Universidad de las Ciencias Informáticas

“Facultad 2’’

Título: Sistema para la autenticación, autorización y administración de perfiles.

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

Autores: Yenlidie González Batista.
Luis Alberto Martínez Morales.

Tutor: Ing. Danis López Naranjo.
Co-tutor: Ing. Oiner Gómez Baryolo.

Ciudad de La Habana, julio de 2008

“Año 50 de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos al Ministerio de las Fuerzas Armadas Revolucionarias (MINFAR) y a la Universidad de las Ciencias Informáticas (UCI) para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

_____ _____ _____

Firma del Autor Firma del Autor Firma del Autor

_____ _____

Firma del Tutor Firma del Tutor

Agradecimientos

De Yenly

Son muchas las personas a las que les debo estar donde estoy. Primeramente agradecer a mi mamá, a mi padrastro, a mi abuela Aurora y a mi abuela María por quererme e inculcarme todos los valores necesarios para convertirme en una persona de bien. Agradecer además a mi papá, a mis tíos y a toda mi familia por creer en mi y apoyarme.

A Luisi, una de las personas más importantes en mi vida, por quererme, comprenderme y estar a mi lado todos estos años.

A mis amigos y compañeros de aula, a Denyse, a Yunielsy, a mi tutor, a Oiner mi co-tutor, a Verdecia, a Rolando, a Sánchez, en fin a todos los que de una forma u otra me han apoyado y ayudado. .

De Luis

Quiero agradecer a todas la personas que de una forma u otra me ayudaron en la realización de la tesis, a mis compañeros de aula, a los que no estaban en el grupo pero que son grandes amigos, a mi tutor Danis y a mi co-tutor Oiner que se portaron muy bien los dos, a Rolando y a Sánchez que los molesté muchísimas veces y siempre me ayudaron. De forma muy especial a mi novia Yenly y a mi familia que son las cosas más importantes que tengo en la vida, gracias a mi abuelita y abuelito que me criaron como si fueran mi mamá y mi papá y me hicieron lo que soy, para ellos mis más sinceros agradecimientos.

De Yenly

Dedico este trabajo a toda mi familia por apoyarme siempre y por su confianza, a Luisi hacer que mis días sean cada vez mejores y por quererme tanto y a todos mis amigos.

De Luis

Este trabajo se lo dedico a mis familiares, a mis compañeros de aula, a mi novia que tanto me ayudó y que tanta fuerza me dio. A todos los que me dijeron algo para darme ánimos cuando las cosas se ponían difíciles y a todos los que me ayudaron cuando los molesté para lograr terminar el trabajo.

"El mundo camina hacia la era electrónica... Todo indica que esta ciencia se constituirá en algo así como una medida del desarrollo; quien la domine será un país de vanguardia. Vamos a colocar nuestros esfuerzos en este sentido con audacia revolucionaria"

"Ernesto Che Guevara"

RESUMEN

La seguridad es un factor fundamental en cualquier institución y los datos son una parte fundamental en las mismas, mucho más si se trata de las instituciones del ministerio de las Fuerzas Armadas Revolucionarias (FAR) ya que en ellas se maneja mucha información lo mismo táctica que logística y es de vital importancia asegurarla de forma tal que no hayan violaciones de la seguridad. Es por esto que se requiere una buena administración y control de la información y medios materiales, así como una adecuada seguridad acorde a los intereses de nuestro país.

En la actualidad todas las aplicaciones que se encuentran en centros de las FAR implementan su propia seguridad lo cual resulta trabajoso y trae como consecuencia que los usuarios tengan que memorizar muchas contraseñas y nombres de usuarios para las distintas aplicaciones a las que necesiten conectarse. Además no se realiza una administración centralizada de perfiles de usuario y es muy difícil controlar que no ocurran violaciones de seguridad en los sistemas y en caso de que ocurran poder detectarlas.

En la Unidad de Compatibilización, Integración y Desarrollo de Productos para la Defensa (UCID) se están desarrollando una serie de aplicaciones y muchas de estas se encuentran en proceso de terminación. Entre estas se encuentra la aplicación de seguridad, el objetivo de la misma es lograr una seguridad centralizada de todas las demás aplicaciones.

El año pasado se hizo una aplicación similar pero hubo una serie de aspectos que no se tuvieron en cuenta a la hora de concebir el sistema, es por esto que surge la necesidad de realizar nuevamente el análisis y diseño del mismo con el objetivo de mejorar lo que se hizo anteriormente.

Tabla de Contenidos

INTRODUCCIÓN

A medida que van pasando los años las tecnologías de la investigación y las comunicaciones se van desarrollando y por ende se van perfeccionando. Nuestro país aunque se encuentra fuertemente bloqueado por el imperialismo hace todo lo posible para estar a la altura de los países desarrollados en materia de las Tecnologías de la Informática y las Comunicaciones (TIC), llevando a cabo la informatización de todos los sectores del país. El ministerio de las Fuerzas Armadas Revolucionarias (FAR), quien es el encargado de la defensa de nuestra patria también desea sumarse a este proceso por todas las ventajas que trae consigo el mismo.

En las FAR por la cantidad de información que se maneja, lo mismo táctica que logística es de vital importancia una buena administración y control de la información y medios materiales, así como una adecuada seguridad acorde a los intereses de nuestro país. Es por esto que las aplicaciones que se realizan en centros de las FAR requieren de una seguridad estricta y bien concebida que permita controlar y monitorear la información para evitar que las aplicaciones sean atacadas y en caso de que esto ocurra que el ataque no sea fructífero y atrapar al atacante en el menor tiempo posible. Es por este motivo que los procesos de autenticación, autorización y administración de perfiles de usuarios en las aplicaciones son muy importantes y constituyen una parte fundamental de las mismas.

Actualmente la seguridad en las aplicaciones que se encuentran en las instituciones de las FAR resulta trabajosa ya que cada sistema la controla de forma diferente, o sea, cada uno implementa su propia seguridad. Esto trae como consecuencia que los usuarios tienen que memorizar muchas contraseñas y nombres de usuarios para las distintas aplicaciones a las que necesiten conectarse. Además no se realiza una administración centralizada de perfiles de usuario y es muy difícil controlar que no ocurran violaciones de seguridad en los sistemas y si ocurren poder detectarlas.

Es necesario que los sistemas conciban una política de seguridad adecuada para logar un buen desempeño de sus objetivos. Cuando se logre una arquitectura centralizada de la seguridad de las aplicaciones se facilitará mucho el control de las mismas además de una disminución en el tiempo de desarrollo de las aplicaciones debido a que no tendrán que implementar su seguridad pues se hará de forma centralizada.

UCID (Unidad de Compatibilización, Integración y Desarrollo de Productos para la Defensa) es un centro donde trabajan en conjunto las Fuerzas Armadas Revolucionarias y la Universidad de las Ciencias Informáticas, actualmente en este centro se están desarrollando una serie de aplicaciones y muchas de estas se encuentran en proceso de terminación. Entre estas se encuentra la aplicación de seguridad, el objetivo de la misma es lograr una seguridad centralizada de todas las demás aplicaciones. El pasado año se logró realizar un sistema con las características del sistema de seguridad que se desea hacer hoy, la aplicación funcionó y cumplió su objetivo que era logar una administración centralizada de la seguridad de los sistemas y es por esta seguridad que se rigen las aplicaciones en estos momentos.

A pesar de cumplir sus objetivos se detectaron una serie de aspectos que no se tuvieron en cuenta a la hora de concebir el sistema de seguridad existente, además las interfaces de usuarios diseñadas presentaron problemas a la hora de realizar las actualizaciones deseadas por el usuario y no eran las más apropiadas ya que no mostraban un entorno amigable para el usuario a la hora de realizar todas las acciones deseadas en el sistema.

El análisis y diseño se realizó nuevamente con el objetivo de mejorar lo que se hizo anteriormente.

El trabajo esta elaborado con tecnología más reciente que el anterior lo que permite dar una solución óptima a este problema de la seguridad centralizada.

Luego de realizar un profundo análisis sobre la importancia que tiene este proyecto por sus características y por la necesidad de tener una administración lo más fiable posible nos hemos planteado el siguiente **problema a resolver:** ¿Cómo lograr una administración centralizada de perfiles de usuarios, autenticación y autorización en un entorno de varias aplicaciones que sea fácil de usar, de controlar y más seguro?

Para ello tendremos como **Objeto de estudio:** La administración de la seguridad en sistemas web.

Nuestro **campo de acción** es: El proceso de control de autenticación, autorización y la administración de perfiles de usuarios en los sistemas de las FAR, ya que nuestra investigación se centra más específicamente en este tema.

Para resolver el problema planteado nos hemos propuesto como **objetivo general:** Analizar y diseñar un sistema independiente de las aplicaciones que garantice la autenticación, autorización y administración de perfiles de usuarios.

Para cumplir nuestro objetivo general nos hemos trazado los siguientes **objetivos específicos:**

- Realizar el diseño teórico de la investigación.
- Realizar la modelación del negocio.
- Identificar los requerimientos funcionales y no funcionales del sistema.
- Modelar los casos de usos del sistema.
- Realizar el análisis y diseño de una administración centralizada de la seguridad.

Hipótesis: Si se realiza una administración centralizada de la autenticación, autorización y perfiles de usuarios entonces se lograría un sistema más fácil de usar, de controlar y más seguro.

Con este trabajo de diploma se pretende realizar el análisis y diseño de un sistema que garantice la seguridad de varias aplicaciones de forma centralizada. El mismo quedará estructurado en tres capítulos, donde el **Capítulo 1** abarca todo lo referente a la **Fundamentación teórica**, en el mismo se analizó todo lo referente a la seguridad en aplicaciones web, así como el proceso de autentificación, autorización y administración de perfiles. Además se realiza un estudio sobre las tecnologías, lenguajes y herramientas utilizadas, teniendo en cuenta las ventajas que cada una de ellas nos brinda. En el **Capítulo 2** se realizó todo lo referente a los flujos de trabajo **Negocio y Requerimientos**, se hizo un estudio sobre el funcionamiento del sistema para un mayor entendimiento a la hora de realizar la modelación del negocio, además de sacar los requerimientos del sistema. También se realizaron las descripciones textuales de los casos de uso del sistema y el diseño de los prototipos de interfaz de usuario. Por último el **Capítulo 3** recoge los flujos de trabajo de **Análisis y Diseño,** en él se encuentran todos los artefactos referentes a estos flujos como son los diagramas de clases del análisis, los diagramas de interacción y los diagramas de clases del diseño.

CAPITULO1: FUNDAMENTACIÓN TEÓRICA

1.1. Introducción.

Este capítulo brinda información sobre la seguridad en aplicaciones Web y sobre las tendencias y tecnologías actuales que se tuvieron en cuenta a la hora de diseñar el sistema. Se dará una breve descripción de los lenguajes de programación y los gestores de bases de datos seleccionados de acuerdo a las necesidades del sistema que se desea realizar, exponiendo sus ventajas. Además se expone un ejemplo de solución que propone una seguridad centralizada de la seguridad.

1.2. Seguridad en aplicaciones Web

Se puede definir seguridad informática como un conjunto de métodos y herramientas destinados a proteger los bienes informáticos de cualquier institución.

La seguridad de las aplicaciones web siempre ha estado en entredicho. El problema de la seguridad de las aplicaciones web es una consecuencia de como se escribe el software, de que técnicas son utilizadas. Por ejemplo, la complejidad de la plataforma es un factor importante; pero aún más es saber cuando utilizar una herramienta en vez de la otra. En pocas palabras no es solamente un problema de código, firewalls o de si usemos o no un software para revisar el código de manera automática sino de cómo se programa. [1]

Para lograr que un sistema sea fuerte desde el punto de vista de su seguridad se debe garantizar tener un buen balance entre estos aspectos:

- Confidencialidad: La información o los activos informáticos son accedidos solo por las personas autorizadas.
- Integridad: Los activos o la información solo pueden ser modificados por las personas autorizadas y de la forma autorizada.
- Disponibilidad: Los activos informáticos son accedidos por las personas autorizadas en el momento requerido. [2]

A la hora de desarrollar una aplicación, generalmente nos centramos más en la funcionalidad que en la seguridad. Lo que trae como consecuencia que los atacantes se aprovechen de esto y atenten contra cualquiera de estos tres aspectos.

En la seguridad de aplicaciones juegan un papel fundamental los procesos de autenticación y autorización, ya que permiten un mejor control en el acceso a la información.

Autenticación: Proceso utilizado en los mecanismos de control de acceso con el objetivo de verificar la identidad de un usuario, dispositivo o sistema mediante la comprobación de credenciales de acceso.

Autorización: Es el proceso por el que permite o deniega el acceso de un usuario a un recurso.

1.2.1. Tipos de ataques.

Un ataque no es más que la realización de una amenaza. Diremos que se entiende por amenaza una condición del entorno del sistema de información (persona, máquina, suceso o idea) que, dada una oportunidad, podría dar lugar a que ocurra una violación de la seguridad (confidencialidad, integridad, disponibilidad).

Las cuatro categorías generales de amenazas o ataques son las siguientes:

- **Intercepción:** Una entidad no autorizada consigue acceso a un recurso. Este es un ataque contra la confidencialidad. La entidad no autorizada podría ser una persona, un programa o un ordenador. Ejemplos de este ataque son pinchar una línea para tomar los datos que circulen por la red y la copia ilícita de ficheros o programas (intercepción de datos), o bien la lectura de las cabeceras de paquetes para desvelar la identidad de uno o más de los usuarios implicados en la comunicación observada ilegalmente (intercepción de identidad). [3]
- **Modificación:** Una entidad no autorizada no sólo consigue acceder a un recurso, sino que es capaz de manipularlo. Este es un ataque contra la integridad. Ejemplos de este ataque son el cambio de valores en un archivo de datos, alterar un programa para que funcione de forma diferente y modificar el contenido de mensajes que están siendo transferidos por la red.[3]

- **Interrupción:** Un recurso del sistema es destruido o se vuelve no disponible. Este es un ataque contra la disponibilidad. Ejemplos de este ataque son la destrucción de un elemento hardware como un disco duro, cortar una línea de comunicación o deshabilitar el sistema de gestión de ficheros.[3]
- **Fabricación:** Una entidad no autorizada inserta objetos falsificados en el sistema. Este es un ataque contra la autenticidad. Ejemplos de este ataque son la inserción de mensajes ilegítimos en una red o añadir registros a un archivo. Estos ataques se pueden asimismo clasificar de forma útil en términos de ataques pasivos y ataques activos.[3]

Ataques pasivos: En los ataques pasivos el atacante no altera la comunicación, sino que únicamente la escucha o monitoriza, para obtener información que está siendo transmitida.

Sus objetivos son la intercepción de datos y el análisis de tráfico, una técnica más sutil para obtener información de la comunicación, que puede consistir en: Obtención del origen y destinatario de la comunicación, leyendo las cabeceras de los paquetes monitorizados. Control del volumen de tráfico intercambiado entre las entidades monitorizadas, obteniendo así información acerca de actividad o inactividad inusuales. Control de las horas habituales de intercambio de datos entre las entidades de la comunicación, para extraer información acerca de los períodos de actividad. [3]

Los ataques pasivos son muy difíciles de detectar, ya que no provocan ninguna alteración de los datos. Sin embargo, es posible evitar su éxito mediante el cifrado de la información y otros mecanismos.

Ataques activos: Estos ataques implican algún tipo de modificación del flujo de datos transmitido o la creación de un falso flujo de datos, pudiendo subdividirse en cuatro categorías:

- **Suplantación de identidad:** El intruso se hace pasar por una entidad diferente. Normalmente incluye alguna de las otras formas de ataque activo. Por ejemplo, secuencias de autenticación pueden ser capturadas y repetidas, permitiendo a una entidad no autorizada acceder a una serie de recursos privilegiados suplantando a la entidad que posee esos privilegios, como al robar la contraseña de acceso a una cuenta.[3]
- **Reactuación:** Uno o varios mensajes legítimos son capturados y repetidos para producir un efecto no deseado, como por ejemplo ingresar dinero repetidas veces en una cuenta dada.[3]

- **Modificación de mensajes:** Una porción del mensaje legítimo es alterada, o los mensajes son retardados o reordenados, para producir un efecto no autorizado. Por ejemplo, el mensaje "Ingresa un millón de pesos en la cuenta A" podría ser modificado para decir "Ingresa un millón de pesos en la cuenta B".[3]
- **Degradación fraudulenta del servicio:** Impide o inhibe el uso normal o la gestión de recursos informáticos y de comunicaciones. Por ejemplo, el intruso podría suprimir todos los mensajes dirigidos a una determinada entidad o se podría interrumpir el servicio de una red inundándola con mensajes espurios. Entre estos ataques se encuentran los de denegación de servicio, que consisten en paralizar temporalmente el servicio de un servidor de correo, Web, FTP, etc.[3]

1.2.2. Control de acceso. Las contraseñas.

Un concepto técnico o lógico de acceso es la interacción entre un sujeto y un objeto que resulta en un flujo de información de uno al otro. El sujeto es la entidad que recibe o modifica la información o los datos contenidos en los objetos; puede ser un usuario, programa, proceso, entre otros.

El control de acceso vigila por autenticar la identidad de los usuarios o grupos de usuarios y autorizar el acceso a recursos. Los controles de accesos son necesarios para proteger la confidencialidad, integridad y disponibilidad de los objetos y la información que contienen implícitamente, ya que restringe a los usuarios el paso sólo a la información que necesitan para su trabajo o la que por jerarquía tienen derecho a obtener. [10]

Las contraseñas

Una contraseña o clave es una forma de autenticación que utiliza información que solamente el individuo conoce, o sea, secreta, para controlar el acceso hacia algún recurso protegido. Está compuesta por un código alfanumérico y en ocasiones solamente numérico (PIN). [10]

Las contraseñas crean una seguridad contra los usuarios no autorizados mientras que el sistema de seguridad sólo verifica y confirma si la contraseña es válida y permite el acceso si lo es, pero no

identifica si el usuario que está en posesión de dicha contraseña está autorizado a utilizarla. Es por esta razón que los usuarios deben proteger su contraseña, pues con ello también protegen su identidad.

Se plantea una relación inversa entre seguridad y facilidad de uso o conveniencia como también se expresa en bibliografía especializada, para las técnicas de autenticación y en este caso para las contraseñas. O sea, si algún objeto, con su información asociada o algún recurso esta protegido por una contraseña, se incrementan la seguridad y asociada a ello la pérdida de conveniencia para los usuarios del sistema.

La seguridad de las contraseñas se ve afectada por diversos factores:

1. Fortaleza de la contraseña: Deben ser largas, normalmente más de 7 caracteres, y se deben usar combinaciones de letras mayúsculas y minúsculas, números y símbolos. Ejemplos de contraseñas fuertes serían las siguientes: tastY=wheeT34, pArtei@34! y #23kLLflux.
2. Formas de almacenar las contraseñas: Se debe usar un algoritmo criptográfico irreversible (o función resumen), los más comunes son MD5 y SHA1.
3. Método de retransmisión de la contraseña: Deben ser transmitidas mediante algún método criptográfico.
4. Longevidad de la contraseña: Deben ser cambiadas con cierta periodicidad.[10]

1.3. Soluciones que proponen una seguridad centralizada para el control de acceso.

Actualmente existen soluciones para el control de la seguridad de varias aplicaciones de manera centralizada, o sea, en un entorno de varias aplicaciones controlarlas a todas de igual forma, pero hay que destacar que estas propuestas son incipientes todavía por lo que en muchas ocasiones son miradas con recelos por los clientes.

Una de las soluciones que proponen una seguridad centralizada es:

AccessMaster IAM (Gestión de Identidades y Acceso) & SSO (Single Sign-On).

Los sistemas de información corporativos incluyen un número creciente de aplicaciones heterogéneas y recursos alojados en diversos sistemas abiertos. Mientras esta variedad y heterogeneidad facilita los procesos de negocio, 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. [4]

El principio de las soluciones de Gestión de Identidades y Accesos (AccessMaster IAM) es el poder establecer, mediante políticas de control de acceso, qué usuarios pueden acceder a que aplicaciones y recursos, de manera que un usuario no pueda usar aplicaciones o entrar en recursos para los que no está autorizado. [4]

Otro valor añadido de esta solución es la gestión centralizada y segura de usuarios y contraseñas para los distintos servicios de la organización.

El Single Sign-On (SSO) supone efectuar en lugar del usuario la operación de identificarse mediante login y password frente a las aplicaciones y recursos corporativos. Los usuarios se autentican una única vez, contra el sistema de IAM y SSO, y después este sistema se encarga de forma transparente de las autenticaciones subsiguientes en su lugar, según se van produciendo los accesos correspondientes. [4]

1.4. Tendencias y tecnologías actuales a considerar.

1.4.1. Protocolo Simple de Acceso a Objetos (SOAP).

Es un protocolo de mensajes entre computadoras. SOAP especifica el formato de mensaje que accede e invoca a los objetos y permite solucionar los problemas de las tecnologías que desarrollan aplicaciones que trabajen sobre Internet, estos problemas son la falta de interoperabilidad, la dependencia a la arquitectura de trabajo, así como al lenguaje de programación.

SOAP es un protocolo ligero para el intercambio de información en un entorno distribuido y descentralizado. Esta basado en el protocolo XML.

Algunas de las Ventajas de SOAP son:

- No esta asociado con ningún lenguaje
- No se encuentra fuertemente asociado a ningún protocolo de transporte
- Permite la interoperabilidad entre múltiples entornos. [5]

1.4.2. Arquitectura en capas.

Una buena arquitectura de software debe facilitar los requerimientos de mantenimiento, reusabilidad, escalabilidad, y robustez del mismo. Al concertar la solución de un problema como una serie de capas, cada capa debe ocuparse de un subconjunto de responsabilidades fuertemente acopladas y tener poca cohesión con las demás. Los cambios internos en cualquier capa deben ocasionar la menor cantidad posible de cambios en las restantes. [1]

La primera capa se denomina capa de presentación y normalmente consiste en una interfaz gráfica de usuario de algún tipo. La capa intermedia, consiste en la aplicación o lógica del negocio, y la tercera capa, la capa de datos, contiene los datos necesarios para la aplicación.

La capa intermedia (lógica de aplicación) es básicamente el código al que recurre la capa de presentación para recuperar los datos deseados. La capa de presentación recibe entonces los datos y los formatea para su presentación. Esta separación entre la lógica de aplicación de la interfaz de usuario añade una enorme flexibilidad al diseño de la aplicación. Pueden construirse y desplegarse múltiples interfaces de usuario sin cambiar en absoluto la lógica de aplicación siempre que esta presente una interfaz claramente definida a la capa de presentación. [6]

La tercera capa contiene los datos necesarios para la aplicación. Estos datos consisten en cualquier fuente de información, incluido una base de datos de empresa, un conjunto de documentos XML o incluso un servicio de directorio como el servidor LDAP (Lightweight Directory Access Protocol). [6]

Una ventaja evidente de este modelo es que la capa de presentación puede desarrollarse de variadas maneras simultáneamente: cliente Web, aplicación Windows, aplicación para otro Sistema Operativo, entre otras. Mientras menos responsabilidades recaigan en esta capa mayor será la facilidad de

desarrollar múltiples versiones de la misma. Otra ventaja sería la posibilidad de emigrar de servidor de bases de datos ocasionándole mínimos cambios en el sistema, en tal caso los cambios se concentrarían en la capa de datos, quizás hubiera que hacer pequeños ajustes en la capa de negocio, pero nunca en la capa de presentación.

1.4.3. AJAX.

AJAX, acrónimo inglés de Asynchronous Javascript and XML (Javascript y XML asíncrono) es una técnica de desarrollo Web para crear aplicaciones. [7]

El hecho de que el intercambio de datos se realice de forma asíncrona permite que las aplicaciones Web funcionen de una manera casi transparente al usuario en términos de comunicación con el servidor. En el modelo clásico, cada vez que se quiere cargar una nueva página Web con nuevos datos, se envía una petición al servidor Web, y este devuelve la página entera, que incluye tanto los datos a mostrar como la presentación de la misma.

Sin embargo, al utilizar el modelo AJAX, cuando se quiere cargar datos nuevos, se envía una petición HTTP al servidor Web que devuelve únicamente los datos necesarios. Con este sistema se consigue reducir el volumen de tráfico entre cliente y servidor, y también que no se tengan que cargar páginas HTML enteras cada vez que se quieren representar nuevos datos. [7]

En resumen, el uso de la técnica AJAX proporciona las siguientes ventajas:

- Las aplicaciones son más interactivas y responden a las peticiones del usuario más rápidamente, al estilo escritorio.
- Estas aplicaciones tienen una apariencia muy similar a las aplicaciones de escritorio tradicionales sin depender de plugins o características específicas de los navegadores.
- Se reduce el tamaño de la información intercambiada (muchas micro-peticiones, pero el flujo de datos global es inferior)
- Se libera de procesamiento a la parte servidora (se realiza en la parte cliente). [7]

1.4.4. WSDL

Web Services Description Language (WSDL) describe la interfaz pública a los servicios Web. Está basado en XML y describe la forma de comunicación, es decir, los requisitos del protocolo y los formatos de los mensajes necesarios para interactuar con los servicios listados en su catálogo. Las operaciones y mensajes que soporta se describen en abstracto y se ligan después al protocolo concreto de red y al formato del mensaje.

1.4.5. LDAP

LDAP (Lightweight Directory Access Protocol) es un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red. [8]

Habitualmente, almacena la información de login (usuario y contraseña) y es utilizado para autenticarse aunque es posible almacenar otra información (datos de contacto del usuario, ubicación de diversos recursos de la red, permisos, certificados, entre otros).

¿Qué es un directorio?

Un directorio es una base de datos, pero en general contiene información más descriptiva y más basada en atributos. La información contenida en un directorio normalmente se lee mucho más de lo que se escribe. Como consecuencia los directorios no implementan normalmente los complicados esquemas para transacciones o esquemas de reducción que las bases de datos utilizan para llevar a cabo actualizaciones complejas de grandes volúmenes de datos, Las actualizaciones en un directorio son usualmente cambios sencillos de todo o nada, si es que permiten algo. [8]

1.4.6. Lenguajes de Programación Web a utilizar.

La programación Web, parte de las siglas WWW, que significan World Wide Web o telaraña mundial. [14]

Con el comienzo de Internet y la programación web, se desfasaron los diseños gráficos tradicionales, con lo que se empezaron a diseñar interfaces concretas para este medio, buscando ficheros pequeños para facilitar la carga de los mismos. La programación web se orientaba a un diseño muy cargado e interactuando con el usuario, mientras que al empezar a competir con millones de webs se ha optado más por el diseño sencillo y de fácil comprensión.

Siempre es difícil elegir el lenguaje sobre el que se va a trabajar cuando se trata de aplicaciones web, porque existen varios y todos tienen características positivas y distintivas que influyen la hora de tomar la decisión.

Para la realización de esta aplicación se decidió usar PHP por las características que posee.

PHP es el acrónimo de Hipertext Preprocesor. Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, seguro, con una gran librería de funciones y mucha documentación. Se escribe dentro del código HTML, lo que lo hace realmente fácil de utilizar. [9]

Es independiente de plataforma, puesto que existe un módulo de PHP para casi cualquier servidor web. Esto hace que cualquier sistema pueda ser compatible con el lenguaje, lo que hace de esto una ventaja importante, ya que permite exportar el sitio desarrollado en PHP de un sistema a otro sin prácticamente ningún trabajo.

Por último señalábamos la seguridad, en este punto también es importante el hecho de que en muchas ocasiones PHP se encuentra instalado sobre servidores Unix o Linux, que son conocidos como más veloces y seguros que el sistema operativo Windows NT o 2000. Además, PHP permite configurar el servidor de modo que se permita o rechacen diferentes usos, lo que puede hacer al lenguaje más o menos seguro dependiendo de las necesidades de cada cual.

Este lenguaje de programación está preparado para realizar muchos tipos de aplicaciones web gracias a la extensa librería de funciones con la que está dotado. La librería de funciones cubre desde cálculos matemáticos complejos hasta tratamiento de conexiones de red, por poner dos ejemplos.

Algunas de las más importantes capacidades son: compatibilidad con las bases de datos más comunes, como MySQL y Oracle, por ejemplo. Además incluye funciones para el envío de correo electrónico, upload de archivos, crear dinámicamente en el servidor imágenes en formato GIF, incluso animadas y una lista interminable de utilidades adicionales.

JavaScript es un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento.

Tiene muchas posibilidades, permite la programación de pequeños scripts, pero también de programas más grandes, orientados a objetos, con funciones, estructuras de datos complejas, etc. Además, Javascript pone a disposición del programador todos los elementos que forman la página web, para que éste pueda acceder a ellos y modificarlos dinámicamente.

Tiene como características principales las siguientes:

- Es interpretado (que no es compilado) por el cliente.
- Está basado en objetos. No es, como Java, un lenguaje de programación orientada a objetos (OOP).
- JavaScript no emplea clases ni herencia, típicas de la OOP.
- No es necesario declarar los tipos de variables que van a utilizarse.
- Las referencias a objetos se comprueban en tiempo de ejecución, por lo tanto no se compila.
- No puede escribir automáticamente al disco duro.

Una ventaja que presenta JavaScript es que permite crear páginas más dinámicas, haciéndolas más atractivas para el usuario.

1.4.7. Sistema Gestor de Bases de Datos (SGBD) a utilizar.

Rapidez, efectividad en los procesos y los grandes flujos de información están como primera necesidad a la hora de optimizar servicios y productos. Ante esta notable demanda de soluciones informáticas han surgido muchos gestores de bases de datos, estos programas permiten manejar la información de modo sencillo y prestan servicios para el desarrollo y el manejo de bases de datos.

Un SGBD debe proporcionar a los usuarios la capacidad de almacenar datos en la base de datos, acceder a ellos y actualizarlos. Esta es la función fundamental de un SGBD.

Debe proporcionar un mecanismo que garantice que todas las actualizaciones correspondientes a una determinada transacción se realicen, o que no se realice ninguna. Una transacción es un conjunto de acciones que cambian el contenido de la base de datos.

Debe proporcionar un mecanismo que garantice que sólo los usuarios autorizados pueden acceder a la base de datos. La protección debe ser contra accesos no autorizados, tanto intencionados como accidentales.

PostgreSQL es un sistema de base de datos profesional típico de Unix. Dispone de una serie de funcionalidades que caracterizan a las bases de datos de altas prestaciones que lo hacen apto para la mayoría de las aplicaciones. Es más avanzado que MySQL, el sistema de base de datos estándar que se emplea en blogs, portales, foros, webs personales, etcétera, aunque es más lento, y sus capacidades no se aprovechan normalmente, por lo que es menos popular que MySQL. Al igual que MySQL, Postgre es gratuito.

Entre las facilidades que brinda PostgreSQL podemos mencionar:

- Restauración continua de la base de datos. Es decir, puedes volver a un punto concreto. Es de suponer que esto supone una carga más para el sistema, pero es una opción interesante.
- Mejoras de rendimiento y decisiones sobre el sistema de ficheros donde quieres guardar tus cosas.
- Cambio de tipos de campo con alter table.

Debido a sus características y a las facilidades que brinda, se escogió este como sistema gestor de bases de datos. Además cuenta con la característica de ser software libre y es el SGBD que se usa en las instalaciones de UCID.

1.4.8. Proceso Unificado de desarrollo de software.

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.

Contiene muchas de las mejores prácticas en el desarrollo de software. Le proporciona a cada miembro del equipo las pautas, plantillas y herramientas, ayudándolos a que produzcan, dentro de un horario predecible, con un presupuesto razonable y con alta calidad para satisfacer las necesidades de los usuarios.

RUP tiene tres características fundamentales:

- Guiado por los casos de uso.
- Centrado en la arquitectura.
- Iterativo e incremental.

1.4.9. Lenguaje de modelación.

UML es un lenguaje de modelado de sistemas de software ampliamente reconocido y utilizado. Es un lenguaje gráfico que se utiliza para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software.

Este además ofrece un estándar para describir un plano del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables. Puede soportar diferentes metodologías de desarrollo de software, como es el caso de RUP, pero no especifica cual de ellas usar. También permite modelar desde complejos sistemas para empresas hasta sistemas basados en web o sistemas de tiempo real. Es un lenguaje muy expresivo, que cubre todas las vistas necesarias para desarrollar un sistema.

1.4.10. Herramienta de Modelado Visual y Desarrollo a utilizar.

Visual Paradigm para UML es una herramienta de modelado profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. [11]

El diseño es centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad. Esta herramienta usa un lenguaje estándar común para todo el equipo de desarrollo que facilita la comunicación y disponibilidad de múltiples versiones, para cada necesidad.

Entre las principales ventajas que proporciona el Visual Paradigm para UML podemos citar:

- Integración completa con Microsoft Office.
- Es multiplataforma y muy útil para la generación de código fuente en PHP.
- Tiene la capacidad de crear el esquema de clases a partir de una base de datos y crear la definición de base de datos a partir del esquema de clases.
- Permite invertir código fuente de programas, archivos ejecutables y binarios en modelos UML al momento, creando de forma simple toda la documentación.
- Incorpora el soporte para trabajo en equipo, que permite que varios desarrolladores trabajen a la vez en el mismo diagrama y vean en tiempo real los cambios hechos por sus compañeros de equipo.

1.4.11. Ext.

Ext es un framework JavaScript del lado del cliente para el desarrollo de aplicaciones web. Tiene un sistema dual de licencia: Comercial y Open Source. Este framework puede correr en cualquier plataforma que pueda procesar POST y devolver datos estructurados (PHP, Java, .NET y algunas otras). [12]

Basa toda su funcionalidad en JavaScript a través de librerías ya muy conocidas. En tiempo de ejecución carga y crea todos los objetos html a través del uso intenso de DOM. [13]

Ventanas, mensajes emergentes, date pickers y muchas otras utilidades son todas creadas en tiempo de ejecución. Alguien preguntará ¿y los datos? Los datos son obtenidos con mucho AJAX a través de XML y/o JSON.

Ventajas:

- La orientación a objetos intensa te hará modular todos tus scripts
- El diseño está completamente separado de la funcionalidad.
- Funciones comunes como validación, comboxes editables, ventanas arrastables (con minimizar y maximizar)y grillas editables, son muy fáciles de implementar.
- Buena y amplia documentación, así como también su comunidad. [13]

1.5. Conclusiones.

En este capítulo se realizó un estudio referente a la seguridad en aplicaciones Web. También se hizo una investigación para conocer las ventajas que ofrece el lenguaje de programación escogido para realizar la aplicación que en este caso fue PHP por todas las facilidades que brinda, además por ser el lenguaje usado en las aplicaciones desarrolladas en UCID. También se decidió usar PostgreSQL como sistema gestor de base de datos por sus características y la metodología de desarrollo de software a utilizar fue el Proceso Unificado de desarrollo de software.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.1. Introducción

En este capítulo se muestra como se realiza el control de la seguridad en las instituciones de las FAR, los conceptos fundamentales del problema y todo lo referente a los requerimientos. Además se definen los casos de uso del sistema, los trabajadores que intervienen en los mismos y los diagramas de casos de uso del sistema.

2.2. Objeto de estudio

La seguridad en las aplicaciones web se basa en muchos aspectos, entre ellos se encuentran los procesos de autenticación, autorización y administración de perfiles, los mismos son de gran importancia ya que los recursos más importantes que tiene una empresa o institución son los datos. Es necesario mantener una seguridad estricta para lograr que accedan a los recursos sólo las personas autorizadas. Se puede implementar una seguridad para cada aplicación web, pero esta opción no es la ideal. La mejor opción sería realizar una administración centralizada de estos tres aspectos.

2.2.1. Problema y situación problémica:

En la actualidad en los centros de las FAR no se cuenta con un sistema que controle los procesos de autenticación, autorización y administración de perfiles de forma centralizada, o sea, cada sistema implementa su propia seguridad, esto trae consigo que se invierta mucho más tiempo de desarrollo en cada aplicación y que los usuarios que tengan que acceder a más de una aplicación tengan que memorizar varios nombres de usuarios y contraseñas. Además en las instituciones de las FAR por la sensibilidad de la información que se maneja se requiere una seguridad mucho más estricta para evitar fugas de información.

El año pasado en UCID se realizó un sistema similar al que se diseñó en esta ocasión. A pesar de funcionar y cumplir sus objetivos se detectaron una serie de aspectos que no se tuvieron en cuenta a la hora de concebirlo, como la autenticación contra LDAP. Además las interfaces de usuarios

diseñadas presentaron problemas a la hora de realizar las actualizaciones deseadas por el usuario y no eran las más apropiadas ya que no mostraban un entorno amigable para el usuario a la hora de realizar todas las acciones deseadas en el sistema. Es por esto que se decidió realizar nuevamente el análisis y diseño del mismo ya que es este sistema el que se piensa implantar en todos los centros de las FAR.

El objetivo estratégico de la organización es lograr una seguridad acorde a la que se requiere en las instituciones de las FAR.

2.2.2. Objeto de automatización

Los procesos a automatizar son la autenticación, autorización y administración de perfiles de manera centralizada.

Autenticación: Proceso utilizado en los mecanismos de control de acceso con el objetivo de verificar la identidad de un usuario, dispositivo o sistema mediante la comprobación de credenciales de acceso.

Autorización: Es el proceso por el que permite o deniega el acceso de un usuario a un recurso.

Administración de Perfiles: Los perfiles de usuario consisten en información sobre el usuario que puede utilizar la aplicación para personalizar su comportamiento. Un perfil de usuario puede incluir preferencias de la interfaz de usuario (por ejemplo, colores de fondo).

2.2.3. Información que se maneja

El sistema de seguridad debe manejar toda la información sobre los sistemas a los cuales se le brinda seguridad. Por cada sistema se registran: las funcionalidades, las acciones, los roles, los usuarios, los tipos de usuarios, los servicios que brindan cada uno de los sistemas y los servicios que consumen.

2.3. Propuesta de sistema

Para que la aplicación de seguridad brinde los servicios de autenticación, autorización y administración de perfiles primeramente debe tener registrada toda la información referente a cada uno de los sistemas que van a consumir estos servicios, para con esta información restringir el acceso de los usuarios a los recursos y lograr que sólo accedan a lo que le está permitido según su rol. Una vez registradas toda la información necesaria y publicados los servicios que brindan seguridad sólo queda consumir los mismos en el momento necesario para cada sistema externo.

2.4. Modelo del dominio.

En caso de que no se logren determinar los procesos del negocio con claridad, además de identificar quienes son las personas que lo inician y quienes son las que desarrollan las actividades en cada uno de estos procesos. Se realiza el modelo de dominio.

Un modelo de dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las “cosas” que existen o los eventos que suceden en el entorno en el que trabaja el sistema. El modelo de dominio se describe mediante diagramas de UML (especialmente mediante diagramas de clases). De manera general el Modelo del Dominio ayuda a los usuarios, clientes, desarrolladores e interesados a utilizar un vocabulario común para poder entender el contexto en que se sitúa el sistema.

Las clases del dominio aparecen de tres formas típicas:

- Objetos del negocio que representan cosas que se manipulan en el negocio.
- Objetos del mundo real y conceptos de los que el sistema debe hacer seguimiento.
- Sucesos que ocurrirán o han ocurrido.

En este caso se decidió realizar un Modelo de dominio, ya que no se logró identificar con claridad los procesos del negocio con fronteras bien establecidas donde se logren ver claramente. Los conceptos fundamentales que se encuentran en nuestro dominio se relacionan a continuación.

