datos del usuario desde el subsistema Web Wiki.

- ***CC_Registrar_Wiki:***

Clase de control que manipula la información necesaria para efectuar el registro del usuario en la base de datos del subsistema Web Wiki y establece la relación con la clase de control CC_SSO quien registra, a su vez, los nuevos datos en el directorio activo de usuarios (LDAP).

3.1.4- Diagramas de clases del análisis de los casos de uso del paquete de administración:

- ***Diagrama de clases del análisis del caso de uso "Autentificar Administrador".***

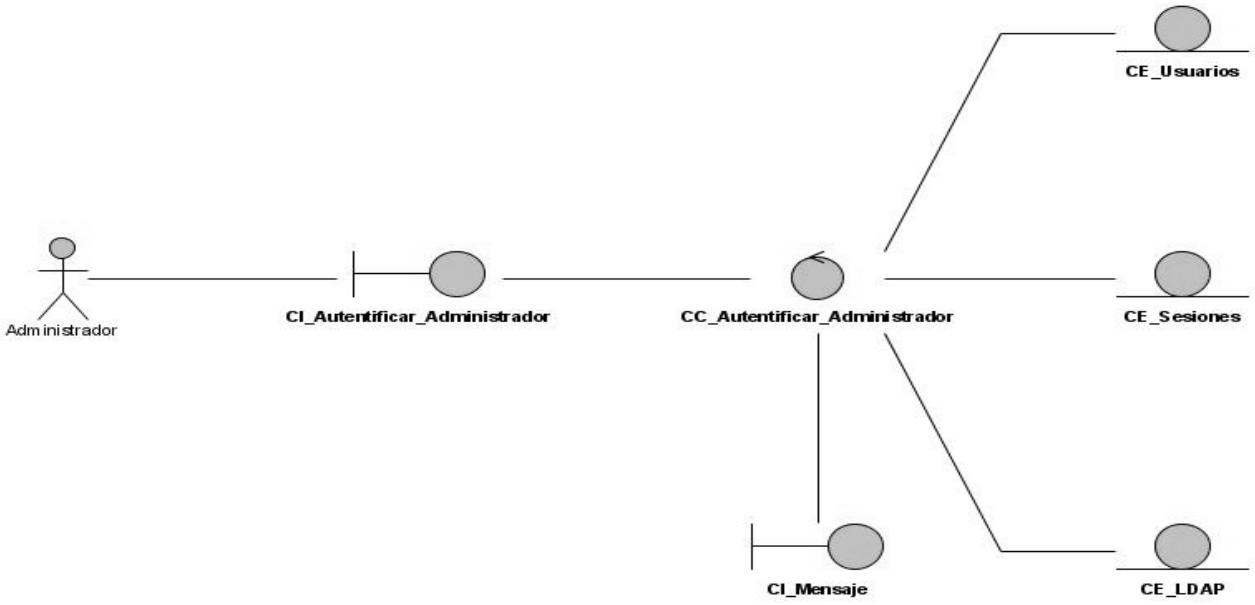

Fig.22: Diagrama de clases del análisis. CU-Autentificar Administrador.

- ***Administrador:***

Actor que interactuará directamente con la interfaz de administración del SSO.
Es el responsable de iniciar los casos de uso:

- Autentificar Administrador.
- Actualizar Administrador.
- Crear Administrador.
- Agregar Nombre de Dominio.
- Agregar IP.
- Actualizar Nombre de Dominio.
- Actualizar IP.
- Eliminar Nombre de Dominio.
- Eliminar IP
- Consultar direcciones IP

- Consultar Nombre de Dominio.

- ***CI_Autentificar_Administrador:***

Clase que provee la interfaz gráfica necesaria para la autentificación del administrador en el sistema SSO.

- ***CC_Autentificar_Administrador:***

Es la clase de control, que se encarga de realizar las operaciones de autentificación a través del sistema SSO, para ello accede de forma directa a la tabla de usuarios registrados en el sistema SSO y al directorio activo de usuarios del sistema (LDAP). Manipula los datos del administrador en el proceso de su autentificación.

- ***CE_Usuarios:***

Clase entidad que va a representar una tabla de la base de datos del SSO, donde se guardará y consultará la información del administrador.

- ***CE_LDAP:***

Clase entidad que guardará datos de los usuarios de todo el sistema, incluyendo el administrador, en la base de datos central. Directorio activo de usuarios.

- ***Diagrama de clases del análisis del caso de uso "Crear Administrador".***

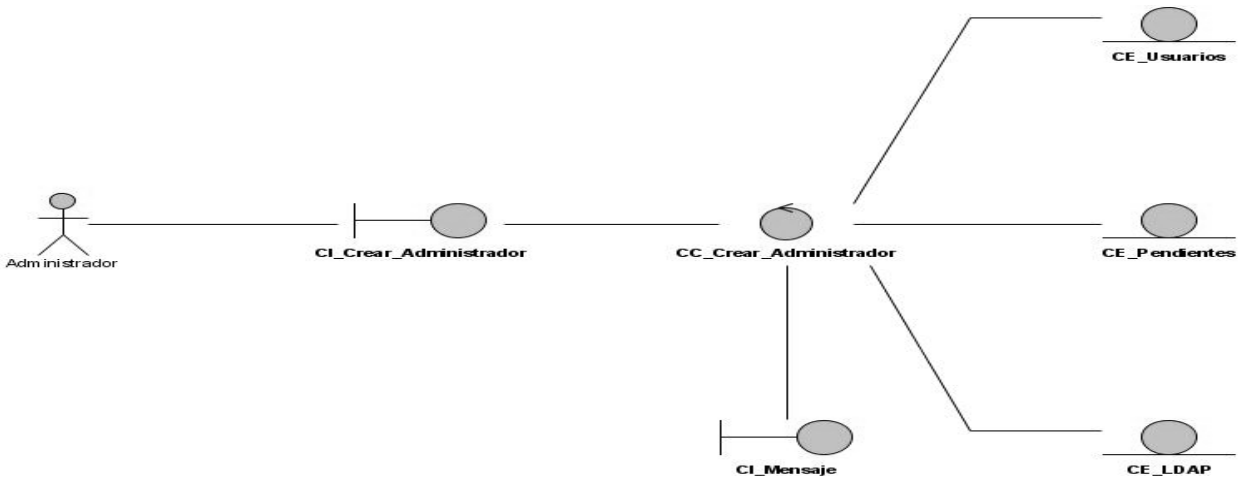

Fig.23: Diagrama de clases del análisis. CU-Crear Administrador.

- ***CI_Crear_Administrador:***

Clase que provee la interfaz gráfica necesaria para la ceración de una cuenta de administración en el sistema SSO. Esta interfaz es construida durante la instalación del SSO.

- ***CC_Crear_Administrador:***

Es la clase de control, que se encarga de realizar las operaciones de creación cuentas de administración a través del sistema SSO, para ello accede de forma directa a la tabla de usuarios registrados en el sistema SSO y al directorio activo de usuarios del sistema (LDAP). Manipula los datos del administrador en el proceso de su creación.

❖ ***Diagrama de clases del análisis del caso de uso "Actualizar Administrador".***

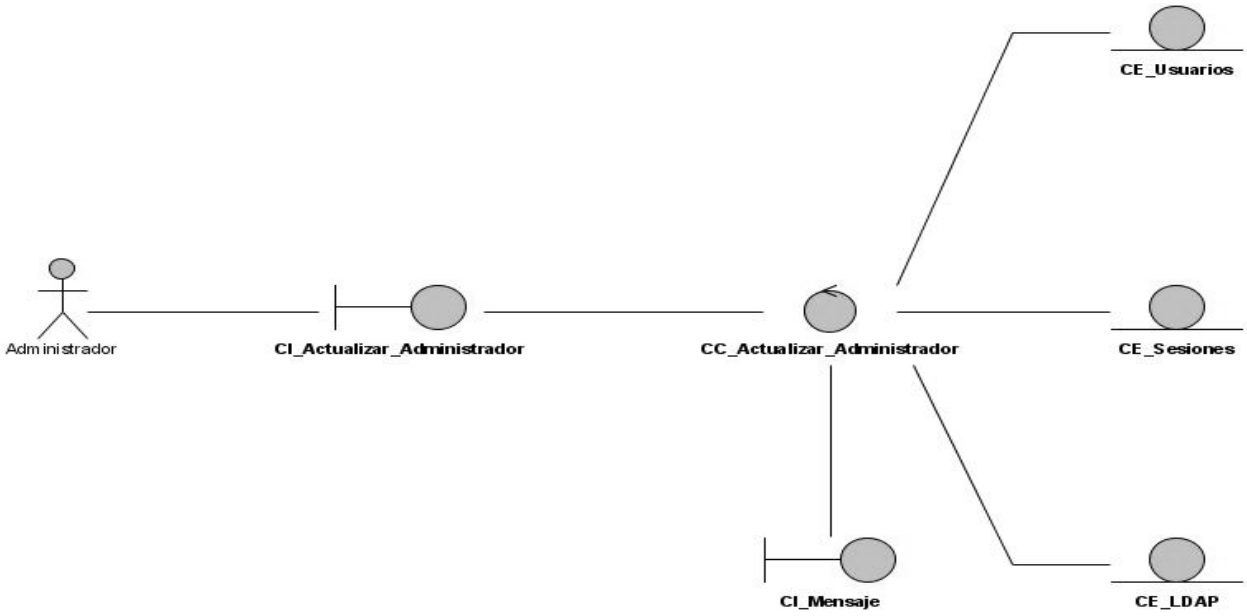

Fig.24: Diagrama de clases del análisis. CU-Actualizar Administrador.

- ***CI_Actualizar_Administrador:***

Clase que provee la interfaz gráfica necesaria para la actualización de una cuenta de administración en el sistema SSO. Esta interfaz es construida durante la instalación del SSO.

- ***CC_Actualizar_Administrador:***

Es la clase de control, que se encarga de realizar las operaciones de actualización de cuentas de administración a través del sistema SSO, para ello accede de forma directa a la tabla de usuarios registrados en el sistema SSO y al directorio activo de usuarios del sistema (LDAP). Manipula los datos del administrador en el proceso de su actualización.

❖ ***Diagrama de clases del análisis del caso de uso "Gestionar Nombre de Dominio".***

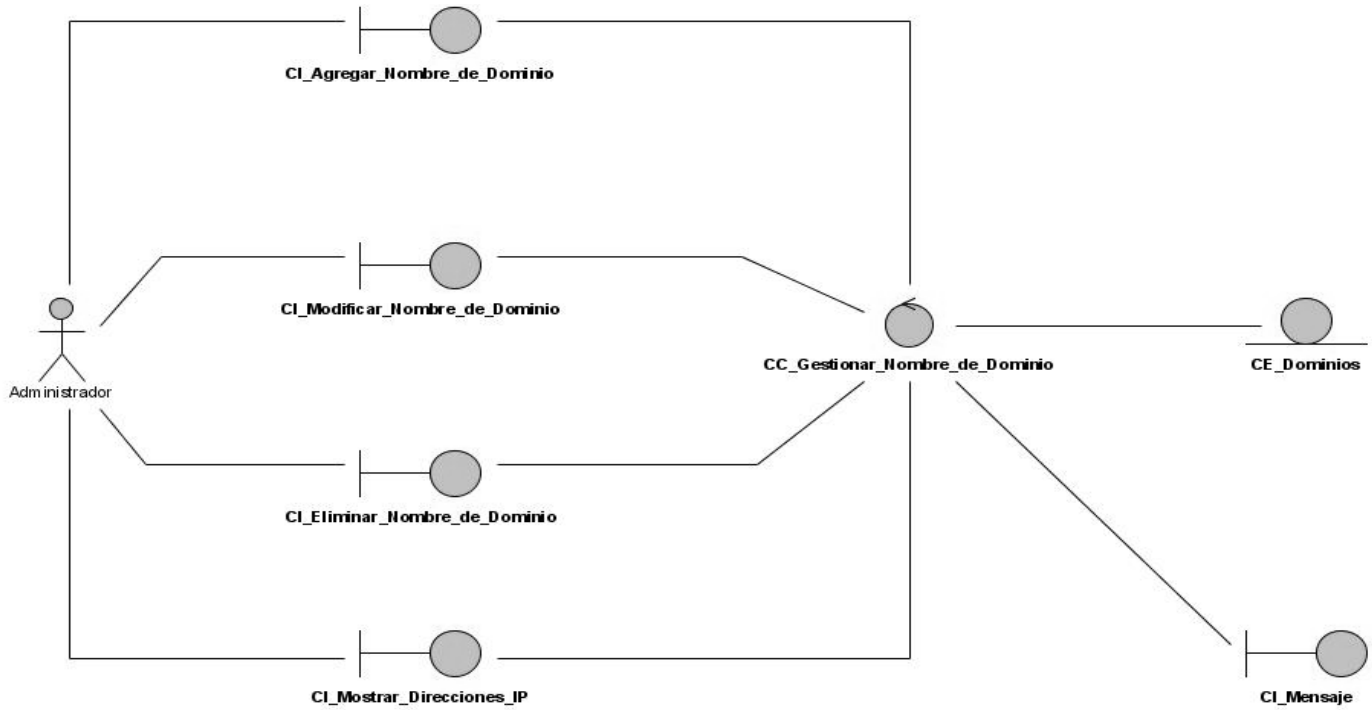

Fig.25: Diagrama de clases del análisis. CU-Gestionar Nombre de Dominio.

- ***CI_Agregar_Nombre_de_Dominio:***

Clase interfaz mediante la cual el administrador registra, los nombre de dominio con que trabaja el SSO.

- ***CI_Modificar_Nombre_de_Dominio:***

Clase interfaz mediante la cual el administrador modifica los datos de nombres de dominio con que trabaja el SSO.

- ***CI_Eliminar_Nombre_de_Dominio:***

Clase interfaz mediante la cual el administrador elimina nombres de dominio con que trabaja el SSO.

- ***CI_Mostrar_Direcciones_IP:***

Clase interfaz mediante la cual el administrador obtiene un listado de todas las direcciones IP que forman parte de un nombre de dominio, escogido previamente, con que trabaja el SSO.

- ***CC_Gestionar_Nombre_de_Dominio:***

Es la clase de control, que se encarga de realizar las operaciones de gestión de nombres de dominio (agregar, actualizar, eliminar, mostrar), para ello accede de forma directa a la tabla de nombres de dominios registrados en el sistema SSO. Manipula los datos de nombres de dominio en el proceso de su gestión.

- ***CE_Dominios:***

Clase entidad que representa una tabla de la base de datos del SSO, donde se guarda y consulta la información referente a los nombres de dominio autorizados en el sistema.

