Universidad de la Ciencias Informáticas

Facultad 9

Securitas

Sistema de Seguridad Centralizada de Aplicaciones Web

TRABAJO DE DIPLOMA PARA OPTAR POR EL

Título de Ingeniero en Ciencias Informáticas

AUTORES: Yadeny Aquino Jiménez

Laura M. Palomino Mariño

TUTOR: Ing. Yoenis Pantoja Zaldívar

Ciudad de La Habana, junio, 2007.

"Año 49 de la Revolución"

"- ¿Qué significa *habla*, *amigo* y *entra*? -preguntó Merry."

-Es bastante claro- dijo Gimli-. Si eres un amigo, dices la contraseña y las puertas se abren y puedes entrar.

-Sí -dijo Gandalf-, es probable que estas puertas estén gobernadas por palabras…

El Señor de Los Anillos

J.J.R . Tolkien

DECLARACIÓN DE AUTORÍA

Nosotras, Laura Maria Palomino Mariño y Yadeny Aquino Jiménez, declaramos que somos las únicas autoras de este trabajo y autorizamos a la Facultad 9 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

_____ _____

Laura Maria Palomino Mariño (Autor) Yadeny Aquino Jiménez (Autor)

_____

Ing. Yoenis Pantoja Zaldívar (Tutor)

DATOS DE CONTACTO

Síntesis del Tutor: Ing. Yoenis Pantoja Zaldívar

Jefe de la Disciplina de Programación

Facultad 9 UCI

Síntesis del Asesor: Ana Luisa Vigó Mitjans

Licenciada en Educación Especialidad Lengua Inglesa

Profesora Auxiliar

Facultad 9 UCI

OPINIÓN DEL TUTOR

Trabajo de Diploma Sistema de Seguridad Centralizada de Aplicaciones Web

Como tutor del Trabajo de Diploma “Sistema de Seguridad Centralizada de Aplicaciones Web”, luego de haber culminado la realización del mismo, considero que las autoras Laura María Palomino Mariño y Yadeny Aquino Jiménez, han desarrollado un conjunto de habilidades que les permitirán darle solución adecuadamente a cualquier tipo de necesidad de informatización que se presente en su vida profesional.

Durante la realización del presente trabajo las estudiantes han demostrado un alto grado de responsabilidad ante el cumplimiento en tiempo de las tareas que se les programaron. Han trabajado organizadamente dando muestras de poseer responsabilidad y compromiso en la realización de su tesis. Su desempeño ha manifestado que han desarrollado un valioso nivel de asimilación de nuevas metodologías, llegando a alcanzar un profundo conocimiento y una gran capacidad para la toma de decisiones correctas.

La originalidad, la elegancia en el trabajo y la independencia, han sido cualidades dignas de destacar a lo largo de la realización del trabajo realizado. Las estudiantes manifestaron laboriosidad a lo largo del cumplimiento de las tareas programadas, logrando resultados satisfactorios a pesar de disponer de poco tiempo de desarrollo.

Por otra parte, el elemento investigativo del documento, estuvo desde el inicio muy bien orientado y estructurado, basado en una gran cantidad de bibliografía actualizada. Cada contenido se ha expuesto con claridad y aporta grandes conocimientos al lector.

Por todo lo anteriormente planteado, considero que las diplomantes están aptas para ejercer como Ingenieras Informáticas; y propongo al tribunal que se le otorgue al Trabajo de Diploma la máxima calificación.

Tutor: Ing. Yoenis Pantoja Zaldívar

AGRADECIMIENTOS

A nuestro Tutor Ing. Yoenis Pantoja por su ayuda en todos los momentos que lo necesitamos.

A nuestro amigo Andrés por su apoyo incansable y colaboración.

A nuestros padres por su grano de maiz y por estar siempre presentes.

A todas esas personas que de una forma u otra ayudaron en nuestra formación profesional a lo largo de estos 5 años.

A nuestros compañeros de aula que siempre nos brindaron apoyo ante el estudio y por haber compartido juntos estos lindos años.

A los amigos, los mas cercanos, y los que están lejos también, gracias.

DEDICATORIA

A nuestros padres y demás familiares, por entregarnos su amor y dedicación, su apoyo en todo momento y por habernos encaminado de forma correcta en la vida.

RESUMEN

El progresivo desarrollo informático y tecnológico ha acrecentado consigo los riesgos en un medio donde prácticamente cualquier intercambio ocurre con la intervención de algún sistema de cómputo: -La Universidad de Ciencias Informáticas (UCI)-, protagonista de estos avances en nuestro país, sin dudas no escapa a la necesidad de preservar la información.

La Facultad 9, pieza de ésta institución, desarrolla un interesante número de aplicaciones como parte del proceso de digitalización que se lleva a cabo actualmente, enfocado en facilitar el trabajo del personal y optimizar el aprovechamiento de las tecnologías con que se cuenta.

Cada sistema realizado, o en proyecto, implementa su seguridad de forma independiente, lo que conlleva a diferencias en las vulnerabilidades y a su vez un aumento de estas. Por lo tanto, surge la necesidad de desarrollar un sistema genérico capaz de proveer servicios de seguridad a las aplicaciones web de la facultad de forma centralizada, logrando mayor integridad, fácil mantenimiento en términos de seguridad, uniformidad, así como promover la reutilización de código, solucionando de esta forma muchos problemas administrativos.

El presente trabajo consta de un estudio detallado del arte, análisis de las principales herramientas y tecnologías propuestas para el desarrollo del software. Contiene toda la documentación UML incorporada a la solución presentada, el estudio de factibilidad y los beneficios asociados a la puesta en marcha del sistema propuesto.

PALABRAS CLAVES

- Seguridad
- Aplicaciones web
- Servicios web
- Sistemas de seguridad

ABSTRACT

The progressive computer and technological development has increased because of the risks in a means where practically any exchange happens with the intervention of some computation system: - The University of Computer Sciences (UCI)-, as the main character of these advances in our country, without doubts it doesn't escape to the necessity from preserving the information.

The School 9, piece of this institution, develops an interesting number of applications like part of the digitization process that it is carried out at the moment, focused in to facilitate the personnel's work and to optimize the use of the technologies with which it is counted.

Each carried out system, or in project, it implements their security in an independent way, what bears to differences in the vulnerabilities and in turn an increase of these. Therefore, the necessity arises of developing a generic system able to provide services of security to the applications web of the School in a centralized way, achieving a bigger integrity, easy maintenance in terms of security, uniformity, as well as to promote the code reutilization, solving this way many administrative problems.

The present work consists of a detailed study of the art, analysis of the main tools and technologies proposed for the development of the software. It contains the whole documentation incorporate UML to the presented solution, the study of feasibility and the benefits associated to the start of the proposed system.

KEYWORDS

- Security
- Web applications
- Web services
- Security systems

TABLA DE CONTENIDOS

ÍNDICE DE TABLAS

Capitulo 3

Anexos

ÍNDICE DE FIGURAS

Capitulo 2

Capitulo 3

Capitulo 4

Anexos

INTRODUCCIÓN

El vertiginoso desarrollo alcanzado en las nuevas tecnologías de la informática y las comunicaciones ha llevado a la sociedad a entrar en lo que se ha dado en llamar “era de la información”. La información puede existir de muchas formas, impresa o escrita en papel, almacenada digitalmente, transmitida por correo postal o utilizando medios digitales, presentada en imágenes o expuesta en una conversación. Cualquiera sea la forma que adquiere la información es un recurso que, como el resto de los activos importantes tiene un gran valor. (1)

En la actualidad, prácticamente todos los sistemas de cómputo se hayan conectados, enviando y recibiendo información generalmente valiosa, que de ser manipulada podría significar cuanto menos pérdida en términos económicos.

La evolución de la informática y las telecomunicaciones ha traído numerosos beneficios pero también aumento de los riesgos, dado que existen personas y entidades que se dedican a perjudicar la integridad y confidencialidad de los datos que viajan por la red. Por lo tanto, paralelo a este desarrollo es necesario llevar a cabo métodos cada vez mas eficaces para dar seguridad a las aplicaciones y con ello proteger los intereses de quienes confían en la Web como un medio de comunicación e intercambio.

En la Universidad de las Ciencias Informáticas (UCI) la información es un recurso con valor incalculable, por significar la “vida” de la universidad; es por eso que debe ser debidamente protegida teniendo en cuenta que el centro tiene una amplia conexión interna y acceso pleno a internet por la mayoría del personal.

Existen actualmente en la institución, numerosos procesos ya automatizados y otros por automatizar que contienen importantes flujos de información ya sea docente, científico o laboral, siendo necesario restringir acceso a los mismos y preservar la información.

La facultad 9 se encuentra desarrollando varios sistemas con el objetivo de facilitar el trabajo y hacer un mejor uso de las tecnologías con que cuenta mediante proyectos e investigaciones científicas.

Una de estas investigaciones se refiere al desarrollo de un sistema genérico proveedor de servicios de seguridad a las aplicaciones web de forma centralizada, cuyo objetivo es lograr uniformidad en este aspecto y promover la reutilización de código.

De manera general la situación problémica se enuncia en los siguientes puntos:

- La implementación de sistemas web en la facultad conlleva al desarrollo repetitivo de un módulo de seguridad, cuyas funcionalidades son semejantes en cada caso, esto implica considerable aumento y variedad de las vulnerabilidades cuando debería fomentarse la concentración de esfuerzos para lograr un sistema centralizado que implemente alta seguridad.
- Cada sistema debe ser atendido por un administrador, cuando, utilizando los servicios del dominio UCI y la Web, pudiera centralizarse.
- A la hora de actualizar contra ataques, se hace necesario hacerlo para cada uno de los sistemas, mientras que de la forma que se propone solo deberá hacerse para uno: el centralizado.
- No se reutiliza el código y no se estandariza el término de seguridad informática para las aplicaciones web dentro de la misma facultad.

La importancia de una investigación con temáticas asociadas a seguridad y el aporte de un sistema centralizado que asegure los recursos informáticos de la facultad se justifican con los siguientes beneficios:

- Los usuarios de los sistemas auditados se sentirían mas confiados al saber que existe un software especializado que brinda servicios de seguridad de manera organizada, logrando mantener la integridad y confidencialidad de la información que a ellos interese.
- Aporte de una investigación minuciosa acerca de temáticas tan importantes como la seguridad informática, recopilando importantes conceptos a tener en cuenta y haciendo un estudio de sistemas de seguridad.
- Como un aporte tecnológico se puede considerar el modelado del sistema propuesto que a su vez posee valor de interés social, por la misión protectora que encapsula el mismo.
- El carácter de los datos que pueda proteger el sistema propuesto conlleva a un alcance superior en cuanto a la importancia del mismo. Los datos de carácter científico, docente y laboral requieren de alto nivel de protección.

No se podría dejar de mencionar la existencia en Cuba y la UCI de sistemas de seguridad con este mismo propósito, los cuales han servido de apoyo a esta investigación. No obstante, éstos no resuelven el **problema actual**, el cual se expone de la siguiente forma*:*

¿Cómo centralizar la seguridad en las diferentes aplicaciones web de la Facultad 9?

Asimismo, el **objeto de estudio** lo constituyen *los sistemas proveedores de seguridad para aplicaciones web* y el **campo de acción** *el proceso de seguridad en los sistemas de la facultad 9*.

Se persigue con esta investigación lograr el siguiente **objetivo general:** *Modelar un mecanismo genérico de seguridad para las aplicaciones web de la Facultad 9*.

Para lo que se proponen las siguientes actividades a realizar**:**

1. *Recopilación de información acerca de seguridad informática y seguridad en sistemas web.*
2. *Recopilación, estudio del arte de las principales y más actuales tecnologías utilizadas para el diseño y desarrollo de aplicaciones de seguridad.*
3. *Estudio de lo planteado en los estándares nacionales e internacionales para el desarrollo de aplicaciones web.*
4. *Entrevistas a personal experimentado en seguridad en la UCI.*
5. *Confección de la documentación completa UML*
6. *Diseño de la Base de datos.*
7. *Análisis y diseño del sistema genérico de seguridad.*

Métodos de investigación científica utilizados**:**

Métodos Teóricos:

Histórico lógico: Permite el estudio acerca de la seguridad informática desde el punto de vista histórico.

Análisis y la síntesis: Se realiza una síntesis de toda la bibliografía consultada y analizada para el desarrollo del trabajo.

Modelación: Se modelan una serie de diagramas para el diseño de la aplicación web que nos permite abstraernos a la realidad.

Métodos Empíricos:

Entrevistas: Se realizan entrevistas a personal especializado en el tema para profundizar en el mismo y dar solución al problema a resolver.

Análisis de documentos: Se realiza la revisión de documentos utilizados para la investigación.

Como **resultados esperados** de la investigación se tienen:

1. *Estado del arte de las principales y más actuales tecnologías utilizadas para el diseño y desarrollo de aplicaciones de seguridad en Cuba y en el mundo.*
2. *Documentación completa UML del sistema de seguridad.*
3. *Diseño de la Base de datos.*

Idea a defender:

La concepción de un Sistema de seguridad centralizada de aplicaciones web para la facultad 9 es una aproximación a la solución de problemas administrativos que surgen en el área de seguridad en sistemas web trayendo como beneficios tangibles el aseguramiento de información de manera similar en los sistemas web, enfoque unidireccional e integración corporativa.

El presente trabajo se estructura en 5 capítulos descritos a continuación:

Capítulo 1. Fundamentación teórica: Se exponen conceptos asociados a seguridad informática y seguridad en sistemas web. Se visualizan los principales problemas que motivan el desarrollo de esta investigación. Se especifican otros sistemas de seguridad existentes en el Mundo, Cuba, y la UCI.

Capitulo 2. Tendencias y tecnologías actuales a considerar: En este capitulo se hace un estudio de las principales tendencias actuales para resolver problemas de seguridad, así como la fundamentación de la elección que se hizo para darle solución efectiva a la problemática planteada en el capitulo anterior.

Capítulo 3. Descripción de la solución propuesta: En este capitulo se expone la solución propuesta para asegurar los recursos de la facultad mediante un modelo del dominio y se describe el futuro sistema mediante los artefactos requeridos.

Capítulo 4. Construcción de la solución propuesta: Se tratan otros aspectos relacionados con la solución que se propuso anteriormente, se modela el sistema mediante diagramas de clases, implementación y despliegue, así como el diseño de la base de datos asociada.

Capítulo 5. Estudio de Factibilidad: Se abordan los aspectos relacionados con el estudio realizado de la factibilidad, la planificación previamente realizada, los costos del proyecto, y los beneficios del mismo ya sean tangibles o no.

CAPÍTULO 1 Fundamentación Teórica.

1.1 Introducción

Diseñar sistemas seguros es una tarea compleja, pues las amenazas y los ataques son, en muchos casos, poco cuantificables y muy variados. La aplicación de medidas de seguridad para proteger un sistema supone un análisis y cuantificación previa de los riesgos o vulnerabilidades del sistema (2).

En este capitulo se hace un estudio de los principales conceptos a tener en cuenta a la hora de diseñar sistemas con propósitos de asegurar información. Se plantean de manera detallada las dificultades esenciales que resuelve un sistema centralizado de seguridad. Se citan otros sistemas y empresas que se dedican a la búsqueda de mecanismos de perfeccionamiento de la seguridad.

1.2 Elementos asociados a la seguridad de sistemas

1.2.1 De Seguridad informática

Se puede definir *seguridad informática* como un conjunto de métodos y herramientas destinados a proteger los bienes informáticos de una institución (3).

Según estudios realizados, este término está íntimamente relacionado con los siguientes aspectos en cualquier sistema de cómputo:

- *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. (3).

Especialistas en el tema consideran que un sistema puede decirse seguro si establece un balance entre estos tres conceptos. Sin embargo, un sistema puede mantener la confidencialidad si nadie tiene acceso a

él y sería un sistema bastante seguro, pero viola definitivamente este balance, pues tendrá, sin lugar a dudas, una disponibilidad nula. Es por ello, la importancia de implementar un apropiado sistema de controles, que pudieran ser políticas, prácticas, procedimientos, estructuras organizacionales y funciones de software.

Un sistema de computación tiene tres componentes fundamentales a proteger: hardware, software y datos. Sin embargo, se consideran los datos como el activo informático más preciado para cualquier institución.

Se ha descrito el valor de la información como superior al del hardware y software, pues pese a que estos son bastante caros se podrían reponer en caso de pérdida; sin embargo, determinada información podría ser muy difícil de recuperar.

Se propone como un mecanismo de análisis de seguridad en un sistema, el estudio de las principales formas en las que este pudiera sufrir perdidas o daños, y como resultado obtener las debilidades de dicho sistema.

Una *vulnerabilidad* es una debilidad en el sistema de seguridad, por ejemplo, en procedimientos, diseños e implementaciones que pueden ser explotados para causar algún daño. Una *amenaza* a un sistema de computación es un grupo de circunstancias que tienen el potencial para causar algún daño o pérdida. Alguien que explote una vulnerabilidad estaría realizando un *ataque* contra un sistema (3).

Para evitar los ataques se usan los denominados controles o mecanismos de defensa como medida de protección. Un *control* es una acción, dispositivo o procedimiento que elimina o reduce una vulnerabilidad (3).

Como resultado de este epígrafe se describe la relación entre los principales conceptos que se han establecido. Se podría decir que una amenaza puede convertirse en ataque si afectara la confidencialidad, disponibilidad o integridad de la información y es posible bloquearla si se aplican controles a las vulnerabilidades de los sistemas.

1.2.2 De ataques informáticos.

Cuando se materializan las amenazas estamos en presencia de un *ataque* (2). Existen dos tipos de ataques: *pasivos* y *activos.*

Pasivos: No se altera la comunicación, sino que únicamente se escucha o monitoriza para obtener información que está siendo transmitida. Su objetivo es el análisis de tráfico. Los ataques pasivos son muy difíciles de detectar ya que no provocan ninguna alteración de los datos.

Activos: Implican algún tipo de modificación del mensaje transmitido o la creación de un falso mensaje (4).

Estos tipos de ataques pueden a su vez agruparse de la siguiente forma:

- *Intercepción*: Acceso a la información por personas no autorizadas. Es un ataque de tipo pasivo contra la confidencialidad de la información. Un ejemplo de este tipo de ataque es el robo de contraseñas o la copia ilícita de programas.
- *Modificación*: Acceso no autorizado a la información en el que se produce una modificación de la misma. Es un ataque activo contra la integridad de los datos. La desfiguración de un sitio web es un ejemplo de este tipo de ataque.
- *Interrupción*: Deja de funcionar total o parcialmente un sistema informático. Es un ataque activo contra la disponibilidad de la información. Un ejemplo del mismo sería el bloqueo de los servicios de servidores web o de correo electrónico (3).
- *Fabricación*: Inserción de objetos falsificados en el sistema. Este es un ataque activo contra la autenticidad. Ejemplo de este ataque es cuando se insertan mensajes ilegítimos en una red o cuando se añaden registros a un archivo (4).

1.2.3 Control de acceso. Mecanismos.

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.

Un *objeto* es la entidad que provee o contiene la información o los datos, puede ser un fichero, una base de datos, una computadora, un programa, una impresora o un dispositivo de almacenamiento (1).

Se plantea que los roles de sujeto y objeto pueden cambiarse en dos entidades que se comunican para cumplir una tarea. Ejemplo de entidades intercambiables son un programa con una base de datos, un proceso y un fichero.

El siguiente concepto propuesto de control de acceso está más orientado al campo informático y se plantea como el proceso de conceder permisos a usuarios o grupos de acceder a objetos, tales como ficheros o impresoras en la red. El control de acceso está basado en tres conceptos fundamentales: identificación, autenticación y autorización.

Se denomina *identificación* a la acción por parte de un usuario de presentar su identidad a un sistema, usualmente se usa un identificador de usuario. Establece además que el usuario es responsable de las acciones que lleve a cabo en el sistema. Esto está relacionado con los registros de auditorías que permiten guardar las acciones realizadas dentro del sistema y rastrearlas hasta el usuario autenticado.

Autenticación: Es la verificación de la validez del usuario y su contraseña, ya que usualmente se implementa una contraseña para iniciar una sesión. Existen cuatro tipos de técnicas que permiten realizar la autenticación de la identidad del usuario, las cuales pueden ser utilizadas individualmente o combinadas (autenticación de varios factores):

1. Algo que solamente el individuo conoce: Por ejemplo una contraseña.
2. Algo que la persona posee: Por ejemplo una tarjeta magnética.
3. Algo que el individuo es y que lo identifica unívocamente: Por ejemplo las huellas digitales.
4. Algo que solo el individuo es capaz de hacer: Por ejemplo los patrones de escritura.

Autorización: En este proceso se comprueba que los usuarios con identidad válida tengan entrada solamente a aquellos recursos a los que tienen derechos de acceso. En otras palabras, la autorización es una simple revisión que se ejecuta en todas las fases del procesamiento de solicitudes en el servidor web. De esta forma se garantiza que únicamente se acceda a los recursos permitidos (1).

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 solo a la información que necesitan para su trabajo o la que por jerarquía tienen derecho a obtener.

1.2.3.1 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)* (1).

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:

- *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.
- *Formas de almacenar las contraseñas*: Se debe usar un algoritmo criptográfico irreversible (o función resumen), los más comunes son MD5 y SHA1.
- *Método de retransmisión de la contraseña*: Deben ser transmitidas mediante algún método criptográfico, en el caso de las redes locales se usa con mucha frecuencia Kerberos.
- *Longevidad de la contraseña*: Deben ser cambiadas con cierta periodicidad (1).

1.2.3.2 Tokens de seguridad o autenticación

Un *token de seguridad* es un objeto físico, o sea, un pequeño dispositivo de hardware que los usuarios cargan consigo para autorizar el acceso a un servicio de red. El dispositivo puede ser en forma de una tarjeta inteligente o puede estar incorporado en un objeto utilizado comúnmente, como un llavero. Existe más de una clase de token de autenticación, están los bien conocidos generadores de contraseñas dinámicas (*one time password*) y los que comúnmente se denominan *tokens USB*, los cuales no sólo permiten almacenar contraseñas y certificados digitales, sino que permiten llevar la identidad digital de la persona.

Los *token* proveen un nivel de seguridad adicional utilizando el método conocido como *autenticación de dos factores*: el usuario tiene un número de identificación personal (PIN), que le autoriza como el propietario del dispositivo; luego el dispositivo despliega un número que identifica en forma única al usuario ante el servicio, permitiéndole ingresar (1).

1.2.3.3 Autenticación biométrica

Se basa en rasgos personales distintivos con capacidad de identificar a una persona. Se clasifican en:

- *Fisiológicos*: Huella dactilar, iris, retina, cara, geometría de la mano, huella palmar, estructura de la venas, estructura de la oreja, termografía facial.
- *Conductuales*: Voz, escritura, firma manuscrita, modo de teclear, modo de andar (1).

Del estudio de este epígrafe se puede resumir que no existe un mecanismo de acceso que defina un balance apropiado entre seguridad y conveniencia aunque sí definitivamente algunos de ellos lo suavizan, sin embargo son mucho más caros en términos económicos. Se sitúan entre estos la autenticación biométrica de escritura-firma, voz y geometría de la mano.

1.2.4 Control de acceso. Modelos

Como ya anteriormente se expuso la *autorización* es el mecanismo para establecer si el usuario o proceso preliminarmente identificado y autenticado tiene permitido el acceso a determinados recursos. Se implementa con uno de los siguientes modelos de control de acceso.

1.2.4.1 Control de Acceso Obligatorio (MAC)

En este modelo es el sistema quien protege los recursos, comparando las etiquetas del sujeto que accede frente al recurso accedido, o sea, la autorización para que un sujeto acceda a un objeto depende de los niveles de seguridad que tengan, ya que estos indican qué permiso de seguridad tiene el sujeto y el nivel de sensibilidad del objeto (1).

Todos los sujetos y objetos del sistema tienen una etiqueta de seguridad que se compone de:

- Una clasificación o nivel de seguridad como un número en un rango, o un conjunto de clasificaciones discretas (Desclasificado, Confidencial, Secreto y Sumamente Secreto).
- Una o más categorías o compartimentos de seguridad como Contabilidad, Ventas, I+D... etc.

Lo anterior se conoce como política de seguridad multinivel pues sigue el modelo de clasificación de la información militar donde la confidencialidad de la información es lo más relevante. Por ejemplo, los militares clasifican los documentos en Confidencial, Secreto y Sumamente Secreto (1).

Análogamente un usuario puede obtener permisos de seguridad Confidencial, Secreto y Sumamente Secreto y con éstos acceder a objetos archivados con niveles iguales o inferiores a los permisos. O sea, si el permiso de seguridad que se le asignó fuera Sumamente Secretos, puede acceder a todos los recursos clasificados como Secretos Confidenciales, y Sumamente Secretos. En cambio, si el permiso de seguridad que adquiere es Confidencial solo podrá acceder a los recursos de tipo Confidencial.

1.2.4.2 Control de acceso Discrecional (DAC)

El modelo DAC se ha venido usando profusamente en sistemas operativos de propósito general y en virtualmente todos los sistemas de bases de datos y sistemas de comunicaciones de propósito comercial. En este modelo un usuario bien identificado, típicamente el creador o propietario del recurso, decide cómo protegerlo estableciendo cómo compartirlo, mediante controles de acceso impuestos por el sistema. Lo esencial es que el propietario del recurso puede cederlo a un tercero.

En sus inicios estos sistemas eran excesivamente simples, al permitir un conjunto limitado de operaciones posibles sobre un recurso (*rwx* por propietario, grupo o resto de usuarios, como en Unix), pero

rápidamente se añadieron las famosas listas de control de accesos (ACLs), listas de usuarios y grupos con sus permisos específicos (1).

1.2.4.3 Control de acceso basado en Roles (RBAC)

En este modelo a los usuarios le son asignados uno o varios roles mientras que los permisos y privilegios se asignan a estos roles. Por tanto, las políticas de control de accesos basado en roles regulan el acceso de los usuarios a la información en términos de sus actividades y funciones de trabajo (roles), representándose así de forma natural la estructura de las organizaciones. Este modelo permite la construcción jerárquica de estas políticas de acceso, por herencia o especialización, por ello tiene el potencial de reducir la complejidad y el coste de la administración de seguridad en entornos heterogéneos.

Dada la alta integración entre los roles y las responsabilidades de los usuarios, se pueden seguir los principios del mínimo privilegio y de la separación de responsabilidades. Estos principios son vitales para alcanzar el objetivo de integridad, al requerir que a un usuario no se le otorguen mayores privilegios que los necesarios para efectuar su trabajo (1). Este modelo es ampliamente aceptado e incluso recomendado como una buena práctica. Sistemas como Microsoft Active Directory, SELinux, Solaris, Oracle DBMS, y otros, lo implementan de cierta forma.

Con el estudio de los modelos de control de acceso se ha llegado a la conclusión de que DAC y MAC, por sí solos no llegan a cubrir las necesidades de la mayoría de las organizaciones. Determinándose que el modelo MAC es demasiado rígido y DAC es débil si se quiere controlar el acceso de forma efectiva.

RBAC, modelo que intenta unificar los clásicos DAC y MAC, basándose en roles, impone el control de acceso de mejor forma que los anteriores.

1.2.5 Criptografía

Desde que el hombre ha necesitado comunicarse con los demás ha tenido la necesidad de que algunos de sus mensajes solo fueran conocidos por las personas a quien estaban destinados. La necesidad de poder enviar mensajes de forma que solo fueran entendidos por los destinatarios hizo que se crearan sistemas de cifrado, de forma que un mensaje después de un proceso de transformación, lo que se le llama cifrado, solo pudiera ser leído siguiendo un proceso de descifrado (5).

La criptografía (del griego kryptos, "escondido", y graphein, "escribir"), el arte de enmascarar los mensajes con signos convencionales, que sólo cobran sentido a la luz de una clave secreta, nació con la escritura. Su rastro se encuentra ya en las tablas cuneiformes y los papiros demuestran que los primeros egipcios, hebreos, babilonios y asirios conocieron y aplicaron sus inescrutables técnicas, que alcanzan hoy su máxima expresión gracias al desarrollo de los sistemas informáticos y de las redes mundiales de comunicación (6).

1.2.6 Mecanismos y medidas de seguridad informática

Los mecanismos de defensa pueden ser muy diversos pero de manera general el objetivo de los mismos siempre es uno de los siguientes:

- *Prevención*: Aumentar la seguridad del sistema previniendo la ocurrencia de violaciones a la seguridad.
- *Detección*: Detectar la ocurrencia de una violación a la seguridad en el momento en que se produce la misma.
- *Recuperación*: Retornar el sistema a su normal funcionamiento después de una violación. (3).

Las medidas de seguridad pueden ser a su vez de tres tipos fundamentales: lógicas, físicas y administrativas.

Medidas físicas: Se aplican mecanismos para impedir el acceso directo o físico no autorizado al sistema. También protegen al sistema de desastres naturales o condiciones medioambientales adversas.

Medidas lógicas: Incluye un conjunto de medidas de acceso lógico a los recursos y a la información, garantizando el uso correcto de los mismos. Se refiere más a la protección de la información almacenada.

Medidas administrativas: Son aquellas que deben ser tomadas por las personas encargadas de definir la política de seguridad de la institución para ponerla en práctica, hacerla viable y vigilar su correcto funcionamiento (1).

1.3 Sistemas proveedores de seguridad para aplicaciones web como objeto de estudio.

1.3.1 Descripción General

La existencia en el mundo de numerosos sistemas vinculados a la administración de la seguridad brinda hoy la posibilidad de realizar un estudio de la concepción de los mismos, la necesidad por la que surgen y el contexto en que se enmarcan.

La labor fundamental que realizan es brindar seguridad de una manera centralizada a través de servicios con el objetivo de ayudar a las aplicaciones web a mantener la confidencialidad, integridad y disponibilidad de la información sensible. Para esto, implementan sofisticados mecanismos de control de acceso como la autenticación y la autorización.

Para lograr acceso a información de acuerdo a la jerarquía que posea el usuario los sistemas aplican los métodos de control de acceso, generalmente basado en roles (RBAC). Esto posibilita que los usuarios accedan solo a la información disponible según los permisos que posean, como consecuencia de las necesidades y restricciones de la institución.

Los sistemas de seguridad implementan mecanismos para auditar y registrar en un historial o bitácora cualquier acción que se realice o intente realizarse, ya sea maliciosa o no.

Generalmente estos sistemas incluyen la capacidad de analizar datos de comportamiento de los usuarios en cuanto a la solicitud de información, envíos y entradas que realiza el usuario en determinada aplicación.

Se implementan mecanismos para controlar la disponibilidad de los recursos.

Se aplican políticas de seguridad flexibles, de fácil configuración.

Se centraliza la administración de la seguridad de los sistemas web, siendo más robusta al estar manejada por un especialista en seguridad.

Todas estas características traen mejoras consigo, permiten obtener una visión unificada de la seguridad de la red, evitar errores de configuración, mínimos costos al implementarse una sola vez el módulo

correspondiente a seguridad, máxima integración entre los sistemas pues se logra mayor compatibilidad, un módulo de seguridad mucho mas robusto al ser concebido y manejado por especialistas, menos vulnerabilidades pues al pensarse en la seguridad no como un algo complementario a una aplicación sino como un sistema independiente para proveer a las aplicaciones web se hace un mejor estudio de los conceptos y necesidades orientados a seguridad.

1.3.2 Elementos de seguridad en sistemas web como dominio actual del problema.

Con el grado de interconectividad existente hoy en el mundo ya evolucionando desde hace algún tiempo, la web ha venido a corroborar, estableciéndose como un medio de comunicación e intercambio de información. Por lo tanto, la necesidad de implementar mecanismos para asegurar los datos que viajan por la red, es inminente.

Según Simson Garfinkel en “Web security, privacy & commerce” la seguridad en la web tiene tres etapas esenciales:

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

1.3.2.1 Tipos de ataques en aplicaciones web

Debido a la evolución de las tecnologías y el creciente desarrollo de aplicaciones web, de las que dependen numerosas empresas en el mundo para vender sus productos mediante el comercio electrónico**,** los atacantes han implementado mecanismos complejos e igualmente desarrollados para materializar sus amenazas, cometiendo delitos informáticos como robos y fraudes a diferentes niveles.

A continuación se muestran algunas formas de ataques específicos para aplicaciones web.

Autenticación

- *Fuerza bruta:* Es un proceso automatizado de prueba y error utilizado para adivinar un nombre de usuario, contraseña, número de tarjeta de crédito o clave criptográfica.
- *Autenticación insuficiente:* Ocurre cuando un sitio web permite a un atacante acceder a contenido sensible o funcionalidades sin haberse autenticado correctamente.
- *Débil validación en la recuperación de contraseñas:* Se produce cuando un sitio web permite a un atacante obtener, modificar o recuperar, de forma ilegal, la contraseña de otro usuario.

Autorización

- *Predicción de credenciales/sesión:* Es un método de secuestro o suplantación de un usuario del sitio web.
- *Autorización insuficiente:* Se produce cuando un sitio web permite acceso a contenido sensible o funcionalidades que deberían requerir un incremento de las restricciones en el control de acceso.
- *Expiración de sesión insuficiente:* Se produce cuando un sitio web permite a un atacante reutilizar credenciales de sesión o IDs de sesión antiguos para llevar a cabo la autorización.
- *Fijación de sesión:* Es una técnica de ataque que fuerza al ID de sesión de un usuario a adoptar un valor determinado

Ataques en la parte cliente

- *Suplantación de contenido:* Es una técnica de ataque utilizada para engañar al usuario haciéndole creer que cierto contenido que aparece en un sitio web es legítimo, cuando en realidad no lo es.
- *Cross-site scripting (XSS):* Es una técnica de ataque que fuerza a un sitio web a repetir código ejecutable facilitado por el atacante y que se carga en el navegador del usuario.

Ejecución de comandos

- *Desbordamiento de buffer:* Es un ataque que altera el flujo de una aplicación sobrescribiendo partes de la memoria.

- *Ataques de formato de cadena:* Alteran el flujo de una aplicación utilizando las capacidades proporcionadas por las librerías de formato de cadenas para acceder a otro espacio de memoria.
- *Inyección LDAP:* Es una técnica de ataque usada para explotar sitios web que construyen sentencias LDAP a partir de datos de entrada suministrados por el usuario.
- *Comandos de Sistema Operativo:* Es una técnica de ataque utilizada para explotar sitios web mediante la ejecución de comandos de sistema operativo a través de la manipulación de las entradas a la aplicación.
- *Inyección de código SQL*: Es una técnica de ataque usada para explotar sitios web que construyen sentencias SQL a partir de entradas facilitadas por el usuario.
- *Inyección de código SSI (Server-side Include):* Es una técnica de explotación en la parte servidora que permite a un atacante enviar códigos a una aplicación web, que posteriormente será ejecutado localmente por el servidor web.
- *Inyección XPath:* Es una técnica de ataque utilizada para explotar sitios web que construyen consultas XPath con datos de entrada facilitados por el usuario.

Revelación de información

- *Indexación de directorio:* Es una función del servidor web que lista todos los ficheros del directorio solicitado si no se encuentra presente el fichero de inicio habitual.
- *Fuga de información:* Se produce cuando un sitio web revela información sensible como comentarios de los desarrolladores o mensajes de error, que puede ayudar a un atacante para explotar el sistema.
- *Técnica de ataque Path Traversal:* Fuerza el acceso a ficheros, directorios y comandos que potencialmente residen fuera del directorio “document root” del servidor web.
- *Localización de recursos predecibles:* Es una técnica de ataque usada para descubrir contenido y funcionalidades ocultas en el sitio web.

Ataques lógicos

- *Abuso de funcionalidad:* Es una técnica de ataque que usa las propias capacidades y funcionalidades de un sitio web para consumir, estafar o evadir mecanismos de control de acceso.
- *Denegación de servicio (Denial of Service, DoS):* Es una técnica de ataque cuyo objetivo es evitar que un sitio web permita la actividad habitual de los usuarios.
- *Anti-automatización insuficiente:* Se produce cuando un sitio web permite a un atacante automatizar un proceso que sólo debe ser llevado a cabo manualmente.
- *Validación de proceso insuficiente:* Se produce cuando un sitio web permite a un atacante evadir o engañar el flujo de control esperado por la aplicación (8).

1.3.3 Situación Problemática

La siguiente investigación surge por la creciente necesidad de aunar esfuerzos para combatir problemas de seguridad. En un ambiente de intranet, donde además se depende de servicios informáticos sólidos y, principalmente seguros, y existe relación entre los sistemas que se desarrollan, es indudable la puesta en marcha de un mecanismo capaz de dar soporte a los requerimientos de seguridad de estos sistemas de forma centralizada.

Los administradores saben por experiencia propia que las amenazas de los hackers y los delincuentes continúan en aumento, lo que los obliga a dedicar mucho tiempo a la administración de la seguridad. En este punto se plantea la primera problemática y polémica, por qué no crear un módulo específicamente de seguridad que se use por los demás sistemas y que tenga su propio administrador, persona especializada en esta temática que pueda introducir mejoras a un diseño original.