Concepto	Descripción

Usuario	Cualquier persona que interactúa con el sistema y juega un rol determinado en el mismo.
Rol	Papel que juega el usuario dentro del sistema, se le atribuyen uno o más roles a los usuarios para restringir o no las funcionalidades y páginas a las que este tendrá acceso.
Funcionalidad	Hace referencia a un conjunto de acciones, pero no realiza ninguna de forma específica, ejemplo: gestionar usuario. Los usuarios en dependencia de su rol pueden acceder a ciertas funcionalidades.
Acciones	Actividad o acción concreta que realiza un sistema. Generalmente es un método del sistema.
Sistema	Es la aplicación web como tal. Está compuesto por el o los módulos que sean necesarios.
Servicios que brinda	Servicios que brinda una aplicación. Es una acción que hace el sistema y que la publica para que otros la utilicen
Servicios que consume	Servicios que consume una aplicación: hace referencia a una acción que consume un sistema externo que se encuentra publicada.

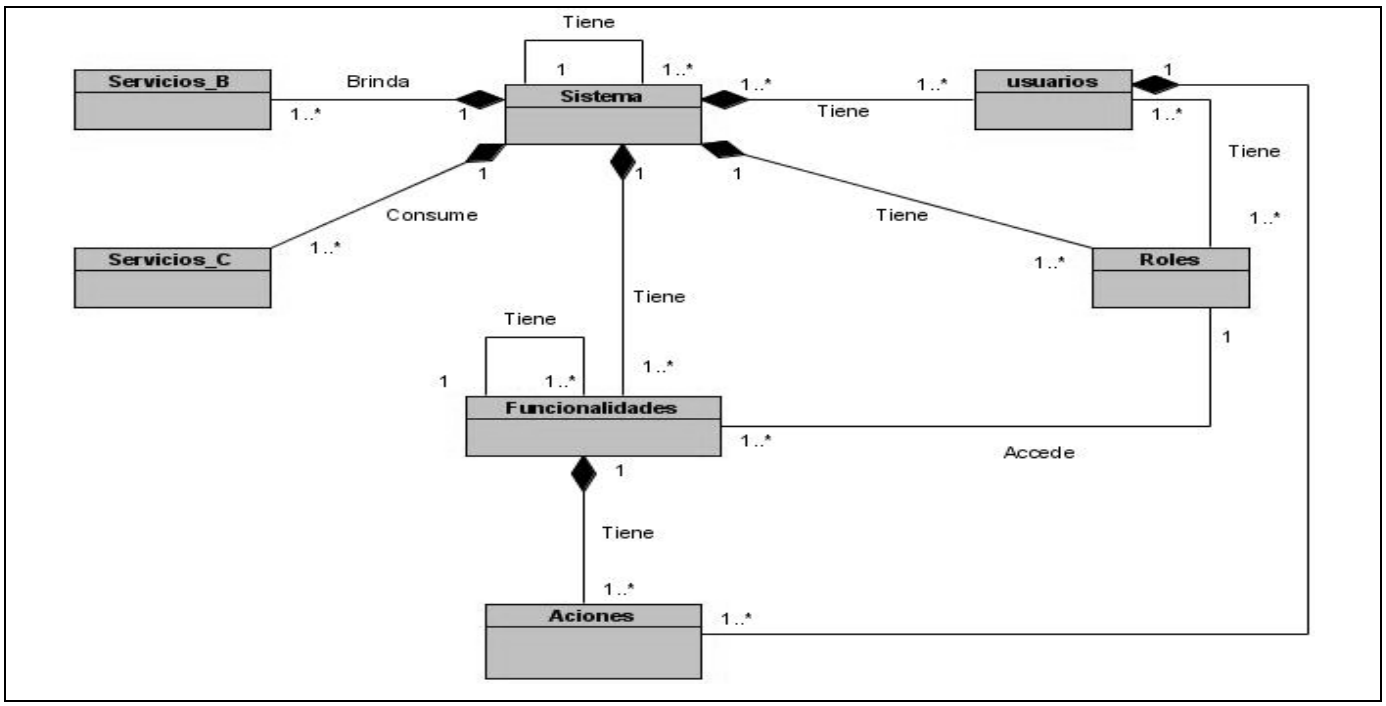


Figura 2.1 Modelo de clases del dominio

2.5. Requisitos que debe cumplir el sistema.

Para dar solución al problema se identificaron los siguientes requisitos con los que debe cumplir el sistema:

2.5.1. Requisitos funcionales.

Los requisitos funcionales son las capacidades o condiciones que el sistema debe cumplir. A continuación se muestran todos los requisitos que debe cumplir nuestro sistema.

1. Gestionar Sistemas.
   1.1. Registrar nuevo sistema.
      1.1.1. Solicitar datos del sistema.
   1.2. Modificar datos de un sistema previamente insertado.
   1.3. Eliminar un sistema previamente insertado.
   1.4. Cambiar padre de un sistema.

2. Gestionar funcionalidades.
   2.1. Registrar nueva funcionalidad.
      2.1.1. Seleccionar sistema y entrar datos.
   2.2. Modificar una funcionalidad.
   2.3. Eliminar una funcionalidad.
   2.4. Cambiar padre de una funcionalidad.

3. Gestionar acciones por funcionalidades.
   3.1. Registrar una nueva acción.
      3.1.1. Seleccionar el sistema, seleccionar la funcionalidad y entrar datos.
   3.2. Modificar una acción.
   3.3. Eliminar una acción.

4. Gestionar roles por sistemas.
   4.1. Registrar nuevo rol.
      4.1.1. Seleccionar el sistema, entrar datos y funcionalidades a las que puede acceder.
   4.2. Modificar un rol previamente insertado.
   4.3. Eliminar un rol previamente insertado.

5. Gestionar usuarios por sistemas.
   5.1. Registrar nuevo usuario.
      5.1.1. Seleccionar sistema, tipo de usuario, entrar datos y rol que juega el usuario en ese sistema.
   5.2. Modificar un usuario previamente insertado.
   5.3. Eliminar un usuario previamente insertado.

5.4. Restringir acciones de las funcionalidades asignadas cuando se registró el usuario.

6. Brindar servicio para autenticar un usuario en el marco de trabajo utilizando LDAP o según la información registrada en nuestro sistema.
   6.1. Recibir petición con parámetros: usuario a autenticar, contraseña.
   6.2. Comprobar validez de datos suministrados y caducidad de la contraseña.
      6.2.1. En caso positivo, autenticar el usuario en el marco de trabajo, se crea certificado digital.
      6.2.2. En caso negativo devolver mensaje de error.

7. Brindar servicio para autorizar acceso de un usuario a los sistemas a los que tiene acceso.
   7.1. Recibir petición con parámetros: certificado digital.
   7.2. Comprobar validez y correspondencia de los datos suministrados.
      7.2.1. En positivo: permitir acceso.
      7.2.2. En caso negativo: devolver mensaje de error.

8. Brindar servicio para cargar el menú de un usuario.
   8.1. Recibir petición con parámetros: certificado digital y sistema.
   8.2. Comprobar validez y correspondencia de los datos suministrados.
      8.2.1. En positivo: permitir acceso.
      8.2.2. En caso negativo: devolver mensaje de error.

9. Brindar servicio para autorizar acceso de un usuario sobre funcionalidades de un sistema registrado.
   9.1. Recibir petición con parámetros: sistema, funcionalidad, certificado digital.
   9.2. Comprobar validez y correspondencia de los datos suministrados.
      9.2.1. En positivo: permitir acceso.
      9.2.2. En caso negativo: devolver mensaje de error.

10. Brindar servicio para autorizar acceso a una acción dentro de una funcionalidad.
    10.1. Recibir petición con parámetros: sistema, funcionalidad, acción, certificado digital.
    10.2. Comprobar validez y correspondencia de los datos suministrados.
       10.2.1. En positivo: permitir acceso.
       10.2.2. En caso negativo: devolver mensaje de error.

11. Brindar servicio para autorizar a un sistema a usar servicios que brindan otros sistemas.
    11.1. Recibir petición con parámetros: sistema consumidor, sistema proveedor, servicio al que se desea acceder y certificado.
    11.2. Comprobar que el sistema consumidor esté autorizado a acceder a ese servicio.
       11.2.1. En positivo: permitir acceso.
       11.2.2. En caso negativo: devolver mensaje de error.

12. Brindar servicio para que un usuario edite su perfil.
    12.1. Recibir petición con parámetros: certificado, sistema, estilo.
    12.2. Comprobar validez de datos suministrados.
       12.2.1. En caso positivo guardar los cambios.
       12.2.2. En caso negativo devolver mensaje de error.

13. Brindar servicio para que un usuario de un sistema externo cambie su contraseña.

13.1. Recibir petición con parámetros: usuario, certificado, sistema, contraseña anterior y la nueva contraseña.
13.2. Comprobar validez de datos suministrados.
13.2.1. En caso positivo cambiar la contraseña.
13.2.2. En caso negativo devolver mensaje de error.

14. Permitir la configuración de las contraseñas.

15. Gestionar servicios que brinda un sistema.
15.1. Registrar un nuevo servicio.
15.2. Modificar un servicio.
15.3. Eliminar un servicio.

16. Gestionar servicios que consume un sistema.
16.1. Registrar un nuevo servicio.
16.2. Modificar un servicio.

17. Brindar servicio para cerrar sesión de un usuario de un sistema externo.
17.1. Recibir petición con parámetros: certificado y sistema
17.2. Comprobar validez de datos suministrados.
17.2.1. En caso positivo cerrar sesión.
17.2.2. En caso negativo devolver mensaje de error.

18. Gestionar tipos de usuarios
18.1. Registrar un nuevo tipo de usuario.
18.2. Modificar tipo de usuario.
18.3. Eliminar un tipo de usuario.

2.5.2. Requisitos no funcionales.

Apariencia o interfaz externa:

- El sistema debe tener una interfaz fácil de usar y amigable para que pueda ser utilizada sin mucha preparación por el usuario.
- Empleo de imágenes y colores identificados con el negocio donde se implantará el sistema.
- Estará diseñado para resolución de 800x600, aunque deberá verse en cualquier resolución superior a esta.

Usabilidad:

- El sistema podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de la computadora.

Rendimiento:

- Los tiempos de respuesta y velocidad de procesamiento de la información serán rápidos, no mayores de 5 segundos para las actualizaciones y 20 para las recuperaciones.

Soporte:

- La aplicación contará antes de su puesta en marcha con un período de pruebas, se le dará mantenimiento, configuración y se brindará el servicio de instalación.

Portabilidad:

- El sistema debe ser multiplataforma, haciendo énfasis en Linux y Windows.

Seguridad:

- Autenticación (Contraseña de acceso).
- Protección contra acciones no autorizadas o que puedan afectar la integridad de los datos.
- La atención al sistema incluyendo, el mantenimiento de las bases de datos así como la salva de la información se realizarán de forma centralizada por el administrador.
- Verificación sobre las acciones irreversibles (eliminaciones).

Políticos culturales:

- El sistema solo podrá ser utilizado en territorio cubano y por las entidades autorizadas por el Ministerio de las FAR.
- El producto no debe contener palabras en otros idiomas.
- El producto debe respetar los términos empleados normalmente por los especialistas en el tema de la esfera que se automatiza.

Legales:

- El sistema está avalado por los tres documentos rectores emitidos en el país para la certificación y validación de los sistemas contables:
  - La Resolución Orden #4 del Ministro de las Fuerzas Armadas Revolucionarias.

Software:

- Para el cliente:
  - Navegador Mozila Firefox.

- Sistema operativo Windows 98 o superior o Linux.

- Para el servidor:
  - Sistema operativo Windows Advancer Server (2000 o superior) o Linux en cualquiera de sus distribuciones.
  - Un servidor Apache 2.0 o superior con módulo PHP 5.0 disponible, este debe estar configurado con la extensión “pgsql” incluida.
  - Un servidor de base de datos PostgreSQL 8.0 o superior.

Hardware:

- Para el servidor:
  - Requerimientos mínimos: Procesador Pentium III a 1GHz de velocidad de procesamiento y 1Gb de memoria RAM.
  - Al menos 40Gb de espacio libre en disco duro.
  - Tarjeta de red.
- Para el cliente:
  - Requerimientos mínimos: Procesador Pentium II a 133Mhz con 128 Mb de memoria RAM.
  - Tarjeta de red.

Restricciones para el diseño e implementación:

- Utilizar los estándares establecidos (codificación, diseño, entre otros).
- Emplear como servidores Web y de bases de datos Apache y PostgreSQL respectivamente.
- Utilizar como lenguaje del lado del servidor al PHP 5.0 o superior y del lado del cliente el JavaScript.

2.6. Descripción del sistema propuesto.

2.6.1. Descripción de los autores del sistema.

Nombre del actor	Descripción
Administrador	Es el administrador del sistema de seguridad, es el que maneja toda la información referente a los

	sistemas y los usuarios.
Sistema externo	Es el sistema que va a usar los servicios que brinda el sistema de seguridad.

2.6.2. Casos de usos del sistema.

Los casos de uso son artefactos narrativos que describen el comportamiento del sistema desde el punto de vista del usuario. Establece un acuerdo entre clientes y desarrolladores sobre las condiciones y posibilidades (requisitos) que debe cumplir el sistema. A continuación se muestran los casos de uso del sistema identificados teniendo en cuenta los requisitos funcionales.

1. Gestionar sistemas.
2. Gestionar funcionalidades.
3. Gestionar acciones.
4. Gestionar roles.
5. Gestionar usuarios.
6. Autenticar usuario.
7. Autorizar acceso a sistemas.
8. Cargar menú.
9. Autorizar acceso a funcionalidad.
10. Autorizar acceso a acción.
11. Autorizar acceso a servicio.
12. Editar perfil.
13. Cambiar contraseña.
14. Configurar contraseña.
15. Gestionar servicios que brinda.
16. Gestionar servicios que consume.
17. Cerrar sesión de usuario.
18. Gestionar tipos de usuarios

2.6.3. Paquetes.

Partiendo de que algunos casos de usos tienen características similares y para un mejor entendimiento de los diagramas se decidió agruparlos en dos paquetes: Sistema y Servicio.

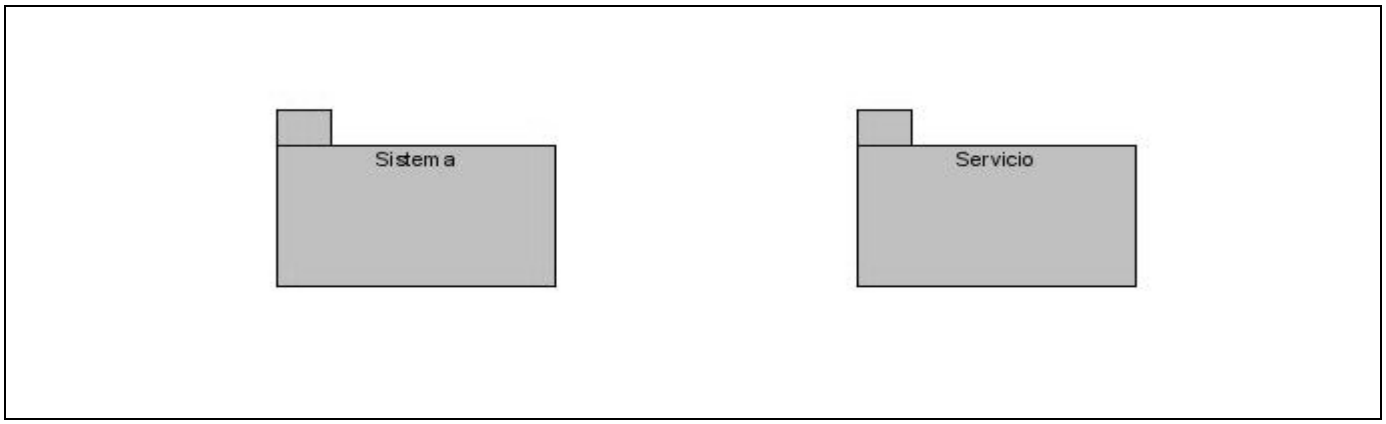


Figura 2.2 Modelo de Paquetes

2.6.4. Diagramas de Casos de Uso del sistema.

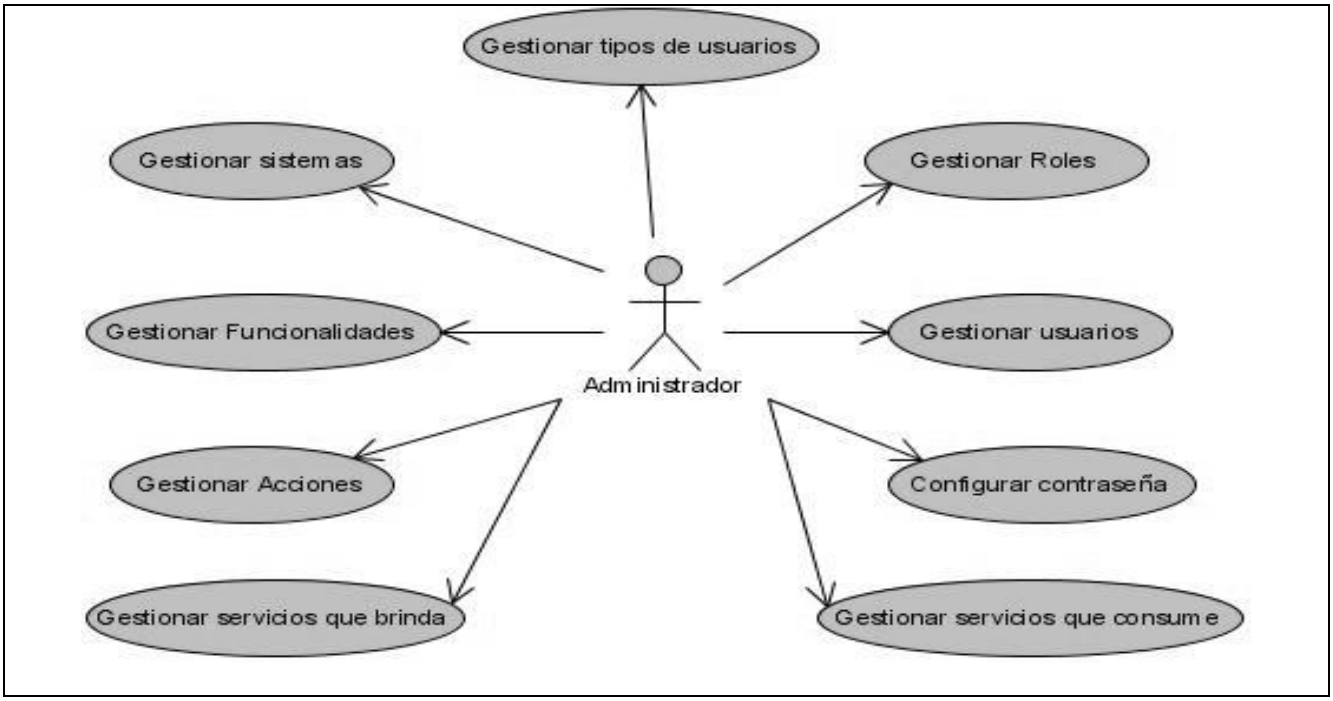


Figura 2.3 Diagrama de CUS Paquete de Sistema

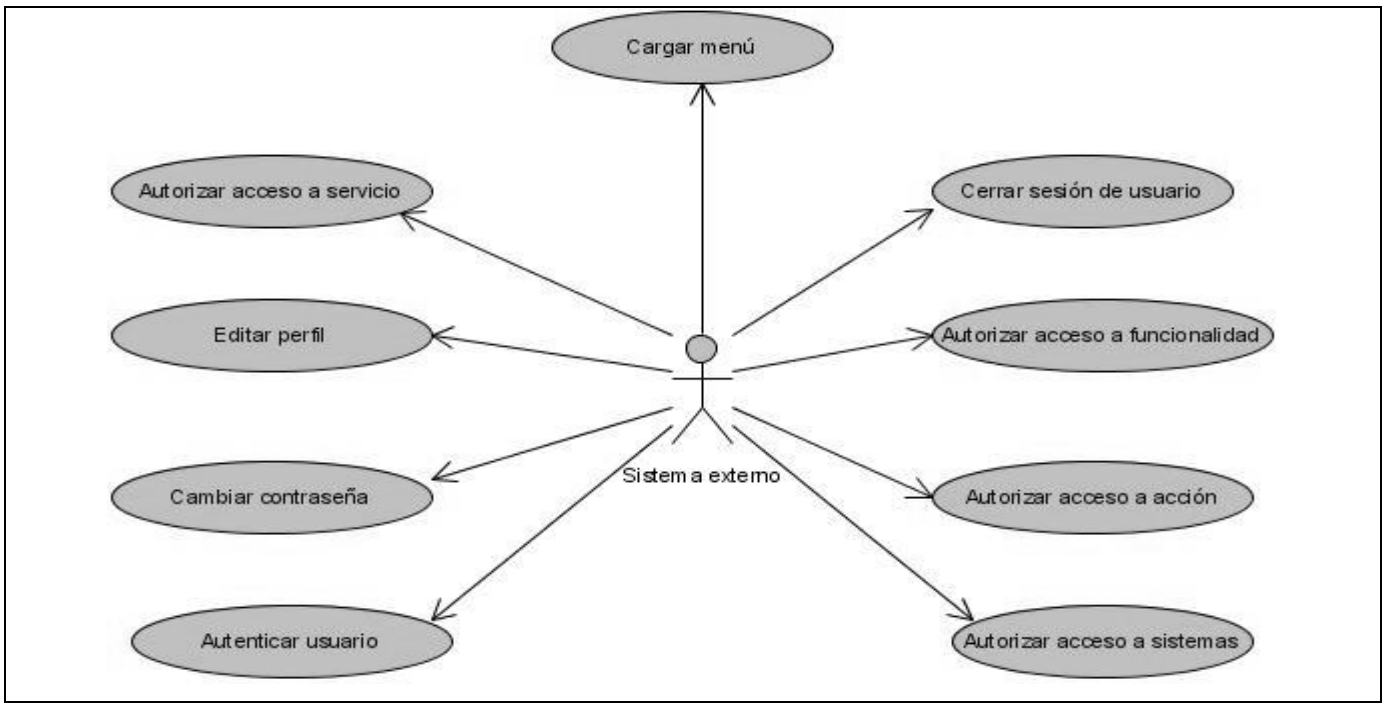


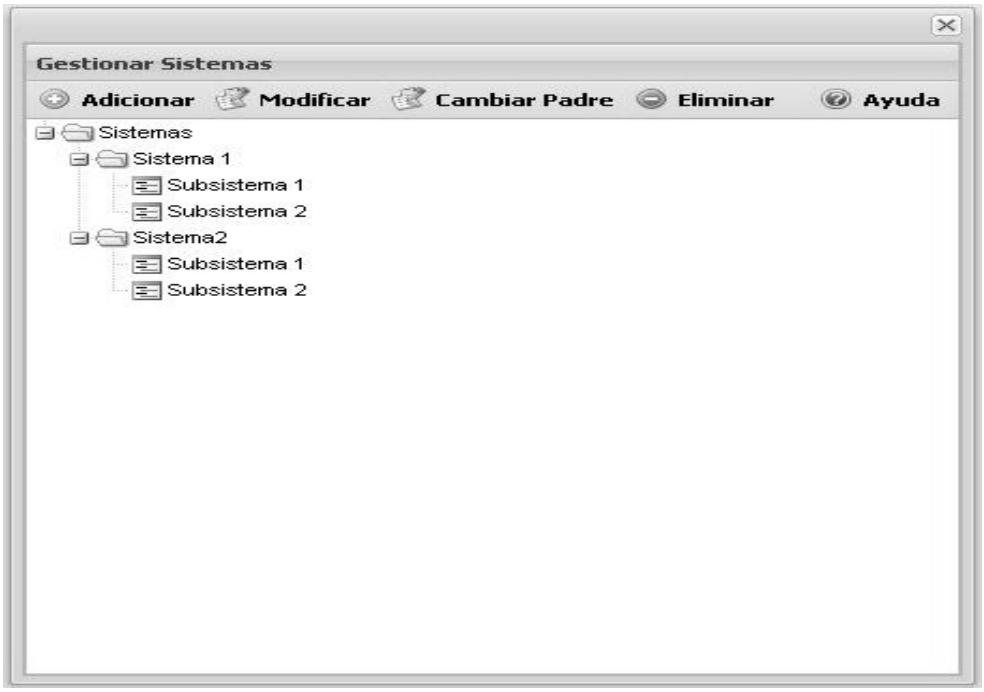
Figura 2.4 Diagrama de CUS Paquete de servicio

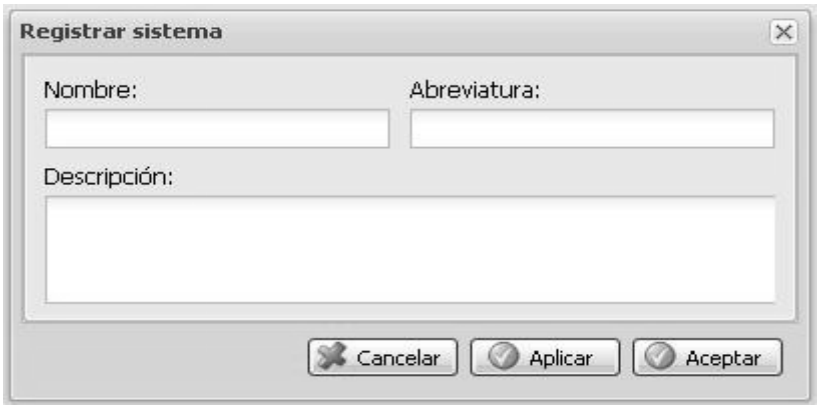
2.6.5. Descripción de los Casos de uso del sistema.

A continuación se muestra la descripción de los Casos de uso del sistema.

Descripción del CUS Gestionar sistemas

Caso de Uso	Gestionar sistemas
Actores	Administrador
Propósito	Permitir Gestionar sistemas
Resumen	El caso de uso se inicia cuando el administrador selecciona la opción Gestionar sistema para registrar, modificar o eliminar un sistema, luego el sistema evalúa la posibilidad de realizar la operación solicitada. El caso de uso termina cuando se realiza o no la acción seleccionada por el actor.
Responsabilidades	R1
CU asociados	
Precondiciones	El administrador debe estar autenticado en el sistema.
Requisitos especiales	-
Descripción	
Interfaz I y II	



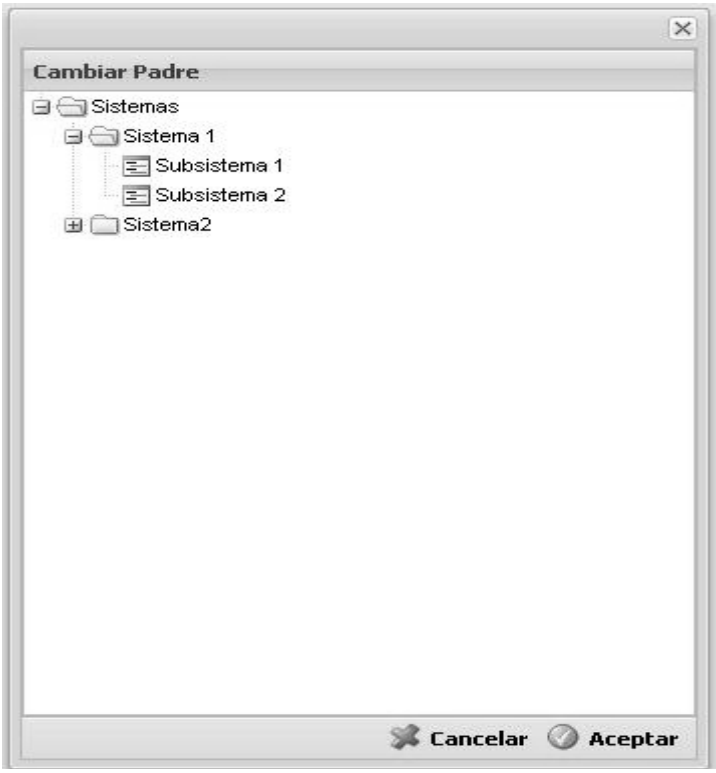


Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. El actor selecciona la opción Gestionar sistemas del menú principal.	2. El sistema muestra la interfaz I con el botón Adicionar activo, además con todos los sistemas en forma de árbol.
3. El actor selecciona la opción: • Adicionar (flujo básico) • Modificar (Ver sección "Modificar sistema") • Eliminar (Ver sección "Eliminar sistema") • Cambiar padre(Ver sección "Cambiar	

padre")	
4. El actor puede registrar el sistema de dos formas: 4.1. El actor acciona el botón Adicionar. 4.2. El actor selecciona el sistema en el cual desea registrar un sistema y acciona el botón Adicionar	5. El sistema muestra la interfaz II con los campos nombre, abreviatura y descripción.
6. El actor llena los campos y acciona el botón Aceptar.	7. El sistema verifica que no existan campos vacíos, que los datos sean correctos y adiciona el sistema.
Flujos alternos	
7. En caso de dejar campos en blanco el sistema muestra un mensaje: "Debe llenar todos los campos." Aceptar. • En caso de que los datos introducidos por el usuario no sean correctos el sistema muestra el mensaje: "Ha introducido datos incorrectos". Aceptar.	
Sección: Modificar sistema	
Acción del Actor	**Respuesta del Sistema**
1. El actor selecciona el sistema que desea modificar.	2. El sistema activa los botones "Modificar, "Eliminar" y "Cambiar padre".
3. El actor acciona el botón Modificar.	4. El sistema muestra la interfaz II con todos los datos del sistema seleccionado.
5. El actor modifica los campos que desea y acciona el botón Aceptar.	6. El sistema verifica que los datos sean válidos, que no existan campos vacíos y guarda los cambios realizados.
Flujos alternos	
6. En caso de que los datos no sean válidos muestra el mensaje: "Los datos introducidos son incorrectos" Aceptar.	
Sección: Eliminar sistema	
Acción del actor	**Respuesta del Sistema**
1. El actor selecciona el sistema que desea eliminar.	2. El sistema activa los botones "Modificar, "Eliminar" y "Cambiar padre", de la interfaz I.
3. El actor acciona el botón Eliminar.	4. El sistema muestra la alerta: "Está seguro que desea eliminar el sistema

	seleccionado." Aceptar Cancelar
5. El actor acciona el botón Aceptar.	6. El sistema elimina el sistema seleccionado.

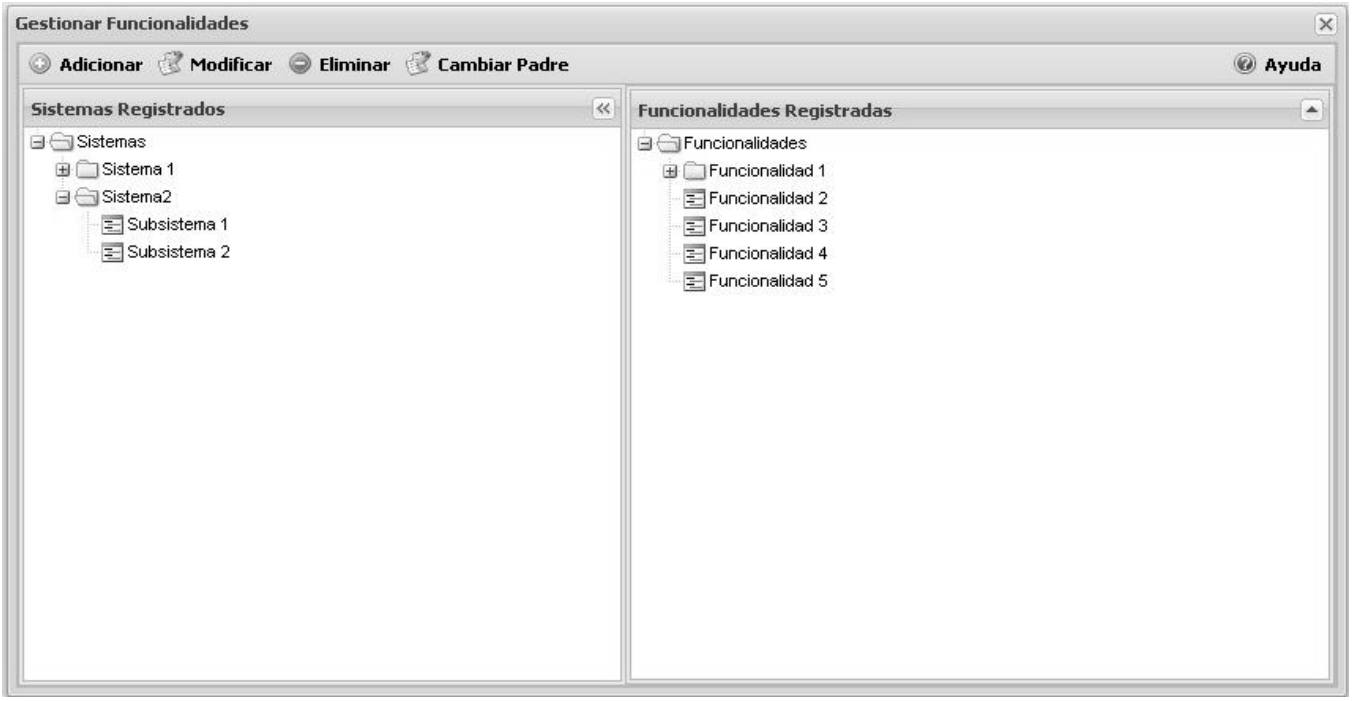
Flujos Alternos
5. En caso de que el actor accione el botón Cancelar se cancela la operación.
Sección: Cambiar padre.
Interfaz III

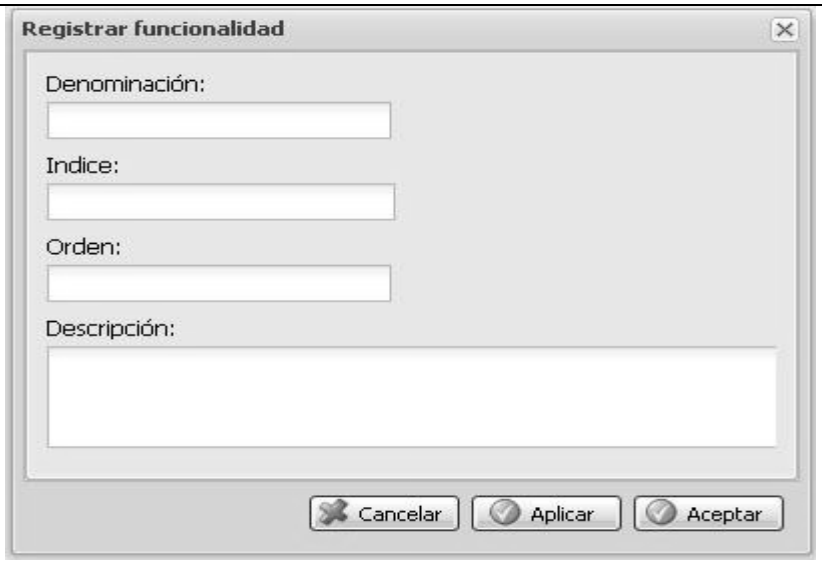


Acción del actor	**Respuesta del Sistema**
1. El actor selecciona el sistema al cual le desea cambiar el padre.	2. El sistema activa todos los botones de la interfaz I.
3. El actor acciona el botón Cambiar padre.	4. El sistema muestra la interfaz III en la que aparece un árbol con todos los sistemas.
5. El actor selecciona el sistema que será padre del anteriormente seleccionado y acciona el botón Aceptar.	6. El sistema cambia el padre.
Postcondiciones	Se actualiza toda la información referente a los sistemas.
Prioridad	Crítica

Descripción del CUS Gestionar funcionalidades.

Caso de Uso	Gestionar funcionalidades
Actores	Administrador
Propósito	Permitir gestionar funcionalidades
Resumen	El caso de uso se inicia cuando el administrador selecciona la opción Gestionar funcionalidades para registrar, modificar o eliminar una funcionalidad, luego el sistema evalúa la posibilidad de realizar la operación solicitada. El caso de uso termina cuando se realiza la acción seleccionada por el actor.
Responsabilidades	R2
CU asociados	-
Precondiciones	El administrador debe estar autenticado en el sistema. Debe haberse registrado un sistema
Requisitos especiales	-
Descripción	
Interfaz I y II	

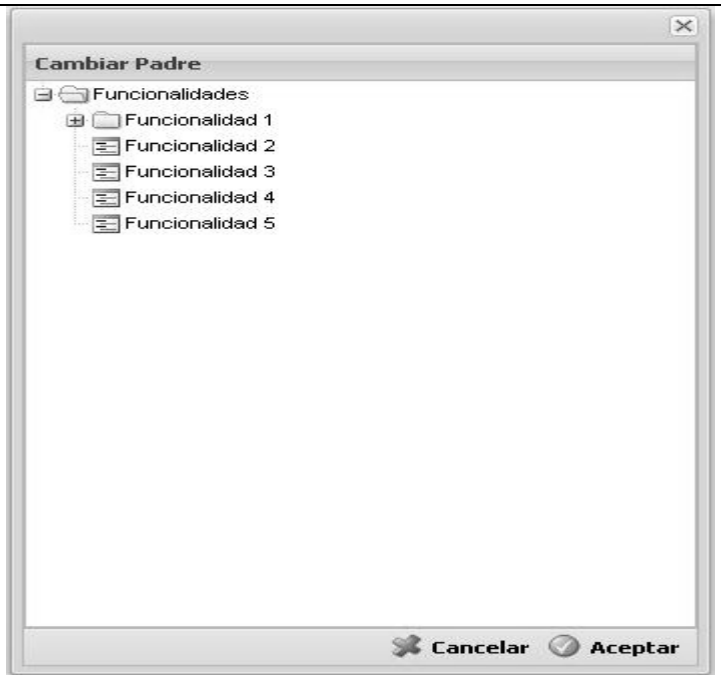




Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. El actor selecciona la opción gestionar funcionalidades del menú principal.	2. El sistema carga los sistemas registrados y los muestra en la interfaz I.
3. El actor selecciona la opción: • Adicionar (flujo básico). • Modificar (Ver sección "Modificar funcionalidad"). • Eliminar (Ver sección "Eliminar funcionalidad"). • Cambiar padre (Ver sección "Cambiar padre").	
4. El actor puede registrar la funcionalidad de dos formas: 4.1. El actor selecciona el sistema en el cual desea registrar una funcionalidad. 4.2. El actor selecciona el sistema y la funcionalidad padre.	5. El sistema muestra las funcionalidades del sistema seleccionado y activa el botón Adicionar
6. El actor acciona el botón Adicionar	7. El sistema muestra la interfaz II con todos los campos (denominación, índice, orden, descripción).
8. El actor introduce los campos y acciona el botón Aceptar.	9. El sistema verifica que los campos hayan sido llenados correctamente y registra la nueva funcionalidad.
Flujos Alternos	
9. En caso de dejar campos en blanco muestra un mensaje: "Debe llenar todos los campos."	

Aceptar. • En caso de que los datos introducidos por el usuario no sean correctos muestra el mensaje: "Ha introducido datos incorrectos" Aceptar.

