

005.8
Pro
A
TD-0180-06

TD-0180-06

Instituto Superior Politécnico "José Antonio Echeverría"



Facultad de Ingeniería Industrial
Centro de Estudios de Ingeniería y Sistemas

ARQUITECTURA DE SEGURIDAD PARA APLICACIONES WEB EMPRESARIALES

Trabajo de diploma para optar por el título de Ingeniería en Informática

Autor: Yuniel Eliades Proenza Arias.

Tutor: Ing. Alexander Cárdenas Pérez.

Ciudad de La Habana, Cuba
Junio, 2006

Resumen

Actualmente la forma de manejar la seguridad dentro de la entidad no es de una manera centralizada, se propone, con esta herramienta, lograr el control centralizado de la seguridad y a la vez lograr la interoperabilidad entre las aplicaciones, dando la posibilidad de que sea accesible desde cualquier plataforma bajo la que este implementado el software. Se logra, con el desarrollo de esta aplicación asegurar el elemento más importante de la institución: el manejo seguro de información puramente confidencial y sensible.

Con este trabajo se da un paso de avance en la automatización del proceso de control de la seguridad al permitir la creación de un único sistema capaz de regular los procesos de acceso y gestión de la información en las aplicaciones empresariales

Índice

INTRODUCCIÓN.....	1
1 FUNDAMENTACIÓN.....	7
1.1 INTRODUCCIÓN.....	7
1.1.1 <i>La seguridad en aplicaciones Web</i>	7
<i>Ataques pasivos</i>	10
<i>Ataques activos</i>	10
1.1.2 <i>Metodología a utilizar</i>	13
1.1.3 <i>Tecnologías actuales</i>	15
1.2 OBJETO DE ESTUDIO.....	26
1.2.1 <i>Objetivos estratégicos de la organización</i>	26
1.2.2 <i>Flujo actual de los procesos</i>	26
1.2.3 <i>Análisis crítico de la ejecución de los procesos</i>	27
1.3 PROCESOS OBJETO DE AUTOMATIZACIÓN.....	27
1.4 SISTEMAS AUTOMATIZADOS EXISTENTES VINCULADOS AL CAMPO DE ACCIÓN.....	29
1.5 FUNDAMENTACIÓN DE LOS OBJETIVOS.....	30
1.6 TENDENCIAS Y TECNOLOGÍAS ACTUALES.....	32
1.7 CONCLUSIONES.....	36
2 MODELO DEL DOMINIO.....	37
2.1 INTRODUCCIÓN.....	37
2.2 DEFINICIÓN DE LAS ENTIDADES Y LOS CONCEPTOS PRINCIPALES.....	37
2.3 REPRESENTACIÓN DEL MODELO DEL DOMINIO.....	38
2.4 CONCLUSIONES.....	39
3 REQUERIMIENTOS.....	40
3.1 INTRODUCCIÓN.....	40
3.2 ACTORES DEL SISTEMA A AUTOMATIZAR.....	40
3.3 DEFINICIÓN DE LOS REQUISITOS FUNCIONALES Y NO FUNCIONALES.....	40
3.3.1 <i>Requisitos no funcionales</i>	40
3.3.2 <i>Requisitos funcionales</i>	43
3.4 PAQUETES Y SUS RELACIONES.....	46
3.5 DIAGRAMA DE CASOS DE USO DEL SISTEMA A AUTOMATIZAR.....	47
3.6 DESCRIPCIÓN DE LOS CASOS DE USO.....	49
3.7 DESCRIPCIÓN EXPANDIDA DE LOS CASOS DE USO PRINCIPALES.....	56
3.8 CONCLUSIONES.....	69

4	DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA	70
4.1	INTRODUCCIÓN	70
4.2	PRINCIPIOS DE DISEÑO	70
4.2.1	<i>Descripción de las clases utilizadas</i>	70
4.2.2	<i>Mecanismo de Acceso a datos</i>	71
4.3	DIAGRAMA DE CLASES DEL DISEÑO	73
4.3.1	<i>Caso de uso: Configurar Sistemas</i>	73
4.3.2	<i>Caso de uso: Configurar Grupo de usuarios</i>	74
4.3.3	<i>Caso de uso: Configurar Roles</i>	75
4.3.4	<i>Caso de uso: Insertar Usuario</i>	76
4.3.5	<i>Caso de uso: Actualizar Usuario</i>	77
4.3.6	<i>Caso de uso: Configurar Páginas</i>	78
4.3.7	<i>Caso de uso: Configurar Funcionalidades</i>	79
4.3.8	<i>Caso de uso: Configurar Eventos</i>	80
4.3.9	<i>Caso de uso: Configurar Módulos</i>	81
4.3.10	<i>Caso de uso: Crear Perfil</i>	82
4.3.11	<i>Caso de uso: Eliminar Perfil</i>	83
4.3.12	<i>Caso de uso: Registrar Evento</i>	84
4.3.13	<i>Caso de uso: Autenticar usuario</i>	85
4.3.14	<i>Caso de uso: Visualizar Registro</i>	86
4.4	DIAGRAMAS DE COMPONENTES	87
4.5	AYUDA	88
4.6	TRATAMIENTO DE ERRORES	88
4.7	ESTÁNDAR DE IMPLEMENTACIÓN	88
4.8	DISEÑO DE LA BASE DE DATOS	99
4.8.1	<i>Modelo lógico de datos</i>	99
4.8.2	<i>Modelo físico de datos</i>	100
4.9	DIAGRAMA DE DESPLIEGUE	101
4.10	CONCLUSIONES	101
5	ESTUDIO DE FACTIBILIDAD	102
5.1	INTRODUCCIÓN	102
5.2	PLANIFICACIÓN BASADA EN CASOS DE USO	102
5.3	BENEFICIOS TANGIBLES E INTANGIBLES	104
5.4	ANÁLISIS DE COSTOS Y BENEFICIOS	104
5.5	CONCLUSIONES	104
	CONCLUSIONES	105
	RECOMENDACIONES	106

BIBLIOGRAFÍA.....	107
GLOSARIO DE TÉRMINOS	108
ANEXO 1 DESCRIPCIÓN EXPANDIDA DE LOS CASOS DE USO	I
ANEXO 2 DIAGRAMAS DE COMPONENTES.....	XIV
ANEXO 3 DIAGRAMA DE CLASES DEL DISEÑO VALIDAR USUARIO	XVIII

Índice de tablas

Tabla 1. Definición de actores del sistema a automatizar 40

Índice de figuras

Figura 1. Diagrama de los casos de uso del paquete: <i>Administrar Sistemas</i>	47
Figura 2. Diagrama de los casos de uso del paquete: <i>Administrar Usuarios</i>	48
Figura 3. Diagrama de los casos de uso del paquete: <i>Servicios</i>	48
Figura 4. Diagrama de los casos de uso del paquete: <i>Autenticación</i>	49
Figura 5. Diagrama de clases del caso de uso	73
Figura 6. Diagrama de clases del caso de uso	74
Figura 7. Diagrama de clases del caso de uso	75
Figura 8. Diagrama de clases del caso de uso	76
Figura 9. Diagrama de clases del caso de uso	77
Figura 10. Diagrama de clases del caso de uso	78
Figura 11. Diagrama de clases del caso de uso	79
Figura 12. Diagrama de clases del caso de uso	80
Figura 13. Diagrama de clases del caso de uso	81
Figura 14. Diagrama de clases del caso de uso	82
Figura 15. Diagrama de clases del caso de uso	83
Figura 16. Diagrama de clases del caso de uso	84
Figura 17. Diagrama de clases del caso de uso	85
Figura 18. Diagrama de clases del caso de uso	86
Figura 19 Diagrama de componentes.	87
Figura 20. Diagrama de despliegue	101

Introducción

El Ministerio de las Fuerzas Armadas Revolucionarias cuenta con un conjunto de sistemas para el procesamiento de la información, implementados en diferentes plataformas, cada uno concebido de manera independiente. Surge entonces la necesidad de relacionar todos los sistemas y a la vez controlar el acceso de los usuarios a estos de una manera centralizada. La manera en que está implementada la seguridad implica numerosas dificultades.

Actualmente en la institución se registran una serie de problemas que afectan directamente a los usuarios que realizan operaciones sobre los sistemas y a nuestro juicio constituyen la situación problemática a las que haremos frente con el presente trabajo:

- Cada sistema implementa su propia arquitectura y elementos de seguridad.
- Los usuarios se ven obligados a memorizar varios nombres de usuario y claves de acceso para acceder a los diferentes sistemas.
- No existe una definición clara sobre los sucesos e informaciones a auditar en cada uno de los sistemas.
- No existe la posibilidad de realizar la administración centralizada de los perfiles de usuario para las aplicaciones desarrolladas en la entidad.
- No existe la posibilidad por parte de los órganos responsables de analizar violaciones o riesgos de violaciones en el acceso a la información.

Todo lo anterior está provocado, básicamente, por **la no existencia de una arquitectura centralizada de seguridad y auditoría aplicable a los productos de software de gestión que se desarrollan en la entidad**, esto se puede definir como la causa que provoca la situación problemática y por ende **el problema a resolver**. La definición de una adecuada política de seguridad y

auditoría en las aplicaciones constituye un elemento crucial para el buen desempeño de las mismas en un entorno empresarial. Al lograr lo anteriormente planteado, se produce una disminución de los tiempos de desarrollo del resto de las soluciones, los usuarios finales se desenvuelven bajo una política que les permite un único acceso controlado a aquellas aplicaciones con derecho y además las labores de administración y manejo de los perfiles de dichos usuarios se mantienen bajo un control exhaustivo a partir de los roles que poseen los mismos en los módulos de los diferentes sistemas. Con la implementación de una arquitectura centralizada se hace menos engorrosa la labor de administración de la seguridad de los sistemas, además se definen de manera general todos los posibles elementos a auditar sobre el acceso a los sistemas y modificación de los datos dentro de los mismos. Se brinda una herramienta útil a los órganos responsables para analizar violaciones de acceso a la información disponible en las aplicaciones.

El trabajo presenta una solución a la altura de las tecnologías actuales, utilizando una manera óptima para lograr resolver el problema del control de acceso en las aplicaciones de la institución. Este último es la causa de la poca eficiencia de dicho proceso, siendo su inmediata solución una tarea de alta prioridad por los riesgos que entraña y las molestias que causa a los usuarios de todos los sistemas.

Durante los estudios realizados en el momento de acometer la ejecución de este trabajo se llegó a la conclusión de la no existencia en la organización de antecedente alguno sobre el mismo, en el país y en el mundo aunque existen sistemas que tocan el tema estos no se ajustan a los requerimientos de la entidad. Existen sistemas para el control de la seguridad pero no enfocan el asunto a manera de servicio Web.

Este trabajo brindará un valioso aporte a la institución al permitir nuevas concepciones en lo referente al control de la seguridad de cada una de las

aplicaciones desarrolladas en la institución, siendo impersonal a cada una de estas y así revoluciona todos los conceptos de seguridad y auditoría que se tenían hasta el momento, lo cual brinda soporte a las políticas de seguridad de los distintos sistemas con que cuenta la entidad.

Objeto de estudio

Nuestro objeto de estudio se basa en el control de seguridad y auditoría en aplicaciones empresariales.

Campo de acción

Se deben tener en cuenta los siguientes aspectos:

1- El proceso de control de los accesos de los usuarios a través de:

- **Dominios de aplicación:** Especifica el nivel jerárquico al que pertenece el usuario dentro de la institución.
- **Especialidad:** El área a que pertenece dentro de la empresa, para un conjunto de usuarios con esta información se garantiza el acceso único a la información de su área.
- **Grupos de usuarios:** Se dividen los tipos de usuarios según los privilegios que se otorgan, así se asegura que los usuarios realicen solamente acciones inherentes a su grupo, que desempeña determinados roles.
- **Usuarios Independientes:** Este tipo de usuario cumple roles específicos dentro de las aplicaciones. Puede no pertenecer a ningún grupo.
- **Roles:** Papel que juega un usuario en la aplicación. Garantizan el control de los usuarios descrito en las especificaciones.
- **Especificaciones de acceso para Roles:**

-
- *A funcionalidades de sistemas:* Se especifican accesos a funcionalidades que tienen alcance a nivel de sistema.
 - *A funcionalidades de módulos:* Se especifican accesos a funcionalidades que tienen alcance solamente a nivel de módulos.

2- El proceso de archivar la información del estado del sistema y de los eventos realizados por los diferentes usuarios a través de:

- **Información sobre cantidad de usuarios que en un instante dado:**
 - *Usan el servicio:* Se especifica el número de usuarios que usan el servicio en un instante dado.
 - *Acceden a un sistema:* Se especifica el número de usuarios y el sistema a que acceden en un instante dado.
- **Información del usuario:** Se especifica todo referente al usuario.
- **Tipo de acción realizada:**
 - *Acción de visualización sin generar salida:* Se basa en una consulta a datos que no genera ficheros de salida, solo se muestra la información en el navegador.
 - *Acción de visualización que genera salida:* La consulta genera un fichero de salida.
 - *acción de actualización:* Se basa en consultas que realizan las operaciones básicas sobre datos (insertar, modificar o eliminar).
 - *Acción de acceso a ficheros:* El usuario puede acceder a la información presente en un fichero. Para este caso se debe archivar la dirección de fichero accedido.
- **Contenido y resultado de la acción:** Se especifica el contenido de la consulta realizada y el resultado en caso de generar salida. Si la salida es un fichero se especifica su identificación.

- **Momento de ocurrencia:** Especifica el instante exacto de ocurrencia de la acción.
- **Información de los datos accedidos:** Especifica los datos accedidos.
- **Información de los sistemas o módulos accedidos:** Especifica el nombre del sistema sobre el que se realiza la acción y/o el módulo.

✓ Teniendo en cuenta todo lo anteriormente planteado proponemos la siguiente hipótesis:

-Si se realiza un producto de software basado en una arquitectura centralizada de seguridad y auditoría, usando la tecnología de Servicios Web se garantiza una gestión eficiente de estos elementos en las aplicaciones Web empresariales.

Para lograr la completa realización del producto proponemos como objetivos:

General:

- Confeccionar un producto de software multiplataforma basado en una arquitectura centralizada de seguridad y auditoría para su uso en las aplicaciones Web empresariales.

Específicos:

- Realizar todo el proceso de análisis y diseño de la aplicación a implementar.
- Lograr la separación en diferentes capas de las funcionalidades del software.

- Desarrollar servicios Web accesibles a todas las aplicaciones desarrolladas en la institución e independiente a la plataforma en que están implementadas, para el manejo de las políticas de seguridad y auditoría.
- Lograr un software que contribuya a mejoras en el proceso de administración.

1 Fundamentación

1.1 Introducción

En este capítulo se analizan los aspectos fundamentales a tener en cuenta para la realización del producto propuesto, teniendo en cuenta los elementos del objeto de estudio y el campo de acción. Además se realiza un análisis de las tecnologías a utilizar para su desarrollo teniendo en cuenta las ventajas que ofrecen y la relación directa con dicho producto. Todo parte de un análisis de la situación actual del control de la seguridad en la institución, las tecnologías usadas y las disponibilidades a escala mundial.

1.1.1 La seguridad en aplicaciones Web

La seguridad es un aspecto crítico de las aplicaciones Web. Las aplicaciones Web, por definición, permiten el acceso de usuarios a recursos centrales, el servidor Web y, a través de éste, a otros como los servidores de base de datos. Con los conocimientos y la implementación correcta de medidas de seguridad, puede proteger sus propios recursos así como proporcionar un entorno seguro donde los usuarios trabajen cómodos con su aplicación. Los elementos más importantes a tener en cuenta son:

- *Autenticación* confirma que los usuarios son quienes dicen que son. Por ejemplo, un usuario debe proporcionar un nombre de usuario y una contraseña que se comprueban en una autoridad (por ejemplo, una base de datos o un servidor de dominios de Windows).
- *Autorización* es el proceso de conceder o denegar el acceso a los recursos a usuarios específicos.

Estos conceptos pueden resultar familiares debido a su uso en Windows. En cualquier tipo de red, se debe iniciar la sesión (es decir, autenticarse). Una vez iniciada la sesión de forma satisfactoria, se puede tener acceso a archivos, carpetas, impresoras y otros recursos específicos (es decir, se dispone de autorización para obtener acceso a los recursos en función de las credenciales de inicio de sesión). Windows ofrece un sofisticado sistema de seguridad que permite a los administradores crear cuentas de usuario y autorizar a dichos usuarios a obtener acceso a carpetas, archivos, etc.

Las aplicaciones Web, por definición, permiten el acceso de usuarios a recursos centrales, el servidor Web y, a través de éste, a otros como los servidores de base de datos. Comprender e implementar las medidas de seguridad adecuadas permite:

- Proteger los recursos propios contra accesos no autorizados.
- Restringir los niveles de acceso por usuario o por rol.
- Establecer la integridad y confidencialidad de los datos, ofreciendo un entorno seguro en el que los usuarios se encuentren cómodos trabajando con la aplicación.
- Establecer control sobre cómo la aplicación obtiene acceso a recursos restringidos.
- Garantizar que el código de la aplicación se ejecuta de la forma esperada.

Amenazas a la Seguridad

Se entiende por amenaza una condición del entorno del sistema de información (persona, máquina, suceso o idea) que, dada una oportunidad, podría dar lugar a que se produjese una violación de la seguridad (confidencialidad, integridad, disponibilidad o uso legítimo).

Las amenazas a la seguridad en una red pueden caracterizarse modelando el sistema como un flujo de información desde una fuente, como por ejemplo un fichero o una región de la memoria principal, a un destino, como por ejemplo otro fichero o un usuario. Un ataque no es más que la realización de una amenaza.

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

- **Interrupción:** un recurso del sistema es destruido o se vuelve no disponible. Este es un ataque contra la disponibilidad. Ejemplos de este ataque son la destrucción de un elemento hardware, como un disco duro, cortar una línea de comunicación o deshabilitar el sistema de gestión de ficheros.
- **Intercepción:** una entidad no autorizada consigue acceso a un recurso. Este es un ataque contra la confidencialidad. La entidad no autorizada podría ser una persona, un programa o un ordenador. Ejemplos de este ataque son pinchar una línea para hacerse con datos que circulen por la red y la copia ilícita de ficheros o programas (intercepción de datos), o bien la lectura de las cabeceras de paquetes para desvelar la identidad de uno o más de los usuarios implicados en la comunicación observada ilegalmente (intercepción de identidad).
- **Modificación:** una entidad no autorizada no sólo consigue acceder a un recurso, sino que es capaz de manipularlo. Este es un ataque contra la integridad. Ejemplos de este ataque son el cambio de valores en un archivo de datos, alterar un programa para que funcione de forma diferente y modificar el contenido de mensajes que están siendo transferidos por la red.
- **Fabricación:** una entidad no autorizada inserta objetos falsificados en el sistema. Este es un ataque contra la autenticidad. Ejemplos de este ataque son la inserción de mensajes espurios en una red o añadir registros a un archivo.

Estos ataques se pueden asimismo clasificar de forma útil en términos de ataques pasivos y ataques activos.

Ataques pasivos

En los ataques pasivos el atacante no altera la comunicación, sino que únicamente la escucha o monitoriza, para obtener información que está siendo transmitida. Sus objetivos son la interceptación de datos y el análisis de tráfico, una técnica más sutil para obtener información de la comunicación, que puede consistir en:

- **Obtención del origen y destinatario** de la comunicación, leyendo las cabeceras de los paquetes monitorizados.
- **Control del volumen de tráfico** intercambiado entre las entidades monitorizadas, obteniendo así información acerca de actividad o inactividad inusuales.
- **Control de las horas habituales** de intercambio de datos entre las entidades de la comunicación, para extraer información acerca de los periodos de actividad.

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

Ataques activos

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

- **Suplantación de identidad:** el intruso se hace pasar por una entidad diferente. Normalmente incluye alguna de las otras formas de ataque activo. Por ejemplo, secuencias de autenticación pueden ser capturadas y repetidas, permitiendo a una entidad no autorizada acceder a una serie de

recursos privilegiados suplantando a la entidad que posee esos privilegios, como al robar la contraseña de acceso a una cuenta.

- **Reactuación:** uno o varios mensajes legítimos son capturados y repetidos para producir un efecto no deseado, como por ejemplo ingresar dinero repetidas veces en una cuenta dada.
- **Modificación de mensajes:** una porción del mensaje legítimo es alterada, o los mensajes son retardados o reordenados, para producir un efecto no autorizado. Por ejemplo, el mensaje "Ingresa un millón de pesetas en la cuenta A" podría ser modificado para decir "Ingresa un millón de pesetas en la cuenta B".
- **Degradación fraudulenta del servicio:** impide o inhibe el uso normal o la gestión de recursos informáticos y de comunicaciones. Por ejemplo, el intruso podría suprimir todos los mensajes dirigidos a una determinada entidad o se podría interrumpir el servicio de una red inundándola con mensajes espurios. Entre estos ataques se encuentran los de **denegación de servicio**, consistentes en paralizar temporalmente el servicio de un servidor de correo, Web, FTP, etc.