❖ ***Diagrama de clases del análisis del caso de uso "Gestionar IP".***

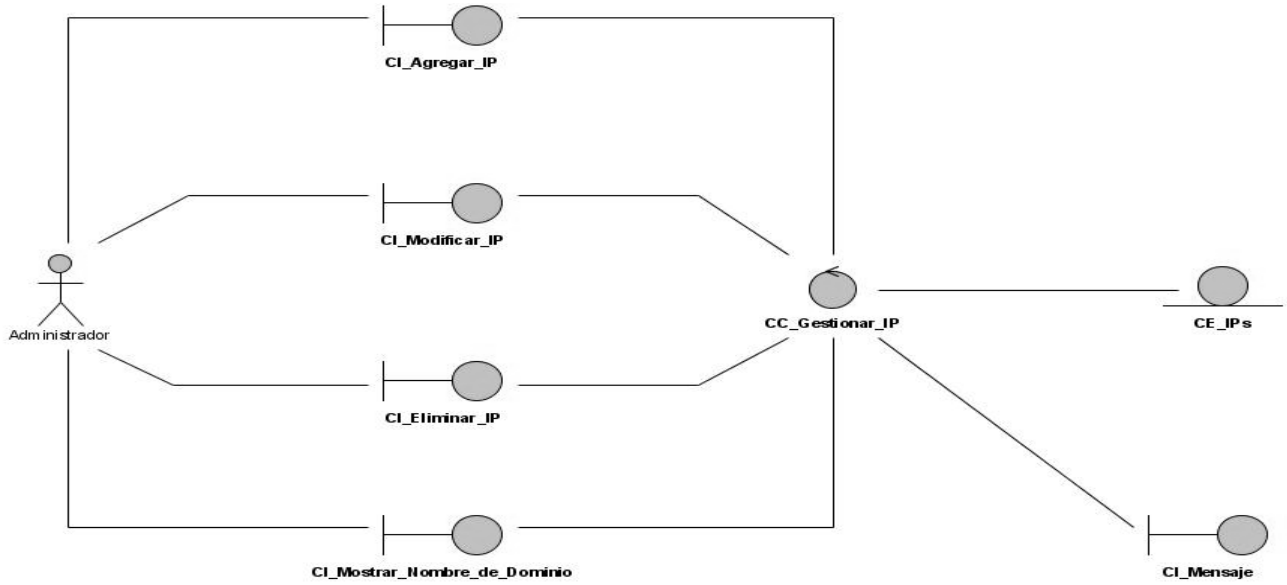

Fig.26: Diagrama de clases del análisis. CU-Gestionar IP.

- ***CI_Agregar_Nombre_de_Dominio:***

Clase interfaz mediante la cual el administrador registra, las direcciones IP con que trabaja el SSO.

- ***CI_Modificar_Nombre_de_Dominio:***

Clase interfaz mediante la cual el administrador modifica los datos de las direcciones IP con que trabaja el SSO.

- ***CI_Eliminar_Nombre_de_Dominio:***

Clase interfaz mediante la cual el administrador elimina direcciones IP con que trabaja el SSO.

- ***CI_Mostrar_Direcciones_IP:***

Clase interfaz mediante la cual el administrador obtiene el nombre de dominio al cual pertenece la dirección IP, escogida previamente, que se quiere consultar.

- ***CC_Gestionar_Nombre_de_Dominio:***

Es la clase de control, que se encarga de realizar las operaciones de gestión de las direcciones IP (agregar, actualizar, eliminar, mostrar), para ello accede de forma directa a la tabla de direcciones IP registradas en el sistema SSO. Manipula los datos las direcciones IP en el proceso de su gestión.

- ***CE_Tabla_IP:***

Clase entidad que representa una tabla de la base de datos del SSO, donde se guarda y consulta la información referente a las direcciones de IP pertenecientes a los dominio autorizados con que trabaja el sistema.

3.2- Modelo de Diseño del SSO.

El modelo de diseño de un sistema:

- Describe la realización física de los casos de uso.
- Se centra en como los requisitos funcionales y no funcionales tienen impacto en el sistema.

RUP propone que el artefacto *Modelo de Diseño* básicamente contenga:

- **Introducción:** Una descripción textual que sirve como breve introducción al modelo.

- **Paquetes y Subsistemas de Diseño:** Los paquetes y subsistemas del diseño representados en una jerarquía y una breve descripción de ellos.
- **Diagramas:** los diagramas de clases del diseño y diagramas de interacción (colaboración y/o secuencia) del diseño. Estos últimos también llamados realización de casos de uso.
- **Clases, interfaces, relaciones**, contenidas en los paquetes y una breve descripción de ellos.[14]

3.2.1- Diagramas de interacción:

Los diagramas de interacción muestran una interacción concreta: un conjunto de objetos y sus relaciones, junto con los mensajes que se envían entre ellos.

- Modelan el comportamiento dinámico del sistema; el flujo de control en una operación.
- Describen la interacción entre objetos; los objetos interactúan a través de mensajes para cumplir ciertas tareas.
- Las interacciones proveen un «comportamiento» y típicamente implementan un caso de uso.

Existen dos tipos de diagramas de interacción en UML:

- Diagramas de Secuencia (dimensión temporal).
- Diagramas de Colaboración (dimensión estructural). [15]

3.2.1.1- Diagramas de Colaboración:

Los diagramas de colaboración muestran las interacciones que ocurren entre los objetos que participan en una situación determinada. Esta es, más o menos, la misma información mostrada por los diagramas de secuencia, pero destacando la forma en que las operaciones se producen en el tiempo. Los diagramas de colaboración fijan el interés en las relaciones entre los objetos y su topología.

- Son útiles en la fase exploratoria para identificar objetos.
- La distribución de los objetos en el diagrama permite observar adecuadamente la interacción de un objeto con respecto de los demás.
- La estructura estática viene dada por los enlaces; la dinámica por el envío de mensajes por los enlaces.

Por cada realización de casos de uso se realizan diagramas de interacción. Estos pueden escogerse entre diagramas de secuencia y diagramas de colaboración.

Los diagramas de colaboración orientados a la realización del SSO se encuentran en los anexos de este trabajo *(Ver Anexo 2)*.

3.2.2- Diagramas de clases del diseño:

Los diagramas de clases son los más utilizados en el modelado de sistemas orientados a objetos.
Un diagrama de clases muestra un conjunto de clases, interfaces y colaboraciones, así como sus relaciones.
Para la modelación de la solución se proponen como clases del diseño los estándares para modelar aplicaciones Web.

- **Página servidora:** Representa la página Web que tiene código que se ejecuta en el servidor. Este código interactúa con recursos en el servidor. Las operaciones representan las funciones del código y los atributos las variables visibles dentro del alcance de la página.
- **Página cliente:** Una instancia de página cliente es una página Web, con formato HTML. Mezcla de datos, presentación y lógica. Son interpretadas por el navegador. Sus atributos son las variables declaradas dentro del *script* que son accesibles para cualquier función dentro de la página.
- **Formulario:** Colección de elementos de entrada que son parte de una página cliente. Se relaciona directamente con la etiqueta de igual nombre del HTML. El diseño de sistemas se ocupa de desarrollar las directrices propuestas durante el análisis en términos de aquella configuración que tenga más posibilidades de satisfacer los objetivos planteados tanto desde el punto de vista funcional como del no funcional. [16]

En esta tesis se elaboró un diagrama de clases Web para cada uno de los casos de uso del sistema *(Ver Anexo 3).*

3.2.3- Diseño de la base de datos:

El diagrama Entidad-Relación es un mecanismo formal de representar y manipular información de manera general y sistemática. Refleja tan solo la existencia de los datos, no lo que se hace con ellos, es independiente del sistema operativo y del gestor de base de datos en que se empleen posteriormente. Es independiente de las restricciones de almacenamiento y de tiempo de ejecución, sus elementos son entidades, atributos y relaciones. [17]
El diagrama Entidad-Relación que representa el diseño de la base de datos del SSO se encuentra ilustrado en los anexos de este trabajo *(Ver Anexo 4)*.

3.2.3.1- Descripción de las tablas:

Tabla 16: Descripción de la tabla Dominios de la base de datos.

Nombre: Dominios		
Descripción: Almacena los datos de nombres de dominio con que trabaja el SSO.		
Atributo	**Tipo**	**Descripción**
iddominio	int(11)	Identificador del nombre de dominio
dominio	varchar(254)	Nombre de dominio

Tabla 17: Descripción de la tabla IPs de la base de datos.

Nombre: IPs		
Descripción: Almacena los datos de las direcciones IP con que trabaja el SSO.		
Atributo	**Tipo**	**Descripción**
idip	int(11)	Identificador de la dirección IP.
iddominio	int(11)	Identificador del nombre de dominio al que pertenece dicha dirección IP.
ip	varchar(40)	Dirección IP.

Tabla 18: Descripción de la tabla Peticiones de la base de datos.

Nombre: Peticiones		
Descripción: Almacena los datos de las peticiones de cambio de clave.		
Atributo	**Tipo**	**Descripción**
idpeticion	int(11)	Identificador de la petición.
usuario	varchar(254)	Usuario que hace la petición
fecha	datetime	Fecha en que se hace la petición.
code	varchar(254)	Código que se le asigna.
confirmada	int(1)	Confirmación de la petición.
fecha_c	timestamp	Fecha en que se realiza la confirmación.

Tabla 19: Descripción de la tabla Pendientes de la base de datos.

Nombre: Pendientes		
Descripción: Almacena los datos del proceso de activación de cuentas de usuarios.		
Atributo	**Tipo**	**Descripción**
idpendiente	int(11)	Identificador de la activación pendiente.
usuario	varchar(254)	Usuario pendiente por activación.
keycode	varchar(254)	Código de activación que se le asigna.
confirmada	int(1)	Confirmación de la activación de cuenta.
fecha_confirmacion	datetime	Fecha en que se realiza la confirmación.

Tabla 20: Descripción de la tabla Sesiones de la base de datos.

Nombre: Sesiones		
Descripción: Almacena los datos de sesiones de usuarios.		
Atributo	**Tipo**	**Descripción**
idsesion	int(11)	Identificador de la sesión.
ip	varchar(40)	Dirección IP de donde se inicia la sesión de cara al sistema.
ip_x	varchar(40)	Dirección IP de donde se inicia la sesión detrás del Proxy, en caso de existir.
hash	varchar(254)	Token único de identificación del usuario.
time	varchar(254)	Momento en que se activa la sesión.
user_agent	varchar(254)	Navegador por el que se accede.
username	varchar(254)	Nombre de usuario.
password	varchar(254)	Calve del usuario
last_site	varchar(254)	Último sitio visitado.
last_time	timestamp	Hora de la última acción realizada.

Tabla 21: Descripción de la tabla Usuarios de la base de datos.

Nombre: Usuarios		
Descripción: Almacena los datos de los administradores del SSO.		
Atributo	**Tipo**	**Descripción**
idusuario	int(11)	Identificador del usuario.
usuario	varchar(254)	Identificador del usuario del sistema.
clave	varchar(254)	Clave del usuario del sistema.
nombres	varchar(254)	Nombre del usuario
apellidos	varchar(254)	Apellidos del usuario.
fecha_creacion	datetime	Fecha de registro.
fecha_ultima_m	timestamp	Fecha de la última modificación de los datos.
ip	varchar(40)	Dirección de IP de acceso.

En este capitulo se establecieron las pautas de diseño necesarias para la construcción del sistema obtenida en el capítulo anterior. Se modelaron los diagramas de clases del análisis correspondiente a cada caso de uso del sistema, diagramas de interacción (del tipo colaboración en este caso) además de las clases (Web) del diseño.

Para la construcción de estos diagramas se utilizaron patrones de diseño para mejorar la calidad del software. Con éstos quedan sentadas las bases para el trabajo en la implementación del SSO de RINDE.

CONCLUSIONES.

Tras una intensa investigación, la cual devino en la tarea de recopilar información de sistemas (SSO) semejantes a este pero con distintas funcionalidades, se llegó a la conclusión que esta era la vía correcta para la solución inmediata del problema existente: la autentificación centralizada e integración de sistemas Web independientes.

La realización de este proyecto trajo consigo la fusión de un gran sistema Web (RINDE) que contaba con tres subsistemas independientes: Un portal Web informativo (Drupal), una herramienta principal para el desarrollo colaborativo de aplicaciones libres (GForge) y un medio de documentación constante y libre para los usuarios (Wiki), logrando un alto nivel de seguridad y coherencia de datos.
Es un trabajo que se pone en las manos de todos los que consideran que la seguridad es parte fundamental del mundo de la informática hoy en día.

1. El proyecto RINDE, en su primera etapa, provee una solución de software para la instalación y mantenimiento de herramientas de desarrollo colaborativo (Drupal, GForge y Wiki), personalizadas para el uso del gobierno venezolano.
2. Single Sign-On (SSO) es un procedimiento de autentificación que habilita al usuario para acceder a varios sistemas con una sola instancia de identificación.
3. En una segunda etapa se crea un SSO para unificar todo el sistema y garantizar un proceso fácil, fluido y con mayor grado de seguridad.
4. La *arquitectura de administración y almacenamiento de credenciales centralizados* fue por la que se optó para llevar a cabo la implementación de nuestro SSO.
5. Se obtuvo una lista de las funcionalidades que debe tener el software representadas mediante un diagrama de casos de uso dividido en paquetes, y se describieron todas las acciones que realizan los actores y el sistema en general.
6. Se realizaron los modelos de análisis y diseño del SSO.
7. Los modelos de análisis y de diseño realizados responden a las necesidades establecidas por el cliente.

RECOMENDACIONES.

Los objetivos propuestos para el presente trabajo de diploma han sido cumplidos satisfactoriamente, no obstante se incluyen tres recomendaciones a tomar en cuenta para futuras implementaciones de sistemas SSO.

- Crear un manual de usuario para garantizar el soporte a los administradores que trabajan en la configuración y mantenimiento del sistema.
- Aplicar el principio de la mejora continua a través del perfeccionamiento de las funcionalidades del sistema, a partir de las variaciones que puedan surgir en cuanto a las necesidades del cliente durante el tiempo de explotación.
- Implementar un sistema de este tipo en la Universidad de las Ciencias Informáticas para lograr una mayor coherencia, facilidad de uso y seguridad en los servicios Web acorde con los estándares internacionales.

REFERENCIAS BIBLIOGRÁFICAS.

[1]. Aladdin Knowledge Systems. Gestión de contraseñas. Single Sign On, 2006. [Disponible en: http://www.aladdin.es/news/2006/etoken/gestion_contrasenyas.asp].

[2]. Eliurkis Díaz Terrero. Single Sign On: Sistema de Autentificación Único, 2007. [Disponible en: http://www.deepinphp.com/2007/09/01/single-sign-on-sistema-de-autenticacion-unico/].

Single Sign-On, 2006. [Disponible en: http://es.wikipedia.org/wiki/Single_Sign-On].

[3]. Miqel Trilla. Single Sign-On, 2006.

[4]. Iván M. Caballero, Jeimy J. Cano. Consideraciones para Implementar una Arquitectura Single Sign-On, 2005.

[5]. P. Letelier. Rational Unified Process (RUP), 2008. [Disponible en: https://pid.dsic.upv.es/C1/Material/Documentos%20Disponibles/Introducci%C3%B3n%20a%20RUP.doc].

[6]. Modelo Vista Controlador (MVC). 2008. [Disponible en: http://es.wikipedia.org/wiki/Modelo_Vista_Controlador].

[7]. Ing. Febe Ángel Ciudad Ricardo - Ing. Ninet Soto López. [Conferencia 2], Fase de inicio. modelo del negocio, 2006.

[8]. Ing. Febe Ángel Ciudad Ricardo - Ing. Ninet Soto López. [Conferencia 3], Fase de inicio. Levantamiento de requisitos, 2006.

[9]. IDEM 6.

[10]. IDEM 6.

[11]. Andreas Kaiser. Software Libre. 2000. [Disponible en: http://www.atela.net/contenido/articulos/software-libre.html]

[12]. IDEM 9.

[13]. Ing. Febe Ángel Ciudad Ricardo - Ing. Ninet Soto López. [Conferencia 5], Fase de Elaboración. Análisis - Diseño, 2006.