Sección: Modificar funcionalidades	
Acción del Actor	**Respuesta del Sistema**
1. El actor selecciona el sistema en el cual desea modificar una funcionalidad.	2. El sistema muestra en la interfaz I un árbol con las funcionalidades del sistema seleccionado.
3. El actor selecciona la funcionalidad que desea modificar.	4. El sistema muestra la interfaz II con los botones "Modificar", "Eliminar" y "Cambiar padre" activos.
5. El actor acciona el botón Modificar.	6. El sistema muestra la interfaz II con todos los datos (denominación, índice, orden, descripción) de la funcionalidad seleccionada y con el botón Aceptar y Cancelar activo.
7. El actor realiza los cambios deseados y acciona el botón Aceptar.	8. El sistema verifica que los datos sean válidos y guarda los cambios realizados.
Flujos Alternos	
8. En caso de que los datos no sean válidos muestra el mensaje: "Los datos introducidos son incorrectos" Aceptar.	
Sección: Eliminar funcionalidades	
Acción del actor	**Respuesta del Sistema**
1. El actor selecciona el sistema en el cual desea eliminar una funcionalidad.	2. El sistema muestra las funcionalidades del sistema seleccionado.
3. El actor selecciona la funcionalidad que desea eliminar.	4. El sistema muestra la interfaz II con los botones "Modificar", "Eliminar" y "Cambiar padre" activos.
5. El actor acciona el botón Eliminar.	6. El sistema muestra un la alerta: "Está seguro que desea eliminar el sistema seleccionado." Aceptar Cancelar
7. El actor acciona el botón Aceptar.	8. El sistema elimina la funcionalidad.
Flujos Alternos	
2. En caso de que el actor accione el botón Cancelar se cancela la operación.	
Sección: Cambiar padre	
Interfaz IV	



Acción del actor	Respuesta del Sistema
1. El actor selecciona el sistema en el cual se encuentra la funcionalidad que se desea cambiar padre.	2. El sistema muestra todas las funcionalidades del sistema seleccionado.
3. El actor selecciona la funcionalidad a la que le desea cambiar el padre y acciona el botón Cambiar padre	4. El sistema carga todas las funcionalidades registradas en ese sistema y las muestra en la interfaz IV.
5. El actor selecciona la funcionalidad que será padre y acciona el botón Aceptar.	6. El sistema establece como padre de la funcionalidad, la funcionalidad seleccionada.
Postcondiciones	Se actualiza toda la información referente a las funcionalidades.
Prioridad	Crítica

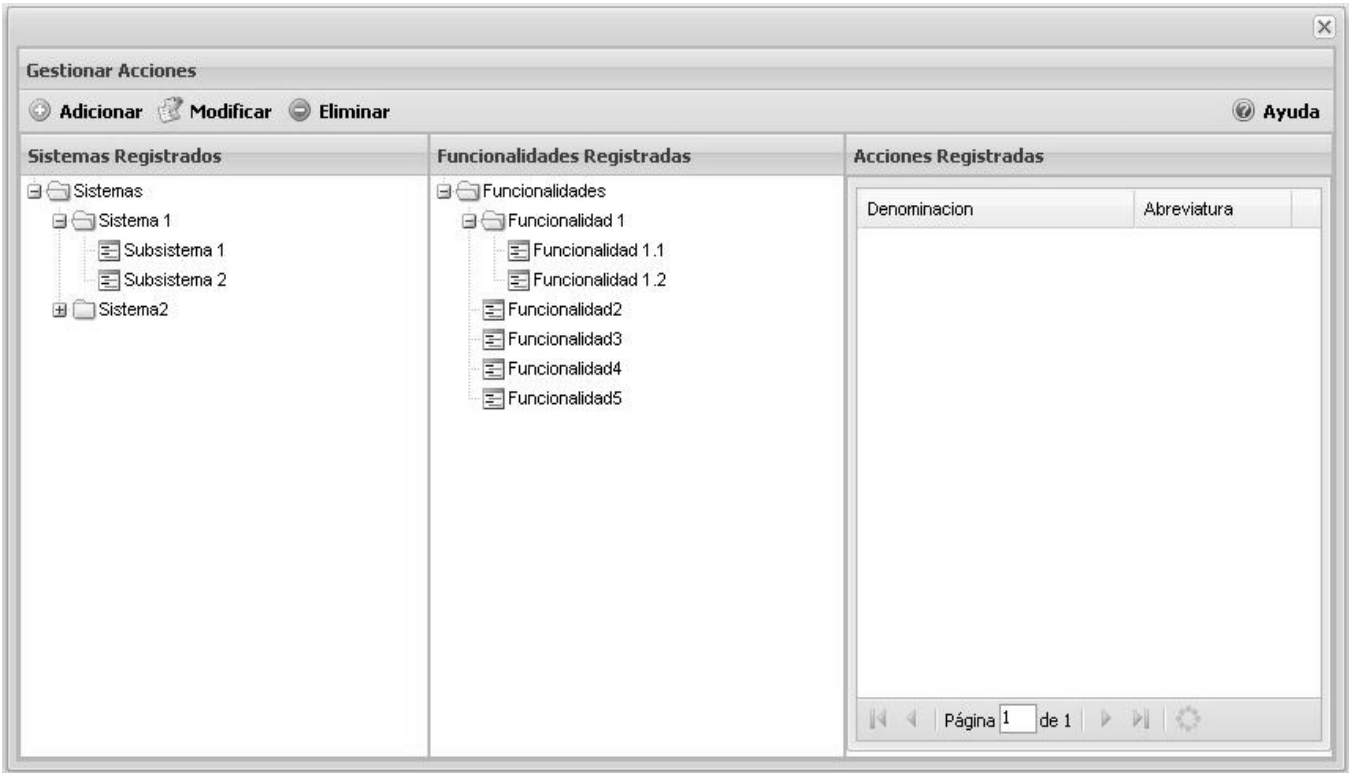
Descripción del CUS Gestionar acciones

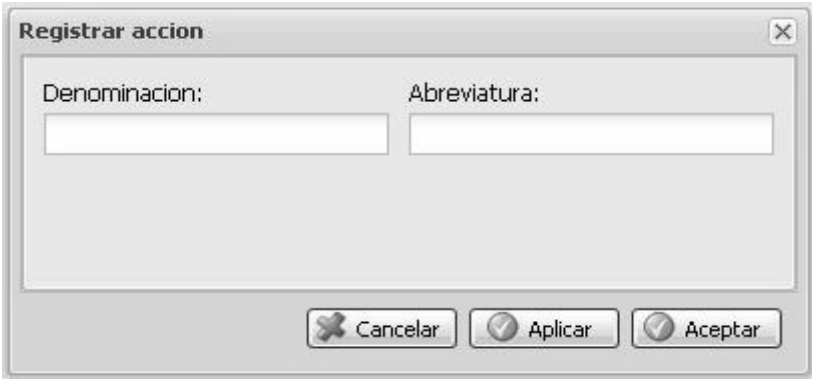
Caso de Uso	Gestionar acciones
Actores	Administrador
Propósito	Permitir insertar, modificar y eliminar acciones.
Resumen	El caso de uso se inicia cuando el administrador selecciona la opción Gestionar acciones para registrar, modificar o eliminar una acción, luego el sistema evalúa la posibilidad de realizar la operación solicitada. El caso de uso termina cuando se realiza la acción seleccionada por el actor.
Responsabilidades	R3
CU asociados	-
Precondiciones	El administrador debe estar autenticado en el sistema, debe haberse registrado

	al menos un sistema y funcionalidades en el mismo.
Requisitos especiales	-

Descripción

Interfaz I y II





Flujo Normal de Eventos

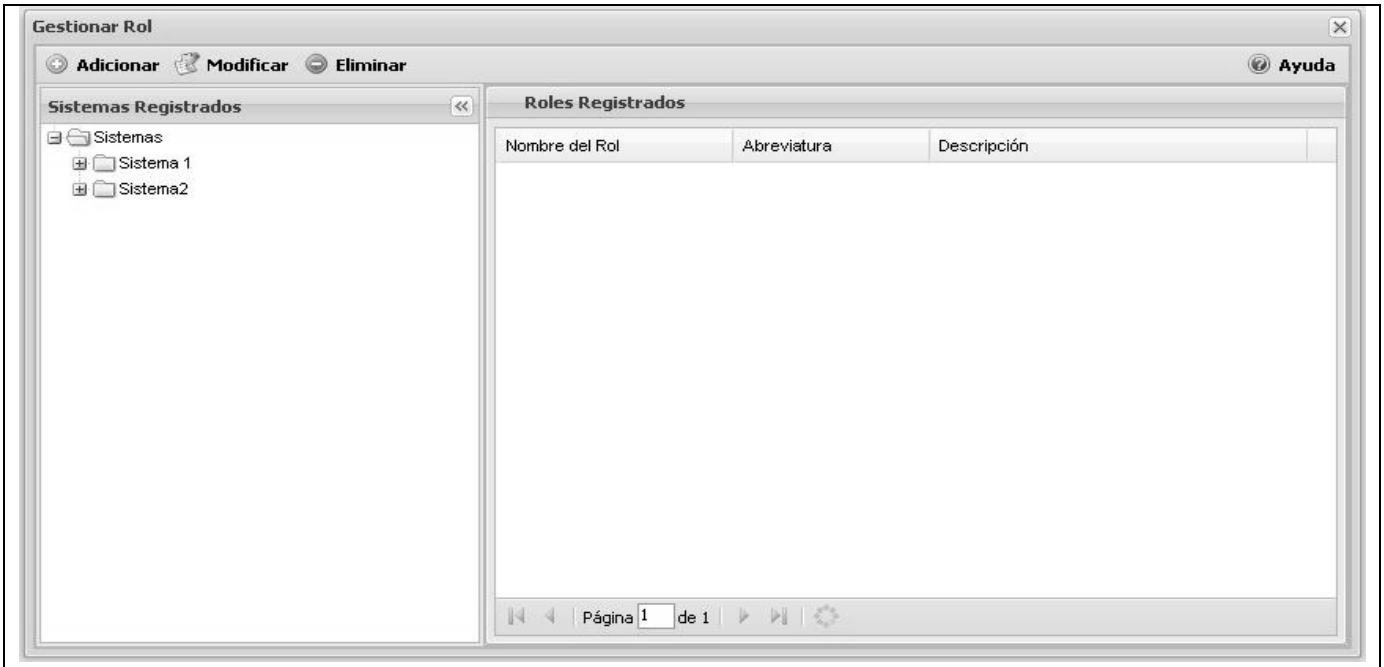
Acción del Actor	Respuesta del Sistema
1. El actor selecciona la opción gestionar acciones del menú principal.	2. El sistema carga los sistemas registrados y los muestra en la interfaz I en forma de árbol.
3. El actor selecciona el sistema en el cual	4. El sistema muestra en la interfaz I un

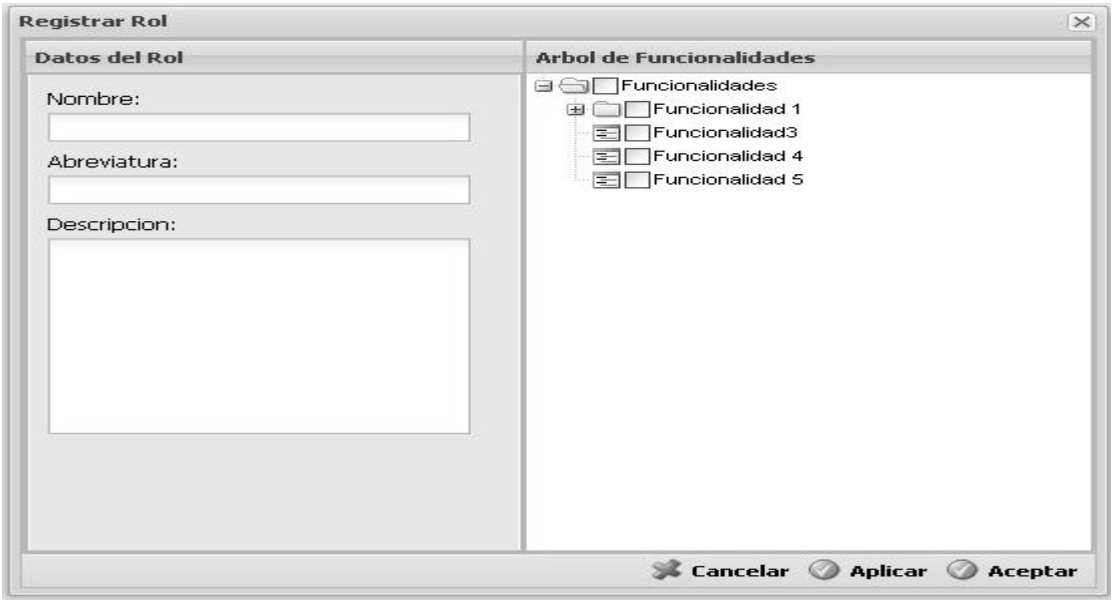
desea gestionar acciones.	árbol con las funcionalidades del sistema seleccionado.
5. El actor selecciona la funcionalidad en la cual desea gestionar acciones.	6. El sistema activa el botón Adicionar y muestra en la interfaz I una tabla con todas las acciones correspondientes a la funcionalidad seleccionada.
7. El actor selecciona: • Adicionar (Flujo normal). • Modificar (Ver sección "Modificar acción"). • Eliminar (Ver sección "Eliminar acción").	8. El sistema muestra la interfaz II con todos los campos (denominación, abreviatura, icono).
9. El actor llena los campos mostrados y acciona el botón Aceptar.	10. El sistema comprueba la validez de los datos y registra la nueva acción.
Flujos Alternos	
10. En caso de dejar campos en blanco muestra el mensaje: "Debe llenar todos los campos." Aceptar • En caso de que los datos introducidos por el usuario no sean correctos muestra el mensaje: "Ha introducido datos incorrectos" Aceptar	
Sección: Modificar acción	
Acción del Actor	**Respuesta del Sistema**
1. El actor selecciona el sistema en el cual está la acción que quiere modificar.	2. El sistema muestra en el árbol de funcionalidades todas las correspondientes al sistema seleccionado.
3. El actor selecciona la funcionalidad en la que se encuentra la acción que desea modificar.	4. El sistema muestra una tabla con todas las acciones que tiene la funcionalidad seleccionada.
5. El actor selecciona la acción que desea modificar.	6. El sistema muestra la interfaz I con los botones "Modificar" y "Eliminar" activos.
7. El actor acciona el botón Modificar	8. El sistema muestra la interfaz II con todos los datos de la acción seleccionada (denominación, abreviatura, icono).
9. El actor modifica los campos de su interés y acciona el botón Aceptar.	10. El sistema comprueba la validez de los datos y guarda los cambios realizados.
Flujos Alternos	
10. En caso de que los datos no sean válidos muestra el mensaje: "Los datos introducidos son incorrectos" Aceptar.	
Sección: Eliminar acción	
Interfaz (Relacionada con la sección si existe)	
Acción del actor	**Respuesta del Sistema**
1. El actor selecciona el sistema en el que se encuentra la acción que desea eliminar.	2. El sistema muestra en el árbol de funcionalidades todas las

	correspondientes al sistema seleccionado
3. El actor selecciona la funcionalidad en la que se encuentra la acción que desea eliminar.	4. El sistema muestra una tabla con todas las acciones correspondientes a la funcionalidad seleccionada.
5. El actor selecciona la acción que desea eliminar.	6. El sistema muestra la interfaz I con los botones “Modificar” y ”Eliminar” activos.
7. El actor acciona el botón Eliminar.	8. El sistema muestra la alerta: ¿Está seguro que desea eliminar la acción? Aceptar Cancelar
9. El actor acciona el botón Aceptar.	10. El sistema elimina la acción seleccionada.
Flujos Alternos	
9. En caso de que el actor accione el botón Cancelar se cancela la operación.	
Postcondiciones	Se actualiza toda la información referente a las acciones.
Prioridad	Crítica

Descripción del CUS Gestionar roles

Caso de Uso	Gestionar roles
Actores	Administrador
Propósito	Permitir gestionar los roles.
Resumen	El caso de uso se inicia cuando el administrador selecciona la opción Gestionar roles para registrar, modificar o eliminar un rol, luego el sistema evalúa la posibilidad de realizar la operación solicitada. El caso de uso termina cuando se realiza a o no la acción seleccionada por el actor.
Responsabilidades	R4
CU asociados	-
Precondiciones	El administrador debe estar autenticado en el sistema. Debe existir al menos un sistema.
Requisitos especiales	-
Descripción	
Interfaz I y II	





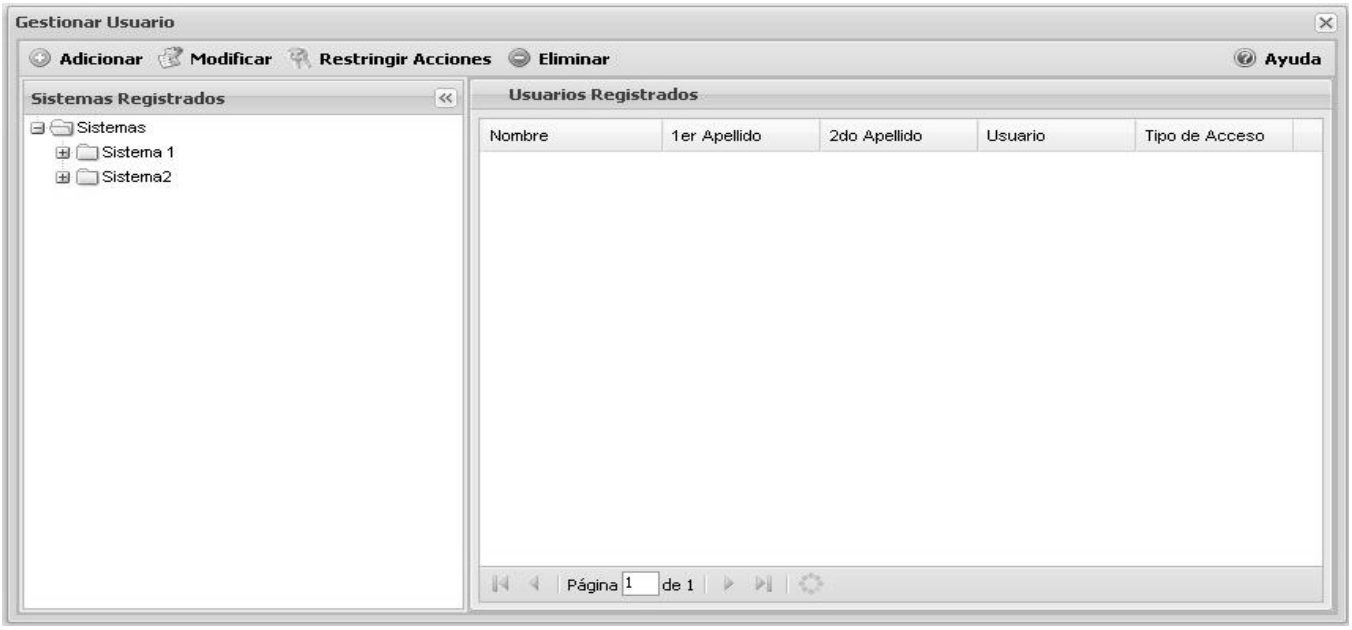
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. El actor selecciona la opción Gestionar Roles del menú principal.	2. El sistema muestra la Interfaz I en la que aparecen todos los sistemas registrados en forma de árbol y una tabla para mostrar los roles de un sistema dado.
3. El actor selecciona un sistema.	4. El sistema activa el botón “Adicionar” y muestra en la tabla los roles correspondientes al sistema seleccionado.

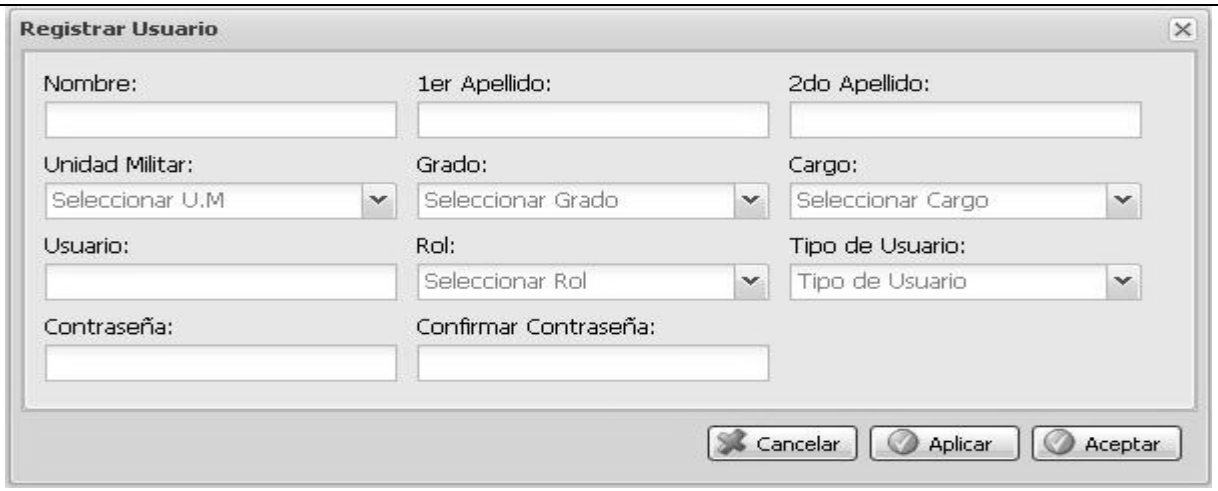
5. El actor decide: • Adicionar (flujo básico). • Modificar (ver sección "Modificar rol"). • Eliminar (ver sección "Eliminar rol").	6. El sistema muestra la interfaz II con los campos nombre, abreviatura, descripción y una tabla con las funcionalidades correspondientes al sistema en el cual se va a adicionar el nuevo rol.
7. El actor llena los campos, selecciona las funcionalidades que le va a asignar al rol y acciona el botón Aceptar.	8. El sistema comprueba la validez de los datos, que no existan campos vacíos y adiciona el nuevo rol.
Flujos Alternos	
8. En caso de que los datos no sean válidos el sistema muestra el mensaje: "Ha introducido datos incorrectos". Aceptar. • En caso de que existan campos vacíos el sistema muestra el mensaje: "Debe llenar todos los campos." Aceptar.	
Sección: Modificar Rol	
Acción del Actor	**Respuesta del Sistema**
1. El actor selecciona el sistema en el que se encuentra el rol que desea modificar.	2. El sistema muestra en la tabla todos los roles del sistema seleccionado.
3. El actor selecciona el rol que desea modificar	4. El sistema activa el botón "Modificar".
5. El actor acciona el botón Modificar.	6. El sistema muestra la interfaz II. En los campos los valores correspondientes al rol seleccionado y u árbol, con las funcionalidades del sistema.
7. El actor modifica los campos de su interés y acciona el botón Aceptar.	8. El sistema comprueba la validez de los datos, que no existan campos vacíos y modifica el rol.
Flujos Alternos	
8. En caso de que los datos no sean válidos el sistema muestra el mensaje: "Ha introducido datos incorrectos". Aceptar. • En caso de que existan campos vacíos el sistema muestra el mensaje: "Debe llenar todos los campos." Aceptar.	
Sección: Eliminar rol	
Acción del actor	**Respuesta del Sistema**
1. El actor selecciona el sistema en el que se encuentra el rol que desea eliminar.	2. El sistema muestra en la tabla todos los roles del sistema seleccionado.
3. El actor selecciona el rol que desea eliminar.	4. El sistema activa el botón "Eliminar"
5. El Actor acciona el botón Eliminar.	6. El sistema muestra la alerta: "Está seguro que desea eliminar el rol

	seleccionado” Aceptar Cancelar
7. El Actor acciona el botón Aceptar.	8. El sistema elimina el rol seleccionado.
Flujos Alternos	
6. En caso de que el actor accione Cancelar se cancela la acción.	
Pos condiciones	Se actualiza toda la información referente a los roles.
Prioridad	Crítica

Descripción del CUS Gestionar usuarios

Caso de Uso	Gestionar usuarios.
Actores	Administrador
Propósito	Permitir Gestionar usuarios.
Resumen	El caso de uso se inicia cuando el administrador selecciona la opción gestionar usuarios del menú principal para registrar, modificar o eliminar un usuario, el sistema evalúa la posibilidad de realizar la operación solicitada. El caso de uso termina cuando se realiza o no la acción seleccionada por el actor.
Responsabilidades	R5
CU asociados	-
Precondiciones	El administrador debe estar autenticado en el sistema, debe haberse registrado un sistema.
Requisitos especiales	-
Descripción	
Interfaz I y II	





Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. El actor selecciona la opción Gestionar Usuario del menú Principal.	2. El sistema muestra la interfaz I en la que aparece un árbol con todos los sistemas y una tabla para mostrar los usuarios.
3. El actor selecciona un sistema.	4. El sistema muestra en la tabla todos los usuarios correspondientes al sistema seleccionado y activa el botón "Adicionar"
5. El actor decide: • Adicionar (flujo Normal). • Modificar (ver sección "Modificar Usuario"). • Eliminar (ver sección "Eliminar Usuario") • Restringir acciones (ver sección "Restringir acciones a usuario").	6. El sistema solicita a estructura y composición la unidad militar, el cargo y el grado. El sistema muestra la interfaz II con los campos nombre, 1er apellido, 2do apellido, unidad militar, grado, cargo, usuario, rol, tipo de usuario, contraseña y confirmar contraseña.
7. El actor llena los campos y acciona el botón Aceptar.	8. El sistema comprueba la validez de los datos, que no existan campos vacíos y adiciona el usuario.
Flujos Alternos	
12. En caso de que los datos no sean válidos el sistema muestra el mensaje: "Ha introducido datos incorrectos". Aceptar. • En caso de que existan campos vacíos el sistema muestra el mensaje: "Debe llenar todos los campos." Aceptar.	
Sección: Modificar Usuario.	
Interfaz III.	

Acción del Actor	Respuesta del Sistema
1. El actor selecciona el sistema en el que está el usuario que desea modificar.	2. El sistema muestra en la tabla todos los usuarios correspondientes al sistema seleccionado.
3. El actor selecciona el usuario que desea modificar.	4. El sistema activa el botón “Modificar”.
5. El actor acciona el botón Modificar.	6. El sistema muestra la interfaz III en la que aparecen todos los datos del usuario
7. El actor modifica los campos de su interés y acciona el botón aceptar.	8. El sistema comprueba la validez de los datos, que no existan campos vacíos y modifica el usuario.

Flujos Alternos.

7. En caso de que el actor accione cancelar se cancela la acción y se cierra la interfaz.

8. En caso de que los datos no sean válidos el sistema muestra el mensaje:

“Ha introducido datos incorrectos”.
Aceptar.

- En caso de que existan campos vacíos el sistema muestra el mensaje:

“Debe llenar todos los campos.”
Aceptar.

Sección: Eliminar Usuario.

Interfaz Eliminar usuario.

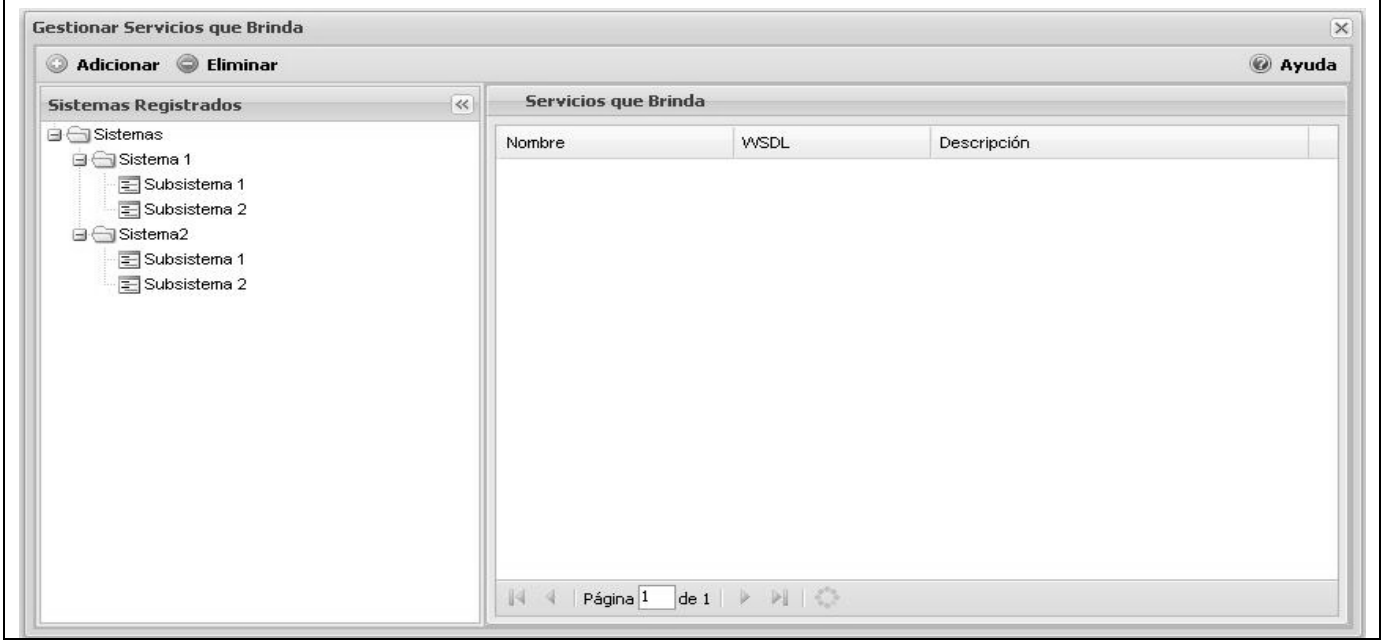
Acción del actor	Respuesta del Sistema
1. El actor selecciona el sistema en el que se encuentra el usuario que desea eliminar.	2. El sistema muestra en la tabla los usuarios correspondientes al sistema seleccionado.
3. El actor selecciona el usuario que desea eliminar.	4. El sistema activa el botón “Eliminar”.
5. El actor acciona el botón Eliminar	6. El sistema muestra una alerta en la que parecen dos mensajes para seleccionar uno: • Eliminar del sistema seleccionado. • Eliminar de todos los sistemas.

	Y los botones Aceptar y Cancelar.
7. El actor decide: • Seleccionar "Eliminar del sistema seleccionado", y accionar el botón Aceptar (flujo normal). • Seleccionar "Eliminar de todos los sistemas" y accionar el botón Aceptar. (ver subsección "Eliminar de todos los sistemas").	8. El sistema elimina el usuario pero solo de ese sistema en el que fue seleccionado.
Flujos Alternos	
6. En caso de que el actor accione el botón Cancelar el sistema cancela la operación.	
Subsección: "Eliminar de todos los sistemas".	
Acción del actor	**Respuesta del Sistema**
1. El actor decide Seleccionar "Eliminar de todos los sistemas" y accionar el botón Aceptar.	2. El sistema elimina el usuario de todos los sistemas de los cuales él es usuario.
Flujos alternos	
Sección: "Restringir acciones a usuario".	
Interfaz IV	
	
Acción del actor	**Respuesta del Sistema**
1. El actor selecciona el sistema en el que se encuentra el usuario al que le quiere restringir acciones.	2. El sistema muestra en la tabla todos los usuarios correspondientes al sistema seleccionado.
3. El actor selecciona el usuario al que le quiere restringir acciones.	4. El sistema activa el botón "Restringir Acciones".
5. El actor acciona el botón Restringir acciones	6. El sistema muestra la interfaz IV en la que aparece una tabla con todas las acciones a las que el usuario tiene acceso.

7. El actor selecciona las acciones a las cuales el usuario no va a tener acceso y acciona el botón Aceptar	8. El sistema actualiza las acciones a las que el usuario tiene acceso.
Flujos alternos	
Postcondiciones	Se actualiza toda la información referente a los usuarios.
Prioridad	Crítica

Gestionar servicios que brinda un sistema

Caso de Uso	Gestionar Servicios que brinda un sistema
Actores	Administrador
Propósito	Permitir Gestionar los servicios que brinda un sistema.
Resumen	El caso de uso se inicia cuando el administrador selecciona la opción gestionar servicios que brinda luego el sistema evalúa la posibilidad de realizar la operación solicitada. El caso de uso termina cuando se realiza o no la acción seleccionada por el actor.
Responsabilidades	R15
CU asociados	
Precondiciones	El administrador debe estar autenticado en el sistema, debe haberse registrado al menos un sistema.
Requisitos especiales	-
Descripción	
Interfaz I y II	



Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. El actor selecciona la opción Gestionar servicios que brinda del menú principal.	2. El sistema muestra la interfaz I en la que aparecen un árbol con todos los sistemas y una tabla para mostrar todos los servicios que brinda un sistema dado.
3. El actor selecciona un sistema.	4. El sistema activa el botón "Adicionar" y se muestran en la tabla los servicios correspondientes al sistema seleccionado.
5. El actor decide: • Adicionar (flujo normal) • Modificar (ver sección "Modificar servicio"). • Eliminar (ver sección "Eliminar servicio").	6. El sistema muestra la interfaz II con los campos nombre, wsdl y descripción del servicio.
7. El actor llena los campos y acciona el botón Aceptar.	8. El sistema comprueba que los datos son válidos, que no existan campos vacíos y adiciona el nuevo servicio.

Flujos alternos.

7. En caso de que el actor accione el botón Cancelar se cancela la operación.

8. En caso de que los datos no sean válidos el sistema muestra el mensaje:

"Ha introducido datos incorrectos".
Aceptar.

- En caso de que existan campos vacíos el sistema muestra el mensaje:

"Debe llenar todos los campos."
Aceptar.

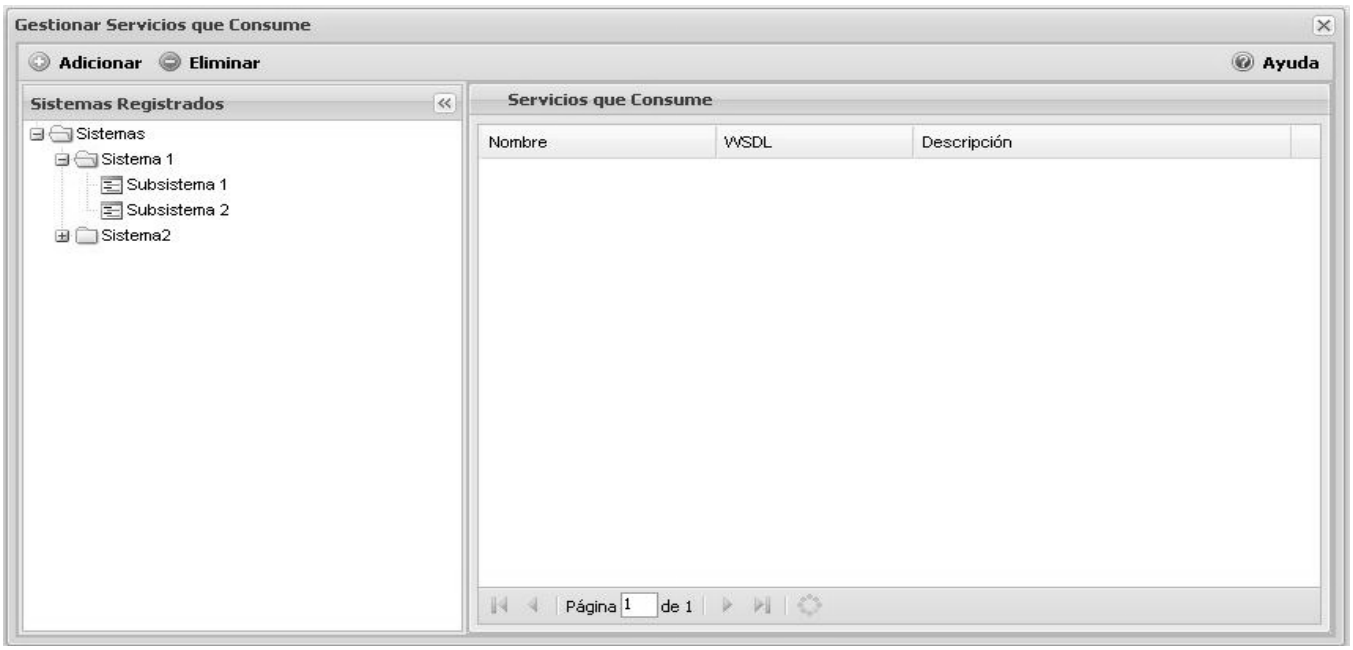
Sección: "Modificar servicio"	
Interfaz modificar servicio	
Acción del actor	**Respuesta del sistema**
1. El actor selecciona el sistema en el que se	2. El sistema muestra en la tabla todos los

encuentra el servicio que desea modificar.	servicios correspondientes al sistema seleccionado.
3. El actor selecciona el servicio que de desea modificar.	4. El sistema activa los botones “Modificar”, y “Eliminar”.
5. El actor acciona el botón Modificar.	6. El sistema muestra la interfaz II con los campos nombre, wsdl y descripción del servicio.
7. El actor modifica los cambios de su interés y acciona el botón Aceptar.	8. El sistema verifica que los datos son válidos, modifica y registra los datos del servicio.
Flujos alternos	
7. En caso de que el actor accione el botón cancelar, se termina la acción. 8. En caso de que los datos no sean válidos el sistema muestra el mensaje: “Ha introducido datos incorrectos”. Aceptar.	
Sección: “Eliminar servicio”	
Acción del actor	**Respuesta del sistema**
1. El actor selecciona el sistema en el que se encuentra el servicio que desea eliminar.	2. El sistema muestra en la tabla todos los servicios que brinda el sistema seleccionado.
3. El actor selecciona el servicio que desea eliminar.	4. El sistema Activa los botones “Eliminar” y “Modificar”
5. El Actor acciona el botón Eliminar.	6. El sistema muestra la alerta: “Está seguro que desea eliminar el servicio seleccionado” Aceptar Cancelar
7. El actor acciona el botón Aceptar.	8. El sistema elimina el servicio seleccionado.
Flujos alternos.	
7. En caso de que el actor accione el botón Cancelar, se termina la acción.	
Postcondiciones	Se actualiza toda la información referente a los servicios que brinda un sistema.
Prioridad	Crítica

Descripción del CUS Gestionar servicios que consume un sistema.

Caso de Uso	Gestionar servicios que consume.
Actores	Administrador
Propósito	Permitir Gestionar los servicios que consume un sistema.
Resumen	El caso de uso se inicia cuando el administrador selecciona la opción gestionar servicios que consume luego el sistema evalúa la posibilidad de realizar la operación solicitada. El caso de uso termina cuando se realiza o no la acción seleccionada por el actor.
Responsabilidades	R16

CU asociados	
Precondiciones	El administrador debe estar autenticado en el sistema, debe haberse registrado al menos un sistema.
Requisitos especiales	-
Descripción	
Interfaz I y II	



Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. El actor selecciona la opción Gestionar servicios que consume del menú principal.	2. El sistema muestra la interfaz I en la que aparecen un árbol con todos los sistemas y una tabla para mostrar todos los servicios que consume un sistema dado.
3. El actor selecciona un sistema.	4. El sistema activa el botón Adicionar y se muestran en la tabla los servicios que consume el sistema seleccionado.
5. El actor decide: • Adicionar (flujo normal) • Eliminar (ver sección "Eliminar servicio que consume").	6. El sistema muestra la interfaz II en la que aparecen un árbol con todos los sistemas y una tabla para mostrar los servicios que brinda un sistema dado.
7. El actor selecciona el sistema en el que se encuentra el servicio que desea registrar.	8. El sistema muestra en la tabla todos los servicios que brinda el sistema seleccionado.
9. El actor selecciona el servicio que desea adicionar y acciona el botón aceptar.	10. El sistema adiciona el nuevo servicio a consumir en el sistema previamente seleccionado.
Flujos alternos.	
9. En caso de que el actor accione el botón Cancelar se cancela la operación.	
Sección: "Eliminar servicio que consume"	
Interfaz Eliminar servicio que brinda	
Acción del actor	**Respuesta del sistema**
1. El actor selecciona el sistema en el que se encuentra el servicio que desea eliminar.	2. El sistema muestra en la tabla todos los servicios que brinda el sistema seleccionado.
3. El actor selecciona el servicio que desea eliminar.	4. El sistema activa el botón "Eliminar".
5. El actor acciona el botón Eliminar.	6. El sistema muestra la alerta: "Está seguro que desea eliminar el servicio seleccionado" Aceptar Cancelar
7. El actor acciona el botón Aceptar.	8. El sistema elimina el servicio seleccionado.
Flujos alternos.	
7. En caso de que el actor accione el botón Cancelar, se termina la acción.	
Postcondiciones	Se actualiza toda la información referente a los servicios que consume un sistema.
Prioridad	Crítica

Descripción del CUS Autenticar usuario.