Servicios de Seguridad

Para hacer frente a las amenazas a la seguridad del sistema se definen una serie de servicios para proteger los sistemas de proceso de datos y de transferencia de información de una organización. Estos servicios hacen uso de uno o varios mecanismos de seguridad. Una clasificación útil de los servicios de seguridad es la siguiente:

- **Confidencialidad:** requiere que la información sea accesible únicamente por las entidades autorizadas. La confidencialidad de datos se aplica a todos los datos intercambiados por las entidades autorizadas o tal vez a sólo porciones o segmentos seleccionados de los datos, por ejemplo mediante cifrado. La confidencialidad de flujo de tráfico protege la identidad

del origen y destino(s) del mensaje, por ejemplo enviando los datos confidenciales a muchos destinos además del verdadero, así como el volumen y el momento de tráfico intercambiado, por ejemplo produciendo una cantidad de tráfico constante al añadir tráfico espurio al significativo, de forma que sean indistinguibles para un intruso.

- **Autenticación:** requiere una identificación correcta del origen del mensaje, asegurando que la entidad no es falsa. Se distinguen dos tipos: de entidad, que asegura la identidad de las entidades participantes en la comunicación, mediante biométrica (huellas dactilares, identificación de iris, etc.), tarjetas de banda magnética, contraseñas, o procedimientos similares; y de origen de información, que asegura que una unidad de información proviene de cierta entidad, siendo la firma digital el mecanismo más extendido.
- **Integridad:** requiere que la información sólo pueda ser modificada por las entidades autorizadas. La modificación incluye escritura, cambio, borrado, creación y reactuación de los mensajes transmitidos. La integridad de datos asegura que los datos recibidos no han sido modificados de ninguna manera, mientras que la integridad de secuencia de datos asegura que la secuencia de los bloques o unidades de datos recibidas no ha sido alterada y que no hay unidades repetidas o perdidas.
- **No repudio:** ofrece protección a un usuario frente a que otro usuario niegue posteriormente que en realidad se realizó cierta comunicación. Esta protección se efectúa por medio de una colección de evidencias irrefutables que permitirán la resolución de cualquier disputa. El no repudio de origen protege al receptor de que el emisor niegue haber enviado el mensaje, mientras que el no repudio de recepción protege al emisor de que el receptor niegue haber recibido el mensaje. Las firmas digitales constituyen el mecanismo más empleado para este fin.
- **Control de acceso:** requiere que el acceso a los recursos (información, capacidad de cálculo, nodos de comunicaciones, entidades físicas, etc.) sea controlado y limitado por el sistema destino, mediante el uso de contraseñas o llaves hardware.

-
- **Disponibilidad:** requiere que los recursos del sistema informático estén disponibles a las entidades autorizadas cuando los necesiten.

1.1.2 Metodología a utilizar

Actualmente existen ciertas tendencias fundamentadas en la idea de construir sistemas más grandes y complejos. Se quiere un software que este mejor adaptado a nuestras necesidades lo que a su vez hace que el software sea más complejo, pero no solo eso, sino que además lo queremos lo más rápido posible.

Sin embargo la mayoría de los desarrolladores siguen haciendo software con los mismos métodos de hace 20 años, sin percatarse de la importancia que tiene la presencia de un proceso bien definido y bien gestionado, siendo este elemento el que marca una diferencia esencial entre proyectos hiperproductivos y otros que fracasan.

¿Que es entonces un proceso de desarrollo de software?

Es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software.

RUP

Rational Unified Process (RUP) es una propuesta de proceso para el desarrollo de software basada en la orientación a objetos, el desarrollo iterativo y el modelamiento visual usando Unified Modeling Language (UML) para describir un sistema, lo cual permite incorporar al proceso de desarrollo de software un mejor control de los requerimientos y cambios. Posibilita la distribución del trabajo en diversos frentes de forma simultánea.

A pesar de ser una metodología desarrollada directamente para el trabajo con clases y objetos brinda amplias posibilidades con el manejo eficiente del tiempo de diseño e implementación de aplicaciones Web.

El Proceso Unificado fue desarrollado por Philippe Kruchten, Ivar Jacobson y otros de la Rational como el proceso complementario al UML.

Hay que destacar que el RUP capacita a las organizaciones de muchas maneras, la más significativa es que proporciona la forma en la que el equipo de proyecto puede trabajar de una forma más conjunta con los clientes y demás implicados. Lo que favorece una mayor organización y entendimiento de lo que realmente el cliente necesita y una excelente proyección del proyecto.

Características de RUP

- Manejado por Casos de Uso.

Un sistema software ve la luz para dar servicio a sus usuarios. Por tanto, para construir un sistema con éxito debemos conocer lo que sus futuros usuarios necesitan y desean.

Un caso de uso es un fragmento de funcionalidad del sistema que proporciona al usuario un resultado importante. Representan requerimientos funcionales. Todos los casos de uso juntos constituyen el modelo de casos de uso el cual describe la funcionalidad total del sistema. Los casos de uso guían la arquitectura del sistema y la arquitectura del sistema influye en la selección de los casos de uso.

- Centrado en la arquitectura.

La arquitectura en un sistema software se describe mediante diferentes vistas del sistema en construcción. Este concepto incluye los aspectos estáticos y dinámicos más significativos del sistema. Esta se refleja en los casos de uso pues cada producto tiene tanto una función como una forma, ninguna es suficiente por si sola.

- Iterativo e incremental.

Resulta práctico dividir el trabajo en partes más pequeñas o mini proyectos los que no son más que iteraciones que resultan en un incremento.

Una iteración es una secuencia de actividades con un plan establecido y criterios de evaluación, cuyo resultado es una versión del software

Beneficios de la iteración:

- Reduce el coste del riesgo al coste de un solo incremento.
- Menos riesgo de no sacar el producto al mercado en fecha.
- Acelera el ritmo de desarrollo.
- Las necesidades del usuario y correspondientes requisitos no pueden definirse completamente al principio. Se requieren iteraciones sucesivas.

1.1.3 Tecnologías actuales

1.1.3.1 Aplicaciones Web

Una Aplicación Web es un conjunto de páginas Web enlazadas que visualizan diferentes partes de la información que se quiere mostrar a través de ella. Constituye una de las mejores herramientas para divulgar, gestionar y compartir la información por lo que trae consigo un aumento de la eficiencia en cuanto a la manipulación de gran cantidad de datos.

Actualmente se ha incrementado el uso de las aplicaciones distribuidas, y con ello el uso de la Internet por las grandes ventajas que esta ofrece. El futuro de las grandes empresas es integrarse entre ellas y los caminos a la integración se abren por el uso de aplicaciones Web.

Existen numerosas aplicaciones de Software Libre para la implementación de aplicaciones en plataforma Web, generando esto una gran ventaja para el desarrollo de dichas aplicaciones y siendo de gran utilidad para los que desean incursionar en este ventajoso mundo.

1.1.3.2 Servicios Web

Según la W3C (el organismo que se encarga de desarrollar gran parte de los estándares de Internet), se define de la siguiente forma: "Un servicio Web es una aplicación software identificada mediante una URI, cuyo interfaz (y uso) es capaz de ser definido, descrito y descubierto mediante artefactos XML, y soportar interacciones directas con otras aplicaciones software usando mensajes basados en XML y protocolos basados en Internet".

Los servicios Web, no son por tanto aplicaciones con una interfaz gráfica con la que las personas puedan interactuar, sino que son software accesible en Internet (o en redes privadas que usen tecnologías Internet) por otras aplicaciones. De esta forma podemos desarrollar aplicaciones que hagan uso de otras aplicaciones que estén disponibles en Internet interactuando con ellas.

Básicamente los servicios Web permiten que diferentes aplicaciones, realizadas con diferentes tecnologías, y ejecutándose en toda una variedad de entornos, puedan comunicarse e integrarse, lo cual es muy importante para el sistema que se quiere implementar.

Un servicio Web es un componente software que se basa en las siguientes tecnologías:

-Un formato que describa la interfaz del componente (sus métodos y atributos) basado en XML. Por lo general este formato es el WSDL (Web Service Description Language), Lenguaje de descripción de servicios Web.

-Un protocolo de aplicación basado en mensajes y que permite que una aplicación interactúe (use, instancia, llame, ejecute) al servicio. Por lo general este protocolo es SOAP (Simple Object Access Protocol), Protocolo Simple de Acceso a Datos.

-Un protocolo de transporte que se encarga de transportar los mensajes por Internet. Por lo general este protocolo de transporte es HTTP (Hiper-Text Transport Protocol), Protocolo de Transferencia de Hipertexto, que es exactamente el mismo que usamos para navegar por la Web.

1.1.3.3 Lenguajes de Programación Web

ASP

ASP no es realmente un lenguaje como tal, es el acrónimo de Active Server Pages, el lenguaje usado en realidad para programar ASP es Visual Basic Script o JScript. El mayor inconveniente de ASP es que se trata de un sistema propietario que es usado nativamente sólo por Microsoft Internet Information Server (IIS). Esto limita su disponibilidad a servidores basados en Win32. Existe un par de proyectos en desarrollo que permiten que ASP corra en otros entornos y servidores Web: InstantASP (comercial) y ChiliSoft ASP (comercial). Se dice que ASP es un lenguaje más lento y pesado que PHP, y también menos estable. Algunas de las ventajas de ASP consisten en que debido a que usa principalmente VBScript, es relativamente simple tratar con el lenguaje si usted ya conoce cómo programar en Visual Basic. El soporte de ASP también se encuentra habilitado por defecto en el servidor IIS, facilitando su instalación y ejecución. Los componentes integrados en ASP son bastante limitados, de modo que si necesita usar características "avanzadas", como interactuar con servidores FTP, necesita comprar componentes adicionales.

Perl

Perl (**P**ractical **E**xtraction and **R**eport **L**anguage) es un lenguaje de programación desarrollado por Larry Wall (lwall at netlabs.com) inspirado en otras herramientas de UNIX como son: sed, grep, awk, c-shell, para la administración de tareas propias de sistemas UNIX.

No establece ninguna filosofía de programación concreta. No se puede decir que sea orientado a objetos, modular o estructurado aunque soporta directamente todos estos paradigmas; su punto fuerte son las labores de procesamiento de textos y archivos.

No es ni un compilador ni un intérprete, está en un punto intermedio, cuando mandamos a ejecutar un programa en Perl, se compila el código fuente a un código intermedio en memoria que se optimiza como si se fuera a elaborar un programa ejecutable pero es ejecutado por un motor, como si se tratase de un intérprete.

Lenguaje de programación basado en scripts portable a casi cualquier plataforma. Es muy utilizado para escribir CGI's. Uno de sus elementos más potentes son las expresiones regulares, que a partir de su versión en Perl han sido adoptadas por otros lenguajes y plataformas como .NET o Javascript.

PHP

Es un lenguaje de programación gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación. Su creador Rasmus Lerdorf ha recibido muchas contribuciones de otros desarrolladores debido a su política de código abierto [ALV04]. Actualmente PHP se encuentra en su versión PHP 5.

PHP puede ser utilizado en cualquiera de los principales sistemas operativos del mercado, incluyendo Linux, muchas variantes Unix (incluido HP-UX, Solaris y OpenBSD), Microsoft Windows, Mac OS X, RISC OS y probablemente alguno más. PHP soporta la mayoría de servidores Web de hoy en día, incluyendo Apache, Microsoft Internet Information Server, Personal Web Server, Netscape

e iPlanet, entre otros. PHP tiene módulos disponibles para la mayoría de los servidores, para aquellos otros que soporten el estándar CGI, PHP puede usarse como procesador CGI [PHP02].

En el desarrollo de Servicios Web, PHP no se ha quedado atrás, aunque no poseen una extensión que se encargue de ello. Basándose en SOAP, el protocolo más popular para la creación y consumo de Servicios Web su principal implementación es NuSOAP. La mejor opción es el uso del PHP 5.00 que desde sus inicios incorpora la extensión `php_soap` para el manejo del protocolo SOAP, además de ser orientado a objetos, ganando en eficiencia, estabilidad y facilidades de uso.

Por qué utilizar PHP y no otras opciones:

- PHP no soporta directamente punteros, como el C, de forma que no existen los problemas de depuración provocados por estos.
- Se pueden hacer grandes cosas con pocas líneas de código. Lo que hace que merezca la pena aprenderlo.
- Viene acompañado por una excelente biblioteca de funciones que permite realizar cualquier labor (acceso a base de datos, encriptación, envío de correo, gestión de un comercio electrónico, XML, creación de PDF ...)
- Está siendo utilizado con éxito en varios millones de sitios Web.
- Es multiplataforma, funciona en todas las plataformas que soporten apache.
- **Es software libre.** Se puede obtener en la Web y su código está disponible bajo la licencia GPL. Este es uno de los aspectos fundamentales para nuestra elección de este lenguaje.

Las cuatro grandes características: Velocidad, estabilidad, seguridad y simplicidad.

- **Velocidad:** No solo la velocidad de ejecución, la cual es importante, sino además no crear demoras en la máquina. Por esta razón no debe requerir

demasiados recursos de sistema. PHP se integra muy bien junto a otro software, especialmente bajo ambientes Unix, cuando se configura como módulo de Apache, está listo para ser utilizado.

- **Estabilidad:** La velocidad no sirve de mucho si el sistema se cae cada cierta cantidad de ejecuciones. Ninguna aplicación es 100% libre de bugs, pero teniendo de respaldo una increíble comunidad de programadores y usuarios es mucho más difícil para los bugs sobrevivir. PHP utiliza su propio sistema de administración de recursos y dispone de un sofisticado método de manejo de variables, conformando un sistema robusto y estable.
- **Seguridad:** El sistema debe poseer protecciones contra ataques. PHP provee diferentes niveles de seguridad, estos pueden ser configurados desde el archivo `.ini`. *¿Cual es?*
- **Simplicidad:** Se les debe permitir a los programadores generar código productivamente en el menor tiempo posible. Usuarios con experiencia en C y C++ podrán utilizar PHP rápidamente.

1.1.3.4 Gestores de Bases de Datos

PostgreSQL

Este gestor de bases de datos posee una estabilidad y confiabilidad legendaria nunca ha presentado caídas en varios años de operación de alta actividad. Están disponible 34 plataformas Unix en la última versión estable, existe una versión para Windows usando la plataforma Cygwin. Fue diseñado para ambientes de alto volumen. Escala muy bien al aumentar el número de CPUs y la cantidad de RAM. Soporta transacciones y desde la versión 7.0, claves ajenas con comprobaciones de integridad referencial. Tiene mejor soporte para vistas y procedimientos almacenados en el servidor, además tiene ciertas características orientadas a objetos.

MySQL

Según definición de sus autores: es un servidor de base de datos muy rápido, robusto, multitarea y multiusuario. Tiene enfoque relacional y soporta AnsiSQL. Es gratis y se puede bajar de www.mysql.com.

Soporta clientes en C, C++, Eiffel, Java, Perl, PHP, Python y Tcl. Trabaja en diferentes plataformas además de soportar múltiples idiomas Completo y optimizado uso del SQL y contiene un MyODBC.

En cuanto a seguridad confía en la propia del sistema a efectos de robo de las bases de datos, caída del sistema. Usa Listas de Control de Acceso para todas las conexiones, consultas y otras operaciones

En fin, es un servidor de Base de datos relacional, tiene interfases para desarrollo de clientes en diversos lenguajes, está disponible en diferentes plataformas, es gratis, se puede disponer del código fuente.

Razones para usar Postgres y no MySQL :

-

-PostgreSQL es una base de datos diseñada para ser de tipo empresarial.
-Algunas características de PostgreSQL aún no están disponibles o estables en MySQL:

- Triggers (5.1 rudimentarios)
- Vistas (5.0)
- Secuencias
- Herencia
- Cursores (5.0)
- Procedimientos almacenados (5.0 lenguaje único)

SQL Server

Microsoft SQL Server 7.0 constituye un lanzamiento determinante para los productos de bases de datos de Microsoft. SQL Server es el RDBMS de elección para una amplia gama de clientes corporativos y Proveedores Independientes de Software (ISVs) que construyen aplicaciones de negocios. Las necesidades y requerimientos de los clientes han llevado a la creación de innovaciones de producto significativas para facilitar la utilización, escalabilidad, confiabilidad y almacenamiento de datos.

Razones para usar Postgres y no SQL-Server :

- Estabilidad, Confiabilidad, SQL Server corre solo en MS-Windows.
- SQL Server no tiene Multiversiónado
- Adherencia a Estándares

usar una clase, accediendo a sus métodos y atributos. Para usar una función específica debemos instanciarla, es decir llamarla pasando por referencia valores o parámetros, luego esperamos un resultado, si es que existe, y en este caso ese es el rol que juega SOAP el de describir esta operación, llamar a la función, pasar los parámetros requeridos y describir cómo será la respuesta y cómo la recibiremos.

Algunas de las Ventajas de SOAP² son:

- **No está asociado con ningún lenguaje:** los desarrolladores involucrados en nuevos proyectos pueden elegir desarrollar con el último y mejor lenguaje de programación que exista pero los desarrolladores responsables de mantener antiguas aflicciones heredadas podrían no poder hacer esta elección sobre el lenguaje de programación que utilizan. SOAP no especifica una API, por lo que la implementación de la API se deja al lenguaje de programación, como en Java, y la plataforma, como Microsoft

como ya se ha mencionado SOAP no define un medio de transporte de los mensajes; los mensajes de SOAP se pueden asociar a los protocolos de transporte existentes como HTTP y SMTP.

- **Permite la interoperabilidad entre múltiples entornos:** SOAP se desarrollo sobre los estándares existentes de la industria, por lo que las aplicaciones que se ejecuten en plataformas con dicho estándares pueden comunicarse mediante mensaje SOAP con aplicaciones que se ejecuten en otras plataformas. Por ejemplo, una aplicación de escritorio que se ejecute en una PC puede comunicarse con una aplicación del back-end ejecutándose en un mainframe capaz de enviar y recibir XML sobre HTTP.

WSDL

Cubre las funcionalidades de descripción necesarias para las tecnologías de Servicios Web. Como evolución de los XML Schemas, actúa como mecanismo formal para la definición de los servicios, sus interfaces, puntos de acceso, etc. Uno de los aspectos fundamentales es facilitar la generación de código a partir de descripciones WSDL.

¿Qué es XML?

XML, es el estándar de Extensible Markup Language. XML no es más que un conjunto de reglas para definir etiquetas semánticas que nos organizan un documento en diferentes partes. XML es un metalenguaje que define la sintaxis utilizada para definir otros lenguajes de etiquetas estructurados.

A pesar de su sencillez aparente, XML está transformando completamente la creación y el uso de software. El Web revolucionó la comunicación entre usuarios y aplicaciones. XML está revolucionando la comunicación entre aplicaciones o, de forma más general, la comunicación entre equipos, pues ofrece un formato de datos universal que permite adaptar o transformar fácilmente la información.

¿Por qué XML es utilizado en los servicios Web?

- Es un estándar abierto es decir que es reconocido mundialmente ya que muchas compañías tecnológicas integran en su software compatibilidad con dicho lenguaje. Esto quiere decir que la gran mayoría de software de escritorio de sistema operativo, aplicaciones móviles permiten la compatibilidad con XML esto lo hace muy potente a la hora de permite la comunicación entre distintas plataformas de software y hardware (y si bien recordamos este es el sentido final de los Servicios Web).
- Simplicidad de sintaxis esto quiere decir que es muy fácil de escribir código en XML y la representación de los datos es casi entendible por cualquier ser humano. Esto lo hace muy flexible a la hora de querer repensar datos de cualquier especie, bastara con contar con cualquier editor de texto y

1.2 Objeto de estudio

1.2.1 Objetivos estratégicos de la organización

El objetivo estratégico de la organización es la preparación del país para la defensa y la lucha armada ante una posible agresión extranjera. Dentro de este objetivo general nuestro trabajo se concentra en el aumento del control, mediante la automatización, de todos los recursos materiales, financieros y humanos de la entidad.

1.2.2 Flujo actual de los procesos

Control de Seguridad

Actualmente el control de acceso lo realiza cada aplicación de manera independiente, teniendo en cuenta los requerimientos de seguridad planteados. Muchas de las aplicaciones trazan su política de seguridad en los principios básicos de crear usuarios de la aplicación con determinados permisos, brindando opciones de acceso y controlados por un administrador del sistema que en ocasiones es manejado por más de una persona. Otras definen permisos de acceso por una especialidad específica que se encarga de restringir el acceso a la información disponible solo a los usuarios que pertenecen a dicha especialidad, unas pocas aplican grupos de usuarios con permisos para el acceso a la información.

En el entorno Web se definen algunos de los permisos controlando el acceso mediante direcciones IP dentro de un rango permitido o un IP dado. Además de usar las potencialidades brindadas por la seguridad de NTFS, teniendo en cuenta, a la hora de publicar, la flexibilidad del acceso. Existen algunas aplicaciones en MS-DOS que no tienen implementado control de acceso.

La auditoría

El mayor porcentaje de las aplicaciones no presentan gestión de auditorías, las que lo tienen implementado lo hacen de manera independiente utilizando diferentes criterios y formas. Las realizan a nivel de gestor usando triggers, controlando acciones o eventos que ocurren sobre la Base de Datos, acciones sobre ficheros, páginas visitadas, etc.