Tradicionalmente cuando se concibe un sistema web se le asocia un módulo de seguridad cuya filosofía no difiere en gran medida a la puesta en práctica por otros sistemas a la hora de implementar seguridad. Sin embargo, el aumento de las vulnerabilidades debido a diseños e implementaciones deficientes por no prestársele suficiente atención a la seguridad, el crecimiento de técnicas para quebrantar la misma que viene como anexo al desarrollo tecnológico e informático, el personal no comprometido con la seguridad y especializado en esta temática, el desconocimiento de medidas orientadas al tema, así como la no reutilización y optimización de código son efectos secundarios que se suceden constantemente.

Cómo se realiza hoy la actualización contra ataques, pues todos los sistemas implementan sus propias formas, y se repite el proceso para cada uno. Por otra parte la poca o ninguna estandarización del término de seguridad informática para las aplicaciones web es otro problema que se plantea, pues no se siguen definiciones asociadas a seguridad y no se hace un estudio importante cuando se concibe el módulo que se destina para dar seguridad a los sistemas web.

De manera general el problema se reduce a la existencia de múltiples aplicaciones y como consecuencia de módulos de seguridad asociados a cada sistema lo que se traduce en diferencias en cuanto a las vulnerabilidades y crecimiento en este aspecto, cuando debería fomentarse la concentración de esfuerzos para lograr un sistema centralizado que implemente alta seguridad.

1.4 Análisis de otras soluciones existentes

Como consecuencia del desarrollo de las aplicaciones web, y las redes también se han perfeccionado los ataques y las medidas de seguridad. Numerosas aplicaciones se han creado con el objetivo de ayudar a sus clientes a ser menos vulnerables en cuanto a ataques informáticos. Empresas dedicadas al desarrollo de estas aplicaciones en el mundo son Open Web Application Security Project (OWASP) y su consorcio Web Application Security Center (WASC), quienes facilitan herramientas, documentación y códigos de forma gratuita y aseguran la buena calidad y garantía de sus productos. La SPI Dinamics y la TB-Security tienen gran fortaleza en el lanzamiento específicamente para seguridad de aplicaciones web.

Algunos productos orientados a la seguridad lanzados al mercado son:

La línea llamada Microsoft Forefront

La familia de productos de seguridad para empresas Microsoft Forefront ofrece protección y control de la seguridad de la infraestructura informática orientada a empresas.

Su completo conjunto de productos de seguridad, que se integran entre sí y con la infraestructura informática de la empresa, puede complementarse e interoperar con soluciones de terceros. De esta manera, se obtienen soluciones de seguridad muy completa y de defensa en profundidad.

Gracias a que se simplifican la administración, la generación de informes, el análisis y la implementación, es posible proteger de manera más eficiente y efectiva los recursos informáticos de la empresa y brindar un acceso más seguro a las aplicaciones y servidores.

Con una protección que brinda una excelente respuesta y cuenta con el respaldo de las guías técnicas de Microsoft, Forefront ayuda a enfrentar con confiabilidad las amenazas en constante cambio y las demandas de negocios cada vez mayores (7).

Cams

El Cams, de Cafesoft, es un software para la Web que centraliza la autenticaron de usuarios, el control de acceso y la administración. Provee seguridad para recursos que están hospedados en todos los líderes Web y aplicaciones J2EE servidor: están incluidos Apache, Microsoft IIS, BEA WebLogic, IBM WebSphere, JBoss, Oracle 9iAs, Pramati y Tomcat. Los recursos protegidos por Cams pueden residir en una intranet corporativa, extranet o Internet, y pueden ser documentos estáticos y JSP/servlet, ASP.Net, PHP, Cold Fusion, y aplicaciones Web CGI.

Citrix Application Firewall

Citrix® Application Firewall protege las aplicaciones web del creciente número de ataques a nivel de las aplicaciones, incluidas las violaciones de desborde de memoria intermedia, los ataques con la técnica de inyección SQL, reescritura de comandos entre sitios y otros. Además de probadas defensas contra ataques, Citrix Application Firewall brinda protección contra robo de identidades al asegurar la información confidencial de la empresa y los datos confidenciales de los clientes.

Es la solución de seguridad para aplicaciones web que mejor funciona en la industria, con la capacidad para proteger a los servidores web sin disminuir el rendimiento ni los tiempos de respuesta de las aplicaciones. Citrix Application Firewall ofrece el rendimiento más adecuado para cualquier instalación de la empresa o del centro de datos (10).

SeguriNet

Consiste en el análisis, diseño e implementación de un sistema genérico de seguridad de aplicaciones Web y las acciones a tomar para su adaptación al “Sistema de Gestión de la Información” del Instituto

Nacional de Meteorología. Pretende garantizar de manera eficiente, íntegra y consistente la seguridad de sus datos así como la creación, configuración, autenticación y control de usuarios, reportes de accesibilidad, historial e información de las operaciones del sistema. Desarrollado en la UCI, Cuba, actualmente no se encuentra funcionando aunque si cumplió sus objetivos (11).

En la UCI, existe un subsistema de gestión de la seguridad implantado en el Proyecto Akademos, asegurando la confidencialidad de la información con carácter docente manejada en el proyecto anteriormente mencionado, reduciendo costes asociados a perdidas de información, por incidentes de seguridad y brindando servicios a los demás módulos pertenecientes al sistema.

Sin embargo, pese a la existencia de estos y otros productos, y la buena calidad de los mismos, no es posible adaptarlos a las necesidades de la facultad 9 en la UCI, tanto por cuestiones de costo como por incompatibilidades en cuanto a los entornos y fines para los que fueron concebidos.

1.5 Conclusiones

Teniendo como punta de partida el estudio de los fundamentales conceptos asociados a la seguridad informática, el análisis de la situación problemática, así como la descripción del objeto de estudio se han consolidado los conocimientos más importantes para el buen desarrollo de un sistema centralizado que brinde servicios a los sistemas web de la facultad 9.

CAPÍTULO 2 Tendencias y tecnologías actuales a considerar.

2.1 Introducción

En este capitulo se realiza un estudio de las principales tecnologías actuales a utilizar para realizar el sistema de seguridad, así como la fundamentación de las metodologías y herramientas seleccionadas para dar solución al problema propuesto en el capitulo anterior.

2.2 Metodología de Desarrollo de Software.

El desarrollo vertiginoso que desde hace algunos años ha venido experimentado la industria de la informática hace que, que los sistemas informáticos sean cada vez más complejos y que los usuarios exijan por la calidad de los mismos. Por otro lado, el alto nivel de competitividad existente en este mercado determina que pequeños retrasos en la entrega de un sistema pueda implicar la pérdida de gran cantidad de usuarios finales e importantes clientes.

Para lograr un proceso bien definido y bien gestionado los desarrolladores hacen uso de métodos de desarrollo de software que permitan lograr proyectos que no fracasen.

2.2.1 El Proceso Unificado de desarrollo de software

El Proceso Unificado es un proceso de desarrollo de software. Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software. El proceso unificado o RUP es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyectos. (12)

RUP es el resultado de la unión de varias metodologías de desarrollo de software y permite sacar el máximo provecho de los conceptos asociados a la orientación a objetos y modelado visual. La correcta aplicación de RUP permite reducir los tiempos de desarrollo, aumentar la calidad de las aplicaciones y disminuir los costes de mantenimiento.

RUP utiliza el Lenguaje Unificado de Modelado UML, para preparar todos los esquemas de un sistema de software. UML es una parte esencial del Proceso Unificado, pues fueron desarrollados de forma paralela. (12)

2.2.1.1 Características del Proceso Unificado de Software

El Proceso Unificado se resume en tres frases claves: dirigidos por casos de uso, centrado en la arquitectura, e iterativo e incremental. Esto es lo que hace único al Proceso Unificado.

- **Dirigido por los casos de uso:** La razón de ser de un sistema es brindar servicios a los usuarios, RUP define casos de uso como el conjunto de acciones que debe realizar el sistema para dar un resultado de valor a un determinado usuario. Los casos de uso además de especificar los requisitos de un sistema, guían su diseño, implementación y prueba; guían el proceso de desarrollo.

- **Centrado en la arquitectura:** La arquitectura es una vista del diseño completo con las características más importantes. Esta además de incluir las necesidades de los usuarios e inversores, incluye aspectos técnicos del hardware, sistema operativo, sistema de gestión de base de datos, protocolos de red, con los que debe coexistir el sistema. La arquitectura representa la forma del sistema, la cual va madurando en su interacción con los casos de uso hasta llegar a un equilibrio entre funcionalidad y características técnicas.

- **Iterativo e Incremental:** El alto nivel de complejidad de los sistemas actuales hacen que sea factible dividir el proceso en varios mini-proyectos. Cada uno de estos se les denomina iteración y pueden o no representar un incremento en el grado de terminación del producto completo. En cada iteración los desarrolladores seleccionan un grupo de casos de uso, los cuales se diseñan, implementan y prueban.

Todas estas características son de gran importancia. La arquitectura proporciona la estructura sobre la cual guiar las iteraciones, mientras que los casos de uso definen los objetivos y dirigen el trabajo de cada iteración. La eliminación de una de las tres ideas reduciría drásticamente el valor de Proceso Unificado.

2.2.1.2 El Lenguaje Unificado de Modelado

El Lenguaje Unificado de Modelado (UML) es un lenguaje estándar de modelado para software. Es un lenguaje para la visualización, especificación, construcción y documentación de los artefactos de sistemas en los que el software juega un papel importante. Básicamente, UML permite a los desarrolladores visualizar los resultados de su trabajo en esquemas o diagramas estandarizados.

UML (por sus siglas en inglés, Unified Modelling Language) es el lenguaje de modelado de sistemas de software más conocido en la actualidad. Permite visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para escribir un "plano" del sistema, 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. UML cuenta con varios tipos de modelos, los cuales muestran diferentes aspectos de las entidades representadas y se ha convertido en el estándar de facto para construir software orientado a objetos.

2.3 Rational Rose como herramienta CASE

Rational Rose es la herramienta CASE que comercializan los desarrolladores de UML (Booch, Rumbaugh y Jacobson) y que soporta de forma completa la especificación del UML. Esta herramienta propone la utilización de cuatro tipos de modelos para realizar un diseño del sistema, utilizando una vista estática y otra dinámica de los modelos del sistema, uno lógico y otro físico. Permite crear y refinar estas vistas creando de esta forma un modelo completo que representa el dominio del problema y el sistema de software:

- **Desarrollo Iterativo:** Utiliza un proceso de desarrollo iterativo controlado donde el desarrollo se lleva a cabo en una secuencia de iteraciones. Cada iteración comienza con una primera aproximación del análisis, diseño e implementación para identificar los riesgos del diseño. Cuando la implementación pasa todas las pruebas que se determinan en el proceso, ésta se revisa y se añaden los elementos modificados al modelo de análisis y diseño. Una vez que la actualización del modelo se ha modificado, se realiza la siguiente iteración.

- **Trabajo en Grupo:** Permite que haya varias personas trabajando a la vez en el proceso iterativo controlado, para ello posibilita que cada desarrollador opere en un espacio de trabajo privado que contiene el modelo completo y tenga un control exclusivo sobre la propagación de los cambios en ese espacio de trabajo.
- **Generador de Código:** Se puede generar código en distintos lenguajes de programación a partir de un diseño en UML.
- **Ingeniería Inversa:** Proporciona mecanismos para realizar la denominada Ingeniería Inversa, es decir, a partir del código de un programa, se puede obtener información sobre su diseño.

2.4 Microsoft .NET: plataforma de desarrollo

.NET es una plataforma de desarrollo de software con énfasis en transparencia de redes, con independencia de plataforma y que permite un rápido desarrollo de aplicaciones. Microsoft desarrolló una estrategia horizontal donde integra todos sus productos, desde el Sistema Operativo hasta las herramientas de mercado. (14)

Con esta plataforma Microsoft incursiona de lleno en el campo de los Servicios Web y establece el XML como norma en el transporte de información en sus productos y lo promociona como tal en los sistemas desarrollados utilizando sus herramientas.

.NET ofrece una manera rápida y económica pero a la vez segura y robusta de desarrollar aplicaciones o como la misma plataforma las denomina, soluciones permitiendo a su vez una integración más rápida y ágil entre empresas y un acceso más simple y universal a todo tipo de información desde cualquier tipo de dispositivo. (14)

La parte de .NET que permite desarrollar estas aplicaciones con mayor rendimiento, seguridad y confiabilidad es el **Framework .NET**:

Fundamento de la siguiente generación de aplicaciones basadas en Windows que son fáciles de construir, emplear e integrar con otros sistemas en red. Simplifica el desarrollo de software Windows. Proporciona a los desarrolladores un único enfoque para construir tanto las aplicaciones de escritorio llamadas a veces

aplicaciones inteligentes para el cliente y aplicaciones basadas en Web. Puede minimizar los conflictos entre aplicaciones al ayudar a coexistir a componentes de software incompatibles. (15)

Consta de dos partes principales: El lenguaje común en tiempo de ejecución (Common Language Runtime) y la biblioteca de clases de .NET Framework.

Lenguaje común en tiempo de ejecución (Common Language Runtime): Otorga los servicios comunes a las aplicaciones de .NET Framework. Los programas pueden estar escritos en casi cualquier lenguaje, C, C++, C#, y Microsoft Visual Basic, así como algunos lenguajes más antiguos tales como Fortran. El runtime simplifica la programación al asistir con muchas tareas triviales de escritura de código. Estas tareas incluyen administración de la memoria que puede ser un importante generador de errores en el sistema (bugs), administración de la seguridad y manejo de errores.

Biblioteca de clases de .NET Framework: La biblioteca incluye conjuntos pre empaquetados de funcionalidades que los desarrolladores pueden usar para ampliar con mayor velocidad las capacidades de su propio software. La biblioteca incluye tres componentes claves:

- ASP.NET para ayudar a construir aplicaciones y servicios Web.
- Windows Forms para facilitar el desarrollo de interfaces de usuario para clientes inteligentes.
- ADO.NET para ayudar a conectar las aplicaciones a bases de datos.

Las ventajas claves de .NET Framework:

- Ayuda a los profesionales a integrar mejor a los sistemas existentes con su soporte nativo para servicios Web.
- Asiste tanto a los usuarios como a los servidores Web con el uso del software.
- Facilita el desarrollo de software con confiabilidad, escalabilidad, rendimiento y seguridad mejorada.
- Ayuda a los desarrolladores a ser más productivos al hacer más fácil para ellos el reutilizar código existente. Les permite integrar con mayor facilidad componentes escritos en cualquiera de los más de 20 lenguajes de programación que soporta y los ayuda a construir software con mayor facilidad para una amplia gama de dispositivos que utilizan las mismas habilidades y herramientas. (15)

2.4.1 Lenguaje C#

El lenguaje C# fue diseñado por Microsoft especialmente para la plataforma .Net, y a pesar de ser un lenguaje joven tiene incorporado las mejores características de otros lenguajes así como nuevas potencialidades, además de que se plantea su compilador es el más depurado y optimizado de los incluidos en el Framework .Net. Fue diseñado para lograr una combinación idónea de simplicidad, expresividad y desempeño eficiente.

C# tiene una sintaxis similar a la de C++, sin embargo incorpora un modelo de referencia a objetos parecido al de Delphi o Java, eliminando la necesidad del engorroso trabajo con punteros (aunque ofrece las herramientas para usarlos en caso de extrema necesidad). El listado de características del C# abarca mucho más de lo que se enumera a continuación:

- Los tipos básicos son tratados como clases.
- Cuenta con gestión automática de memoria.
- Implementa una fuerte política de seguridad de tipos,
- Brinda mecanismos como los índices y la instrucción foreach, que hacen más fácil e intuitivo el trabajo.
- Elimina la herencia múltiple (ofrece el uso de interfaces).
- Facilita el trabajo con propiedades y eventos.(16)

2.4.2 Asp.NET

Microsoft ha desarrollado una nueva tecnología denominada ASP.NET, como parte de la estrategia .NET para el desarrollo Web, con el objetivo de resolver las limitaciones de ASP y posibilitar la creación de software como servicio.

ASP.NET es un marco de trabajo de programación generado en Common Language Runtime que puede utilizarse en un servidor para generar eficaces aplicaciones Web. (17)

Para tener una idea más clara de lo que implica el desarrollo de las aplicaciones utilizadas y distribuidas por Internet y las características que representan a la plataforma ASP.NET, es importante hacer mención de algunos aspectos con los que cuentan las aplicaciones actuales:

Aplicaciones cliente/servidor: Estas aplicaciones son típicamente en un formato de ejecutables compilados. Éstos pueden integrar toda la riqueza de una interfaz de usuario, tal es el caso de las aplicaciones de desempeño y productividad, pero no se reúne la lógica de negocio como un recurso que se pueda reutilizar. Además, acostumbran ser menos gestionables y escalables que las demás aplicaciones.

Aplicaciones que utilizan el navegador: Dichas aplicaciones están caracterizadas por contar con una interfaz de web rica y muy útil. La interfaz gráfica integra varias tecnologías, las cuales son el HTML, XHTML, scripting, entre otras, siempre y cuando el navegador que se esté utilizando soporte estas tecnologías.

ASP.NET ofrece varias ventajas importantes acerca de los modelos de programación Web anteriores:

Mejor rendimiento: ASP.NET es un código de Common Language Runtime compilado que se ejecuta en el servidor. A diferencia de sus predecesores, ASP.NET puede aprovechar las ventajas del enlace anticipado, la compilación just-in-time, la optimización nativa y los servicios de caché desde el primer momento. Esto supone un incremento espectacular del rendimiento antes de siquiera escribir una línea de código.

Compatibilidad con herramientas de primer nivel: El marco de trabajo de ASP.NET se complementa con un diseñador y una caja de herramientas muy completos en el entorno integrado de programación (Integrated Development Environment, IDE) de Visual Studio. La edición WYSIWYG, los controles de servidor de arrastrar y colocar y la implementación automática son sólo algunas de las características que proporciona esta eficaz herramienta. (17)

Eficacia y flexibilidad: Debido a que ASP.NET se basa en Common Language Runtime, la eficacia y la flexibilidad de toda esa plataforma se encuentra disponible para los programadores de aplicaciones Web. La biblioteca de clases de .NET Framework, la Mensajería y las soluciones de acceso a datos se encuentran accesibles desde la Web de manera uniforme. ASP.NET es también independiente del

lenguaje, por lo que puede elegir el lenguaje que mejor se adapte a la aplicación o dividir la aplicación en varios lenguajes. Además, la interoperabilidad de Common Language Runtime garantiza que la inversión existente en programación basada en COM se conserva al migrar a ASP.NET.

Simplicidad: ASP.NET facilita la realización de tareas comunes, desde el sencillo envío de formularios y la autenticación del cliente hasta la implementación y la configuración de sitios. Por ejemplo, el marco de trabajo de página de ASP.NET permite generar interfaces de usuario, que separan claramente la lógica de aplicación del código de presentación, y controlar eventos en un sencillo modelo de procesamiento de formularios de tipo Visual Basic. Además, Common Language Runtime simplifica la programación, con servicios de código administrado como el recuento de referencia automático y el recolector de elementos no utilizados. (17)

Facilidad de uso: ASP.NET emplea un sistema de configuración jerárquico, basado en texto, que simplifica la aplicación de la configuración al entorno de servidor y las aplicaciones web. Debido a que la información de configuración se almacena como texto sin formato, se puede aplicar la nueva configuración sin la ayuda de herramientas de administración local. Esta filosofía de "administración local cero" se extiende asimismo a la implementación de las aplicaciones ASP.NET Framework. Una aplicación ASP.NET Framework se implementa en un servidor sencillamente mediante la copia de los archivos necesarios al servidor. No se requiere el reinicio del servidor, ni siquiera para implementar o reemplazar el código compilado en ejecución.

Escalabilidad y disponibilidad: ASP.NET se ha diseñado teniendo en cuenta la escalabilidad, con características diseñadas específicamente a medida, con el fin de mejorar el rendimiento en entornos agrupados y de múltiples procesadores. Asimismo, el motor de tiempo de ejecución de ASP.NET controla y administra los procesos de cerca, por lo que si uno no se comporta adecuadamente (filtraciones, bloqueos), se puede crear un proceso nuevo en su lugar, lo que ayuda a mantener la aplicación disponible constantemente para controlar solicitudes. (17)

Posibilidad de personalización y extensibilidad: ASP.NET presenta una arquitectura bien diseñada que permite a los programadores insertar su código en el nivel adecuado. De hecho, es posible extender o reemplazar cualquier subcomponente del motor de tiempo de ejecución de ASP.NET con su propio

componente escrito personalizado. La implementación de la autenticación personalizada o de los servicios de estado nunca ha sido más fácil.

Seguridad: Con la autenticación de Windows integrada y la configuración por aplicación, se puede tener la completa seguridad de que las aplicaciones están a salvo.

2.5 Servicios Web

Los servicios Web son unidades de código discretas, cada una de las cuales se encarga de un conjunto limitado de tareas. Están basados en XML, el lenguaje universal del intercambio de información en Internet y pueden utilizarse en cualquier plataforma o sistema operativo, independientemente del lenguaje de programación utilizado. Aunque los servicios web XML son independientes uno entre si pueden vincularse y formar un grupo de colaboración para realizar un tarea determinada. (18)

Los servicios Web XML también permiten que los programadores puedan elegir entre generar todas las partes de sus aplicaciones o utilizar servicios Web XML creados por otros. De este modo, una empresa no necesita crear todas las partes de una solución completa. Como tiene la capacidad para anunciar y ofrecer sus propios servicios Web XML, se crean nuevos flujos de ingresos para la empresa.

Un beneficio clave en la nueva arquitectura de servicios Web es la habilidad de entregar soluciones integradas e interoperables. Ayudar a proteger la integridad, confidencialidad y seguridad de los servicios Web a través de la aplicación de un modelo comprensivo de seguridad es crítico. (18)

2.5.1 Protocolos que usan los Servicios Web.

Los servicios Web XML se invocan en Internet por medio de protocolos estándar tales como SOAP, XML y UDDI. Estos protocolos los definen organizaciones de estándares públicos como el consorcio W3C.

XML

Extensible Markup Language no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades. Es un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable.

XML es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.

SOAP

Simple Object Access Protocol, es un protocolo estándar creado por Microsoft IBM, define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML, es decir que permite la comunicación entre aplicaciones a través de mensajes por medio de Internet. Es independiente de la plataforma, y del lenguaje, además de ser un protocolo abierto, ya que es una especificación abierta, construido sobre tecnologías también abiertas como XML y HTTP.

SOAP es un marco extensible y descentralizado que permite trabajar sobre múltiples protocolos de redes informáticas. Los procedimientos de llamadas remotas pueden ser modelados en la forma de varios mensajes SOAP interactuando entre sí, funciona sobre cualquier protocolo de Internet, generalmente http.

UDDI

Universal Discovery Description and Integration son las páginas amarillas de los servicios Web. Al igual que con las páginas amarillas tradicionales, se puede buscar una empresa que ofrezca los servicios que se necesitan, obtener información sobre el servicio ofrecido y contactar con alguien para más información. Naturalmente, también se puede ofrecer un servicio Web sin registrarlo en UDDI, al igual que se puede abrir un negocio en el sótano y confiar en la publicidad boca a boca, pero si se desea que se conozca tales servicios por un numero considerable de personas interesadas, se necesita UDDI.

WSDL

Web Services Description Language 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.

Ventajas de los Servicios Web.

- Aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen.
- Los servicios Web fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.
- Al apoyarse en HTTP, los servicios Web pueden aprovecharse de los sistemas de seguridad firewall sin necesidad de cambiar las reglas de filtrado.
- Permiten que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados.
- Permiten la interoperabilidad entre plataformas de distintos fabricantes por medio de protocolos estándar.

2.6 Otros protocolos

HTTP

El protocolo de transferencia de hipertexto (*HyperText Transfer Protocol*) es el protocolo usado en cada transacción de la Web (WWW). El hipertexto es el contenido de las páginas web, y el protocolo de transferencia es el sistema mediante el cual se envían las peticiones de acceso a una página y la respuesta con el contenido. También sirve el protocolo para enviar información adicional en ambos sentidos, como formularios con campos de texto.

HTTP es un protocolo sin estado, es decir, que no guarda ninguna información sobre conexiones anteriores. Al finalizar la transacción todos los datos se pierden. Por esto se popularizaron las cookies, que son pequeños ficheros guardados en el propio ordenador que puede leer un sitio web al establecer conexión con él, y de esta forma reconocer a un visitante que ya estuvo en ese sitio anteriormente. Gracias a esta identificación, el sitio web puede almacenar gran número de información sobre cada visitante, ofreciéndole así un mejor servicio.

HTTP dispone de una variante cifrada mediante SSL llamada HTTPS.

SSL

SSL proporciona autenticación y privacidad de la información entre extremos sobre Internet mediante el uso de criptografía. Habitualmente, sólo el servidor es autenticado (es decir, se garantiza su identidad) mientras que el cliente se mantiene sin autenticar; la autenticación mutua requiere un despliegue de infraestructura de claves públicas (o PKI) para los clientes. Los protocolos permiten a las aplicaciones cliente-servidor comunicarse de una forma diseñada para prevenir escuchas (eavesdropping), la falsificación de la identidad del remitente (phising) y mantener la integridad del mensaje.

SSL se ejecuta en una capa entre los protocolos de aplicación como HTTP, SMTP, NNTP y sobre el protocolo de transporte TCP, que forma parte de la familia de protocolos TCP/IP. Aunque pueda proporcionar seguridad a cualquier protocolo que use conexiones de confianza (tal como TCP), se usa en la mayoría de los casos junto a HTTP para formar HTTPS. HTTPS es usado para asegurar páginas World Wide Web para aplicaciones de comercio electrónico, utilizando certificados de clave pública para verificar la identidad de los extremos.

SSL adopta una variedad de medidas de seguridad:

- Numera todos los registros y usa el número de secuencia en el código de autenticación del mensaje (MAC).
- Usa un resumen de mensaje mejorado con una clave (de forma que solo con dicha clave se pueda comprobar el MAC).
- Proteje contra varios ataques conocidos (incluidos ataques man in the middle attack), como los que implican un degradado del protocolo a versiones previas (por tanto, menos seguras), o conjuntos de cifrados más débiles.
- El mensaje que finaliza el protocolo *handshake* (*Finished*) envía un *hash* de todos los datos intercambiados y vistos por ambas partes.
- La función pseudo aleatoria divide los datos de entrada en 2 mitades y las procesa con algoritmos hash diferentes (MD5 y SHA), después realiza sobre ellos una operación XOR. De esta forma se protege a sí mismo de la eventualidad de que alguno de estos algoritmos se revelen vulnerables en el futuro.

HTTPs

El protocolo HTTPS es la versión segura del protocolo HTTP. El sistema HTTPS utiliza un cifrado basado en las Secure Socket Layers (SSL) para crear un canal cifrado (cuyo nivel de cifrado depende del servidor remoto y del navegador utilizado por el cliente) más apropiado para el tráfico de información sensible que el protocolo HTTP. Cabe mencionar que el uso del protocolo HTTPS no impide que se pueda utilizar HTTP. Es aquí, cuando nuestro navegador nos advertirá sobre la carga de elementos no seguros (HTTP), estando conectados a un entorno seguro (HTTPS).

Los protocolos https son utilizados por navegadores como: Safari (navegador), Internet Explorer, Mozilla Firefox, Opera, entre otros.

Es utilizado principalmente por entidades bancarias, tiendas en línea, y cualquier tipo de servicio que requiera el envío de datos personales o contraseñas.

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. (20)

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...).

¿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

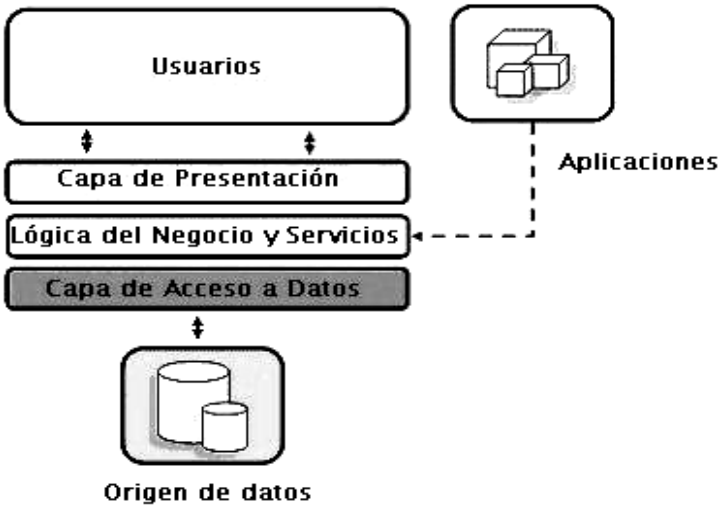
grandes volúmenes de datos, Las actualizaciones en un directorio son usualmente cambios sencillos de todo o nada, si es que permiten algo. (20)

2.7 Arquitectura propuesta

2.7.1 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. (12)

Fig 2. 1 Arquitectura en capas.



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. (19)

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 como Oracle o Sybase, un conjunto de documentos XML o incluso un servicio de directorio como el servidor LDAP. (19)

Una ventaja evidente de este modelo es que la capa de presentación puede desarrollarse de variadas maneras simultáneas: cliente Web, aplicación Windows, aplicación para otro Sistema Operativo, entre otras. Mientras menos responsabilidades recaigan en esta capa tanto mayor será la facilidad de desarrollar múltiples versiones de la misma. Otra ventaja sería la posibilidad de emigrar de servidor de bases de datos con un mínimo de 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.

2.8 Microsoft SQL Server 2000: como gestor de base de datos.

Microsoft SQL Server 2000 es un sistema de gestión de base de datos relacionales (SGBD) basada en el lenguaje SQL, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea. Entre sus características figuran:

- Soporte de transacciones.
- Gran estabilidad.
- Gran seguridad.
- Escalabilidad.
- Soporta procedimientos almacenados.
- Incluye también un potente entorno gráfico de administración que permite el uso de comandos DDL y DML gráficamente.
- Permite trabajar en modo cliente- servidor donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo acceden a la información.

- Además permite administrar información de otros servidores de datos.

Para el desarrollo de aplicaciones más complejas (tres o más capas), Microsoft SQL Server 2000 incluye interfaces de acceso para la mayoría de las plataformas de desarrollo, cuenta con un lenguaje (Transact-SQL) para programar procedimientos almacenados y triggers; permite definir tablas, índices, vistas, etc., es distribuido y escalable, con soporte para 32 procesadores y 64 GB de RAM, es más fácil de usar que el Oracle y más potente que MySQL.

SQLServer es un gestor de base de datos fácil de utilizar para construir, administrar e implementar aplicaciones de negocios. Esto significa tener que poner a disposición un modelo de programación rápido y sencillo para desarrolladores, eliminando la administración de base de datos para operaciones estándar, y suministrando herramientas sofisticadas para operaciones más complejas.

2.9 Aplicaciones de Internet Ricas.

Una Aplicación de Internet Rica o RIA (*Rich Internet Application*) es un nuevo tipo de aplicación Web cuyo objetivo es el de incrementar y mejorar las opciones y capacidades de las aplicaciones Web tradicionales.

Las limitaciones en la capa de presentación de los actuales navegadores Web y del lenguaje HTML ha impulsado a los desarrolladores a utilizar este nuevo tipo de aplicaciones que permiten, entre otras cosas, mejorar la experiencia entre el usuario y la aplicación, la ejecución de contenido multimedia y la carga de aplicaciones online/offline, dependiendo de la tecnología RIA que se utilice.

2.9.1 AJAX.

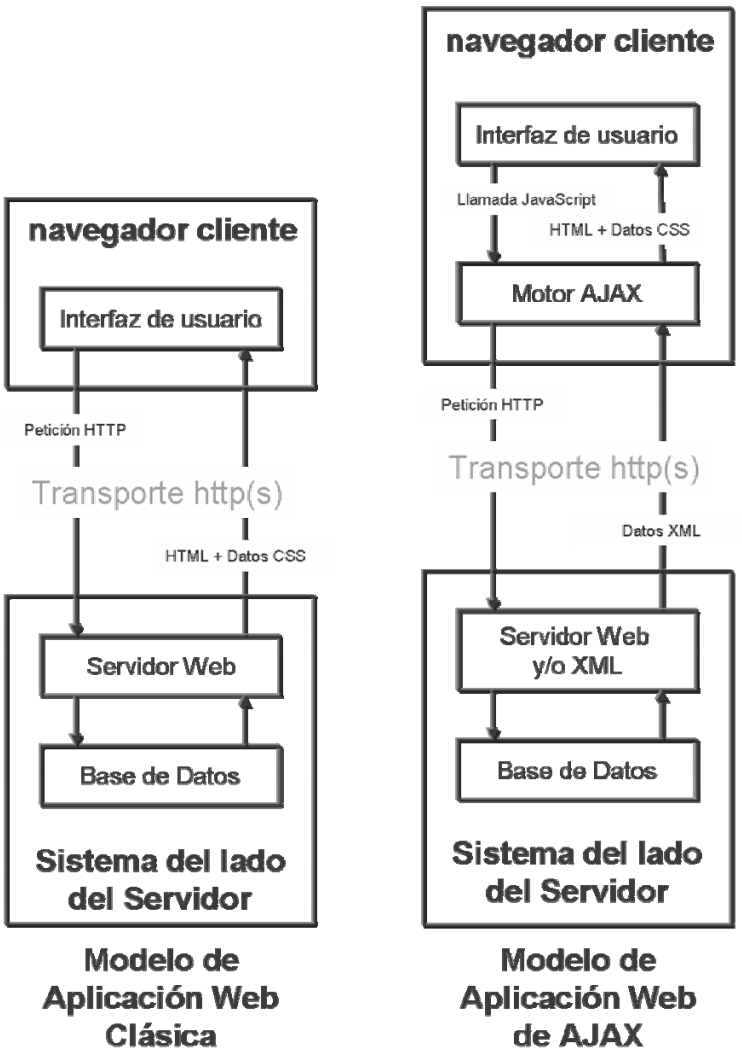
AJAX, acrónimo inglés de *Asynchronous Javascript and XML* (Javascript y XML asíncrono) es una técnica de desarrollo Web para crear aplicaciones ricas. (21)

AJAX propone el desarrollo de aplicaciones Web de la siguiente forma:

- Presentación basada en estándares: *XHTML* y *CSS*
- Cambios dinámicos en la visualización del contenido mostrado y control de eventos del usuario a través del *DOM* (Javascript)

- Intercambio de datos y manipulación usando *XML* y *XSLT*
- Obtención de datos de forma asíncrona a través de *XMLHttpRequest*
- *Javascript* para unirlo todo

Fig 2. 2 Modelo clásico frente a alternativa usando AJAX



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. (21)

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). (21)

2.10 Conclusiones

En el presente capitulo se realizó un análisis de la metodología, así como las herramientas y lenguaje a utilizar para dar solución al problema propuesto. Se fundamentó de forma concreta las ventajas e importancia que presentan las tecnologías actuales que más se usan para desarrollar aplicaciones para controlar la seguridad de diferentes sistemas.

CAPÍTULO 3 Presentación de la solución propuesta.

3.1 Introducción

Este capitulo presenta la solución a la problemática planteada en la introducción y capitulo 1 haciendo uso de las herramientas seleccionadas para dar respuesta a la misma.

Se exponen mediante un modelo del dominio los principales, objetos, eventos, entidades y participantes que describen la situación actual. Se muestran las funcionalidades que cubrirá el sistema en desarrollo, así como las cualidades del mismo mediante una descripción exhaustiva. Se modela la solución mediante diagramas de casos de uso del sistema describiéndose cada uno, y se presentan además los subsistemas formados.

3.2 Entorno donde trabajará el sistema

3.2.1 Principales conceptos, eventos, objetos y participantes del entorno del dominio.

Facultad: Se considera el entorno físico donde tiene lugar el dominio

Sistema web: Representa todas las aplicaciones que apliquen y se rijan por los conceptos y características de sitio web /aplicación web.

Recursos: Concepto agrupador que se refiere a datos, base de datos, componentes, y otros a las que un usuario puede acceder en dependencia de su rol.

Permisos: Pueden limitar o dar acceso sobre un recurso. Se le otorgan a los usuarios en dependencia de los roles.

Rol: Concepto que determina los permisos que tiene el usuario sobre un objeto. Papel que juegan los usuarios dentro del sistema.

Usuario: Interactua con una aplicación y juega un rol determinado en un sistema.

Módulo: Estructura que contiene recursos. Se utiliza para dividir los sistemas software, llamados también subsistemas.

Acción: Se realiza sobre un recurso. Los usuarios pueden realizar una acción en dependencia de su rol y los permisos que éste le conceda.

Administrador: Usuario que juega rol con todos los permisos posibles sobre los recursos.

Bitácora: Entidad en la que se registran todas las acciones de los usuarios. También se le puede llamar Historial.