Caso de Uso	Autenticar usuario.
Actores	Sistema externo.
Propósito	Permitir acceso de un usuario al marco de trabajo.
Resumen	El caso de uso se inicia cuando un usuario desea acceder al marco de trabajo. A nuestro sistema responsable de la seguridad le llega el usuario y la contraseña. El caso de uso termina cuando se permite o deniega el acceso.
Responsabilidades	R6
CU asociados	
Precondiciones	
Requisitos especiales	-

Descripción	
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. El actor solicita usar el servicio de autenticación que brinda el sistema de seguridad, pasando los parámetros usuario y contraseña.	2. El sistema verifica si existe un servidor LDAP. • No existe (ver sección "Autenticar Sin LDAP"). • Existe (flujo normal). 3. El sistema envía los parámetros usuario y contraseña al servidor LDAP y espera la respuesta. 4. El sistema crea un certificado digital para el usuario. 5. El sistema envía un XML con el certificado.
Flujos Alternos	
4. Si la respuesta del servidor LDAP es fallida el sistema envía un XML con mensaje de error. En caso de que no concuerden los parámetros el sistema envía un XML con mensaje de error. En caso de que la contraseña haya caducado el sistema envía un XML con mensaje de información.	
Sección: "Autenticar Sin LDAP"	
Acción del actor.	**Respuesta del sistema.**
1. El actor solicita usar el servicio de autenticación que brinda el sistema de seguridad, pasando los parámetros usuario y contraseña.	2. El sistema verifica que el usuario existe. 3. El sistema verifica que la contraseña sea válida. 4. El sistema verifica si la contraseña no ha caducado 5. El sistema crea un certificado digital para

	el usuario. 6. El sistema envía un XML con el certificado.
Flujos alternos.	
2. En caso de que el usuario no exista el sistema envía un XML con mensaje de error. 3. En caso de que la contraseña no sea igual el sistema envía un XML con mensaje de error. 4. En caso de que la contraseña haya caducado el sistema envía un XML con mensaje de información.	
Postcondiciones	Se autentica el usuario en el marco de trabajo.
Prioridad	Crítica

Descripción del CUS Autorizar acceso a sistemas.

Caso de Uso	Autorizar acceso a sistemas.
Actores	Sistema externo.
Propósito	Devolver todos los sistemas a los que un usuario tiene acceso.
Resumen	El caso de uso se inicia cuando se solicitan los sistemas a lo que tiene acceso un usuario previamente autenticado. El caso de uso termina cuando se devuelven o no los sistemas.
Responsabilidades	R7
CU asociados	
Precondiciones	El usuario tiene que estar autenticado.
Requisitos especiales	-
Descripción	
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. El actor solicita usar el servicio de autorizar acceso a sistemas enviando como parámetro el certificado de usuario.	2. El sistema comprueba que el certificado es válido. 3. El sistema verifica a que usuario pertenece ese certificado. 4. El sistema encuentra todos los sistemas a los que el usuario tiene acceso 5. El sistema devuelve un XML con todos los sistemas.
Flujos alternos.	
2. En caso de el que el certificado no sea válido el sistema envía un XML de error.	
Postcondiciones	Se devuelven los sistemas a los que tiene acceso un usuario

Prioridad	Crítica

Descripción del CUS Cargar menú.

Caso de Uso	Cargar Menú
Actores	Sistema externo.
Propósito	Devolver el menú de un usuario en un sistema.
Resumen	El caso de uso se inicia cuando se solicita el menú de un usuario en un sistema. El caso de uso termina cuando se devuelve el menú del usuario.
Responsabilidades	R8
CU asociados	
Precondiciones	El usuario tiene que estar autenticado.
Requisitos especiales	-

Descripción	
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. El actor solicita el servicio de cargar menú pasando los parámetros, sistema y certificado.	2. El sistema comprueba que el certificado es válido. 3. El sistema verifica a que usuario corresponde el certificado. 4. El sistema verifica el rol que le corresponde al usuario en el sistema enviado. 5. El sistema devuelve el menú correspondiente al usuario en ese sistema según su rol en forma de XML.
Flujos alternos.	
2. En caso de el que el certificado no sea válido el sistema envía un XML de error.	

Postcondiciones	Se le muestra el menú al usuario.
Prioridad	Crítica

Descripción del CUS Autorizar acceso a funcionalidad.

Caso de Uso	Autorizar acceso a funcionalidad.
Actores	Sistema externo.
Propósito	Permitir acceso de un usuario a una funcionalidad de un sistema externo registrado.
Resumen	El caso de uso se inicia cuando un usuario desea acceder a una funcionalidad de un sistema externo. El sistema externo le pide a nuestro sistema responsable

	de la seguridad que verifique si el usuario tiene permiso de acceso a esa funcionalidad. El caso de uso termina cuando se permite o deniega el acceso.
Responsabilidades	R9
CU asociados	
Precondiciones	El usuario debe estar autenticado.
Requisitos especiales	-

Descripción	
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. El actor solicita el servicio de verificación de acceso a funcionalidad que brinda nuestro sistema enviando los parámetros sistema, certificado y funcionalidad a la que se quiere acceder.	2. El sistema verifica que el certificado sea válido. 3. El sistema con los parámetros enviados verifica si la funcionalidad a la que quiere acceder el usuario se encuentra dentro de las funcionalidades a las que tiene acceso según el rol que le corresponde al usuario en el sistema enviado. 4. EL sistema envía un XML autorizando a usar la funcionalidad.
Flujos alternos.	
3. En caso de que el certificado no sea válido el sistema envía un XML de error. 4. En caso de que no tenga acceso a la funcionalidad el sistema envía un XML de error.	

Postcondiciones	Permitir acceso a funcionalidad a solicitud del sistema externo.
Prioridad	Crítica

Descripción del CUS Autorizar acceso a acción.

Caso de Uso	Autorizar acceso a Acción.
Actores	Sistema externo.
Propósito	Permitir acceso de un usuario a una acción de un sistema externo registrado.
Resumen	El caso de uso se inicia cuando un usuario desea acceder a una acción de un sistema externo. El sistema externo le pide a nuestro sistema responsable de la seguridad que verifique si el usuario tiene permiso de acceso a esa acción. El caso de uso termina cuando se permite o deniega el acceso.
Responsabilidades	R10
CU asociados	
Precondiciones	El usuario tiene que estar autenticado..
Requisitos	-

especiales	
Descripción	
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. El actor solicita el servicio de verificación de acceso a una acción que brinda nuestro sistema enviando los parámetros sistema, funcionalidad, certificado y acción a la que quiere acceder el usuario.	2. El sistema verifica que el certificado sea válido. 3. El sistema con los parámetros enviados verifica si la acción a la que quiere acceder el usuario esta comprendida dentro de las acciones que el puede hacer en ese sistema. 4. EL sistema envía un XML autorizando a usar la acción.
Flujos alternos.	
3. En caso de que el certificado no sea válido el sistema envía un XML de error. 4. En caso de que no tenga acceso a la acción el sistema envía un XML de error.	
Postcondiciones	El usuario es autorizado a acceder a la acción solicitada a solicitud del sistema externo.
Prioridad	Crítica

Descripción del CUS autorizar acceso a servicio.

Caso de Uso	Autorizar acceso a servicio.
Actores	Sistema externo.
Propósito	Permitir acceso de una aplicación a un servicio publicado de otra aplicación.
Resumen	El caso de uso se inicia cuando un sistema desea consumir un servicio de otro sistema registrado. El sistema externo le pide a nuestro sistema responsable de la seguridad que verifique si el si el sistema tiene permiso de acceso a ese servicio. El caso de uso termina cuando se permite o deniega el acceso.
Responsabilidades	R11
CU asociados	
Precondiciones	El usuario tiene que estar autenticado.
Requisitos especiales	-
Descripción	
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. El actor solicita el servicio de autorización de servicios enviando los parámetros sistema consumidor, sistema proveedor, servicio que se desea consumir, y certificado del usuario.	2. El sistema verifica que el certificado sea válido. 3. El sistema con los parámetros enviados

	verifica si el sistema consumidor tiene acceso al servicio solicitado. 4. EL sistema envía un XML autorizando a usar el servicio solicitado.
Flujos alternos.	
3. En caso de que el certificado no sea válido el sistema envía un XML de error. 4. En caso de que no tenga acceso al servicio el sistema envía un XML de error.	
Postcondiciones	El usuario es autorizado a usar el servicio solicitado a solicitud del sistema externo.
Prioridad	Crítica

Descripción del CUS Editar perfil.

Caso de Uso	Editar perfil.
Actores	Sistema externo.
Propósito	Permitir que un usuario se edite su perfil.
Resumen	El caso de uso se inicia cuando un usuario desea editar su perfil. El sistema externo solicita a nuestro sistema responsable de la seguridad el servicio editar perfil. El caso de uso termina cuando se ejecuta o se deniega la acción.
Responsabilidades	R12
CU asociados	
Precondiciones	El usuario tiene que estar autenticado.
Requisitos especiales	-
Descripción	
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. El actor solicita el servicio de editar perfil enviando los parámetros sistema, certificado y estilo.	2. El sistema verifica que el certificado sea válido. 3. El sistema con los parámetros enviados actualiza el perfil del usuario.
Flujos alternos.	
3. En caso de que el certificado no sea válido el sistema envía un XML de error.	
Postcondiciones	Se actualiza el perfil del usuario que estaba autenticado a solicitud del sistema externo.
Prioridad	Crítica

Descripción del CUS Cambiar contraseña.

Caso de Uso	Cambiar contraseña.
Actores	Sistema externo.
Propósito	Permitir que un usuario se cambie su contraseña.
Resumen	El caso de uso se inicia cuando un usuario desea cambiarse su contraseña. El sistema externo solicita a nuestro sistema responsable de la seguridad el servicio de cambiar contraseña. El caso de uso termina cuando se ejecuta o se deniega la acción.
Responsabilidades	R13
CU asociados	
Precondiciones	El usuario tiene que estar autenticado.
Requisitos especiales	-

Descripción	
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. El actor solicita el servicio de cambiar contraseña enviando los parámetros sistema, certificado, contraseña anterior, contraseña nueva y confirmación de contraseña nueva.	2. El sistema verifica que el certificado sea válido. 3. El sistema verifica que la contraseña cumple con los requisitos establecidos por el centro. 4. El sistema verifica que la Contraseña no es igual a ninguna de las anteriores. 5. El sistema con los parámetros enviados cambia la contraseña del usuario. 6. EL sistema envía un XML con el mensaje: “Su contraseña se ha cambiado””.
Flujos alternos.	
3. En caso de que el certificado no sea válido el sistema envía un XML de error. 4. En caso de que la contraseña no cumpla con los requisitos del centro de datos el sistema envía un XML con mensaje de “sus contraseña no cumple con las restricciones de seguridad”. 5. En caso de el que la contraseña sea igual a alguna de las anteriores del usuario el sistema envía un XML con el mensaje: “la contraseña debe ser diferente de las anteriores.	

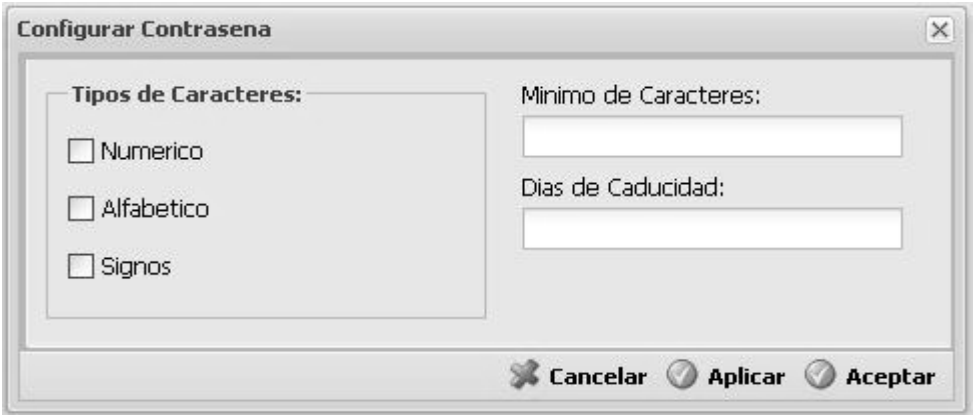
Postcondiciones	Se actualiza la contraseña a petición del sistema externo.
Prioridad	Crítica

Descripción del CUS Configurar contraseña.

Caso de Uso	Configurar contraseña.
Actores	Administrador.
Propósito	Permitir configurar las contraseñas
Resumen	El caso de uso se inicia cuando el administrador desea configurar las contraseñas. El caso de uso termina cuando se ejecuta o se deniega la acción.
Responsabilidades	R14
CU asociados	
Precondiciones	El administrador debe estar autenticado en el sistema.
Requisitos especiales	-

Descripción

Interfaz I



Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1. El actor selecciona la opción Configurar contraseñas del menú principal.	2. El sistema muestra la interfaz I en la que aparecen los campos mínimo de caracteres, días de caducidad, numérico, alfabético y signos.
3. El actor selecciona y llena los campos mostrados y acciona el botón Aceptar.	4. El sistema verifica que los datos son correctos, que no existen campos vacíos y guarda la nueva configuración de contraseñas.

Flujos alternos.

3. En caso de que los datos no sean válidos el sistema muestra el mensaje:

"Ha introducido datos incorrectos".
Aceptar.

• En caso de que existan campos vacíos el sistema muestra el mensaje: "Debe llenar todos los campos." Aceptar.	
Postcondiciones	Se guarda la nueva configuración de las contraseñas.
Prioridad	Crítica

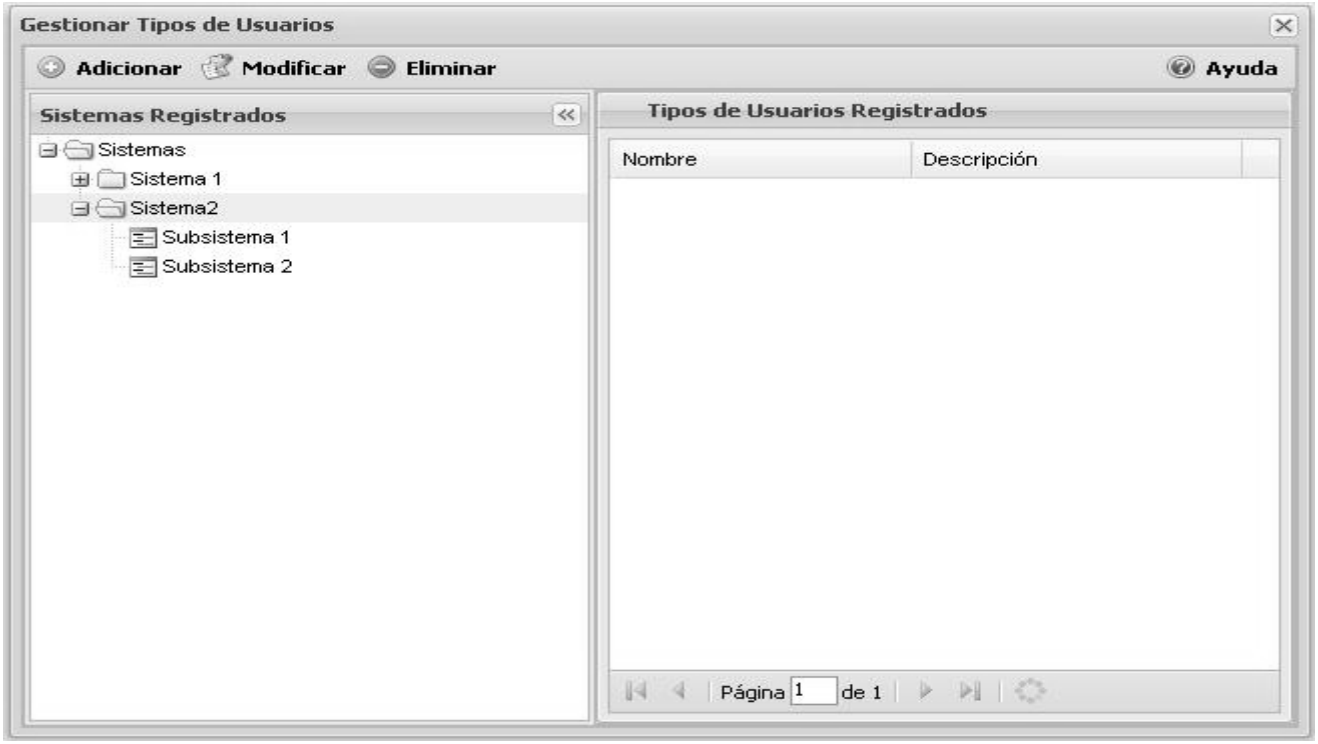
Descripción del CUS Cerrar sesión de usuario.

Caso de Uso	Cerrar sesión de usuario.
Actores	Sistema externo.
Propósito	Cerrar sesión de un usuario en un sistema registrado.
Resumen	El caso de uso se inicia cuando un usuario previamente autenticado desea cerrar la sesión. El caso de uso termina cuando se ejecuta o se deniega la acción.
Responsabilidades	R17
CU asociados	
Precondiciones	El usuario tiene que estar autenticado.
Requisitos especiales	-
Descripción	
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. El actor solicita cerrar la sesión de un usuario previamente autenticado enviando los parámetros: certificado, sistema.	2. El sistema verifica que el certificado sea válido y que corresponda a algún usuario autenticado. 3. El sistema invalida el certificado. 4. El sistema cierra la sesión de usuario y envía un XML con el mensaje: "Usted ha cerrado su sesión".
Flujos alternos.	
3. En caso de el que el certificado no sea válido el sistema envía un XML de error.	
Postcondiciones	Se ha cerrado la sesión de usuario que estaba autenticado a solicitud del sistema.
Prioridad	Crítica

Descripción del CUS Gestionar tipos de usuarios.

Caso de Uso	Gestionar Tipos de Usuario
Actores	Administrador.

Propósito	Permitir gestionar tipos de usuario para establecer grupos de usuarios y controlar que se acceda a la base de datos con los permisos correspondientes a los roles de usuario.
Resumen	El caso de uso se inicia cuando el administrador desea gestionar tipos de usuario. El caso de uso termina cuando se realiza o no la acción del administrador.
Responsabilidades	R18
CU asociados	
Precondiciones	El administrador debe estar autenticado en el sistema, debe haberse registrado al menos un sistema.
Requisitos especiales	-
Descripción	
Interfaz I y II	



Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. El actor selecciona la opción gestionar tipos de usuario del menú principal.	2. El sistema muestra la interfaz I en la que aparecen todos los sistemas en forma de árbol, una tabla para mostrar los tipos de usuario y los botones “Adicionar”, “Modificar” y “Eliminar” desactivados
3. El actor selecciona el sistema en que desea gestionar tipos de usuarios	4. El sistema activa el botón “Adicionar” y muestra todos los tipos de usuario en la tabla.
5. El actor decide: • Adicionar (Flujo normal). • Modificar (ver sección Modificar tipos de usuario). • Eliminar (ver sección Eliminar tipos de usuario).	6. El sistema muestra la interfaz II en la que aparecen dos campos uno para el nombre del tipo de usuario, otro para la descripción y los botones Aceptar y Cancelar
7. El administrador entra los datos y presiona el botón Aceptar.	8. El sistema registra el tipo de usuario.
Flujos alternos.	
7. En caso de que el actor accione el botón Cancelar se cancela la operación.	
Sección: “Modificar tipo de usuario”.	
Acción del actor	**Respuesta del sistema**
1. El actor selecciona el sistema en el que desea gestionar tipos de usuario.	2. El sistema muestra en la tabla todos los tipos de usuarios correspondientes al sistema seleccionado.
3. El actor selecciona el tipo de usuario que desea modificar.	4. El sistema Activa los botones “Modificar” y “Eliminar”.
5. El actor presiona el botón Modificar.	6. El sistema muestra la interfaz II con todos los datos del tipo de usuario seleccionado.
7. El actor modifica los campos de su interés y presiona el botón Aceptar.	8. El sistema comprueba la validez de los datos y modifica el tipo de usuario seleccionado.

Flujos alternos	
7. En caso de que el actor presione Cancelar se cancela la acción. 8. En caso de que los datos no sean válidos el sistema muestra el mensaje: "Ha introducido datos incorrectos". Aceptar.	
Sección: Eliminar Tipos de usuario.	
Acción del actor	**Respuesta del sistema**
1. El actor selecciona el sistema en el que se encuentra el tipo de usuario que desea eliminar.	2. El sistema muestra en la tabla los tipos de usuario correspondientes al sistema seleccionado.
3. El actor selecciona el tipo de usuario que desea eliminar.	4. El sistema activa los botones "Modificar" y "Eliminar".
5. El actor presiona el botón Eliminar	6. El sistema muestra la alerta: "Está seguro que desea eliminar el tipo de usuario seleccionado" Aceptar Cancelar
7. El actor presiona el botón Aceptar	8. El sistema elimina el tipo de usuario.
Flujos alternos	
7. En caso de que el actor accione el botón Cancelar, se termina la acción.	
Postcondiciones	Se actualiza toda la información referente a los tipos de usuarios.
Prioridad	Crítica

2.6. Conclusiones.

En este capítulo se realizó todo lo correspondiente a los flujos de negocio y levantamiento de requisitos. En el mismo están los artefactos que generan estos flujos, como son: modelo de dominio, catálogo de requisitos, modelo de casos de uso, casos de uso del sistema con sus respectivas descripciones y prototipos de interfaces de usuario.

CAPÍTULO 3: ANÁLISIS Y DISEÑO.

3.1. Introducción.

Este capítulo muestra todo lo referente a los flujos análisis y diseño del sistema. En el se encuentran todos los artefactos generados como son Diagrama de clases del análisis, diagramas de clases del diseño y los diagramas de interacción.

3.2. Análisis.

3.2.1. Diagramas de clases del análisis.

A continuación se muestran los diagramas de clases del análisis de los casos de uso Gestionar sistemas, Gestionar funcionalidades, Gestionar acciones, Gestionar servicios que brinda un sistema y configurar contraseña. El resto de los diagramas se encuentran en el **Anexo 1**

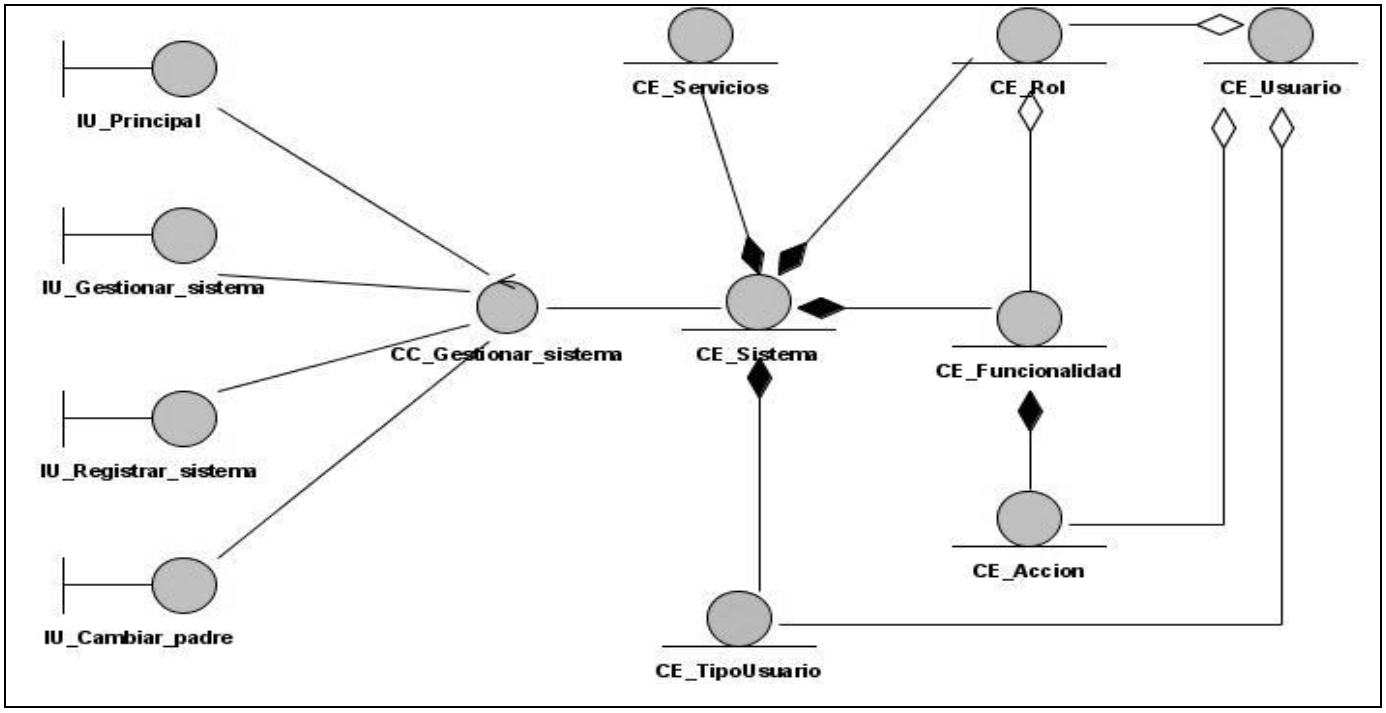


Figura 3.1 Diagrama de clases del análisis CU Gestionar sistemas.

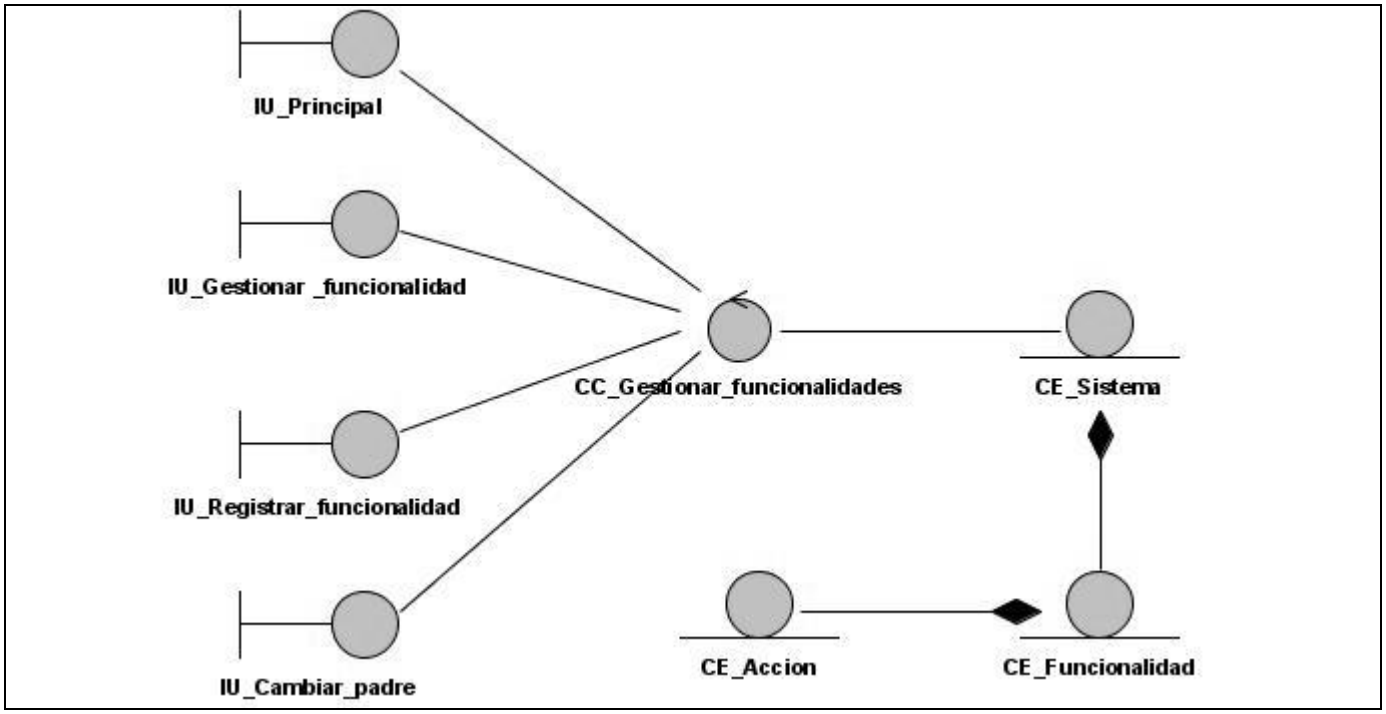


Figura 3.2 Diagrama de clases del análisis CU Gestionar funcionalidades.

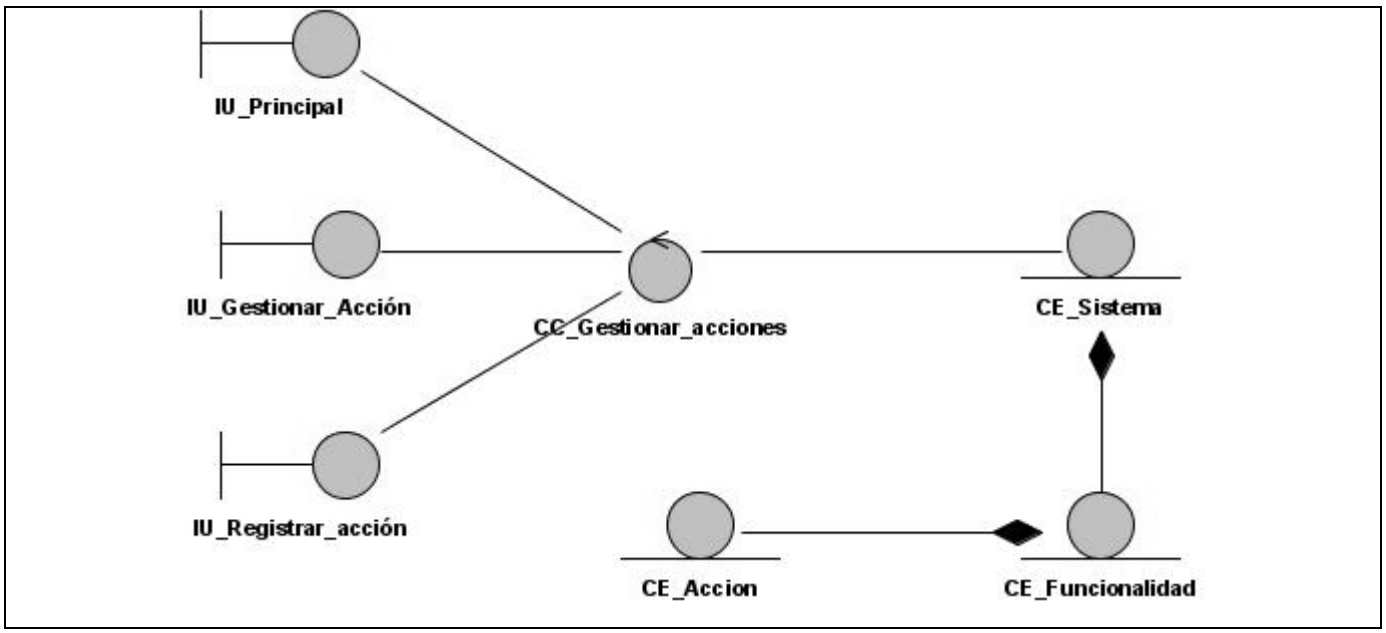


Figura 3.3 Diagrama de clases del análisis CU Gestionar acciones.

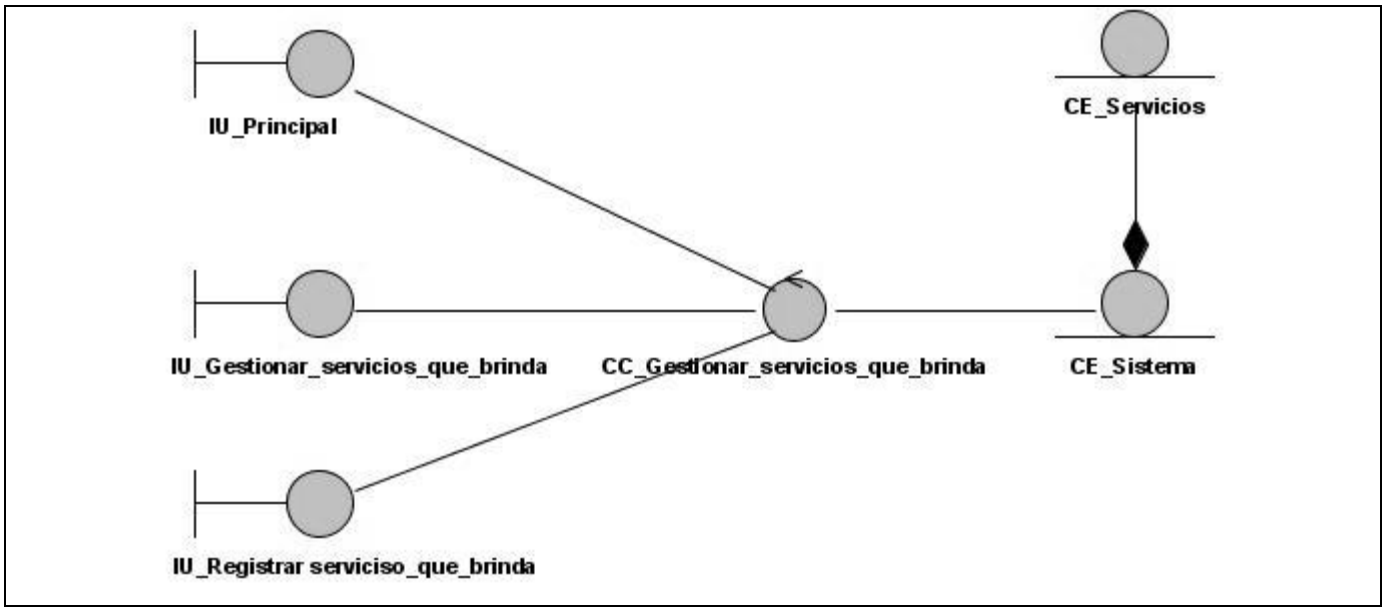


Figura 3.4 Diagrama de clases del análisis CU Gestionar servicio que brinda.

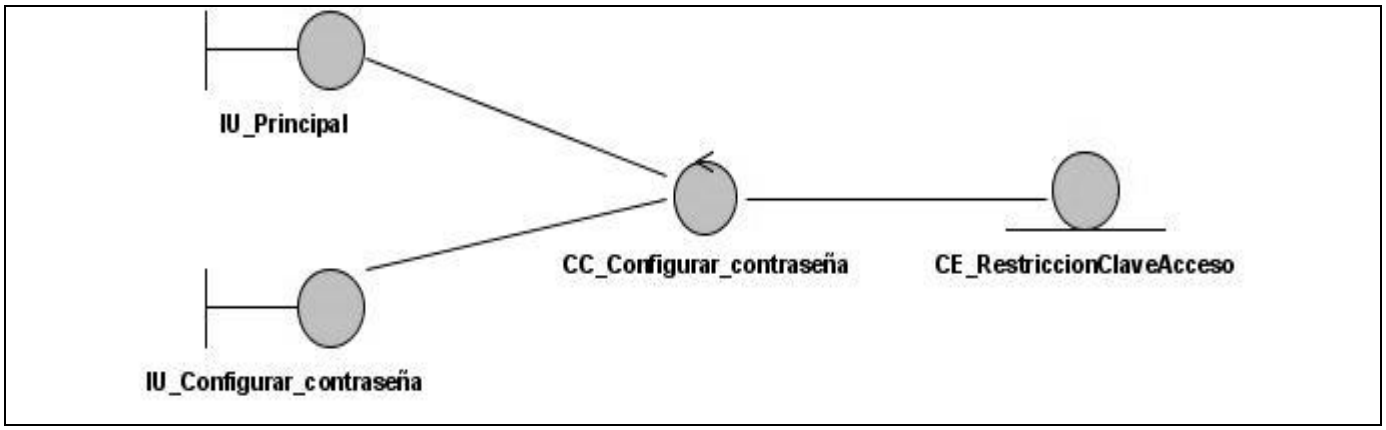


Figura 3.5 Diagrama de clases del análisis CU Configurar contraseña.

3.3. Diseño

El modelo de diseño está muy cercano al de implementación, lo que es natural para guardar y mantener el modelo de diseño a través del ciclo de vida completo del software. En el diseño se modela el sistema y se encuentra su forma para que soporte todos los requisitos, incluyendo los no funcionales y las restricciones que se le suponen. Una entrada esencial en el diseño es el resultado del análisis, proporciona una comprensión detallada de los requisitos. Además impone una estructura del sistema que debemos esforzarnos para conservar lo más fielmente posible cuando demos forma al sistema.

3.3.1. Mecanismo de diseño

Los Mecanismos de diseño son una forma de especificación al Equipo de Desarrollo en los que se esclarecen como llevar a vías de hecho un determinado aspecto de diseño. Estos evitan que se ponga lo mismo en cada uno de los diagramas, ahorrando tiempo y facilitándoles el trabajo a los diseñadores. A continuación se muestra el mecanismo de diseño para el acceso a datos que se definió en el centro UCID.

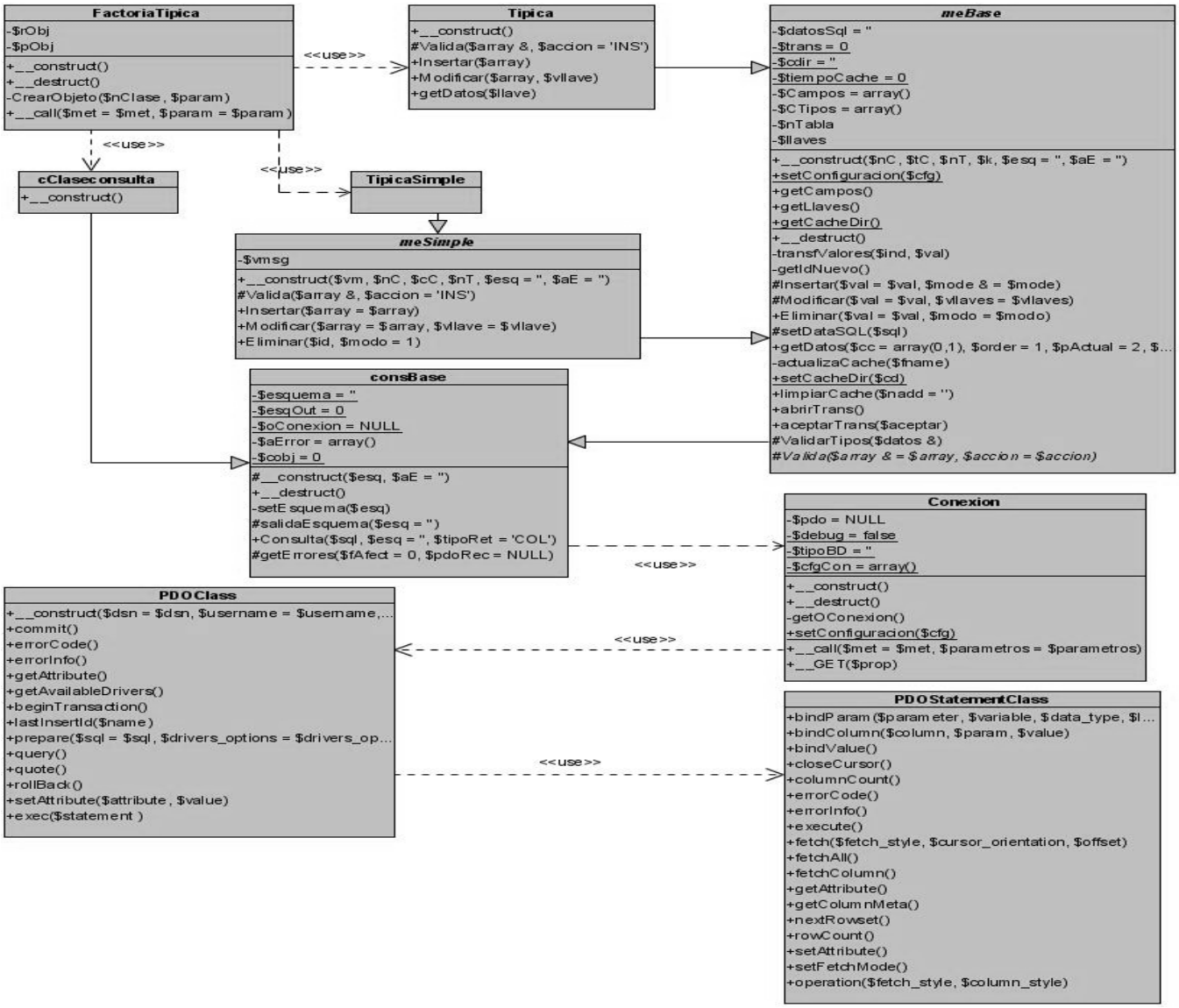


Figura 3.6 Mecanismo para el acceso a datos.