1.2.3 Análisis crítico de la ejecución de los procesos

Como se describe anteriormente, los procesos de control de los accesos a las aplicaciones y la auditoría de los eventos que se generan por dichos accesos no se realizan de manera óptima. Inicialmente, la manera en que se aborda el proceso de control de la seguridad en la gran mayoría de las aplicaciones existentes hoy está alejado de la realidad de dicho proceso.

La manera de distribuir el papel de administrador de los sistemas, en muchas ocasiones manejable por más de una persona, entraña riesgos de seguridad, debido a que se pone en manos de muchos la estabilidad del funcionamiento y la integridad del sistema.

Los sistemas de grupos de usuario han demostrado su eficacia para conceder permisos a numerosos usuarios para controlar el acceso a la información, y aun más a la información que se muestra en aplicaciones Web, que generalmente están disponibles para un gran número de personas en la red. Esta herramienta es poco utilizada en las aplicaciones de la entidad.

1.3 Procesos objeto de automatización

1- El proceso de control de los accesos de los usuarios a través de:

- **Dominios de aplicación:** Especifica el nivel jerárquico al que pertenece el usuario dentro de la institución.

-
- **Especialidad:** El área a que pertenece dentro de la empresa, para un conjunto de usuarios con esta información se garantiza el acceso único a la información de su área.
 - **Grupos de usuarios:** Se dividen los tipos de usuarios según los privilegios que se otorgaran, así se asegura que los usuarios realicen solamente acciones inherentes a su grupo, que desempeña determinados roles.
 - **Usuarios Independientes:** Este tipo de usuario cumple roles específicos dentro de las aplicaciones. Puede no pertenecer a ningún grupo.
 - **Roles:** Papel que juega un usuario en la aplicación. Garantizan el control de los usuarios descrito en las especificaciones.
 - **Especificaciones de acceso para Roles:**
 - *A funcionalidades de sistemas:* Se especifican accesos a funcionalidades que tienen alcance a nivel de sistema.
 - *A funcionalidades de módulos:* Se especifican accesos a funcionalidades que tienen alcance solamente a nivel de módulos.

2- El proceso de archivar la información del estado del sistema y de los eventos realizados por los diferentes usuarios a través de:

- **Información sobre cantidad de usuarios que en un instante dado:**
 - *Usan el servicio:* Se especifica el número de usuarios que usan el servicio en un instante dado.
 - *Acceden a un sistema:* Se especifica el número de usuarios y el sistema a que acceden en un instante dado.
- **Información del usuario:** Se especifica todo referente al usuario.
- **Tipo de acción realizada:**

-
- *Acción de visualización sin generar salida:* Se basa en una consulta a datos que no genera ficheros de salida, solo se muestra la información en el navegador.
 - *Acción de visualización que genera salida:* La consulta genera un fichero de salida.
 - *acción de actualización:* Se basa en consultas que realizan las operaciones básicas sobre datos (insertar, modificar o eliminar).
 - *Acción de acceso a ficheros:* El usuario puede acceder a la información presente en un fichero. Para este caso se debe archivar la dirección de fichero accedido.
- **Contenido y resultado de la acción:** Se especifica el contenido de la consulta realizada y el resultado en caso de generar salida. Si la salida es un fichero se especifica su identificación.
 - **Momento de ocurrencia:** Especifica el instante exacto de ocurrencia de la acción.
 - **Información de los datos accedidos:** Especifica los datos accedidos.
 - **Información de los sistemas o módulos accedidos:** Especifica el nombre del sistema sobre el que se realiza la acción y/o el módulo.

1.4 Sistemas automatizados existentes vinculados al campo de acción

Web Services Enhancements 2.0 para Microsoft .NET permite integrar la autenticación de WS-Security basada en X.509 con características de la seguridad basada en funciones en Microsoft .NET Framework. Utiliza las características funcionales de .Net para desarrollar Servicios Web seguros. Permite crear y utilizar un administrador de tokens de seguridad personalizado con Web Services Enhancements 2.0 para Microsoft .NET con el fin de comprobar certificados X.509, asignarles funciones y rellenar información contextual mediante objetos principal e

identity personalizados. La ventaja de este enfoque basado en WS-Security frente a la seguridad clásica basada en HTTP es que el primero no confía en la seguridad o integridad a nivel de transporte sino que trabaja únicamente con el mensaje SOAP. Esto ofrece capacidades de seguridad de un extremo a otro sobre varios saltos y protocolos. Ofrece numerosas posibilidades pero no es libre, ya que funciona para .Net solamente.

En la entidad existe una aplicación llamada Control de Auditoría. Es un software elaborado en Delphi que sirve como herramienta para administrar la auditoría de los campos y tablas de la Base de Datos de SQL Server según se desee. Utiliza como técnica fundamental la generación de triggers. Dicha aplicación logra concentrar el control de las Base de datos a las cuales se tiene acceso. No es Multiplataforma y sirve solamente para SQL Server. No está implementada a manera de servicio, sino a manera de Aplicación Desktop y no es accesible a todos los sistemas.

En el mundo existen numerosos sistemas para el control de la seguridad, basados en los estándares de calidad e integridad de los datos, que brindan un conjunto de posibilidades para el control del acceso de los usuarios. Dichas aplicaciones no están implementadas al estilo de Servicios Web, tecnología que en este campo brinda infinitas posibilidades.

1.5 Fundamentación de los objetivos

En todo proceso de desarrollo de software es necesario realizar análisis y diseño, dado que son los procesos fundamentales de dicho desarrollo, donde se define qué es lo que se va a realizar y la manera de afrontarlo, es decir **el qué** y **el cómo** del desarrollo del software. En el Análisis se realiza un proceso de refinamiento de los requisitos capturados, permitiendo su estructuración y razonar sobre los aspectos internos del sistema. Además nos proporciona una estructura centrada en el mantenimiento, en aspectos tales como flexibilidad ante cambios y reutilización.

En el diseño modelamos el sistema y encontramos su forma (incluyendo la arquitectura)¹ y así aseguramos que se tengan en cuenta todos los requisitos. Describe la realización física de los casos de uso centrándose en como los requisitos funcionales y no funcionales tienen impacto en el sistema.

La realización de un software accesible a todas las aplicaciones sin importar la plataforma que usen ni bajo la cual estén implementadas es la solución óptima al problema planteado. Debido que se logra una intercomunicación eficiente entre las aplicaciones y se monitorea de manera centralizada el proceso de control de acceso de los usuarios a los sistemas.

En las aplicaciones diseñadas según un modelo de tres capas, el sistema es dividido en datos, lógica de negocio y presentación. La capa de datos representa el mecanismo por el cual se manipula y persiste la información. Consiste en un administrador de bases de datos relacional (RDBMS), y el esquema de datos propio de cada aplicación. Cuando hay varias aplicaciones presentes, los modelos de datos se complementan, evitando la duplicidad de información y aumentando las facilidades que brinda el sistema como un todo. En la capa de lógica de negocio se modela el comportamiento del sistema, basándose en los datos provistos por la capa de datos, y actualizándolos según sea necesario. Esta capa describe los distintos procesos de negocio que tienen lugar en las organizaciones. Finalmente, la capa de presentación contiene todos los elementos que constituyen la interfaz con el usuario. Esta capa incluye todo aquello con lo que el usuario puede interactuar, como por ejemplo las pantallas de las aplicaciones, el modelo de navegación del sistema y los adaptadores para cada modo de acceso (browser, teléfono celular, etc.). Los diseños "three tier" son ampliamente utilizados en el mercado, y a lo largo del tiempo han probado sus ventajas. Las aplicaciones en tres capas típicamente tienen mayor capacidad de crecimiento y son más sencillas de mantener, dada su naturaleza altamente modular.

Ventajas del uso de tres capas:

- La posibilidad de integrar aplicaciones que accedan a las mismas bases de datos de una forma sencilla.
- Separar las reglas de negocio de los interfaces especialmente en entornos multiplataforma permite que las reglas se cambien con un mínimo impacto sobre los usuarios de las aplicaciones.
- El uso de modelos three-tier aumenta increíblemente la flexibilidad a la hora de aplicar las posibilidades de la informática para aspectos específicos de la problemática del cliente.
- Es fácil construir nuevas aplicaciones desde los componentes instalados si las reglas del negocio están en unos servidores de aplicaciones más que en cada aplicación.
- Herramientas como el ODBC, que fuerza a las Bases de Datos a ser abstractas y transformarse en genéricos almacenes de datos, te permiten permanecer completamente flexibles a la hora de determinar donde guardar los datos independientemente de la aplicación que se use.

1.6 Tendencias y tecnologías actuales

Arquitectura y diseño de sistemas Web modernos

Las aplicaciones Web se han convertido en pocos años en complejos sistemas con interfaces de usuario cada vez más parecidas a las aplicaciones de escritorio, dando servicio a procesos de negocio de considerable envergadura y estableciéndose sobre ellas requisitos estrictos de accesibilidad y respuesta. Esto ha exigido reflexiones sobre la mejor arquitectura y las técnicas de diseño más adecuadas. En este artículo se pretende dar un breve repaso a la arquitectura de tales aplicaciones y a los patrones de diseño más aplicables. En los últimos años, la rápida expansión de Internet y del uso de intranets corporativas ha supuesto

una transformación en las necesidades de información de las organizaciones. En particular esto afecta a la necesidad de que:

1. La información sea accesible desde cualquier lugar dentro de la organización e incluso desde el exterior.
2. Esta información sea compartida entre todas las partes interesadas, de manera que todas tengan acceso a la información completa (o a aquella parte que les corresponda según su función) en cada momento.

Estas necesidades han provocado un movimiento creciente de cambio de las aplicaciones tradicionales de escritorio hacia las aplicaciones Web, que por su idiosincrasia, cumplen a la perfección con las necesidades mencionadas anteriormente. Por tanto, los sitios Web tradicionales que se limitaban a mostrar información se han convertido en aplicaciones capaces de una interacción más o menos sofisticada con el usuario. Inevitablemente, esto ha provocado un aumento progresivo de la complejidad de estos sistemas y, por ende, la necesidad de buscar opciones de diseño nuevas que permitan dar con la arquitectura óptima que facilite la construcción de los mismos.

El usuario interactúa con las aplicaciones Web a través del navegador. Como consecuencia de la actividad del usuario, se envían peticiones al servidor, donde se aloja la aplicación y que normalmente hace uso de una base de datos que almacena toda la información relacionada con la misma. El servidor procesa la petición y devuelve la respuesta al navegador que la presenta al usuario. Por tanto, el sistema se distribuye en tres componentes: el navegador, que presenta la interfaz al usuario; la aplicación, que se encarga de realizar las operaciones necesarias según las acciones llevadas a cabo por éste y la base de datos, donde la información relacionada con la aplicación se hace persistente. Esta distribución se conoce como el modelo o arquitectura de tres capas.

Software Libre

Un creciente número de organizaciones están adoptando el "software Free/Open Source" (Programas Gratuitos y/o de Código o Fuente Abierta) para todo, desde redes de computadoras hasta herramientas de productividad en oficinas.

Entre las ventajas del "software Open Source" figuran:

- Efectividad de costo
- seguridad, flexibilidad y adaptabilidad
- apoyo permanente y desarrollo de una comunidad de usuarios
- los programadores pueden resolver problemas y añadir nuevas funciones.

Una vez que un producto de software libre ha empezado a circular, rápidamente está disponible a un coste muy bajo o sin coste. Al mismo tiempo, su utilidad no decrece. Esto significa que el software libre se puede caracterizar como un bien público en lugar de un bien privado. Aunque realmente no lo es en ningún momento.

Puesto que el software libre permite el libre uso, modificación y redistribución, a menudo encuentra un hogar en los países del tercer mundo para los cuales el coste del software no libre es a veces prohibitivo. También es sencillo modificarlo localmente, lo que permite que sean posibles los esfuerzos de traducción a idiomas que no son necesariamente rentables comercialmente.

Tendencias hacia Servicios Web

Las aplicaciones Web actuales ya no son suficientes. El modelo actual de negocio electrónico no facilita la integración de las aplicaciones de Internet con el resto de software de las empresas. Si las compañías quieren extraer el máximo beneficio de Internet, los sitios Web deben evolucionar. Este es el contexto en el que surgen los Servicios Web.

El principal objetivo que se logra, es la interoperabilidad y la integración. Mediante los Servicios Web, las empresas pueden compartir servicios software con sus clientes y sus socios de negocio. Esto ayudará a las compañías a escalar sus negocios, reduciendo el coste en desarrollo y mantenimiento de software, y sacando los productos al mercado con mayor rapidez. La integración de aplicaciones hará posible obtener la información demandada en tiempo real, acelerando el proceso de toma de decisiones. La evolución de Internet hacia los Servicios Web, mejorará los resultados globales de las empresas, reduciendo sus gastos y guiándolas hacia una mejora progresiva de la calidad. La adopción de la tecnología de servicios Web por la industria es el primer paso hacia una economía global.

1.7 Conclusiones

Con el presente trabajo se da solución a un problema que tiene alta prioridad dentro de la institución debido a la gran importancia que tiene el servicio de control del acceso de los usuarios a las diferentes aplicaciones existentes y las que se implementan para el futuro. Debido al uso de tecnologías actuales y el enfoque que se tiene, constituye una herramienta muy importante y un aporte en el campo de las aplicaciones para el control de la seguridad y la auditoría.

2 Modelo del dominio

2.1 Introducción

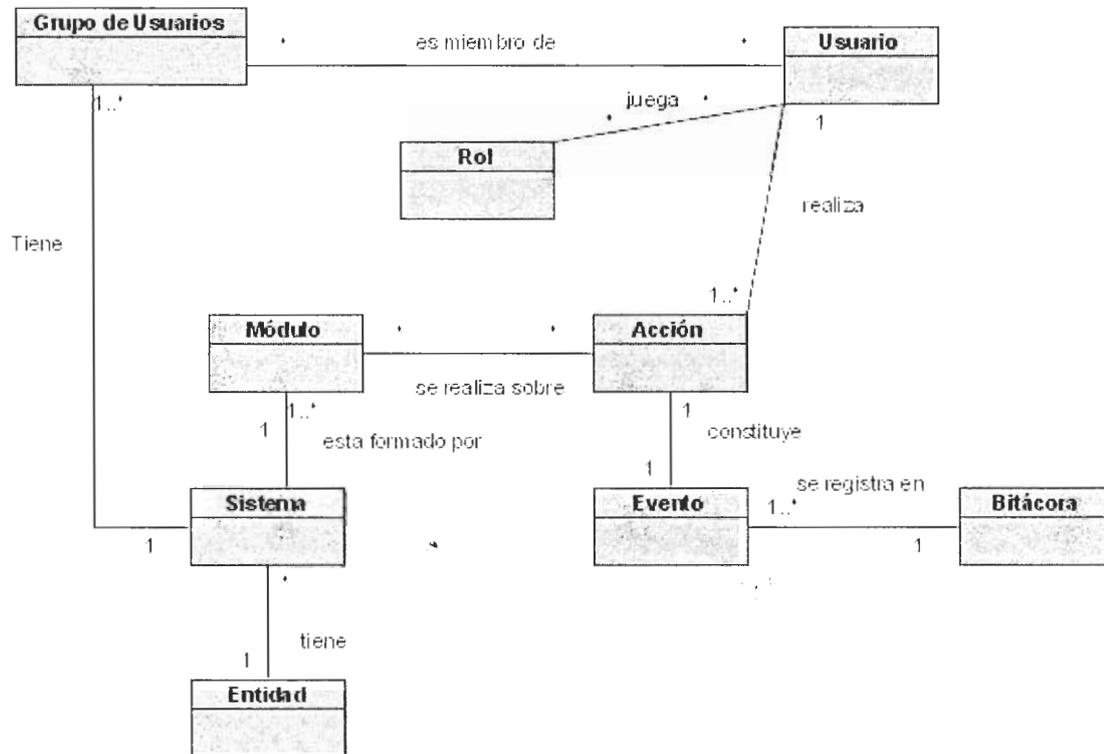
El proceso de control de la seguridad es un proceso de importancia primordial para el uso de las aplicaciones de software con que cuenta el MINFAR. Primeramente por el alto nivel de confidencialidad de la información que se maneja y además la sensibilidad de dicha información. Dicho proceso de control se realiza de manera inalterable en las aplicaciones y se maneja en cada una individualmente. Cuando se chequea la seguridad se tienen en cuenta tres elementos fundamentales: las personas, los sistemas y los datos. Las personas que representan a los usuarios que tendrán que interactuar en algún momento con los sistemas, dichos sistemas que serán afectados directamente por la acción de los usuarios y los datos, que pueden ser afectados de la manera más directa por la acción de los diferentes usuarios del sistema, que pueden alterar o eliminar dicha información. En este capítulo se hace referencia a los principales elementos del proceso de control de la seguridad, definiciones, relaciones etc.

2.2 Definición de las entidades y los conceptos principales

Concepto	Descripción
Entidad	Representa a las dependencias de la Institución, tales como regiones, sectores, unidades militares, sectores militares, etc.
Sistema	Aplicación de software en funcionamiento, accesible al personal de las dependencias de la entidad.
Módulo	Componente de la aplicación de software. Los sistemas están divididos en módulos o subsistemas.
Usuario	Representa a personas o sistemas de la entidad que interactúan con la aplicación.
Rol	Papel que juegan los usuarios dentro del sistema.
Grupo de Usuarios	Los grupos de usuarios se conforman por los usuarios que tienen permisos de acceso idéntico. Los grupos de usuario agrupan funcionalidades comunes.
Acción	Acciones que realizan los usuarios sobre los sistemas, ya sean de acceso al sistema y sus módulos, de acceso a datos, consultas, etc.
Evento	Se registra cada vez que un usuario realiza una acción sobre el sistema,

	es la manera de ver las acciones de la parte del sistema.
Bitácora	Es la estructura donde se almacenan los eventos que ocurren, así con la información específica de la hora, fecha, usuario, la acción realizada y otros datos de interés. Es la base de la auditoría.

2.3 Representación del modelo del dominio



2.4 Conclusiones

El modelo del dominio o modelo conceptual, nos permite de manera visual mostrar al usuario los principales conceptos que se manejan en el dominio del sistema en desarrollo. De manera que se puede comprender mejor cada uno de los conceptos fundamentales que se manejan en la entidad. A partir de este se comienza el desarrollo de la etapa próxima del proceso de desarrollo del software.

3 Requerimientos

3.1 Introducción

La captura de requisitos para el sistema constituye uno de los procesos más importantes en la etapa de desarrollo de software pues es donde se definen las funcionalidades del sistema en cuestión y se analizan detalladamente los requerimientos de software, hardware, etc, para el correcto funcionamiento de la misma.

3.2 Actores del sistema a automatizar

Nombre del actor	Descripción
Administrador	Es el administrador del sistema, que maneja toda la información de configuración de los sistemas y usuarios.
Sistema Externo	Representa los sistemas y módulos externos que interactúan con el sistema de seguridad.

Tabla 1. Definición de actores del sistema a automatizar

3.3 Definición de los requisitos funcionales y no funcionales

3.3.1 Requisitos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable.

Apariencia o interfaz externa:

- Diseño sencillo, con pocas entradas, permitiendo que no sea necesario mucho entrenamiento para utilizar el sistema.
- Empleo de los colores: verde, gris, blanco y azul principalmente, que son los definidos en los estándares del proyecto.

Usabilidad:

- El sistema podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de la computadora y de contabilidad financiera.
- El software tendrá siempre la posibilidad de ayuda disponible para cualquier tipo de usuario, lo que le permitirá un avance considerable en la explotación de la aplicación en todas sus funcionalidades.

Rendimiento:

- Tiempos de respuestas rápidos al igual que la velocidad de procesamiento de la información, no mayor a los 5 segundos en las actualizaciones y no mayor de 20 para las recuperaciones.

Soporte:

- Se requiere un servidor de bases de datos con las siguientes características:
- Soporte para grandes volúmenes de datos y velocidad de procesamiento.
- Tiempo de respuesta rápido en accesos concurrentes.
- Versión de PHP 5.0.
- Por parte del cliente se requiere un navegador capaz de interpretar JavaScript.

Portabilidad:

- Necesidad de que el sistema sea multiplataformas.

Seguridad:

- Autenticación (contraseña de acceso)
- Garantizar que las funcionalidades del sistema se muestren de acuerdo al nivel de usuario que este activo.
- Protección contra acciones no autorizadas o que puedan afectar la integridad de los datos.

-
- Verificación sobre acciones irreversibles (eliminaciones).

Confiabilidad:

- La herramienta de implementación a utilizar tiene soporte para recuperación ante fallos y errores.

Funcionalidad:

- Mínima cantidad de páginas para ejecutar todas las funciones posibles (preferentemente que estén relacionadas).

Implantación

- Entregar toda la documentación asociada al proyecto.
- Organizar el adiestramiento de los usuarios.

Software:

En secciones anteriores se ha mencionado que la construcción de nuestra aplicación funcionará bajo los conceptos de arquitectura cliente/servidor. Por tanto el servidor del usuario final debe tener como requerimientos mínimos de software:

- Una computadora personal con plataforma del sistema operativo Windows Advanced Server 2000 o superior; o Linux.
- Apache 2.0 o superior como servidor Web, con módulo PHP 5 disponible y debe estar configurado con la extensión pgsql incluida.
- PostgreSQL como Sistema Gestor de Base de Datos.
- Y la máquina cliente del usuario debe tener como requerimiento mínimo:
- El navegador Mozilla FireFox.