3.2.2 Diagrama de clases del Modelo de Dominio

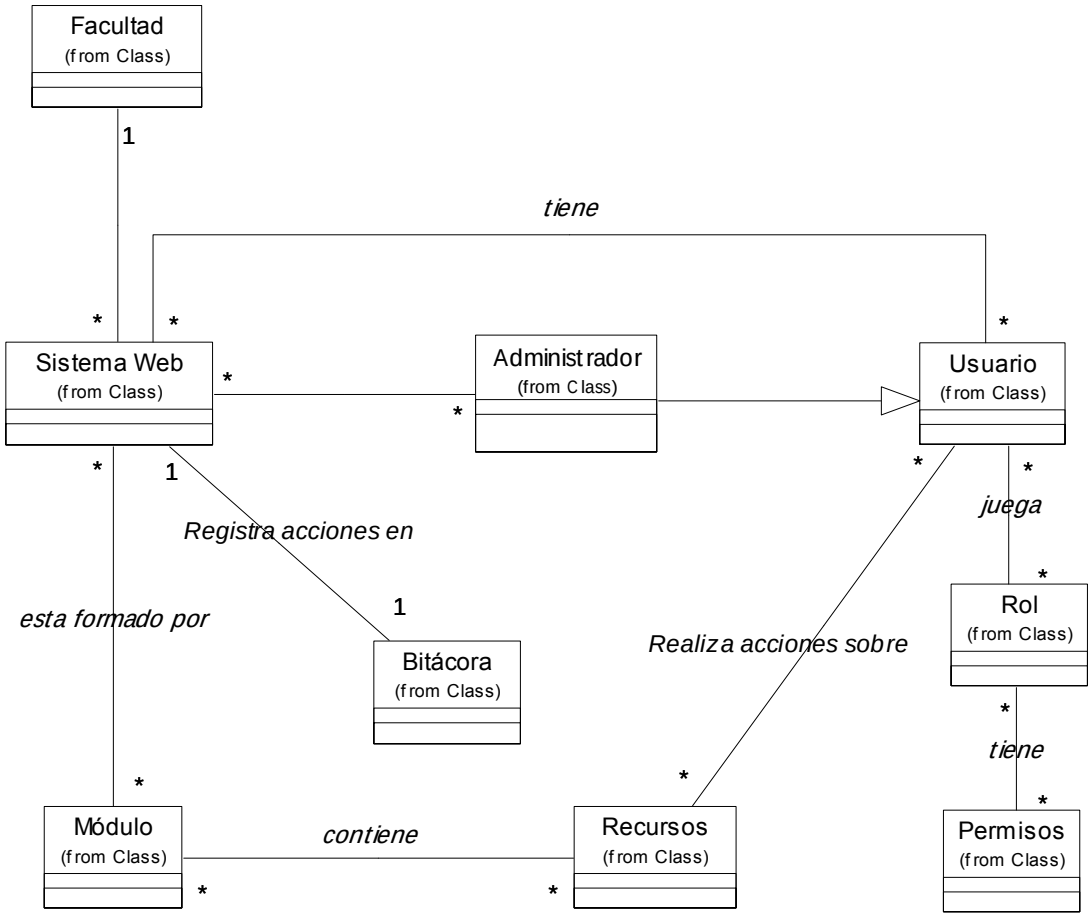
Un modelo del dominio captura los tipos más importantes de objetos que existen o los eventos que suceden en el entorno donde estará el sistema.

Las clases del dominio aparecen en 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 un seguimiento.
- Sucesos que ocurrirán o han ocurrido.

El modelo del dominio se representa mediante diagramas de UML, especialmente mediante diagramas de clases; y muestra las clases y sus relaciones mediante asociaciones.

Fig 3. 1 Diagrama de clases del Modelo de dominio



3.3 Especificación de Requisitos

En éste epígrafe se relaciona la lista de requerimientos funcionales y no funcionales del sistema Securitas.

3.3.1 Requerimientos funcionales

Los requerimientos funcionales son capacidades o condiciones que debe cumplir el sistema, los cuales están enumerados a continuación:

R1. Autenticar administrador

1.1. Solicitar usuario, contraseña y dominio (UCI o local)
1.2. Comprobar credenciales suministradas.
1.3. Dar o denegar acceso al sistema de acuerdo a la validez de las credenciales suministradas.

R2. Cerrar sesión del administrador autenticado.

R3. Gestionar aplicaciones.

3.1. Registrar nueva aplicación en el sistema.
3.1.1. Solicitar nombre de la aplicación, descripción, estado (habilitada o inhabilitada) tiempo máximo de inactividad de sus usuarios y direcciones IP de acceso.
3.2. Modificar datos de una aplicación registrada en el sistema.
3.3. Eliminar aplicación registrada en el sistema.

R4. Gestionar módulos por aplicaciones

4.1. Registrar nuevo módulo.
4.1.1. Solicitar nombre del módulo, descripción, estado (habilitado o inhabilitado) y padre(s) (aplicaciones y/ o módulos en aplicaciones).
4.2. Modificar datos de un módulo registrado.
4.3. Eliminar módulo registrado.

R5. Gestionar recursos por módulo

5.1. Registrar nuevo recurso en módulo.
5.1.1. Solicitar nombre del recurso, descripción y estado (habilitado o inhabilitado).
5.2. Modificar datos de un recurso registrado en un módulo.
5.3. Eliminar recurso registrado en un módulo.

R6. Gestionar operaciones por aplicación.

6.1. Registrar nueva operación en una aplicación.
6.1.1. Solicitar nombre de la operación y descripción.
6.2. Modificar datos de una operación registrada en una aplicación.
6.3. Eliminar una operación registrada en una aplicación.

R7. Gestionar permisos por aplicación.

7.1. Definir nuevo permiso por aplicación.
7.1.1. Solicitar nombre del permiso, descripción, recursos asociados y operaciones asociadas.
7.2. Modificar datos de un permiso definido por aplicación.
7.3. Eliminar permiso definido de una aplicación.

R8. Gestionar roles por aplicación

8.1. Definir nuevo rol en una aplicación.
  8.1.1. Solicitar nombre del rol, descripción y permisos asociados.
8.2. Modificar datos de un rol definido en una aplicación.
8.3. Eliminar rol definido en una aplicación.

R9. Gestionar usuarios por aplicaciones

9.1. Registrar nuevo usuario.
  9.1.1. Solicitar usuario del dominio UCI, estado (Habilitado o Inhabilitado), roles (o rol) asignados.
9.2. Modificar usuario registrado previamente.
9.3. Eliminar usuario registrado previamente.

R10. Administrar bitácora

10.1. Configurar bitácora.
  10.1.1. Por periodo de permanencia de registros en la bitácora.
  10.1.2. Por eventos a auditar.
10.2. Eliminar manualmente entrada de registro de la bitácora.

R11. Generar reporte de módulos

11.1. Filtrar datos de módulos por nombre, descripción, estado, fecha de registro, aplicación, módulo.
11.2. Mostrar reporte en pantalla.

R12. Generar reporte de recursos

12.1. Filtrar datos de recurso por nombre, descripción, estado, fecha de registro, módulo, aplicación.
12.2. Mostrar reporte en pantalla.

R13. Generar reporte de usuarios

13.1. Filtrar datos de usuario por usuario de la uci, estado, fecha de registro, rol, aplicación.
13.2. Mostrar reporte en pantalla.

R14. Generar reporte de roles

14.1. Filtrar datos de roles por nombre, descripción, fecha de registro, aplicación, usuarios
14.2. Mostrar reporte en pantalla.

R15. Generar reporte de aplicaciones

15.1. Filtrar datos de aplicaciones por nombre, descripción, estado, fecha de registro, dirección IP de acceso.
15.2. Mostrar reporte en pantalla.

R16. Generar reporte de bitácora (registros).

16.1. Filtrar datos de registros de bitácora por acceso, aplicación, módulo, recurso, usuario, operación, fecha.

16.2. Mostrar reporte en pantalla.

R17. Gestionar cuentas de administración por dominio UCI

17.1. Adicionar cuenta de administración por dominio UCI

17.1.1. Solicitar usuario del dominio UCI.

17.2. Eliminar cuenta de administración por dominio UCI

R18. Gestionar cuenta de administración local.

18.1. Crear cuenta de administración local. (se realiza solo una vez)

18.1.1. Solicitar usuario de la cuenta local y contraseña.

18.2. Cambiar contraseña de cuenta local.

18.2.1. Solicitar nueva contraseña.

R19. Brindar servicio para autenticar usuario en una aplicación registrada.

19.1. Recibir petición con parámetros: usuario a autenticar, contraseña, aplicación y parámetros de conexión (Referentes a la conexión establecida entre la PC desde donde accede el usuario y la aplicación. Ej.: IP de la PC del usuario, Sistema Operativo...).

19.2. Comprobar la validez y correspondencia de las credenciales y otros datos suministrados.

19.2.1. En caso positivo: autenticar en el sistema al usuario en la aplicación especificada y devolver mensaje con identificador de autenticación (único).

19.2.2. En caso negativo: devolver mensaje de error.

19.3. Registrar acción y datos relacionados en bitácora.

R20. Brindar servicio para cerrar sesión de usuario autenticado en una aplicación registrada.

20.1. Recibir petición con parámetros: identificador de autenticación (entregado en la autenticación), usuario y aplicación (últimos dos parámetros de respaldo, para mayor seguridad).

20.2. Comprobar la validez y correspondencia de los datos suministrados.

20.2.1. En caso positivo: cerrar sesión en el sistema del usuario autenticado y devolver mensaje de acción realizada.

20.2.2. En caso negativo: devolver mensaje de error.

20.3. Registrar acción y datos relacionados en bitácora.

R21. Brindar servicio para autorizar acceso de un usuario sobre recurso de una aplicación registrada.

21.1. Recibir petición con parámetros: recurso, módulo e identificador de autenticación (entregado en la autenticación).

21.2. Comprobar la validez y correspondencia de los datos suministrados.

21.2.1. En caso positivo: devolver mensaje con operaciones que se pueden realizar sobre el recurso.
21.2.2. En caso negativo: devolver mensaje de error.
21.3. Registrar acción y datos relacionados en bitácora.

R22. Brindar servicio para registrar operación de usuario sobre recurso de una aplicación registrada.
22.1. Recibir petición con parámetros: operación, recurso e identificador autenticación (entregado en la autenticación).
22.2. Registrar operación de usuario sobre recurso de una aplicación registrada en bitácora.

3.3.2 Requerimientos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Estas propiedades se ven como las características que hacen al producto atractivo, usable, rápido y confiable; se muestran a continuación:

Requerimientos de implementación:

Se satisface con el uso de la tecnología .Net, como plataforma de desarrollo y la explotación de las facilidades que brinda Microsoft SQL Server 2000 como sistema de gestión de base de datos completamente compatibles.

Requerimientos de apariencia e interfaz externa:

Se propone una interfaz sencilla y amigable, en concordancia con la línea de diseño de la facultad 9, que sea de fácil navegabilidad y entendimiento, y posibilite realizar cómodamente la configuración de los permisos.

Requerimientos de software:

Se propone para el lado del cliente la ejecución de sistema operativo con interfaz gráfica que soporte navegadores de última generación con inclusión del objeto xmlHttpRequest requiriéndose que las opciones Java Script esten habilitadas recomendando internet explorer 6.0 y firefox 1.5 o superior.

Para el lado del servidor la ejecución del sistema operativo Windows 2000 o XP profesional con Service Pack 2.0 o superior. Sistema Gestor de Base de Datos SQL Server 2000. Servidor web: Internet Information Services 5.1 o superior y Net Framework 1.1 o superior.

Requerimientos de Hardware:

Se propone como requerimientos de Hardware para el lado del servidor máquinas Pentium de 600MHz o superior, con 128 MB de memoria RAM y 20 GB de disco duro como mínimo.

Requerimientos de Usabilidad*:*

El sistema debe lograr dar servicios de manera eficiente simultáneamente a numerosas aplicaciones. Debe posibilitar obtener reportes acerca de información relevante. La información debe estar disponible para los usuarios con acceso en todo momento.

Requerimiento Soporte:

Se debe incluir un manual con indicaciones referentes al uso del sistema y acerca de sus servicios. En caso de requerirlo debe darse entrenamiento al personal que de cierta forma se relacione con el sistema y sus servicios.

Requerimientos de Rendimiento:

Se debe lograr que al realizar una acción externa sobre el sistema, en caso de los servicios web que brindará Securitas, el tiempo de respuesta no debe exceder a los 15 segundos y para la aplicación web no debe exceder a los 2 minutos.

Requerimientos de Seguridad:

Securitas, debe cumplir con los principios básicos de seguridad de cualquier sistema informático manteniendo la integridad, confidencialidad y disponibilidad de la información. El sistema debe implementar mecanismo orientados a la prevención, detección y recuperación para el caso de fallas enunciados en las medidas propuestas en el anexo 4.

3.4 Descripción del Sistema Propuesto

3.4.1 Descripción de los actores

Tabla 3.1 Actores del sistema

Nombre del actor	Descripción
Administrador de Securitas	Representa al administrador del sistema de seguridad de la facultad (Securitas).
Sistema Web	Representa a los sistemas web que van a utilizar los servicios del sistema de seguridad de la facultad (Securitas).

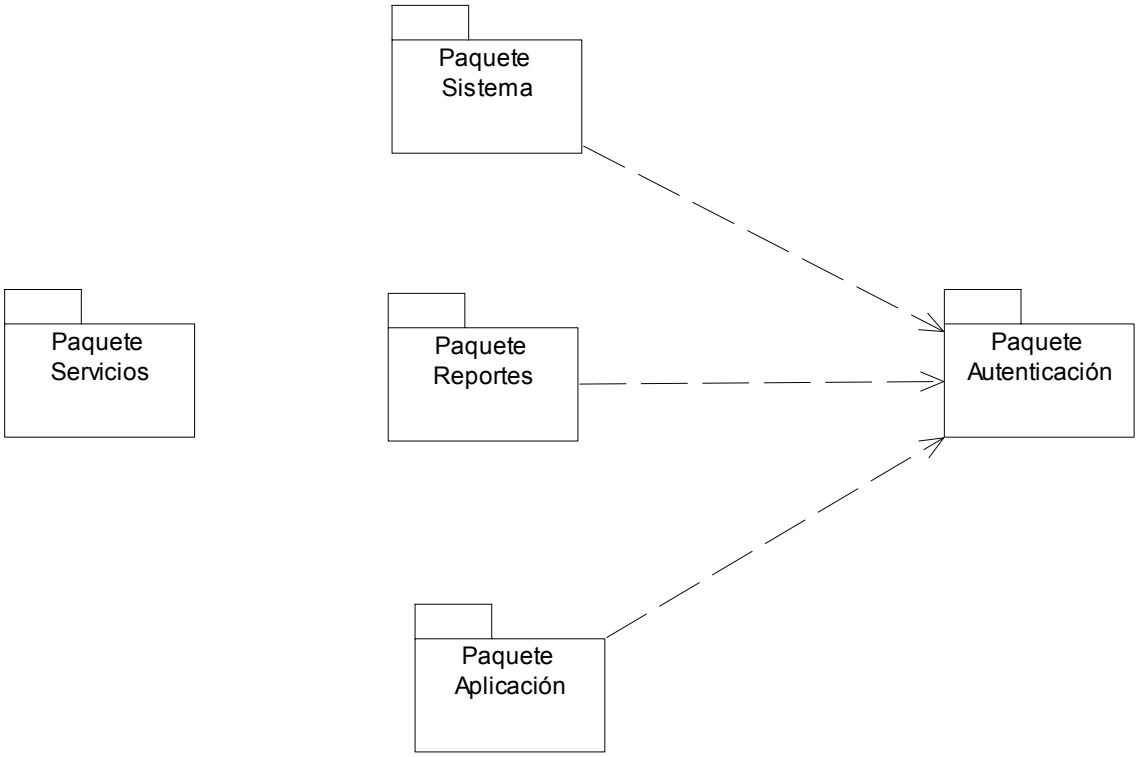
3.4.2 Casos de Uso del Sistema

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.

3.4.2.1 Paquetes y sus relaciones.

Atendiendo a las características similares entre los casos de usos y con el objetivo de comprender mejor la representación de los diagramas de casos de usos del sistema, se decidió agrupar los mismos en diferentes paquetes: Servicios, Aplicación, Sistema, Reporte y Autenticación. Los paquetes y sus relaciones se representan como se muestran en la figura:

Fig 3. 2 Paquetes contenedores de los casos de uso.



3.4.2.2 *Diagramas de Casos de Uso del Sistema.*

Un diagrama de casos de uso del sistema representa gráficamente a los procesos y su interacción con los actores. A continuación se muestran los diagramas de casos de uso del sistema Securitas, los cuales fueron agrupados por paquetes como se mencionó anteriormente.

Fig 3. 3 Paquete Autenticación.

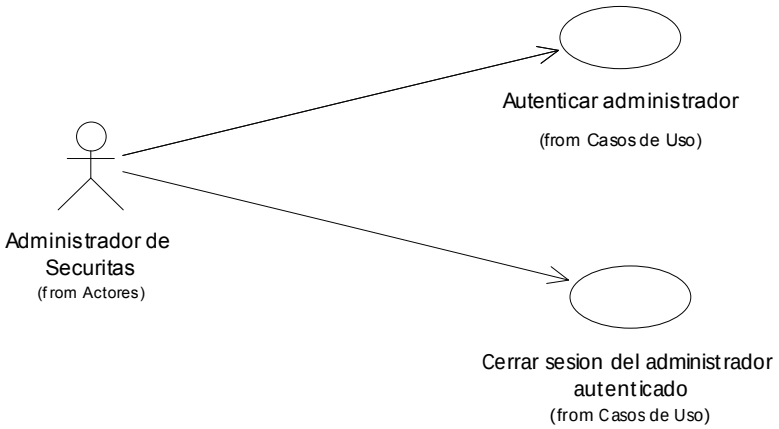


Fig 3. 4 Paquete Aplicación

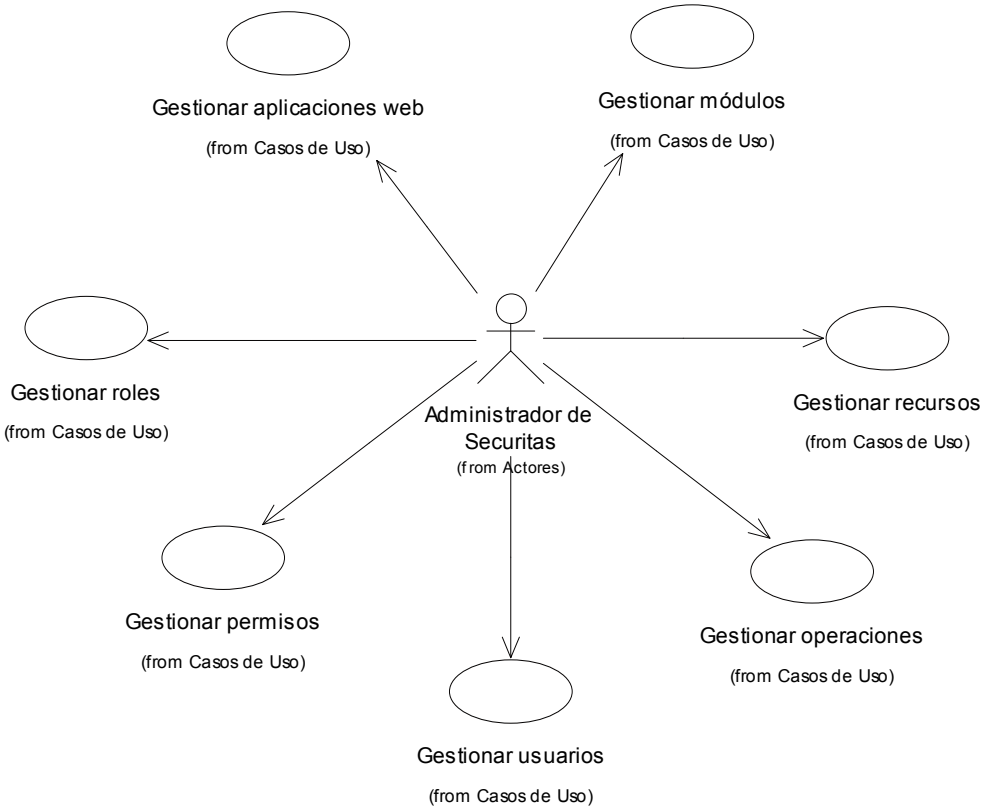


Fig 3. 5 Paquete Reportes.

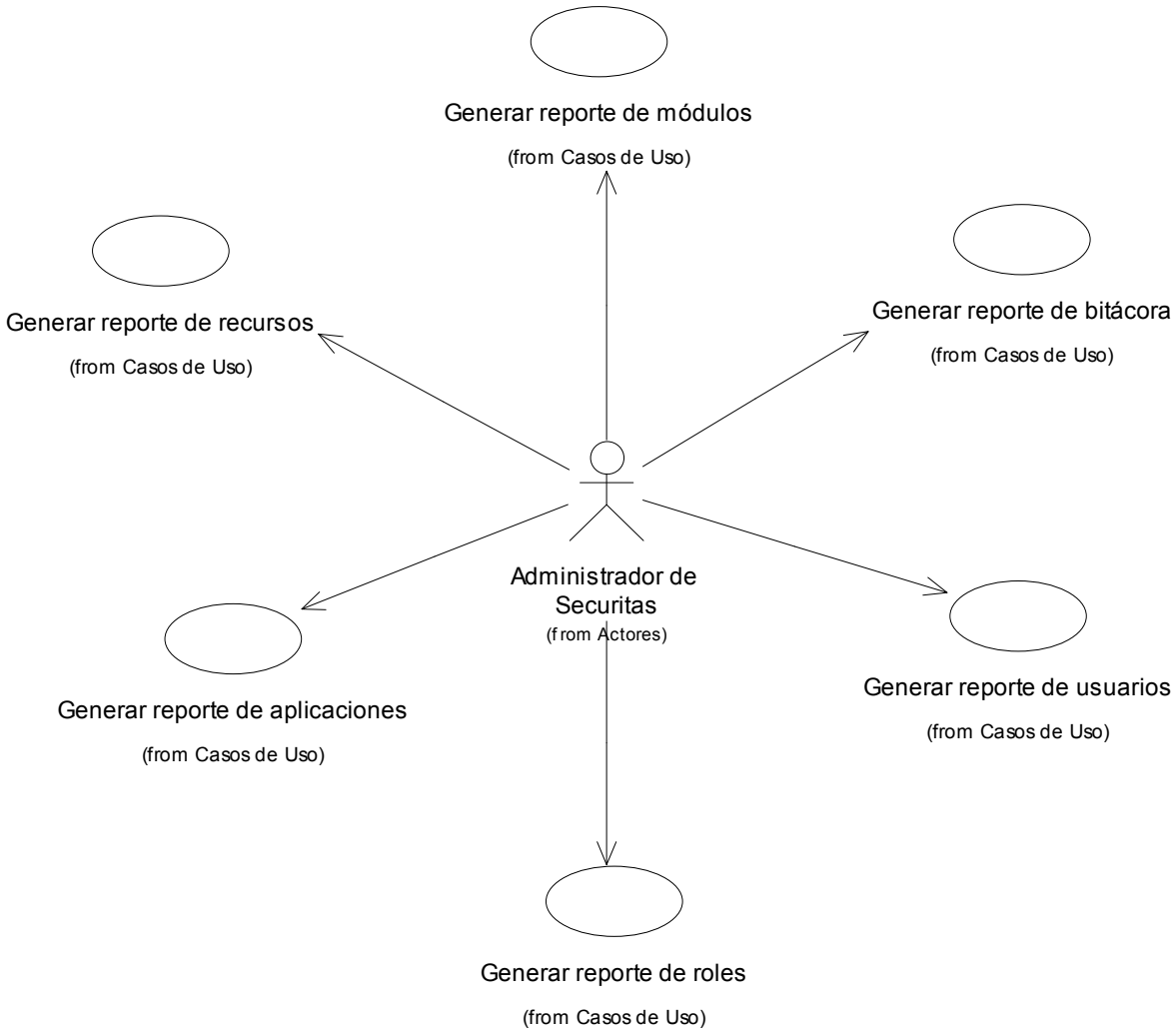


Fig 3. 6 Paquete Servicios.

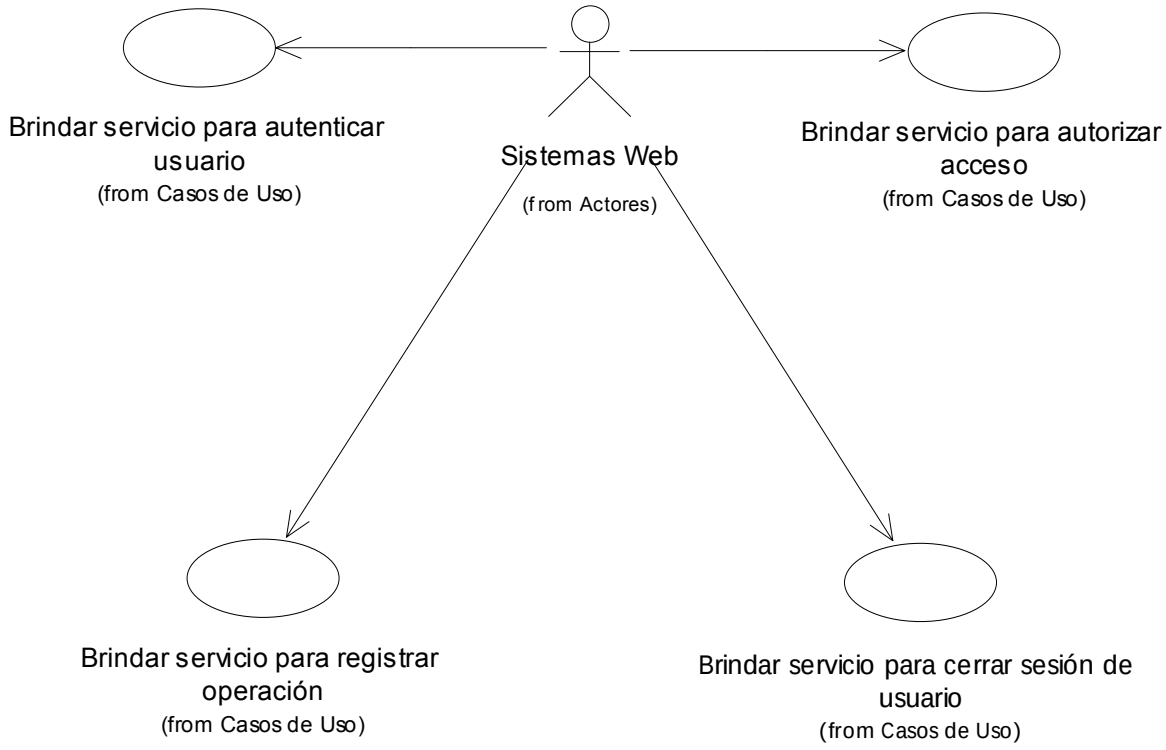
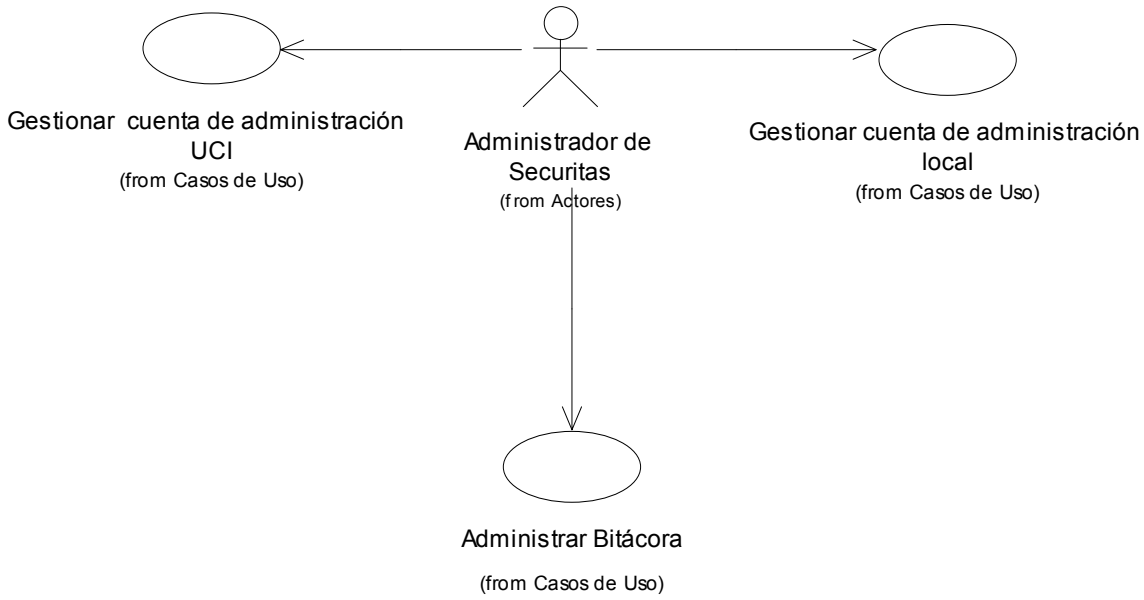


Fig 3. 7 Paquete Sistema.



3.4.3 Descripción de Casos de Uso del Sistema.

A continuación se describen algunos de los casos de uso críticos para el sistema, el resto se expone en el Anexo 1:

Tabla 3. 2 Descripción del CU: Autenticar administrador

Caso de uso:	**Autenticar administrador** **(Sección principal)**
Actores:	Administrador de Securitas
Propósito:	Permitir la entrada del administrador al sistema.
Resumen:	El administrador de Securitas ingresa usuario, contraseña, y selecciona el dominio en que desea trabajar. El sistema valida los datos suministrados

	permitiendo, si los datos son correctos, la entrada al sistema.
Tipo:	Crítico y esencial.
Referencias:	R1
Precondiciones:	
Poscondiciones:	Queda autenticado el administrador de Securitas en el sistema.

Curso normal de eventos	
Acción del actor:	**Respuesta del sistema:**
1 El caso de uso inicia cuando el administrador de Securitas desea acceder al sistema.	**1.1** El sistema solicita la entrada de usuario, contraseña y la selección del dominio local o UCI.
2 El administrador de Securitas introduce los datos correspondientes y selecciona dominio. a) Si se seleccionó dominio local ver *sección Autenticar por dominio local.* b) Si se seleccionó dominio UCI ver *sección Autenticar por dominio UCI.*	
Sección: Autenticar por dominio local	
Curso normal de eventos	
	2.1 El sistema verifica que el usuario y contraseña introducidos sean los correctos para el dominio local.

	2.2 El sistema permite la entrada.
Cursos alternativos	
	2.1 Si el usuario y/o contraseña no se corresponden con los existentes para el dominio local el sistema muestra un mensaje de error.
	2.2 El sistema no permite la entrada.
Sección: Autenticar por dominio UCI	
Curso normal de eventos	
	2.1 El sistema verifica que el usuario pertenezca al sistema.
	2.2 El sistema verifica con el dominio UCI si el usuario y la contraseña son correctos.
	2.3 El sistema permite la entrada.
Cursos alternativos	
	2.1 Si el usuario no pertenece al sistema este muestra un mensaje de error.
	2.2 Si el usuario y/o contraseña no se corresponden con los existentes para el dominio UCI el sistema muestra un mensaje de error.
	2.3 El sistema no permite la entrada.

Tabla 3. 3 Descripción del CU: Gestionar aplicaciones web

Caso de uso:	**Gestionar aplicaciones web.** **(Sección principal)**
Actores:	Administrador de Securitas
Propósito:	Permitir registrar, modificar o eliminar aplicaciones en el sistema.
Resumen:	El administrador de Securitas selecciona una opción para gestionar aplicaciones (Registrar nueva, Modificar o Eliminar). El sistema evalúa la posible realización de la acción comprobando la validez de los datos de entrada. El sistema permite la realización efectiva o no de la acción en dependencia de la validez de los datos.
Tipo:	Crítico y esencial.
Referencias:	R3
Precondiciones:	El Administrador de Securitas se encuentra autenticado en el sistema.
Poscondiciones:	Se actualiza la información en el sistema referente a aplicaciones.

Curso normal de eventos	
Acción del actor:	**Respuesta del sistema:**
1 El caso de uso inicia cuando el administrador de Securitas desea gestionar aplicaciones, seleccionando la opción correspondiente.	**1.1** El sistema muestra un listado de las aplicaciones registradas y las opciones de Registrar nueva, Modificar y Eliminar.

2 El administrador de Securitas escoge una de las opciones. a) Si escoge Registrar nueva ver sección *Registrar nueva aplicación.* b) Si escoge Modificar ver *sección Modificar aplicación* c) Si escoge Eliminar ver *sección Eliminar aplicación*.	
Sección: Registrar nueva aplicación	
Curso normal de eventos	
	2.1 El sistema muestra los campos requeridos para registrar una nueva aplicación: nombre de la aplicación, descripción, tiempo máximo de inactividad de sus usuarios, IPs de acceso y estado.
3 El administrador de Securitas llena los campos requeridos y acepta.	**3.1** El sistema verifica que los datos sean válidos.
	3.2 El sistema registra la nueva aplicación y muestra un mensaje que confirma el desarrollo exitoso de la operación.
Cursos alternativos	
	3.1 Si los datos introducidos no son válidos se muestra un mensaje de error.

Sección: Modificar aplicación	
Curso normal de eventos	
3 El administrador de Securitas selecciona la aplicación que desea modificar.	**3.1** El sistema verifica que se ha seleccionado una aplicación.
	3.2 El sistema muestra los datos actuales y modificables de la aplicación permitiendo edición sobre estos: nombre de la aplicación, descripción, tiempo máximo de inactividad de sus usuarios, IPs de acceso y estado.
4 El administrador de Securitas realiza las modificaciones necesarias y acepta.	**4.1** El sistema verifica que los datos sean válidos.
	4.2 El sistema actualiza los datos de la aplicación y muestra un mensaje que confirma el desarrollo exitoso de la operación.
Cursos alternos	
	3.1 Si no se ha seleccionado una aplicación el sistema muestra mensaje de error.
	4.1 Si los datos entrados no son válidos se muestra un mensaje de error.
Sección: Eliminar aplicación	
Curso normal de eventos	

3 El administrador de Securitas selecciona la aplicación que desea eliminar.	**3.1** El sistema verifica que se ha seleccionado una aplicación.
	3.2 El sistema muestra un mensaje de advertencia solicitando que se confirme si se desea realmente eliminar la aplicación seleccionada.
4 El Administrador de Securitas confirma que desea eliminar la aplicación seleccionada	**4.1** El sistema elimina la aplicación seleccionada.
Cursos alternos	
	3.1 Si no se ha seleccionado una aplicación el sistema muestra mensaje de error.
	4.1 El sistema no realiza ninguna operación.

Tabla 3. 4 Descripción del CU: Gestionar recursos.

Caso de uso:	**Gestionar recursos.** **(Sección principal)**
Actores:	Administrador de Securitas
Propósito:	Permitir registrar, modificar o eliminar recursos en un módulo.
Resumen:	El administrador de Securitas selecciona una opción para gestionar recursos (Registrar nuevo, Modificar o Eliminar). El sistema evalúa la posible realización de la acción comprobando la validez de los datos de entrada. El sistema permite la realización efectiva o no de la acción en

	dependencia de la validez de los datos.
Tipo:	Crítico y esencial.
Referencias:	R5
Precondiciones:	El administrador de Securitas se encuentra autenticado en el sistema. Se ha seleccionado previamente una aplicación. Se ha seleccionado un módulo dentro de la aplicación como futuro contendor de los recursos a gestionar.
Poscondiciones:	Se actualiza la información en el sistema referente a recursos.

Curso normal de eventos	
Acción del actor:	**Respuesta del sistema:**
1 El caso de uso inicia cuando el administrador de Securitas desea gestionar recursos, accediendo a la opción correspondiente.	**1.1** El sistema muestra un listado con todos los recursos incluidos en el módulo contenedor y las opciones de Registrar nuevo, Modificar y Eliminar.
2 El administrador de Securitas selecciona una de las opciones. a) Si selecciona Registrar nuevo ver *sección Registrar nuevo recurso.* b) Si selecciona Modificar ver *sección Modificar recurso.* c) Si selecciona Eliminar ver *sección Eliminar recurso*.	
Sección: Registrar nuevo recurso	

Curso normal de eventos	
	2.1 El sistema muestra los campos requeridos para registrar un nuevo recurso: nombre del recurso, descripción y estado.
3 El administrador de Securitas llena los campos requeridos y acepta.	**3.1** El sistema verifica que los datos sean válidos.
	3.2 El sistema registra el nuevo recurso y muestra un mensaje que confirma el desarrollo exitoso de la operación.
Cursos alternativos	
	3.1 Si los datos introducidos no son válidos se muestra un mensaje de error.
Sección: Modificar recurso	
Curso normal de eventos	
3 El administrador de Securitas selecciona el recurso que desea modificar.	**3.1** El sistema verifica que se ha seleccionado un recurso.
	3.2 El sistema muestra los campos de los datos actuales y modificables del recurso permitiendo edición sobre estos: nombre del módulo, descripción y estado.
4 El administrador de Securitas realiza las modificaciones necesarias y acepta.	**4.1** El sistema verifica que los datos sean válidos.