[14]. IDEM 11.

[15]. Ferrer, F. Diseño de Sistemas 2006. [Disponible en: http://www.daedalus.es/inteligencia-de-negocio/sistemas-complejos/ingenieria-de-sistemas/diseno-de-sistemas/]

[16]. Pinciroli, F. Diagramas de Interacción *Di tutto il mondo*, 2001. Proceso Unificado de Rational.

[17]. IDEM 16.

BIBLIOGRAFÍA.

1. Bionetrix. Enterprise Single Sign-On: Balancing Security & Productivity, 2002.
2. Iván M. Caballero, Jeimy J. Cano. Consideraciones para Implementar una Arquitectura Single Sign-On, 2005.
3. McGraw, G. The Weakest Link, 2002.
4. Miqel Trilla. Single Sign-On, 2006.
5. Svoboda, Z. Securing Web Services with Single Sign-On, 2002.
6. Taylor, L. Understanding Single Sign-on, 1992.
7. Ing. Febe Ángel Ciudad Ricardo - Ing. Ninet Soto López. [Conferencia 5], Fase de Elaboración. Análisis - Diseño, 2006.

GLOSARIO DE TÉRMINOS.

- ***Alta disponibilidad:*** Característica que permite garantizar que un determinado sistema estará disponible cuando un usuario requiera utilizarlo.

- ***Autentificación:*** Llamamos autentificación a la comprobación de la identidad de una persona o de un objeto. La autentificación mediante identificador y contraseña es el sistema más común ya que viene incorporado en los sistemas operativos modernos de todos los ordenadores. *Autenticar:* este verbo se documenta ya en el español medieval y sigue plenamente vigente en el lenguaje legal y administrativo, especialmente en América. Con este mismo sentido se ha creado modernamente el verbo *autentificar*, que se considera también válido y es el usado con preferencia en el español de España y de buena parte de América.

- ***Balanceo de Carga:*** Característica que garantiza el eficiente manejo de una gran cantidad de solicitudes de los clientes en el sistema.

- ***Credenciales:*** Contienen información que le permite al usuario acceder a las diferentes aplicaciones (nombre de usuario, contraseña, etc.).

- ***CRM:*** Customer Relationship Management. Software para la administración de la relación con los clientes. Sistemas informáticos de apoyo a la gestión de las relaciones con los clientes, a la venta y al marketing.

- ***Encripción:*** Consiste en alterar la información, ya sea para ser almacenada o transmitida, utilizando alguna técnica de tal modo que únicamente podrá recuperar el contenido quien está autorizado para leerla o quien la cifró.

- ***E-SSO:*** El Enterprise Single Sign-On, o Legacy Single Sign-On, es un servicio que permite el mapeo de credenciales de seguridad entre Windows y otros sistemas heterogéneos, permitiendo que los usuarios puedan acceder diferentes aplicaciones con un solo conjunto de credenciales, particularmente en escenarios de Enterprise Application Integration (EAI).

- ***Framework:*** Es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Puede ser considerado como el conjunto de procesos y tecnologías usados para resolver un problema complejo. Es el esqueleto sobre el cual varios objetos son integrados para una solución dada.

- ***Gateway:*** Es un punto de acceso que relaciona dos o más puntos de un sistema.

- ***Kerberos:*** Es un protocolo de red que utiliza la criptografía simétrica para proporcionar autentificación para aplicaciones cliente-servidor.

- ***Latencia:*** La latencia de una red es sinónimo de retardo. Es una medida de cuanto tiempo le toma ir de un punto a otro, a un paquete de datos. En algunos casos se acostumbra definir la latencia como el tiempo que le toma a un paquete de datos recorrer el trayecto cerrado (ida hasta algún punto y regreso hasta su remitente). Los factores que contribuyen a la latencia de la red son: Propagación, Transmisión, Enrutamiento, Procesamiento, y algunos retardos asociados a otros dispositivos (switches, discos duros, etc.).

- ***OpenID:*** Es un sistema de identificación digital descentralizado, con el que un usuario puede identificarse en una página Web a través de una URL (o un XRI en la versión actual) y puede ser verificado por cualquier servidor que soporte el protocolo.

- ***Redundancia:*** Información replicada múltiples veces y actualizada en tiempo real con el fin de prevenir su pérdida por falla en el sistema o en los equipos.

- ***Reduced Sign On Systems:*** Sistemas de Autentificación Reducida. Es el segundo término de nombramiento que reciben los Sistemas de Autentificación Únicos o Single Sign-On (SSO). Son conocidos por ambos nombres.

- ***Rinde:*** Red de Integración y Desarrollo Nacional de Software Libre de la República Bolivariana de Venezuela.

- ***SSO:*** Sistema de Autentificación Única. Single Sign-On (SSO) es un procedimiento de autentificación que habilita al usuario para acceder a varios sistemas con una sola instancia de identificación.

- ***Visual Paradigm:*** Visual Paradigm es una suite CASE para el Lenguaje Unificado de Modelado (UML). Su conjunto de herramientas están diseñadas para una amplia gama de usuarios, incluidos los Ingenieros de Software, Analistas de Sistema, Analistas de Negocios, Arquitectos de Sistema, entre otros.

- ***Web-SSO:*** Web Single Sign-On, también llamado Web Access Management (Web-AM), trabaja solo con aplicaciones y recursos accedidos vía Web.

ANEXOS.

ANEXO 1: VISTA GENERAL DE LOS CASOS DE USO DEL SISTEMA.

Fig.27: Diagrama General de Casos de Uso del Sistema.

ANEXO 2: DIAGRAMAS DE COLABORACIÓN.

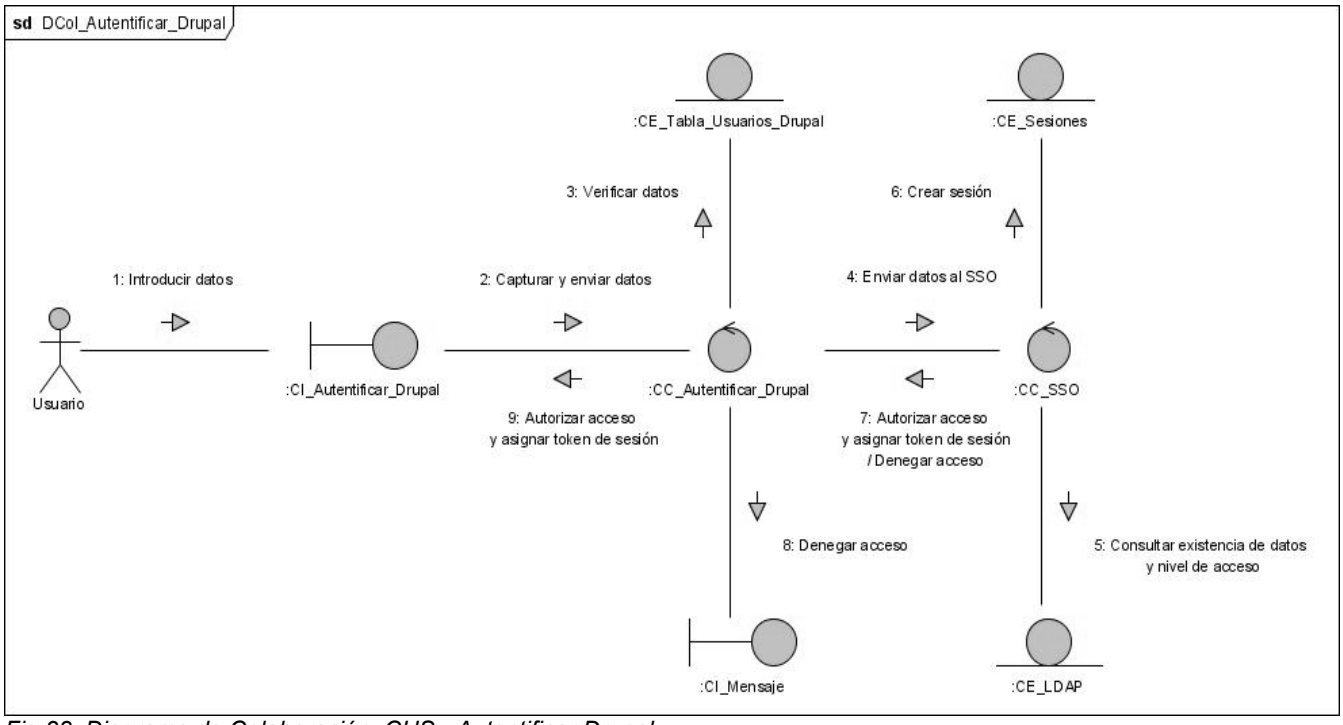

Fig.28: Diagrama de Colaboración. CUS - Autentificar Drupal.

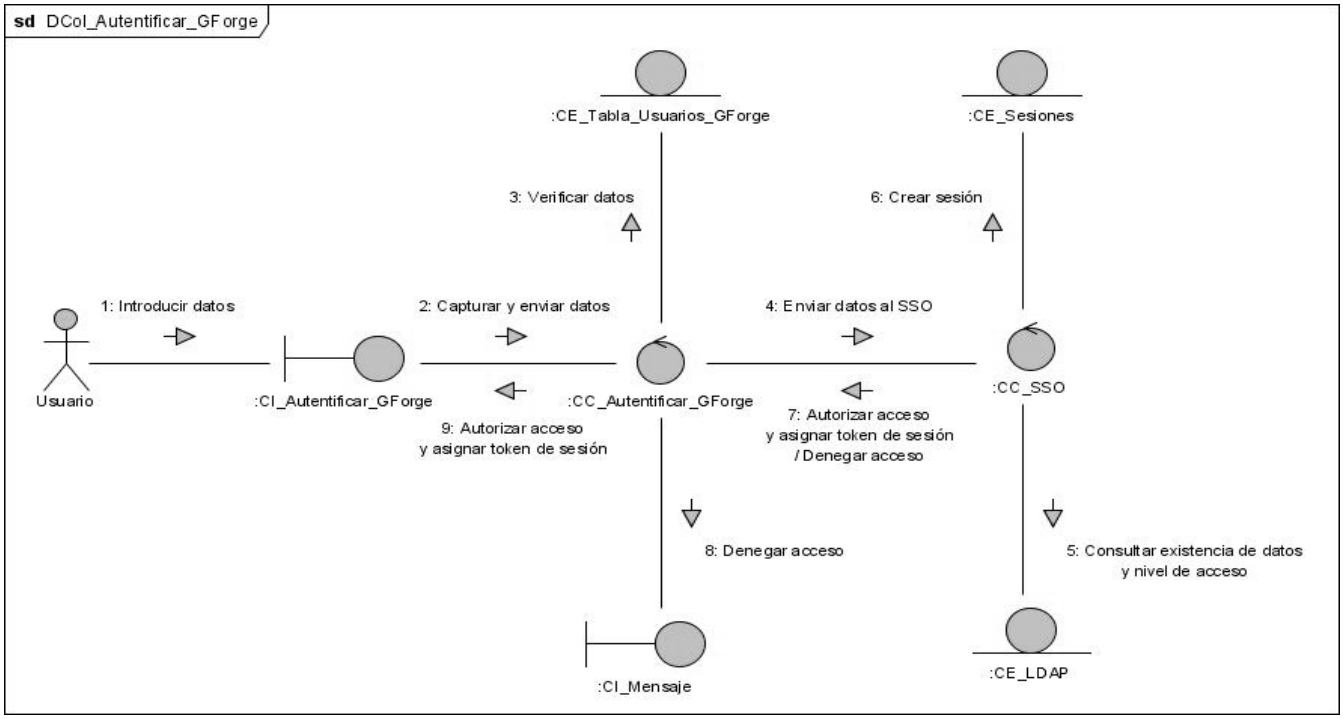

Fig.29: Diagrama de Colaboración. CUS - Autentificar GForge.

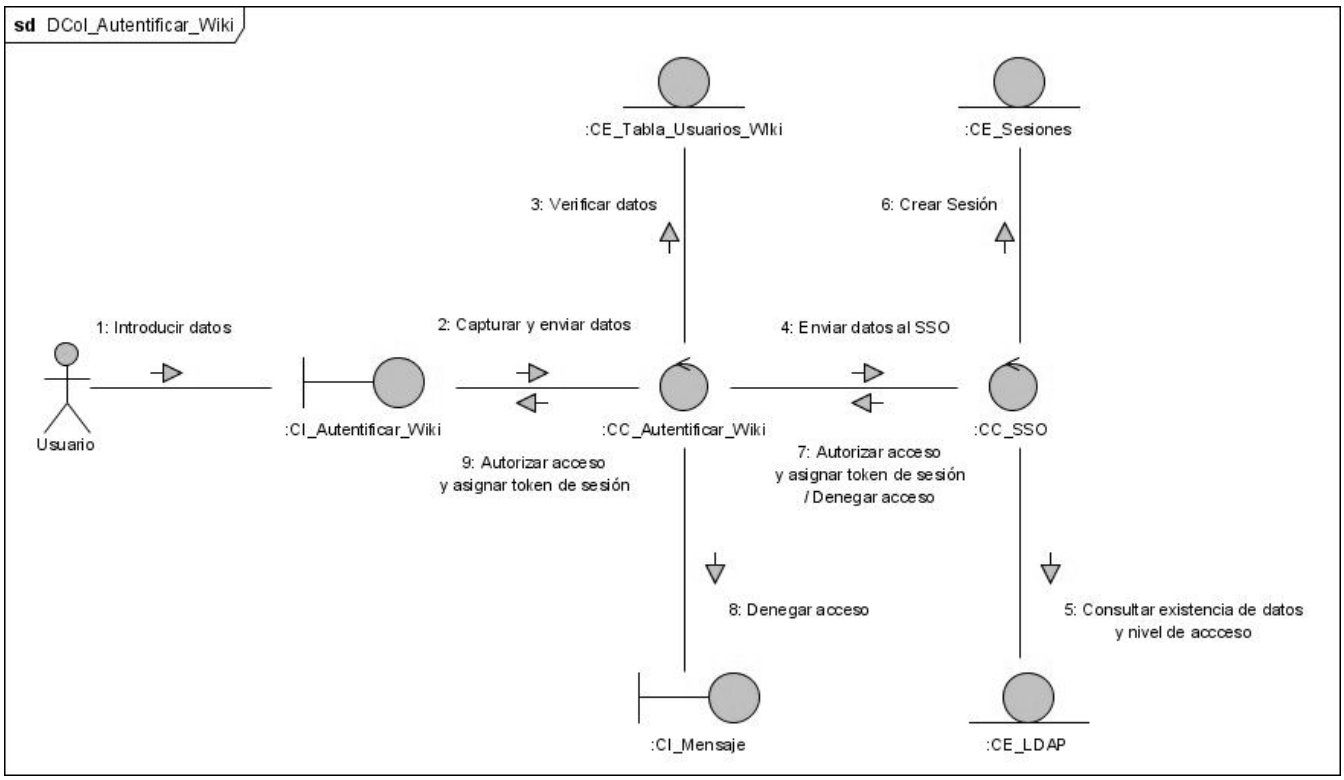

Fig.30: Diagrama de Colaboración. CUS - Autentificar Wiki.