Hardware

Partiendo del mismo supuesto que los requerimientos de software, nuestro modelo ideal (cliente/servidor), para los requerimientos mínimos de hardware, el usuario final debe tener un servidor con las siguientes características:

- Tarjeta de red.
- 128 Mb. de RAM o superior.
- 40 Gb. de disco duro o superior.
- Pentium II a 133 MHz de velocidad en su procesador o más.
- Una computadora que sirva de cliente:
- Pentium a 200 MHz. de velocidad de procesamiento o superior.
- 32 Mb. de memoria RAM superior.
- Tarjeta de red.

3.3.2 Requisitos funcionales

Los requerimientos funcionales como bien nos sugiere, son capacidades o condiciones que el sistema debe cumplir, por lo que en nuestro caso este sistema que se está analizando debe tener la propiedad de satisfacer los requisitos siguientes:

R1. Registrar usuarios.

- Chequear que el usuario que solicita realizar esta operación está realmente autorizado.
- Permitir que el administrador agregue usuarios.
- Permitir que el administrador modifique los usuarios.
- Permitir que el administrador elimine los usuarios.
- Guardar la información introducida o modificada.

R2. Registrar los sistemas.

- Chequear que el usuario que solicita realizar esta operación está realmente autorizado.
- Permitir que el administrador agregue sistemas.
- Permitir que el administrador modifique los sistemas.
- Permitir que el administrador elimine los sistemas.
- Guardar la información introducida o modificada.

R3. Registrar los módulos.

- Chequear que el usuario que solicita realizar esta operación está realmente autorizado.
- Permitir que el administrador agregue módulos.
- Permitir que el administrador modifique los módulos.
- Permitir que el administrador elimine los módulos.
- Guardar la información introducida o modificada.

R4. Registrar Roles a jugar por los grupos de Usuarios en los módulos.

- Chequear que el usuario que solicita realizar esta operación está realmente autorizado.
- Permitir que el administrador agregue Roles por grupos de usuarios en los módulos.
- Permitir que el administrador modifique Roles por grupos de usuarios en los módulos.
- Permitir que el administrador elimine Roles por grupos de usuarios en los módulos.
- Guardar la información introducida o modificada.

R5. Registrar Grupos de Usuarios.

- Chequear que el usuario que solicita realizar esta operación está realmente autorizado.
- Permitir que el administrador agregue grupos de usuarios.
- Permitir que el administrador modifique grupos de usuarios.
- Permitir que el administrador elimine grupos de usuarios.
- Guardar la información introducida o modificada.

R6. Registrar los eventos a auditar.

- Chequear que el usuario que solicita realizar esta operación está realmente autorizado.
- Permitir que el administrador agregue eventos a auditar.
- Permitir que el administrador modifique eventos a auditar.
- Permitir que el administrador elimine eventos a auditar.

-
- Guardar la información de los eventos cada vez que se registren.

R7. Registrar Opciones.

- Chequear que el usuario que solicita realizar esta operación está realmente autorizado.
- Permitir que el administrador agregue opciones.
- Permitir que el administrador modifique opciones.
- Permitir que el administrador elimine opciones.
- Asociar las opciones a los roles.
- Guardar la información introducida o modificada.

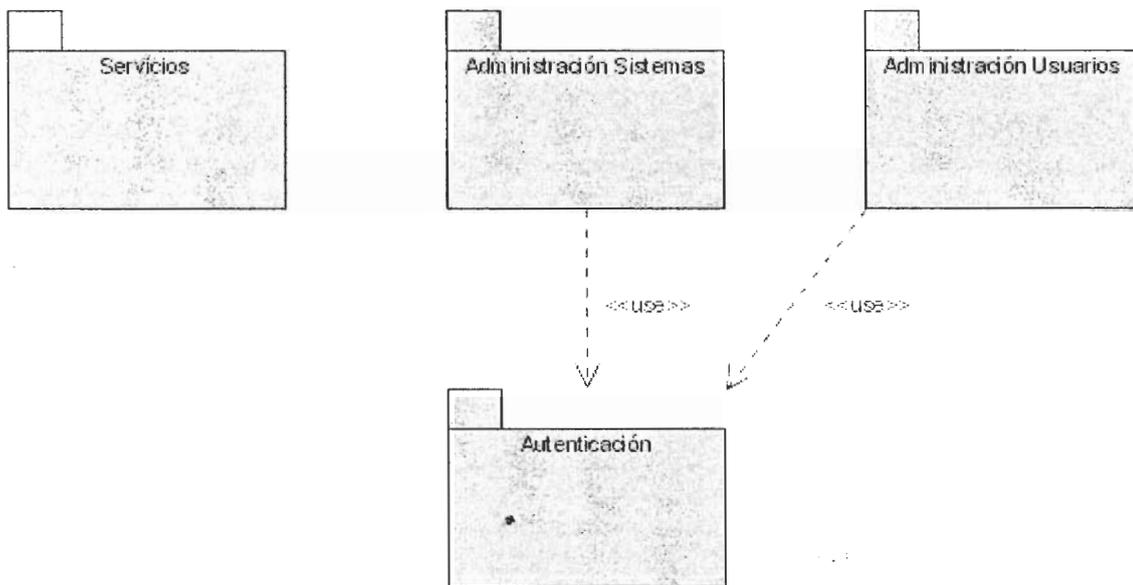
R8. Auditar acceso a los sistemas y módulos.**R9. Auditar acceso a cada opción en los sistemas.****R10. Servicio de autenticación accesible para todos los sistemas independiente de su plataforma.**

- Establecer una comunicación entre los sistemas y el servicio para validar los usuarios.
- Permitir crear, emitir, chequear y eliminar certificados de seguridad.
- Permitir que se puedan obtener los sistemas a los cuales el usuario que se autentifica tiene acceso.

R11. Servicio de auditoría.

- Establecer una comunicación entre los sistemas y el servicio para registrar los eventos que los sistemas generen.
- Permitir que se consulten los datos que se encuentran en la bitácora de la auditoría.

3.4 Paquetes y sus relaciones



3.5 Diagrama de casos de uso del sistema a automatizar

A continuación se muestran los diagramas de Casos de Uso del sistema por cada uno de los paquetes.

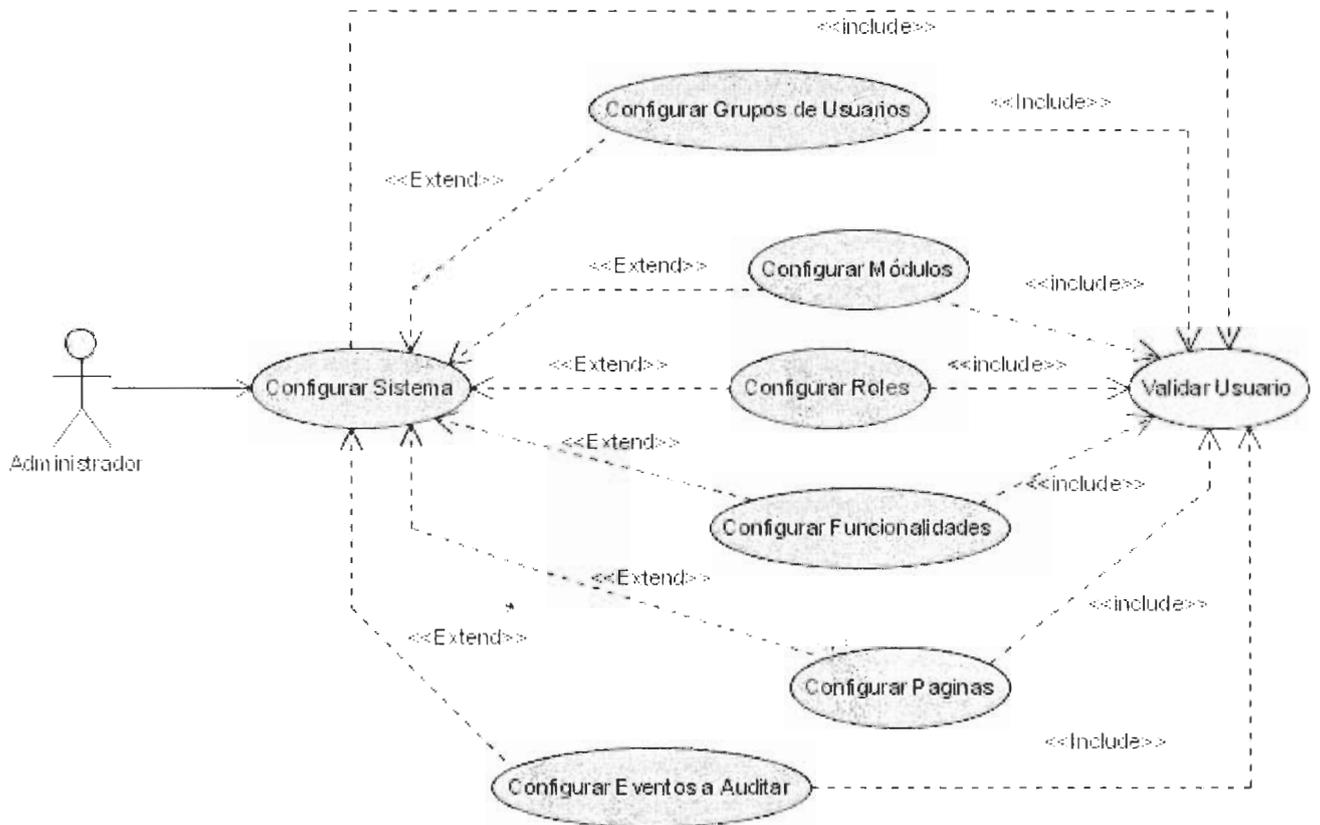


Figura 1. Diagrama de los casos de uso del paquete: *Administrar Sistemas*

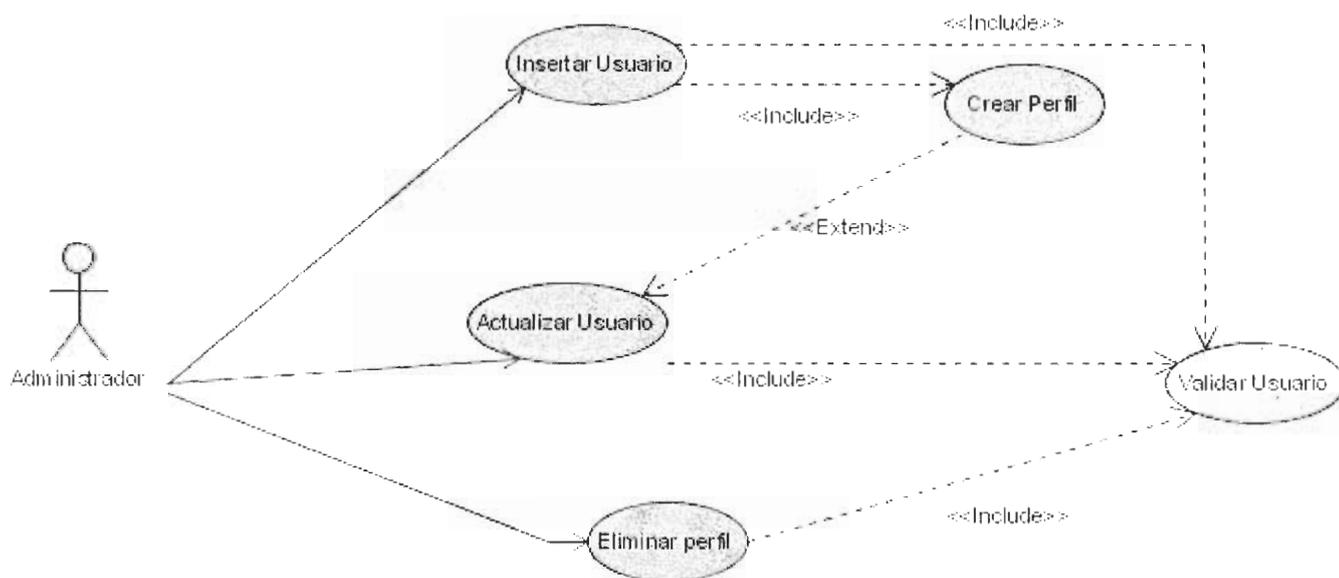


Figura 2. Diagrama de los casos de uso del paquete: *Administrar Usuarios*

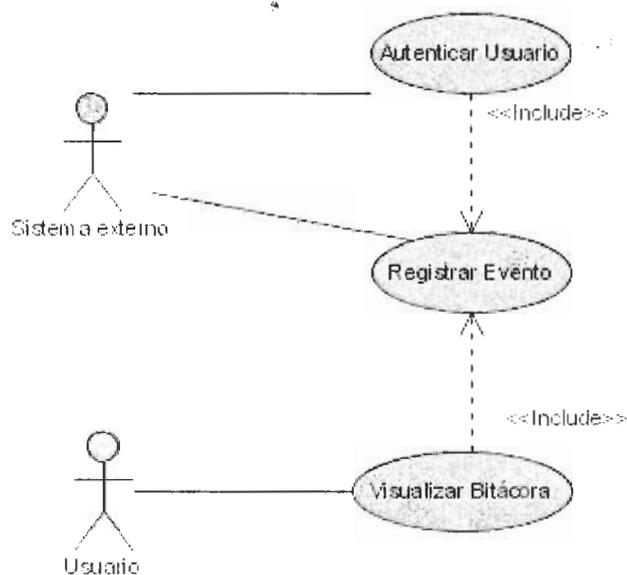


Figura 3. Diagrama de los casos de uso del paquete: *Servicios*

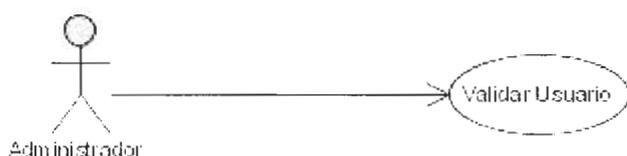


Figura 4. Diagrama de los casos de uso del paquete: *Autenticación*

En los diagramas anteriores se destaca el caso de uso Validar Usuario porque está incluido en los diagramas de Administración de usuarios y de Administración de Sistemas y pertenece al paquete de Autenticación. Se hace la referencia como un caso de uso externo a ese paquete.

3.6 Descripción de los casos de uso

A continuación se muestra la descripción de los casos de uso del sistema:

Nombre del caso de uso	Configurar Sistema
Actores	Administrador del Sistema
Resumen	El caso de uso se inicia cuando se desea incorporar un nuevo sistema y el administrador comienza a procesar la información. Termina cuando se ha almacenado toda la información referente al sistema.

Precondiciones	El Administrador debe estar validado, caso de uso: Validar usuario, que tiene una relación de inclusión.
Poscondiciones	Se almacena en BD toda la información del Sistema registrado.
Requisitos especiales	-

Nombre del caso de uso	Configurar Módulos
Actores	Administrador
Resumen	El caso de uso se inicia cuando se desea configurar un módulo, y termina cuando se ha actualizado dicha información.
Precondiciones	El Administrador debe estar validado, caso de uso: Validar usuario, que tiene una relación de inclusión.
Poscondiciones	Queda actualizada la información del módulo.
Requisitos especiales	

Nombre del caso de uso	Configurar Funcionalidades y menú
Actores	Administrador
Resumen	El caso de uso se inicia cuando se desean insertar, modificar o eliminar funcionalidades para un modulo determinado. Termina cuando se han insertado, modificado o eliminado dichas funcionalidades y la información está actualizada.
Precondiciones	El Administrador debe estar validado, caso de uso: Validar usuario, que tiene una relación de inclusión.
Poscondiciones	Queda actualizada la información de las funcionalidades de un módulo.
Requisitos especiales	-

Nombre del caso de uso	Configurar Roles
Actores	Administrador
Resumen	El Caso de Uso se inicia cuando el Administrador desea configurar un rol, el sistema brinda la posibilidad de insertar, modificar o eliminar, según lo que se seleccione el sistema procede mostrando las <i>interfaces</i> para la interacción con el administrador, el caso de uso termina cuando se actualizan las tablas de la base de datos.

Precondiciones	El Administrador debe estar validado, caso de uso: Validar usuario, que tiene una relación de inclusión.
Poscondiciones	Se actualiza en la BD la información referente a Roles.
Requisitos especiales	

Nombre del caso de uso	Configurar Grupos de Usuarios.
Actores	Administrador del sistema
Resumen	El caso de uso se inicia cuando el actor decide insertar, modificar o eliminar grupos de usuarios. En caso de insertar el sistema solicita la información para el nuevo grupo. Si se decide modificar el sistema permite seleccionar el grupo y muestra las características con posibilidad de ser modificadas. Si se opta por eliminar el sistema permite seleccionar el grupo. El caso de uso termina cuando el sistema inserta, modifica o elimina el grupo y se actualiza la tabla de los grupos de usuarios. Luego se pasa al caso de uso Registrar Evento.
Precondiciones	El Administrador debe estar validado, caso de uso: Validar usuario, que tiene una relación de inclusión.
Poscondiciones	Se inserta, modifica o elimina en la BD un grupo de usuarios.
Requisitos especiales	

Nombre del caso de uso	Insertar usuario.
Actores	Administrador del sistema(Inicia)
Resumen	El caso de uso se inicia cuando el actor decide insertar usuario , el sistema solicita a través de la interfaz los datos necesarios para ello. El caso de uso finaliza cuando ya el usuario ha sido insertado. Luego se pasa al caso de uso Registrar Evento y Crear Perfil.
Precondiciones	El Administrador debe estar validado, caso de uso: Validar usuario, que tiene una relación de inclusión.
Poscondiciones	Se almacena en BD toda la información del usuario insertado.
Requisitos especiales	

Nombre del caso de uso	Configurar Páginas
Actores	Administrador
Resumen	El caso de uso se inicia cuando el administrador desea incorporar toda la información de las páginas que conforman el sitio para el control de acceso a ellas.
Precondiciones	El Administrador debe estar validado, caso de uso: Validar usuario, que tiene una relación de inclusión.
Poscondiciones	Se tendrá almacenada en la BD toda la información de las páginas que tiene el sitio.
Requisitos especiales	

Nombre del caso de uso	Actualizar usuario
Actores	Administrador del sistema(Inicia)
Resumen	El caso de uso se inicia cuando el administrador decide modificar o eliminar un usuario existente. Si se va a modificar el sistema muestra las características del usuario con posibilidad de ser modificadas, luego se actualizará el usuario. Si se va a eliminar se elimina el usuario seleccionado, el caso de uso termina cuando se actualiza la tabla de los usuarios. Luego se pasa a los casos de uso Registrar Evento y Crear Perfil. Si el usuario no es eliminado
Precondiciones	El Administrador debe estar validado, caso de uso: Validar usuario, que tiene una relación de inclusión.
Poscondiciones	El administrador modifica o elimina usuarios.
Requisitos especiales	

Nombre del caso de uso	Crear perfil
Actores	Administrador del Sistema
Resumen	El caso de uso se inicia cuando el administrador inserta un nuevo usuario, a su vez se crea su perfil, esto va a ser un xml se utiliza para mandar toda la información del usuario a los otros módulos de servicios, se guarda como fichero donde se almacena la información de las funcionalidades de los módulos de servicio a las cuales el usuario puede acceder, en dependencia de su rol,

	finalmente se crea el perfil y termina el caso de uso.
Precondiciones	El usuario debe haberse insertado y jugar un rol.
Poscondiciones	Se almacena en un xml toda la información del perfil de usuario creado.
Requisitos especiales	

Nombre del caso de uso	Eliminar perfil.
Actores	Administrador del Sistema
Resumen	El caso de uso se inicia cuando el administrador decide eliminar un perfil de un determinado usuario el sistema muestra una ventana con todos los perfiles que hay creados, el administrador selecciona el solicitado y el sistema finalmente lo elimina.
Precondiciones	-
Poscondiciones	Se elimina el perfil de usuario.
Requisitos especiales	

Nombre del caso de uso	Configurar Eventos Auditables
Actores	Administrador
Resumen	El caso de uso se inicia cuando el administrador desea configurar los eventos auditables de los sistemas, ya sea agregar, modificar o eliminar existentes. El caso de uso termina cuando se ha actualizado toda la información referente a los eventos auditables.
Precondiciones	El Administrador debe estar validado, caso de uso: Validar usuario, que tiene una relación de inclusión.
Poscondiciones	Se actualiza toda la información referente a los eventos auditables en la BD.
Requisitos especiales	

Nombre del caso de uso	Autenticar Usuario
Actores	Sistema externo (Inicia)
Resumen	El caso de uso se inicia cuando el actor invoca al servicio, en la llamada viaja el nombre de usuario y el pass y el sistema, comprueba que este existe, le crea un certificado y en la respuesta le envía un arreglo con los datos del usuario (menos el pass), incluyendo el

	certificado y dos arreglos con los números de las páginas y las funcionalidades que accede respectivamente.
Precondiciones	Se desea acceder a un sistema.
Poscondiciones	Se autentica un usuario.
Requisitos especiales	

Nombre del caso de uso	Registrar Evento
Actores	Sistema externo
Resumen	El caso de uso se inicia cuando el generador de sucesos solicita registrar uno de los eventos configurados para ser auditados. El sistema guarda la información referente al usuario, el evento, la fecha y la hora en que se accedió. El caso de uso termina cuando se actualiza la tabla de Eventos.
Precondiciones	Un actor interactúa con un sistema generando un evento auditable.
Poscondiciones	Se registra un Evento auditable.
Requisitos especiales	-

Nombre del caso de uso	Validar Usuario
Actores	Administrador del Sistema
Resumen	El caso de uso se inicia cuando se accede al sistema para verificar si el usuario es válido y puede entrar al sistema. Termina cuando es autorizado o denegado el acceso al sistema.
Precondiciones	-
Poscondiciones	Se autoriza o deniega el acceso al sistema.
Requisitos especiales	

Nombre del caso de uso	Visualizar Registro
Actores	Sistema Externo
Resumen	El caso de uso se inicia desde un sistema externo se realiza una solicitud de visualización de registro de eventos y termina cuando se devuelve el resultado de la consulta realizada o si la solicitud no está autorizada.
Precondiciones	-
Poscondiciones	Se Visualiza una consulta al registro de eventos
Requisitos especiales	Debe ser una solicitud autorizada.