	4.2 El sistema actualiza los datos del recurso y muestra un mensaje que confirma el desarrollo exitoso de la operación.
Cursos alternos	
	3.1 Si no se ha seleccionado un recurso el sistema muestra mensaje de error.
	4.1 Si los datos entrados no son válidos se muestra un mensaje de error.
Sección: Eliminar recurso.	
Curso normal de eventos	
3 El administrador de Securitas selecciona el recurso que desea eliminar.	**3.1** El sistema verifica que se ha seleccionado un recurso.
	3.2 El sistema muestra un mensaje de advertencia solicitando que se confirme si se desea realmente eliminar el recurso seleccionado.
4 El Administrador de Securitas confirma que desea eliminar el recurso seleccionado.	**4.1** El sistema elimina el recurso seleccionado.
Cursos alternos	
	3.1 Si no se ha seleccionado un recurso el sistema muestra mensaje de error.
	4.1 El sistema no realiza ninguna operación.

Tabla 3. 5 Descripción del CU: Gestionar usuarios.

Caso de uso:	**Gestionar usuarios.** **(Sección principal)**
Actores:	Administrador de Securitas
Propósito:	Permitir registrar, modificar o eliminar usuarios en una aplicación.
Resumen:	El administrador de Securitas selecciona una opción para gestionar usuarios (Registrar nuevo, Modificar o Eliminar). El sistema evalúa la posible realización de la acción comprobando la validez de los datos de entrada. El sistema permite la realización efectiva o no de la acción en dependencia de la validez de los datos.
Tipo:	Crítico y esencial.
Referencias:	R9
Precondiciones:	El administrador de Securitas se encuentra autenticado en el sistema. Se ha seleccionado previamente aplicaciones.
Poscondiciones:	Queda actualizada la información referente a los usuarios en el sistema.

Curso normal de eventos	
Acción del actor:	**Respuesta del sistema:**
1 El caso de uso inicia cuando el administrador de Securitas desea gestionar los usuarios de las aplicaciones registradas.	**1.1** El sistema muestra todos los usuarios de aplicaciónes seleccionadas y las opciones de Registrar nuevo, Modificar y Eliminar.

2 El administrador de Securitas selecciona una de las opciones. a) Si se selecciona Registrar nuevo ver *sección Registrar nuevo usuario.* b) Si se selecciona Modificar ver *sección Modificar usuario.* c) Si se selecciona Eliminar ver *sección Eliminar usuario.*	
Sección: Registrar nuevo usuario	
Curso normal de eventos	
	2.1 El sistema muestra los campos requeridos para registrar nuevo usuario: usuario del dominio UCI, estado y todos los roles existentes.
3 El administrador de Securitas llena los campos requeridos, selecciona los roles y presiona el botón aceptar.	**3.1** El sistema verifica que los datos sean válidos.
	3.2 El sistema verifica con directorio activo UCI si el usuario existe en el dominio UCI.
	3.3 El sistema registra el nuevo usuario y muestra un mensaje que confirma el desarrollo exitoso de la operación.
Cursos alternativos	

	3.1 Si los datos entrados no son válidos se muestra un mensaje de error.
	3.2 Si al comprobar con el directorio activo UCI el usuario no existe en el dominio UCI el sistema emite un mensaje de error.
Sección: Modificar usuario	
Curso normal de eventos	
3 El administrador de Securitas selecciona el usuario que desea modificar.	**3.1** El sistema verifica que se ha seleccionado un usuario.
	3.2 El sistema muestra los campos de los datos actuales y modificables del usuario permitiendo edición: estado y roles asignados.
4 El administrador de Securitas realiza las modificaciones necesarias y acepta.	**4.1** El sistema verifica que los datos sean válidos.
	4.2 El sistema actualiza los datos del usuario y muestra un mensaje que confirma el desarrollo exitoso de la operación.
Cursos alternos	
	3.1 Si no se ha seleccionado un usuario el sistema muestra mensaje de error.
	4.1 Si los datos entrados no son válidos se muestra

	un mensaje de error.
Sección: Eliminar usuario.	
Curso normal de eventos	
3 El administrador de Securitas selecciona el usuario que desea eliminar.	**3.1** El sistema verifica que se ha seleccionado un usuario.
	3.2 El sistema muestra un mensaje de advertencia solicitando que se confirme si se desea realmente eliminar el usuario seleccionado.
4 El Administrador de Securitas confirma que desea eliminar el usuario seleccionado.	**4.1** El sistema elimina el usuario seleccionado.
Cursos alternos	
	3.1 Si no se ha seleccionado un usuario el sistema muestra mensaje de error.
	4.1 El sistema no realiza ninguna actividad.

Tabla 3. 6 Descripción del CU: Gestionar permisos.

Caso de uso:	**Gestionar permisos.** **(Sección principal)**
Actores:	Administrador de Securitas
Propósito:	Permitir definir, modificar o eliminar permisos en una aplicación.

Resumen:	El administrador de Securitas selecciona una opción para gestionar permisos (Definir nuevo, Modificar o Eliminar). El sistema evalúa la posible realización de la acción comprobando la validez de los datos de entrada. El sistema permite la realización efectiva o no de la acción en dependencia de la validez de los datos.
Tipo:	Crítico y esencial.
Referencias:	R7
Precondiciones:	El administrador de Securitas se encuentra autenticado en el sistema. Se ha seleccionado previamente una aplicación.
Poscondiciones:	Se actualiza la información en el sistema referente a permisos.

Curso normal de eventos	
Acción del actor:	**Respuesta del sistema:**
1 El caso de uso inicia cuando el administrador de Securitas desea gestionar permisos en las aplicaciones registradas.	**1.1** El sistema muestra todos los permisos definidos en la aplicación seleccionada y las opciones de Definir nuevo, Modificar y Eliminar.
2 El administrador de Securitas selecciona una de las opciones. a) Si selecciona Definir nuevo ver *sección Definir nuevo permiso.* b) Si selecciona Modificar ver *sección Modificar permiso.* c) Si selecciona Eliminar ver *sección*	

Eliminar permiso.	
Sección: Definir nuevo permiso	
Curso normal de eventos	
	2.1 El sistema muestra los campos requeridos para definir nuevo permiso: nombre del permiso, descripción, todos los recursos existentes y todas las operaciones existentes.
3 El administrador de Securitas procede a llenar los campos y adicionar los recursos y operaciones asociadas, aceptando posteriormente.	**3.1** El sistema verifica que los datos sean válidos.
	3.2 El sistema crea el nuevo permiso y muestra un mensaje que confirma el desarrollo exitoso de la operación.
Cursos alternativos	
	3.1 Si los datos introducidos no son válidos se muestra un mensaje de error.
Sección: Modificar permiso	
Curso normal de eventos	
3 El administrador de Securitas selecciona el permiso que desea modificar.	**3.1** El sistema verifica que se ha seleccionado un permiso.

	3.2 El sistema muestra campos de los datos actuales y modificables del permiso seleccionado, permitiendo edición sobre estos: nombre del permiso, descripción, recursos asociados y operaciones asociadas.
4 El administrador de Securitas realiza las modificaciones necesarias y acepta.	**4.1** El sistema verifica que los datos sean válidos.
	4.2 El sistema actualiza los datos del permiso y muestra un mensaje que confirma el desarrollo exitoso de la operación.
Cursos alternos	
	3.1 Si no se ha seleccionado un permiso el sistema muestra mensaje de error.
	4.2 Si los datos entrados no son válidos se muestra un mensaje de error.
Sección: Eliminar permiso.	
Curso normal de eventos	
3 El administrador de Securitas selecciona el permiso que desea eliminar.	**3.1** El sistema verifica que se ha seleccionado un permiso.
	3.2 El sistema muestra un mensaje de advertencia solicitando que se confirme si se desea realmente eliminar el permiso seleccionado.

4 El Administrador de Securitas confirma que desea eliminar el permiso seleccionado.	**4.1** El sistema elimina el permiso seleccionado.
Cursos alternos	
	3.1 Si no se ha seleccionado un permiso el sistema muestra mensaje de error.
	4.1 El sistema no realiza ninguna actividad.

Tabla 3. 7 Descripción del CU: Gestionar roles.

Caso de uso:	**Gestionar roles.** **(Sección principal)**
Actores:	Administrador de Securitas
Propósito:	Permitir definir, modificar o eliminar roles en una aplicación.
Resumen:	El administrador de Securitas selecciona una opción para gestionar roles (Definir nuevo, Modificar o Eliminar). El sistema evalúa la posible realización de la acción comprobando la validez de los datos de entrada. El sistema permite la realización efectiva o no de la acción en dependencia de la validez de los datos.
Tipo:	Crítico y esencial.
Referencias:	R8
Precondiciones:	El administrador de Securitas se encuentra autenticado en el sistema. Se ha seleccionado previamente una aplicación.

Poscondiciones:	Queda actualizada en el sistema la información referente a roles.
Curso normal de eventos	
Acción del actor:	**Respuesta del sistema:**
1 El caso de uso inicia cuando el administrador de Securitas desea gestionar los roles de las aplicaciones registradas.	**1.1** El sistema muestra todos los roles de la aplicación seleccionada y las opciones Definir nuevo, Modificar y Eliminar.
2 El administrador de Securitas selecciona una de las opciones. a) Si selecciona Definir nuevo ver *sección Definir nuevo rol.* b) Si selecciona Modificar ver *sección Modificar rol.* c) Si selecciona Eliminar ver *sección Eliminar rol.*	
Sección: Definir nuevo rol	
Curso normal de eventos	
	2.1 El sistema muestra los campos requeridos para definir nuevo rol: nombre del rol, descripción y todos los permisos existentes.
3 El administrador de Securitas completa los campos requeridos y acepta.	**3.1** El sistema verifica que los datos sean válidos.

	3.2 El sistema crea el nuevo rol y muestra un mensaje que confirma el desarrollo exitoso de la operación.
Cursos alternativos	
	3.1 Si los datos ingresados no son correctos se muestra un mensaje de error.
Sección: Modificar rol	
Curso normal de eventos	
3 El administrador de Securitas selecciona el rol que desea modificar.	**3.1** El sistema verifica que se ha seleccionado un rol.
	3.2 El sistema muestra los datos actuales y modificables del rol permitiendo edición sobre estos: nombre del rol, descripción y permisos asociados.
4 El administrador de Securitas realiza las modificaciones necesarias y acepta.	**4.1** El sistema verifica que los datos sean válidos.
	4.2 El sistema actualiza los datos del rol y muestra un mensaje que confirma el desarrollo exitoso de la operación.
Cursos alternos	
	3.1 Si no se ha seleccionado un rol el sistema muestra un mensaje de error.

	4.1 Si los datos entrados no son válidos se muestra un mensaje de error.
Sección: Eliminar rol.	
Curso normal de eventos	
3 El administrador de Securitas selecciona el rol que desea eliminar.	**3.1** El sistema verifica que se ha seleccionado un rol.
	3.2 El sistema muestra un mensaje de advertencia solicitando que se confirme si se desea realmente eliminar el rol seleccionado.
4 El Administrador de Securitas confirma que desea eliminar el rol seleccionado.	**4.1** El sistema elimina el rol seleccionado.
Cursos alternos	
	3.1 Si no se ha seleccionado un rol el sistema muestra mensaje de error.
	4.1 El sistema no realiza ninguna actividad.

Paquete Servicios

Tabla 3. 8 Descripción del CU: Brindar servicio para autenticar usuario.

Caso de uso:	**Brindar servicio para autenticar usuario.** **(Sección principal)**

Actores:	Sistema web.
Propósito:	Autenticar un usuario en una aplicación registrada.
Resumen:	Una aplicación a la que se le brinda servicios de seguridad (Sistema web) solicita que se autentique un usuario. Securitas chequea la validez del pedido y autentica devolviendo un identificador de autenticación.
Tipo:	Crítico y esencial.
Referencias:	R19
Precondiciones:	
Poscondiciones:	Se ha autenticado al usuario a solicitud de la aplicación.

Curso normal de eventos	
Acción del actor:	**Respuesta del sistema:**
1 El caso de uso inicia cuando una aplicación a la que se le brinda servicios de seguridad (Sistema web) solicita que se autentique un usuario brindando los parámetros: usuario a autenticar, contraseña, aplicación y parámetros de conexión (Referentes a la conexión establecida entre la PC desde donde accede el usuario y la aplicación. Ej.: IP de la PC del usuario, Sistema Operativo...).	**1.1** El sistema verifica que la aplicación esté registrada.
	1.2 Comprueba que el IP de la PC que hace el pedido coincide con alguno de los IPs de acceso que tiene como dato la aplicación registrada.

	1.3 El sistema verifica que el usuario está registrado en la aplicación.
	1.4 El sistema verifica con directorio activo UCI si el usuario y la contraseña son correctos.
	1.5 El sistema genera un identificador de autenticación único y registra con el mismo al usuario, aplicación, parámetros de la conexión y fecha/hora actual, como entrada de autenticación. Devolviendo el identificador de autenticación.
Cursos alternativos	
	1.1 Si la aplicación no está registrada el sistema emite mensaje de error.
	1.2 Si el IP de la Pc que hace el pedido no coincide con alguno de los IPs de acceso que tiene como dato la aplicación registrada el sistema emite mensaje de error.
	1.3 Si el usuario no está registrado en la aplicación el sistema emite mensaje de error.
	1.4 Si al comprobar con el directorio activo UCI el usuario y/o contraseña no son correctos el sistema emite un mensaje de error.

Tabla 3. 9 Descripción del CU: Brindar servicio para autorizar acceso.

Caso de uso:	**Brindar servicio para autorizar acceso.** **(Sección principal)**
Actores:	Sistema web
Propósito:	Brindar servicio para autorizar acceso de un usuario sobre recurso en aplicación registrada en Securitas
Resumen:	Una aplicación a la que se le brinda servicios de seguridad (Sistema web) requiere comprobar permisos de acceso de un usuario autenticado a un recurso, solicitando los servicios a Securitas.
Tipo:	Crítico y esencial.
Referencias:	R21
Precondiciones:	
Poscondiciones:	

Curso normal de eventos	
Acción del actor:	**Respuesta del sistema:**
1 El caso de uso inicia cuando una aplicación a la que se le brinda servicios de seguridad solicita comprobar acceso de un usuario autenticado a un recurso brindando los parámetros: identificador de autenticación, modulo y recurso.	**1.1** El sistema verifica que el identificador de autenticación hace referencia a algún usuario autenticado.

	1.2 El sistema verifica que el recurso y el módulo existen en la aplicación del usuario autenticado. (según el identificador de autenticación)
	1.3 El sistema verifica que el usuario autenticado tiene permisos de acceso sobre el recurso ubicado en el modulo. (según los parámetros)
	1.4 El sistema devuelve mensaje de acceso autorizado y listado de operaciones que el usuario puede realizar sobre el recurso.
	1.5 El sistema registra acciones en la bitácora.
Cursos alternativos	
	1.1 Si el identificador de autenticación no hace referencia a algún usuario autenticado se emite mensaje de error.
	1.2 Si el recurso no existe se emite mensaje de error.
	1.3 Si usuario autenticado no tiene permisos de acceso sobre el recurso se emite mensaje de error.

3.5 Conclusiones

En este capítulo se definieron todas las entidades del sistema agrupándolas en un Modelo de Dominio, detallándose a la vez los conceptos asociados al mismo. Se obtuvieron todos los requerimientos tanto funcionales como no funcionales del sistema, se identificaron los actores que intervienen, así como todos los casos de usos del sistema, que fueron descritos de forma detallada reflejando las funcionalidades recogidas en los requerimientos. Además, se explicaron a través de los diagramas correspondientes, los casos de uso a automatizar; distribuidos en forma de paquetes para un mejor entendimiento y representación. El desarrollo de este flujo de trabajo y los artefactos obtenidos a partir del mismo, nos permite pasar al diseño de las clases del sistema, el cual será representado en el próximo capítulo.

CAPÍTULO 4 Construcción de la solución propuesta.

4.1 Introducción

En este capítulo se determinan las clases necesarias para llevar a cabo las funcionalidades contenidas en los casos de uso. Se plantea la descripción del diseño de la solución propuesta a través de la representación de los diagramas de clases del diseño, se muestra además el diseño de la base de datos basado en el diagrama de clases persistentes y el modelo de datos, así como los diagramas de despliegue y componentes.

4.2 Diagramas de Clases

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar.

De forma tal que se facilite la comprensión de la relación existente entre los distintos componentes en la realización de los diversos escenarios por casos de uso se elaboró un diagrama de clases para cada uno de estos, presentándose una parte en este capítulo (los más representativos) y el resto en el Anexo 2.

Fig 4. 1 Diagrama de clases. CU Autenticar administrador.

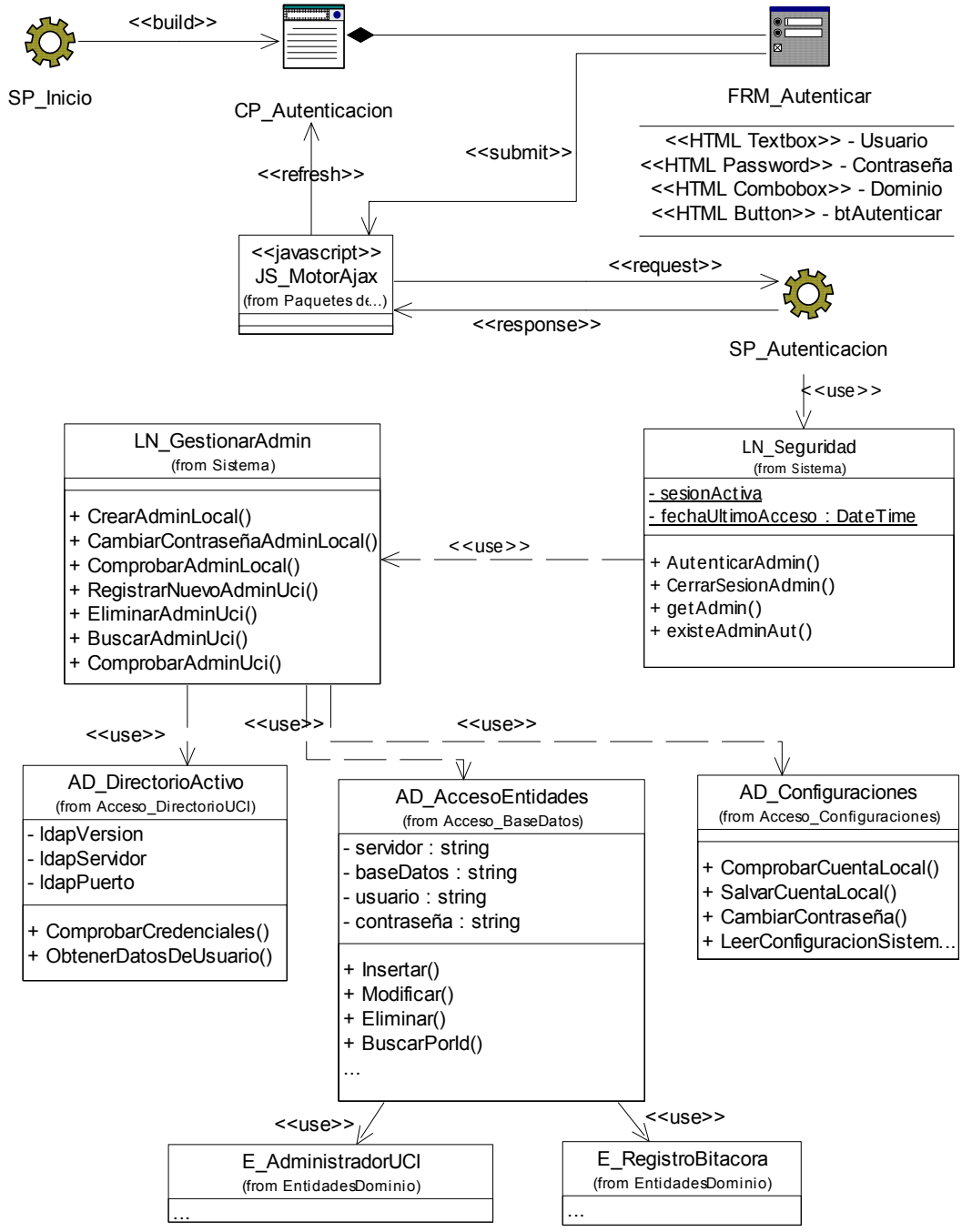


Fig 4. 2 Diagrama de clases. CU Gestionar Aplicaciones

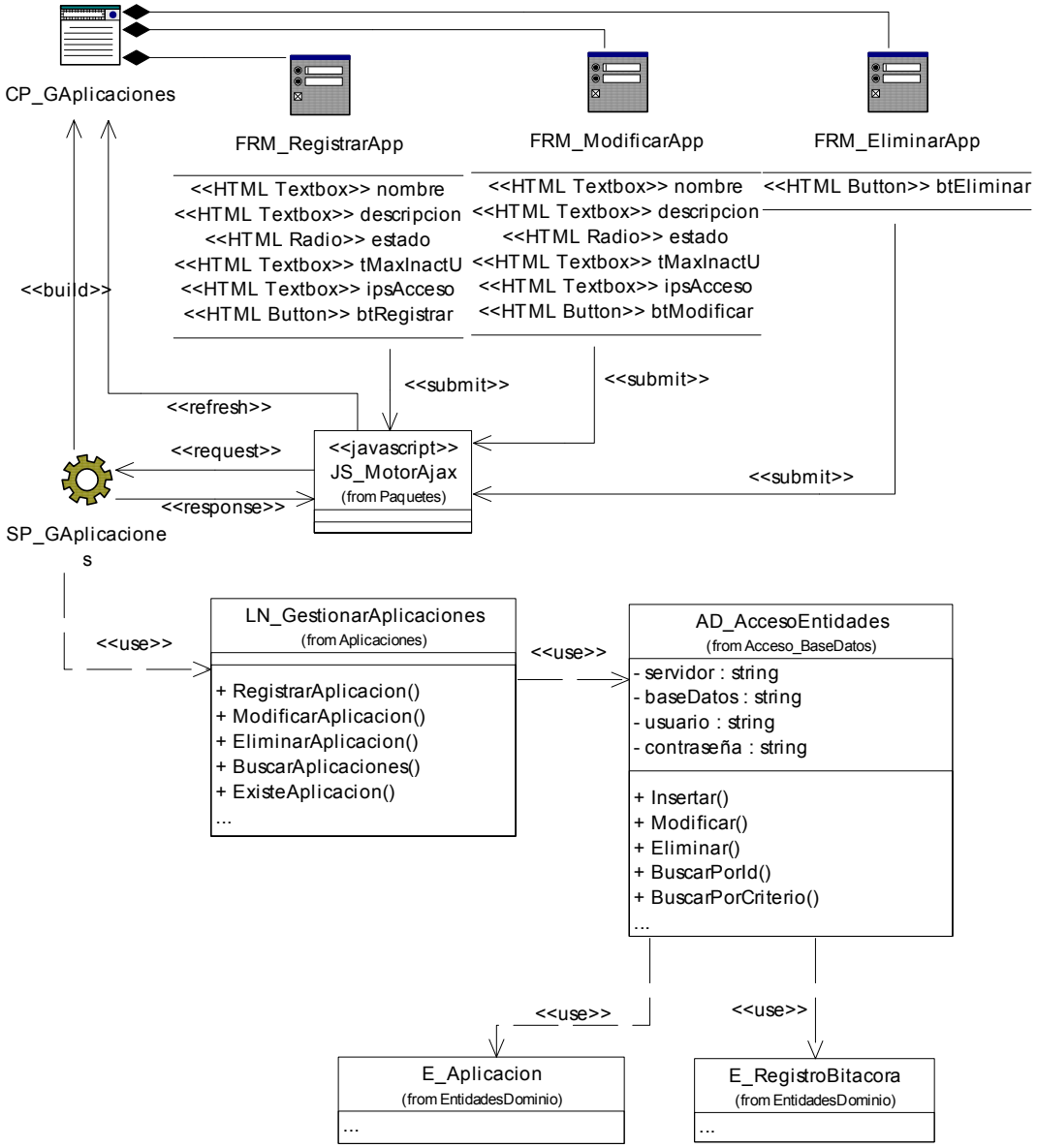


Fig 4. 3 Diagrama de Clases. CU Gestionar Recursos

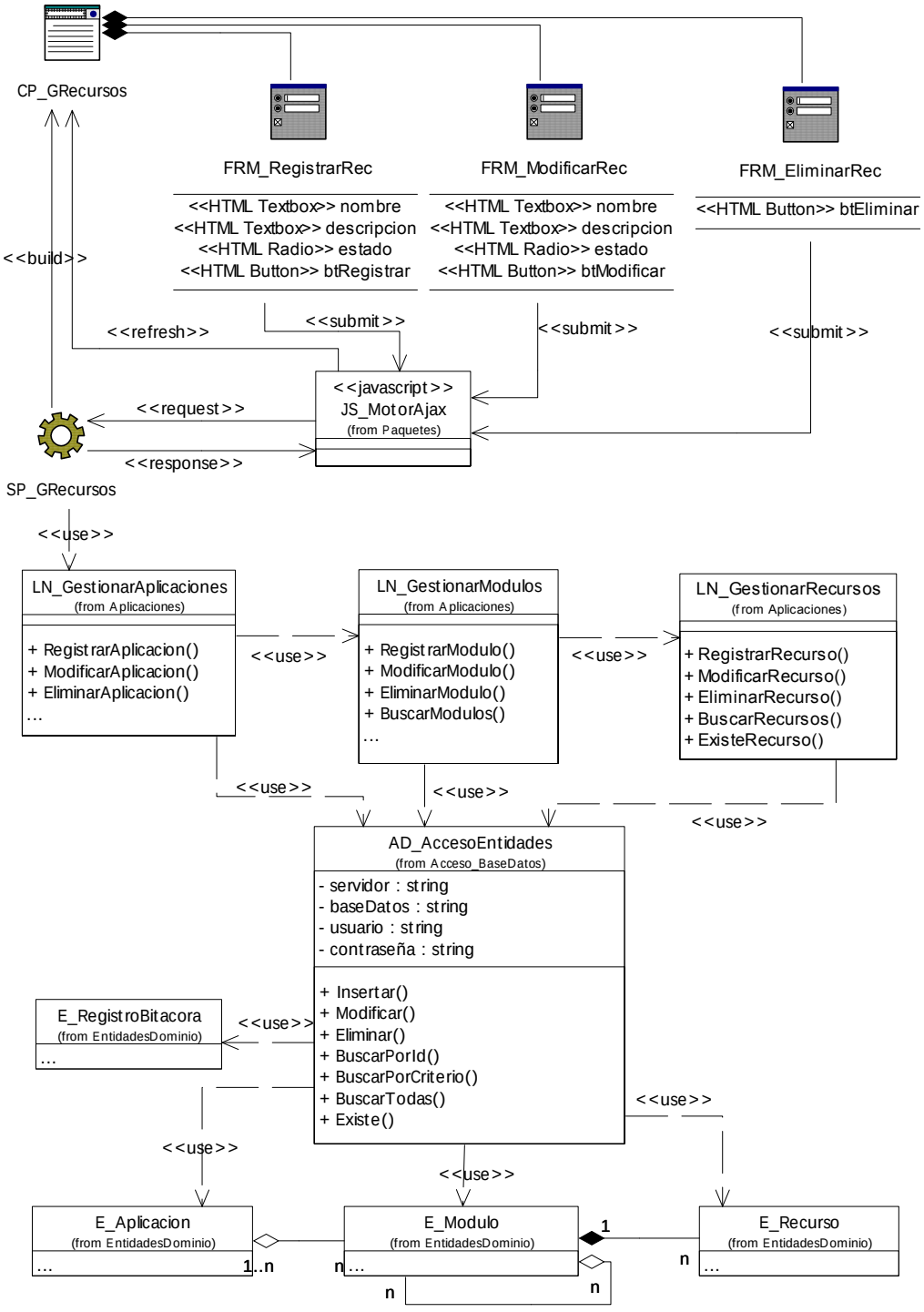


Fig 4. 4 Diagrama de clases. CU Gestionar usuarios.

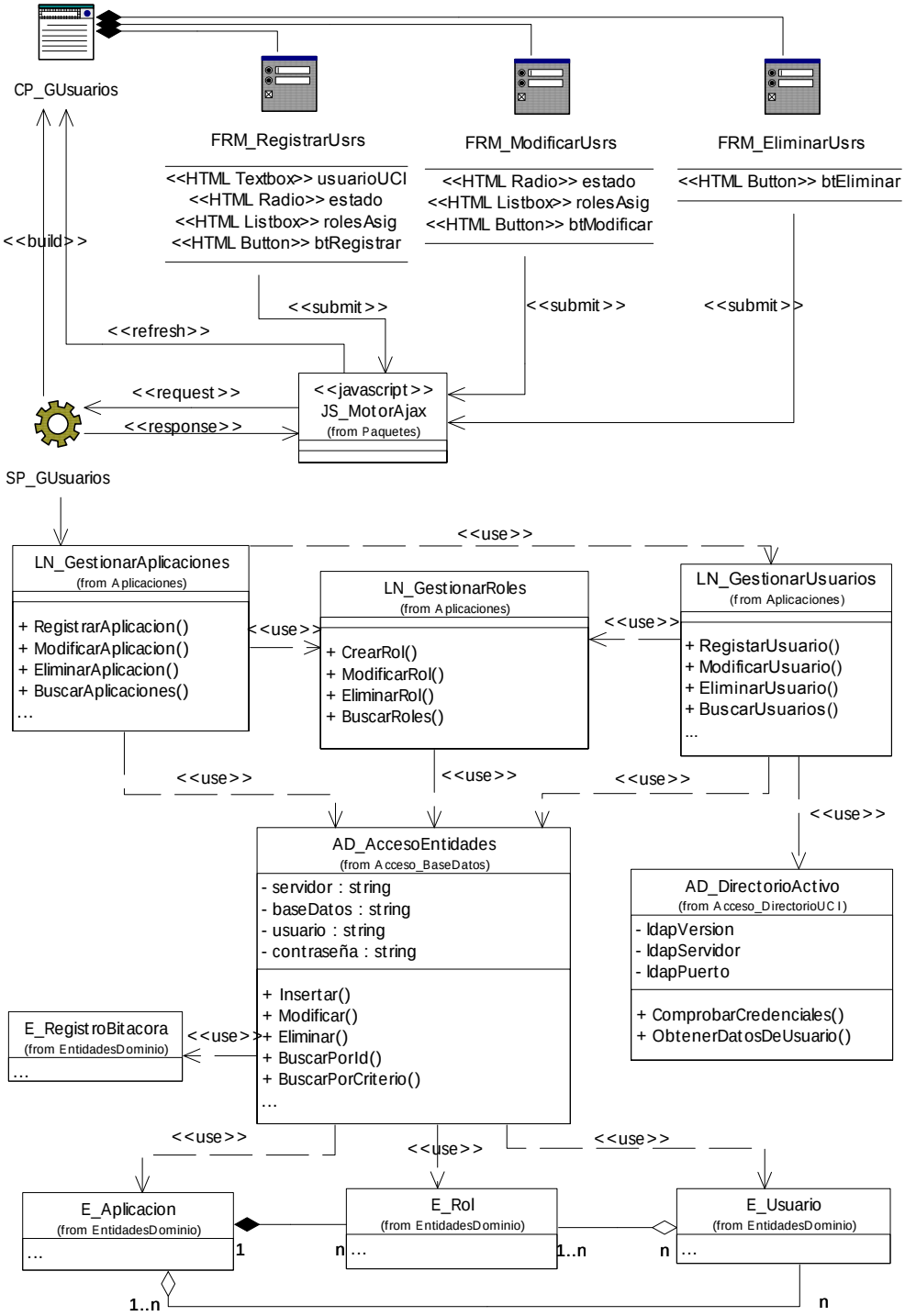


Fig 4.6 Diagrama de clases. CU Gestionar permisos.

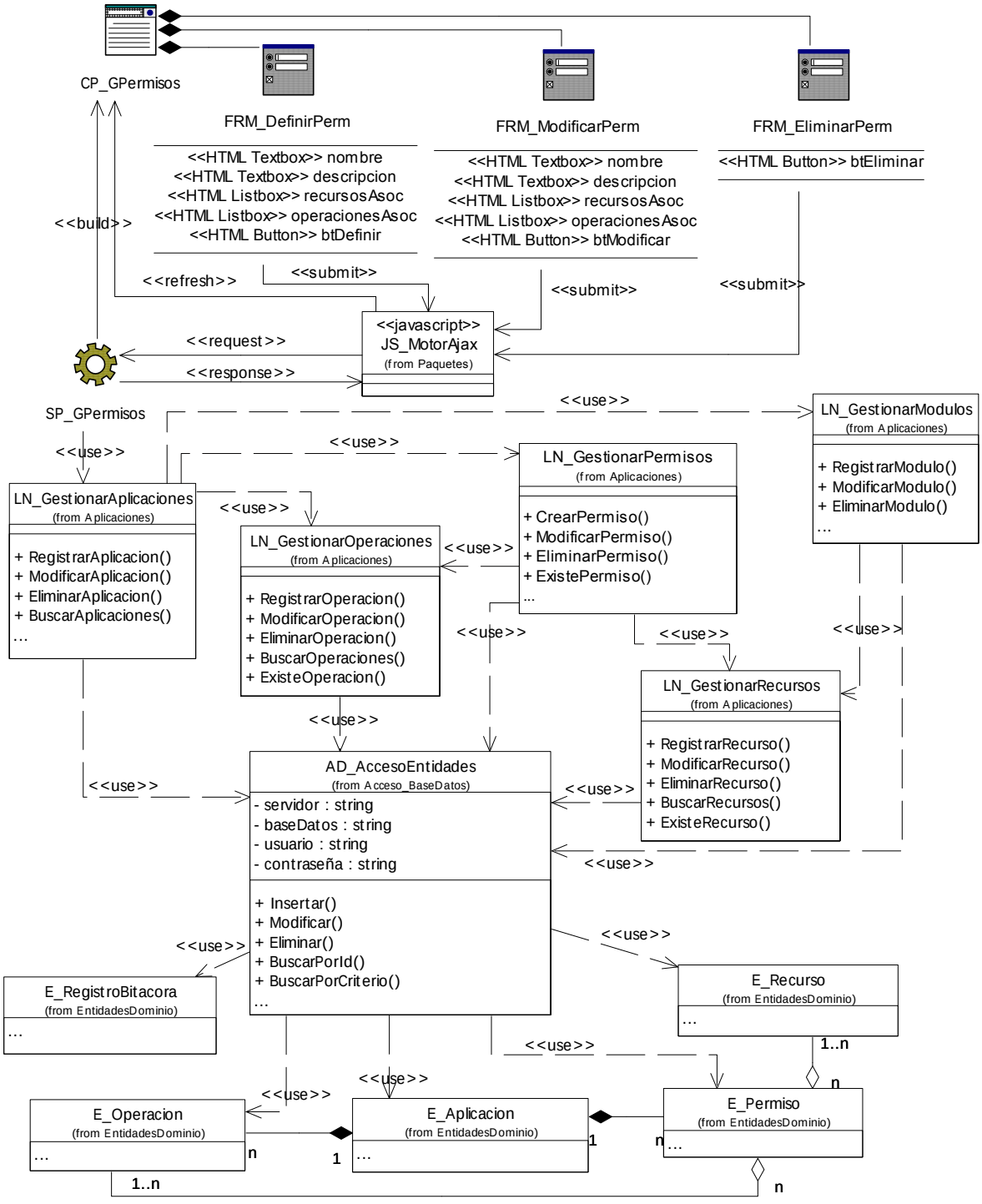


Fig 4. 5 Diagrama de clases. CU Gestionar roles.