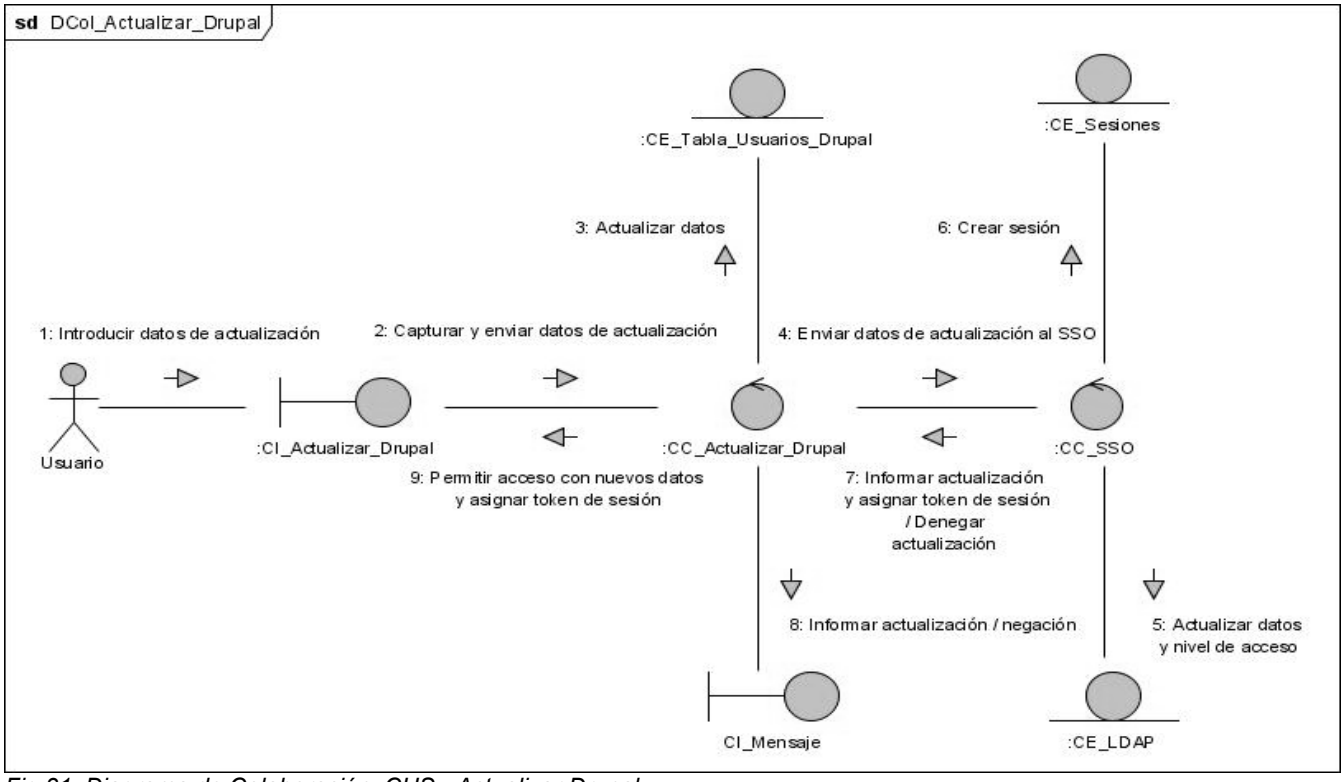

Fig.31: Diagrama de Colaboración. CUS - Actualizar Drupal.

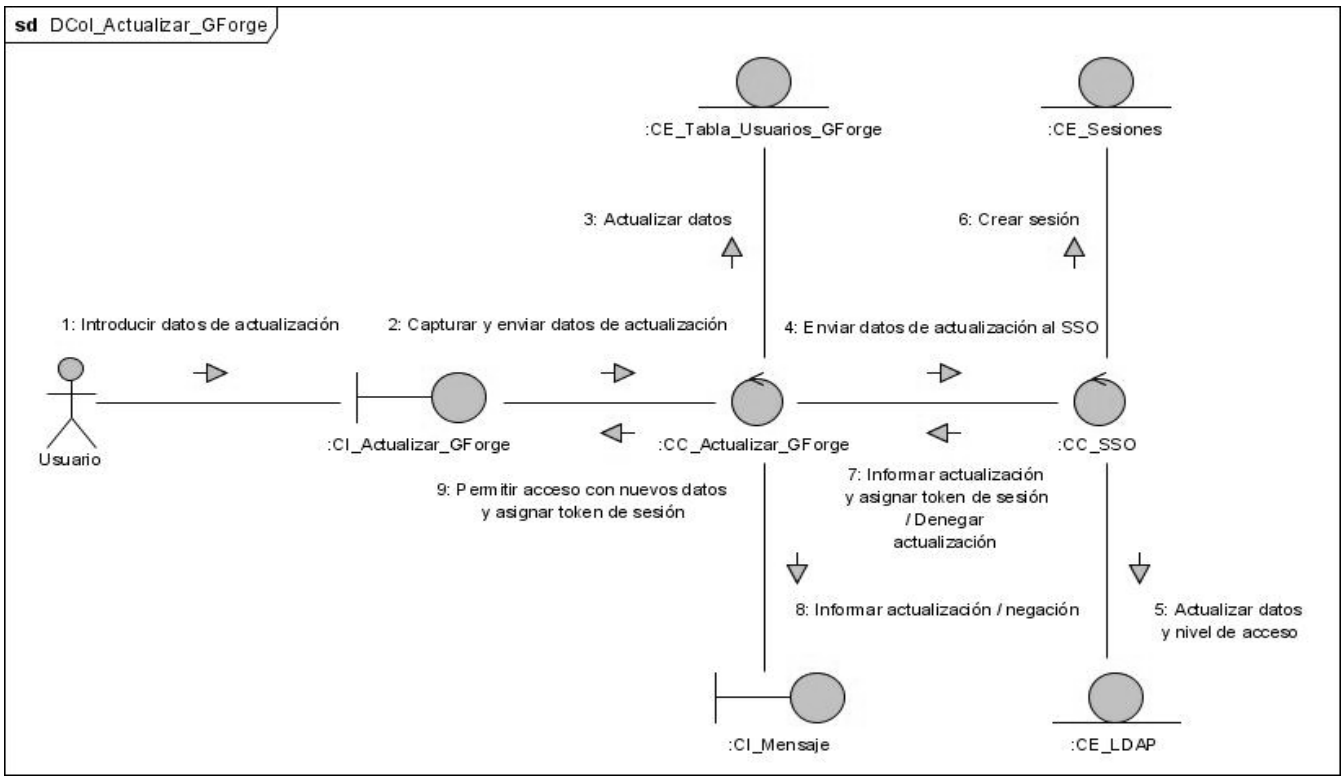

Fig.32: Diagrama de Colaboración. CUS - Actualizar GForge.

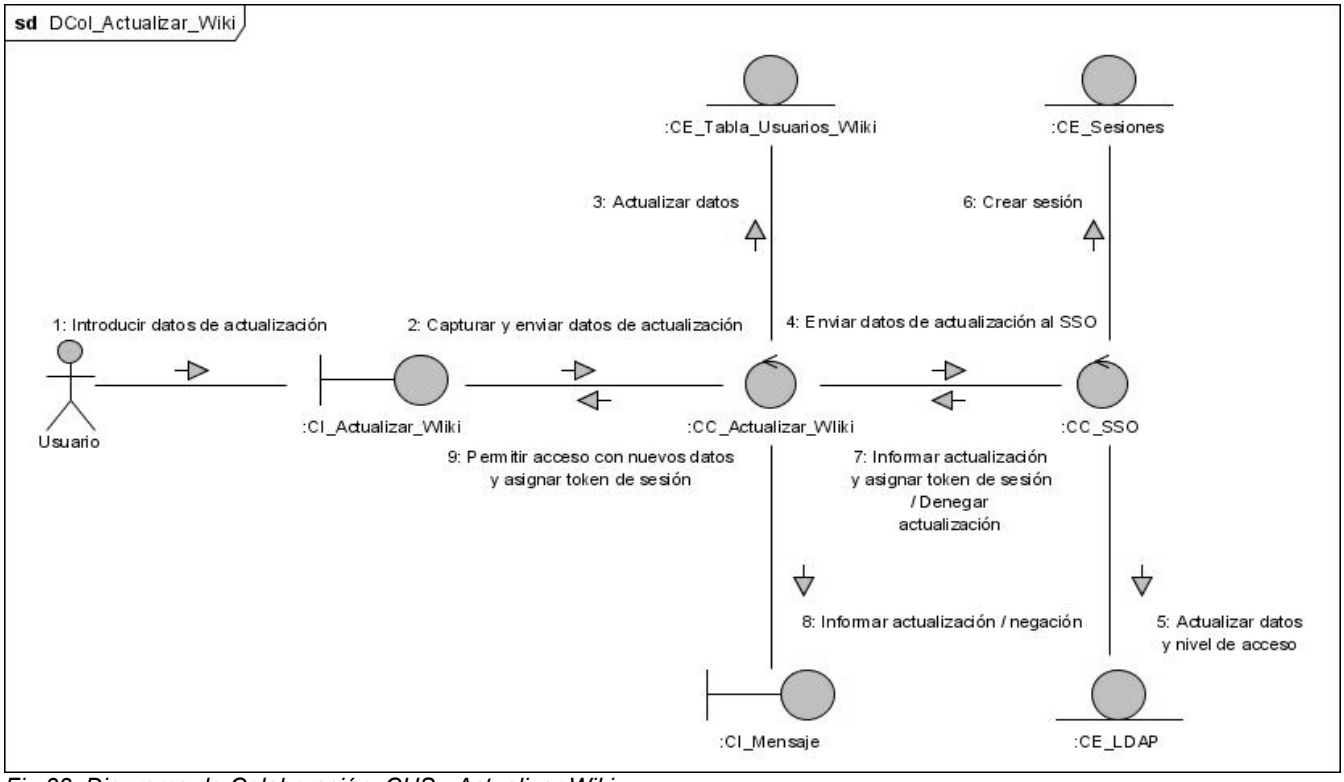

Fig.33: Diagrama de Colaboración. CUS - Actualizar Wiki.

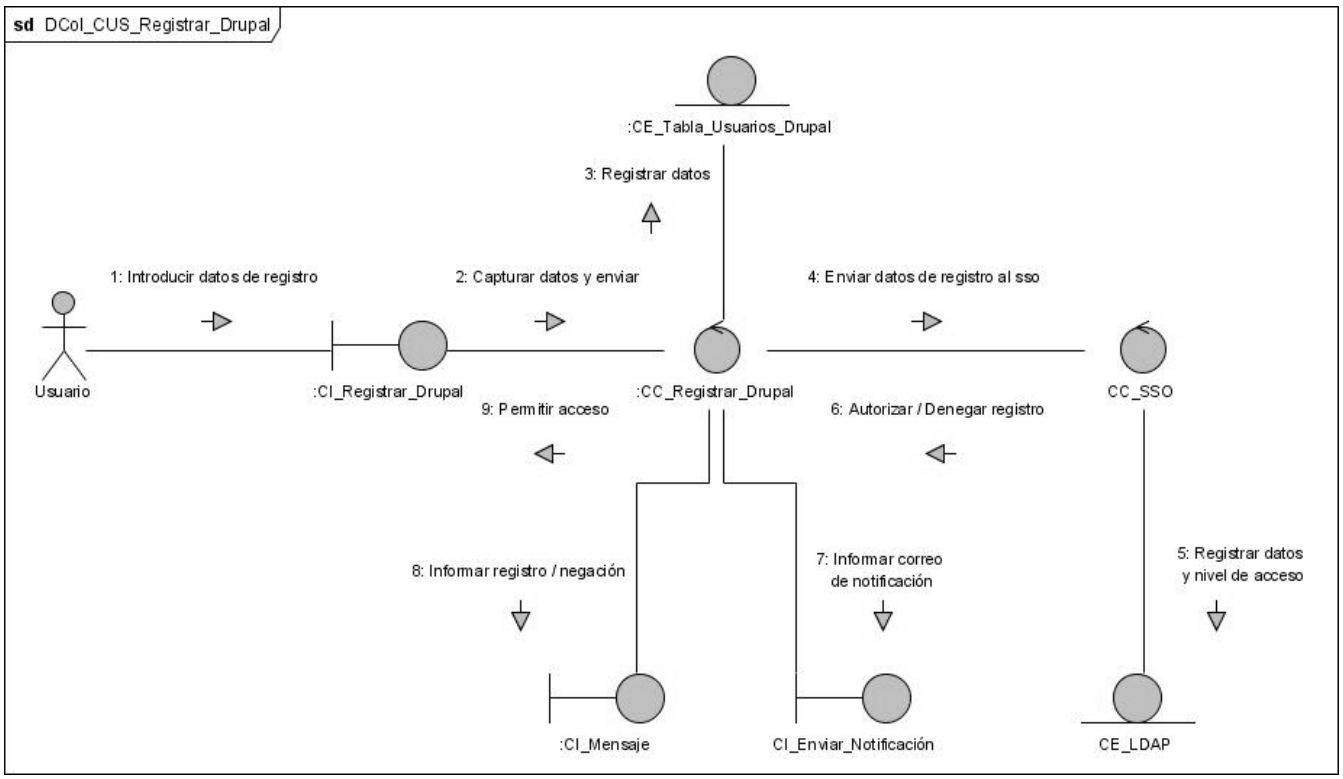

Fig.34: Diagrama de Colaboración. CUS - Registrar Drupal.

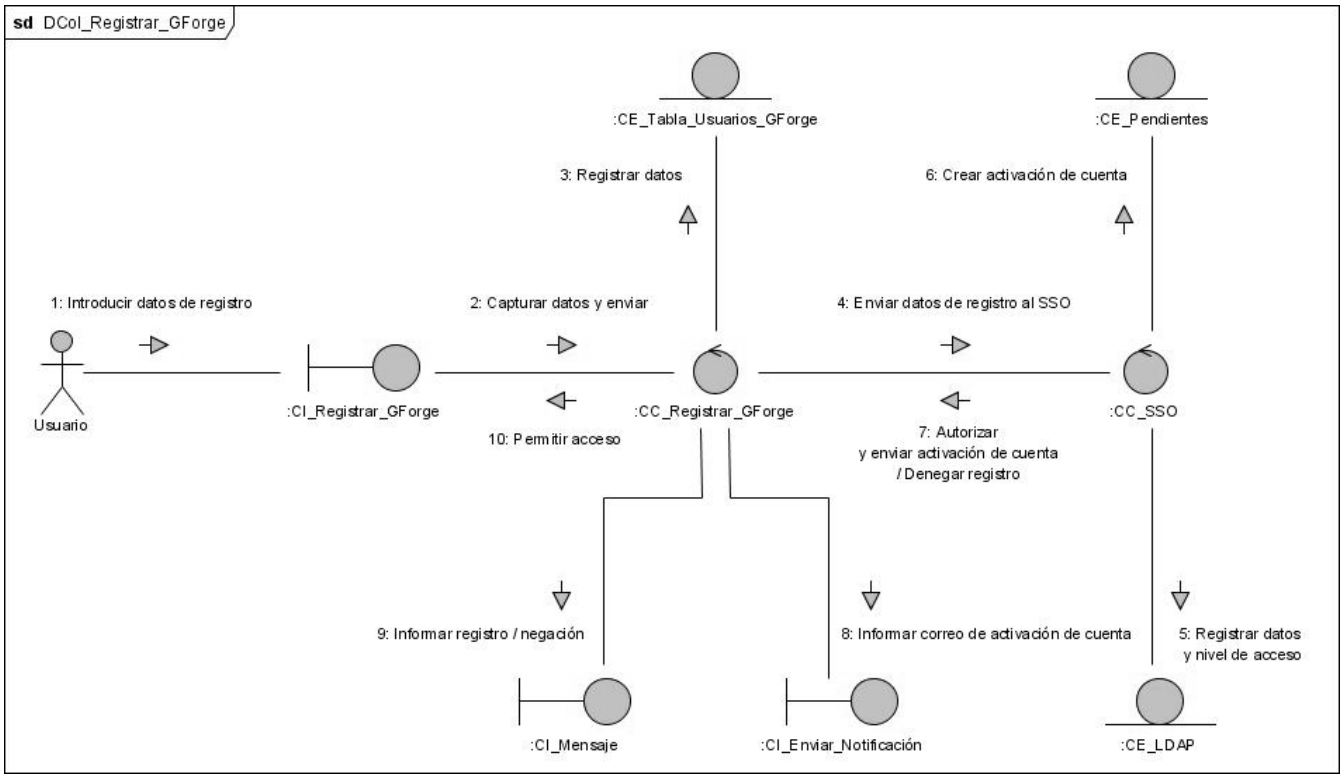

Fig.35: Diagrama de Colaboración. CUS - Registrar GForge.