3.7 Descripción Expandida de los Casos de Uso Principales

Caso de uso:	Configurar Sistema
Actores:	Administrador del Sistema
Propósito: Almacenar toda la información referente a un sistema y registrarlo	
Resumen: El caso de uso se inicia cuando se desea incorporar un nuevo sistema y el administrador comienza a procesar la información. Termina cuando se ha almacenado toda la información referente al sistema y se han realizado los casos de uso: Configurar Módulos, Configurar Roles y configurar funcionalidades.	
Precondiciones:	El Administrador debe estar validado, caso de uso: Validar usuario, que tiene una relación de inclusión.
Referencias:	R2
Poscondiciones:	Se almacena en BD toda la información del Sistema registrado.
Requerimientos Especiales:	-

Interfaces:

Pantalla Opciones Administrativas

Elija opción:

[Configurar Sistema](#)

[Configurar Módulos](#)

[Configurar Funcionalidades y menú](#)

[Configurar Roles](#)

[Configurar Páginas](#)

Para Usuarios:

[Insertar Usuario](#)

[Actualizar Usuario](#)

Para Grupos de Usuarios

[Insertar Grupo](#)

[Modificar Grupo](#)

[Eliminar Grupo](#)

Para Eventos a Auditar

[Insertar Evento a Auditar](#)

[Modificar Evento a Auditar](#)

[Eliminar Evento a Auditar](#)

Pantalla Configurar Sistema

Configurar Sistema

Agregar Sistema A

Nombre

Modificar Sistema B

Nombre

Eliminar Sistema C

D

F

Curso normal de eventos para el caso de uso:	
Acción del actor	Respuesta del sistema
1-Desea Configurar Sistema 3-Selecciona Configurar Sistema 5-Selecciona: <ul style="list-style-type: none"> • Agregar Sistema • Modificar Sistema • Eliminar Sistema 	2-Muestra la interfaz de Opciones Administrativas 4-Muestra interfaz de Configurar Sistema 6-Si el usuario selecciona: <ul style="list-style-type: none"> • Agregar Sistema, ver sección "Agregar Sistema" • Modificar Sistema, ver sección "Modificar Sistema" • Eliminar Sistema, ver sección "Eliminar Sistema"
Sección: Agregar Sistema	
Acción de Actor	Respuesta del Sistema
1-Especifica el nombre del sistema(A) a agregar y presiona el botón (E).	2-Verifica la integridad de los datos brindados por el usuario. 3-Procesa los datos. 4-Muestra mensaje de operación realizada.
Sección: Modificar Sistema	
Acción de Actor	Respuesta del Sistema
2-Selecciona el sistema que desea modificar y especifica el nuevo nombre (C), luego presiona el botón procesar (E).	1-Busca el la BD los sistemas existentes. 3-Verifica la integridad de los datos brindados por el usuario. 4-Procesa los datos. 5-Muestra mensaje de operación realizada.
Sección: Eliminar Sistema	
Acción de Actor	Respuesta del Sistema
2-Selecciona el Sistema a eliminar (D) y presiona el botón procesar (E).	1-Busca el la BD los sistemas existentes.

	<p>3-Verifica la integridad de los datos brindados por el usuario. 4-Procesa los datos. 5-Muestra mensaje de operación</p>
Cursos alternos:	
Si Ocurre un Error en los procesos de inserción, modificación o eliminación de Sistema Muestra mensaje de error, y brinda la posibilidad de corregir datos.	

Caso de uso:	Configurar Módulos
Actores:	Administrador
Propósito:	Realizar la configuración de los módulos que integran un sistema.
Resumen:	El caso de uso se inicia cuando se desea configurar un módulo, y termina cuando se ha actualizado dicha información.
Precondiciones:	El Administrador debe estar validado, caso de uso: Validar usuario, que tiene una relación de inclusión.
Referencias:	R3
Poscondiciones:	Queda actualizada la información del módulo.
Requerimientos Especiales:	-

Interfases:**Interfaz Opciones Administrativas.****Elija opción:**Configurar SistemaConfigurar MódulosConfigurar Funcionalidades y menúConfigurar RolesConfigurar Páginas**Para Usuarios:**Insertar UsuarioActualizar Usuario**Para Grupos de Usuarios**Insertar GrupoModificar GrupoEliminar Grupo**Para Eventos a Auditar**Insertar Evento a AuditarModificar Evento a AuditarEliminar Evento a Auditar

Interfaz Configurar Módulos.

Configurar Módulos

Insertar Nuevo Módulo

Nombre: A Sistema: B

Modificar Módulo: C

Módulo: E Tema: D

Nombre:

Eliminar Módulo F

Módulo: G Sistema:

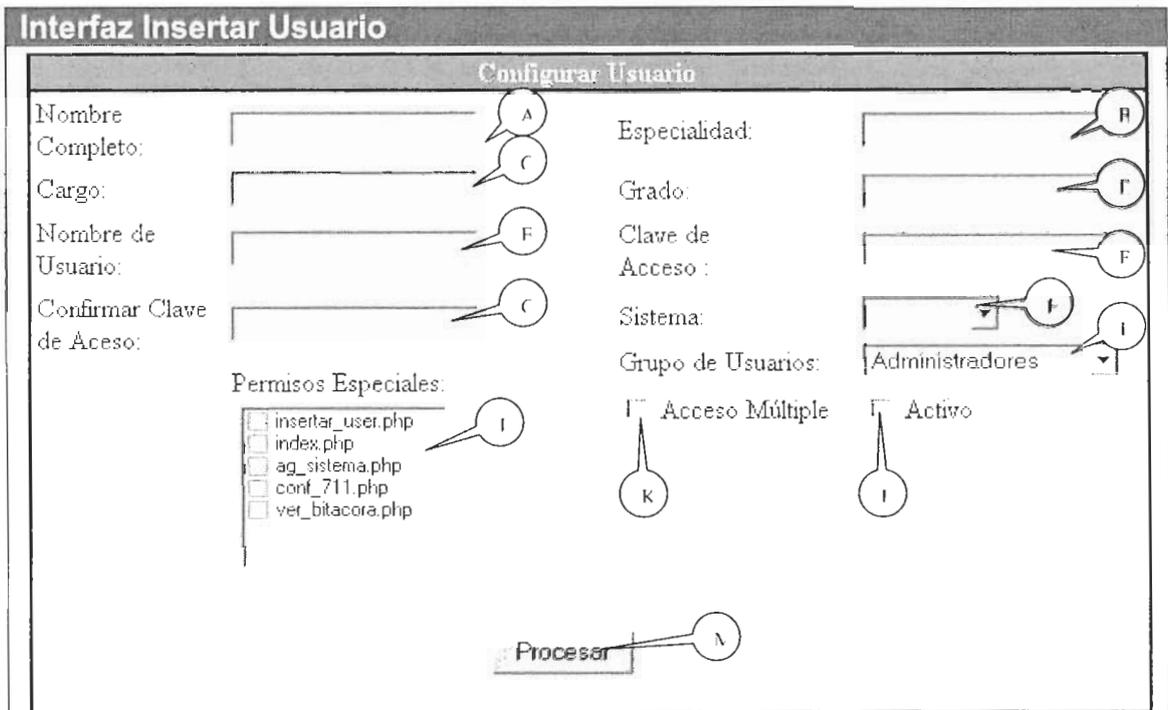
H

Curso normal de eventos:	
Acción del actor	Respuesta del sistema
1-Desea Configurar Modulo	2-Muestra la interfaz de Opciones Administrativas
3-Selecciona Configurar módulo	4-Muestra interfaz de Configurar Módulos, consultando la BD y seleccionando los sistemas y módulos existentes con su información.
5-El usuario Selecciona: <ul style="list-style-type: none"> • Agregar nuevo módulo. • Modificar Módulo. • Eliminar Módulo. 	6-Decide si: <ul style="list-style-type: none"> • El usuario seleccionó Agregar nuevo módulo, ver sección: "Agregar nuevo módulo" • El usuario seleccionó Modificar Módulo, ver sección: "Modificar Módulo" El usuario seleccionó Eliminar Módulo, ver sección: "Eliminar Módulo"

Sección Agregar Módulo	
1- Especifica el nombre del módulo a insertar (A), selecciona el sistema (B) a que pertenece y presiona al botón (H).	2- Verifica la integridad de la información. 3-Procesa la información. 4-Muestra mensaje de operación realizada.
Sección: Actualizar Módulo	
Acción del actor	Respuesta del sistema
1- Selecciona el Sistema (D), luego el módulo (C) que desea modificar y especifica el nuevo nombre (E) que va a tener el módulo. Presiona H.	2-Verifica la integridad de la información a procesar. 3-Procesa la información. 4-Muestra mensaje de operación realizada.
Sección Eliminar módulo	
Acción del actor	Respuesta del sistema
1-Especifica el sistema (G), el módulo (F) que desea eliminar y presiona el botón (J).	2- Verifica la existencia del módulo en el sistema y la información. 3-Procesa la información. 4-Muestra mensaje de operación realizada.
Cursos Alternos:	
<p>Sección: Agregar Módulo Línea 2: Si existe algún problema en la información, muestra mensaje de error. Luego devuelve el control a la línea 1.</p> <p>Sección: Actualizar Módulo Línea 2: Si existe algún problema en la información, muestra mensaje de error. Luego devuelve el control a la línea 1.</p> <p>Sección: Eliminar Módulo Línea 2: Si existe algún problema en la información, muestra mensaje de error. Luego devuelve el control a la línea 1.</p>	

Caso de uso:	Insertar usuario.
Actores:	Administrador del sistema(Inicia)
Propósito: Insertar un nuevo usuario con los datos que se soliciten.	
Resumen: El caso de uso se inicia cuando el actor decide insertar usuario , el sistema solicita a través de la interfaz los datos necesarios para ello. El caso de uso finaliza cuando ya el usuario ha sido insertado. Luego se pasa al caso de uso Registrar Evento y Crear Perfil.	
Precondiciones:	El Administrador debe estar validado, caso de uso: Validar usuario, que tiene una relación de inclusión.
Referencias:	R1
Poscondiciones:	Se almacena en BD toda la información del usuario insertado.
Requerimientos Especiales:	El administrador debe haber completado los datos que se solicitan para poder agregarlo.

Interfaz Opciones Administrativas**Elija opción:**Configurar SistemaConfigurar MódulosConfigurar Funcionalidades y menúConfigurar RolesConfigurar Páginas**Para Usuarios:**Insertar UsuarioActualizar Usuario**Para Grupos de Usuarios**Insertar GrupoModificar GrupoEliminar Grupo**Para Eventos a Auditar**Insertar Evento a AuditarModificar Evento a AuditarEliminar Evento a Auditar



Curso normal de eventos:

Acción del actor	Respuesta del sistema
<p>1. Desea Insertar Usuario</p> <p>3. Selecciona Opción Insertar usuario.</p> <p>5. Especifica nombre completo de usuario, especialidad, cargo, grado, el nombre de usuario, contraseña de acceso y verificación de contraseña, Sistema, Grupo de Usuarios al que pertenece, si va a tener acceso múltiple o no, si está activo, y en caso de ser así, otorgar permisos especiales.</p>	<p>2. Muestra interfaz de opciones Administrativas.</p> <p>4. Muestra Interfaz de insertar Usuario consultando la BD y seleccionando los sistemas, grupos, y páginas.</p> <p>6. Verifica la información e inserta datos en la BD.</p> <p>7. Muestra mensaje de operación realizada.</p>

Cursos alternos:*Línea 6*

Si el sistema verifica que falta algún dato en el envío muestra mensaje de error. El usuario acepta en A y regresa el control a la línea 5.

*Línea 6*

Si el sistema verifica que el Usuario existe, genera mensaje de error, el administrador Acepta en **A** y el sistema devuelve el control a la *línea 5* del flujo normal de eventos.



Caso de uso:		Validar Usuario
Actores:		Administrador del Sistema
Propósito: Validar la entrada al sistema.		
Resumen: El caso de uso se inicia cuando se accede al sistema para verificar si el usuario es válido y puede entrar al sistema. Termina cuando es autorizado o denegado el acceso al sistema.		
Precondiciones:		
Referencias:		R10
Poscondiciones:		Se autentica el usuario.
Interfaz Formulario de autenticación		
<p>The diagram shows a login form with three elements: a text input field labeled 'Usuario' with a callout bubble 'A', a text input field labeled 'Contraseña' with a callout bubble 'B', and a button labeled 'Aceptar' with a callout bubble 'C'.</p>		
Curso normal de eventos para el caso de uso:		
Acción del actor	Respuesta del sistema	
<ol style="list-style-type: none"> 1. Desea entrar al sistema 3. Especifica el usuario (A) y clave de acceso (B). Luego oprime el botón C. 	<ol style="list-style-type: none"> 2. Muestra Interfaz de autenticación. 4. Verifica en la BD los datos del Usuario. 5. Si los datos son correctos permite el acceso al sistema, en caso contrario devuelve el control a la línea 2. 	

Caso de uso:		Autenticar Usuario	
Actores:		Sistema externo (Inicia)	
Propósito: Validar la autenticidad de un usuario al acceder a uno de los sistemas.			
Resumen: El caso de uso se inicia cuando el actor invoca al servicio, en la llamada viaja el nombre de usuario y el pass y el sistema, comprueba que este existe, le crea un certificado y en la respuesta le envía un arreglo con los datos del usuario (menos el pass), incluyendo el certificado y dos arreglos con los números de las páginas y las funcionalidades que accede respectivamente.			
Precondiciones:		Se desea acceder a un sistema.	
Referencias:		R10	
Poscondiciones:		Se autentica un usuario.	
Curso normal de eventos para el caso de uso:			
Acción del actor		Respuesta del sistema	
<ul style="list-style-type: none"> El actor invoca al servicio, en esta llamada viaja el nombre de usuario y la clave y el sistema. 		<ul style="list-style-type: none"> Comprueba que existe. Crea un certificado. <p>4. Envía un arreglo con los datos del usuario (menos la clave), incluyendo el certificado, dos arreglos con los números de las páginas y las funcionalidades que accede respectivamente. En la primera posición del arreglo viaja una bandera booleana que indica si el login falló o no. En caso de fallar, las demás posición del arreglo estarán vacías.</p>	

3.8 Conclusiones

Con los elementos que tenemos luego de concluir este capítulo se está en condiciones de pasar a la próxima fase de propuesta de solución del problema, puesto que se han analizado y recopilado todos los elementos necesarios para llevar a cabo el proceso de realización de la aplicación.

4 Descripción de la solución propuesta

4.1 Introducción

En este capítulo se describe de manera general la solución que se presenta. Se explican los elementos que se tuvieron en cuenta para el diseño de la aplicación y los elementos necesarios para su comprensión. Se explican los mecanismos de diseño utilizados, patrones y estándares. Se muestran de manera resumida los diagramas de clases del diseño de cada caso de uso.

4.2 Principios de diseño

4.2.1 Descripción de las clases utilizadas

Clases	Descripción
Smarty	Es un motor de plantillas para PHP, que permite separar el contenido dinámico y el estático en la capa de presentación.
[Plantillas]	Son las clases con prefijo p y contienen en su estructura todo el contenido estático de una página Web. Estas serán empleadas por Smarty para generar las páginas clientes.
[Aportadora de Contenido]	Son las clases con prefijo ac y tienen la responsabilidad de aportar contenido dinámico a las plantillas procesadas por Smarty. El resultado de este proceso concluye con la generación de las página clientes. (interfaces de usuario)
acGeneral	Constituye la clase base de las aportadoras de contenido.
[Lógica del negocio]	Son las clases con prefijo In y encapsulan en sus métodos todas las operaciones específicas para cada caso de uso del sistema.
Global	Fichero con código PHP que contiene instancias y funciones que serán empleadas por todas las páginas servidoras.
[Típicas]	Son las clases con prefijo t , existe una típica para cada entidad de la base de datos. Implementan las operaciones básicas personalizadas que se realizan sobre las entidades del sistema. (insertar, modificar, eliminar, y consultar)
Factoría Típica	Es la clase encargada de gestionar la creación y utilización de objetos de tipo

	[Típicas].
mEntidad	Es la clase base de la cual heredan todas las típicas. Implementa de forma general las operaciones básicas a realizar con las entidades del sistema y maneja las conexiones a la BD. Emplean las clases PDO y PDOStatement.
mESimples	Es la clase base de la cual heredan todas las Típicas que responden a los nomencladores básicos o simples.
PDO y PDOStatement	Son clases pertenecientes a la librería PHP que permiten la gestión de la conexión y la consulta a la base de datos.

4.2.2 Mecanismo de Acceso a datos

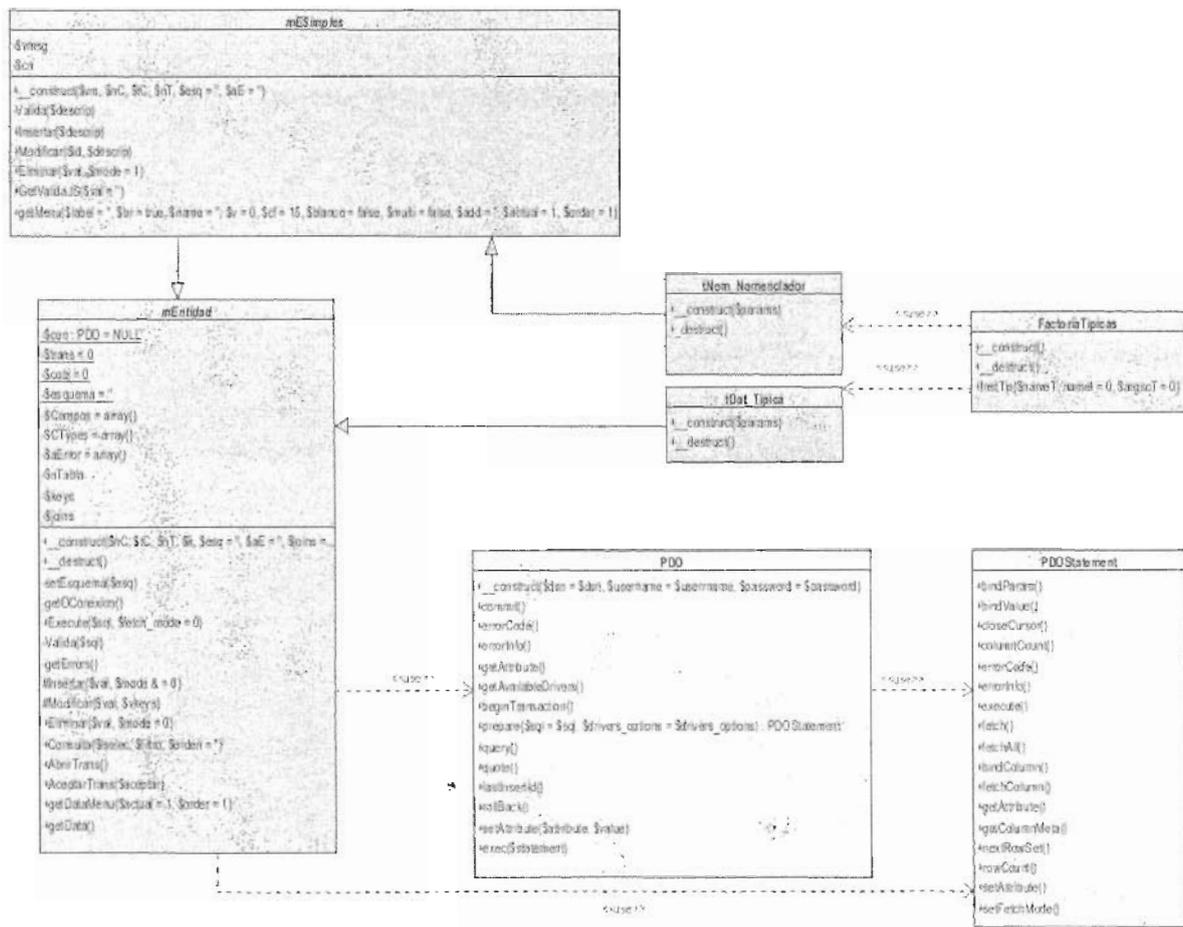
Con el transcurso de los años buscar mecanismos para modelar el acceso a datos siempre ha sido un meta a lograr, es por eso que un mecanismo aplicado para resolver esta situación nunca será igual a otro definido. Muchos son los especialistas que buscan un modelo ideal que sirva para todos los casos y se pueda emplear sin pensarlo dos veces, pero es una realidad que el desarrollo de hoy en día es tan vertiginoso que nos permite buscar soluciones tan buenas como otras ya definidas, convirtiéndose a su vez en mecanismos aplicables bajo ciertas circunstancias. Como todos sabemos el acceso y la manipulación de los datos es algo realmente indispensable a la hora de desarrollar un sistema informático.