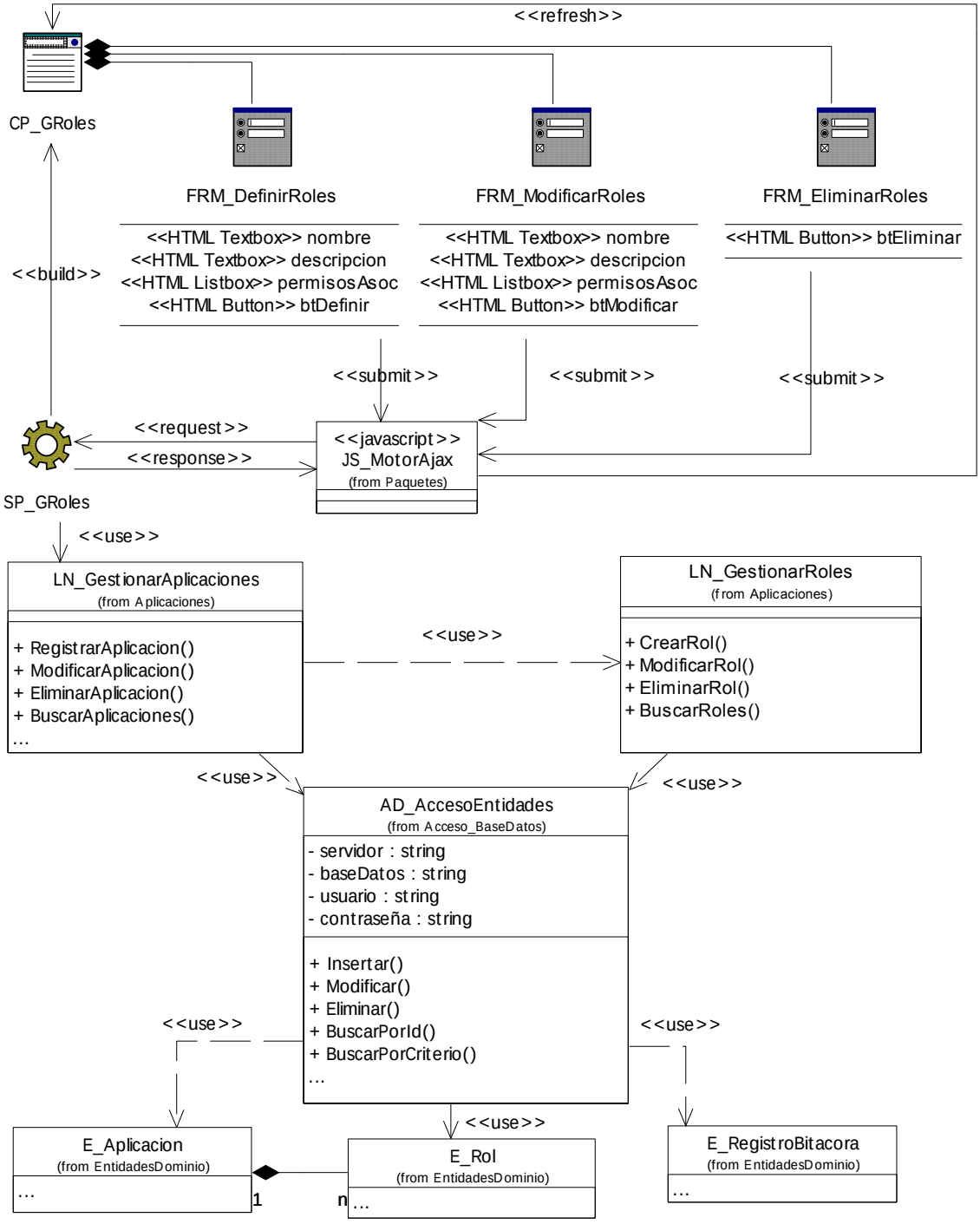


Fig 4. 6 Diagrama de clases. CU Brindar servicio para autenticar usuario.

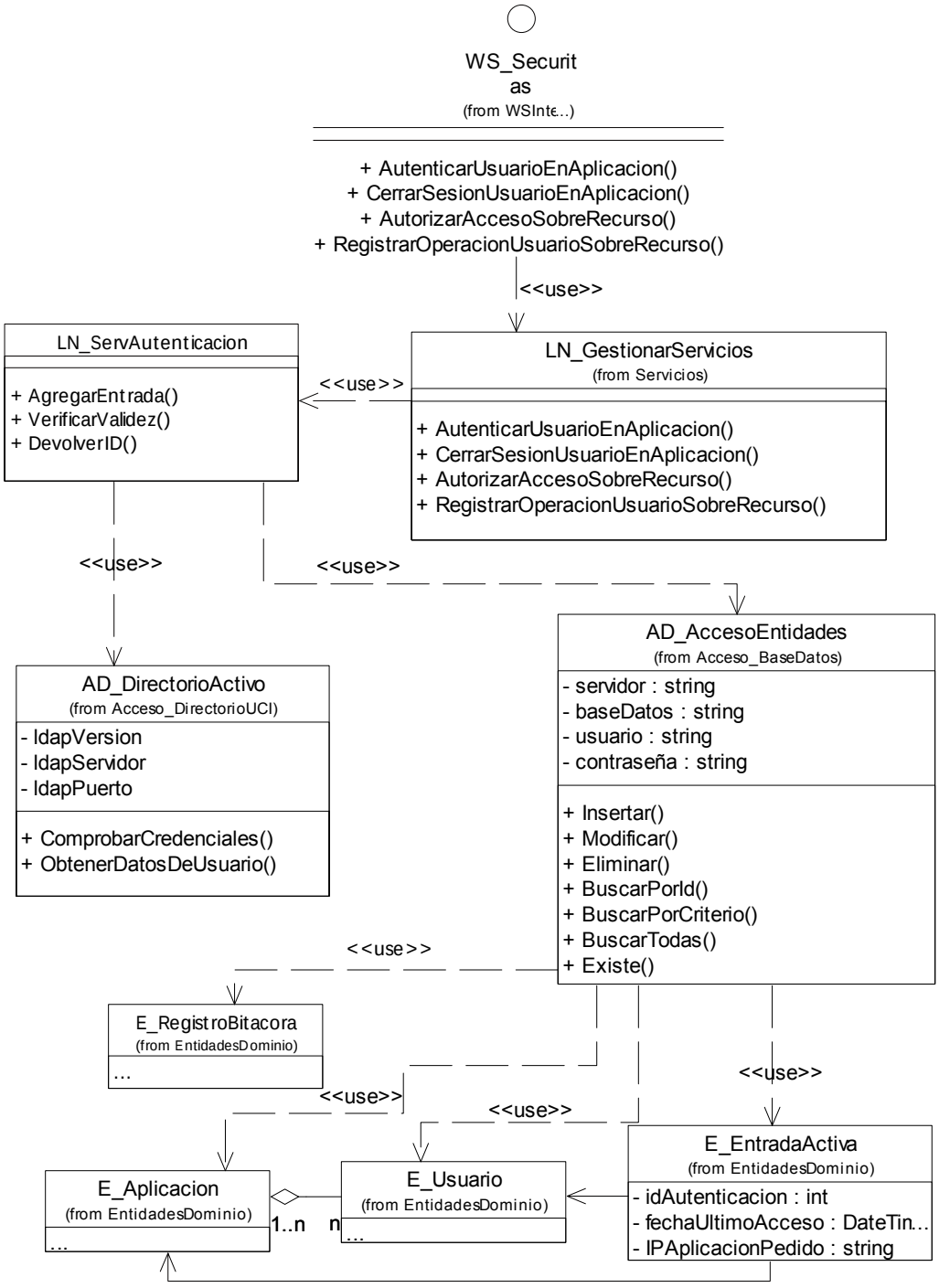
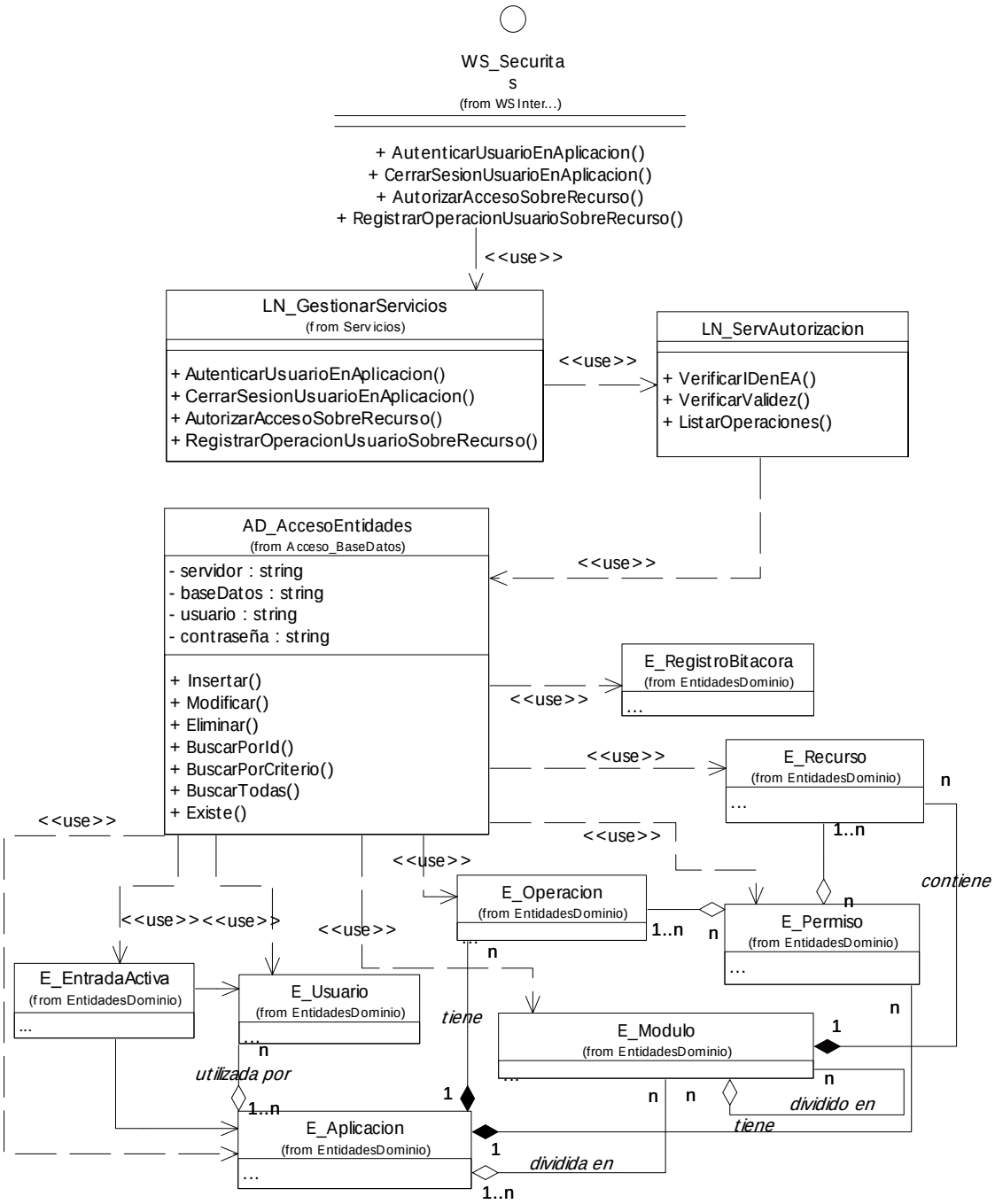


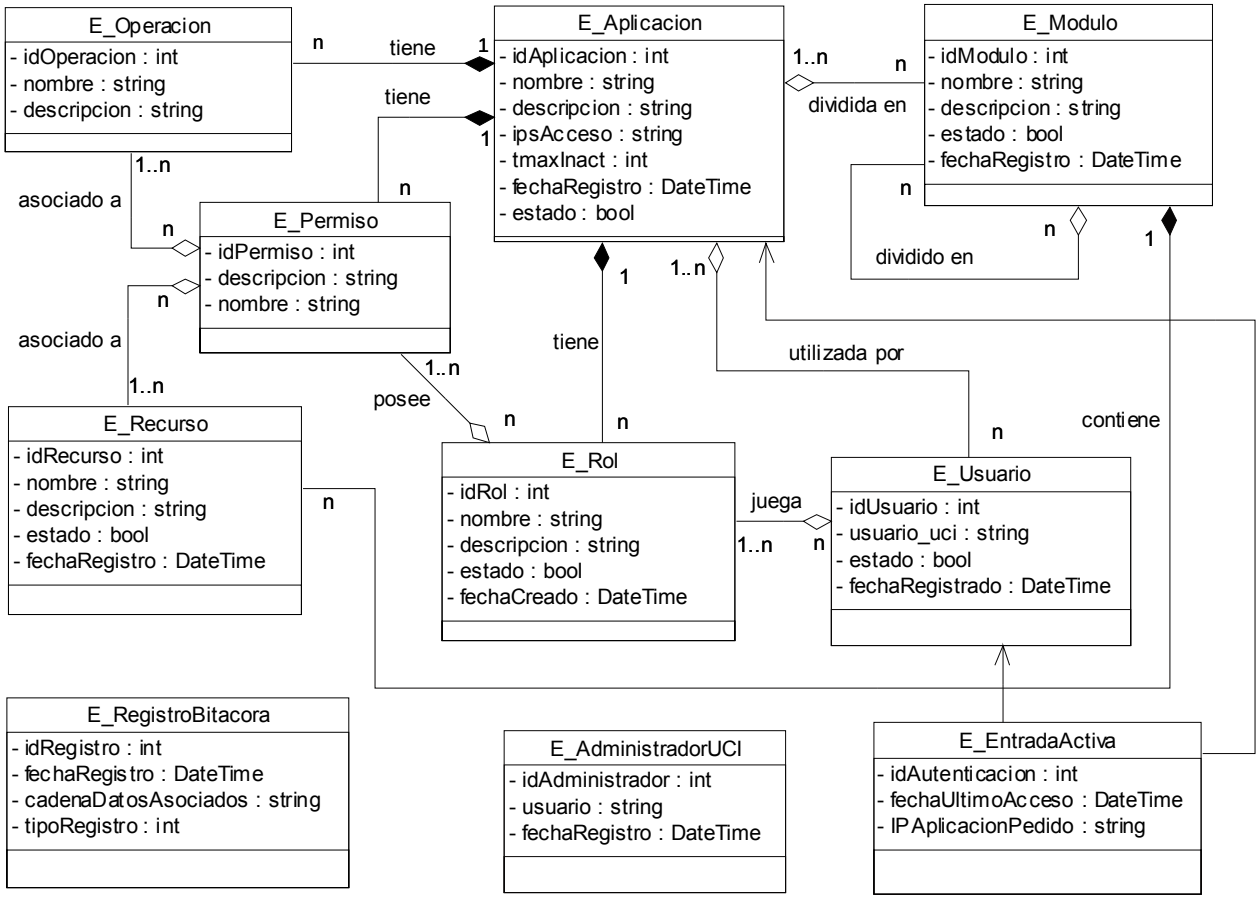
Fig 4. 7 Diagrama de clase. CU Brindar servicio para autorizar acceso.



4.3 Diseño de la Base de Datos

4.3.1 Modelo lógico de datos (Diagrama de clases persistentes)

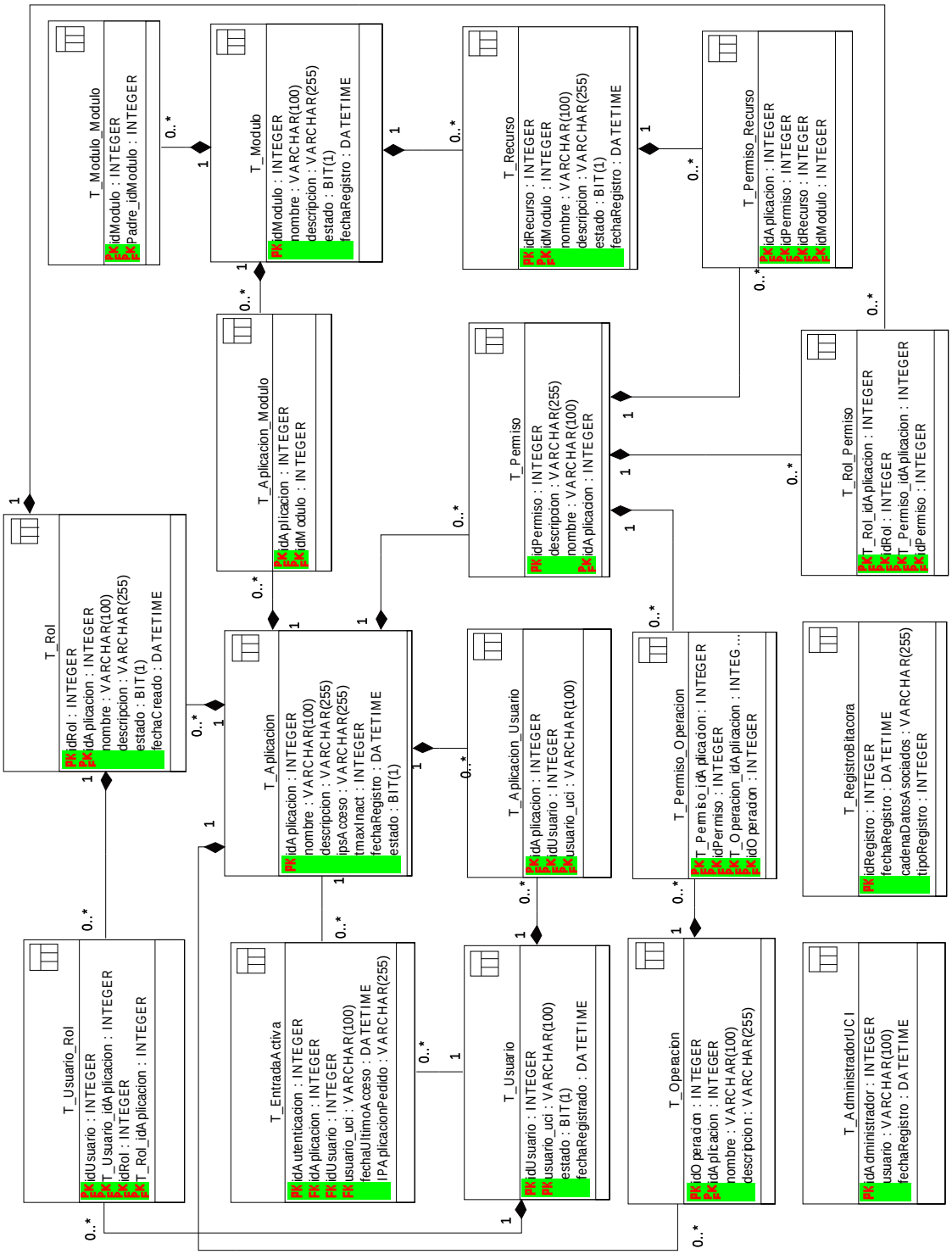
Fig 4. 8 Diagrama de clases persistentes



4.3.2 Modelo físico de datos (Modelo de datos)

El diagrama del modelo de datos se corresponde con la representación física de la base de datos.

Fig 4. 9 Modelo físico de datos.



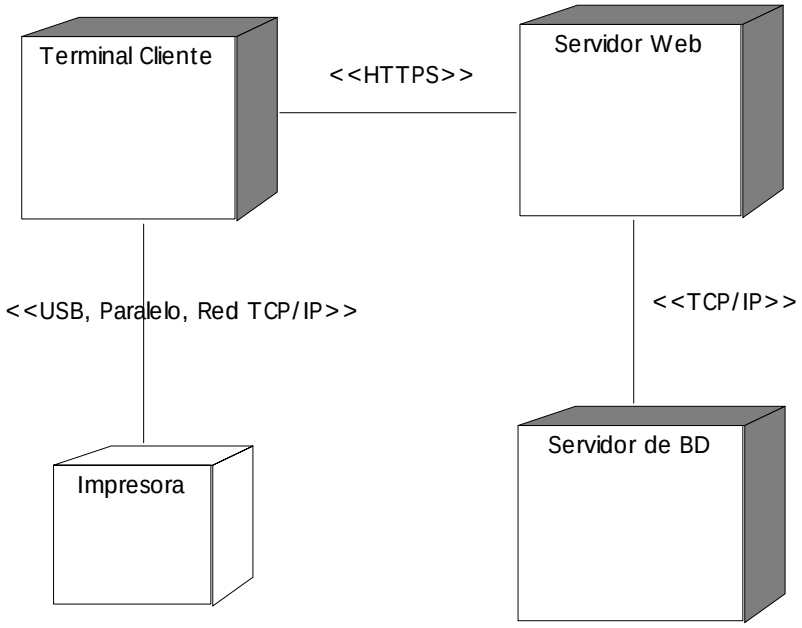
.

4.4 Generalidades de la Implementación

4.4.1 Modelo de Despliegue

El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de computo. El modelo de despliegue se utiliza como entrada fundamental en las actividades de diseño e implementación debido a que la distribución del sistema tiene una influencia principal en su diseño.

Fig 4.10 Diagrama de despliegue



El nodo Servidor de BD representa un servidor SQL Server, en el cual se ubica la información persistente del sistema, en el Servidor Web, Internet Information Services, se ubican íntegramente, las capas de presentación, lógica del negocio y de acceso a datos del sistema, así como los servicios que se brindan.

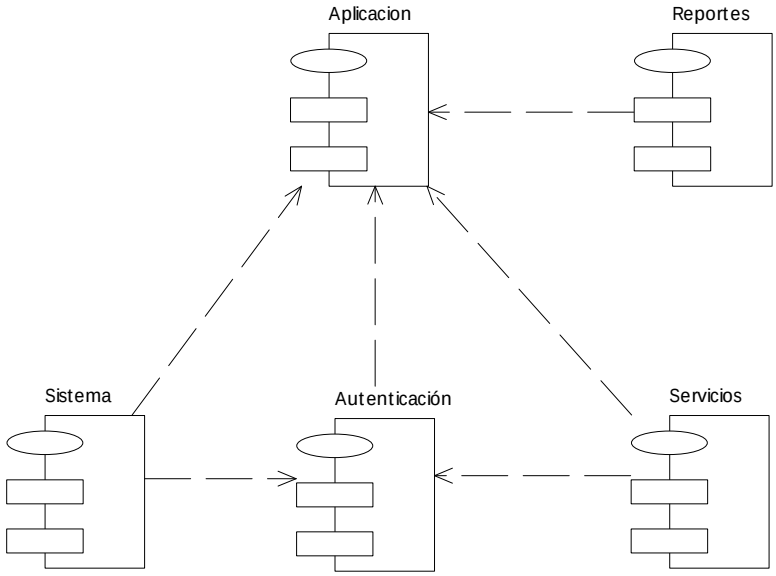
Terminales clientes representan el conjunto de computadoras desde las cuales se puede administrar y acceder a los servicios disponibles, es un nodo con capacidad de procesamiento, al igual que los anteriores, porque es donde se interpretan las respuestas a las solicitudes hechas al servidor en forma de HTML y donde se ejecutan códigos JavaScript para validaciones y encriptaciones de información.

Impresora representa un dispositivo de esta naturaleza que dispone de conexión física con las Terminales Clientes para obtener copias duras de reportes suministrados por Securitas.

4.4.2 Modelo de Implementación

El modelo de implementación es uno de los artefactos que se construye durante la fase de implementación y describe como los elementos del modelo de diseño (clases) se implementan en términos de componentes, ficheros de código fuente, ejecutables entre otros. El modelo de implementación describe también como se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados, y cómo dependen unos de otros. A continuación se representa la relación existente entre los paquetes de los componentes propuesto para el sistema Securitas, mostrando el resto de los diagramas en el Anexo 3.

Fig 4.16 Diagrama de paquetes de implementación.



4.5 Conclusiones

En el capítulo se realizó el diseño del sistema obteniendo como resultados los diagramas de clases del diseño, se muestra el diagrama de clases persistentes de donde se obtuvo el modelo de datos como parte del diseño de la base de datos a utilizar. También se desarrolló el diagrama de despliegue donde quedaron modelados los nodos en los que se distribuye la aplicación, especificando las conexiones de red y protocolos que los unen. Además se desarrolló el modelo de implementación que describe como se organizan los componentes en nuestro sistema.

CAPÍTULO 5 Estudio de Factibilidad.

5.1 Introducción

Desde los primeros momentos del desarrollo de un software, resulta necesario determinar si el mismo resultará factible o no. Para ello es necesario realizar un estudio de los beneficios que aporta y las inversiones que implica tanto en las esferas organizativas, como en la economía y en la técnica, para llevar a cabo su implementación.

El presente capítulo está dedicado al estudio de una de las metodologías tradicionales para la estimación de proyectos software, lo cual da una aproximación de la duración de un proyecto como una variable dependiente de los recursos a emplear. Las estimaciones están asociadas con el esfuerzo y el tiempo con las actividades identificadas del proyecto.

5.2 Planificación basada en Puntos de casos de uso.

La estimación mediante el análisis de Puntos de Casos de Uso es un método propuesto originalmente por Gustav Karner de Objectory AB, y posteriormente refinado por muchos otros autores.

Se trata de un método de estimación de tiempo de desarrollo de un proyecto mediante la asignación de "pesos" a un cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de esos factores. A continuación se detallan los pasos a seguir para la aplicación de este método.

Paso 1: *Identificar los Puntos de casos de uso Desajustados.*

UUCP = UAW + UUCW

Donde:

UUCP: Puntos de Casos de Uso sin ajustar.

UAW: Factor de Peso de los Actores sin ajustar.

UUCW: Factor de Peso de los Casos de Uso sin ajustar.

Cálculo del Factor de Peso de los Actores sin ajustar (UAW):

Este valor se calcula mediante un análisis de la cantidad de actores presentes en el sistema y la complejidad de cada uno de ellos que se establece teniendo en cuenta en primer lugar si se trata de una persona o de otro sistema, y en segundo lugar, la forma en la que el actor interactúa con el sistema.

Los criterios se muestran en la siguiente tabla:

Tipo de actor	***Descripción***	***Factor de peso***	***Actores***	***Total***
Simple	Sistema con sistema a través de interfaz de programación.	1	0	0
Medio	Sistema con sistema mediante protocolo de interfaz basada en texto.	2	1	2
Complejo	Persona que interactúa con el sistema mediante interfaz gráfica.	3	1	3

UAW = S(Factor * Actores) **UAW** **5**

Cálculo del Factor de Peso de los Casos de Uso sin ajustar (UUC)

Este valor se calcula mediante un análisis de la cantidad de Casos de Uso presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los Casos de Uso se establece teniendo en cuenta la cantidad de transacciones efectuadas en el mismo, donde una *transacción* se entiende como una secuencia de actividades atómica, es decir, se efectúa la secuencia de actividades completa, o no se efectúa ninguna de las actividades de la secuencia.

Los criterios se muestran en la siguiente tabla:

Tipo de CU	*Descripción*	*Peso*	*Cantidad de CU*	*Total*
Simple	El caso de uso tiene de 1 a 3 transacciones.	5	22	110
Medio	El caso de uso tiene de 4 a 7 transacciones.	10	0	0
Complejo	El caso de uso tiene más de 8 transacciones.	15	0	0

UUCW = ∑(Factor * CantCU) **UUCW** **110**

UUCP = UAW + UUCW

UUCP= 115

Paso 2: *Cálculo de Puntos de Casos de Uso ajustados.*

Una vez que se tienen los Puntos de Casos de Uso sin ajustar, se debe ajustar este valor mediante la siguiente ecuación:

UCP = UUCP * TCF * EF

Donde:

UCP: Puntos de Casos de Uso ajustados.

UUCP: Puntos de Casos de Uso sin ajustar.

TCF: Factor de complejidad técnica.

EF: Factor de ambiente.

Cálculo del Factor de Complejidad Técnica (TCF):

$TCF = 0.6 + 0.01 \times \Sigma (Peso_i \times Valor\ Asignado)$

Este coeficiente se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de los factores se cuantifica con un valor de 0 a 5.

Significado de los valores

0: No presente o sin influencia.

1: Influencia incidental o presencia incidental.

2: Influencia moderada o presencia moderada.

3: Influencia media o presencia media.

4: Influencia significativa o presencia significativa.

5: Fuerte influencia o fuerte presencia.

En la siguiente tabla se muestra el significado y el peso de cada uno de éstos factores:

Factor	***Descripción***	***Peso***	***Valor asignado***	***Total***
T1	Sistema distribuido	2	0	0
T2	Tiempo de respuesta	1	4	4
T3	Eficiencia del usuario final	1	3	3

T4	Funcionamiento Interno complejo	1	3	3
T5	El código debe ser reutilizable	1	4	4
T6	Facilidad de instalación	0.5	4	2
T7	Facilidad de uso	0.5	5	2.5
T8	Portabilidad	2	0	0
T9	Facilidad de cambio	1	4	4
T10	Concurrencia	1	5	5
T11	Incluye objetivos especiales de seguridad	1	5	5
T12	Provee acceso directo a terceras partes	1	0	0
T13	Se requieren facilidades especiales de entrenamiento de usuarios	1	2	2

$TCF = 0.6 + 0.01 * \Sigma (Peso_i \times Valor\ Asignado)$

$TCF= 0.6 + 0.01 * 34.5$

$TCF= 0.945$

Cálculo del Factor de Ambiente (EF):

$EF = 1.4 - 0.03 * \Sigma (Peso_i * Valor_i)$

El cálculo del Factor de Ambiente es similar al cálculo del Factor de complejidad técnica, es decir, se trata de un conjunto de factores que se cuantifican con valores de 0 a 5.

En la siguiente tabla se muestra el significado y el peso de cada uno de éstos factores.

Factor	***Descripción***	***Peso***	***Valor asignado***	***Total***
E1	Familiaridad con el modelo de proyecto utilizado	1.5	3	4.5
E2	Experiencia en la aplicación	0.5	4	2
E3	Experiencia en la orientación a objetos.	1	3	3
E4	Capacidad del analista líder.	0.5	2	1
E5	Motivación.	1	4	4
E6	Estabilidad de requerimientos	2	4	8
E7	Personal Part–Time	-1	5	-5
E8	Dificultad del lenguaje de programación	-1	1	-1

$EF = 1.4 - 0.03 * \Sigma (Peso_i * Valor_i)$

EF= 1.4 – 0.03 * 17.5

EF= 0.875

Luego:

UCP = UUCP * TCF * EF

UCP= 95.090625

Paso 3: *Estimación de esfuerzo a través de los puntos de casos de uso.*

El esfuerzo en horas-hombre viene dado por: **E = UCP** * **CF**

Donde

E: esfuerzo estimado en horas-hombre.

UCP: Puntos de Casos de Uso ajustados.

CF: factor de conversión.

Para calcular CF:

Para obtener el factor de conversión (CF) se cuentan cuantos valores de los que afectan el factor ambiente (E1...E6) están por debajo de la media (3) y los que están por arriba de la media para los restantes (E7, E8).

CF = 20 Horas-Hombre / Punto de Casos de uso (si $\text{Total}_{EF} \leq 2$)

CF = 28 Horas-Hombre / Punto de Casos de uso (si $\text{Total}_{EF} \geq 3$)

CF = abandonar o cambiar proyecto ya que se considera que el riesgo de fracaso del mismo es demasiado alto. (si $\text{Total}_{EF} \geq 5$)

Por tanto: **CF = 20 Horas-Hombre / Punto de Casos de uso**

Luego:

E = UCP * CF

E= 1901.8125 Horas- Hombre

Paso 4: *Calcular esfuerzo de todo el proyecto.*

Este método proporciona una estimación del esfuerzo en horas-hombre contemplando sólo el desarrollo de la funcionalidad especificada en los casos de uso.

Para una estimación más completa de la duración total del proyecto, hay que agregar a la estimación del esfuerzo obtenida por los Puntos de Casos de Uso, las estimaciones de esfuerzo de las demás actividades relacionadas con el desarrollo de software.

El criterio plantea la distribución del esfuerzo entre las diferentes actividades de un proyecto, según la siguiente aproximación:

Actividad	Porcentaje %	Horas-Hombres
Análisis	10	475.45313
Diseño	20	950.90625
Implementación	40	1901.8125
Pruebas	15	713.17969
Sobrecarga (otras actividades)	15	713.17969
Total	100	4754.5313

Donde:

ET1: Esfuerzo Total (Horas- Hombre)

ET2: Esfuerzo Total (Mes- Hombre)

SM: Salario Promedio Mensual

CH: Cantidad de Hombres

CHM: Costo Hombre- Mes

Costo: Costo Total del Proyecto

Si ET1 = 4754.5313 horas- hombre y cada mes los desarrolladores trabajan como promedio 192 horas (trabajan 8 horas por 24 días en un mes):

Entonces: **ET2= ET1/ horas_ mes** eso daría un:

ET2= 24.763184 Mes- Hombre

Esto quiere decir que 1 persona puede realizar el problema analizado en 25 meses aproximadamente

Costo de proyecto

Se asume que el salario promedio mensual es de $ 100. 00 y trabajan 2 hombres en el proyecto.

CHM = CH * SM

Entonces: **CHM= $ 200.00 /mes**

Costo= SM * ET2

Costo = $ 2476.3184

5.3 Beneficios tangibles e intangibles

El software propuesto trae consigo varios beneficios sobre todo intangibles, pues proveerá servicios de seguridad a los sistemas web de la facultad de forma centralizada, para lograr uniformidad en este

aspecto y promover la reutilización de código. Cada sistema web que utilice los servicios que brinda el sistema de seguridad podrá tener control de los usuarios que acceden a el, así como, podrá registrar cada operación que realice el mismo, con el objetivo de mantener control de los usuarios que hacen uso de la aplicación. Como beneficio tangible una vez terminado el software, puede ser utilizado en las demás facultades, además de que se puede comercializar, pues existen muchas empresas que utilizan varios sistemas y cuentan con gran número de trabajadores, por lo que les sería de gran utilidad manipular el acceso a través de usuarios.

5.4 Análisis de costos y beneficios

Desarrollar un software implica invertir en el mismo, por lo que hay que analizar los beneficios que reportarían su implantación y uso, para poder justificar de forma adecuada su desarrollo.

La utilización del sistema de seguridad centralizada brindaría servicios que aportarían seguridad a las diferentes aplicaciones web de la facultad, manteniendo un control detallado de los accesos a través de usuarios y registrando las operaciones realizadas por los mismos. Este sistema permitirá detectar posibles violaciones de seguridad y reconstruir escenarios ante cualquier novedad, gestionaría la seguridad de una forma natural, según los roles que desempeña cada usuario; y en caso de nuevas aplicaciones optimizaría el proceso de administración de usuarios con el uso del directorio LDAP de la UCI.

Una inversión en seguridad nunca es innecesaria y menos cuando se habla de productos informáticos, donde hay en juego información importante que puede ser utilizada por personas no autorizadas. Solo por el costo de **$ 2476.3184** se puede desarrollar el sistema.

Por todas las ventajas que proporcionaría el sistema se decidió que es factible llevar acabo su desarrollo.

5.5 Conclusiones

En este capitulo se describe el estudio de la factibilidad del sistema propuesto, mediante la planificación de los Puntos de Casos de Uso, como método utilizado. Se logró la estimación de esfuerzo y tiempo de las actividades, así como costo y los beneficios que aporta el uso del software a desarrollar, lo que indica que es factible implementar la herramienta propuesta.

CONCLUSIONES

Con el diseño del Sistema de seguridad centralizada de aplicaciones web se da cumplimiento al objetivo del trabajo, pues ha quedado modelado el sistema que resultaría en una aplicación en la que se reflejarían todos los resultados de la investigación realizada a lo largo del trabajo, logrando a la vez:

- Mantener la integridad, disponibilidad y confidencialidad de la información de los usuarios que utilizarían el sistema.
- Brindar servicios de seguridad a todas las aplicaciones que desee utilizarlos.
- Mantener control de accesos de usuarios, lo cual permite conocer los accesos y las operaciones realizadas por el usuario en una aplicación cualquiera que utilice los servicios.
- Evitar que en el futuro cada sistema tenga que implementar sus propios elementos de seguridad, disminuyendo el costo y tiempo de desarrollo.
- Permitir la concentración de todos los esfuerzos de seguridad en un solo punto, logrando consistencia de la información y garantía contra las violaciones a nivel de sistemas.

RECOMENDACIONES

- Implementar el sistema de seguridad Securitas para mejorar los procesos que se gestionan en la facultad como parte del desarrollo investigativo, docente y laboral en el que está inmerso la universidad.
- Someter al sistema de seguridad una vez implementado a las pruebas correspondientes con el fin de explotar posibles vulnerabilidades y perfeccionar las funcionalidades del mismo.
- La implementación por parte de la universidad de una infraestructura para proporcionar certificados digitales a todos los miembros de la comunidad universitaria de forma fácil y eficaz, obteniendo comunicaciones autenticadas, íntegras y seguras.
- Una vez puesto en práctica se debe establecer las medidas enfocadas en la prevención, detección y recuperación propuestas en el anexo 4.
- Migrar la arquitectura del sistema hacia plataformas libres, en caso necesario, para la mejora de los costos recomendándose ASP.Net, C#, IDE MONO develop, framework MONO, Postgree SQL, y servidor Apache con el objetivo de mantener el diseño original.
- El diseño y la implementación de un módulo que se instale en la aplicación cliente con el objetivo de lograr una mejor integridad entre Securitas y la aplicación web consumidora.

REFERENCIAS BIBLIOGRÁFICAS

1. **Digitales, Dpto. Sistema.** Teleformación. *"Control de acceso. Identificación y autenticación".* [En línea] http://teleformacion.uci.cu.

2. —. Teleformación. *"Introducción a la Seguridad Informática".* [En línea] [Citado el: 3 de Abril de 2007.] http://teleformacion.uci.cu/.

3. —. Teleformación. *"Ataques informáticos y mecanismos de defensa".* [En línea] [Citado el: 5 de Abril de 2007.] http://teleformacion.uci.cu.

4. **Arias, Yuniel Eliades Proenza.** ARQUITECTURA DE SEGURIDAD PARA APLICACIONES WEB EMPRESARIALES. Ciudad de La Habana : s.n., 2006.

5. **Marti, Jose Ramon Esteban.** Seguridad en La red. [En línea] 1998-2006. [Citado el: 7 de Abril de 2007.] http://www.seguridadenlared.org/.

6. El rincon de Quevedo. [En línea] [Citado el: 5 de Abril de 2007.] http://www.inf.utfsm.cl.

7. Microsoft Forefront. [En línea] [Citado el: 6 de Abril de 2007.] http://www.microsoft.com.

8. Kernelnet Informática. [En línea] [Citado el: 6 de Abril de 2007.] http://www.kernelnet.com.

9. **Sumit Siddharth, Pratiksha Doshi.** Security Focus. [En línea] [Citado el: 6 de Abril de 2007.] http://www.securityfocus.com.