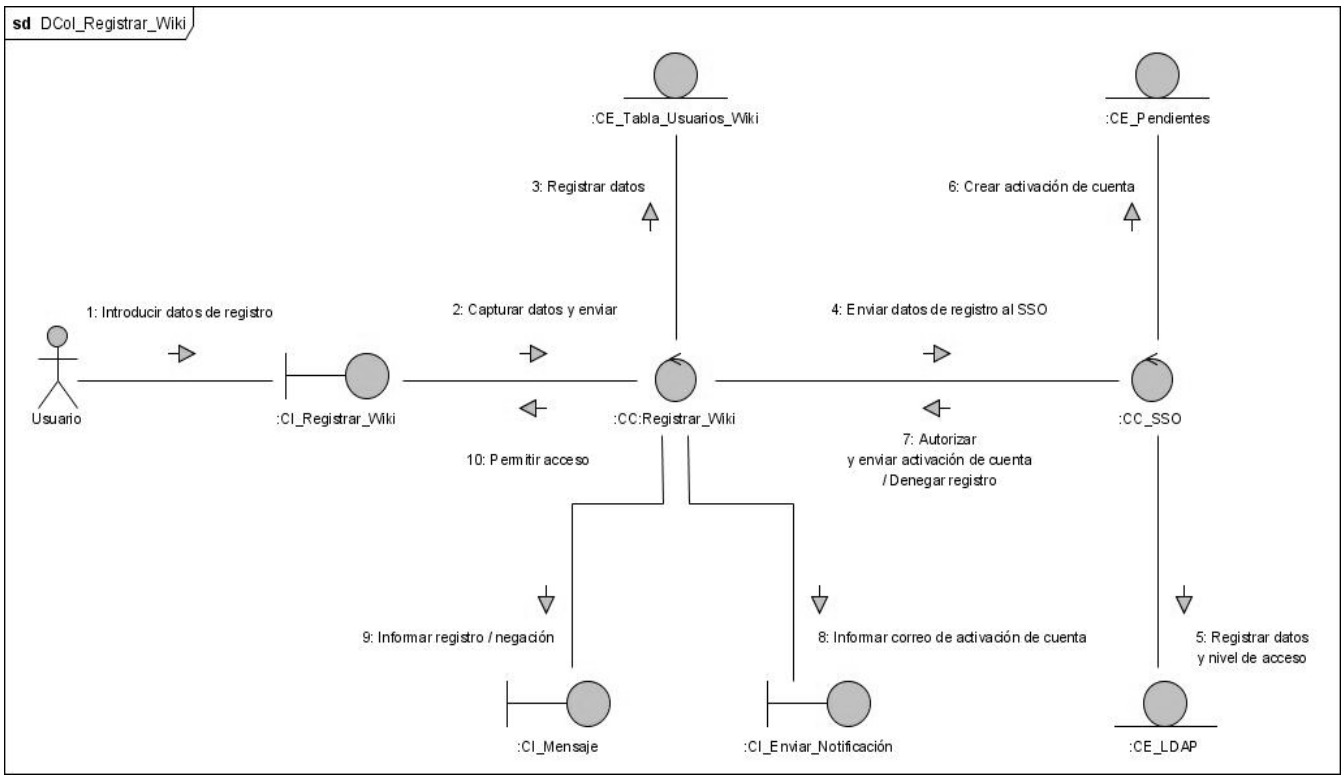

Fig.36: Diagrama de Colaboración. CUS - Registrar Wiki.

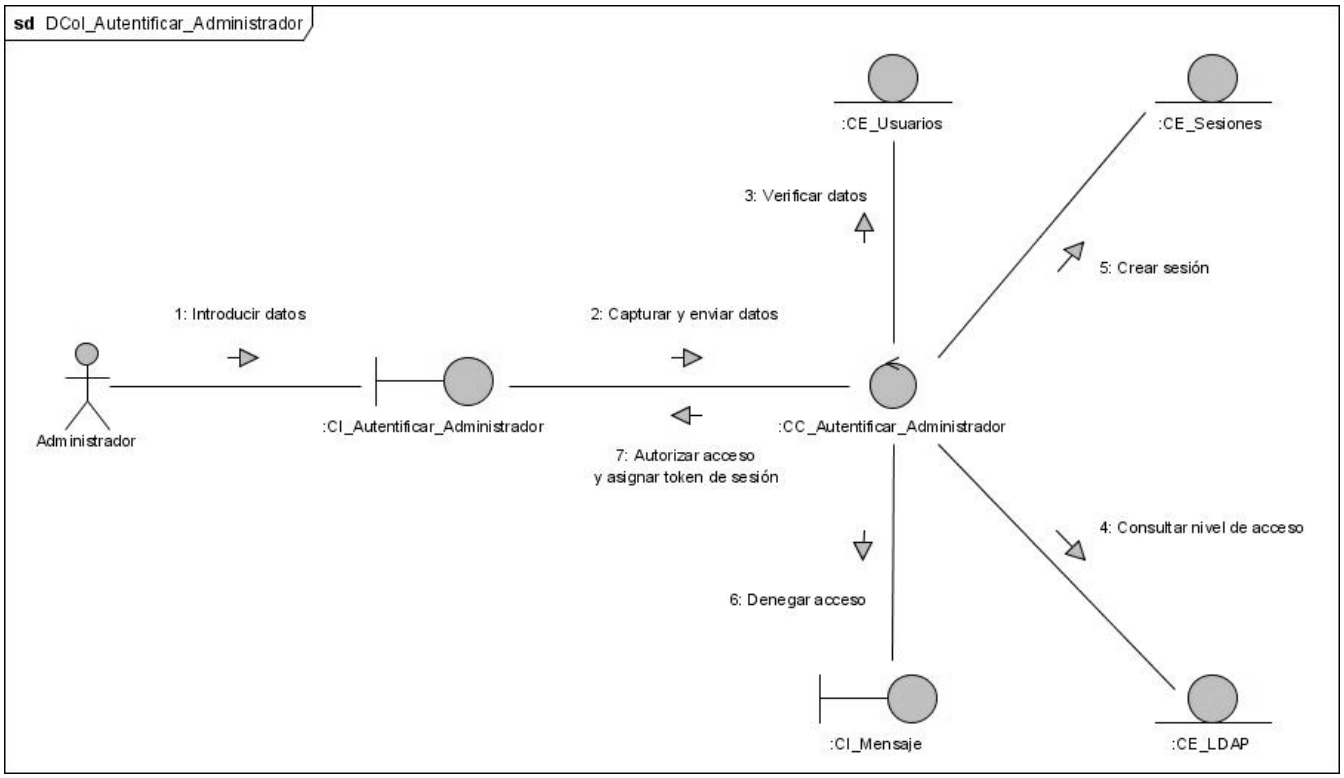

Fig.37: Diagrama de Colaboración. CUS - Autentificar Administrador.

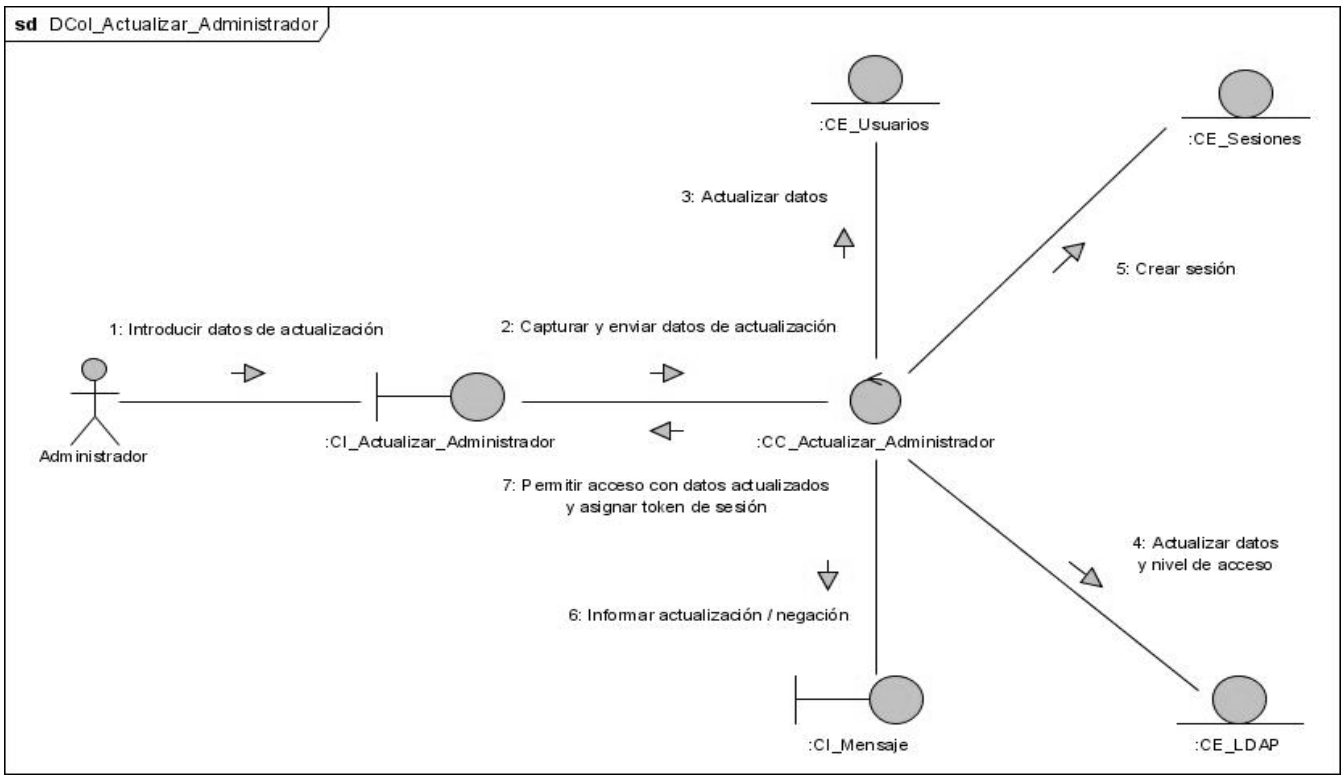

Fig.38: Diagrama de Colaboración. CUS - Actualizar Administrador.

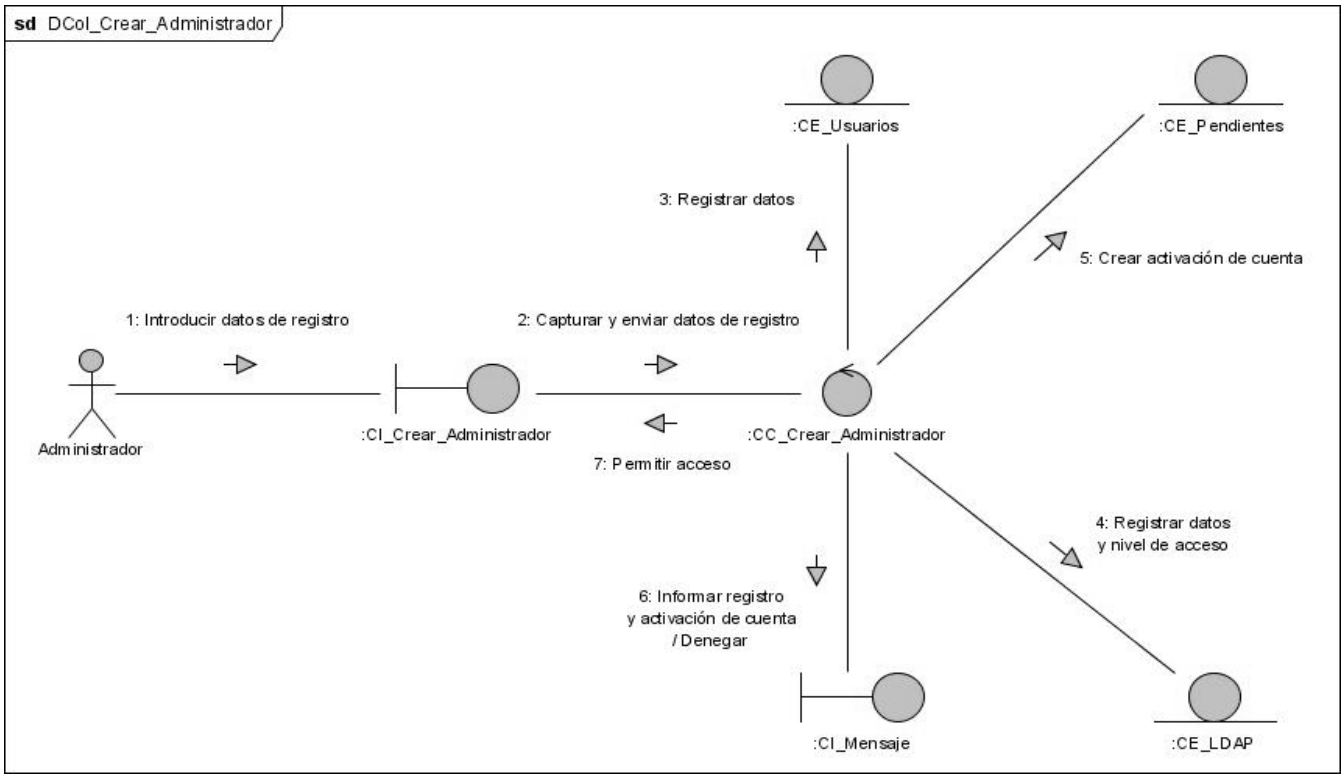

Fig.39: Diagrama de Colaboración. CUS - Crear Administrador.