Por todo esto a raíz de la aparición del concepto de patrones, se han definido algunos que abordan esta problemática., por lo cual nos resulta un punto de partida a la hora de modelar el “acceso a datos”.

¿Por qué definir un mecanismo para el acceso a datos?

Para acceder a los datos siempre están involucrados los mismos objetos y se efectúan un conjunto de operaciones comunes en las realizaciones de algunos casos de uso. De todo esto la necesidad de documentar un mecanismo que simplifique el modelado y que quede como punto de referencia para los desarrolladores. Todo esto nos permitirá obtener diagramas más entendibles, que nos permita una mayor comunicación con nuestro equipo de desarrollo. Pero lo más importante es que nos trazará una línea común, una política a seguir, fomentando algo muy indispensable para lograr eficiencia, la *reutilización*.

Para nuestro sistema en cuestión se plantea el siguiente mecanismo de diseño para modelar el “acceso a datos”.



Vista estática del mecanismo de diseño para persistencia. Conectividad usando entorno desarrollo brindado por el lenguaje PHP.

La vista estática de este mecanismo de acceso a datos muestra un conjunto de clases que interactúan para dar acceso y manipulación de los datos de la persistencia desde el nivel más bajo, es decir utilizando los objetos nativos brindados por el entorno de desarrollo PHP como son PDO y PDOStatement, siguiendo así hasta la abstracción del acceso a datos a través de MEntidad de la cual heredan las clases particulares de nuestro sistema como Típicas y MSimples.

Para dar la responsabilidad a una clase que encapsulara las instancias de estos objetos se definió la clase Factoría típicas ya antes mencionada en el patrón de diseño aplicado a este funcionamiento.