10. Citrix. [En línea] [Citado el: 6 de Abril de 2007.] http://www.citrix.es.

11. **Yoenis Pantoja Zaldívar, Yurisbel Vega Ortiz.** SEGURINET. *SISTEMA GENERICO DE SEGURIDAD.ADAPTACION PARA EL SISTEMA DE GESTION DE INSMET.* Ciudad de La Habana : s.n., 2006.

12. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El Proceso Unificado de Desarrollo Software.* s.l. : Addison Wesley, 2000.

13. **Microsoft.** MSDN en español. [En línea] http://www.microsoft.com/spanish/msdn/arquitectura/das/guias/AppArchCh2.asp.

14. Microsoft.Net [En línea] [Citado:Abril 9, 2007.] http://www.microsoft.com/latam/net/introduccion/quees.asp

15. Microsoft.Net [En línea] [Citado:Abril 9, 2007.] http://www.microsoft.com/latam/net/basics/framework.asp

16. **J. A. G. Seco**, "El lenguaje de programación C#." [En línea] [Citado: Abril 11, 2007.] http://www.josanguapo.com/librocsharp2.zip

17. Microsoft ASP.NET QuickStart Tutorial [En línea] [Citado:Abril 12, 2007.] http://es.gotdotnet.com/quickstart/aspplus/

18. Microsoft.Net [En línea] [Citado:Abril 12, 2007.] http://www.microsoft.com/latam/net/basics/xmlservices.asp

19 Teleformación. *"Preparación para Prueba de Nivel de IGSW, Conferencia: Arquitectura ".* [En línea] [Citado: Abril 15, 2007.] http://teleformacion.uci.cu/.

20. **LDAP_es** [En línea] [Citado:Abril 12, 2007.] http://www.ldap-es.org/node/21

21. AJAX un nuevo acercamiento a Aplicaciones Web [En línea] [Citado:Abril 12, 2007.] http://www.uberbin.net/archivos/internet/ajax-un-nuevo-acercamiento-a-aplicaciones-web.php

.

BIBLIOGRAFÍA

1. **Martin., R.** *Designing SQL Server 2000. Databases for .net Enterprises Servers.* s.l. : Syngress, 2001.

2. **Larman, Craig.** *UML Y PATRONES.Introduccion al análisis y diseño orientado a objetos.* Ciudada de la Habna : Editorial Felix Varela, 2004.

3. **Ivar Jacobson, Grady Booch, James Rumbaugh.** *EL PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE.* Ciudad de La Habana : Felix Varela, 2004.

4. **Pressman., Roger S.** *Ingenieria del Software. Un enfoque practico.* Ciudad de La Habana : Felix Varela, 2005.

5. **V.Lau, P.K.Yuen &.** *PRACTICAL WEB TECNOLOGIES.* Ciudad de La Habana : Felix Varela, 2004.

6. *Criptografía y Seguridad en Computadores.* Jaén : s.n., 1999.

7. CUCERT. [En línea] http://www.cucert.co.cu.

8. Hispasec. [En línea] http://www.hispasec.com.

9. Navegalis. [En línea] http://webmasters.navegalis.com/articulos/mostrar/18/.

10. Segurmatica. [En línea] http://www.segurmatica.co.cu/.

11. **Baader, Rodolfo.** Seguridad en servidores y aplicaciones Web. 2004.

12. **Elvar, David.** El Guille. [En línea] http://www.solotuweb.com/fs~id~5715.html.

13. **Moira J. West-Brown, Don Stikvoort, Klaus-Peter Kossakowski, Klaus-Peter Kossakowski, Robin Ruefle, Mark Zajicek.** *Handbook for Computer Security Incident Response Teams (CSIRTs).* 2003.

14. **Rodríguez, Lázaro Orlando Aneiro.** *ELEMENTO DE ARQUITECTURA Y SEGURIDAD INFORMATICA.* La Habana : Pueblo y Educacion, 2001.

15. **Román Medina-Heigl Hernández.** Análisis de seguridad, optimización y mejora de un portal web basado en PHP y MySQL. Sevilla : s.n., 2002.

16. **Siler Amador Donado, Guefry L. Agredo Méndez, Carolina A. Carrascal Reyes.** Politicas de Seguridad Computacional. 2002.

17. [En línea] http://www.linuxsecurity.com/.

18. [En línea] http://www.microsoft.com/security/default.mspx.

19. [En línea] http://msdn2.microsoft.com/es-es/library/yfe5dwc2(vs.80).aspx.

20. [En línea] http://msdn.microsoft.com/msdnmag/issues/02/09/SecurityTips/default.aspx.

21. [En línea] http://es.tldp.org/Manuales-LuCAS/doc-unixsec/unixsec-html/node279.html.

ANEXOS

Anexo 1. Descripciones de los casos de uso del sistema (no presentadas en el capítulo 3)

Tabla 6. 1 Descripción del CU: Gestionar módulos.

Caso de uso:	**Gestionar módulos.** **(Sección principal)**
Actores:	Administrador de Securitas
Propósito:	Permitir registrar, modificar o eliminar módulos en una aplicación.
Resumen:	El administrador de Securitas selecciona una opción para gestionar módulo (Registrar nuevo, Modificar o Eliminar). El sistema evalúa la posible realización de la acción comprobando la validez de los datos de entrada. El sistema permite la realización efectiva o no de la acción en dependencia de la validez de los datos.
Tipo:	Crítico y esencial.
Referencias:	R4
Precondiciones:	El administrador de Securitas se encuentra autenticado en el sistema. El administrador ha seleccionado aplicaciones previamente.
Poscondiciones:	Se actualiza la información en el sistema referente a módulos.
Curso normal de eventos	
Acción del actor:	**Respuesta del sistema:**

1 El caso de uso inicia cuando el administrador de Securitas desea gestionar módulos, accediendo a la opción correspondiente.	**1.1** El sistema muestra un árbol con todos los módulos de las aplicaciones seleccionadas y las opciones de Registrar nuevo, Modificar y Eliminar.
2 El administrador de Securitas selecciona una de las opciones. a) Si selecciona Registrar nuevo ver *sección Registrar nuevo módulo.* b) Si selecciona Modificar ver *sección Modificar módulo.* c) Si se selecciona Eliminar ver *sección Eliminar módulo*.	
Sección: Registrar nuevo módulo	
Curso normal de eventos	
	2.1 El sistema muestra los campos requeridos para registrar un nuevo módulo: nombre del módulo, descripción y estado.
3 El administrador de Securitas llena los campos requeridos y acepta.	**3.1** El sistema verifica que los datos sean válidos.
	3.2 El sistema registra el nuevo módulo y muestra un mensaje que confirma el desarrollo exitoso de la operación.
Cursos alternativos	

	3.1 Si los datos ingresados no son válidos se muestra un mensaje de error.
Sección: Modificar módulo	
Curso normal de eventos	
3 El administrador de Securitas selecciona el módulo que desea modificar.	**3.1** El sistema verifica que se ha seleccionado un módulo.
	3.2 El sistema muestra los campos de los datos actuales y modificables del módulo permitiendo edición sobre estos: nombre del módulo, descripción y estado.
4 El administrador de Securitas realiza las modificaciones necesarias y acepta.	**4.1** El sistema verifica que los datos sean válidos.
	4.2 El sistema actualiza los datos del módulo y muestra un mensaje que confirma el desarrollo exitoso de la operación.
Cursos alternos	
	3.1 Si no se ha seleccionado un módulo el sistema muestra mensaje de error.
	4.1 Si los datos introducidos no son válidos se muestra un mensaje de error.
Sección: Eliminar módulo.	

Curso normal de eventos	
3 El administrador de Securitas selecciona el módulo que desea eliminar.	**3.1** El sistema verifica que se ha seleccionado un módulo.
	3.2 El sistema muestra un mensaje de advertencia solicitando que se confirme si se desea realmente eliminar el módulo seleccionado.
4 El administrador de Securitas confirma que desea eliminar el módulo seleccionado.	**4.1** El sistema elimina el módulo seleccionado.
Cursos alternos	
	3.1 Si no se ha seleccionado un módulo el sistema muestra mensaje de error.
	4.1 El sistema no realiza ninguna operación.

Tabla 6. 2 Descripción del CU: Gestionar operaciones.

Caso de uso:	**Gestionar operaciones.** **(Sección principal)**
Actores:	Administrador de Securitas
Propósito:	Permitir registrar, modificar o eliminar operaciones en una aplicación.
Resumen:	El administrador de Securitas selecciona una opción para gestionar operaciones (Registrar nueva, Modificar o Eliminar). El sistema evalúa la

	posible realización de la acción comprobando la validez de los datos de entrada. El sistema permite la realización efectiva o no de la acción en dependencia de la validez de los datos.
Tipo:	Crítico y esencial.
Referencias:	R6
Precondiciones:	El administrador de Securitas se encuentra autenticado en el sistema. Se ha seleccionado previamente una aplicación.
Poscondiciones:	Se actualiza la información en el sistema referente a operaciones

Curso normal de eventos

Acción del actor:	**Respuesta del sistema:**
1 El caso de uso inicia cuando el administrador de Securitas desea gestionar operaciones de las aplicaciones registradas.	**1.1** El sistema muestra todas las operaciones de la aplicación seleccionada y las opciones de Registrar nueva, Modificar y Eliminar.
2 El administrador de Securitas selecciona una de las opciones. a) Si se selecciona Registrar nueva ver *sección Registrar nueva operación.* b) Si se selecciona Modificar ver *sección Modificar operación.* c) Si se selecciona Eliminar ver *sección Eliminar operación*.	

Sección: Registrar nueva operación	
Curso normal de eventos	
	2.1 El sistema muestra los campos requeridos para registrar nueva operación: nombre de la operación y descripción.
3 El administrador de Securitas llena los campos requeridos y acepta.	**3.1** El sistema verifica que los datos sean válidos.
	3.2 El sistema registra la nueva operación y muestra un mensaje que confirma el desarrollo exitoso de la operación.
Cursos alternativos	
	3.1 Si los datos introducidos no son válidos se muestra un mensaje de error.
Sección: Modificar operación	
Curso normal de eventos	
3 El administrador de Securitas selecciona la operación que desea modificar.	**3.1** El sistema verifica que se ha seleccionado una operación.
	3.2 El sistema muestra los campos de los datos actuales y modificables de la operación permitiendo edición sobre estos: nombre de la operación y descripción.

4 El administrador de Securitas realiza las modificaciones necesarias y acepta.	**4.1** El sistema verifica que los datos sean válidos.
	4.2 El sistema actualiza los datos de la operación y muestra un mensaje que confirma el desarrollo exitoso
Cursos alternos	
	3.1 Si no se ha seleccionado una operación el sistema muestra mensaje de error.
	4.1 Si los datos entrados no son válidos se muestra un mensaje de error.
Sección: Eliminar operación.	
Curso normal de eventos	
3 El administrador de Securitas selecciona la operación que desea eliminar.	**3.1** El sistema verifica que se ha seleccionado una operación.
	3.2 El sistema muestra un mensaje de advertencia solicitando que se confirme si se desea realmente eliminar la operación seleccionada.
4 El Administrador de Securitas confirma que desea eliminar la operación seleccionada.	**4.1** El sistema elimina la operación seleccionada.
Cursos alternos	
	3.1 Si no se ha seleccionado una operación el

	sistema muestra mensaje de error.
	4.1 El sistema no realiza ninguna actividad.

Tabla 6. 3 Descripción del CU: Brindar servicio para cerrar sesión de usuario.

Caso de uso:	**Brindar servicio para cerrar sesión de usuario.** **(Sección principal)**
Actores:	Sistema web
Propósito:	Cerrar sesión de un usuario autenticado en una aplicación registrada.
Resumen:	Una aplicación a la que se le brinda servicios de seguridad (Sistema web) solicita que se cierre la sesión de un usuario previamente autenticado.
Tipo:	Crítico y esencial.
Referencias:	R20
Precondiciones:	
Poscondiciones:	Se ha cerrado la sesión de usuario que estaba autenticado a solicitud de la aplicación.
Curso normal de eventos	
Acción del actor:	**Respuesta del sistema:**
1 El caso de uso inicia cuando una aplicación	**1.1** El sistema verifica que el identificador de

a la que se le brinda servicios de seguridad (Sistema web) solicita que se cierre la sesión de un usuario autenticado brindando los parámetros: identificador de autenticación, usuario y aplicación.	autenticación hace referencia a algún usuario autenticado.
	1.2 Comprueba que el IP de la PC que hace el pedido coincide con el IPs de la PC que autenticó al usuario (según el identificador de autenticación).
	1.3 El sistema verifica que el usuario autenticado y aplicación correspondiente coinciden con los parámetros usuario y aplicación pasados en la petición.
	1.4 El sistema cierra la sesión del usuario autenticado y devuelve mensaje de operación exitosa.
Cursos alternativos	
	1.1 Si el identificador de autenticación no hace referencia a algún usuario autenticado o se venció el tiempo máximo de inactividad se emite mensaje de error.
	1.2 Si el IP de la Pc que hace el pedido no coincide con el IPs de la PC que autenticó al usuario se emite mensaje de error
	1.3 Si no coinciden el usuario autenticado y aplicación correspondiente con los parámetros usuario y aplicación pasados en la petición el sistema

	emite mensaje de error.

Tabla 6. 4 Descripción del CU: Brindar servicio para registrar operación.

Caso de uso:	**Brindar servicio para registrar operación.** **(Sección principal)**
Actores:	Sistema web
Propósito:	Brindar servicio para registrar operación de un usuario sobre recurso en aplicación registrada en Securitas
Resumen:	Una aplicación a la que se le brinda servicios de seguridad solicita a Securitas registrar operación de un usuario sobre recurso.
Tipo:	Crítico y esencial.
Referencias:	R22
Precondiciones:	
Poscondiciones:	
Curso normal de eventos	
Acción del actor:	**Respuesta del sistema:**
1 El caso de uso inicia cuando una aplicación a la que se le brinda servicios de seguridad solicita registrar operación de un usuario autenticado sobre recurso brindando los parámetros: identificador de autenticación,	**1.1** El sistema verifica que el identificador de autenticación hace referencia a algún usuario autenticado.

operación, módulo y recurso.	
	1.2 El sistema verifica que el recurso existe sobre el modulo (según parámetros) y en la aplicación del usuario autenticado según el identificador de autenticación.
	1.3 El sistema verifica que el usuario autenticado tiene permisos de realizar la operación sobre el recurso.
	1.4 El sistema registra en la bitácora la operación del usuario autenticado sobre el recurso.
Cursos alternativos	
	1.1 Si el identificador de autenticación no hace referencia a algún usuario autenticado se emite mensaje de error y se registra en la bitácora.
	1.2 Si el recurso no existe se emite mensaje de error y se registra en la bitácora.
	1.3 Si usuario autenticado no tiene permisos de realizar la operación sobre el recurso se emite mensaje de error y se registra en la bitácora.

Tabla 6. 5 Descripción del CU: Cerrar sesión del administrador autenticado.

Caso de uso:	**Cerrar sesión del administrador autenticado** **(Sección principal)**
Actores:	Administrador de Securitas
Propósito:	Permitir cerrar la sesión del administrador autenticado.
Resumen:	El administrador de Securitas desea cerrar su sesión desde cualquier lugar del sistema.
Tipo:	Secundario y esencial.
Referencias:	R2
Precondiciones:	
Poscondiciones:	

Curso normal de eventos	
Acción del actor:	**Respuesta del sistema:**
1 El caso de uso inicia cuando el administrador de Securitas procede a cerrar sesión a través de la opción correspondiente.	**1.1** El sistema muestra un mensaje de advertencia solicitando confirmación.
2 El administrador de Securitas confirma que desea cerrar sesión.	**2.1** La sesión queda cerrada.
Cursos alternativos	

	2.1 El sistema no se cierra.
Requerimientos especiales:	

Tabla 6. 6 Descripción del CU: Generar reporte de aplicaciones.

Caso de uso:	**Generar reporte de aplicaciones**
Actores:	Administrador de Securitas.
Propósito:	Generar reportes de aplicaciones (Filtrar, mostrar e imprimir) datos referentes a aplicaciones existentes en el sistema.
Resumen:	El administrador de Securitas desea generar reportes de aplicaciones por nombre, descripción, estado, fecha de registro, dirección IP de acceso.
Tipo:	Secundario y esencial
Referencias:	R15
Precondiciones:	El Administrador de Securitas se encuentra previamente autenticado en el sistema.
Poscondiciones:	
Caso de uso asociado:	
Curso normal de eventos	
Acción del actor:	**Respuesta del sistema:**
1 El caso de uso inicia cuando el Administrador de Securitas desea generar reportes de las aplicaciones existentes en el sistema; para ello puede escoger filtrar por nombre, descripción,	1.1 El sistema filtra estos datos generando un reporte.

estado, fecha de registro, dirección IP de acceso.	
2 El Administrador de Securitas solicita mostrar el reporte en pantalla.	2.1 El sistema muestra el reporte como un documento, con una breve descripción.
Cursos alternativos	
Acción del actor:	**Respuesta del sistema:**
	1.1Si el Administrador de Securitas no aplico filtros el sistema genera un reporte con todos las aplicaciones existentes.
Requerimientos especiales:	

Tabla 6. 7 Descripción del CU: Generar reporte de módulos.

Caso de uso:	**Generar reporte de módulos**
Actores:	Administrador de Securitas.
Propósito:	Generar reportes de módulo (Filtrar, mostrar e imprimir) datos referentes a módulo existentes en el sistema.
Resumen:	El administrador de Securitas desea generar reportes de módulos por nombre, descripción, estado, fecha de registro, módulos por aplicación, submódulos por modulo.
Tipo:	Secundario y esencial
Referencias:	R11
Precondiciones:	El Administrador de Securitas se encuentra previamente autenticado en el

	sistema.
Poscondiciones:	
Caso de uso asociado:	

Curso normal de eventos	
Acción del actor:	**Respuesta del sistema:**
1. El caso de uso inicia cuando el Administrador de Securitas desea generar reportes de módulos de las diferentes aplicaciones existentes en el sistema; para ello puede escoger filtrar por nombre, estado, fecha de registro, módulos por aplicación y submódulos por modulo.	1.1 El sistema filtra estos datos generando un reporte.
2. El Administrador de Securitas solicita mostrar el reporte en pantalla.	2.1 El sistema muestra el reporte como un documento, con una breve descripción.
Cursos alternativos	
Acción del actor:	**Respuesta del sistema:**
	1.1Si el Administrador de Securitas no aplico filtros el sistema genera un reporte con todos los módulos existentes.
Requerimientos especiales:	

Tabla 6. 8 Descripción del CU: Generar reporte de recursos.

Caso de uso:	**Generar reporte de recursos**
Actores:	Administrador de Securitas.
Propósito:	Generar reportes de recursos (Filtrar, mostrar e imprimir) datos referentes a recursos existentes en el sistema.
Resumen:	El administrador de Securitas desea generar reportes de recursos por nombre, descripción, estado, fecha de registro, módulo y aplicación.
Tipo:	Secundario y esencial.
Referencias:	R12
Precondiciones:	El Administrador de Securitas se encuentra previamente autenticado en el sistema.
Poscondiciones:	
Caso de uso asociado:	

Curso normal de eventos	
Acción del actor:	**Respuesta del sistema:**
1 El caso de uso inicia cuando el Administrador de Securitas desea generar reportes de recursos de las diferentes aplicaciones existentes en el sistema; para ello puede escoger filtrar por nombre, descripción, estado, fecha de registro, módulo, aplicación.	1.1 El sistema filtra estos datos generando un reporte.

2 El Administrador de Securitas solicita mostrar el reporte en pantalla.	2.1 El sistema muestra el reporte como un documento, con una breve descripción.
Cursos alternativos	
Acción del actor:	**Respuesta del sistema:**
	1.1Si el Administrador de Securitas no aplico filtros el sistema genera un reporte con todos los recursos existentes.
Requerimientos especiales:	

Tabla 6. 9 Descripción del CU: Generar reporte de usuarios

Caso de uso:	**Generar reporte de usuarios.**
Actores:	Administrador de Securitas.
Propósito:	Generar reportes de usuarios (Filtrar, mostrar e imprimir) datos referentes a usuarios existentes en el sistema.
Resumen:	El administrador de Securitas desea generar reportes de usuarios por usuario de la uci, estado, fecha de registro, fecha de último acceso, rol, aplicación
Tipo:	Secundario y esencial.
Referencias:	R13
Precondiciones:	El Administrador de Securitas se encuentra previamente autenticado en el sistema.
Poscondiciones:	

Caso de uso asociado:	
Curso normal de eventos	
Acción del actor:	**Respuesta del sistema:**
1 El caso de uso inicia cuando el Administrador de Securitas desea generar reportes de usuarios de las diferentes aplicaciones existentes en el sistema; para ello puede escoger filtrar por usuario de la uci, estado, fecha de registro, fecha de último acceso, rol, aplicación	1.1 El sistema filtra estos datos generando un reporte.
2 El Administrador de Securitas solicita generar reporte en pantalla.	2.1 El sistema muestra el reporte como un documento, con una breve descripción.
Cursos alternativos	
Acción del actor:	**Respuesta del sistema:**
	1.1Si el Administrador de Securitas no aplico filtros el sistema genera un reporte con todos los usuarios existentes.
Requerimientos especiales:	

Tabla 6. 10 Descripción del CU: Generar reporte de roles.

Caso de uso:	**Generar reporte de roles**

Actores:	Administrador de Securitas.
Propósito:	Generar reportes de roles (Filtrar, mostrar e imprimir) datos referentes a roles existentes en el sistema.
Resumen:	El administrador de Securitas desea generar reportes de roles por nombre, descripción, estado, fecha de registro, roles por aplicación.
Tipo:	Secundario y esencial.
Referencias:	R14
Precondiciones:	El Administrador de Securitas se encuentra previamente autenticado en el sistema.
Poscondiciones:	
Caso de uso asociado:	

Curso normal de eventos	
Acción del actor:	**Respuesta del sistema:**
1 El caso de uso inicia cuando el Administrador de Securitas desea generar reportes de roles de las diferentes aplicaciones existentes en el sistema; para ello puede escoger filtrar por nombre, estado, fecha de registro, roles por aplicación y roles por usuario.	1.1 El sistema filtra estos datos generando un reporte.
2 El Administrador de Securitas solicita mostrar el reporte en pantalla.	2.1 El sistema muestra el reporte como un documento, con una breve descripción.

Cursos alternativos	
Acción del actor:	**Respuesta del sistema:**
	1.1Si el Administrador de Securitas no aplico filtros el sistema genera un reporte con todos los roles existentes.
Requerimientos especiales:	

Tabla 6. 11 Descripción del CU: Generar reporte de bitácora.

Caso de uso:	**Generar reporte de bitácora.**
Actores:	Administrador de Securitas.
Propósito:	Generar reportes de Bitácora (Filtrar, mostrar e imprimir) datos referentes a la Bitácora.
Resumen:	El administrador de Securitas desea generar reportes de la Bitácora por acceso, aplicación, módulo, recurso, usuario, operación, fecha.
Tipo:	Secundario y esencial.
Referencias:	R16
Precondiciones:	El Administrador de Securitas se encuentra previamente autenticado en el sistema.
Poscondiciones:	
Caso de uso asociado:	

Curso normal de eventos	
Acción del actor:	**Respuesta del sistema:**
1 El caso de uso inicia cuando el Administrador de Securitas desea generar reportes de la Bitácora; para ello puede escoger filtrar por acceso, aplicación, módulo, recurso, usuario, operación, fecha.	1.1 El sistema filtra estos datos generando un reporte.
2 El Administrador de Securitas solicita mostrar el reporte en pantalla.	2.1 El sistema muestra el reporte como un documento, con una breve descripción.
Cursos alternativos	
Acción del actor:	**Respuesta del sistema:**
	1.1Si el Administrador de Securitas no aplico filtros el sistema genera un reporte con todos los registros existentes en la Bitácora.
Requerimientos especiales:	

Tabla 6. 12 Descripción del CU: Gestionar cuenta de administración local.

Caso de uso:	**Gestionar cuenta de administración local** **(Sección: Principal)**
Actores:	Administrador de Securitas.
Propósito:	Crear cuenta de administración local posibilitando cambiar la contraseña a

	voluntad del administrador de Securitas.
Resumen:	El administrador debe crear una cuenta de administración local introduciendo usuario local y contraseña, siendo ésta posible de cambiar cada vez que el administrador lo estime conveniente.
Tipo:	Crítico y esencial.
Referencias:	R18
Precondiciones:	
Poscondiciones:	
Caso de uso asociado:	

Curso normal de eventos	
Acción del actor:	**Respuesta del sistema:**
1 El caso de uso se inicia cuando el Administrador desea gestionar cuenta del dominio local. a. Si selecciona la opción Adicionar, véase sección Adicionar cuenta. b. Si selecciona la opción *Cambiar contraseña*, véase sección *Cambiar contraseña de cuenta local.*	

Cursos alternativos	
Acción del actor:	**Respuesta del sistema:**
Sección: Crear cuenta local.	
Curso normal de eventos	
Acción del actor:	**Respuesta del sistema:**
1 El Administrador de Securitas desea crear una cuenta de administración local.	**1.1** El sistema solicita la entrada de los datos: usuario y contraseña.
2 El administrador de Securitas introduce los datos.	**2.1** El sistema verifica que la contraseña cumpla con los requisitos requeridos por el sistema para la contraseña de administración.
	2.2 El sistema solicita que se repita la entrada de la contraseña.
	2.3 Securitas almacena la nueva contraseña ene le sistema.
Cursos alternativos:	
Acción del actor:	**Respuesta del sistema:**
	2.1 Si la contraseña no cumple con los requisitos impuestos por el sistema para la contraseña de administración, el usuario debe entrar una nueva contraseña.
	2.2 Si al repetirse la nueva contraseña ésta no coincide con la primera entrada, el usuario debe entrar una

	nueva contraseña.
Sección: Cambiar contraseña de cuenta local	
Curso normal de eventos	
Acción del actor:	**Respuesta del sistema:**
1 El Administrador de Securitas introduce usuario y contraseña local a cambiar.	**1.1** El sistema verifica que los datos sean correctos
	1.2 El sistema solicita la entrada de la nueva contraseña.
2 El admisnistrador de Securitas introduce la nueva contraseña.	**2.1** Securitas comprueba que la contraseña cumpla con los requisitos establecidos para la contraseña de administración en el sistema.
	2.2 Securitas solicita que se repita la entrada de la nueva contraseña.
	2.3 Securitas almacena la nueva contraseña en el sistema.
Cursos alternativos:	
Acción del actor:	**Respuesta del sistema:**
	1.1 Si los datos no son correctos no se efectúa el cambio.
	2.1 Si la nueva contraseña no cumple con los requisitos que impone el sistema para la cuenta de administración, no se efectúa el cambio.

	2.2 Si al repetirse la contraseña, esta no coincide con la primera entrada el sistema solicita la entrada de una nueva contraseña.
	2.3 No se almacena la entrada de la nueva contraseña local.
Requerimientos especiales:	

Tabla 6. 13 Descripción del CU: Gestionar cuenta de administración UCI.

Caso de uso:	**Gestionar cuenta de administración UCI** **(Sección: Principal)**
Actores:	Administrador de Securitas.
Propósito:	Adicionar cuenta o eliminar cuenta del dominio UCI.
Resumen:	El Administrador de Securitas desea adicionar cuenta o eliminar cuenta del dominio UCI.
Tipo:	Secundario y esencial
Referencias:	R17
Precondiciones:	El Administrador de Securitas se encuentra autenticado en el sistema.
Poscondiciones:	
Caso de uso asociado:	

Curso normal de eventos	
Acción del actor:	**Respuesta del sistema:**
1 El caso de uso se inicia cuando el Administrador desea gestionar cuenta del dominio UCI. a. Si selecciona la opción Adicionar, véase sección Adicionar cuenta. b. Si selecciona la opción *Eliminar*, véase sección *Eliminar cuenta.*	
Cursos alternativos	
Acción del actor:	**Respuesta del sistema:**
Sección: Adicionar cuenta.	
Curso normal de eventos	
Acción del actor:	**Respuesta del sistema:**
	1.1 El sistema solicita la entrada del usuario del dominio.
2 El Administrador efectúa la entrada del usuario del dominio.	**2.1** El sistema comprueba que el usuario no se encuentre registrado en el mismo.

	2.2 El sistema comprueba que el usuario exista en el directorio LDAP.
	2.3 El sistema incorpora el usuario del dominio como usuario en Securitas.
Cursos alternativos:	
Acción del actor:	**Respuesta del sistema:**
	2.1 Si el usuario ya se encuentra registrado el sistema informa al administrador.
	2.2 Si el sistema no existe en el directorio LDAP muestra un mensaje de error.
	2.3 El sistema no incorpora el usuario del dominio como usuario en Securitas.
Sección: Eliminar cuenta del dominio UCI	
Curso normal de eventos	
Acción del actor:	**Respuesta del sistema:**
	1.1 El sistema muestra las cuentas del dominio UCI existentes en Securitas.
2 El Administrador selecciona la (las) cuenta(s) que desea eliminar.	**2.1** El sistema muestra un mensaje de advertencia solicitando que se confirme si se desea realmente eliminar la (las) cuenta(s) seleccionada(s).
3 El Administrador confirma que desea realmente eliminar la(las) cuenta(s)	**3.1** El sistema elimina la (las) cuenta (s) indicada(s) de

seleccionada(s).	Securitas.
Requerimientos especiales:	

Tabla 6. 14 Descripción del CU: Administrar bitácora.

Caso de uso:	**Administrar bitácora** **(Sección: Principal)**
Actores:	Administrador de Securitas.
Propósito:	Configurar la Bitácora, por periodo de tiempo de permanencia de los elementos en ella, o por eventos a auditar, así como permitir eliminar manualmente elementos de la Bitácora.
Resumen:	El Administrador de Securitas desea configurar la Bitácora, por periodo de tiempo de permanencia de los elementos en ella, o por eventos a auditar, así como eliminar manualmente elementos de la Bitácora.
Tipo	Opcional y esencial.
Referencias:	R10
Precondiciones:	El Administrador de Securitas se encuentra autenticado en el sistema.
Poscondiciones:	
Caso de uso asociado:	
Curso normal de eventos	
Acción del actor:	**Respuesta del sistema:**

1 El caso de uso se inicia cuando el Administrador desea administrar la Bitácora. a. Si selecciona la opción Configurar, véase sección Configurar Bitácora. b. Si selecciona la opción *Eliminar*, véase sección *Eliminar registro de Bitácora.*	
Cursos alternativos	
Acción del actor:	**Respuesta del sistema:**
Sección: Configurar bitácora.	
Curso normal de eventos	
Acción del actor:	**Respuesta del sistema:**
	1.1 El sistema muestra los elementos existentes en la Bitácora.
2 El Administrador da configuración a la Bitácora por eventos a auditar y según la permanencia que quiere dar a los registros en la misma.	**2.1** El sistema guarda la configuración.
Cursos alternativos:	

Acción del actor:	**Respuesta del sistema:**
Sección: Eliminar registro de bitácora.	
Curso normal de eventos	
Acción del actor:	**Respuesta del sistema:**
	1.1 El sistema muestra los registros existentes en la Bitácora
2 El Administrador selecciona el (los) registros(s) que desea eliminar	**2.1** El sistema muestra un mensaje de advertencia solicitando que se confirme si se desea realmente eliminar el(los) registro(s) seleccionado(s).
3 El Administrador confirma que desea realmente eliminar el(los) registo(s) seleccionado(s).	**3.1** El sistema elimina el(los) registro(s) indicado(s) de la Bitácora.

Anexo 2. Diagramas de clases (no presentados en el capitulo 4)

Fig 6. 1 Diagrama de clases. CU Gestionar módulos.

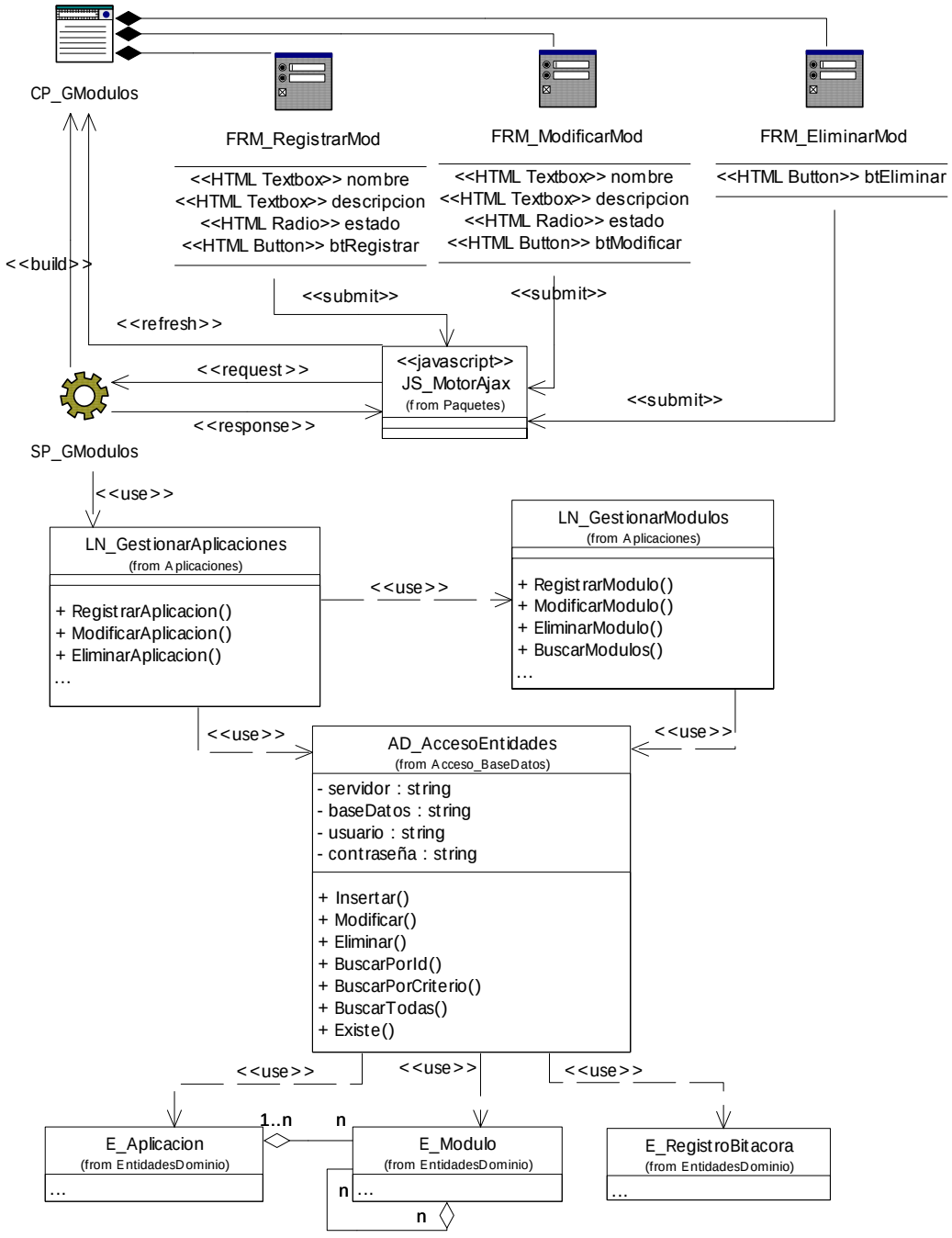


Fig 6. 2 Diagrama de clases. CU Gestionar Operaciones

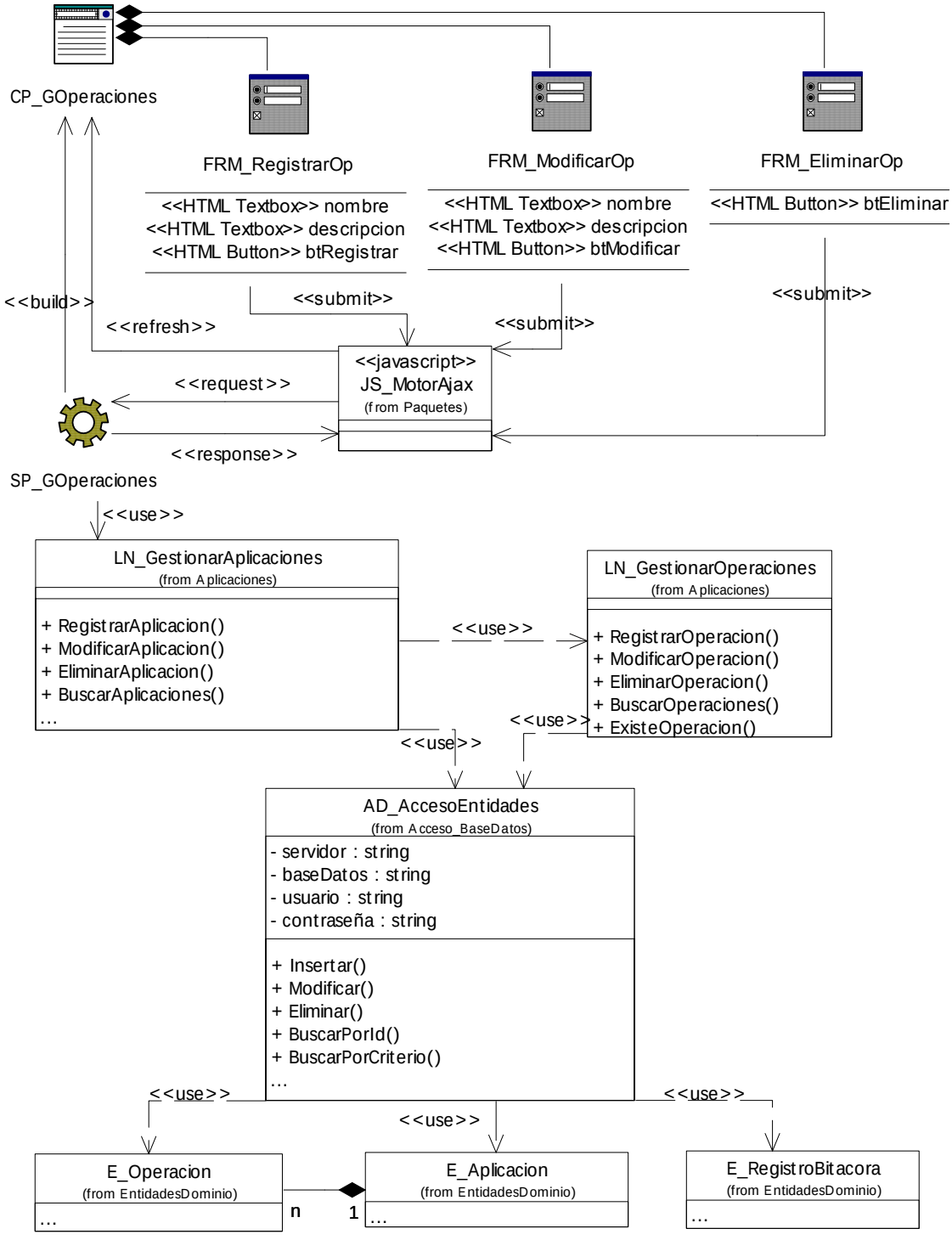


Fig 6. 3 Diagrama de clases. CU Brindar servicio para cerrar sesión de usuario.