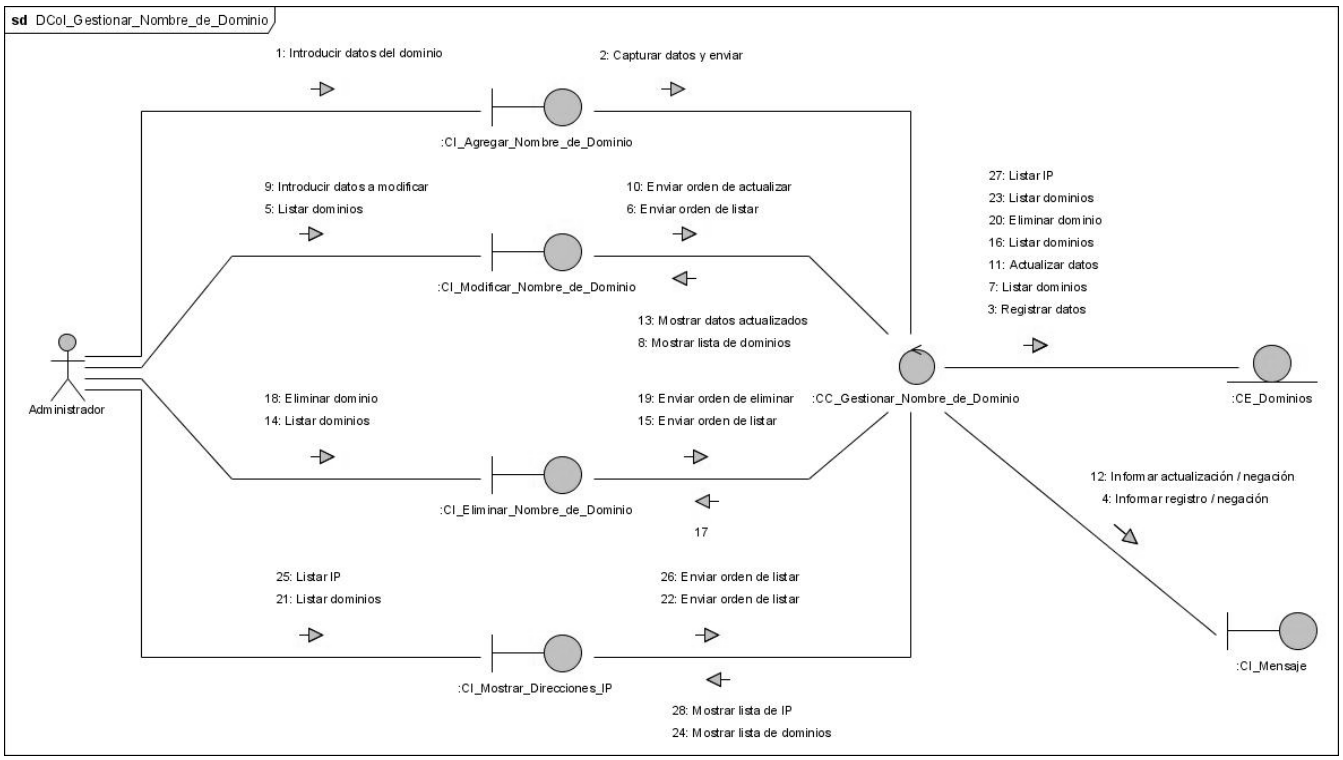

Fig.40: Diagrama de Colaboración. CUS - Gestionar Nombres de Dominio.

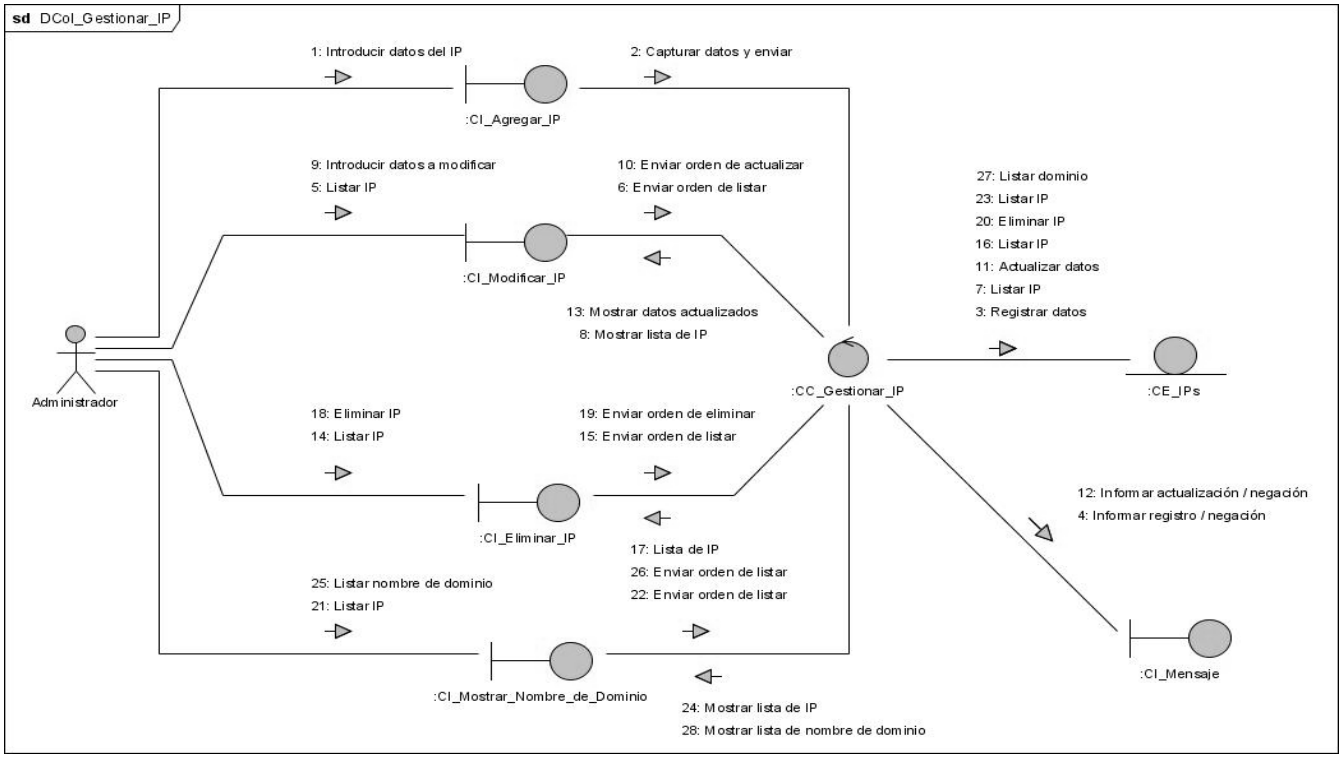

Fig.41: Diagrama de Colaboración. CUS - Gestionar IP.

ANEXO 3: DIAGRAMAS DE CLASES WEB.

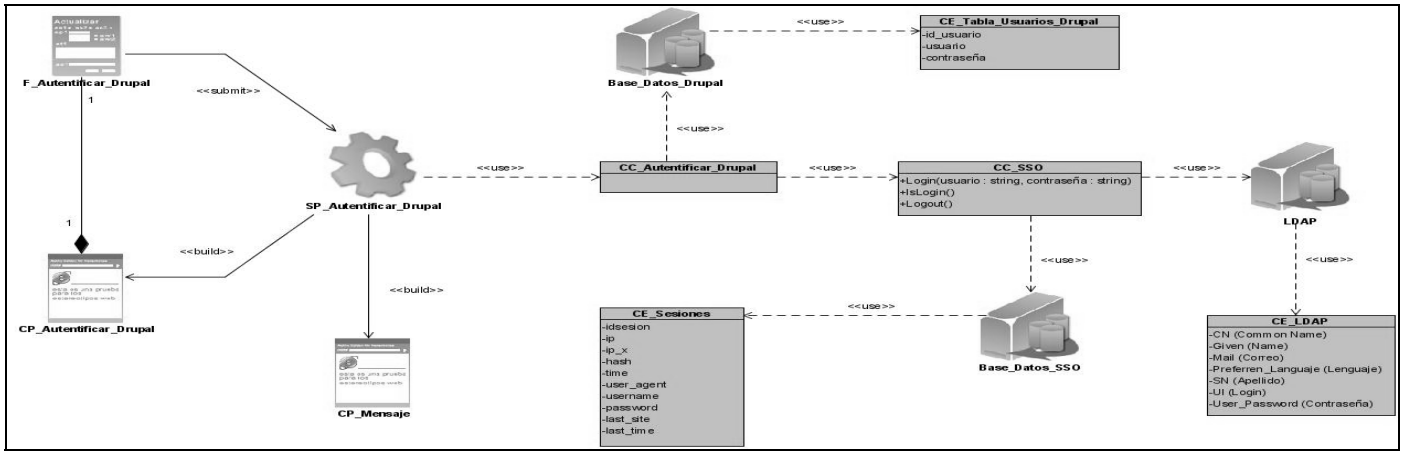

Fig.42: Diagrama de Clases (Web) del Diseño. CUS - Autentificar Drupal.

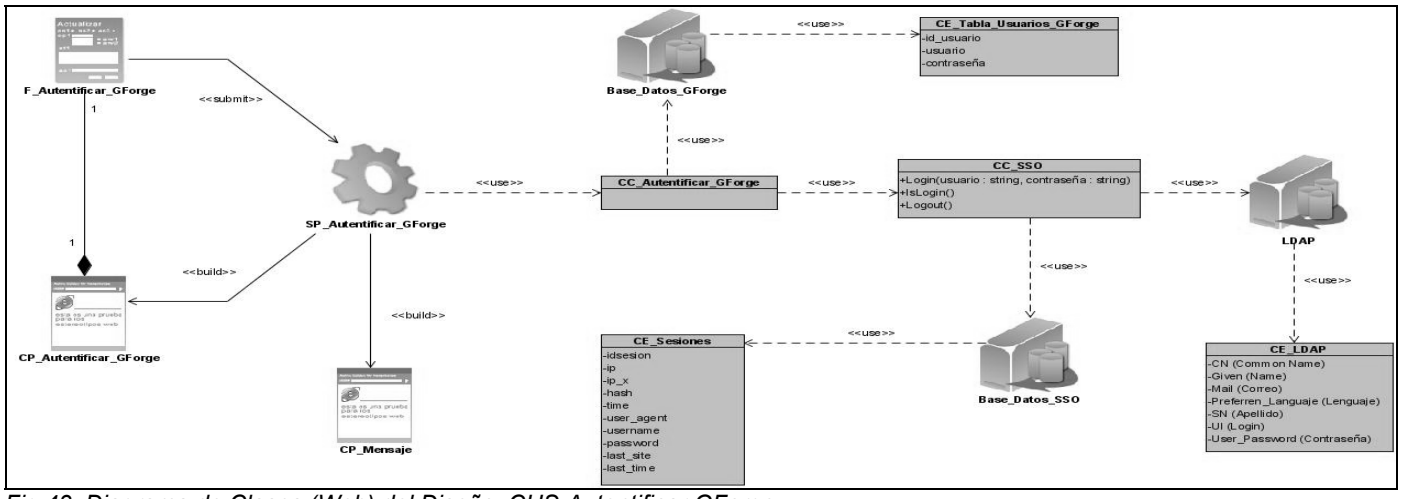

Fig.43: Diagrama de Clases (Web) del Diseño. CUS-Autentificar GForge.

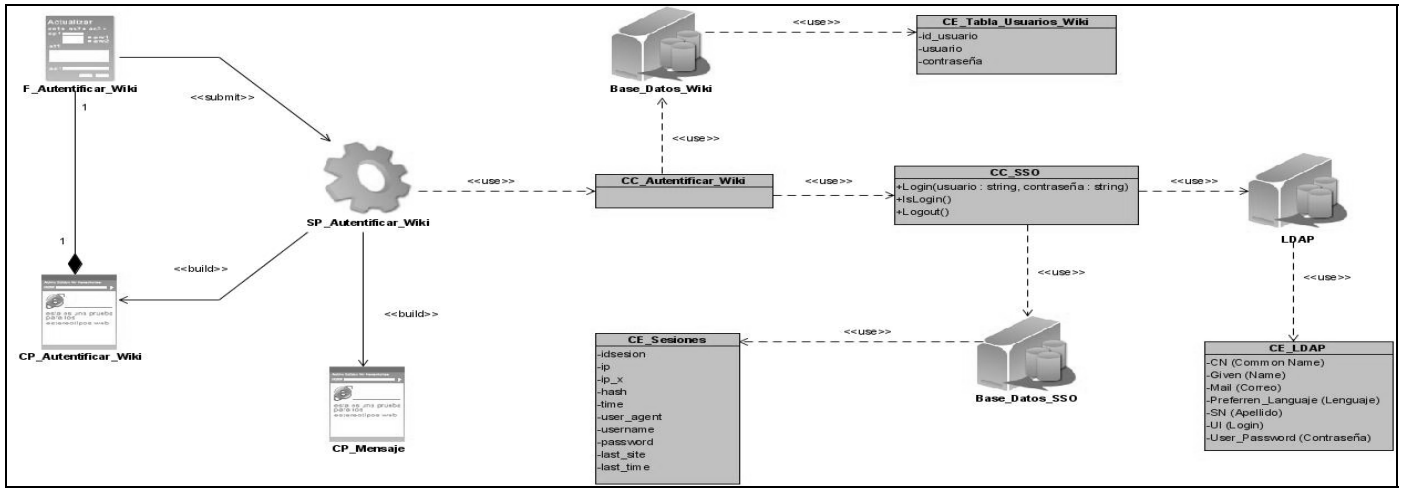

Fig.44: Diagrama de Clases (Web) del Diseño. CUS - Autentificar Wiki.

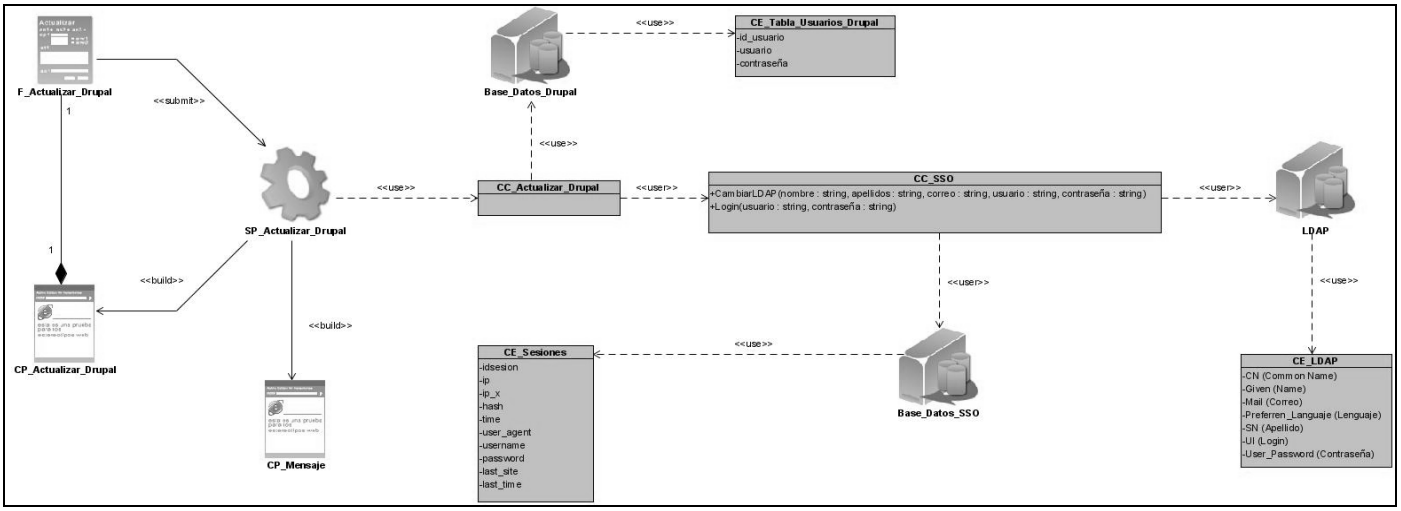

Fig.45: Diagrama de Clases (Web) del Diseño. CUS - Actualizar Drupal.