Para un mayor entendimiento del funcionamiento las clases que intervienen se explican a continuación las responsabilidades de algunas de ellas.

Factoría Típica: Clase que implementa la interfaz del modelo de persistencia con el resto de los subsistemas. A través de esta clase se crean y se manipulan los objetos de las típicas simples, los nomencladores y las demás típicas. Es una puerta entre la capa de Acceso a Datos y la capa de Lógica de Negocio.

Típica: Es una clase que representa a las clases típicas en general de la aplicación. Existe una típica para cada entidad de la base de datos. Sus métodos consisten en las operaciones básicas que se realizan sobre estas tablas (insertar, eliminar y modificar) Hereda de la clase abstracta meBase.

TípicaSimple: Es una clase que representa a las clases típicas para nomencladores simples. Sus métodos consisten en las operaciones básicas que se realizan sobre estas tablas (insertar, eliminar, modificar). Hereda de la clase abstracta meSimple.

CClaseconsulta: Es una clase que representa a las clases consultas en general de la aplicación. Existe una clase consulta para cada entidad de la base de datos. Hereda de la clase abstracta consBase.

meSimple: Clase abstracta, base para la implementación de las típicas que responderán a los nomencladores simples del modelo de persistencia dado. Redefine las operaciones básicas con la funcionalidad de Validación dada.

meBase: Clase abstracta, base para el resto de las que implementen funcionalidades para el trabajo con las entidades del sistema a implementar. Implementa las operaciones básicas que pudieran realizarse a una entidad (insertar, eliminar, modificar). Hereda de consBase la operación de Consulta.

consBase: Esta clase es la base en toda la jerarquía de Acceso a Datos y es empleada para aportar contenido dinámico a las plantillas. Encapsula el objeto conexión. Implementa la operación de Consulta.

Conexión: Esta clase es la encargada de establecer la conexión con el servidor de la BD a través de un objeto PDO de la librería de PHP.

PDO: Nos brinda una capa de abstracción de acceso a los datos, lo que quiere decir que independientemente del tipo de gestor que estemos empleando se emplean las mismas funciones para ejecutar consultas y acceder a los datos.

3.3.2. Diagramas de clases del diseño.

Para una mayor comprensión y para facilitar el trabajo se realizó un diagrama de clases del diseño genérico, el mismo representa las clases del diseño fundamentales que participan en la mayoría de los casos de uso del sistema.

Este diagrama de clases genérico, se realizó con el objetivo de minimizar el trabajo y hacerlo de forma más eficiente. También se desarrolló un diagrama genérico donde se muestra la gestión del portal y donde se puede ilustrar la relación del mismo con el sistema de seguridad (ver figura 3.8).Para la definición de las funciones se utilizó como estándar de codificación la notación CamellCasing.

A continuación se muestran los diagramas de clases del diseño de los casos de uso Gestionar sistemas, Gestionar funcionalidades, Gestionar acciones, Gestionar servicios que brinda un sistema, configurar contraseña y los diagramas genéricos descritos anteriormente. El resto de los diagramas se encuentran en los anexos (Ver el **Anexo 2**).

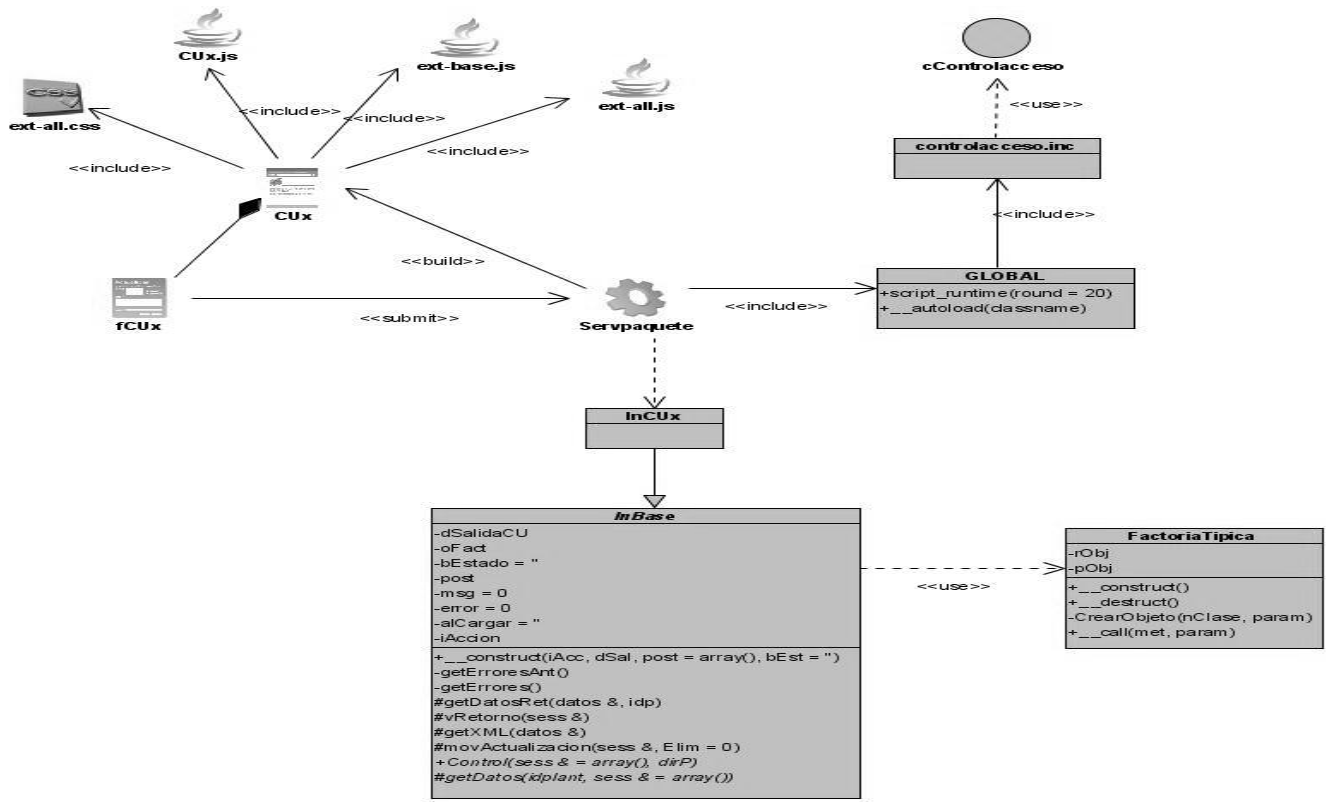


Figura 3.7 Diagrama de clases del diseño genérico.

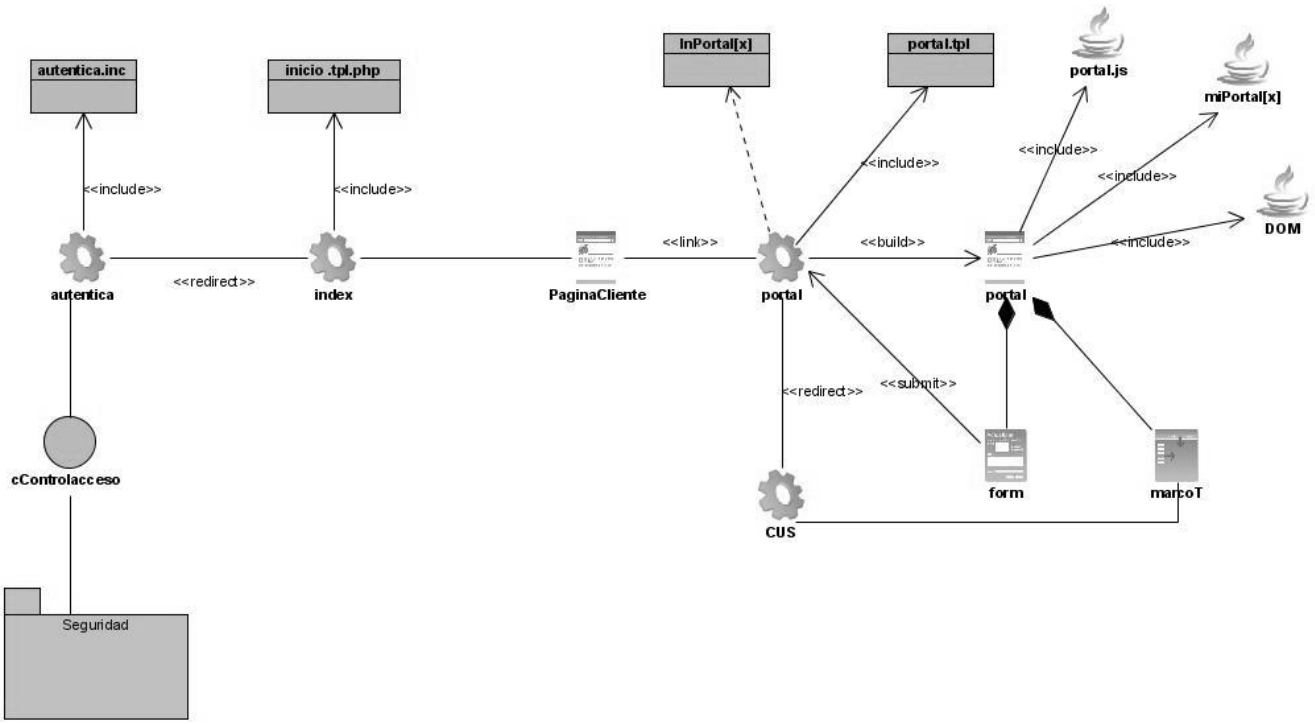


Figura 3.8 Diagrama de clases del diseño genérico para la gestión del portal.

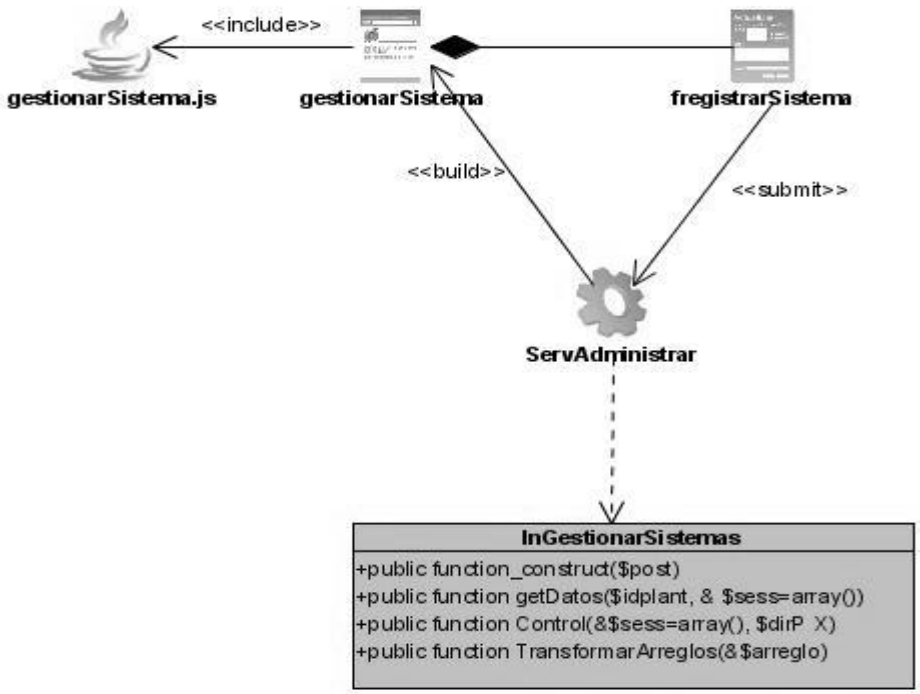


Figura 3.9 Diagrama de clases del diseño CU Gestionar sistema.

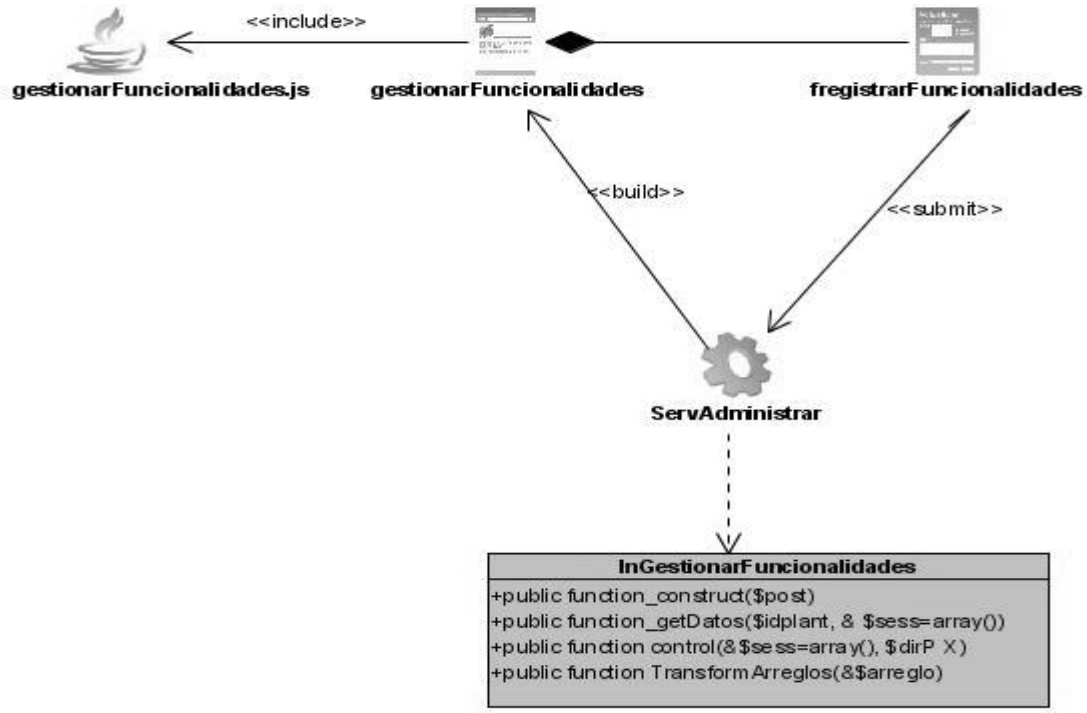


Figura 3.10 Diagrama de clases del diseño CU Gestionar funcionalidades.

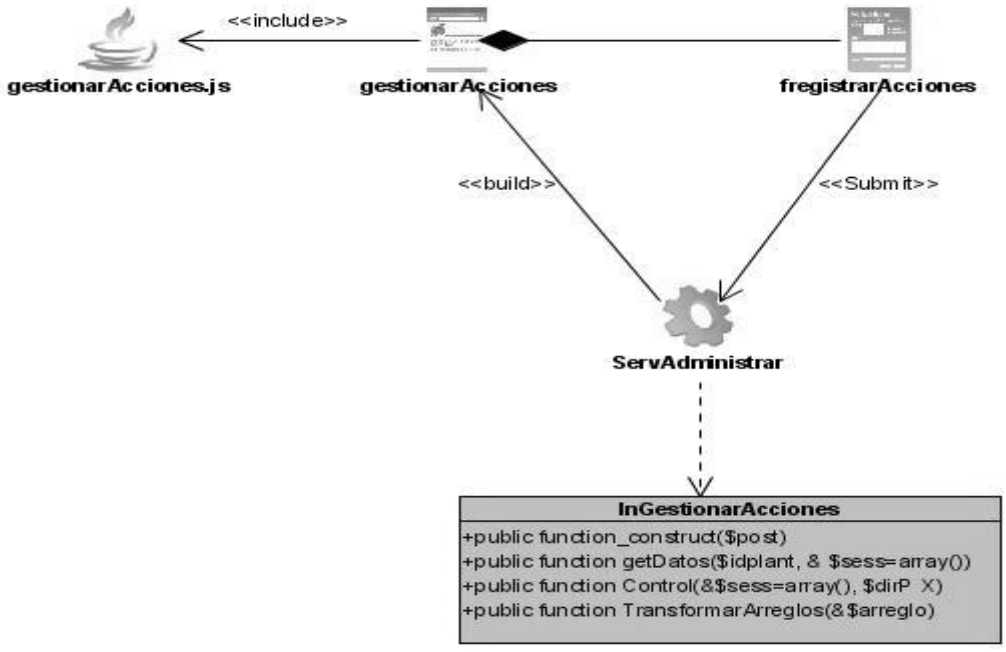


Figura 3.11 Diagrama de clases del diseño CU Gestionar acciones.

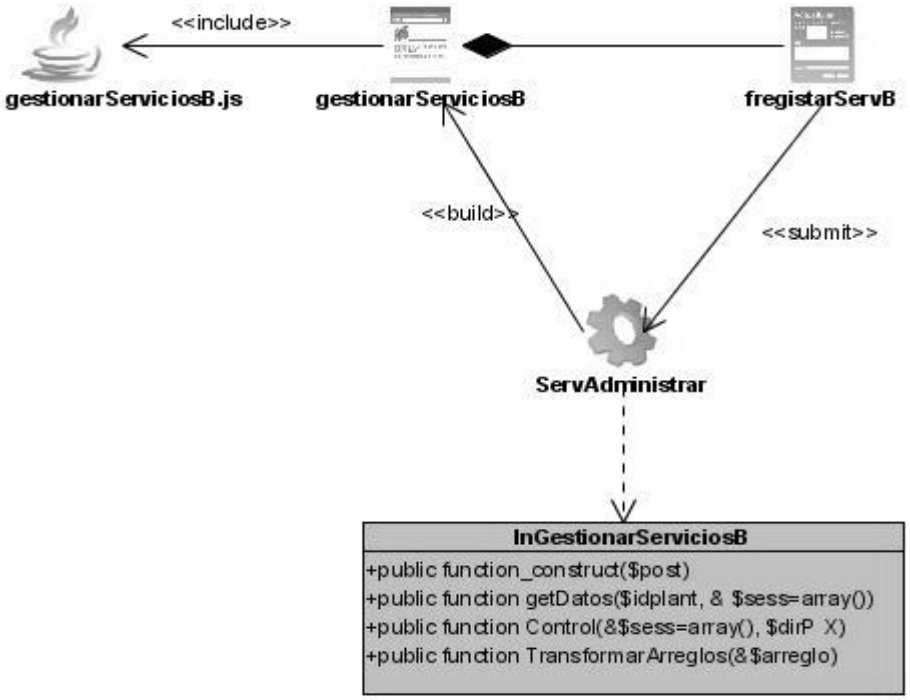


Figura 3.12 Diagrama de clases del diseño CU Gestionar servicio que brinda.

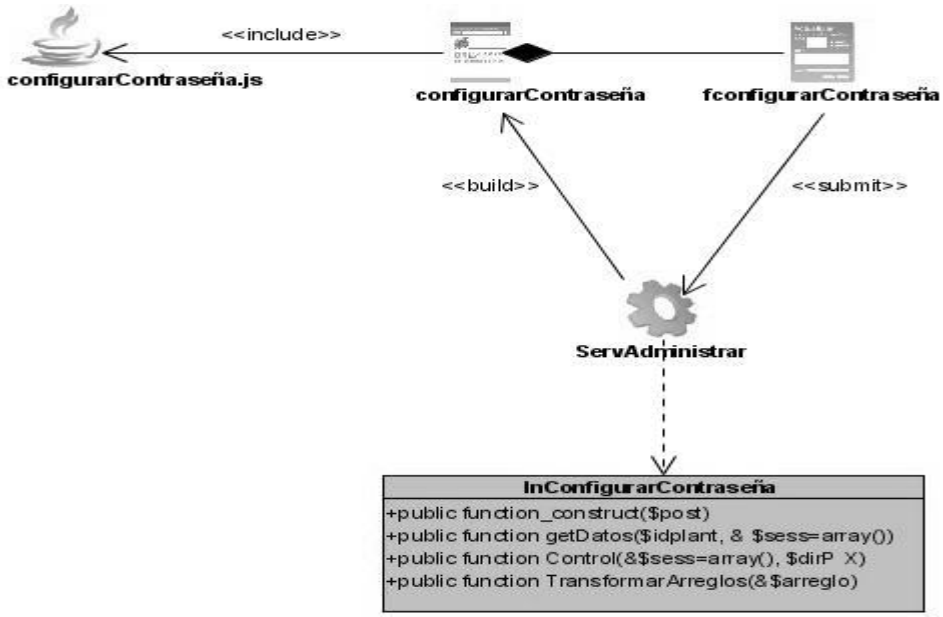


Figura 3.13 Diagrama de clases del diseño CU Configurar contraseña.

3.3.3. Diagramas de secuencia.

A continuación se muestran los diagramas de secuencia de el caso de uso Gestionar sistemas. Para cada caso de uso se realizó un diagrama para cada escenario y uno para el mostrar la interfaz general del caso de uso. El resto de los diagramas se encuentran en los anexos (Ver el **Anexo 3**). Para la definición de las funciones se utilizó como estándar de codificación la notación CamellCasing.