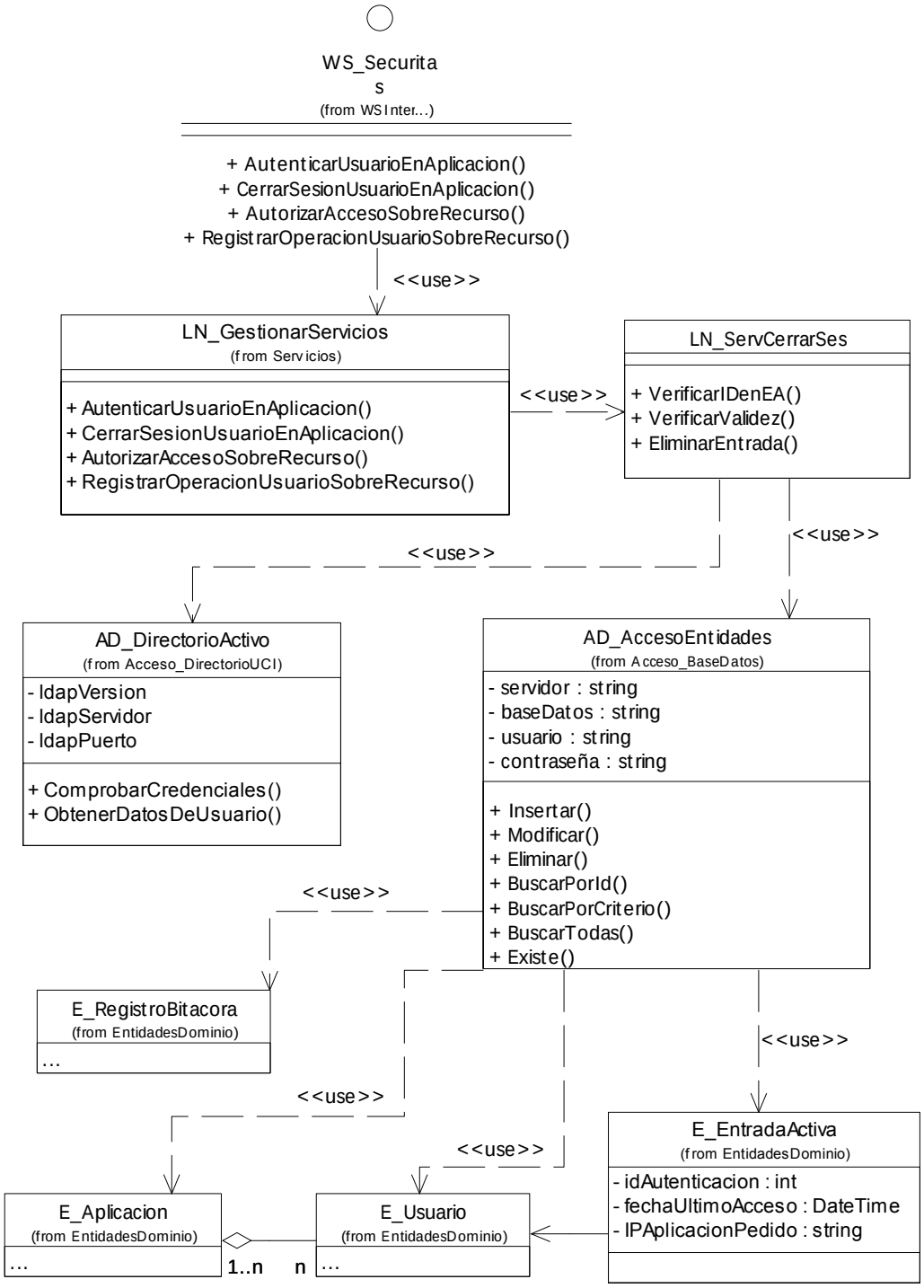


Fig 6. 4 Diagrama de clases. CU Brindar servicio para registrar operación.

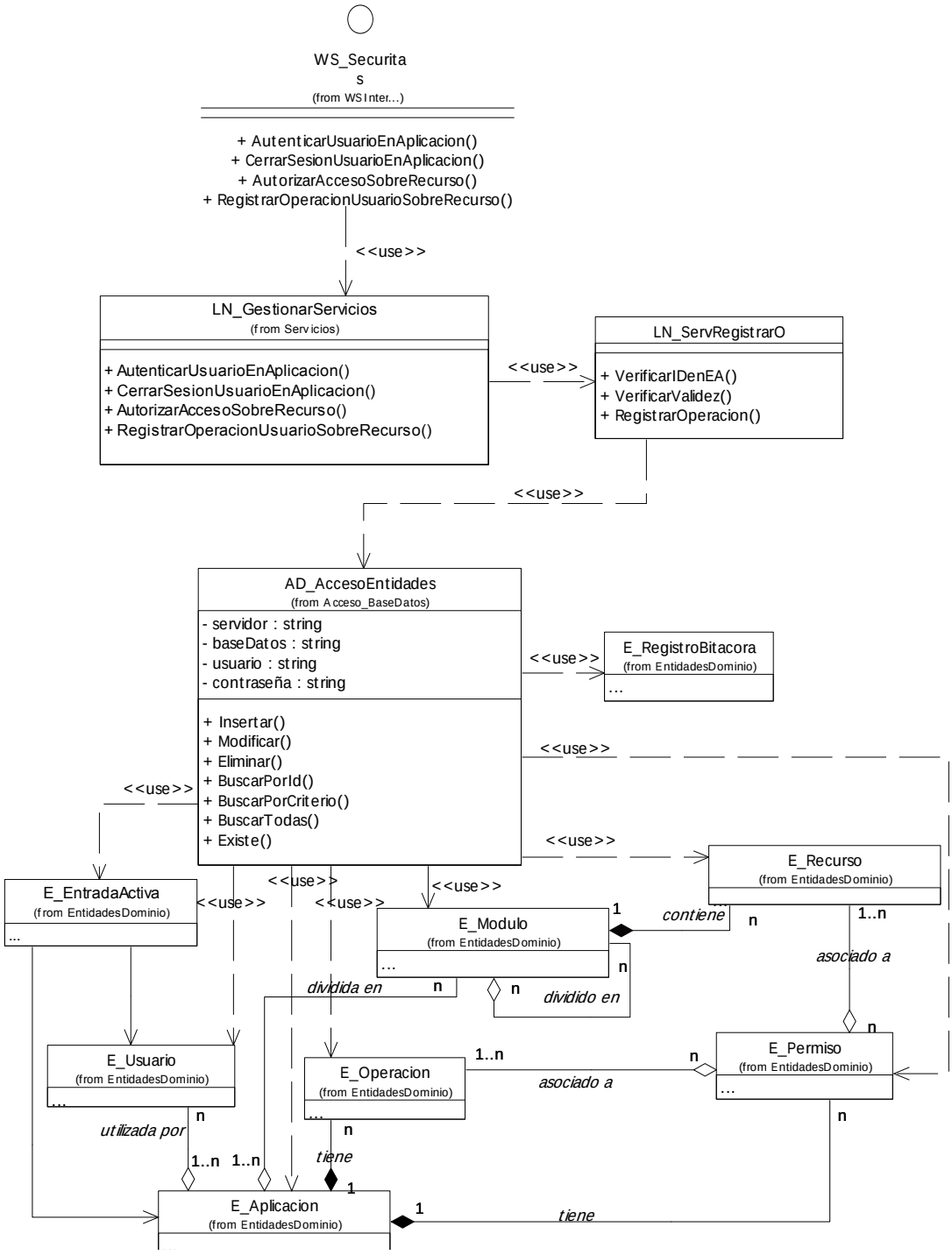


Fig 6. 5 Diagrama de clases. CU Generar reporte de aplicaciones.

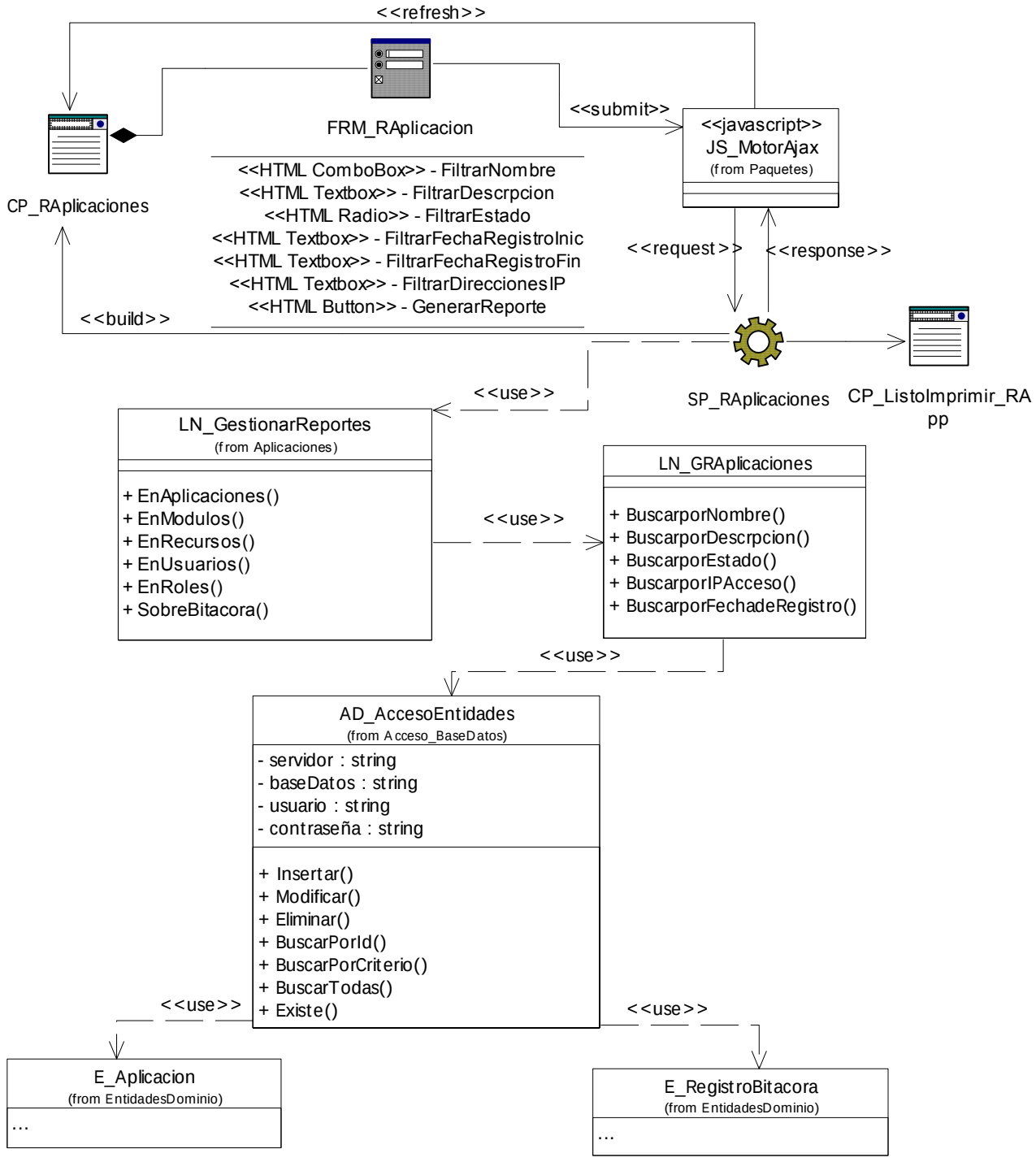


Fig 6. 6 Diagrama de clases. CU Generar reporte de módulos.

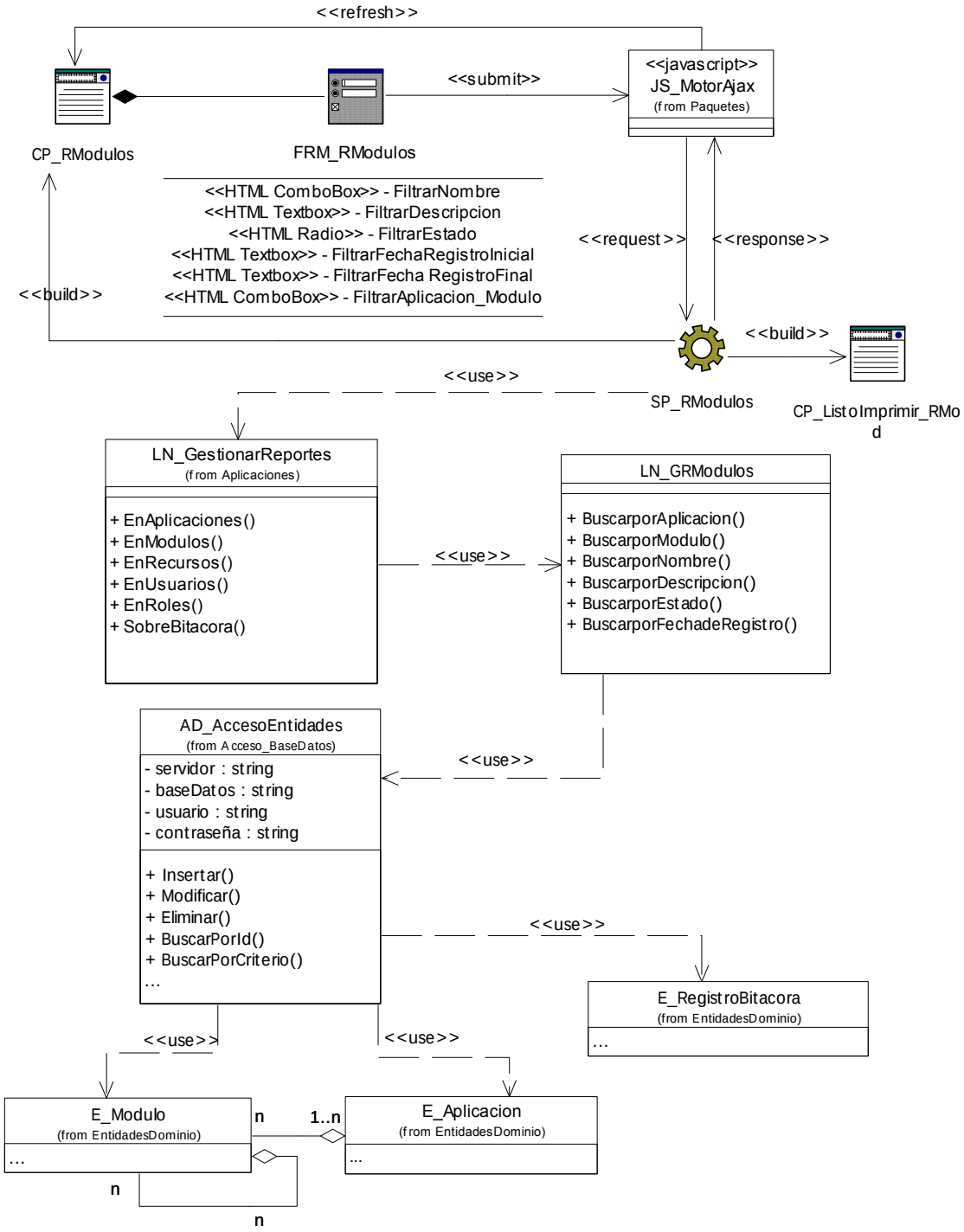


Fig 6. 7 Diagrama de clases. CU Generar reporte de recursos.

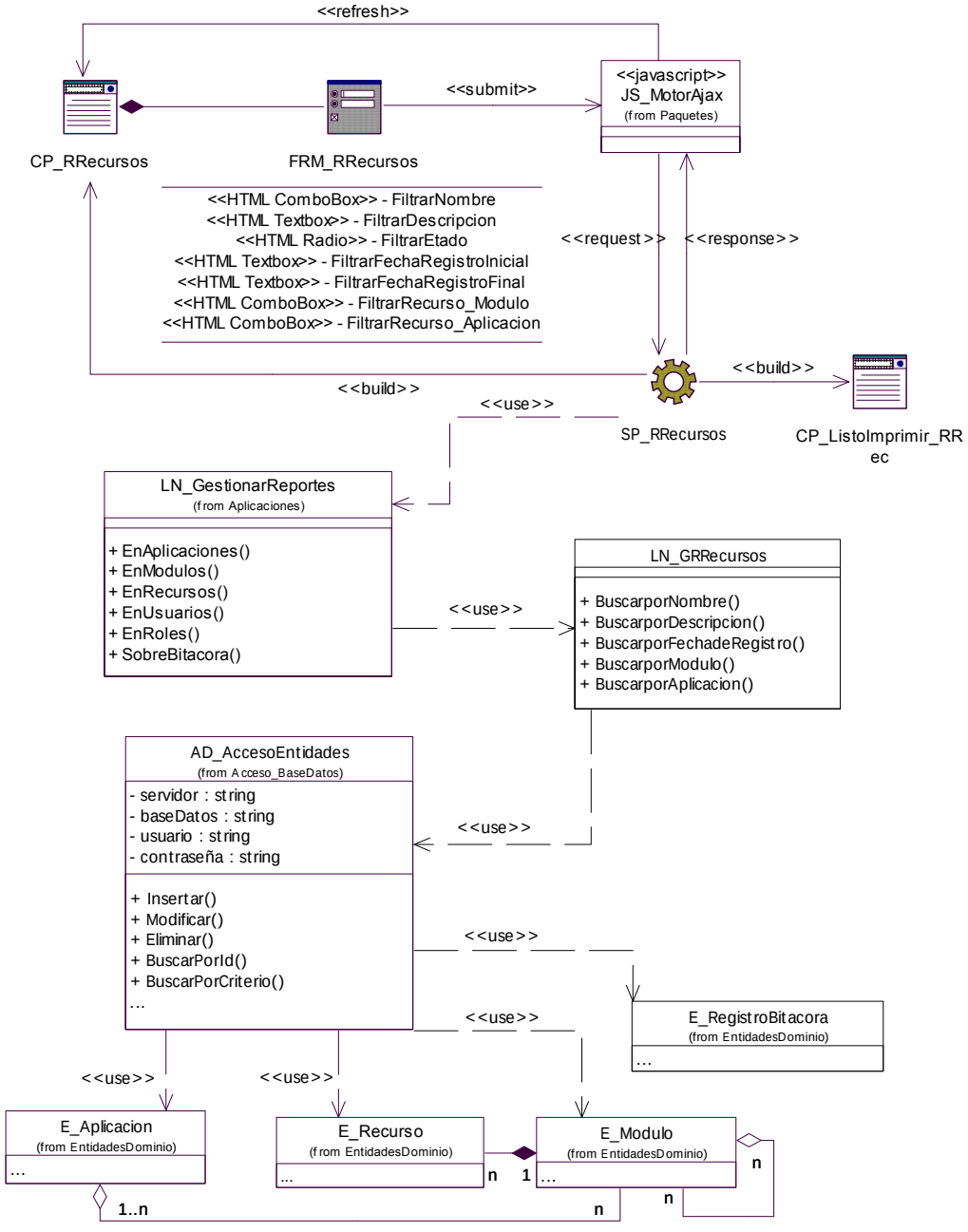


Fig 6. 8 Diagrama de clases. Generar reporte de roles.

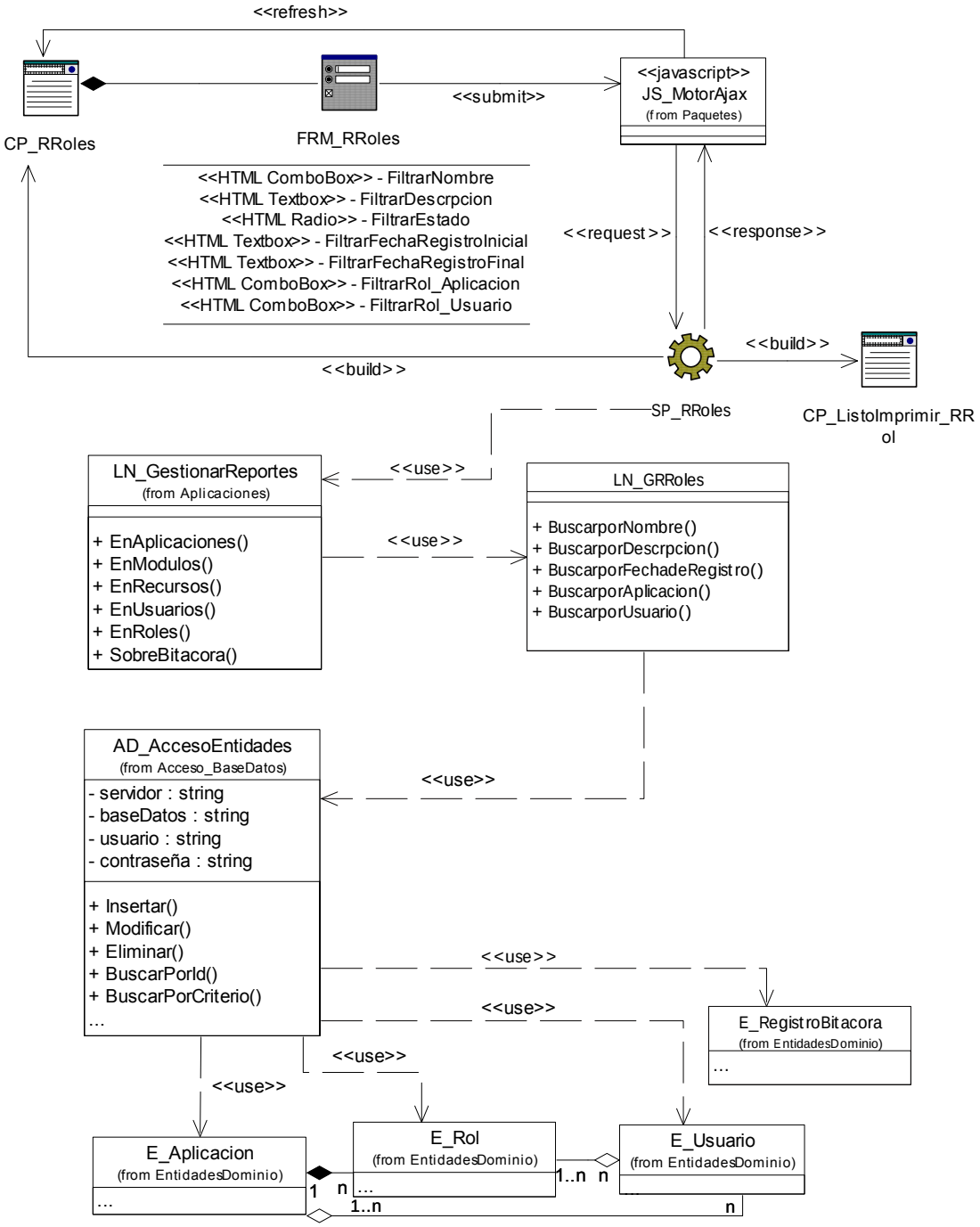


Fig 6. 9 Diagrama de clases. CU Generar reporte de usuarios.

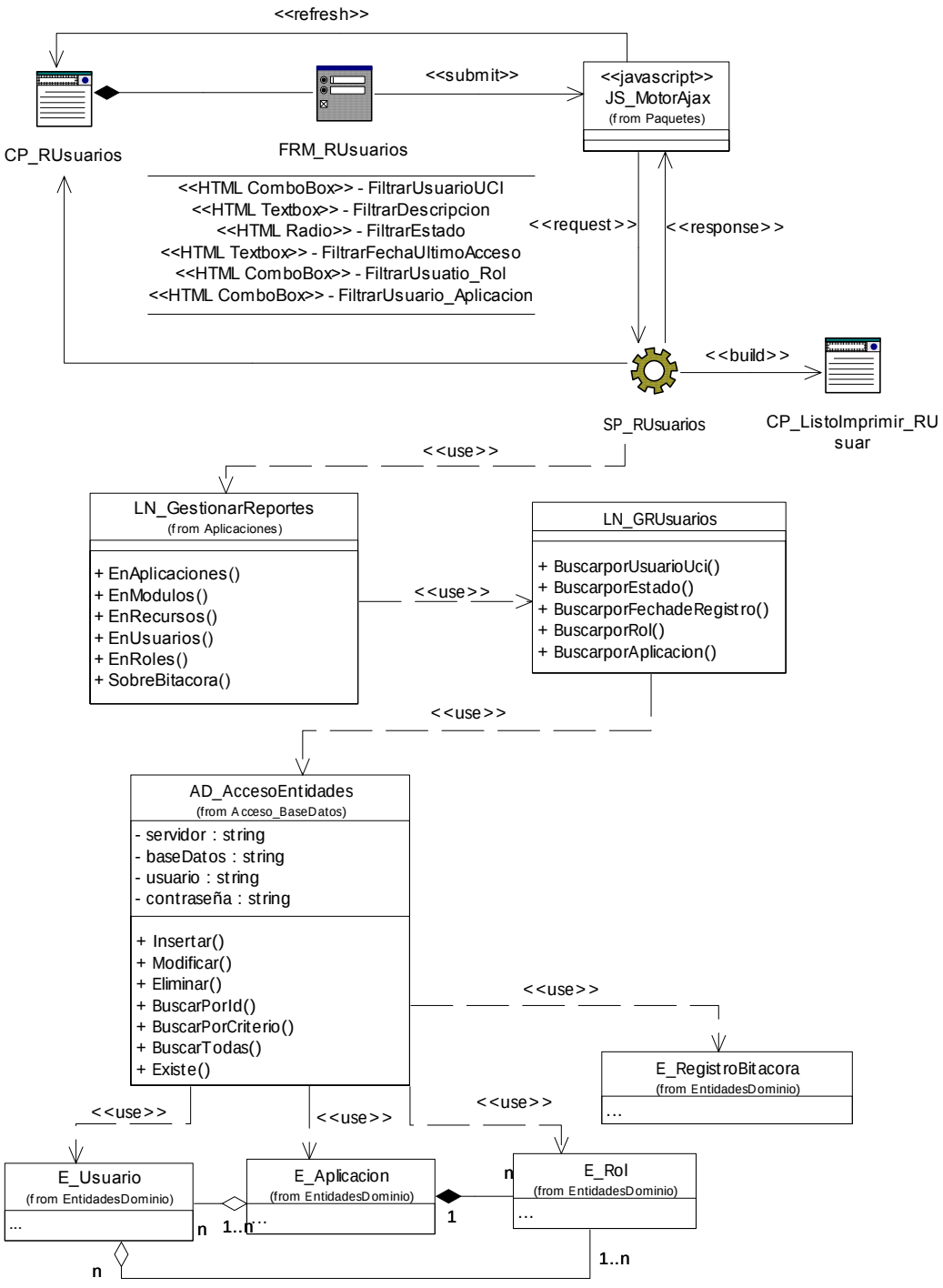


Fig 6. 10 Diagrama de clases. Generar reporte de bitácora.

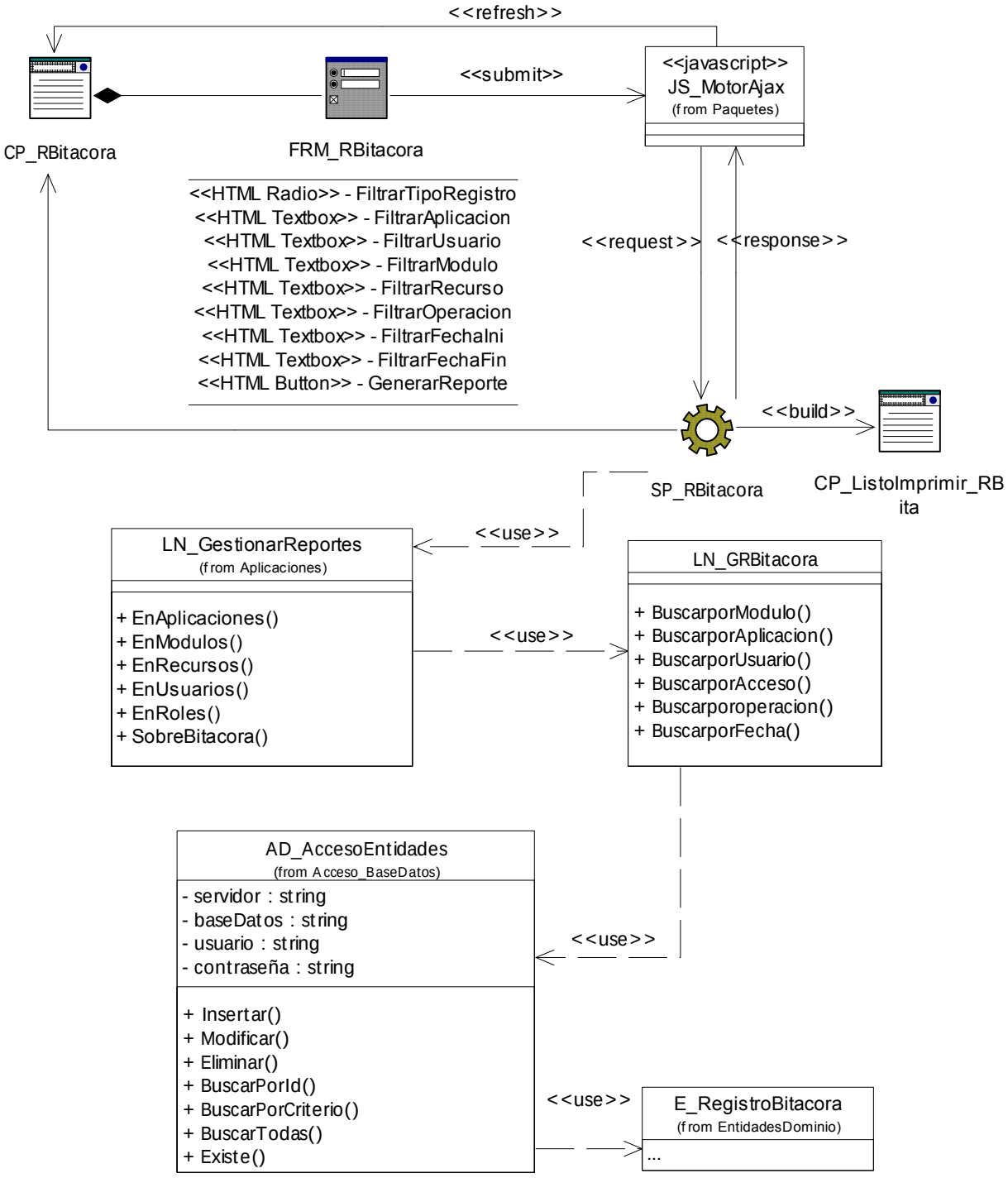


Fig 6. 11 Diagrama de clases. CU Administrar bitácora.

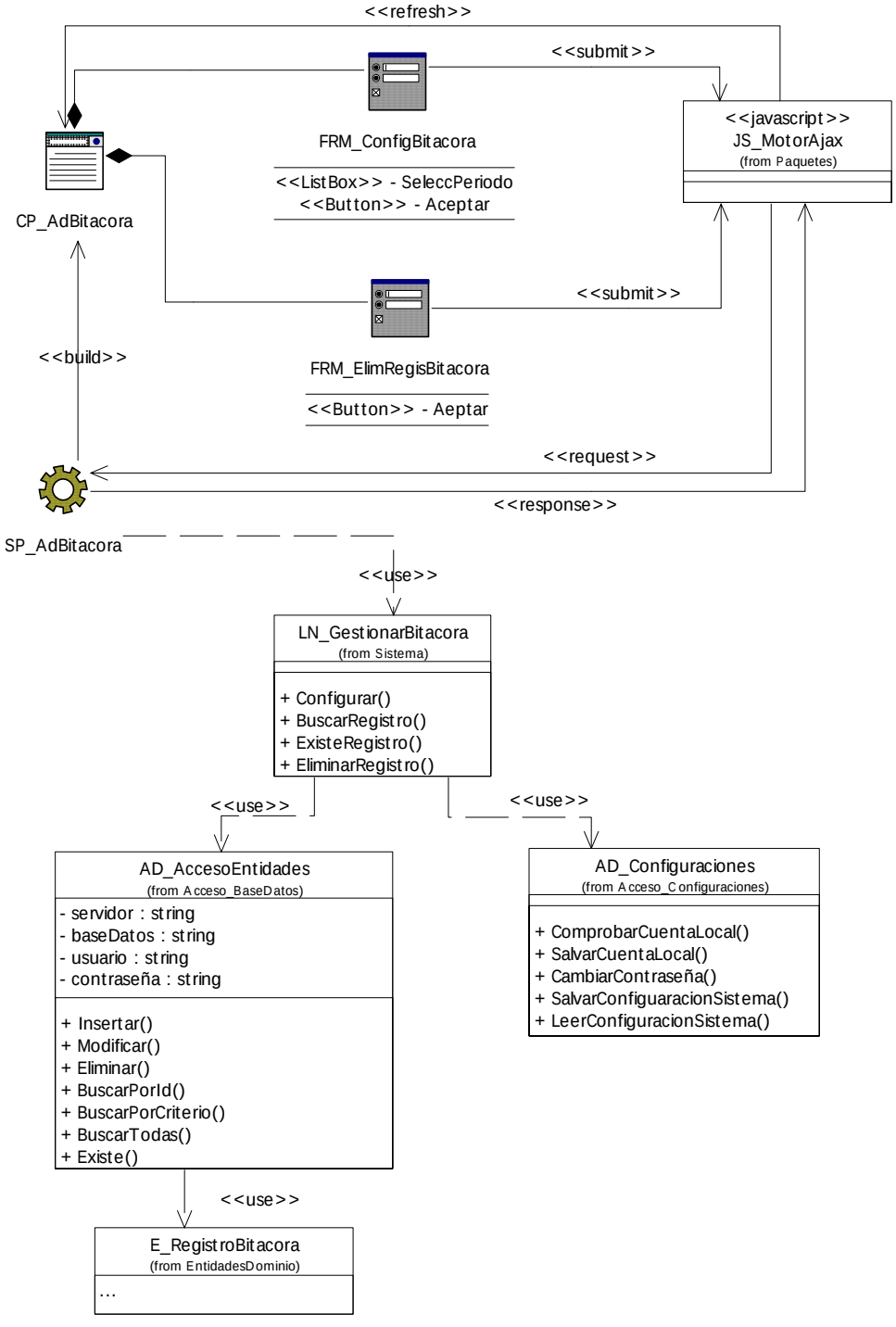


Fig 6. 12 Diagrama de clases. CU Gestionar cuenta de administración local.