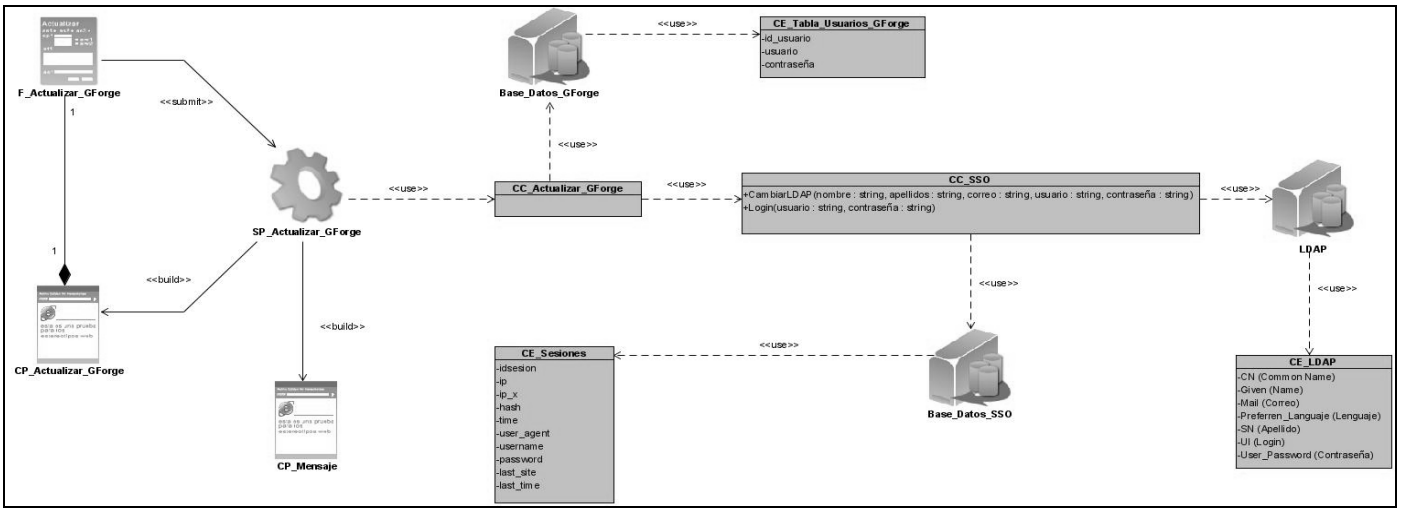

Fig.46: Diagrama de Clases (Web) del Diseño. CUS - Actualizar GForge.

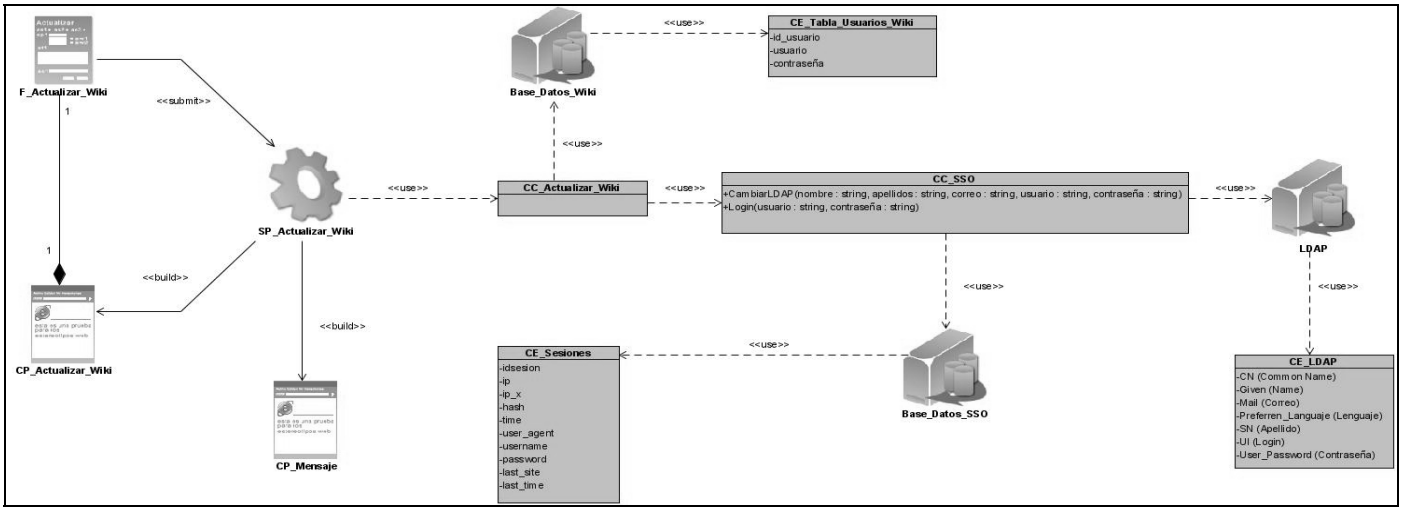

Fig.47: Diagrama de Clases (Web) del Diseño. CUS - Actualizar Wiki.

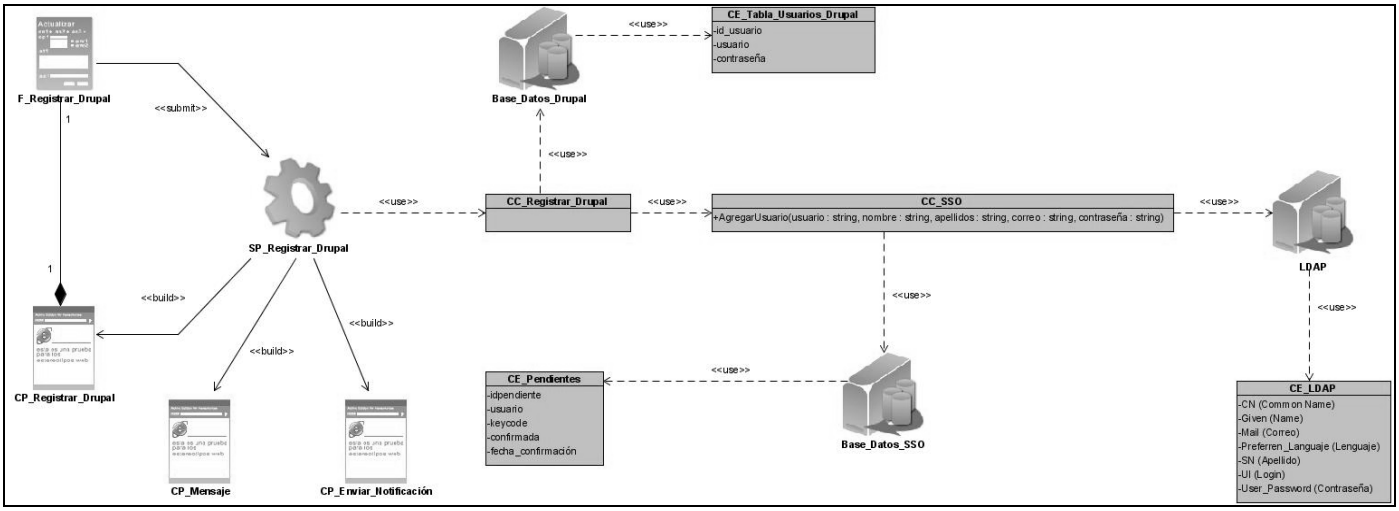

Fig.48: Diagrama de Clases (Web) del Diseño. CUS - Registrar Drupal.

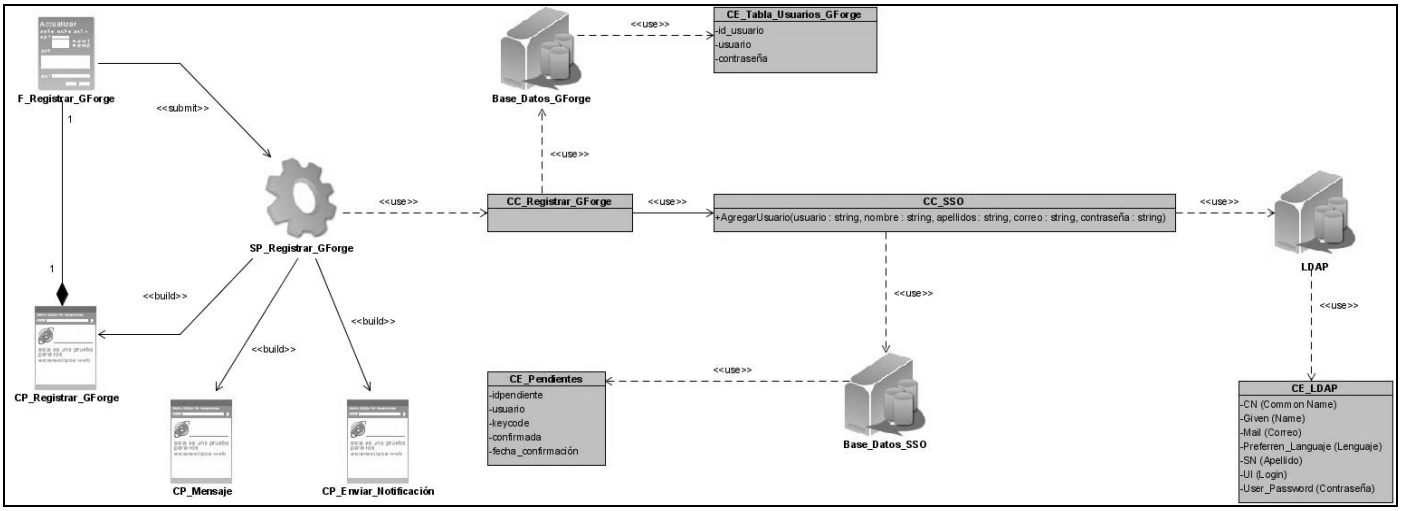

Fig.49: Diagrama de Clases (Web) del Diseño. CUS - Registrar GForge.

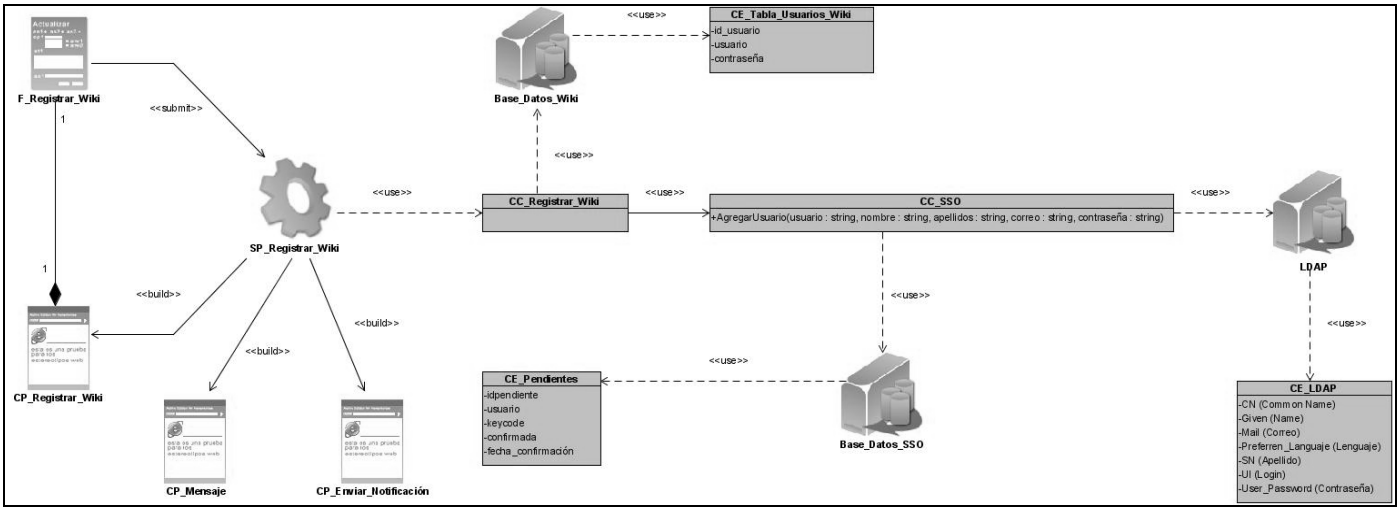

Fig.50: Diagrama de Clases (Web) del Diseño. CUS - Registrar Wiki.

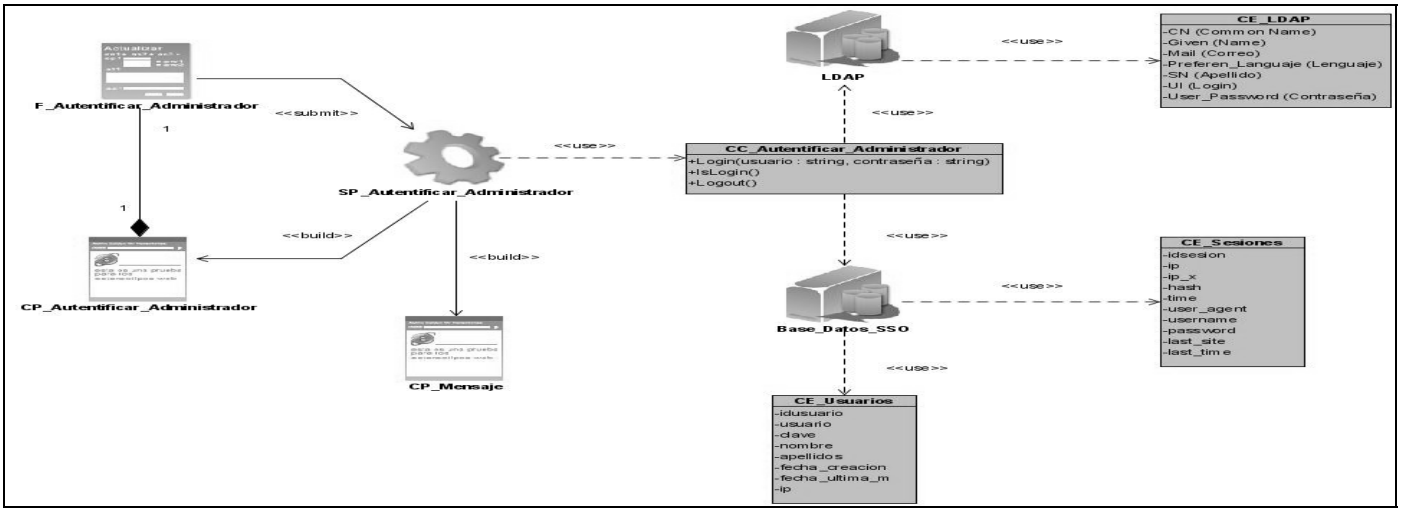

Fig.51: Diagrama de Clases (Web) del Diseño. CUS - Autentificar Administrador.