Diagramas de secuencia CU Gestionar sistemas

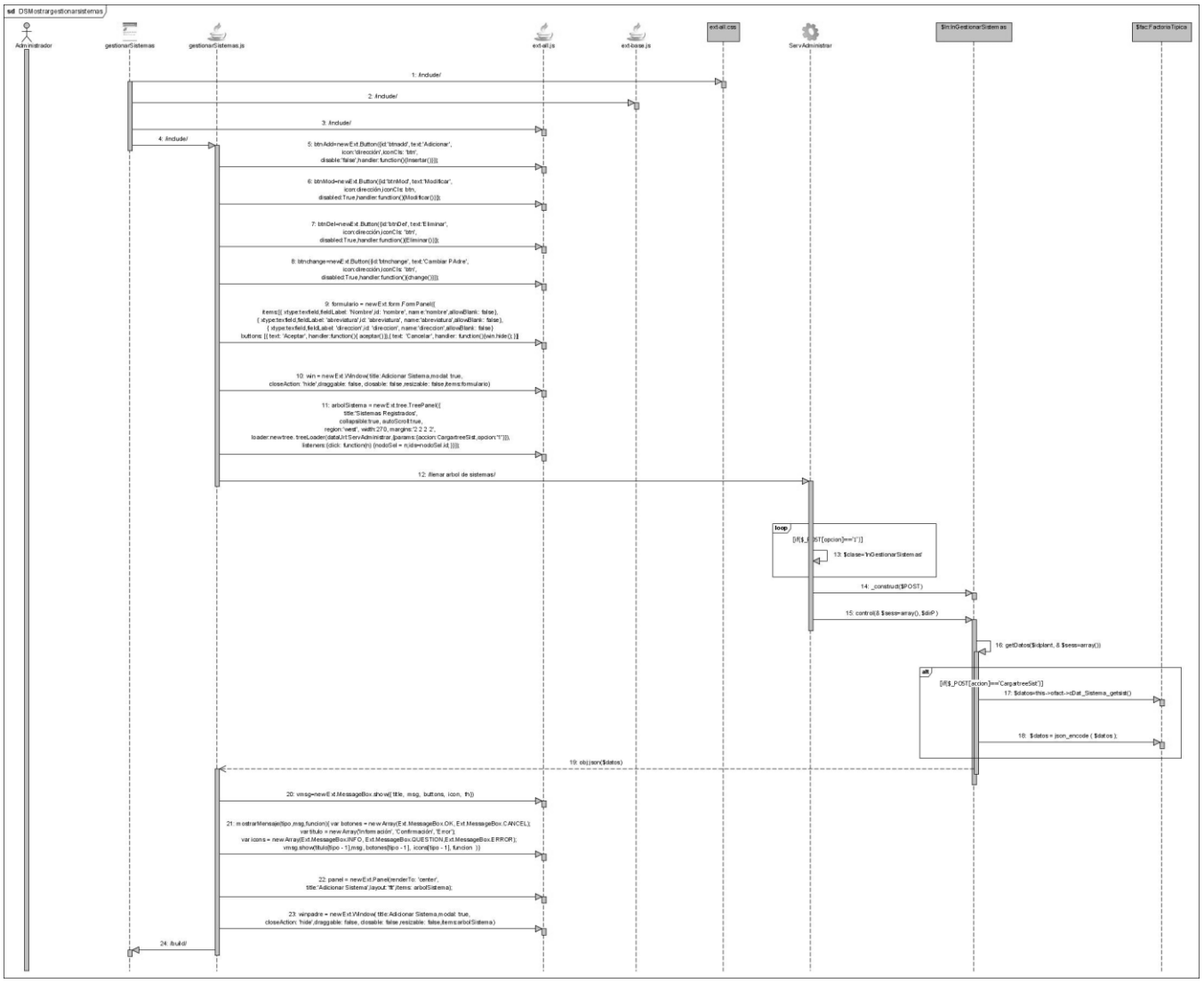


Figura 3.14 Diagrama de secuencia Mostrar Gestionar sistemas

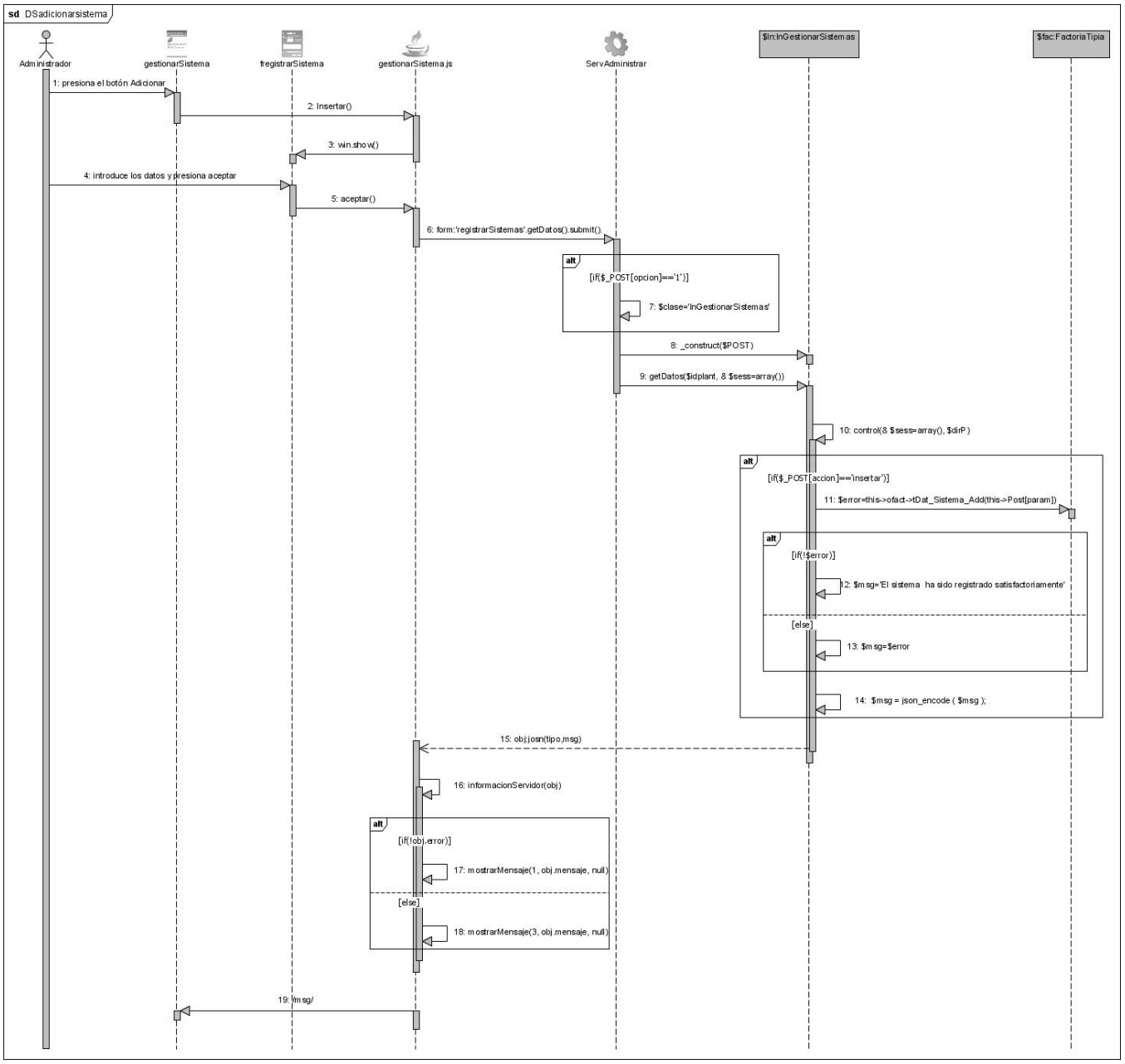


Figura 3.15 Diagrama de secuencia Adicionar sistema

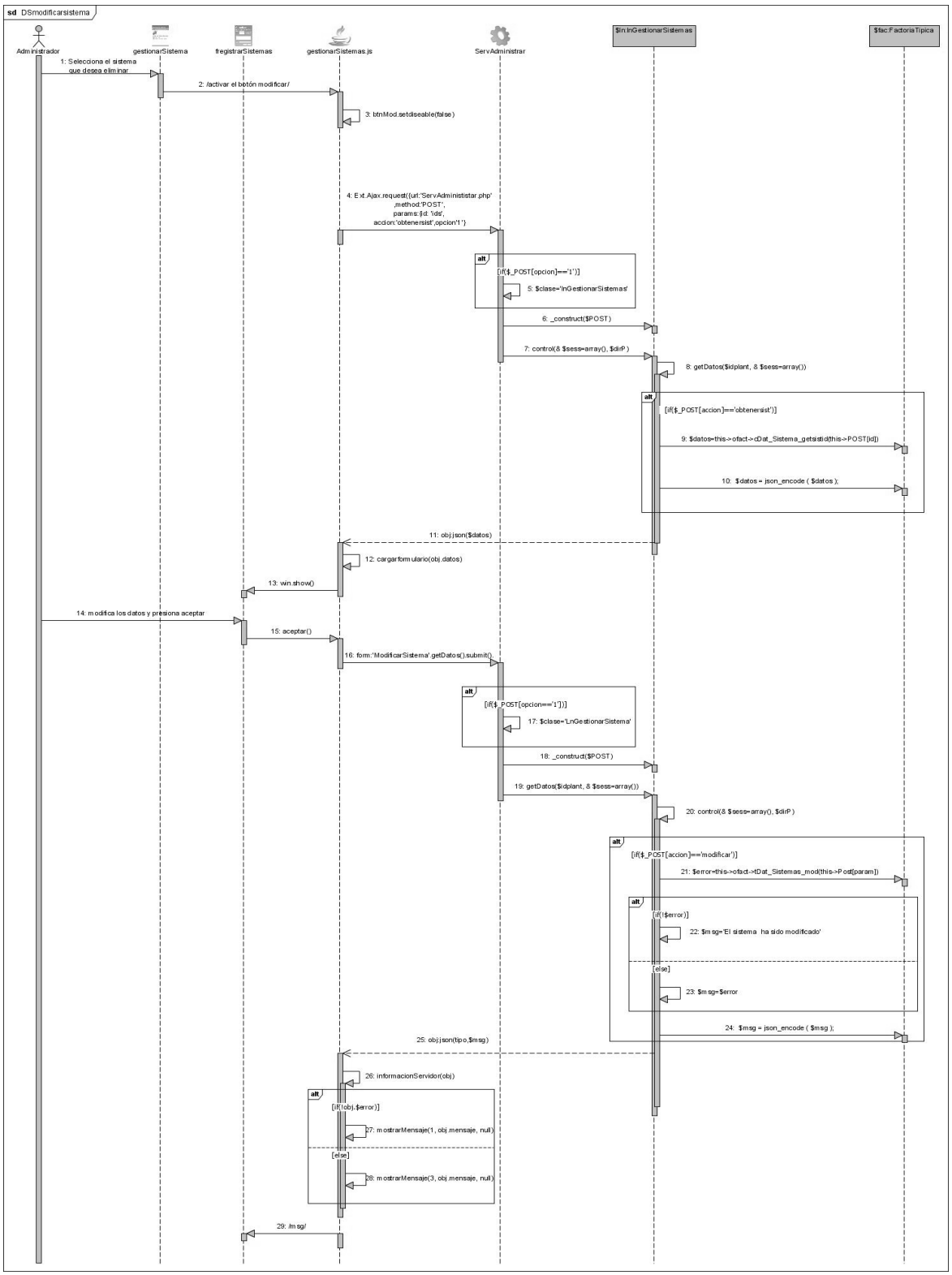


Figura 3.16 Diagrama de secuencia Modificar sistema

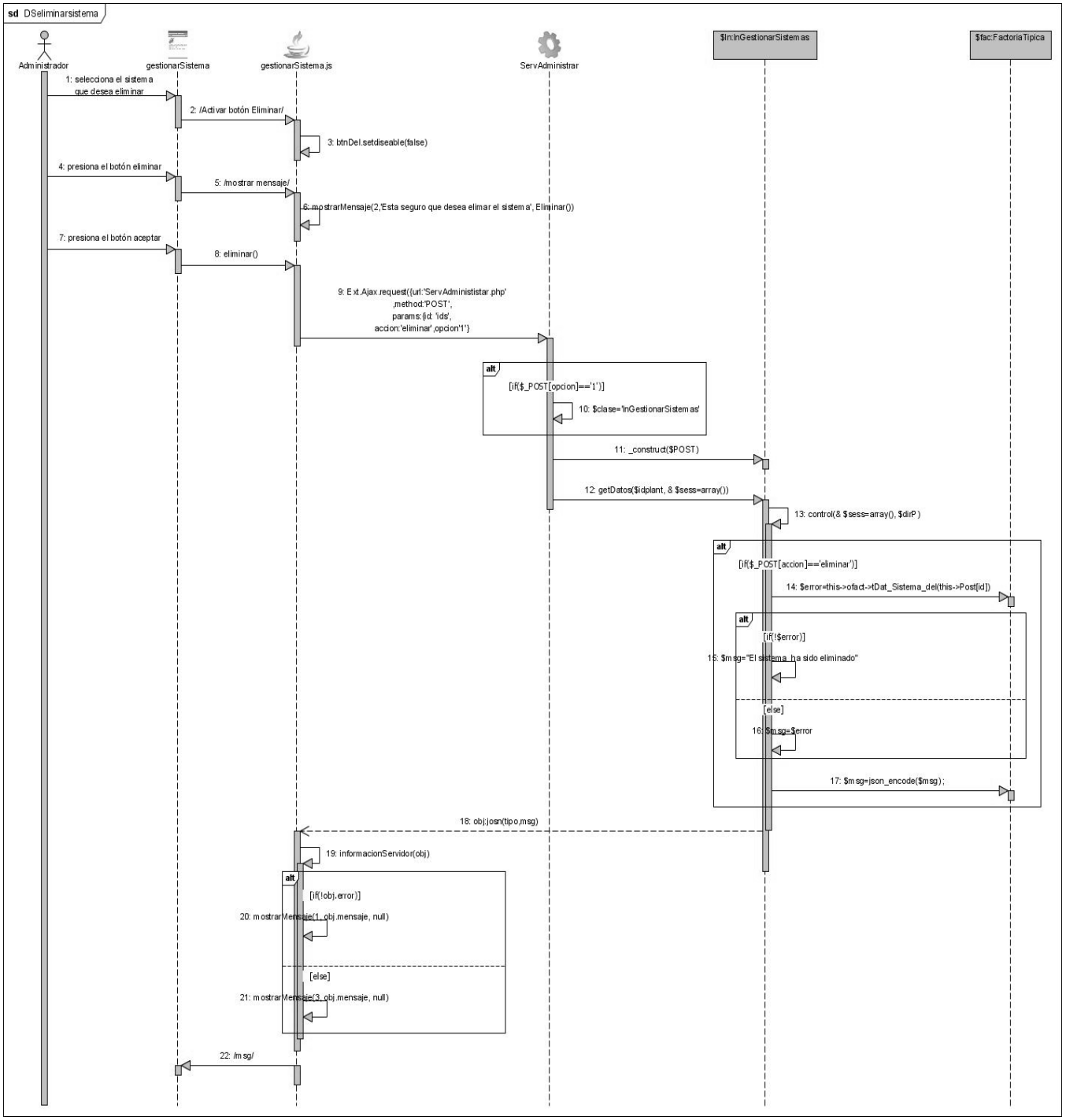


Figura 3.17 Diagrama de secuencia Eliminar sistema

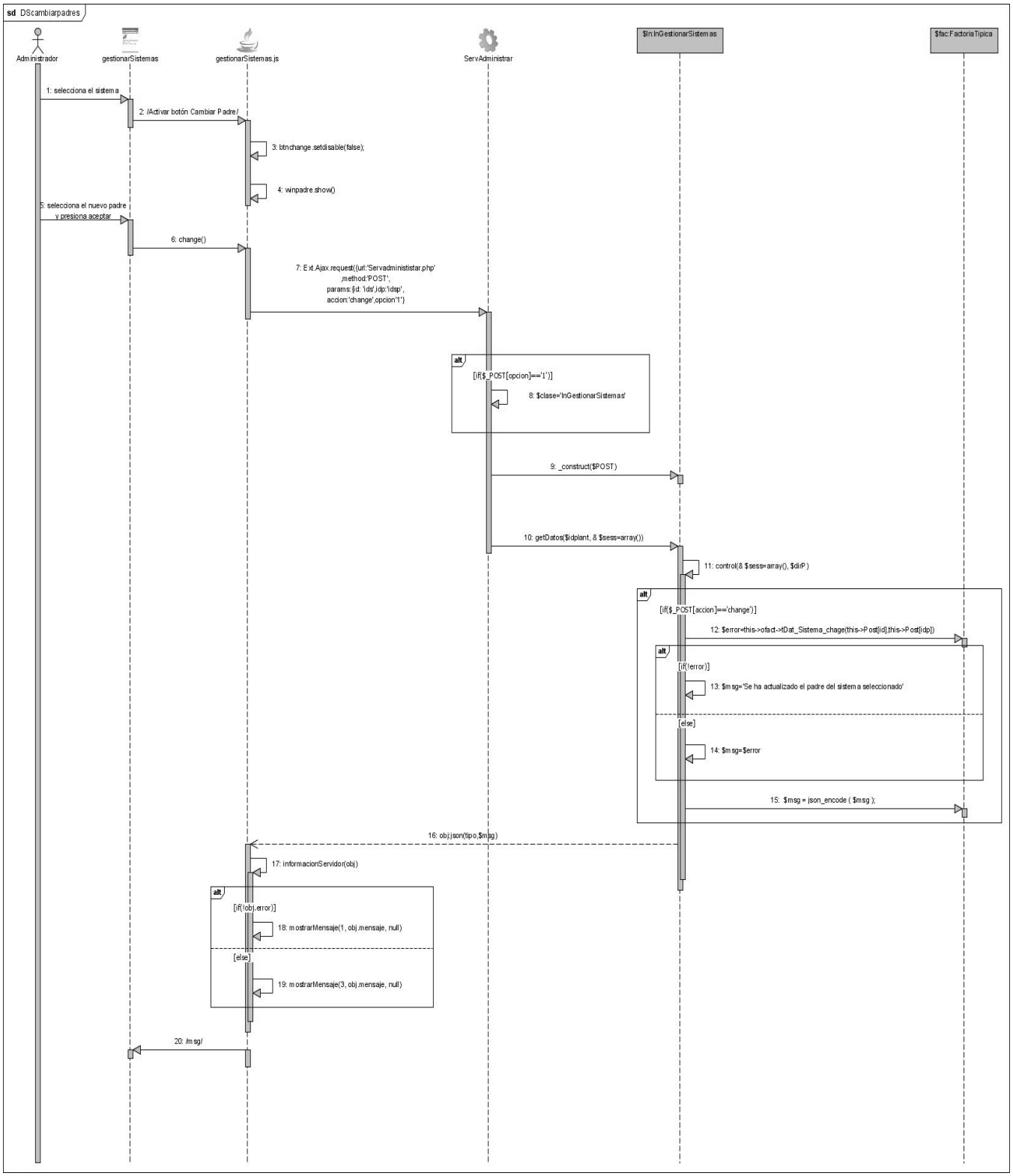
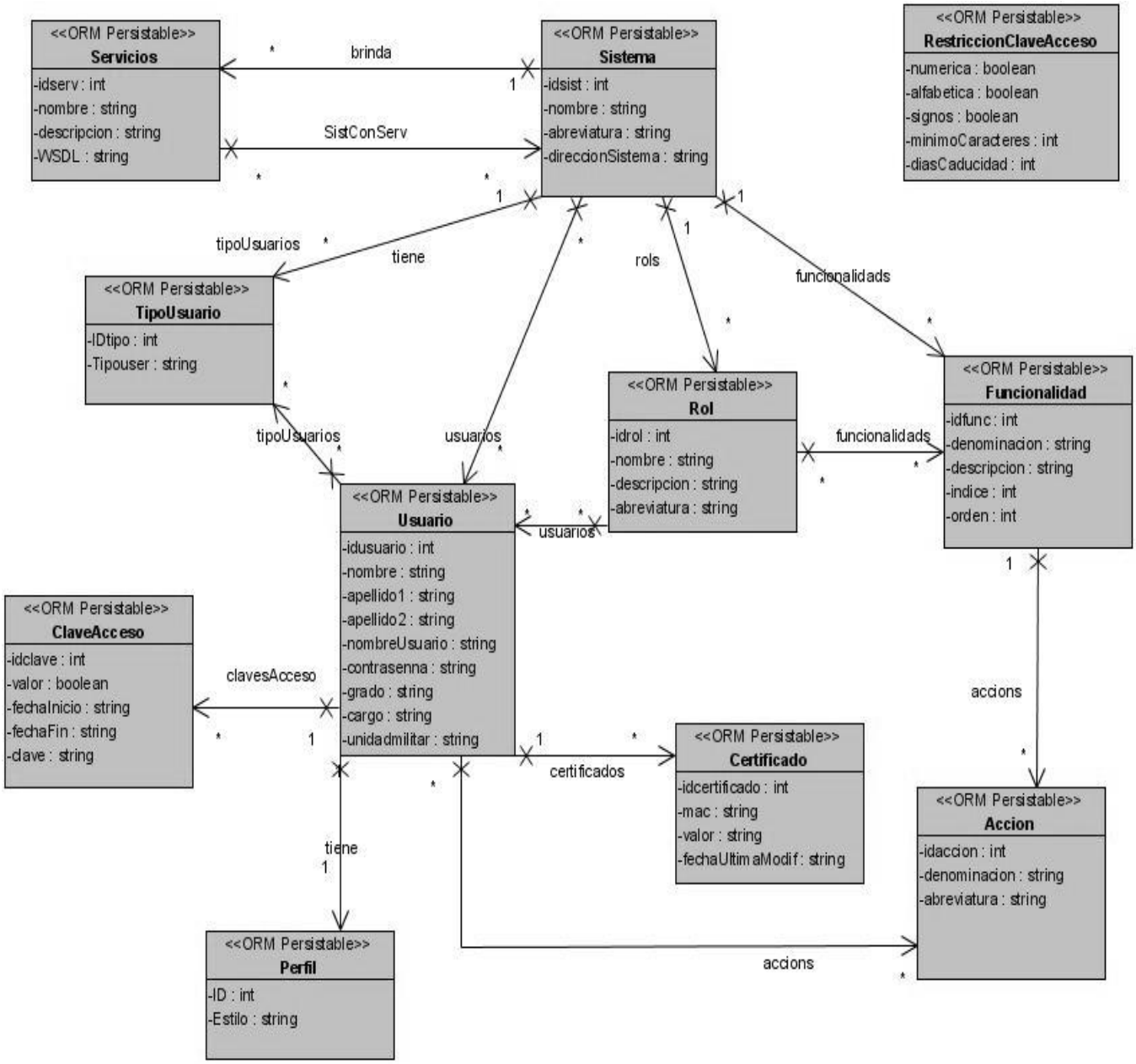


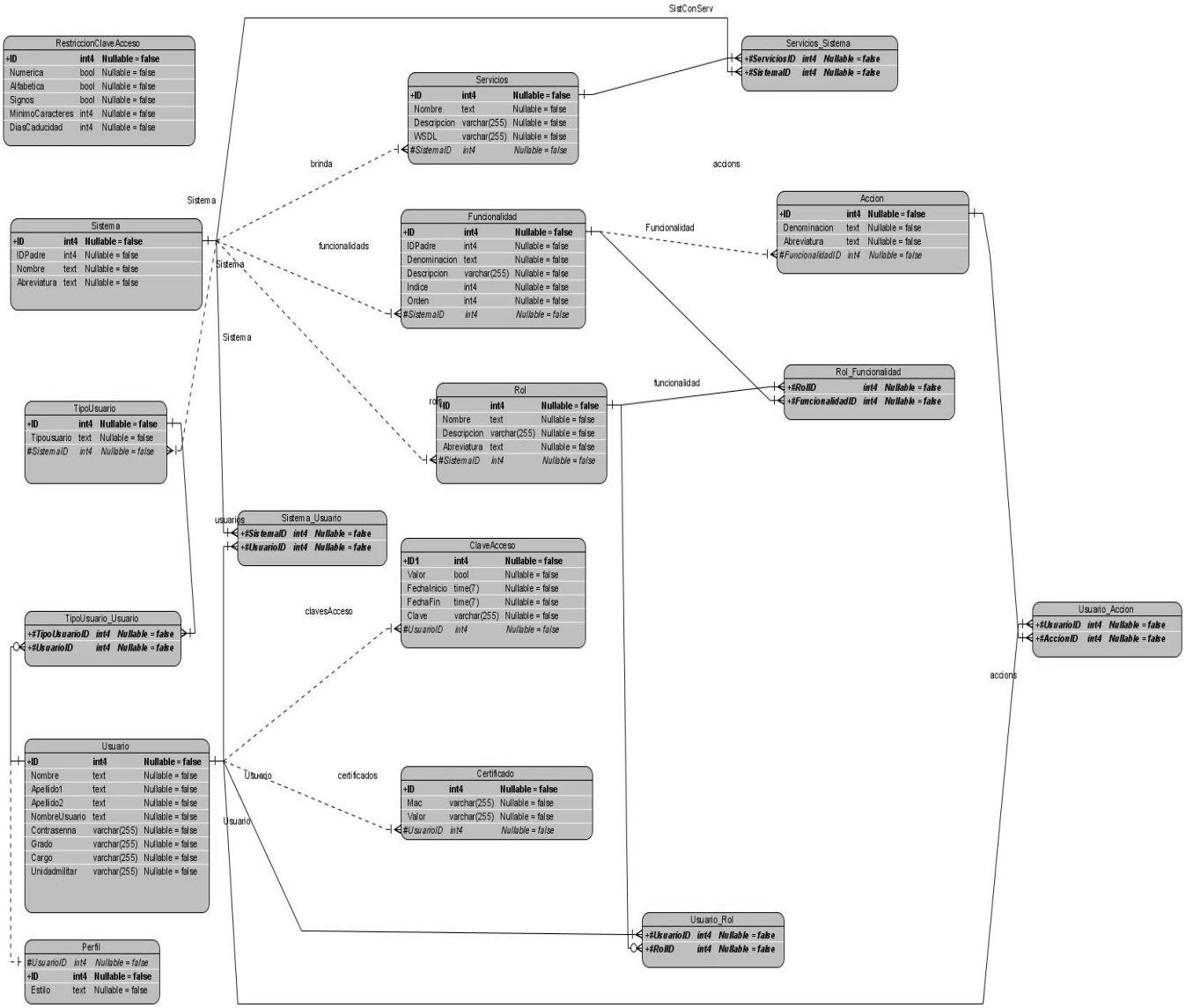
Figura 3.18 Diagrama de secuencia Cambiar padre de un sistema

3.4. Diseño de la base de datos

3.4.1. Diagrama de clases persistentes



3.4.2. Diagrama Entidad Relación de la Base de Datos



3.4.3. Descripción de las tablas

Nombre: Usuario		
Descripción: Esta tabla almacena los datos de los usuarios registrados.		
Atributo	**Tipo**	**Descripción**
id	integer	Identificador del usuario.
Nombre	text	Nombre real del usuario.

Apellido1	text	Primer apellido del usuario.
Apellido2	text	Segundo Apellido del usuario.
NombreUsuario	text	Nombre de usuario.
Contrasenna	varchar	Contraseña de usuario
Grado	varchar	Grado militar.
Cargo	varchar	Cargo militar.
Unidadmilitar	varchar	Unidad a la que pertenece.

Nombre: Sistema		
Descripción: Esta tabla almacena los datos de los sistemas ..		
Atributo	**Tipo**	**Descripción**
ID	integer	Identificador del sistema.
Nombre	text	Nombre del sistema.
Abreviatura	text	Abreviatura del sistema.
DireccionSistema	varchar	Dirección http del sistema.

Nombre: ClaveAcceso		
Descripción: Esta tabla almacena los datos de la clave de acceso de cada usuario.		
Atributo	**Tipo**	**Descripción**
ID	integer	Identificador de la clave.
Valor	bool	Especifica verdadero o falso para esa clave.
FechaInicio	time	Fecha de creada la contraseña.
FechaFin	Time	Fecha en que caduca.
Clave	varchar	Contraseña de usuario.
#UsuarioID	integer	Identificador del usuario que tienes esa contraseña.

Nombre: Certificado		
Descripción: Esta tabla almacena los datos del certificado que se le crea al usuario.		
Atributo	**Tipo**	**Descripción**
ID	integer	Identificador del certificado.
Mac	varchar	Mac de la computadora desde la que accede el usuario.
Valor	varchar	El valor del certificado.
#UsuarioID	integer	Identificar de usuario que le corresponde el certificado.

Nombre: Rol

Descripción: Esta tabla almacena los datos de los roles de los usuarios.		
Atributo	**Tipo**	**Descripción**
ID	integer	Identificador del rol.
Nombre	text	Nombre del rol.
Descripcion	varchar	Descripcion del rol.
Abreviatura	text	Abreviatura del rol.
#SistemaID	integer	Identificador del sistema.

Nombre: Servicios		
Descripción: Esta tabla almacena los datos de los servicios que brindan los sistemas registrados.		
Atributo	**Tipo**	**Descripción**
ID	integer	Identificador del servicio.
Nombre	text	Nombre del servicio.
Descripcion	varchar	Descripción del servicio.
WSDL	varchar	Dirección del WSDL del servicio.

Nombre: Funcionalidad		
Descripción: Esta tabla almacena los datos de las funcionalidades de los sistemas.		
Atributo	**Tipo**	**Descripción**
ID	integer	Identificador de la funcionalidad.
Denominacion	text	Denominación de la funcionalidad.
Descripcion	varchar	Descripción de la funcionalidad.
Indice	integer	Número, índice de la funcionalidad.
Orden	integer	Orden de la funcionalidad.
#SistemaID		Identificador del sistema que tiene la funcionalidad.

Nombre: Accion		
Descripción: Esta tabla almacena los datos de las funcionalidades de los sistemas.		
Atributo	**Tipo**	**Descripción**
ID	integer	Identificador de la acción.
Denominacion	text	Denominación de la acción.
Abreviatura	text	Abreviatura de la accion.
#FuncionalidadID	integer	Identificador de la funcionalidad.

Nombre: RestriccionClaveAcceso.
Descripción: Esta tabla almacena los datos de la restricción de claves de acceso.

Atributo	Tipo	Descripción
ID	integer	Identificador de la restricción.
Numerica	bool	Si requiere caracteres numéricos.
Alfabetica	bool	Si requiere letras.
Signos	bool	Si requiere signos.
MinimoCaracteres	integer	Cantidad de caracteres mínimos de las contraseñas.
DiasCaducidad	integer	Días de caducidad de las contraseñas.

Nombre: TipoUsuario		
Descripción: Esta tabla almacena los datos de los tipos de usuario para acceder a la base de datos.		
Atributo	**Tipo**	**Descripción**
ID	integer	Identificador del tipo de usuario.
Tipousuario	text	Nombre de usuario.
#SistemaID	integer	Sistema al que pertenece ese tipo de usuario.
Descripcion	text	Descripción del tipo de usuario.

Nombre: Perfil		
Descripción: Esta tabla almacena el nombre del estilo css de los usuarios.		
Atributo	**Tipo**	**Descripción**
ID	integer	Identificador del perfil.
Estilo	text	Nombre del estilo.
#UsuarioID	integer	Identificador del usuario que tiene ese perfil.

3.5. Conclusiones.

En este capítulo se presentó lo referente al flujo análisis y diseño del sistema, así como sus artefactos los cuales fueron generados con la herramienta Visual Paradigm para una mejor comprensión de la solución del problema. Este modelo al concluir la fase será de vital importancia para la siguiente ya que es considerado la entrada principal para las siguientes actividades de implementación y prueba.

CONCLUSIONES

Al concluir el presente trabajo de diploma se le ha dado cumplimiento al objetivo propuesto, pues ha quedado diseñado un sistema independiente de las demás aplicaciones que garantiza la autenticación, autorización y administración de perfiles donde quedan reflejados los resultados de la investigación realizada a lo largo del trabajo.

Con la implementación del sistema diseñado se lograría una seguridad centralizada en un entorno de varias aplicaciones, evitando que en el futuro cada sistema tenga que implementar su propia seguridad y que los usuarios tengan que autenticarse más de una vez para acceder a los sistemas a los que tiene acceso. Además permitirá que la protección de los datos se corresponda con el nivel requerido en las instituciones de las FAR.

RECOMENDACIONES

- Implementar el sistema diseñado.
- Que en próximas iteraciones se tenga en cuenta la comunicación segura entre las aplicaciones.
- Confeccionar la ayuda y manual de usuario para facilitarle el trabajo a las personas que un futuro trabajen con el software.

REFERENCIAS BIBLIOGRÁFICAS

[1] . **Pakala, Sangita.** Preguntas Frecuentes sobre Seguridad Informática. [En línea] 2004. [Citado el:06 de Diciembre de 2007.] http://www.um.es/atica/documentos/FAQSeguridadAplicacionesWebOWASP.pdf

[2]. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software.* s.l. : Pearson Prentice Hall, 2004.

[3]. Delitosinformaticos.com. [En línea] 25 de Marzo de 2001. [Citado el: 06 de Diciembre de 2007.] http://www.delitosinformaticos.com/seguridad/clasificacion.shtml.

[4]. AccessMaster IAM (Gestión de Identidades y Acceso) & SSO (Single Sign-On). [En línea] 2003. [Citado el: 10 de Diciembre de 2007.] http://www.bull.es/security/IAM_SSO.htm.

[5]. **González C, Benjamín.** SOAP (Simple Object Access Protocol). [En línea] 2004 [Citado el: 20 de Enero de 2008.] http://www.desarrolloweb.com/articulos/1557.php

[6]. Teleformación.uci.cu. [En línea] [Citado: Diciembre 14, 2007.] http://teleformacion.uci.cu/.

[7]. AJAX un nuevo acercamiento a Aplicaciones Web [En línea] [Citado: 10 de Diciembre de 2007.] http://www.uberbin.net/archivos/internet/ajax-un-nuevo-acercamiento-a-aplicaciones-web.php

[8]. ¿Que es LDAP? [En línea] 2004 [Citado: 10 de Diciembre de 2007.] http://www.ldap-es.org/node/21

[9]. **Alvarez, Miguel Angel**. Qué es PHP [En línea] [Citado: 10 de Diciembre de 2007.] http://www.desarrolloweb.com/articulos/392.php#arriba

[10]. Digitales, Dpto. Sistema. Tele formación. “Control de acceso. Identificación y autenticación”. [En línea] [Citado: 20 de Enero de 2008] http://teleformacion.uci.cu.

[11]. Visual Paradigm for UML.[En línea] 2007 [Citado el: 20 de Enero de 2008.] http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(Iglesia_Anglicana)_%5BMac_OS_X_cuenta_14717_p/

[12]. ext - Framewok JavaScript [En línea] 2007. [Citado el: 13 de Marzo de 2008.] http://pixelco.us/blog/ext-framewok-javascript/

[13]. **Prieto, Vladimir**. Ext 2.0. [En línea] 2007. [Citado el: 10 de diciembre de 2007.] http://www.estadobeta.com/2007/10/28/ext-20/

[14]. Programación Web. [En línea] 2006. [Citado el: 10 de diciembre de 2007.] http://lenguajes-de-programacion.com/programacion-web.shtml

BIBLIOGRAFÍA

Sibiontes.com. [En línea] 23 de Enero de 2005 . [Citado el: 20 de Enero de 2008] http://www.simbiontes.com/archives/tecnologia/ha-salido-postgresql-80.php.

Vegas, Jesús. infor.uva.es. [En línea] 21 de Marzo de 2002. [Citado el: 03 de Junio de 2008.] http://www.infor.uva.es/~jvegas/cursos/buendia/pordocente/node15.html.

Sacristán, Luis. [En línea] Marzo 10, 2008. [Citado el: Junio 2008, 2008.] http://sentidoweb.com/2008/03/10/phpext-libreria-php-para-ext-js.php.

Alvarez, Sara. Sistemas gestores de bases de datos. [En línea] 2007. [Citado el: Diciembre 12, 2007.] http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html.

Jacobson, Ivar, Booch, Grady y Rumbaugh, James. *El Proceso Unificado de Desarrollo de Software.* s.l. : Pearson Prentice Hall, 2004. 2.

Rodriguez, Alejandro. Apuntes DSI. [En línea] [Citado el: Diciembre 12, 2007.] http://petra.euitio.uniovi.es/~i1652585/DSI.pdf

Romero, Francisco. Sistemas Gestores De Bases De Datos [En línea] [Citado el: Diciembre 12, 2007.] http://www.iesromerovargas.net/OASIS2/SGBD/Documentos/T2.pdf

Que es PHP. [En línea] [Citado el: Diciembre 12, 2007.] http://www.phpya.com.ar/temarios/descripcion.php?cod=23