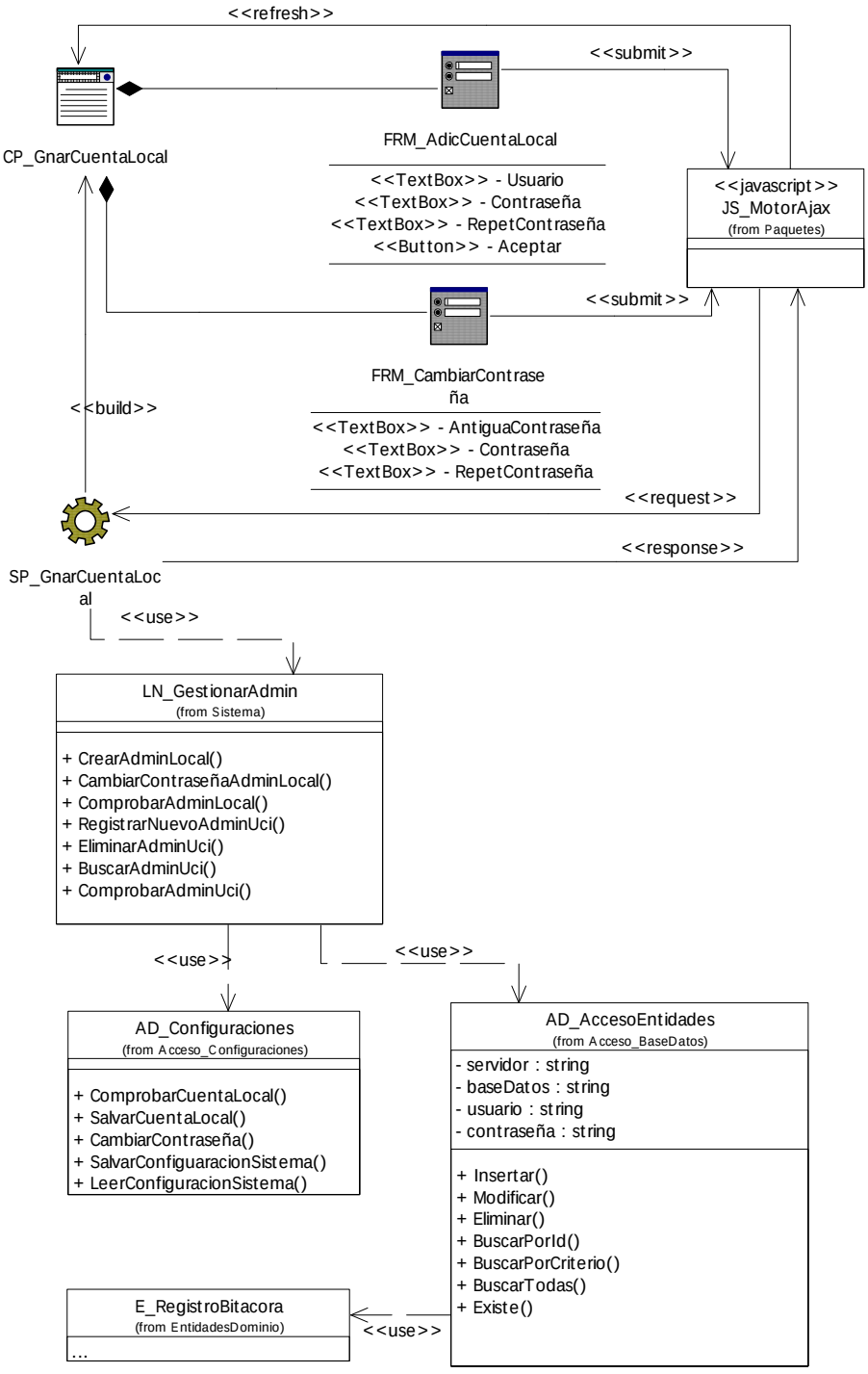
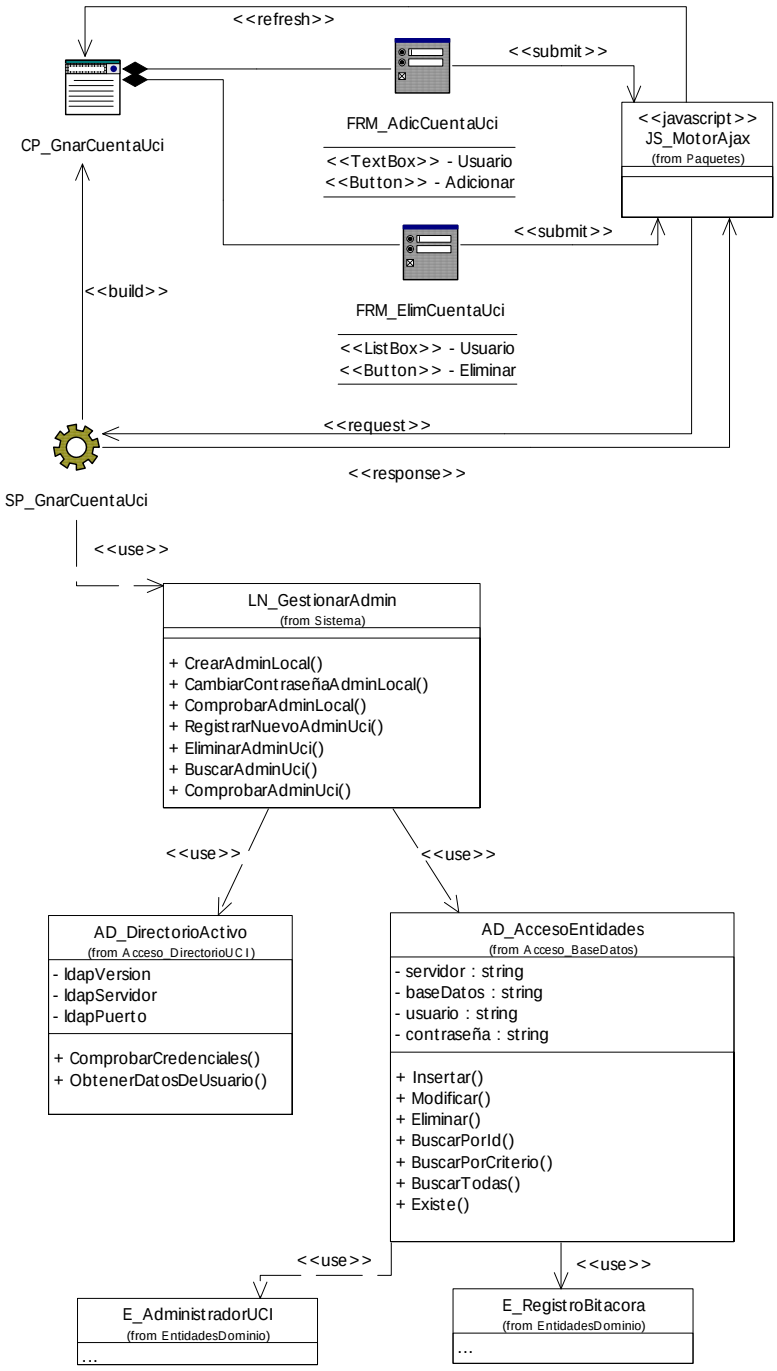


Fig 6. 13 Diagrama de clases. Gestionar cuentas de administrador UCI.



Anexo 3. Diagramas de componentes

Fig 6. 14 Diagrama de componentes. Paquete Aplicación.

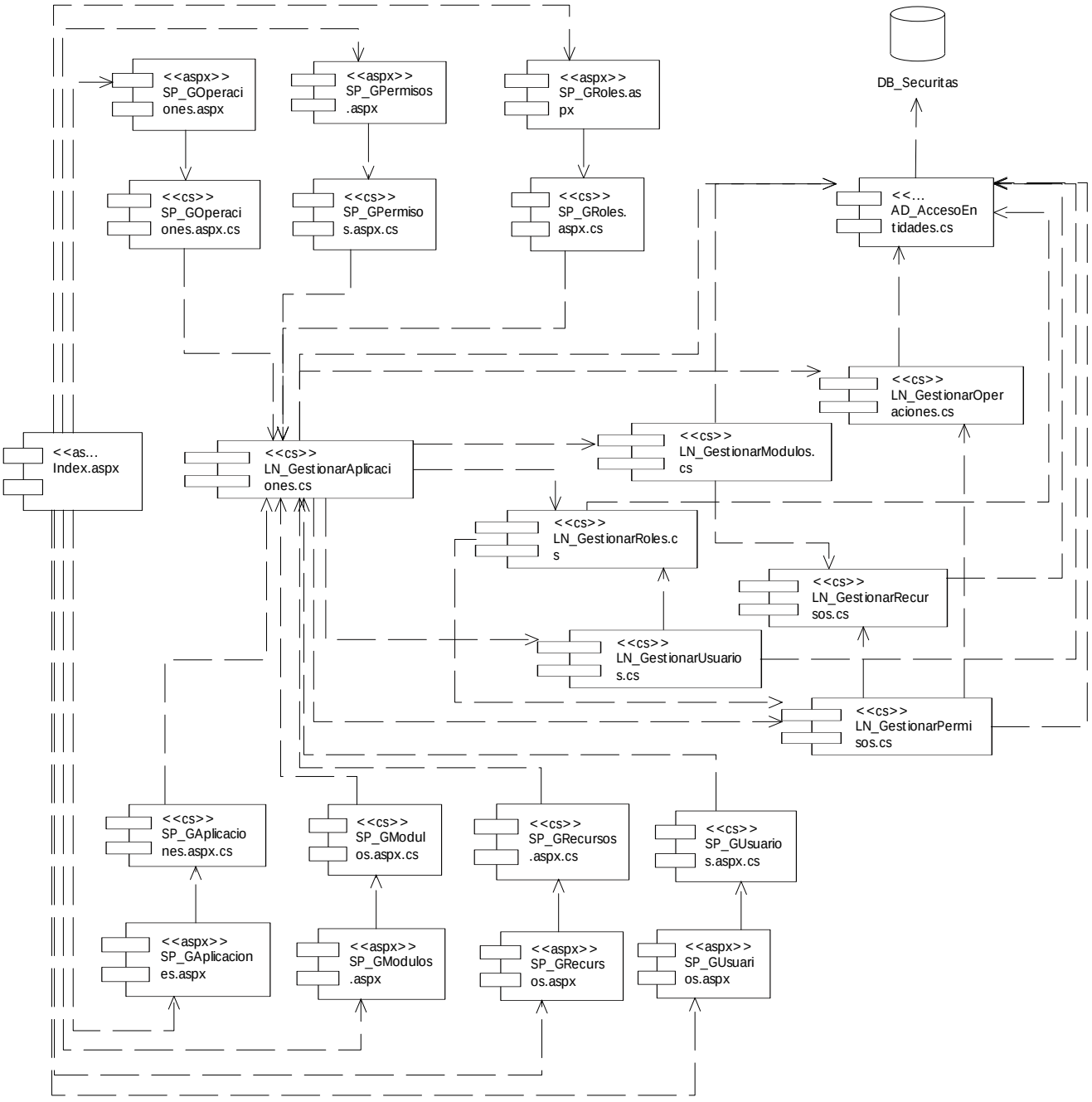


Fig 6. 15 Diagrama de componentes. Paquete Autenticación.

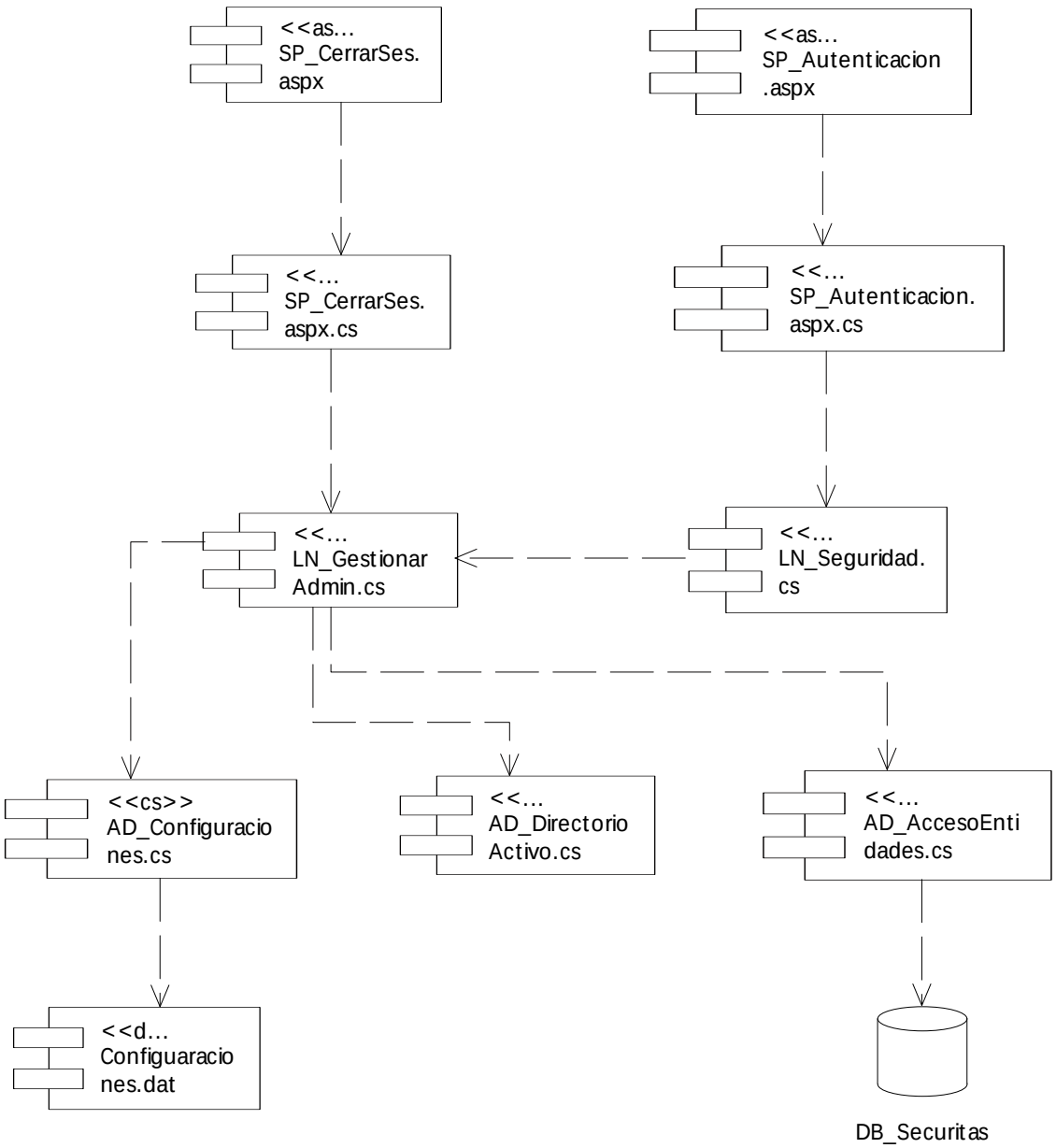


Fig 6. 16 Diagrama de componentes. Paquete Reportes.

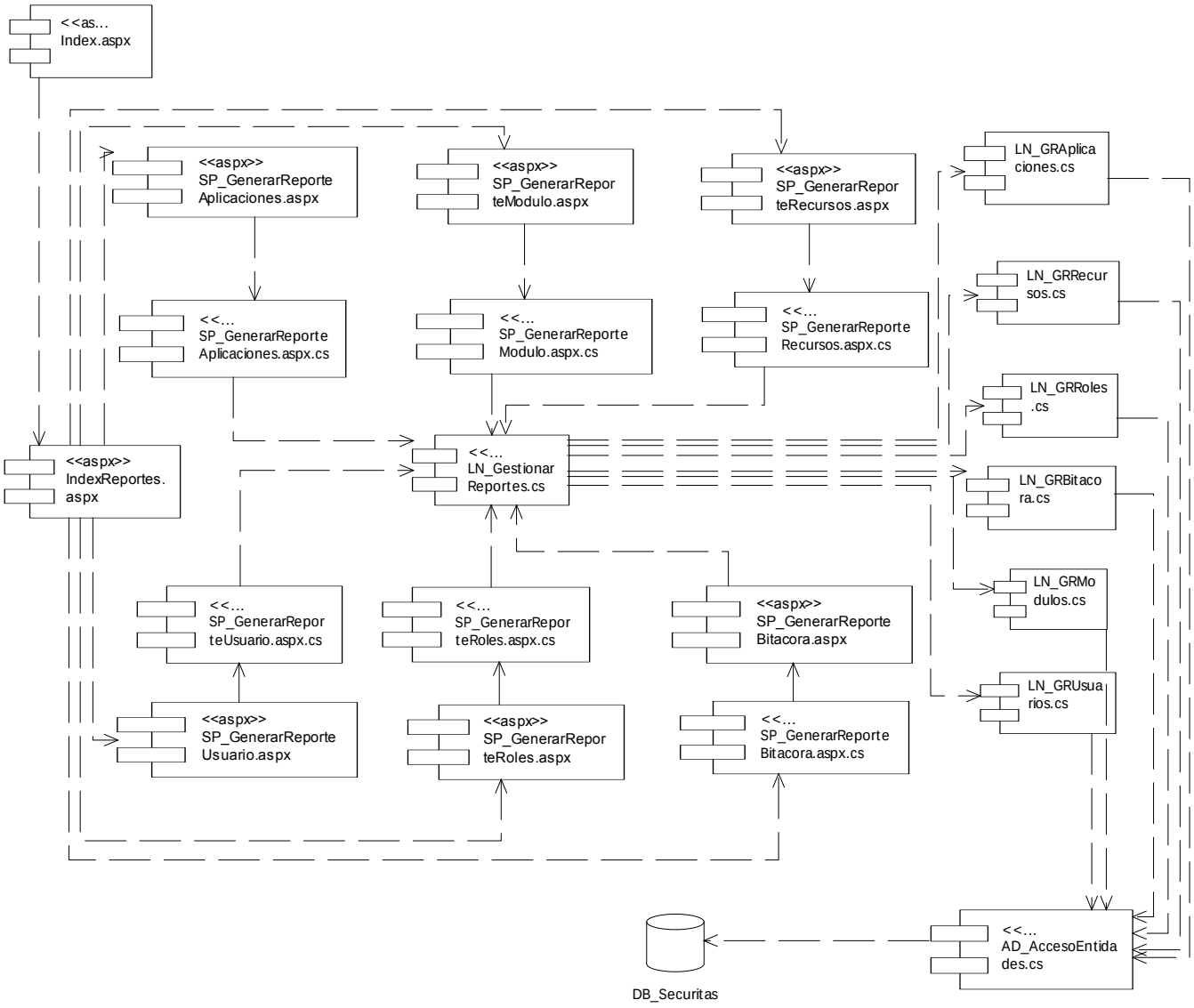


Fig 6. 17 Diagrama de componentes. Paquete Servicios.

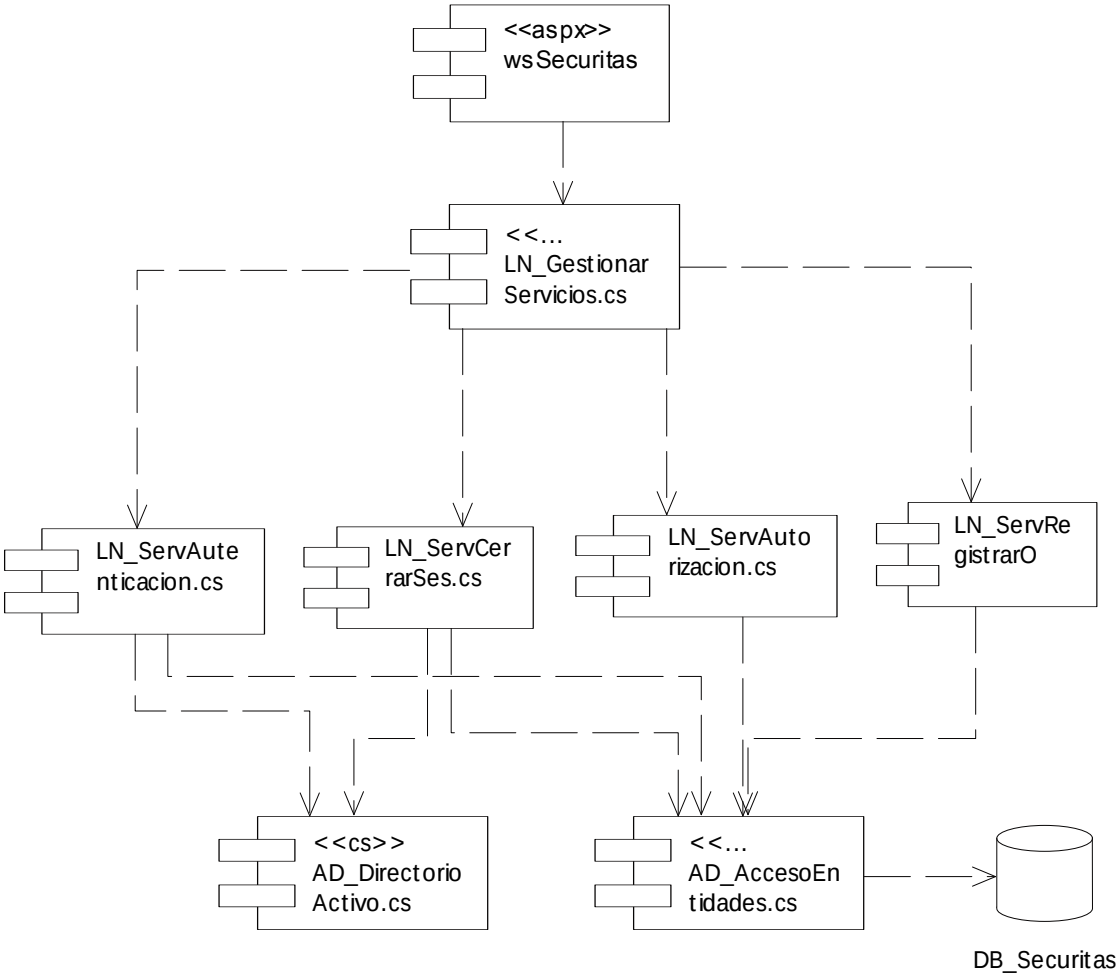
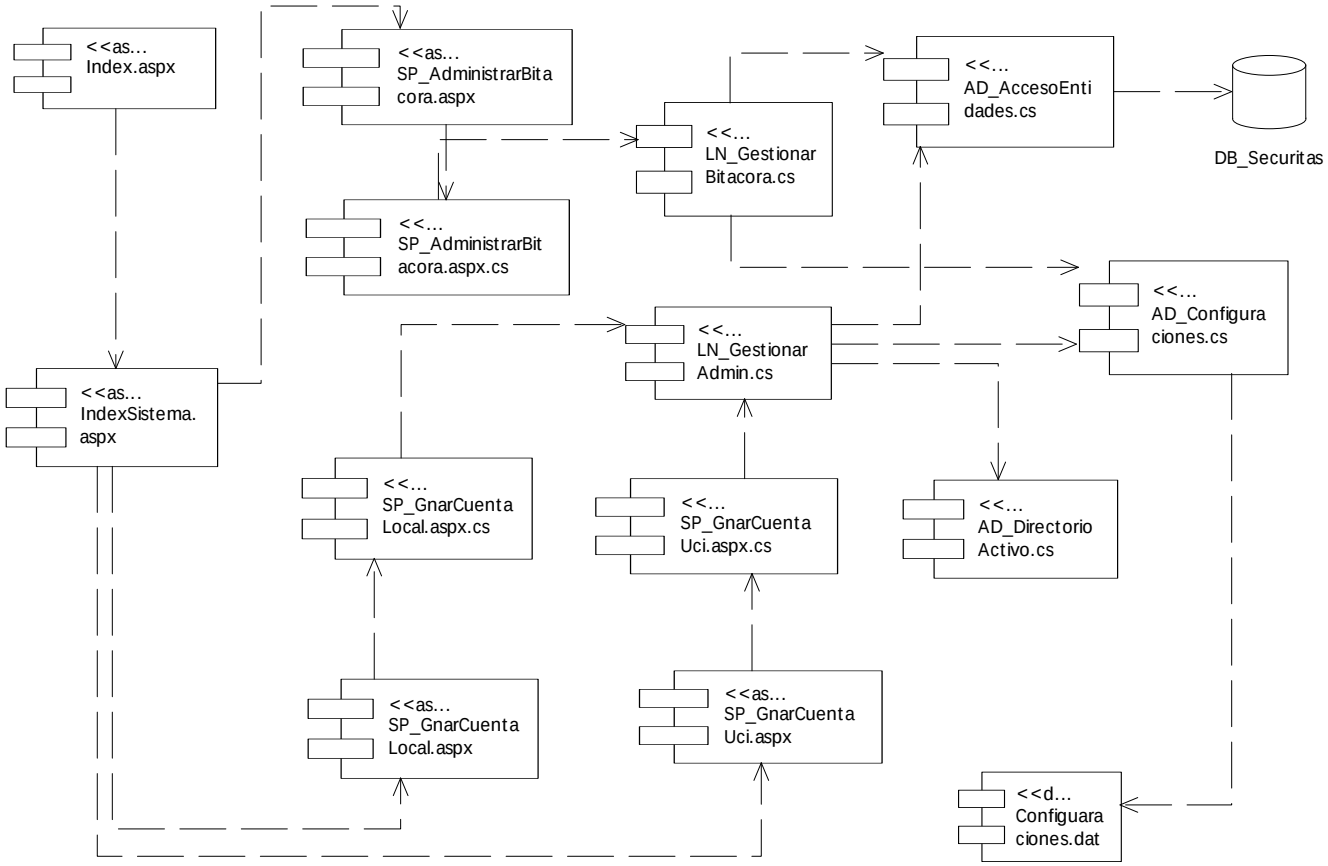


Fig 6. 18 Diagrama de componentes. Paquete Sistema.



Anexo 4. Medidas de seguridad

Medidas de seguridad diseñadas para el Sistema de seguridad centralizada de aplicaciones web de la facultad 9

Medidas físicas

- Restringir el acceso físico al sistema por parte de personal no autorizado.
- Proteger el sistema de cómputo contra daños medioambientales en lugares seguros.
- Establecer mecanismos de recuperación en caso de fallos.

Medidas lógicas

- Implementar el diseño propuesto estableciendo un adecuando control de acceso al sistema.
- Asegurar los datos que viajan desde el cliente hasta el servidor del sistema en cuestión usando protocolo https.
- Realizar copias de respaldo de la información que almacena el sistema periódicamente (cada 15 días)
- Situar en la PC servidora un cortafuegos o firewall como medida preventiva.
- Situar un sistema detector de intrusos en la pc servidora como medida preventiva.
- Implementar correctamente el sistema de auditoría propuesto haciendo uso adecuado de la bitácora.

Medidas administrativas

- Dar a conocer las políticas de seguridad para el sistema al personal inmerso en el proceso.
- Establecer, de requerirlo, un plan de formación en estas políticas para el personal inmerso en el proceso.

Anexo 5 Diferentes vistas del prototipo no funcional

Fig 6. 19 Vista del caso de uso Autenticar administrador

Fig 6. 20 Vista del caso de uso Gestionar aplicaciones Web

Fig 6. 21 Vista del caso de uso Gestionar permiso

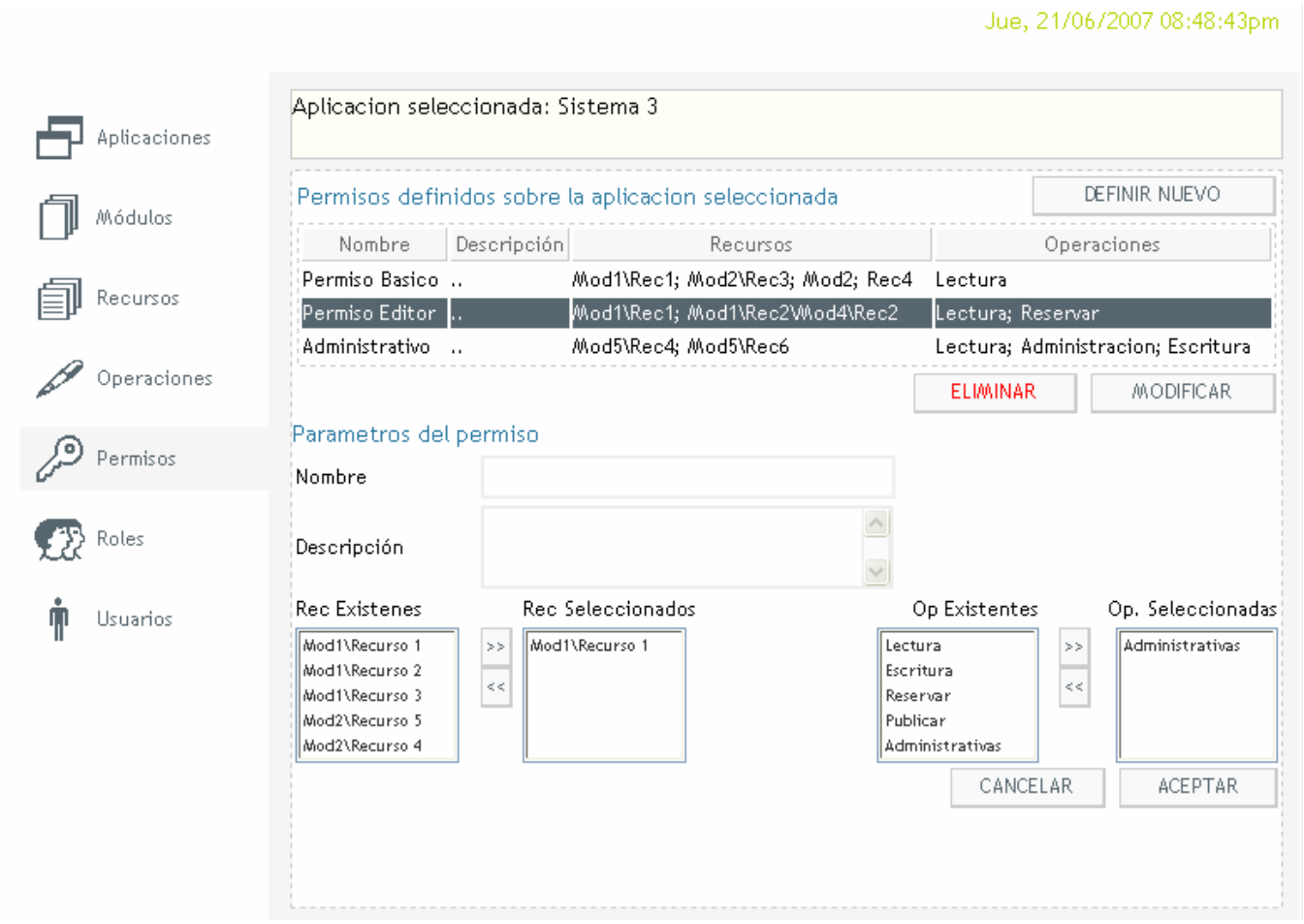


Fig 6. 22 Vista del caso de uso Gestionar operaciones

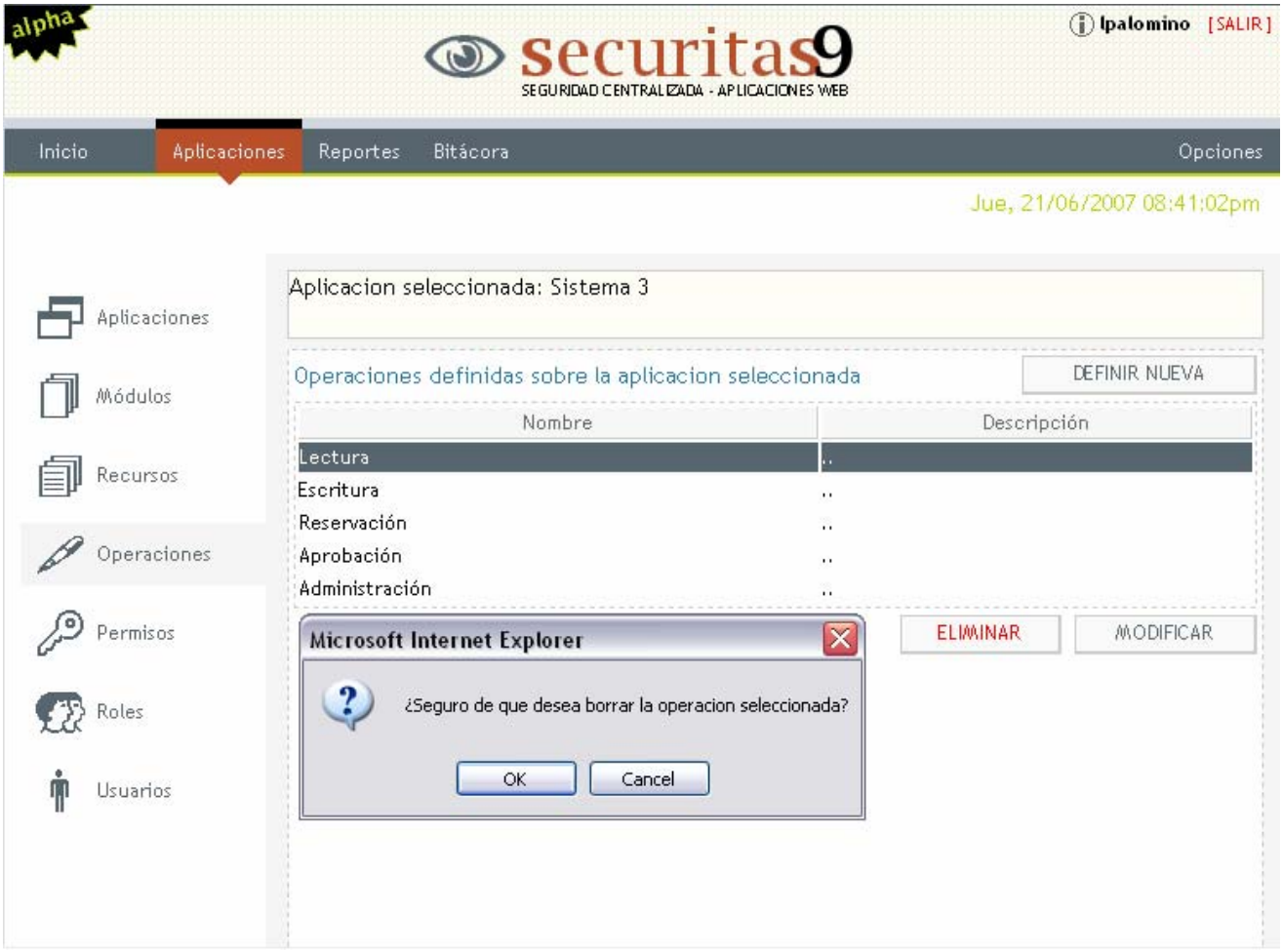


Fig 6. 23 Vista del caso de uso Generar reporte de aplicación

GLOSARIO DE TÉRMINOS

CASE: Acrónimo inglés de *Computer Aided Software Engineering*, que significa Ingeniería de Software Asistida por Ordenador.

CORBA: (*Common Object Request Broker Architecture*) Es un estándar que establece una plataforma de desarrollo de sistemas distribuidos facilitando la invocación de métodos remotos bajo un paradigma orientado a objetos.

CSS: (Cross Site Scripting) Tipo de vulnerabilidad surgida como consecuencia de errores de filtrado de las entradas del usuario en aplicaciones Web. También es conocido como XSS.

DCOM: (*Distributed Component Object Model*) Modelo de Objeto Componente Distribuido. Es un juego de conceptos e interfaces de programa de Microsoft en el cual los objetos de programa del cliente pueden solicitar servicios de objetos de programa servidores en otros ordenadores dentro de una red.

DDL: (*Data Definition Lenguaje*) Lenguaje de definición de datos. Las sentencias DDL son aquellas utilizadas para la creación de una base de datos y todos sus componentes: tablas, índices, relaciones, disparadores, procedimientos almacenados, etcétera.

DML: (*Data Manipulation Lenguaje*) Lenguaje de manipulación de datos. Las sentencias DML son aquellas utilizadas para insertar, borrar, modificar y consultar los datos de una base de datos.

Hacker: pirata de la red.

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.

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.

IIS: *(Internet Information Server)* es el servidor web de Microsoft que corre sobre plataformas Windows.

NNTP: (*Network News Transport Protocol)* protocolo para la transferencia de noticias en red, es una aplicación de internet que consiste en un protocolo usado para la lectura y publicación de artículos de noticias en Usenet.

Reporte: Informe detallado sobre alguna información, o sobre el estado de la información.

RMI: *(Remote Method Invocation)* invocación de métodos remotos, estándar para manejo de objetos distribuidos que es parte de Java.

SMTP: *(Simple Mail Transfer Protocol)* protocolo simple de transferencia de correo electrónico. Protocolo de red basado en texto utilizado para el intercambio de mensajes de correo electrónico entre computadoras y/o distintos dispositivos.

SQL: (*Structured Query Language)* Lenguaje estándar de comunicación con bases de datos.

TCP/IP: es el estándar de protocolo de comunicaciones requerido por las computadoras que acceden a Internet.

Trigger: Un trigger o un disparador es un evento que se ejecuta en una base de datos cuando se cumple una condición establecida al realizar una operación de inserción, actualización o borrado.

Transact-SQL: (T-SQL) es el lenguaje de programación del SQL Server.

Windows: Microsoft Windows es el nombre de una familia de sistemas operativos no libres desarrollados por la empresa de software Microsoft Corporation.

XML: *Extensible Markup Language* (Lenguaje extensible de etiquetas) Es un meta-lenguaje que permite definir lenguajes de marcado adecuado a usos determinados.