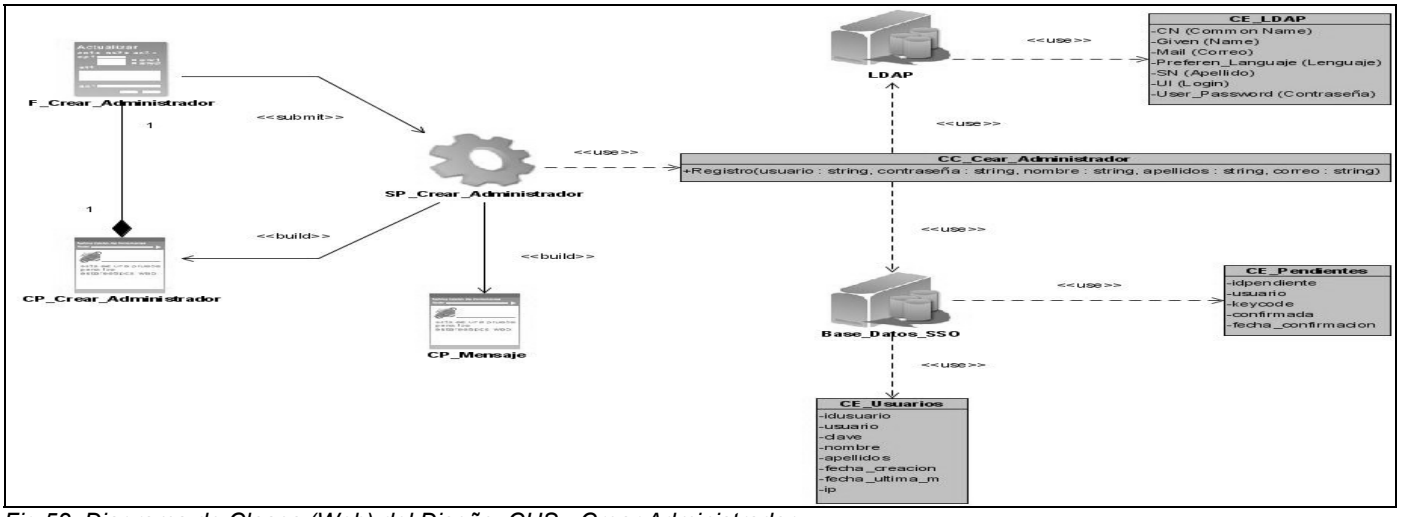

Fig.52: Diagrama de Clases (Web) del Diseño. CUS - Crear Administrador.

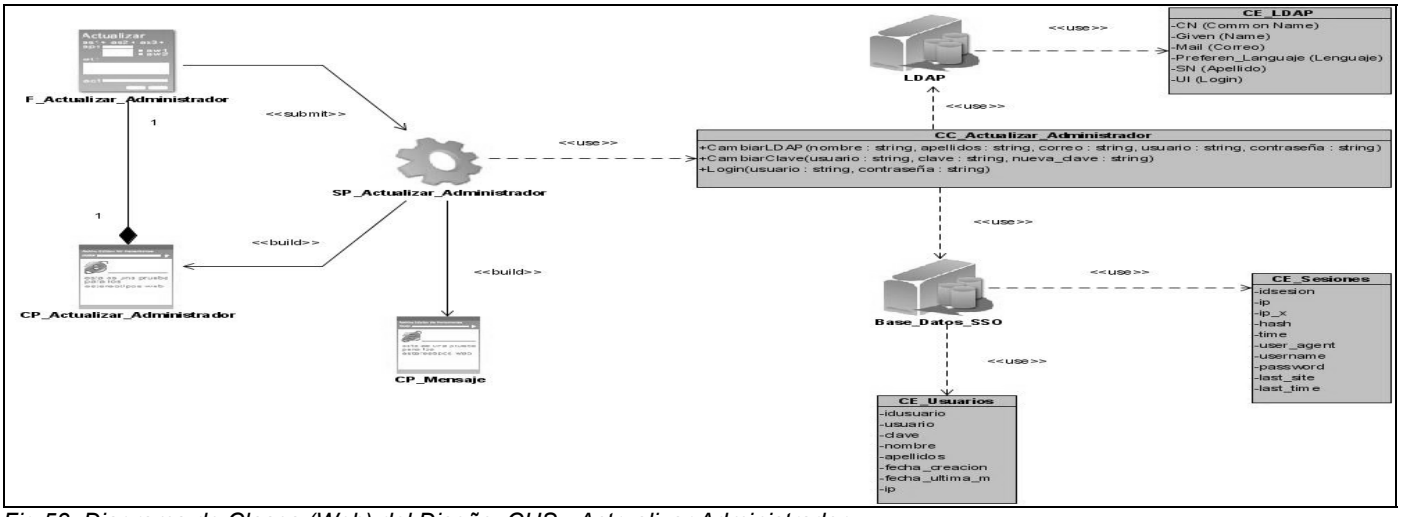

Fig.53: Diagrama de Clases (Web) del Diseño. CUS - Actrualizar Administrador.

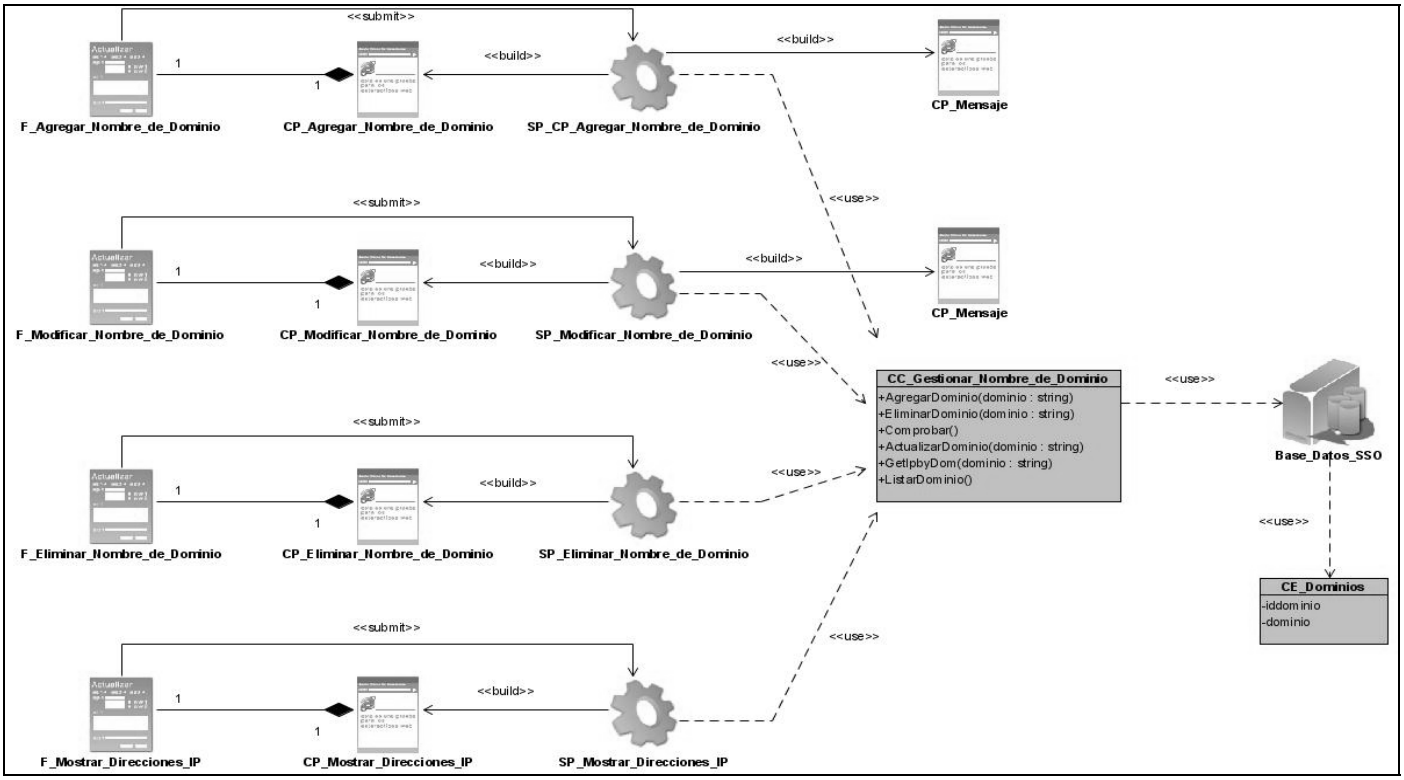

Fig.54: Diagrama de Clases (Web) del Diseño. CUS - Gestionar Nombres de Dominio.

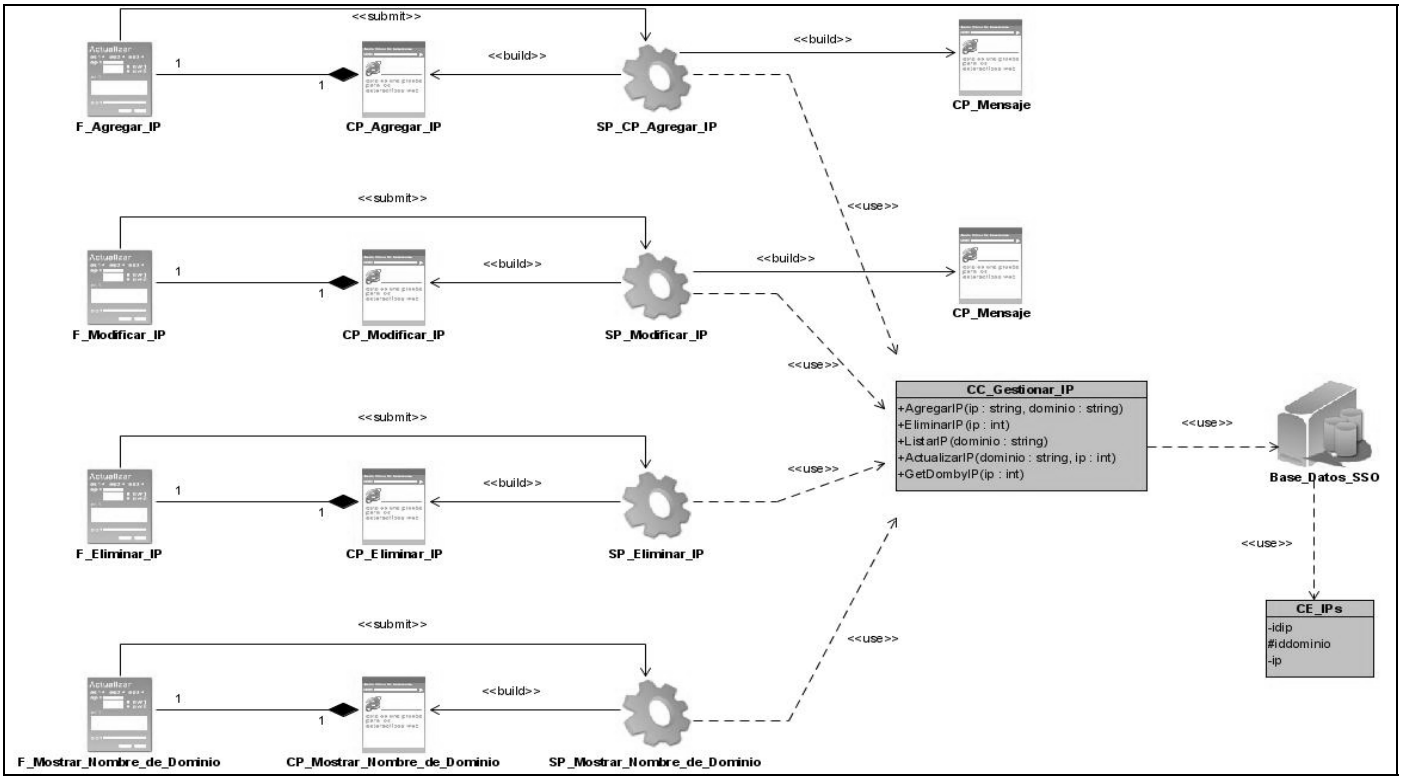

Fig.55: Diagrama de Clases (Web) del Diseño. CUS - Gestionar IP.

ANEXO 4: DIAGRAMA ENTIDAD-RELACIÓN.

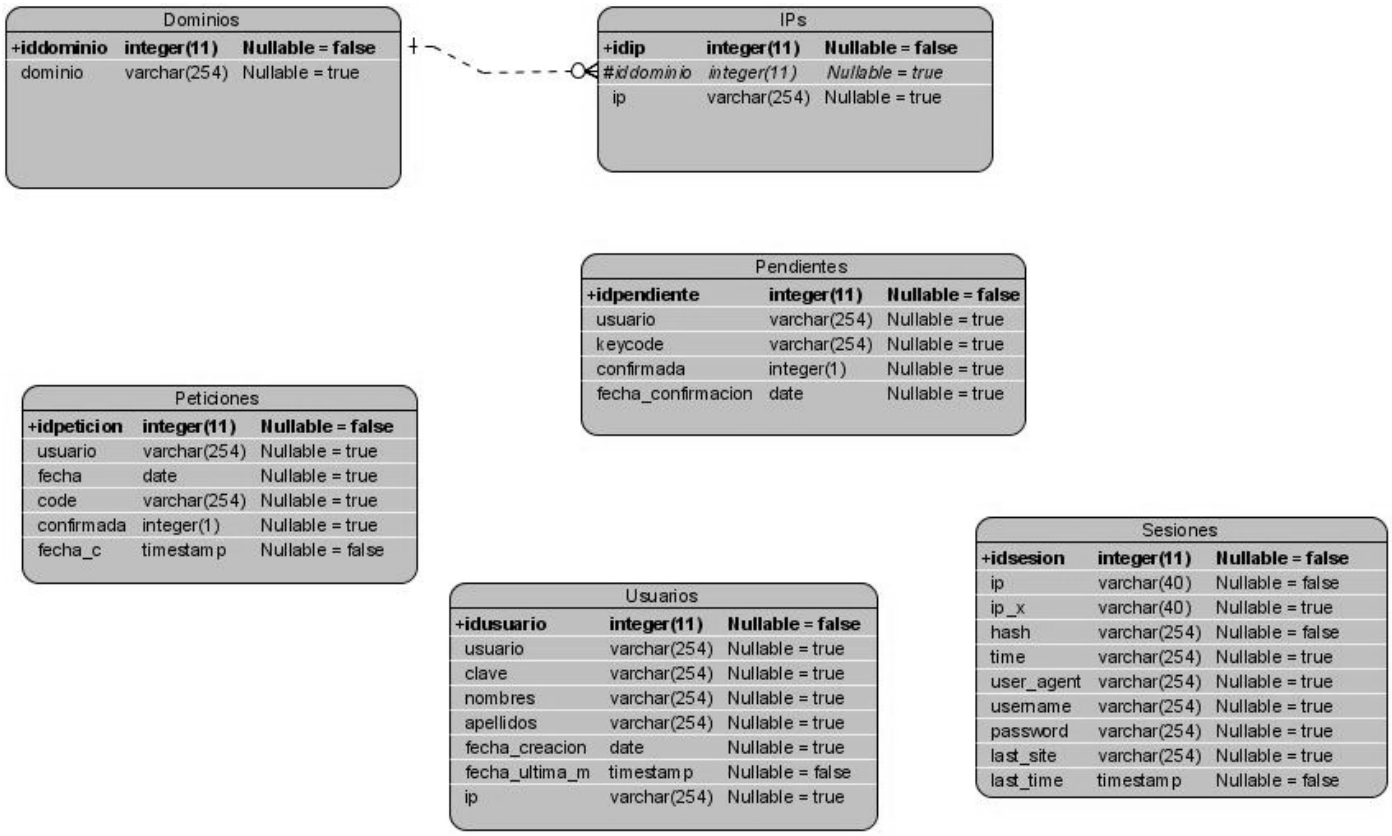

Fig.56: Diagrama Entidad-Relación.