ANEXO 1 DIAGRAMAS DE CLASES DEL ANÁLISIS

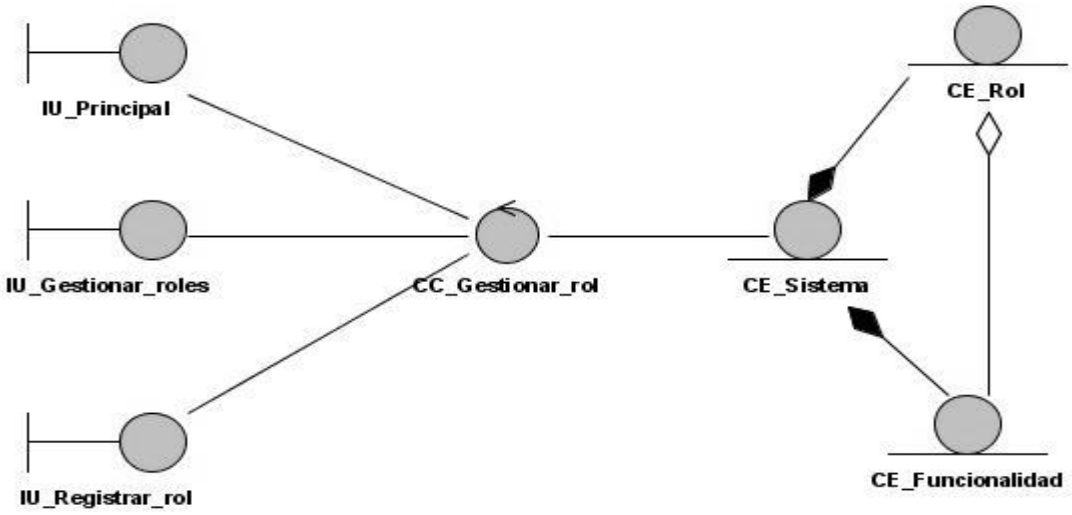


DIAGRAMA DE CLASES DEL ANÁLISIS CU GESTIONAR ROL

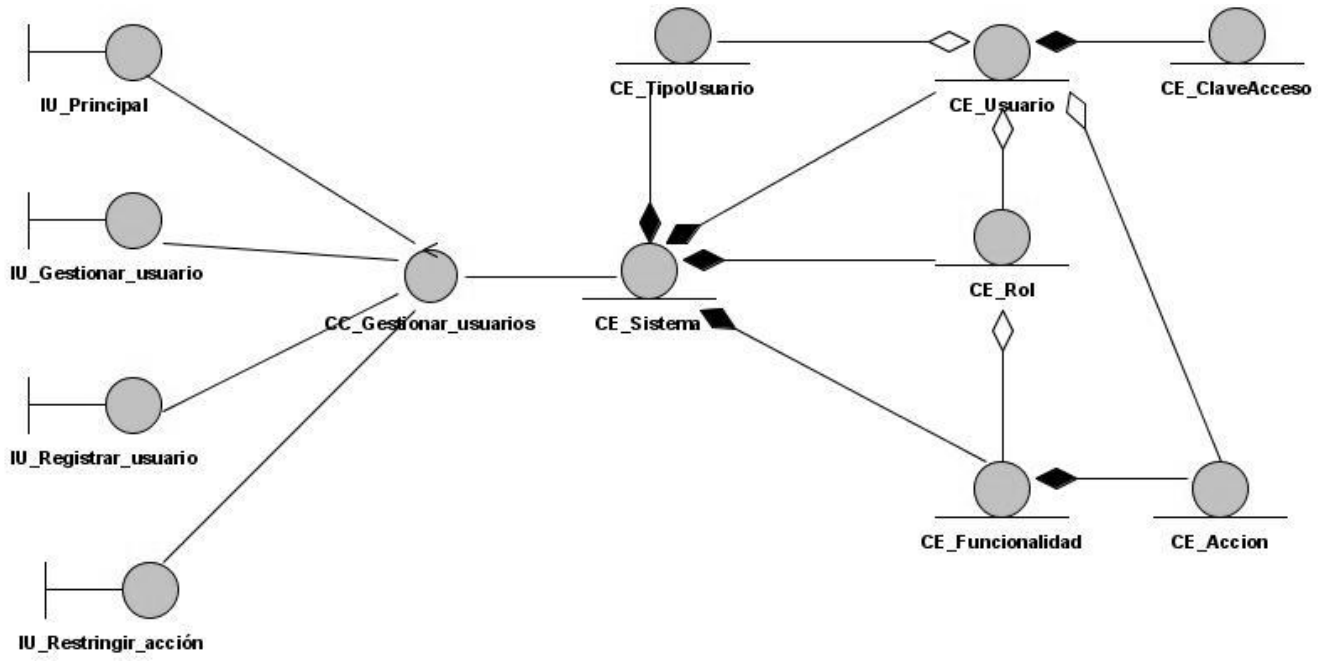


DIAGRAMA DE CLASES DEL ANÁLISIS CU GESTIONAR USUARIOS

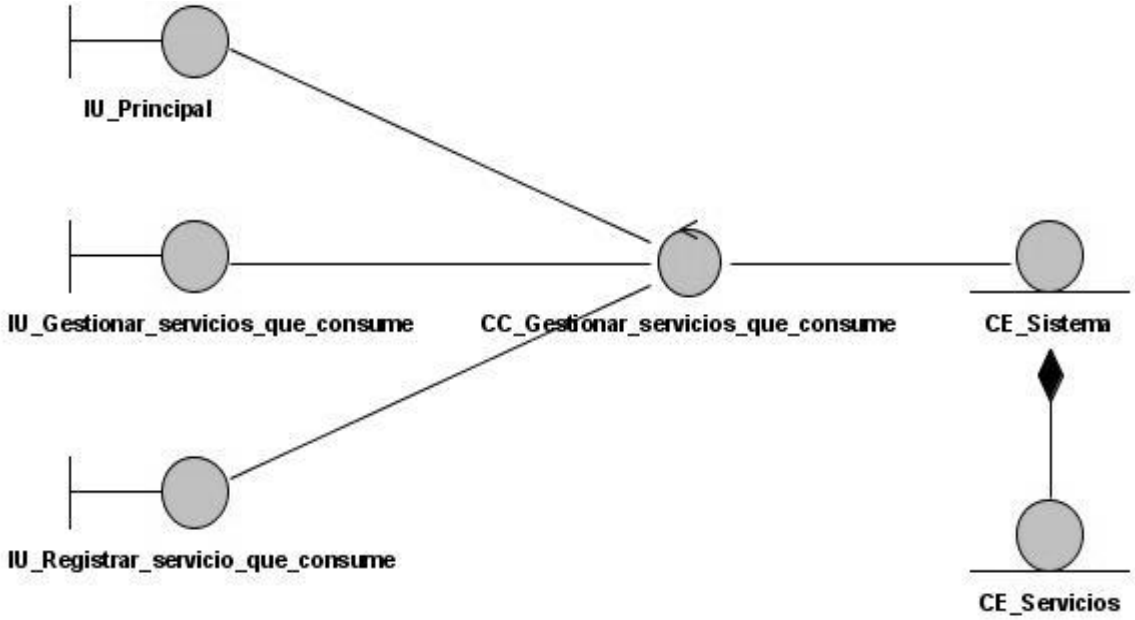


DIAGRAMA DE CLASES DEL ANÁLISIS CU GESTIONAR SERVICIOS QUE CONSUME

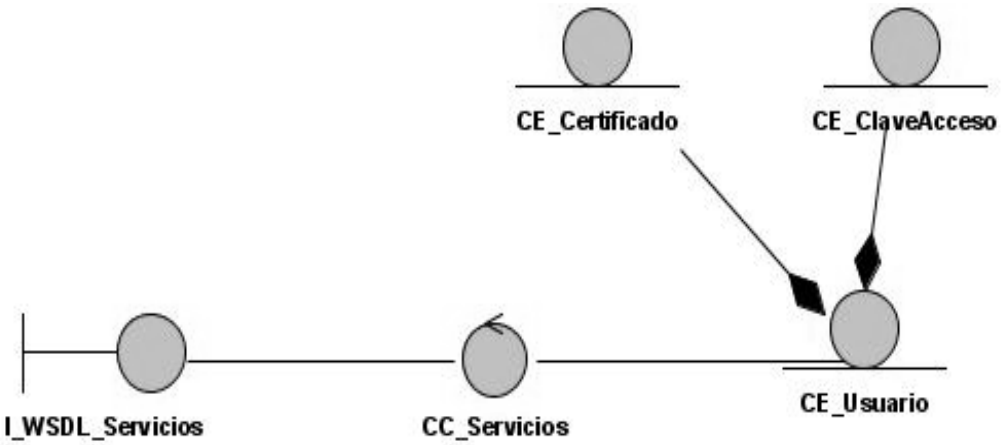


DIAGRAMA DE CLASES DEL ANÁLISIS CU AUTENTICAR USUARIO

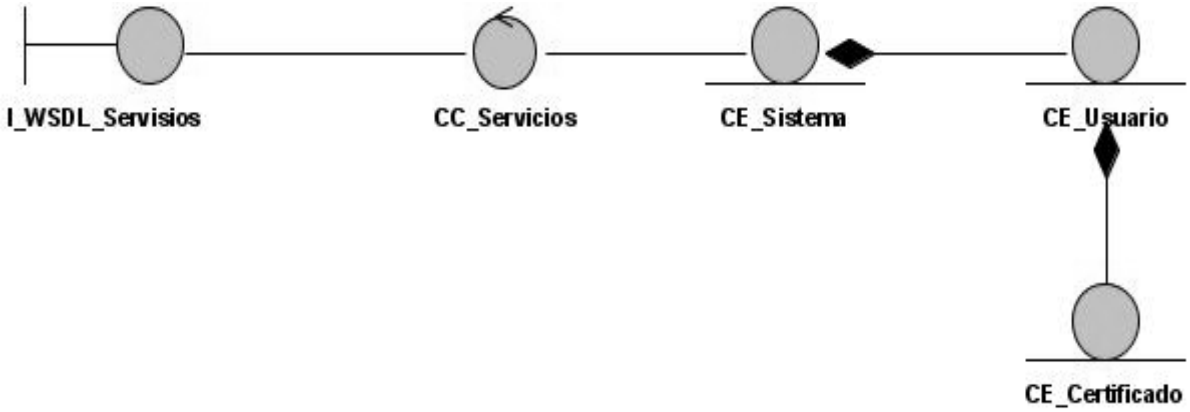


DIAGRAMA DE CLASES DEL ANÁLISIS CU AUTORIZAR ACCESO A SISTEMAS

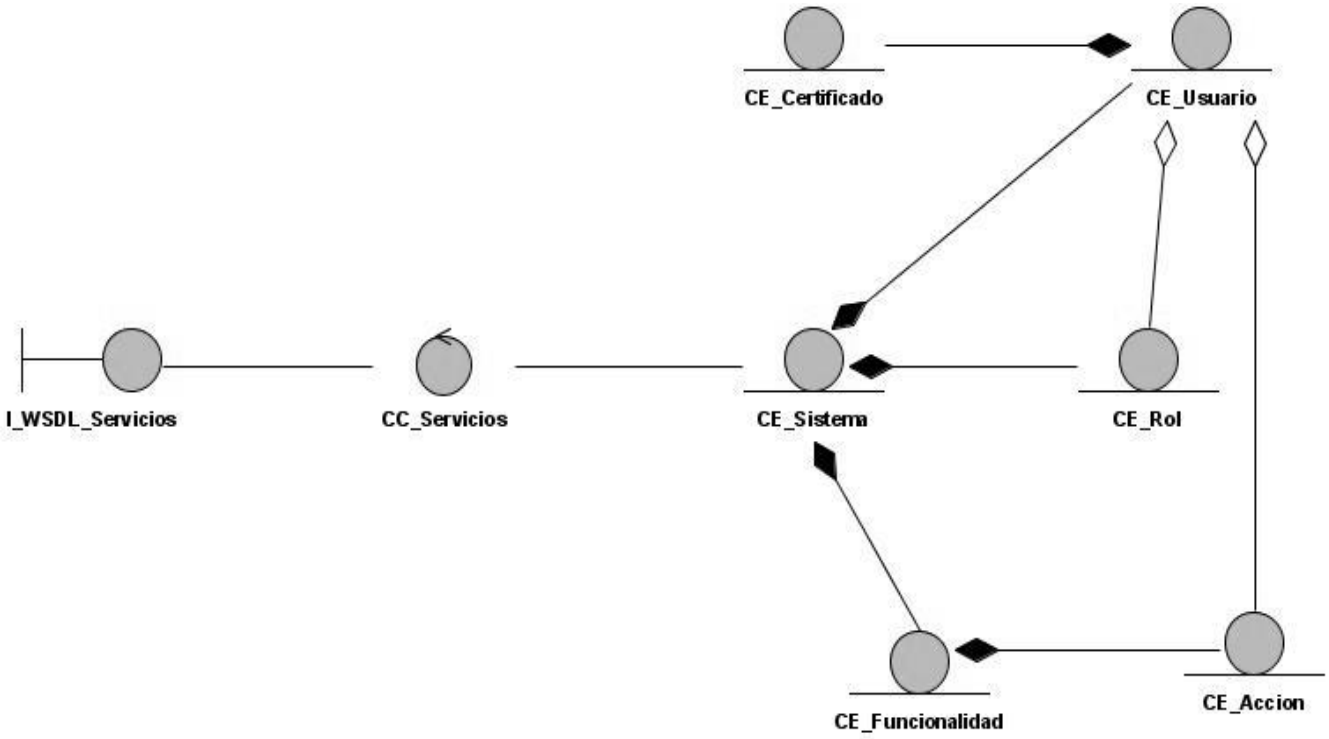


DIAGRAMA DE CLASES DEL ANÁLISIS CU CARGAR MENÚ

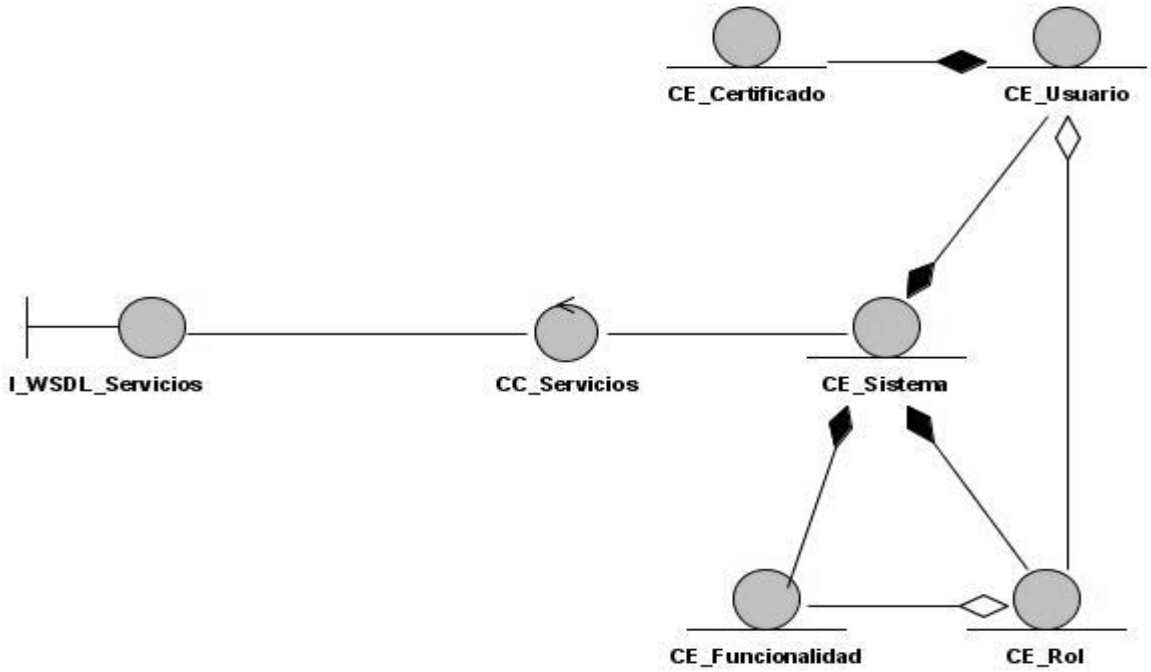


DIAGRAMA DE CLASES DEL ANÁLISIS CU AUTORIZAR ACCESO A FUNCIONALIDAD

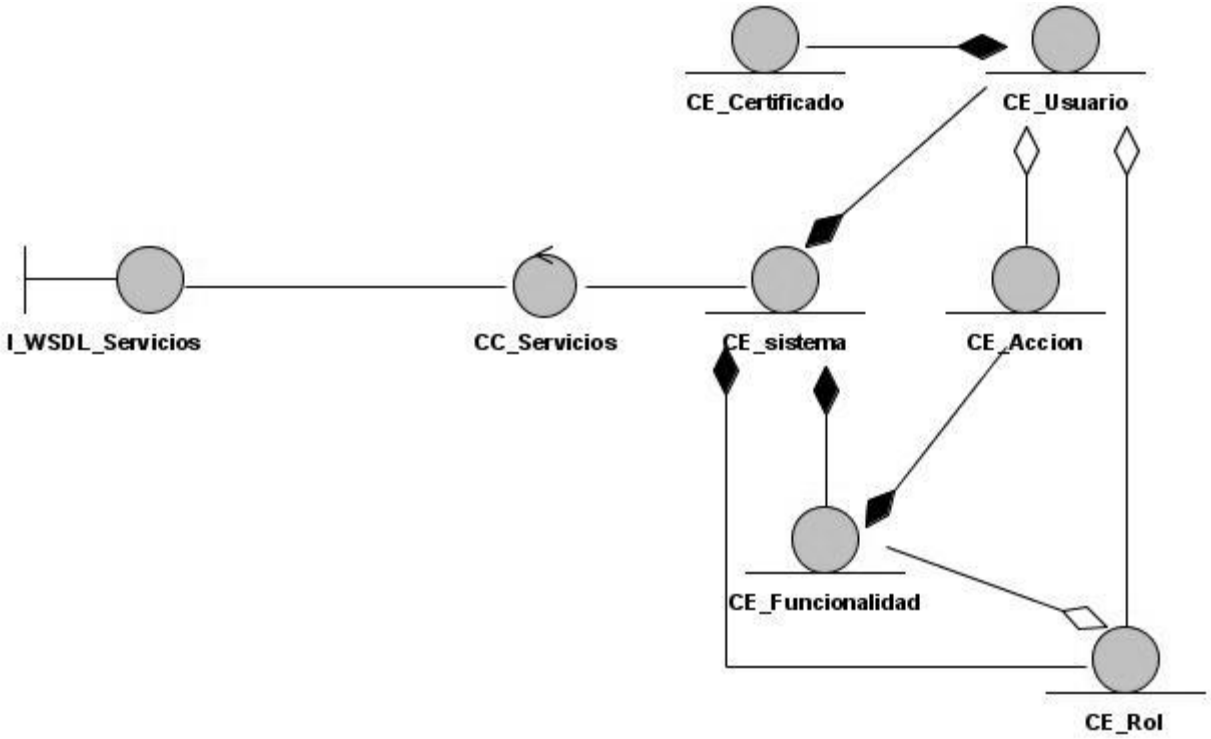


DIAGRAMA DE CLASES DEL ANÁLISIS CU AUTORIZAR ACCESO A ACCIÓN

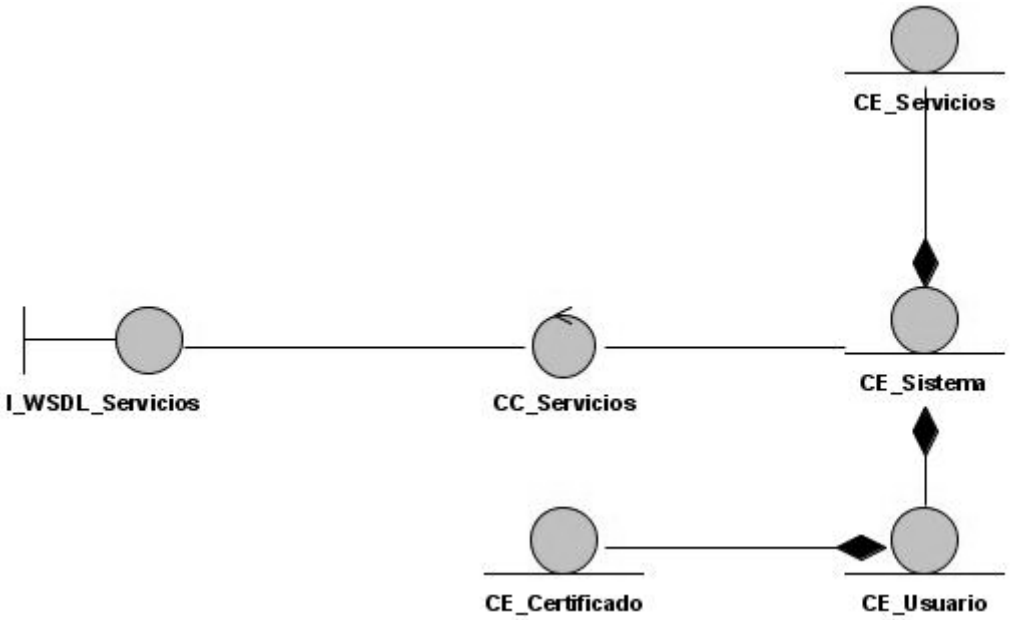


DIAGRAMA DE CLASES DEL ANÁLISIS CU AUTORIZAR ACCESO A SERVICIO

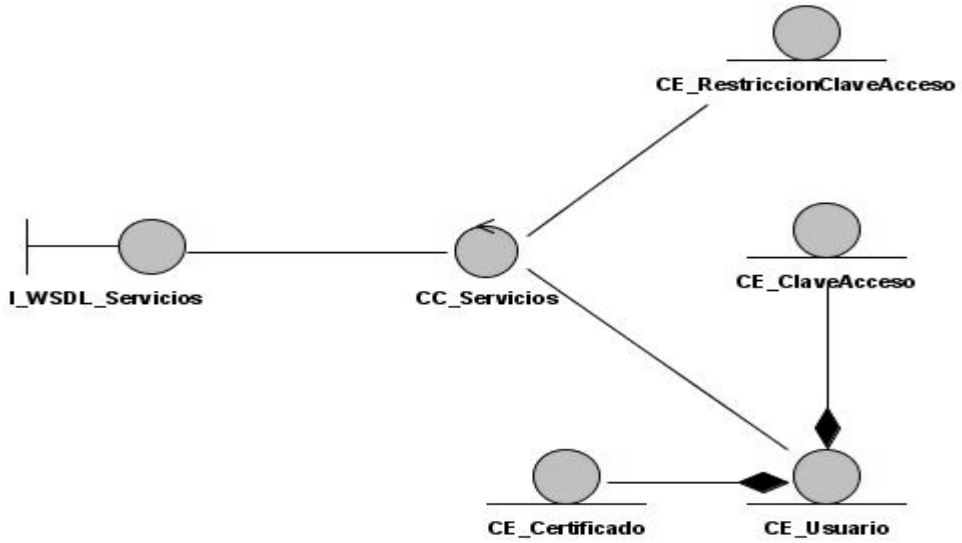


DIAGRAMA DE CLASES DEL ANÁLISIS CU AUTORIZAR CAMBIAR CONTRASEÑA

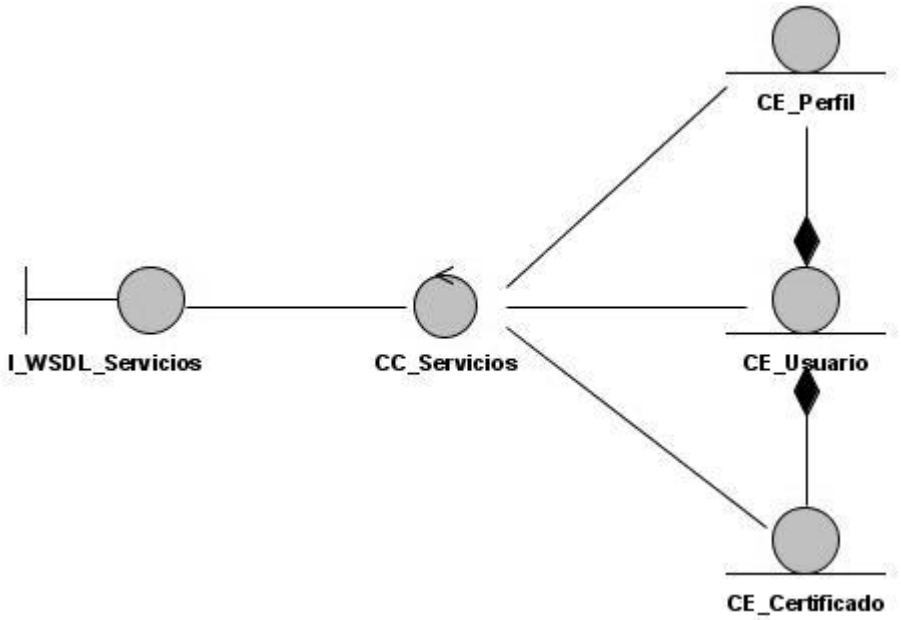


DIAGRAMA DE CLASES DEL ANÁLISIS CU AUTORIZAR EDITAR PERFIL

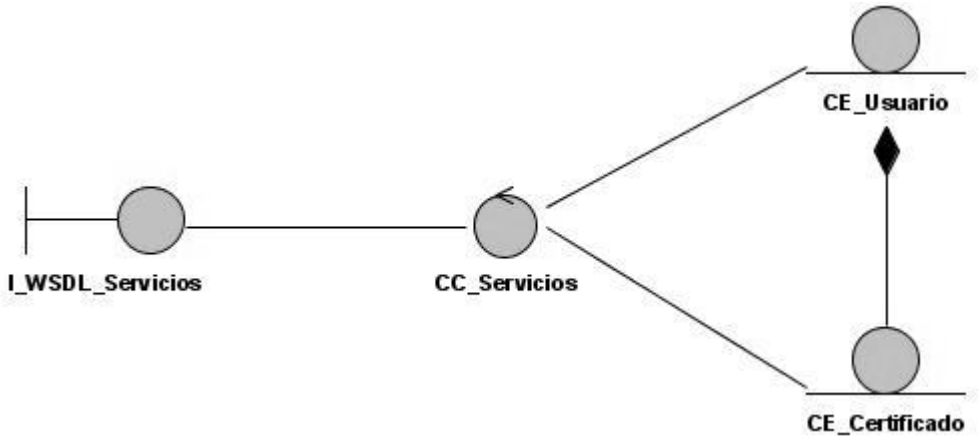


DIAGRAMA DE CLASES DEL ANÁLISIS CU CERRAR SESIÓN DE USUARIO.

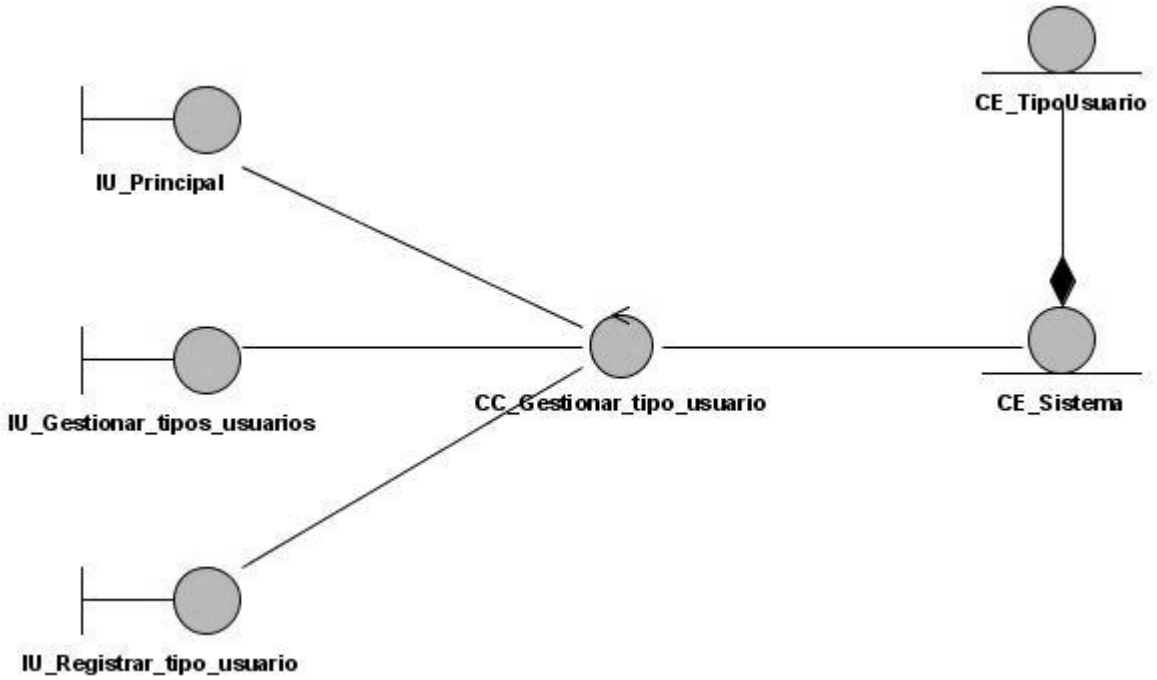


DIAGRAMA DE CLASES DEL ANÁLISIS CU GESTIONAR TIPOS DE USUARIOS.

ANEXO 2 DIAGRAMAS DE CLASES DEL DISEÑO

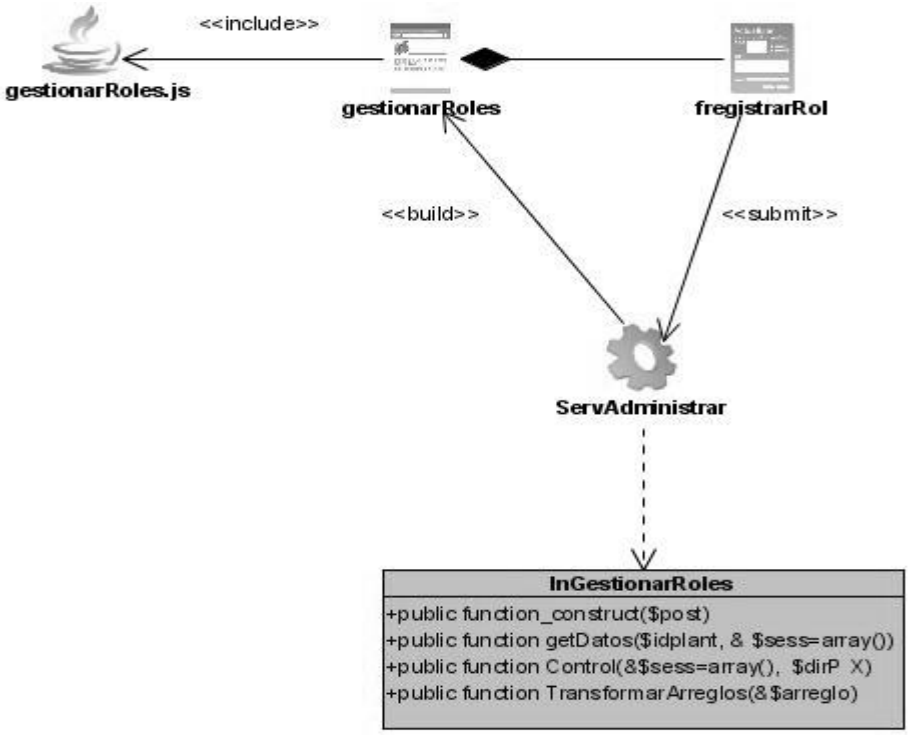


DIAGRAMA DE CLASES DEL DISEÑO CU GESTIONAR ROLES

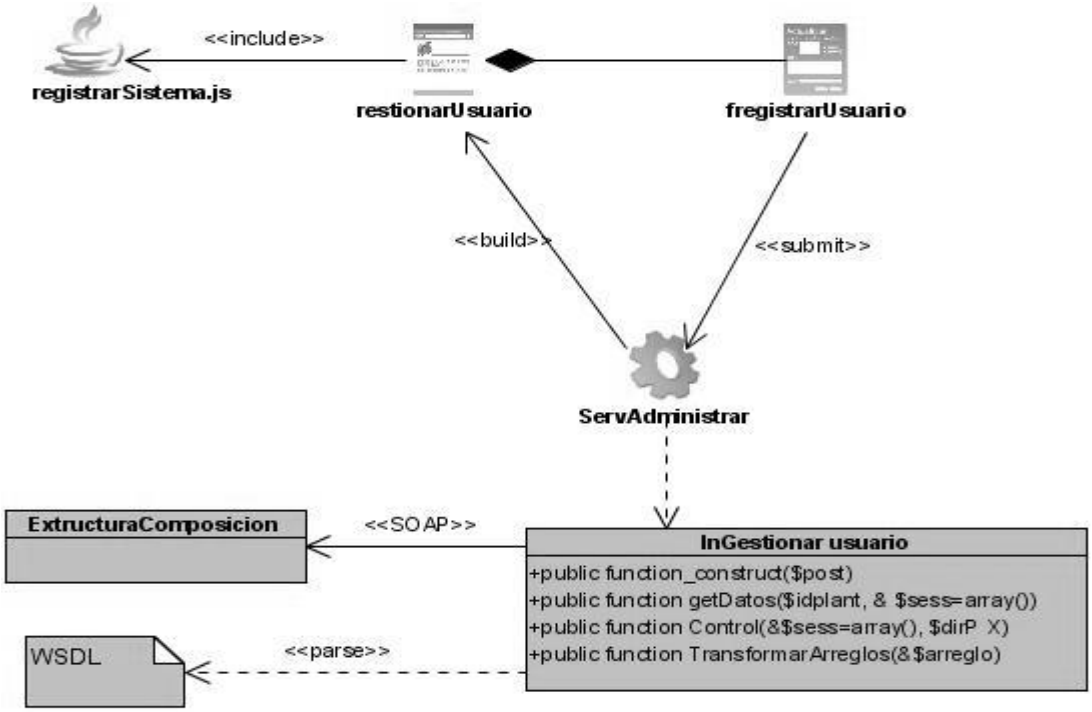


DIAGRAMA DE CLASES DEL DISEÑO CU GESTIONAR USUARIOS

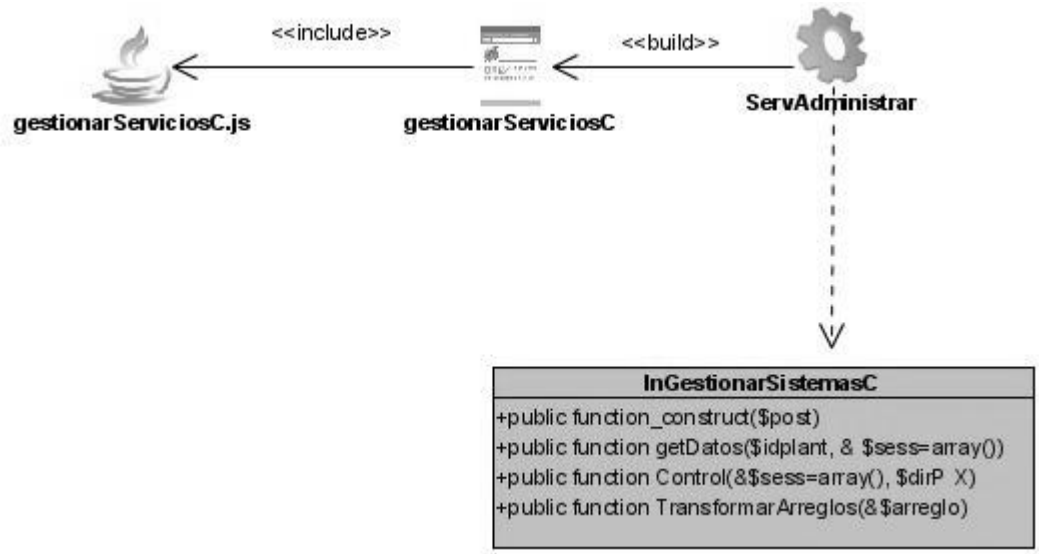


DIAGRAMA DE CLASES DEL DISEÑO CU GESTIONAR SERVICIOS QUE CONSUME

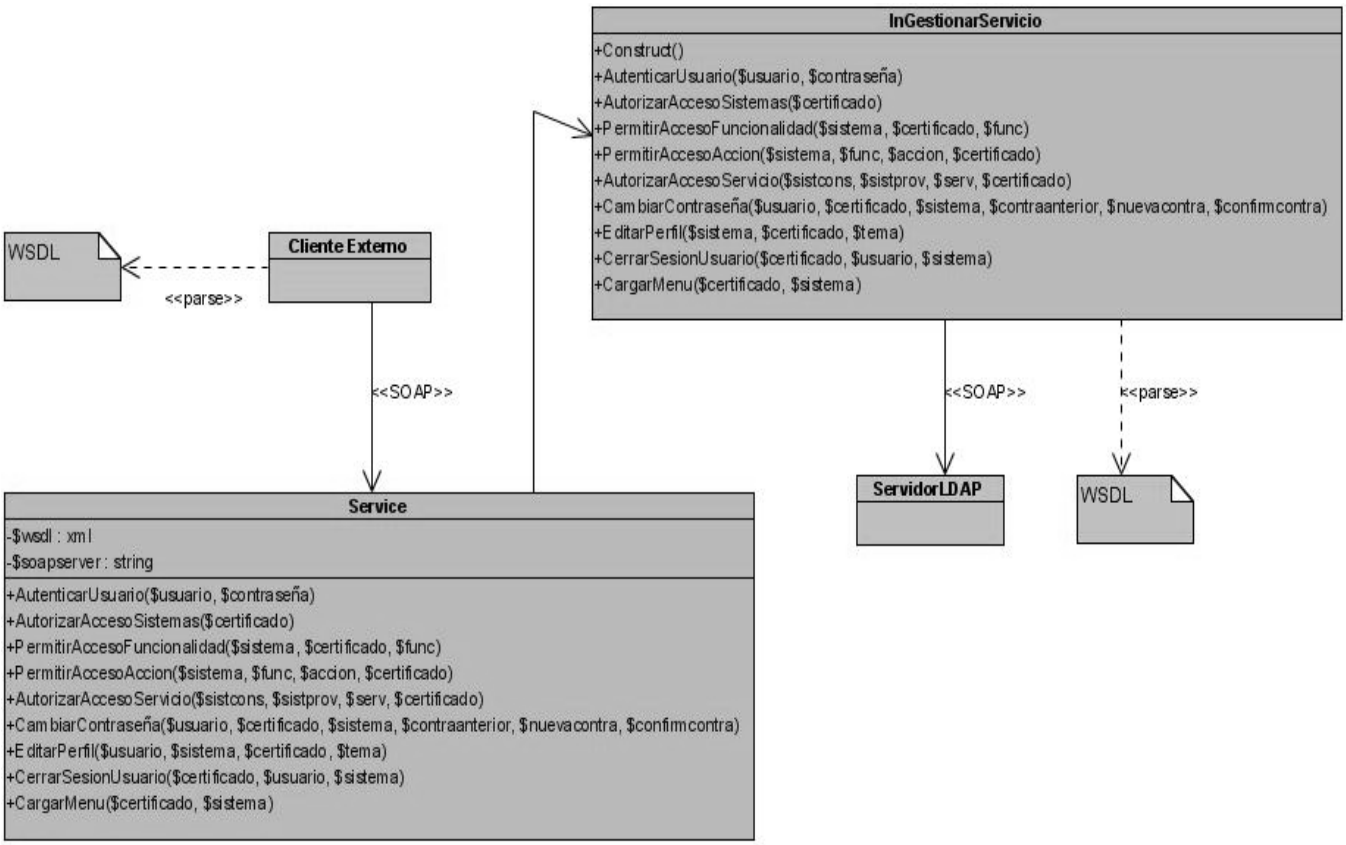


DIAGRAMA DE CLASES DEL DISEÑO DE LOS SERVICIOS

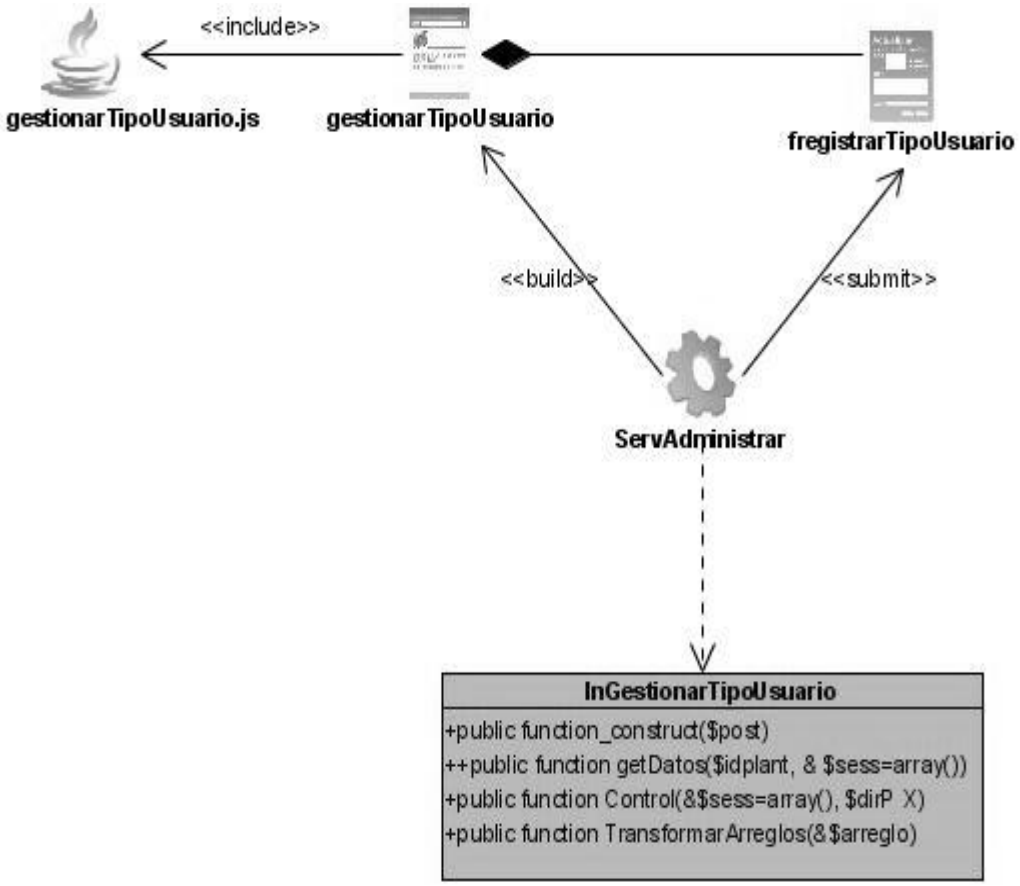


DIAGRAMA DE CLASES DEL DISEÑO CU GESTIONAR TIPOS DE USUARIOS

ANEXO 3 DIAGRAMAS DE SECUENCIA

DIAGRAMAS DE SECUENCIA CU GESTIONAR FUNCIONALIDAD

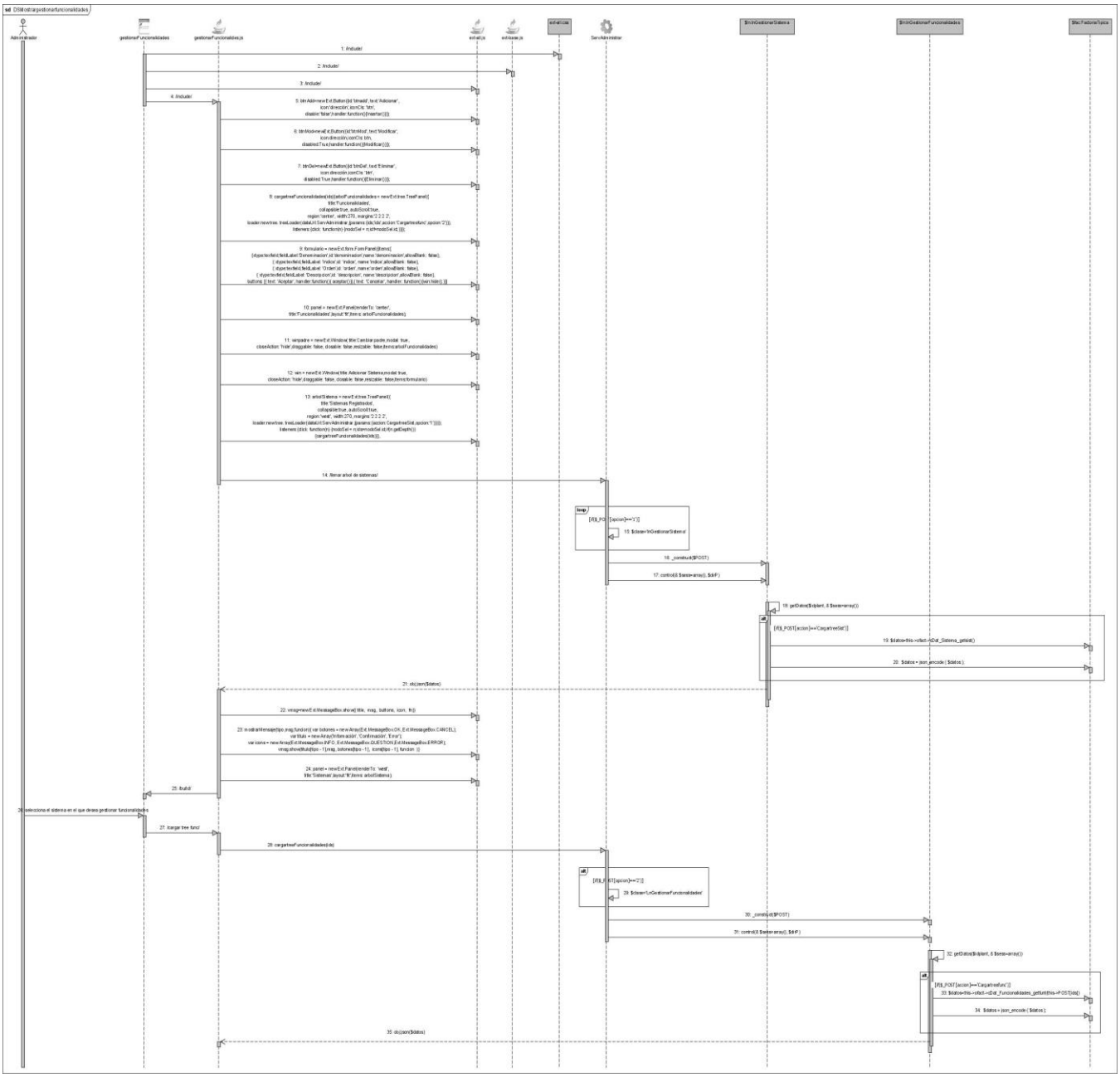


DIAGRAMA DE SECUENCIA MOSTRAR GESTIONAR FUNCIONALIDADES

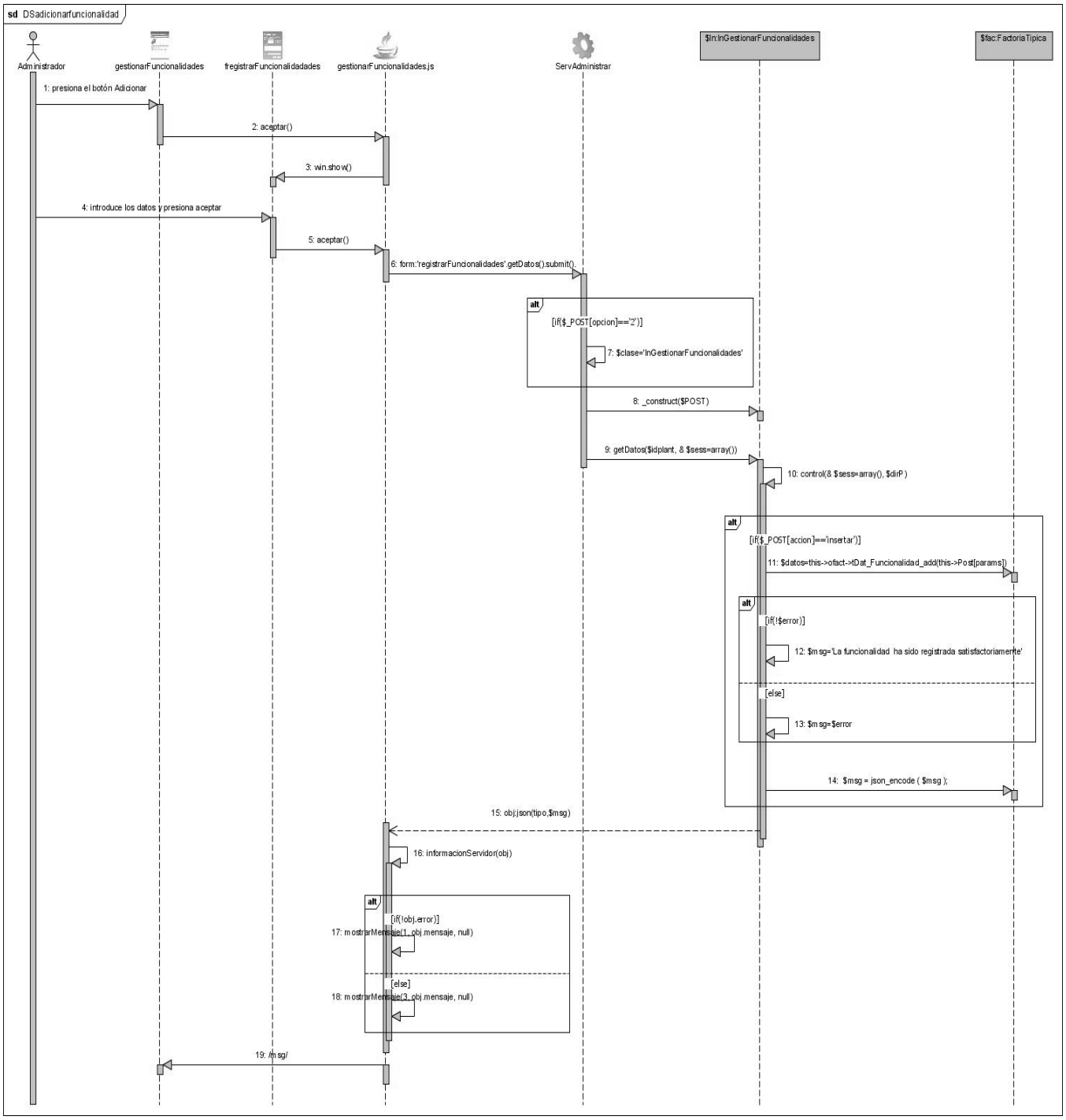


DIAGRAMA DE SECUENCIA REGISTAR FUNCIONALIDAD

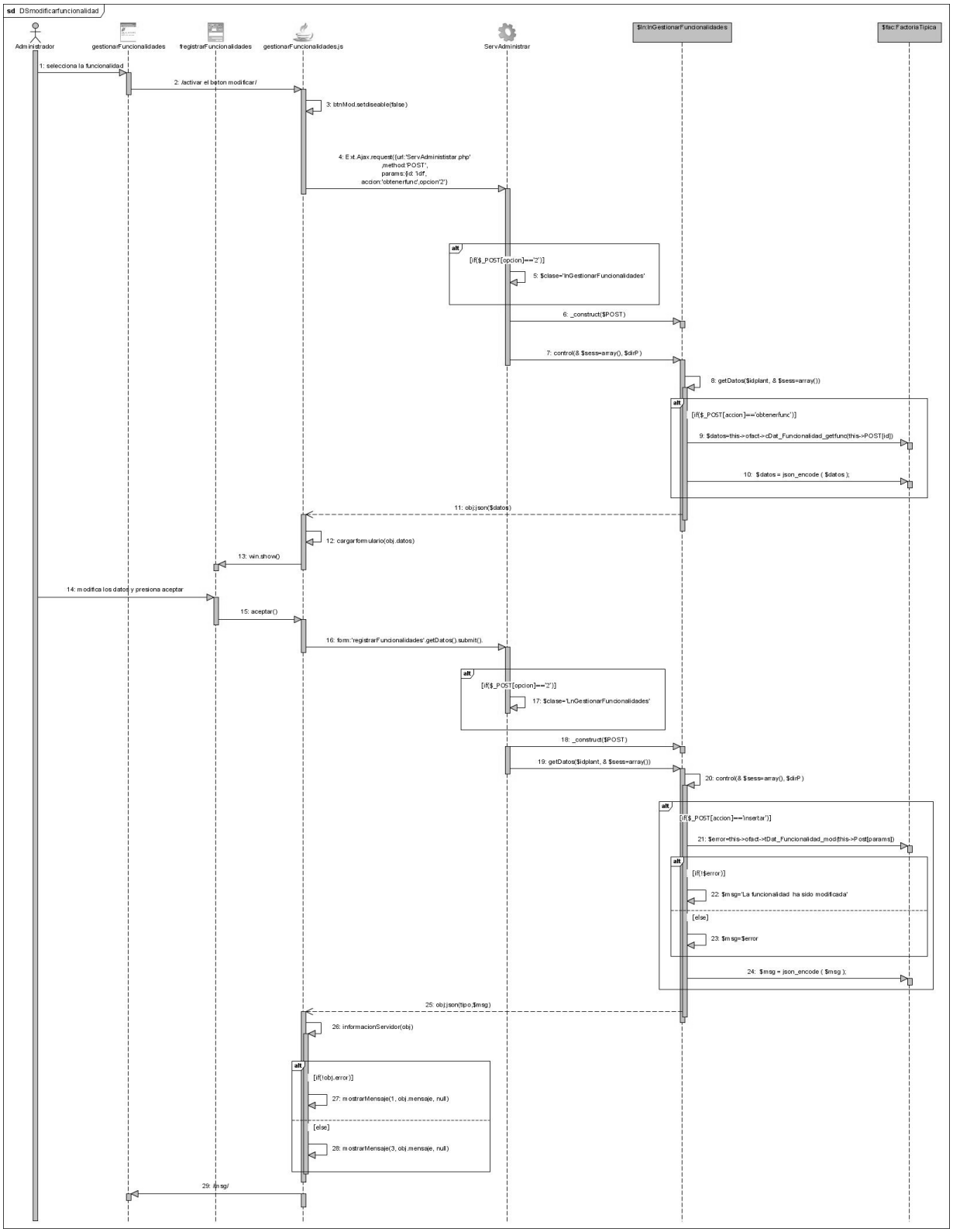


DIAGRAMA DE SECUENCIA MODIFICAR FUNCIONALIDAD

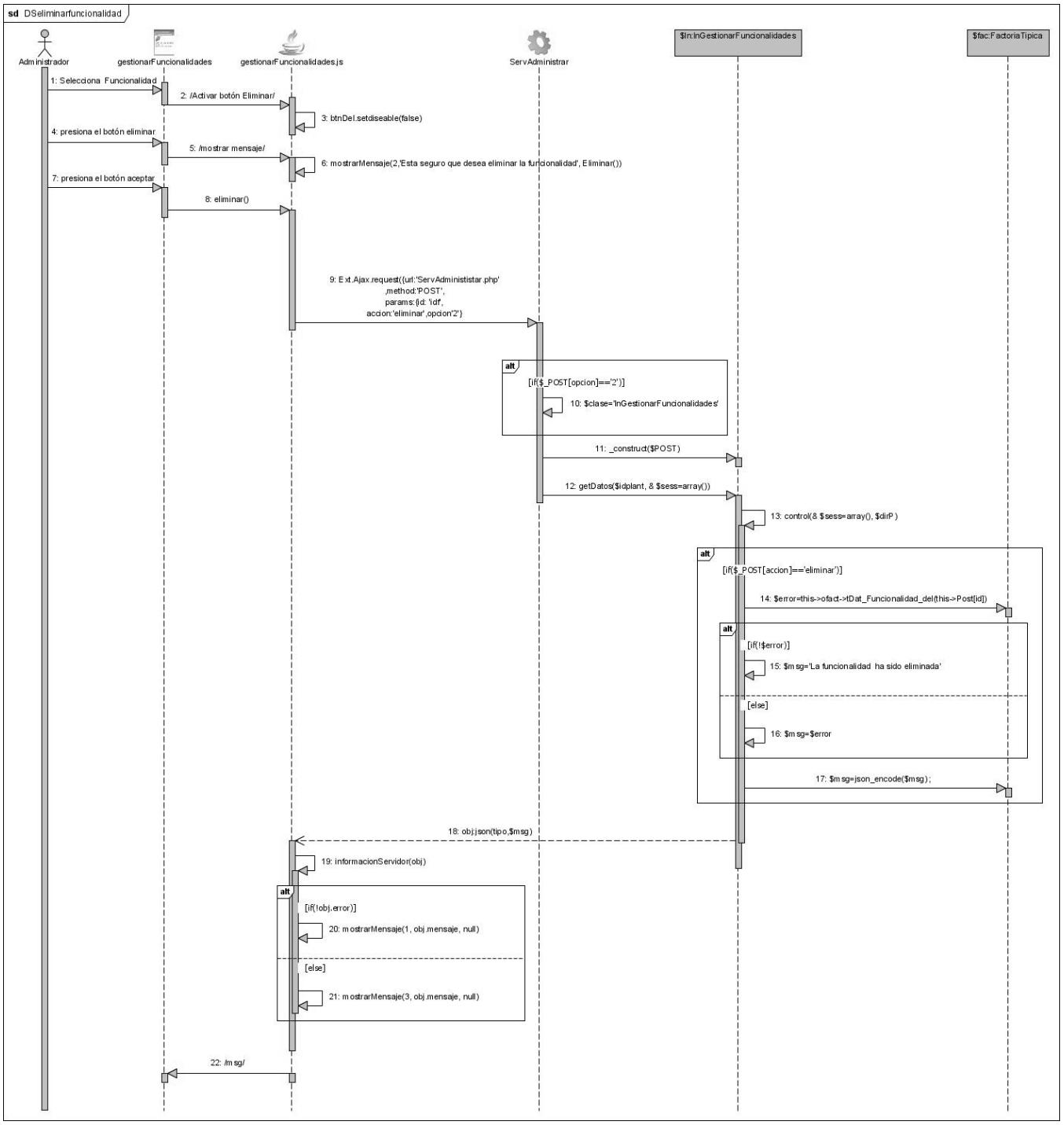


DIAGRAMA DE SECUENCIA ELIMINAR FUNCIONALIDAD

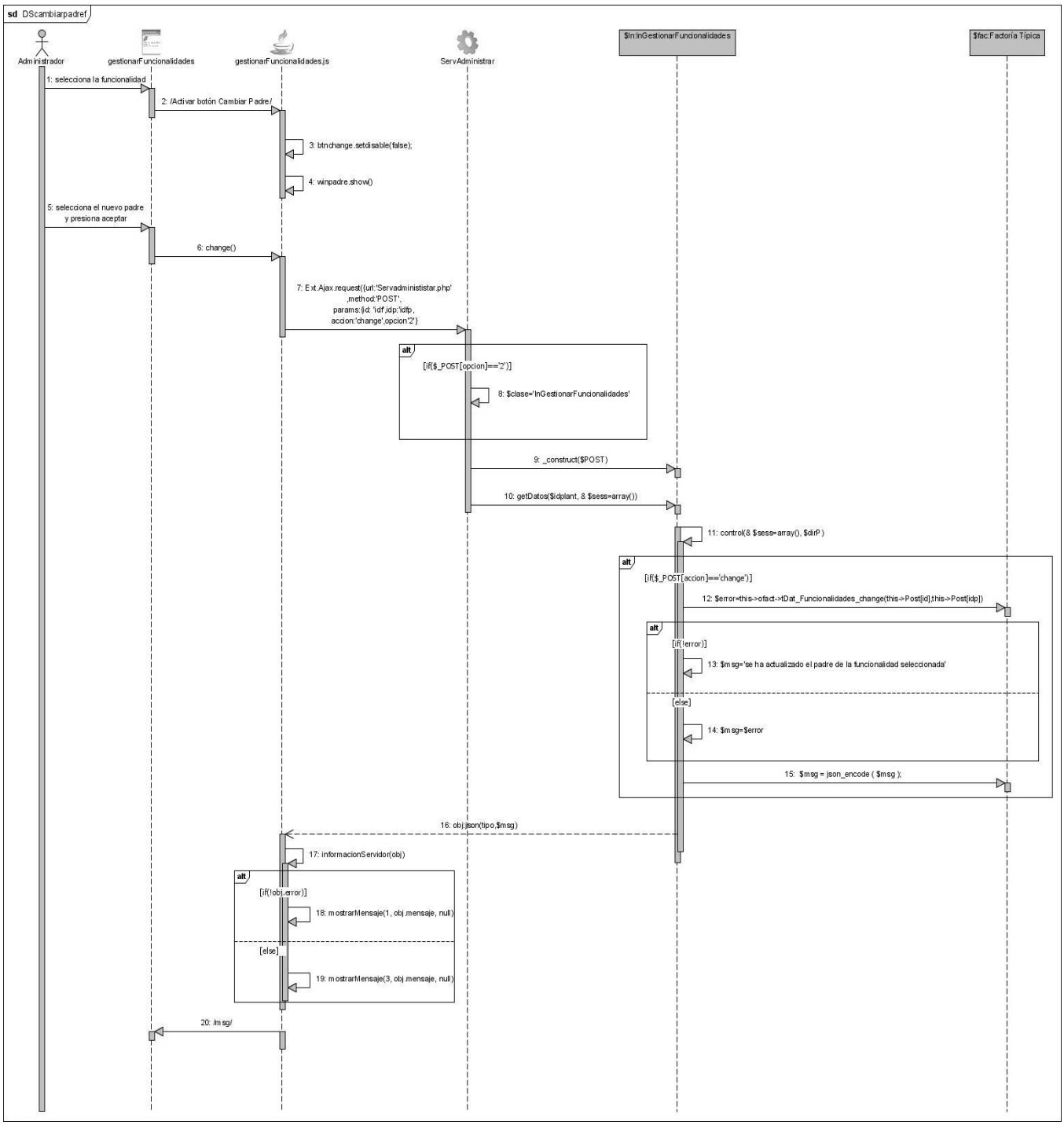


DIAGRAMA DE SECUENCIA CAMBIAR PADRE DE FUNCIONALIDAD

DIAGRAMAS DE SECUENCIA CU GESTIONAR ACCIONES

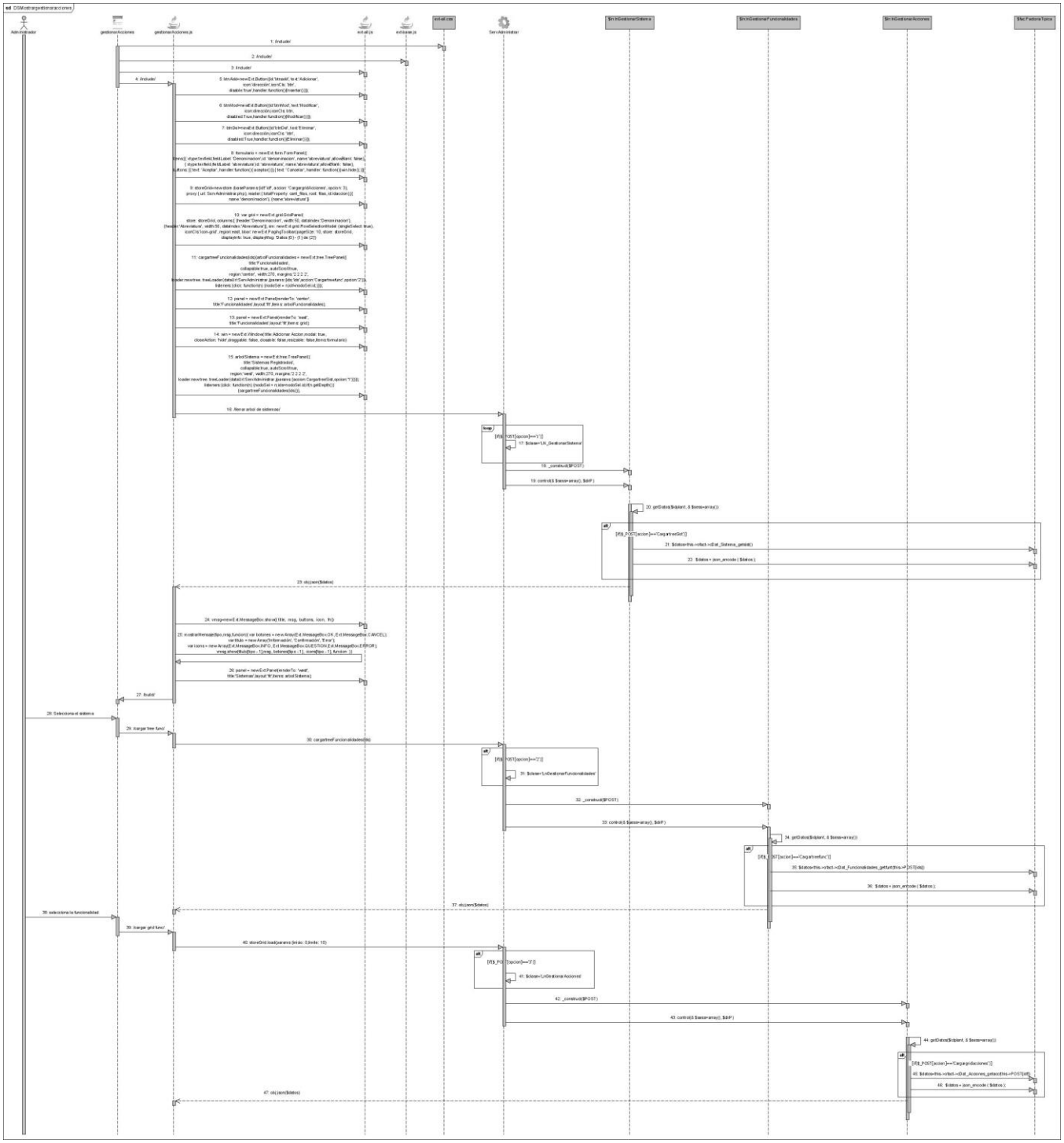


DIAGRAMA DE SECUENCIA MOSTRAR GESTIONAR ACCIONES

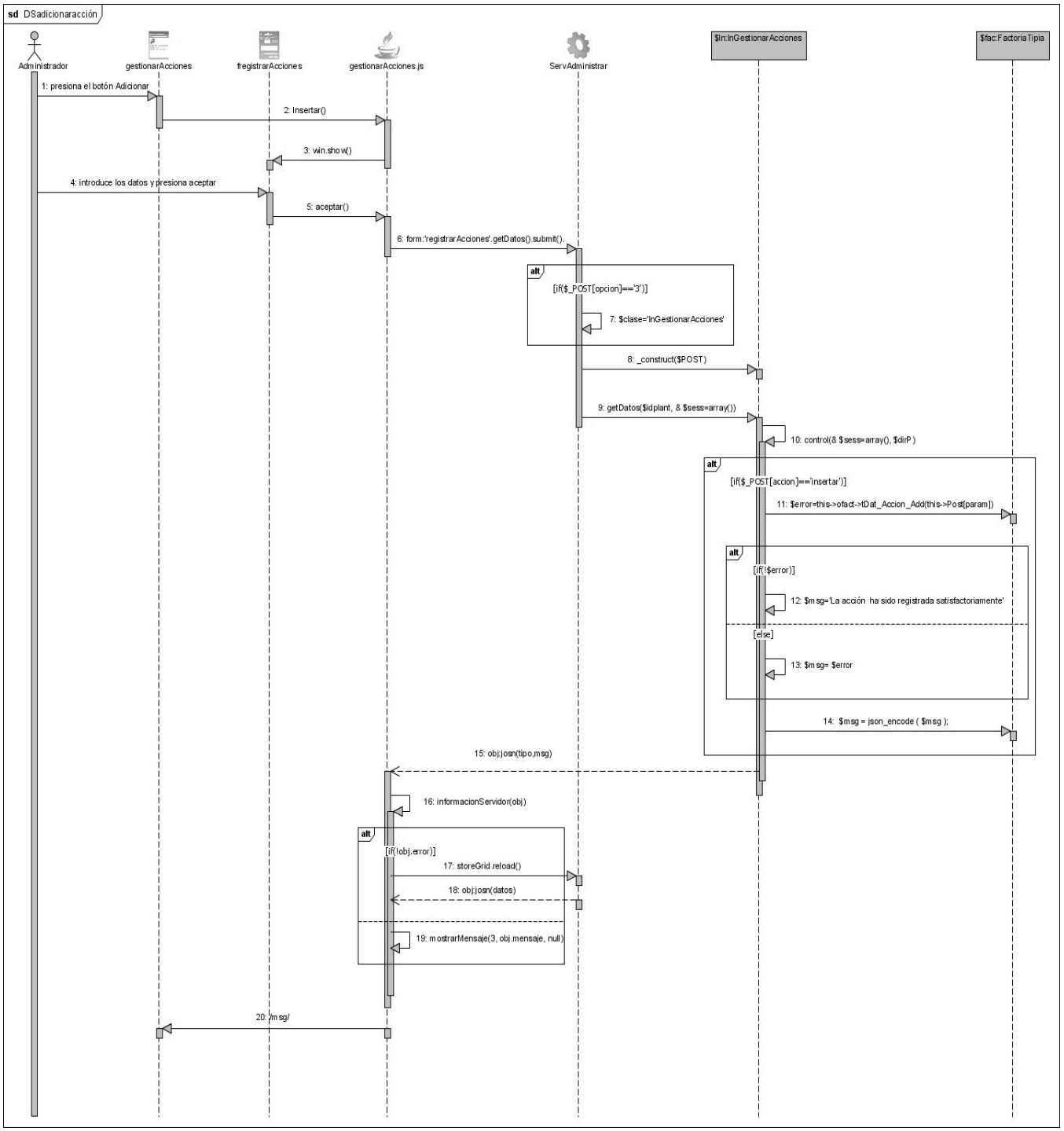


DIAGRAMA DE SECUENCIA REGISTRAR ACCIÓN

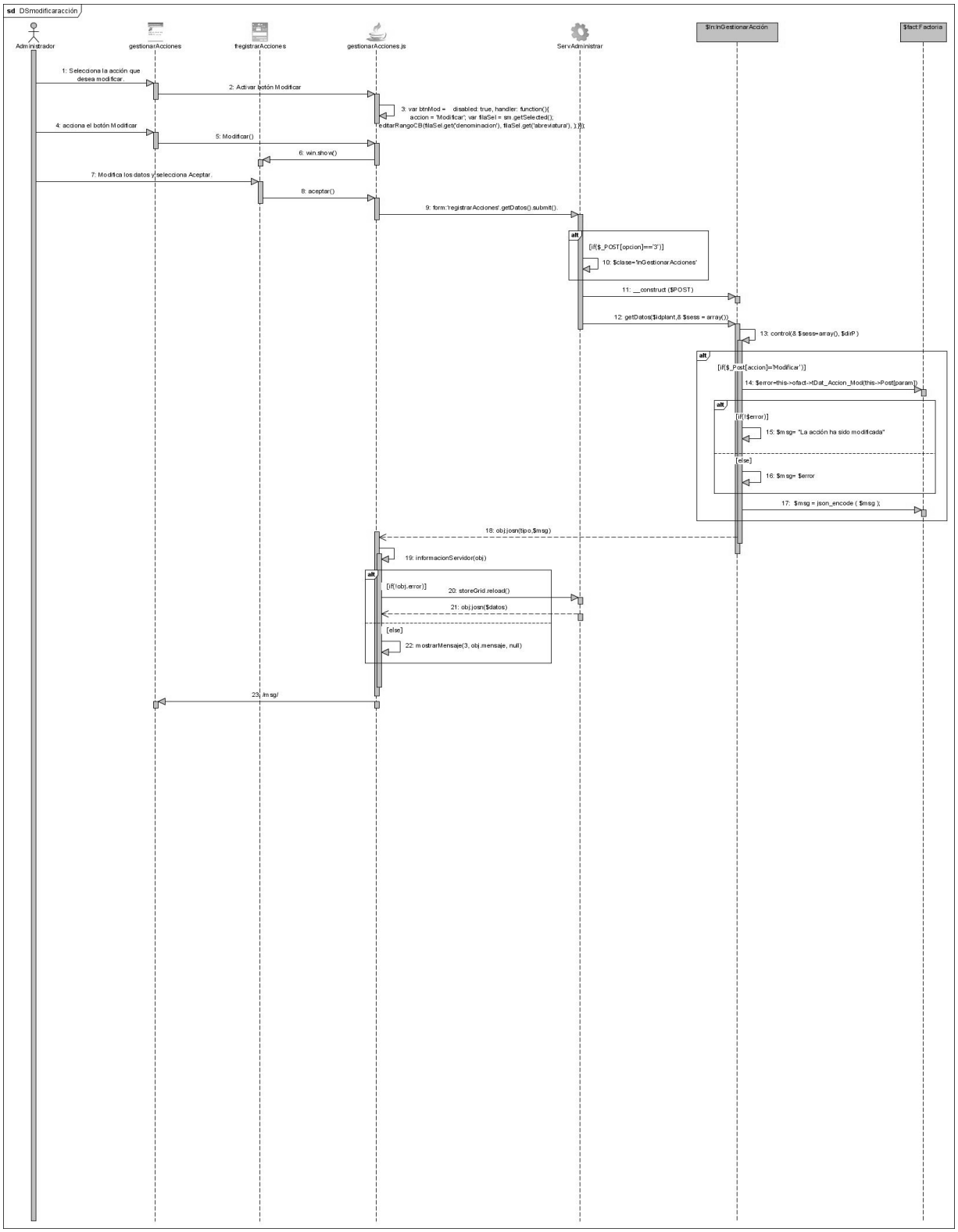


DIAGRAMA DE SECUENCIA MODIFICAR ACCIÓN

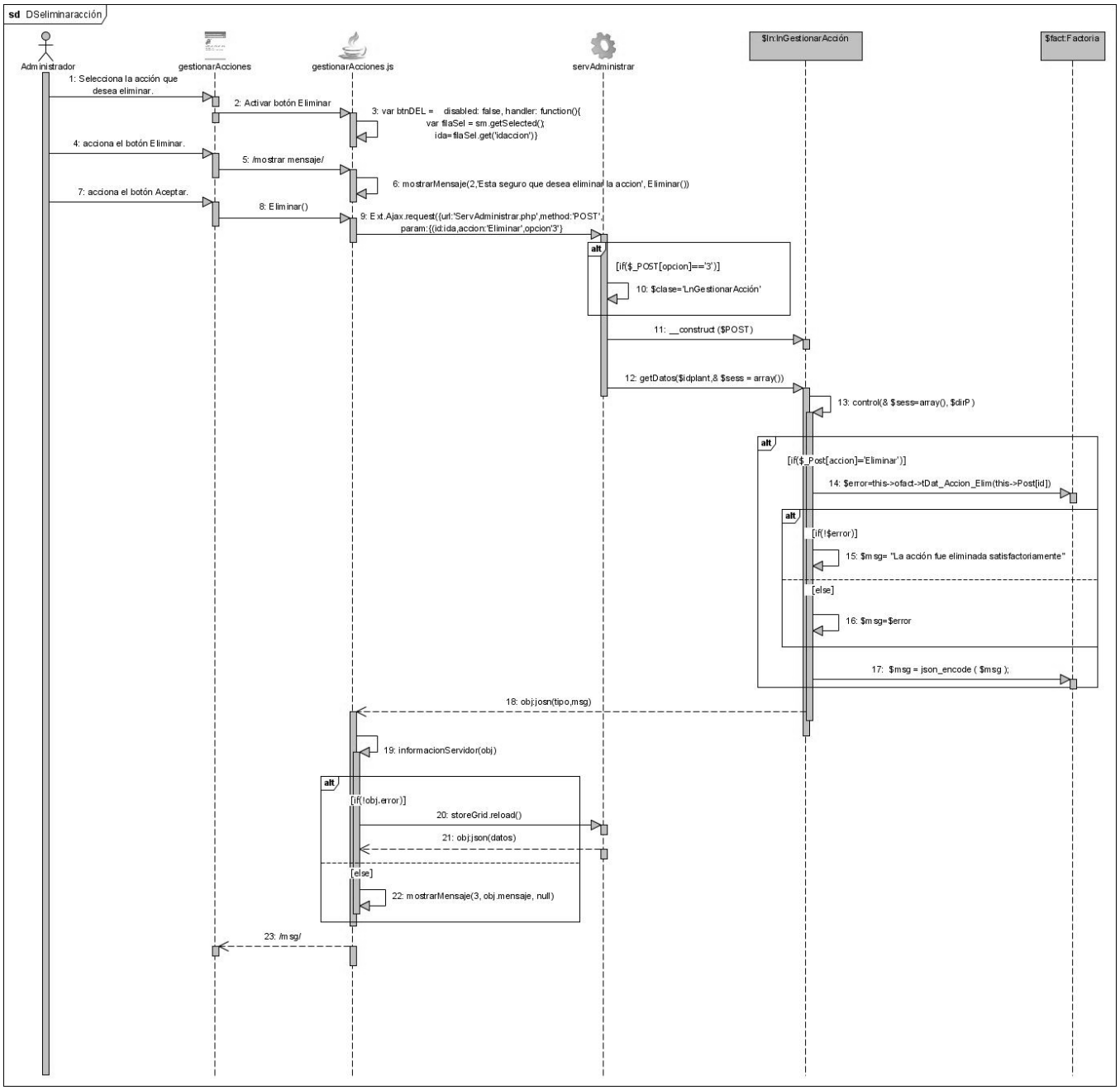


DIAGRAMA DE SECUENCIA ELIMINAR ACCION

DIAGRAMAS DE SECUENCIA CU GESTIONAR SERVICIOS QUE BRINDA

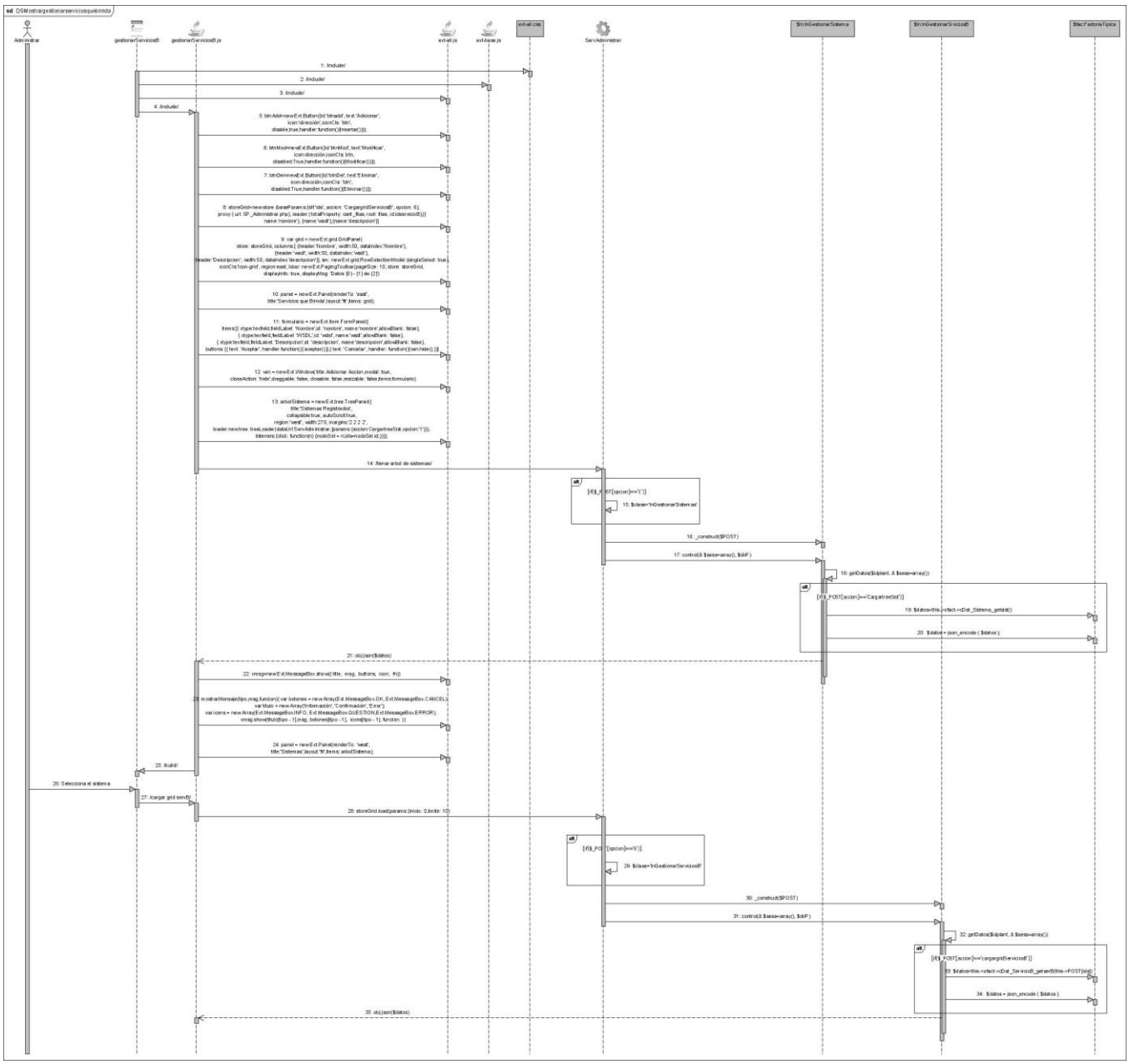


DIAGRAMA DE SECUENCIA MOSTRAR GESTIONAR SERVICIOS QUE BRINDA UN SISTEMA

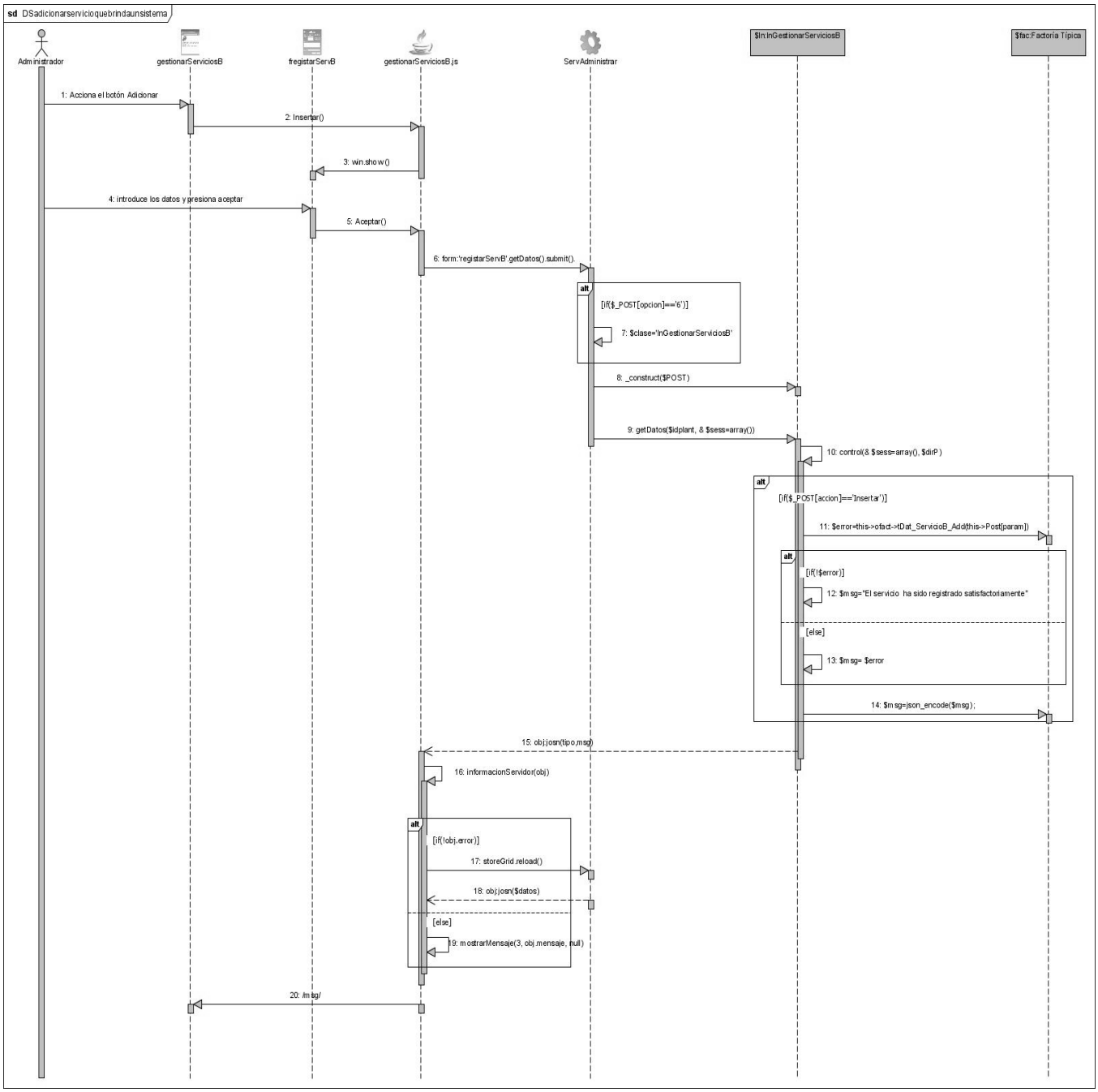


DIAGRAMA DE SECUENCIA REGISTRAR SERVICIOS QUE BRINDA UN SISTEMA

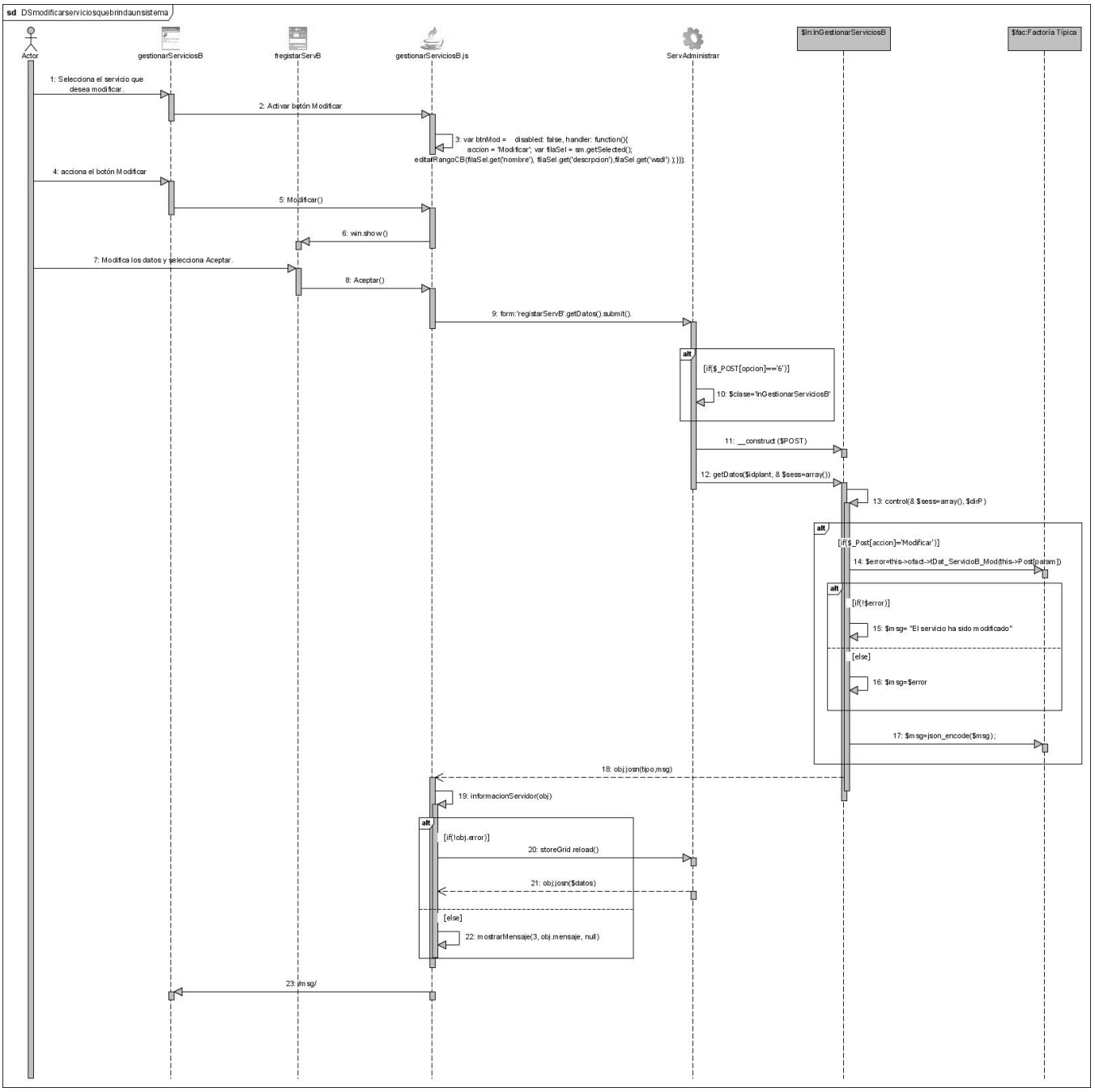


DIAGRAMA DE SECUENCIA MODIFICAR SERVICIO QUE BRINDA UN SISTEMA

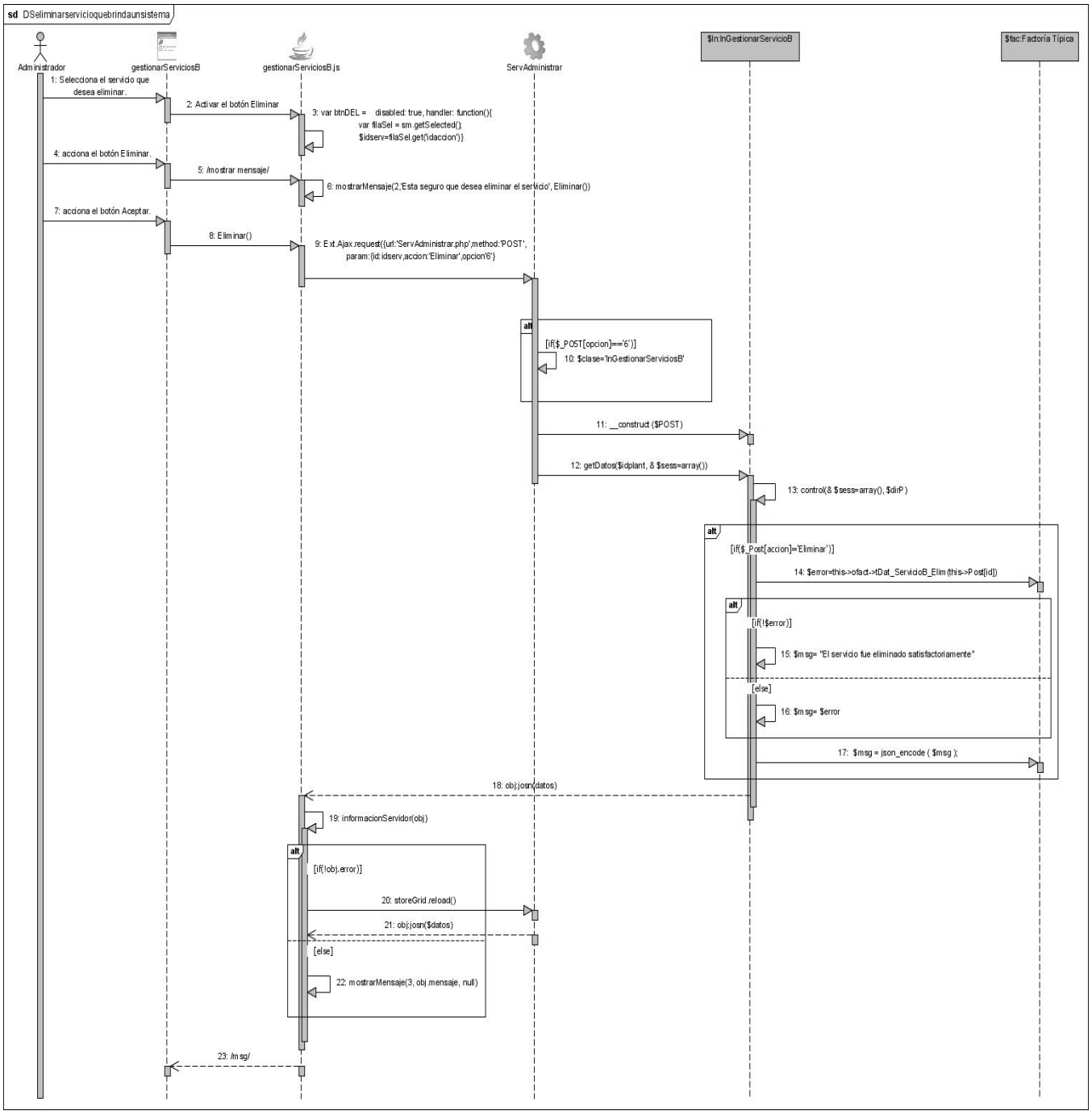
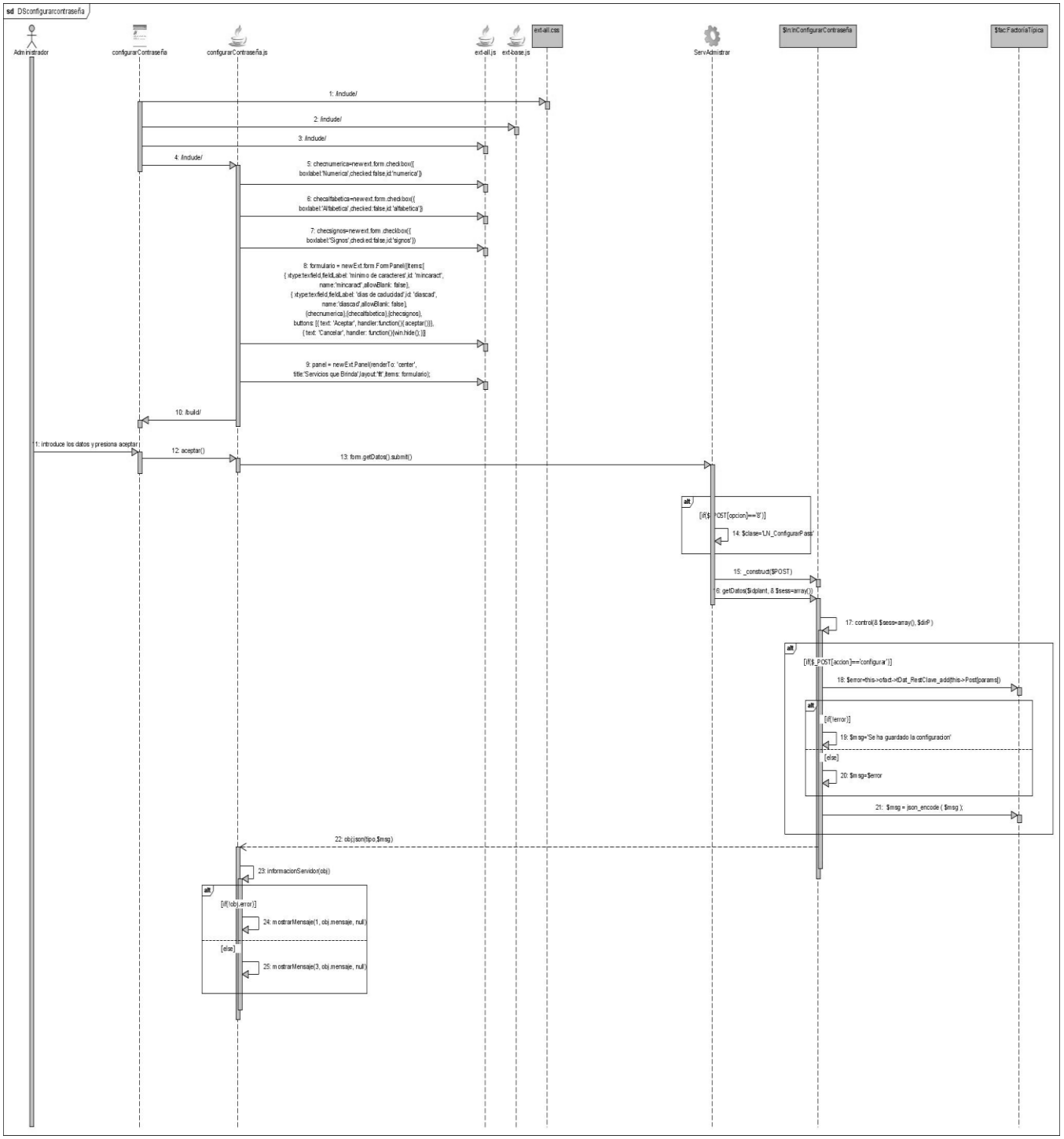


DIAGRAMA DE SECUENCIA ELIMINAR SERVICIO QUE BRINDA UN SISTEMA

DIAGRAMA DE SECUENCIA CU CONFIGURAR CONTRASEÑA



GLOSARIO DE TÉRMINOS

XML: Extensible Markup Language (Lenguaje extensible de etiquetas) Es un meta-lenguaje que permite definir lenguajes de marcado adecuado a usos determinados. Su función principal es describir datos y no mostrarlos.

HTML: HyperText Markup Language (Lenguaje de Marcado de Hipertexto) Lenguaje en el que se escriben las páginas a las que se accede a través de navegadores WWW. Admite componentes hipertextuales y multimedia.

HTTP: Protocolo usado para la transferencia de documentos WWW. Estas transferencias requieren un programa cliente http en un extremo de la comunicación y un servidor http en el otro.

Firewalls: Un cortafuegos es un elemento de hardware o software utilizado en una red de computadoras para controlar las comunicaciones, permitiéndolas o prohibiéndolas según las políticas de red que haya definido la organización responsable de la red.

FTP: File Transfer Protocol es un protocolo de transferencia de archivos entre sistemas conectados a una red TCP basado en la arquitectura cliente-servidor.

Json: JSON, acrónimo de "JavaScript Object Notation", es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML.

Scipts: El guión o archivo de órdenes o archivo de procesamiento por lotes (en inglés script) es un programa usualmente simple, que generalmente se almacena en un archivo de texto plano.

POST: El método POST se refiere normalmente a la invocación de procesos que generan datos que serán devueltos como respuesta a la petición. Se utiliza para mandar una gran cantidad de información al servidor.