4.3 Diagrama de clases del diseño

4.3.1 Caso de uso: Configurar Sistemas

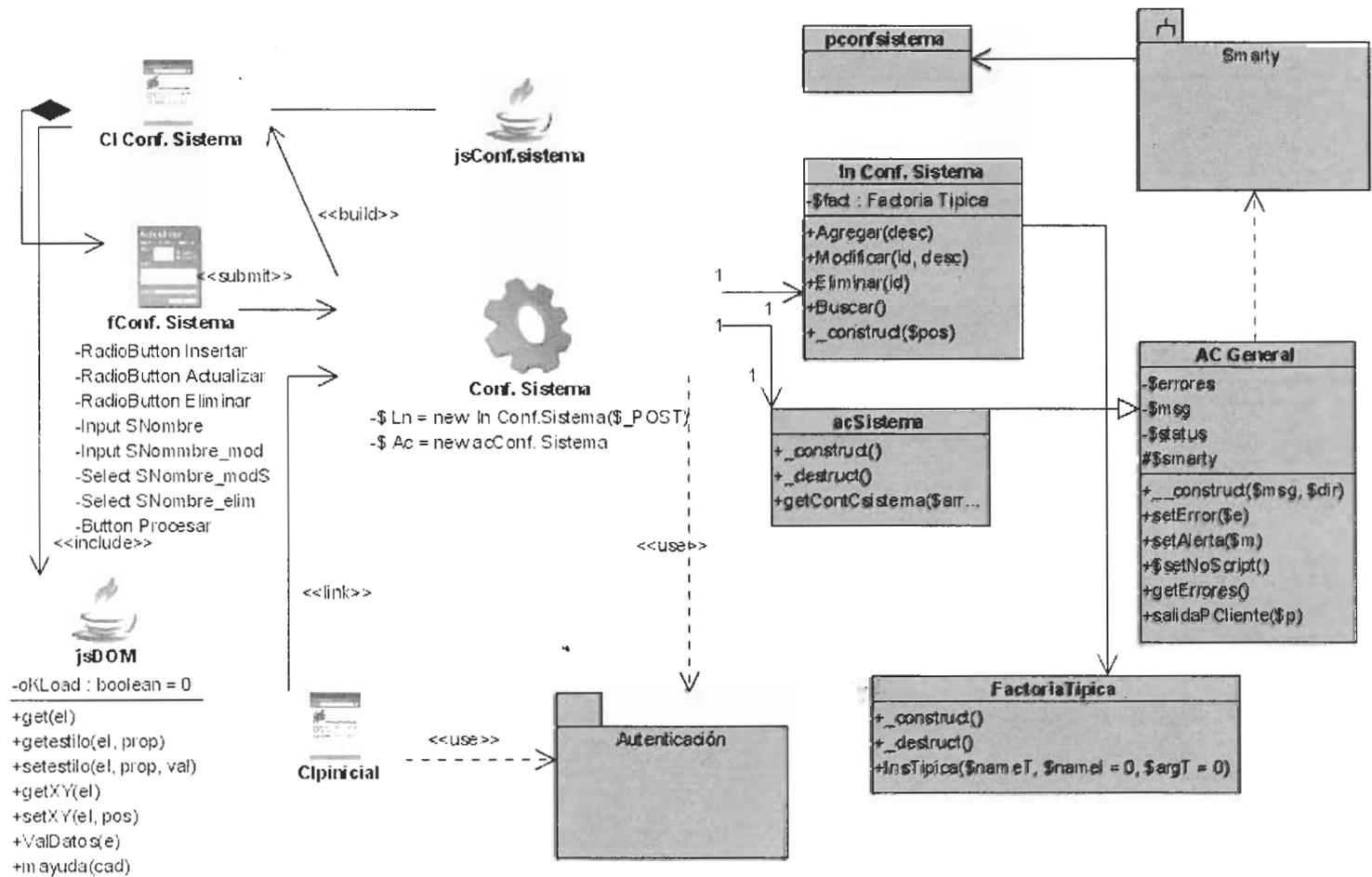


Figura 5. Diagrama de clases del caso de uso

4.3.2 Caso de uso: Configurar Grupo de usuarios

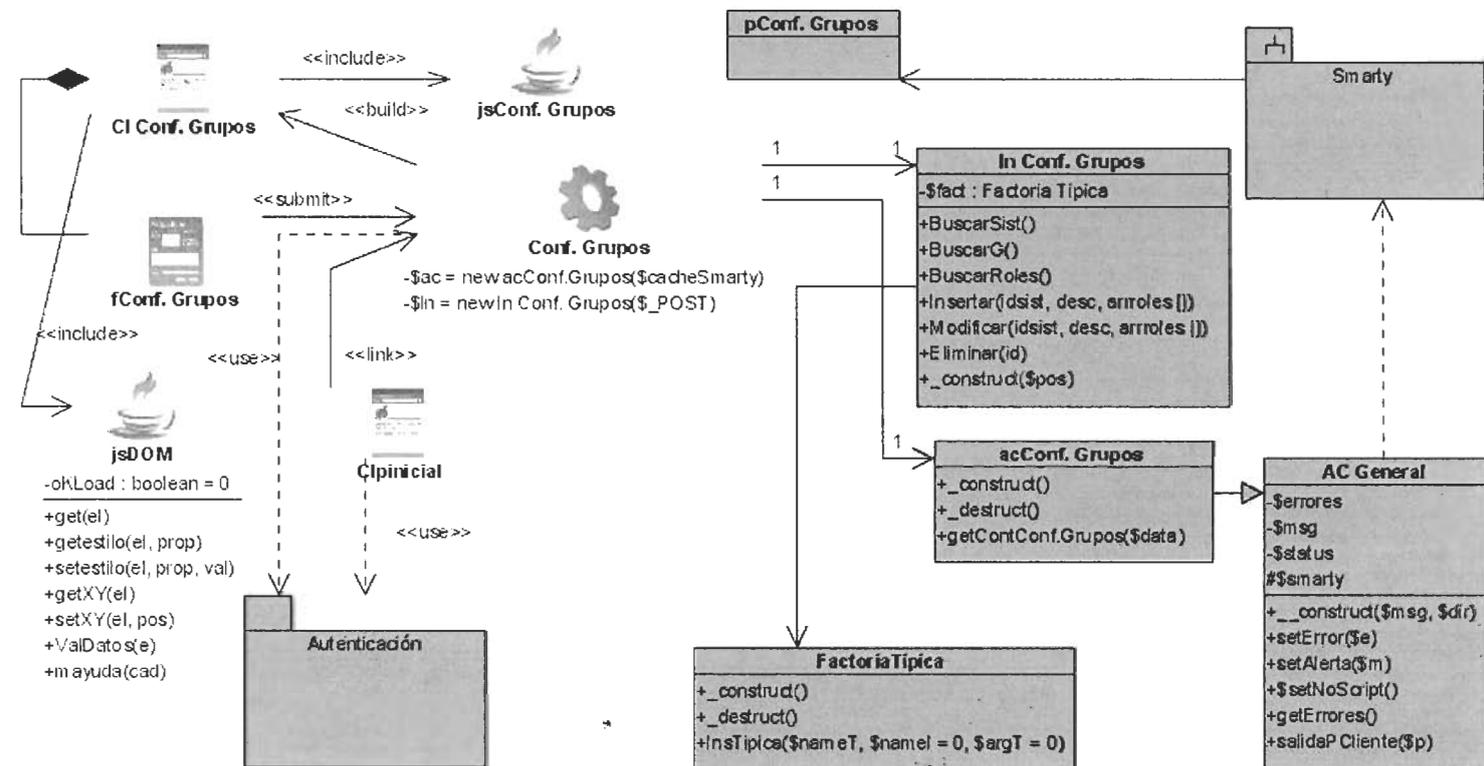


Figura 6. Diagrama de clases del caso de uso

4.3.3 Caso de uso: Configurar Roles

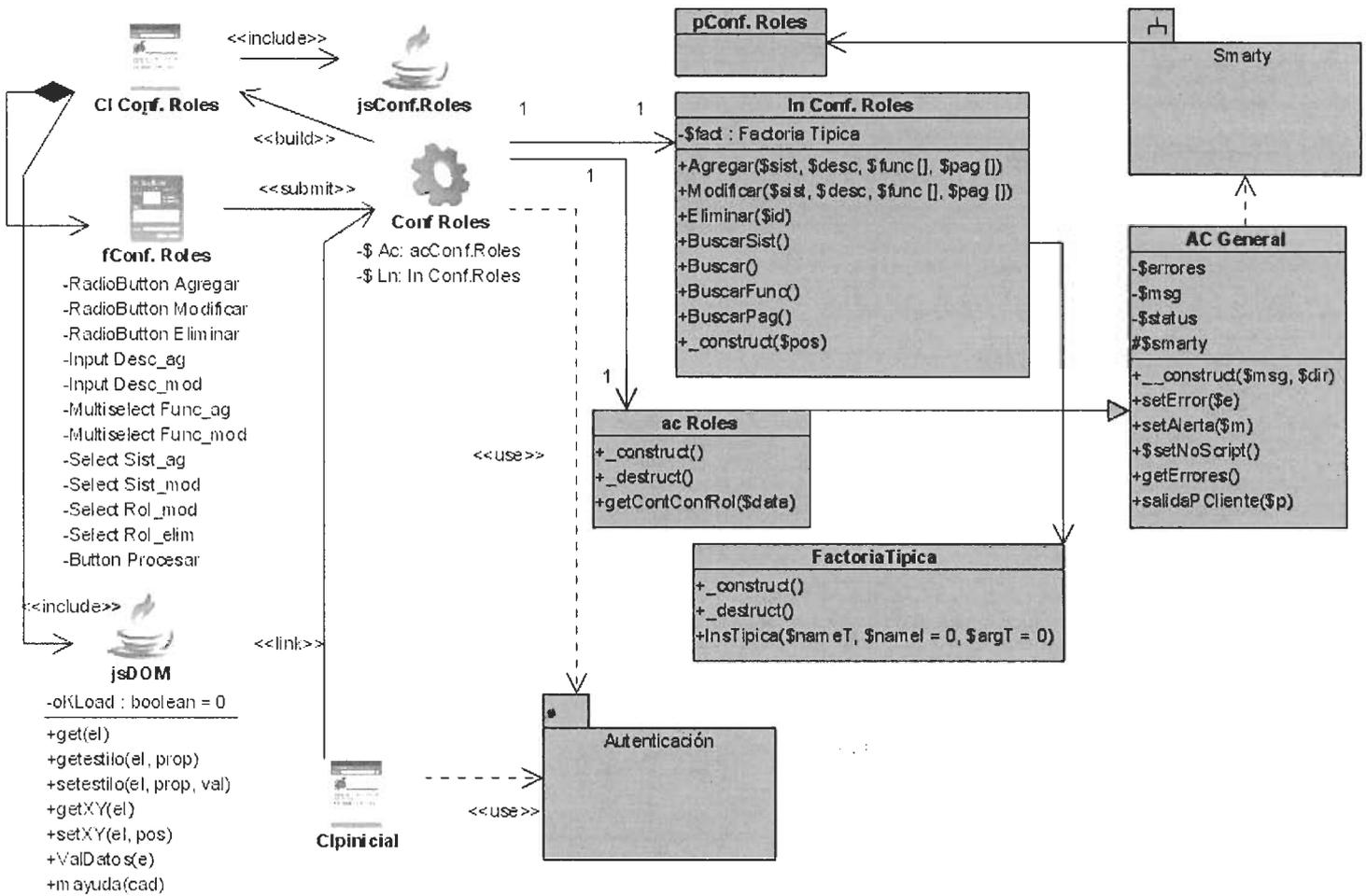


Figura 7. Diagrama de clases del caso de uso

4.3.4 Caso de uso: Insertar Usuario

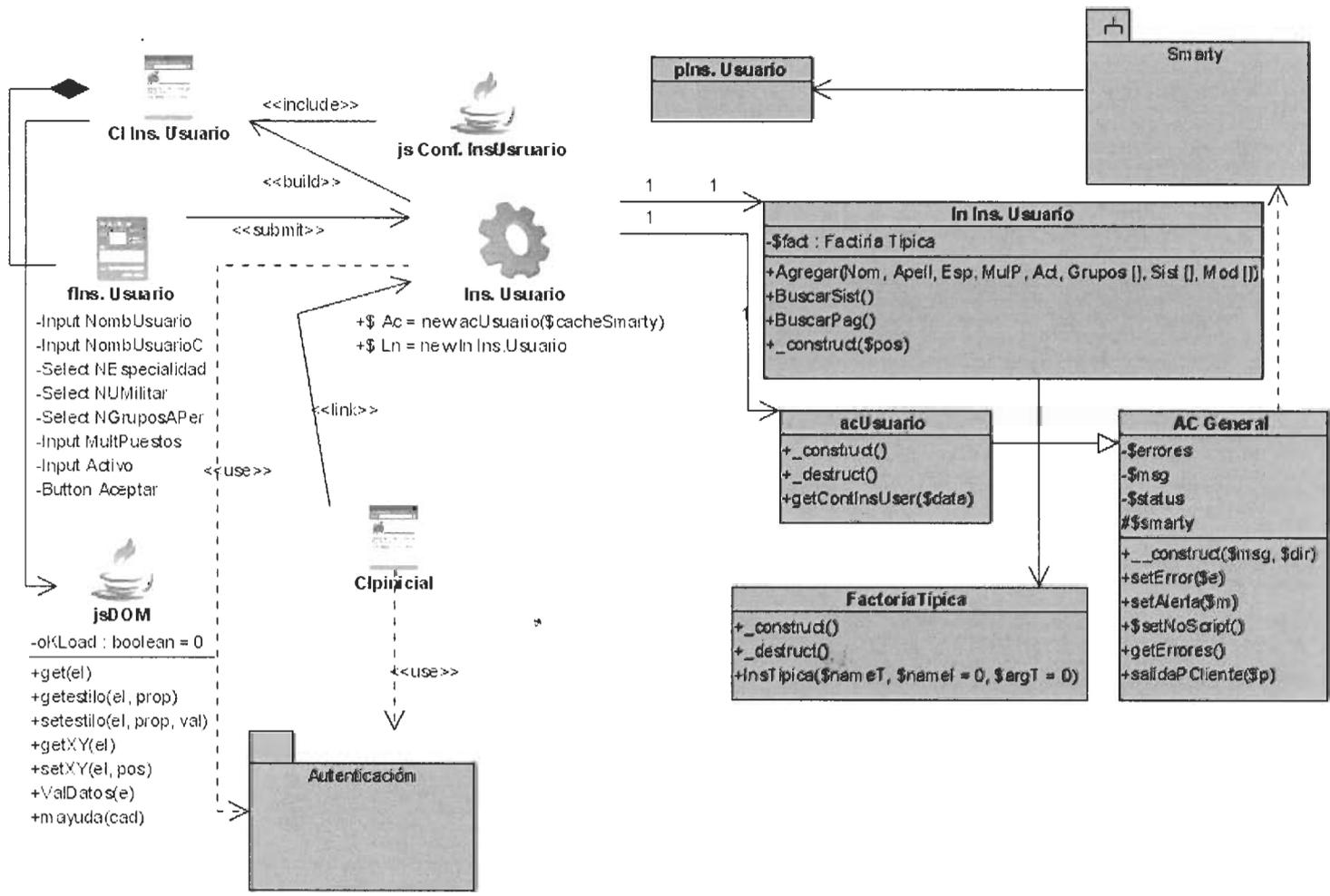


Figura 8. Diagrama de clases del caso de uso

4.3.5 Caso de uso: Actualizar Usuario

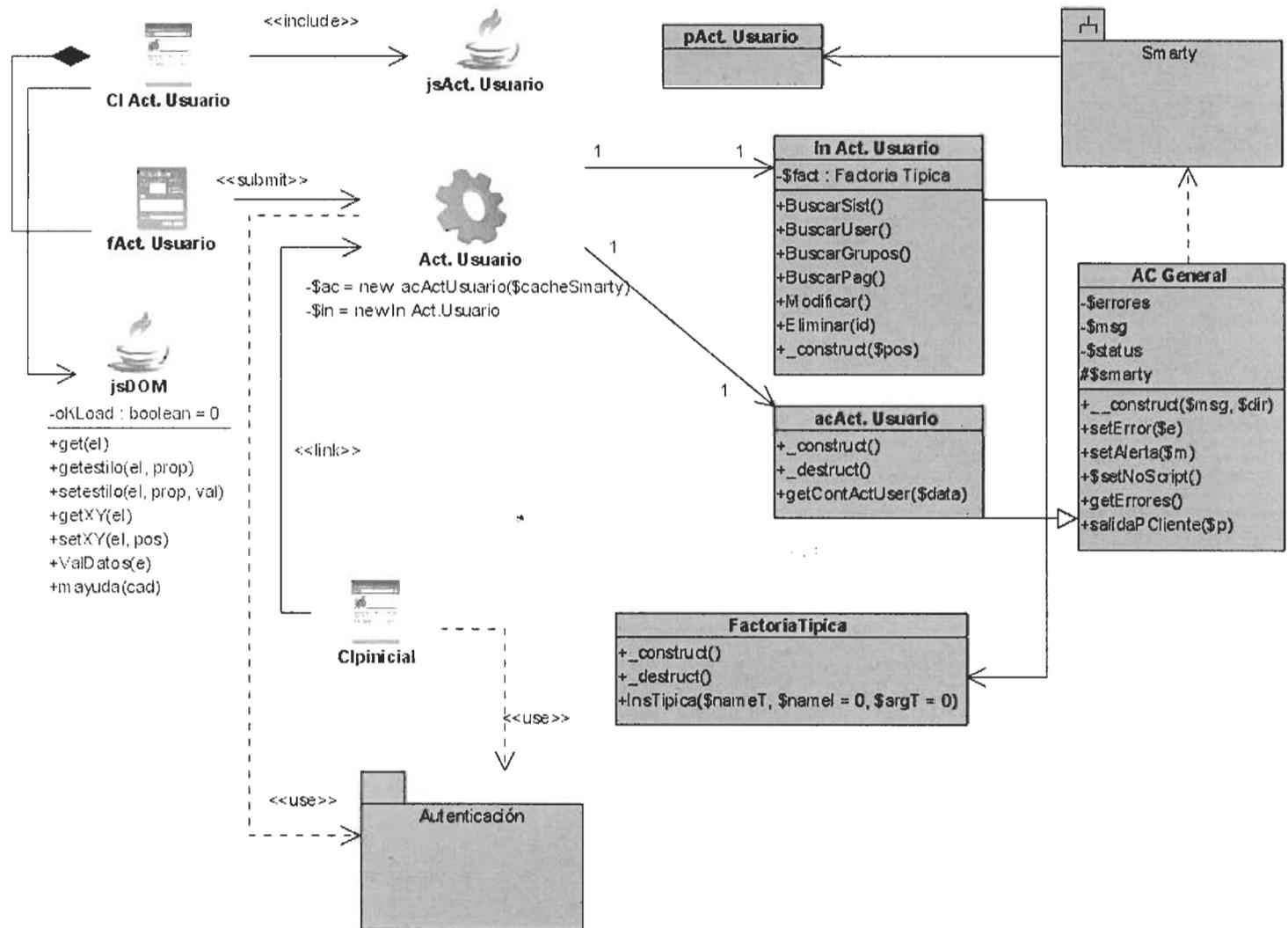


Figura 9. Diagrama de clases del caso de uso

4.3.6 Caso de uso: Configurar Páginas

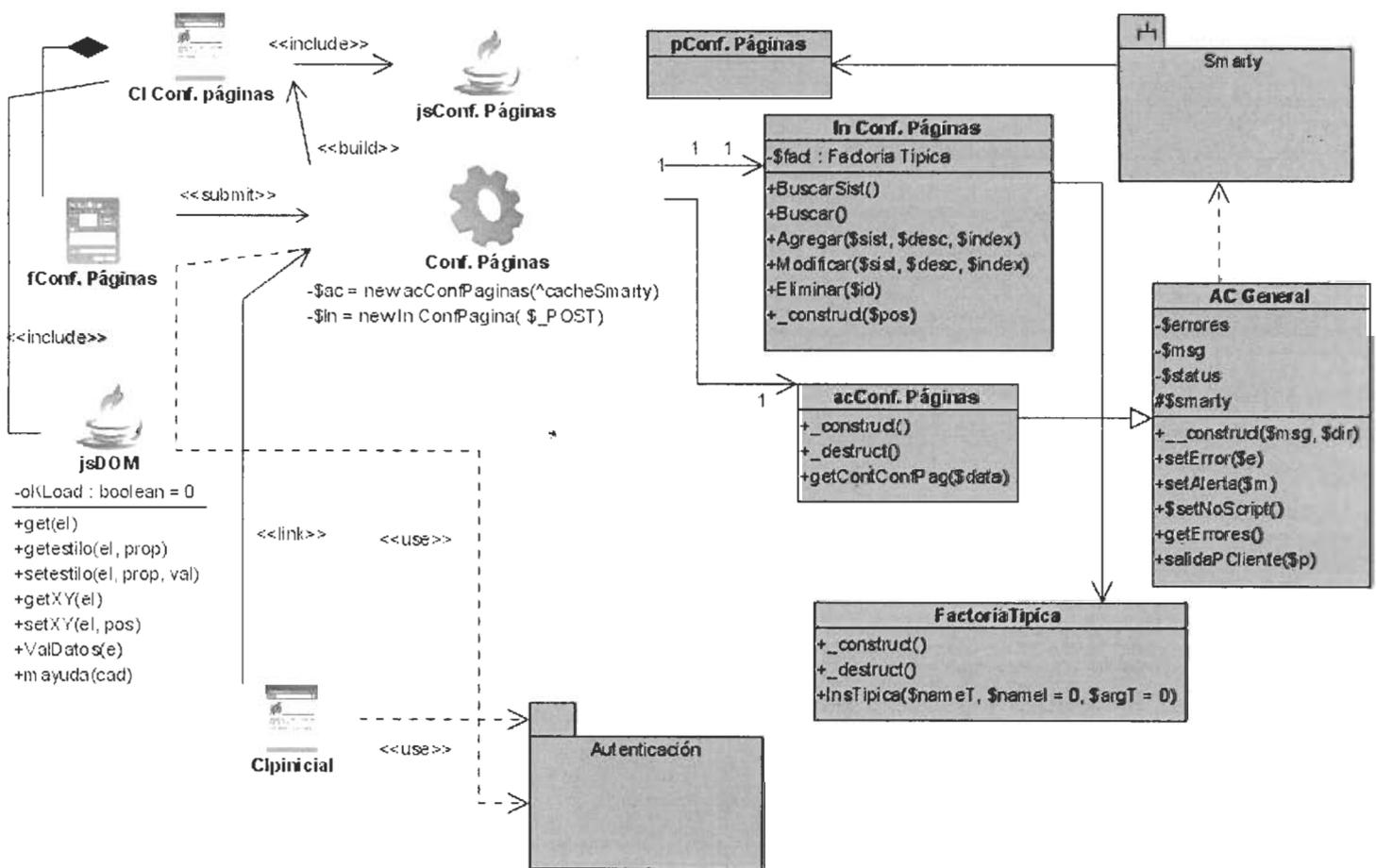


Figura 10. Diagrama de clases del caso de uso

4.3.8 Caso de uso: Configurar Eventos

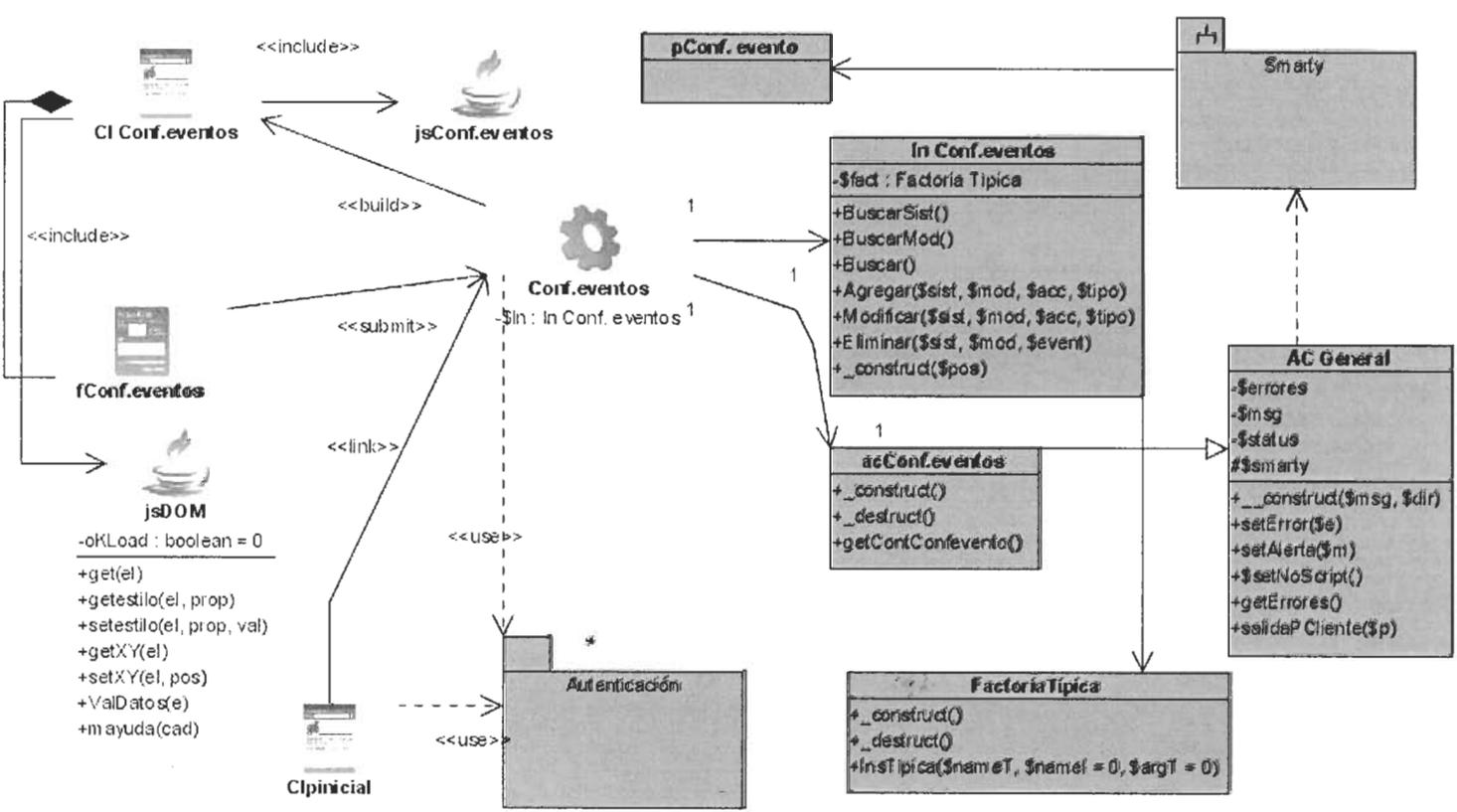


Figura 12. Diagrama de clases del caso de uso

4.3.9 Caso de uso: Configurar Módulos

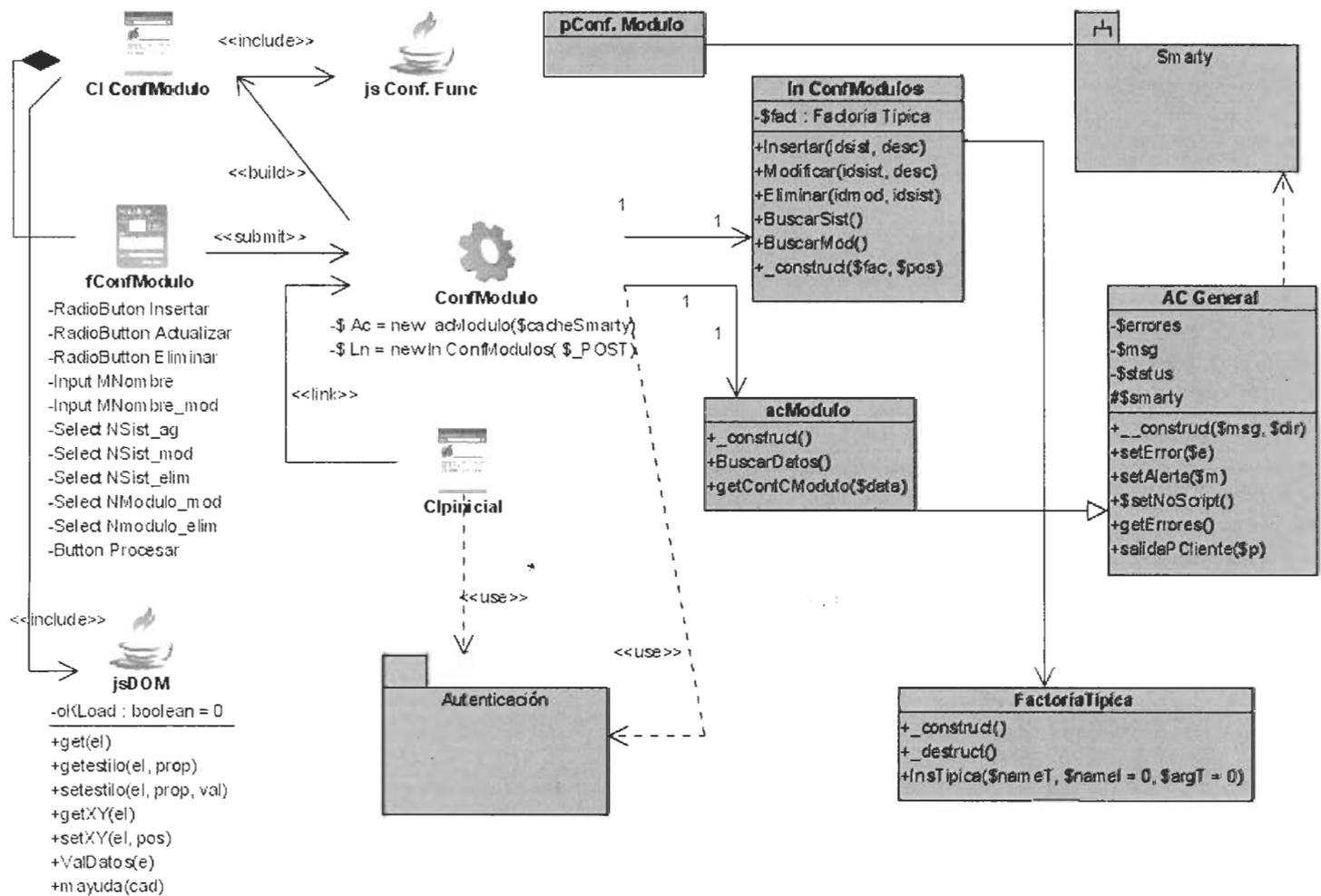


Figura 13. Diagrama de clases del caso de uso

4.3.10 Caso de uso: Crear Perfil

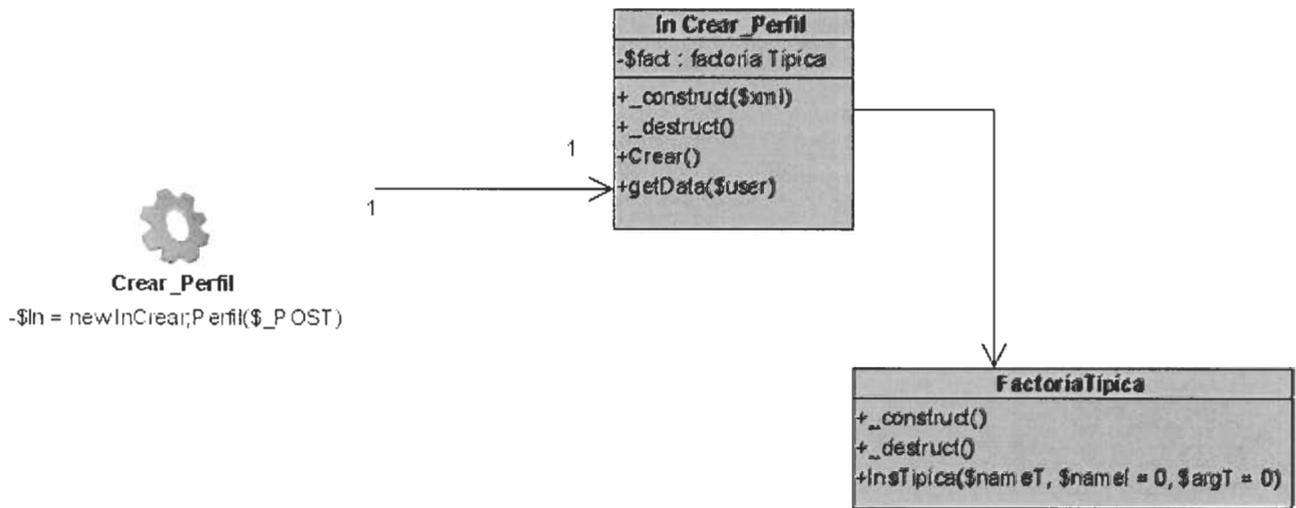


Figura 14. Diagrama de clases del caso de uso

4.3.11 Caso de uso: Eliminar Perfil

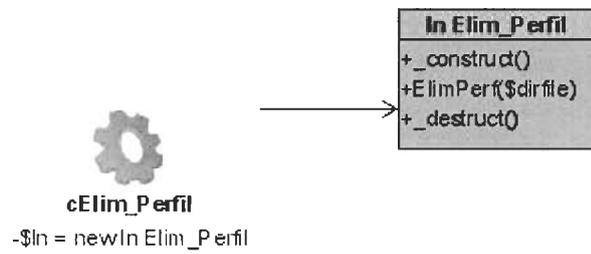


Figura 15. Diagrama de clases del caso de uso

4.3.12 Caso de uso: Registrar Evento

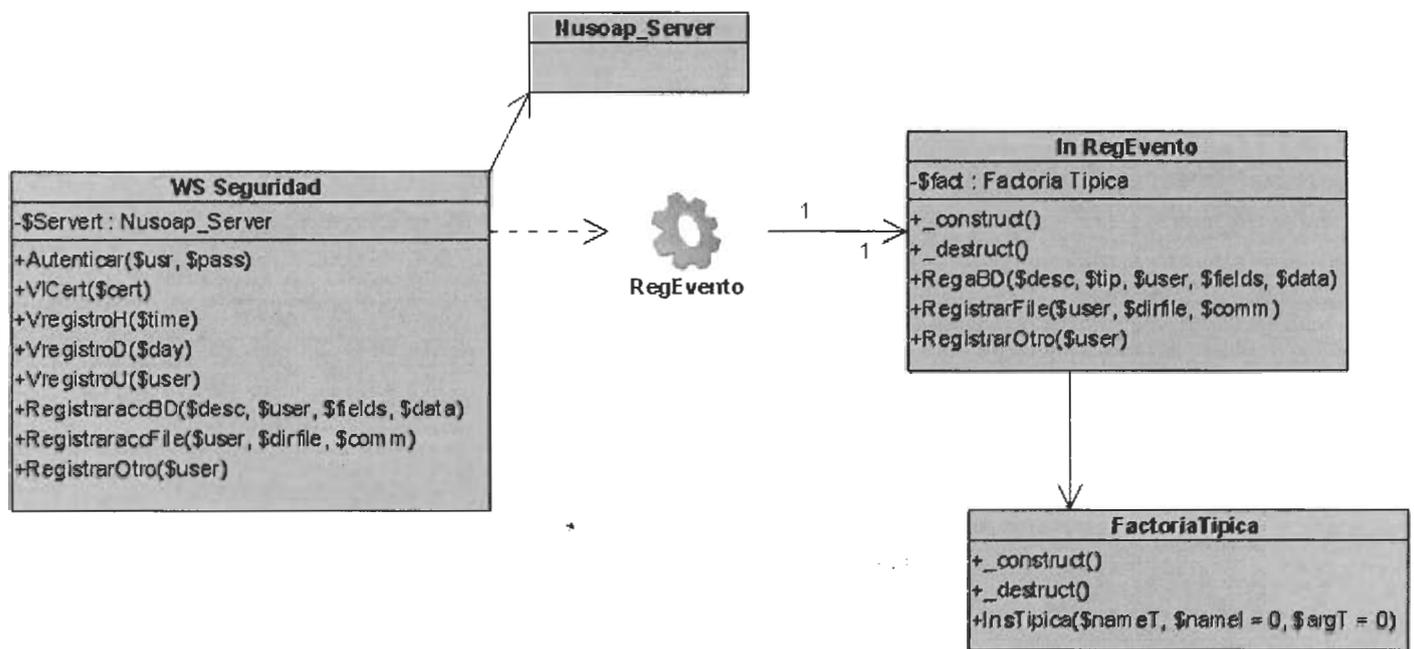


Figura 16. Diagrama de clases del caso de uso

4.3.13 Caso de uso: Autenticar usuario

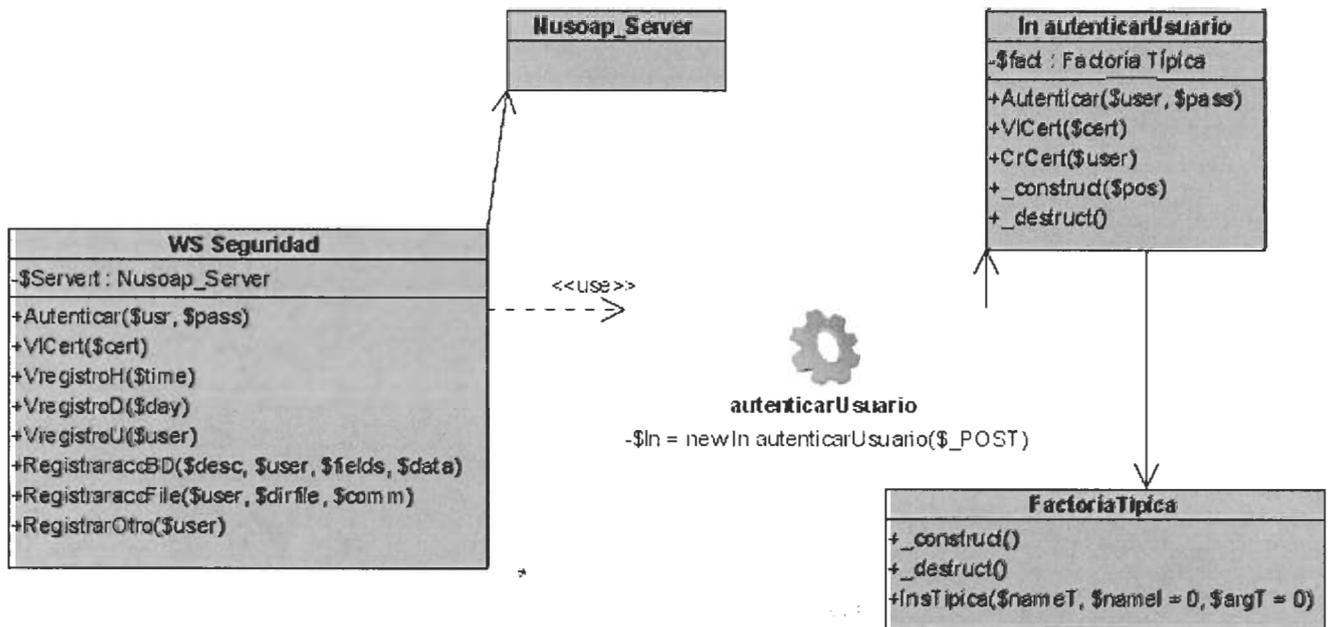


Figura 17. Diagrama de clases del caso de uso

4.3.14 Caso de uso: Visualizar Registro

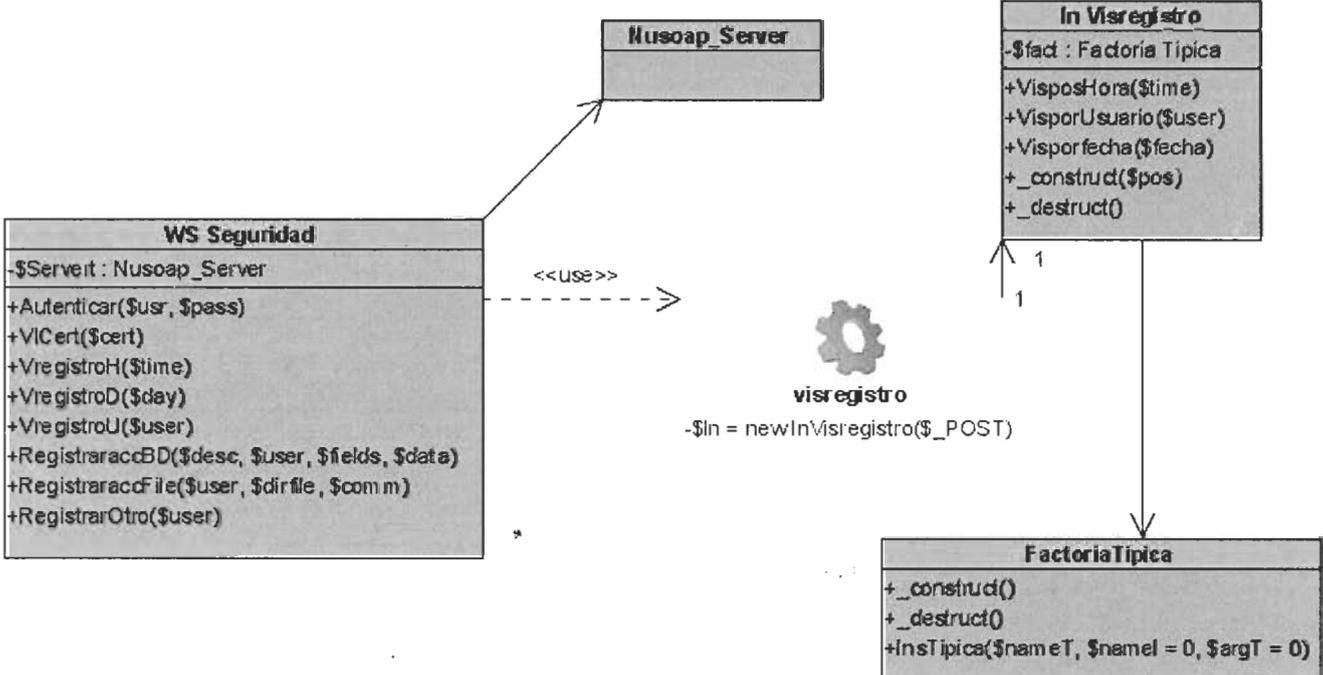


Figura 18. Diagrama de clases del caso de uso

4.4 Diagramas de Componentes

A continuación se muestra el diagrama de componentes:

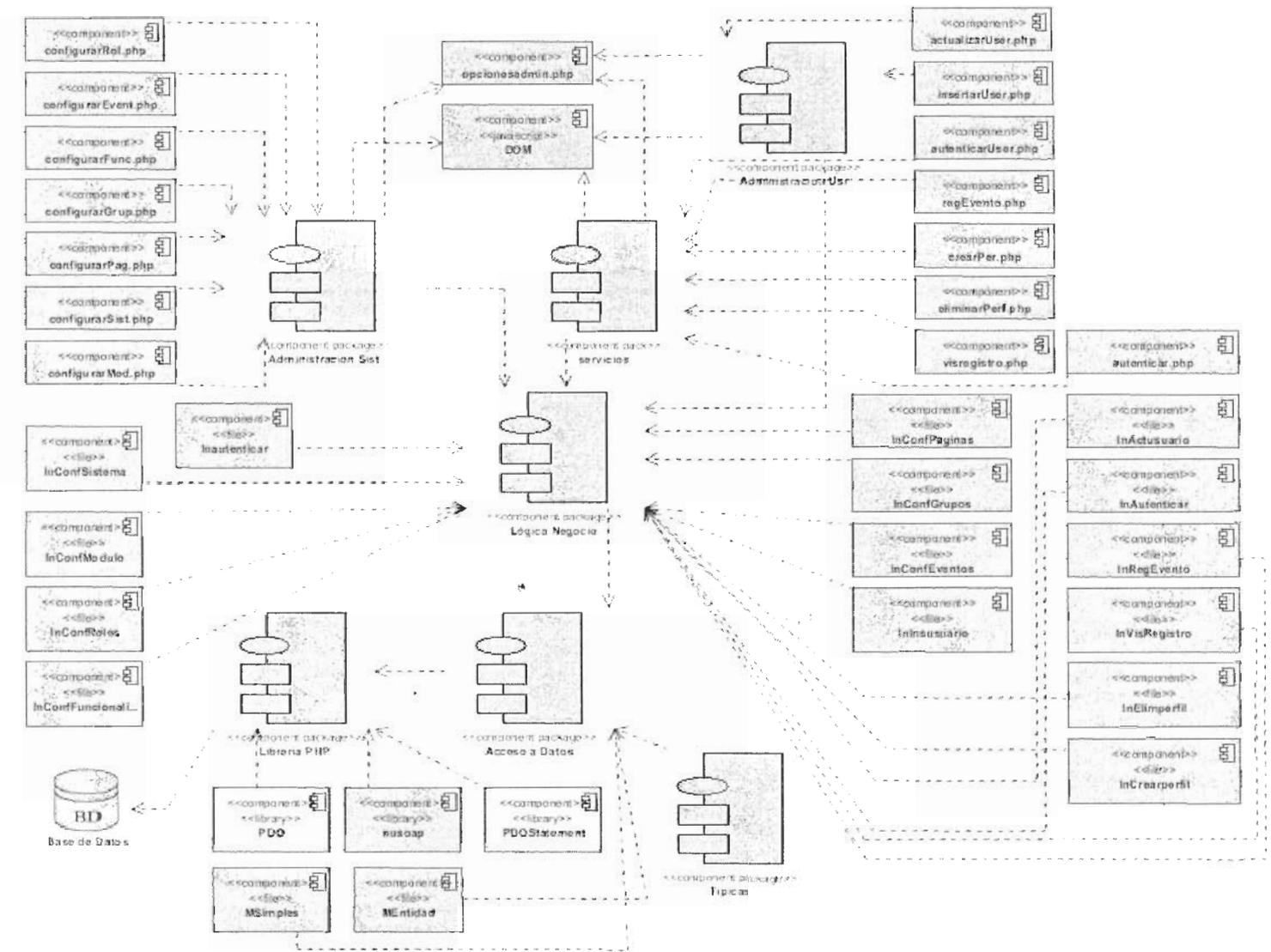


Figura 19 Diagrama de componentes.

4.5 Ayuda

La aplicación cuenta con un material complementario que se pone a disposición de los usuarios para el correcto manejo de la misma. Un manual avanzado de uso para el usuario, el mismo permite el dominio eficiente de cada una de las funcionalidades que brinda la aplicación y un correcto aprovechamiento de sus potencialidades.

4.6 Tratamiento de errores

De manera general los errores serán controlados por funciones JavaScript para evitar la ejecución de las páginas cuando existan errores. Este es el caso de los formularios de inserción/actualización, donde se validará la entrada correcta de los datos, además se incorpora la clase jsDOM que se encarga de la validación de la entrada de los datos en los formularios. En el caso de las eliminaciones se muestra mensaje de confirmación, de manera que no se realice una operación no deseada.

4.7 Estándar de Implementación

HTML

Propiedad name de los controles con igual nombre al del campo de la tabla en el gestor.

Mantener en JavaScript los mismos estándares de código que en PHP.

Formularios y Proyectos			
Objetivo: Nombrar los formularios y los proyectos de forma estándar para todas las aplicaciones.			
Apariencia de formularios y proyectos	de y	Primera letra en mayúscula	Los nombres de los formularios y proyectos comenzarán con la primera letra en mayúscula y el resto en minúscula, en caso de que sea un nombre compuesto se empleará la notación

		PascalCasing*. Ejemplo: MiProyecto.
Carpetas	Los nombres de los directorios serán todos en minúscula y como tendrán la estructura explicada anteriormente.	
Nombre de formularios y proyectos	Relacionados al propósito	El nombre empleado, debe permitir que con sólo leerlo se conozca el propósito del formulario o proyecto. Ejemplo: Sistema para la Demanda de la lucha armada, Sidem.
Clases y Objetos		
Objetivo: Nombrar las clases e instancias de las mismas de forma estándar para todas las aplicaciones. *		
Apariencia de clases y objetos	Primera letra en mayúscula	Los nombres de las clases y las instancias de las mismas deben comenzar con la primera letra en mayúscula y el resto en minúscula, en caso de que sea un nombre compuesto se empleará notación PascalCasing*. Ejemplo: MiClase(). Debemos decir que en ocasiones se utilizarán prefijos para denotar el tipo de la clase siempre en minúsculas. Preferiblemente

		prefijos de una sola letra.
Nombre de clases y objetos	Relacionados al propósito	El nombre empleado, debe permitir que con sólo leerlo se conozca el propósito de la clase o instancia de la misma. Para el caso de las instancias es recomendable que se denoten así: Para la clase: Nomumedida su instancia será \$Oumedida, de forma tal que la primera letra indique que es un objeto y el resto, la clase a la que pertenece.
Apariencia de atributos	Primera letra en minúscula	El nombre que se le da a los atributos de las clases debe comenzar con la primera letra en minúscula, en caso de que sea un nombre compuesto se empleará notación CamelCasing**.
Nombre de atributos	Nemotécnicos	El nombre empleado, debe permitir que con sólo leerlo se conozca el propósito del mismo

		dentro de la clase. Ejemplo: \$nTabla, este atributo denota el nombre de una tabla.
Apariencia de las funciones	Primera letra en mayúscula	Los nombres de las funciones deben comenzar con la primera letra en mayúscula y el resto en minúscula, en caso de que sea un nombre compuesto se empleará notación PascalCasing*. Ejemplo: function BuscaUnidad(). Si son funciones que obtienen un dato se emplea el prefijo get y si fijan algún valor se emplea el prefijo set
Nombre de las funciones	Nemotécnicos	El nombre empleado, debe permitir que con sólo leerlo se conozca el propósito de la misma dentro de la clase.
Declaración de parámetro en funciones	Agrupados por tipos Poner los string 1 numéricos 2, además, agrupar según valores por defecto	Los parámetros que se le pasan a las funciones se recomienda sean declarados de forma tal que estén agrupados por el tipo de dato que contienen. Ejemplo: BuscaUnidad(\$nTabla (string), \$nCampos(string), \$kIndice (entero)).
Variables y constantes		
Apariencia de	Primera letra en	El nombre que se le da a las

variables	minúscula.	variables debe comenzar con la primera letra en minúscula, en caso de que sea un nombre compuesto se empleará notación CamelCasing**.
Apariencia de constantes	Todas sus letras en mayúscula	Se deben declarar las constantes con todas sus letras en mayúscula.
Nombres de las variables y constantes	Nemotécnicos	El nombre empleado, debe permitir que con sólo leerlo se conozca el propósito de la misma. Ejemplo: \$nFields.
Declaración de constantes y asignación a variables	Una por cada línea	Se recomienda declarar una constante por cada línea y con las asignaciones a las variables sucede lo mismo. Ejemplo: define("CONSTANT1","value1"); define("CONSTANT2","value2"); \$nTabla='nomproducto'; \$kIndice=0;
Indentación		
Objetivo: Lograr una estructura uniforme para los bloques de código así como para los diferentes niveles de anidamiento.		
0 espacios en blanco desde la izquierda en	Require Include Class	No se empleará ningún espacio en blanco desde la izquierda para las instrucciones antes mencionadas. Se tomará como inicio de la página el tag PHP <?
2 espacio en blanco	Function	Se dejarán dos espacios en

desde la izquierda en	Define	blanco desde la izquierda en las instrucciones antes mencionadas.
2 espacio en blanco desde la referencia en	Inicio y fin de bloque	Se recomienda dejar dos espacios en blanco desde la instrucción anterior para el inicio y fin de bloque {}. Lo mismo sucede para el caso de las instrucciones If, else, For, While, Do While, Switch, Foreach.
Niveles de anidación	Hasta 5 niveles	Se recomienda emplear hasta 5 niveles de anidación en instrucciones If, For, While.

Ejemplo de indentación

```
<?
require ('class/Interface.php'); *

class MiClase
{
function BuscaUnidad($nTabla, $nFields, $kIndice)
{
    if ($nTabla)
    {
        ...
    }
    for (...)
    {
        ...
    }
}
```

<pre> } } ?> </pre>		
Comentarios, separadores, líneas y espacios en blanco		
Objetivo: Establecer un modo común para comentar el código de forma tal que sea comprensible con sólo leerlo una vez.		
Ubicación de comentarios	Al inicio de cada clase o función y al final de cada bloque de código.	Se recomienda comentar al inicio de la clase o función especificando el objetivo de la misma así como los parámetros que usa (especificar tipos de dato, y objetivo del parámetro) entre otras cosas. Y se comenta también cuando se cierran los ciclos, clases, instrucciones if y otras.
Separador de instrucciones	Se emplea el punto y coma.	Se recomienda usar el separador al final de cada instrucción y no en la línea de abajo. Ejemplo: define ("CONSTANT", "value1");
Líneas en blanco	Se emplean antes de cada función.	Se recomienda dejar una línea en blanco antes de la definición de cada función para dar claridad al código.
Espacios en blanco	Entre operadores lógicos y aritméticos.	Se recomienda usar espacios en blanco entre estos operadores para lograr una mayor legibilidad en el código. Ejemplo:

		\$nTabla = 'nomproducto'; if ((\$nTabla) && (\$nFields))
Bases de Datos, Tablas, esquemas y Campos		
Apariencia de la BD	Primera letra en mayúscula	Los nombres de las BDs deben comenzar con la primera letra en mayúscula y el resto en minúscula, en caso de que sea un nombre compuesto se empleará notación PascalCasing*. Ejemplo: ContMaterial.
Nombres de las BDs	Nemotécnicos y relacionados al propósito.	El nombre empleado, debe permitir que con sólo leerlo se conozca el propósito de la misma.
Apariencia de los esquemas	Todas las letras en minúscula.	El nombre a emplear para los esquemas debe escribirse con todas las letras en minúscula para evitar problemas con el Case Sensitive del gestor. Ejemplo: create schema 'finanzas';
Nombres de los esquemas	Nemotécnicos y relacionados al propósito.	El nombre empleado, debe permitir que con sólo leerlo se conozca el propósito del mismo.
Apariencia de las tablas	Todas las letras en minúscula.	El nombre a emplear para las tablas debe escribirse con todas las letras en minúscula para evitar problemas con el

		Case Sensitive del gestor. Ejemplo: create table 'nomproducto';
Nombres de las tablas	Nemotécnicos y relacionados al propósito. Además clasificando las tablas por su tipo.	El nombre empleado, debe permitir que con sólo leerlo se conozca el propósito del mismo. Se deben clasificar las tablas por su tipo, es decir por los datos que contienen se le coloca un prefijo, se pueden clasificar en: Nomencladores, tablas de datos, de auditoría, de seguridad, de configuración etc... Ejemplo: nomproducto (Nomeclador). seg_usuarios (de seguridad). conf_almacen (de configuración).
Apariencia de los campos	Todas las letras en minúscula.	El nombre a emplear para los campos debe escribirse con todas las letras en minúscula para evitar problemas con el Case Sensitive del gestor. Ejemplo: add field 'idproducto';
Nombre de los campos	Nemotécnicos	El nombre empleado, debe permitir que con sólo leerlo se conozca el propósito del

		<p>mismo. Además se debe incluir un comentario en la descripción del mismo. Ejemplo:</p> <p>cantemb: cantidad de embalajes.</p>
Nombre de las llaves primarias	Nemotécnicos empleando prefijos.	<p>Se nombrarán las llaves primarias de forma que se vea de qué tabla es y que es primaria. Ejemplo:</p> <p>pk_cuenta. (Llave primaria de la tabla cuenta). Si es una llave compuesta se coloca el prefijo y en nemotécnico los campos que la forman.</p>
Nombre de las llaves foráneas.	Nemotécnicos empleando prefijos.	<p>Se nombrarán las llaves foráneas de forma que se vea de qué tabla es y que es foránea. Ejemplo:</p> <p>fk_cuenta. (Llave foránea de la tabla cuenta). Si es una llave compuesta se coloca el prefijo y en nemotécnico los campos que la forman.</p>
Nombre de las secuencias	Nemotécnicos empleando prefijos.	<p>Se nombrarán las secuencias de forma que se vea de qué campo es y que es una secuencia. Ejemplo:</p> <p>seq_idcuenta. (Secuencia del campo idcuenta).</p>
Nombres de las	Nemotécnicos	El nombre empleado, debe

funciones, triggers, y vistas		permitir que con sólo leerlo se conozca el propósito del mismo.
----------------------------------	--	---

- ***Notación PascalCasing:** Los identificadores y nombres de variables, métodos y funciones están compuestos por múltiples palabras juntas iniciando cada palabra con letra mayúscula. Ejemplo: NotacionPascalCasing.
- ****Notación CamelCasing:** Los identificadores y nombres de variables, métodos y funciones están compuestos por múltiples palabras juntas iniciando cada palabra con letra mayúscula excepto la primera palabra que debe iniciar con minúscula. Ejemplo: notacionCamelCasing.

4.8 Diseño de la base de datos

4.8.1 Modelo lógico de datos

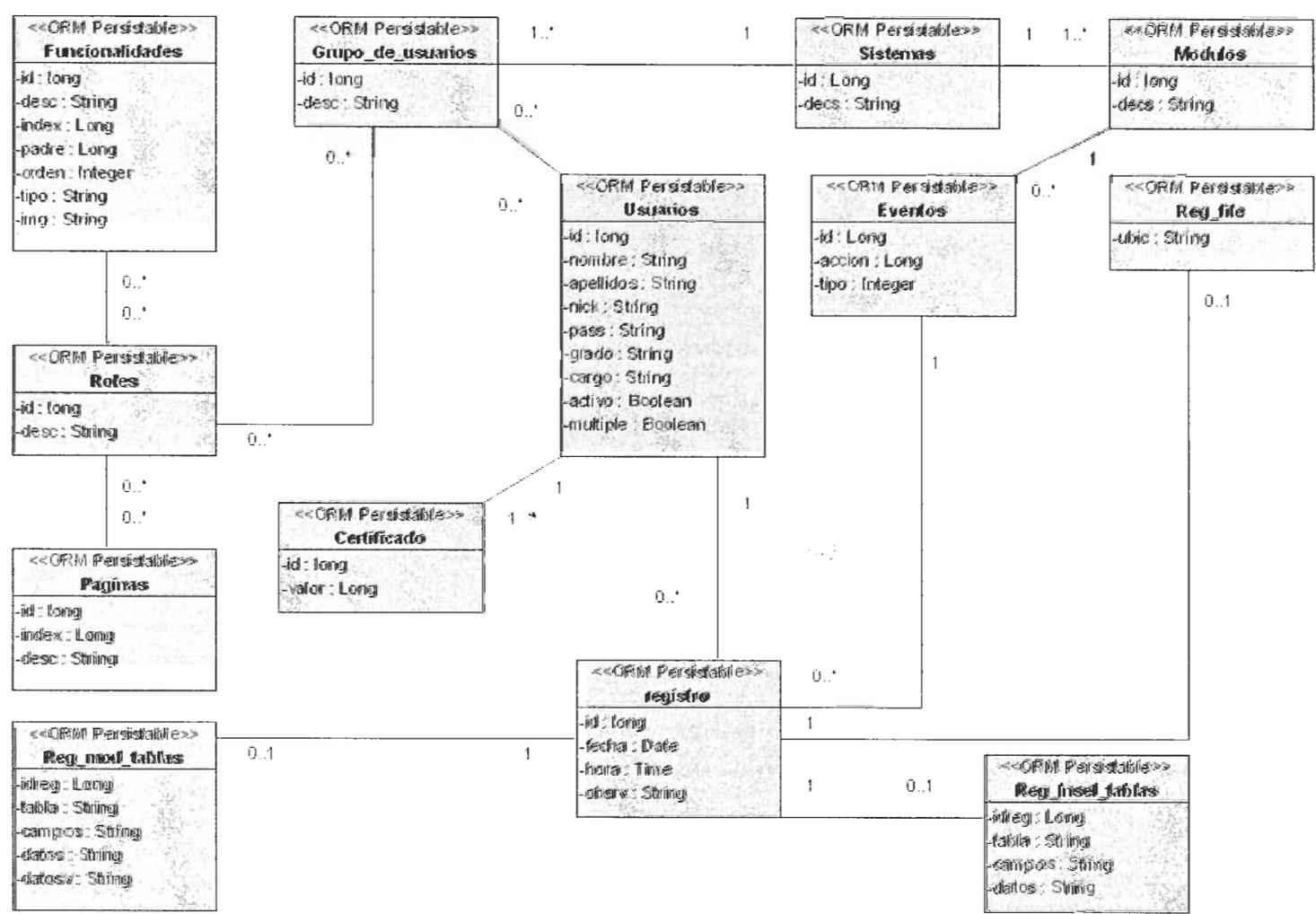
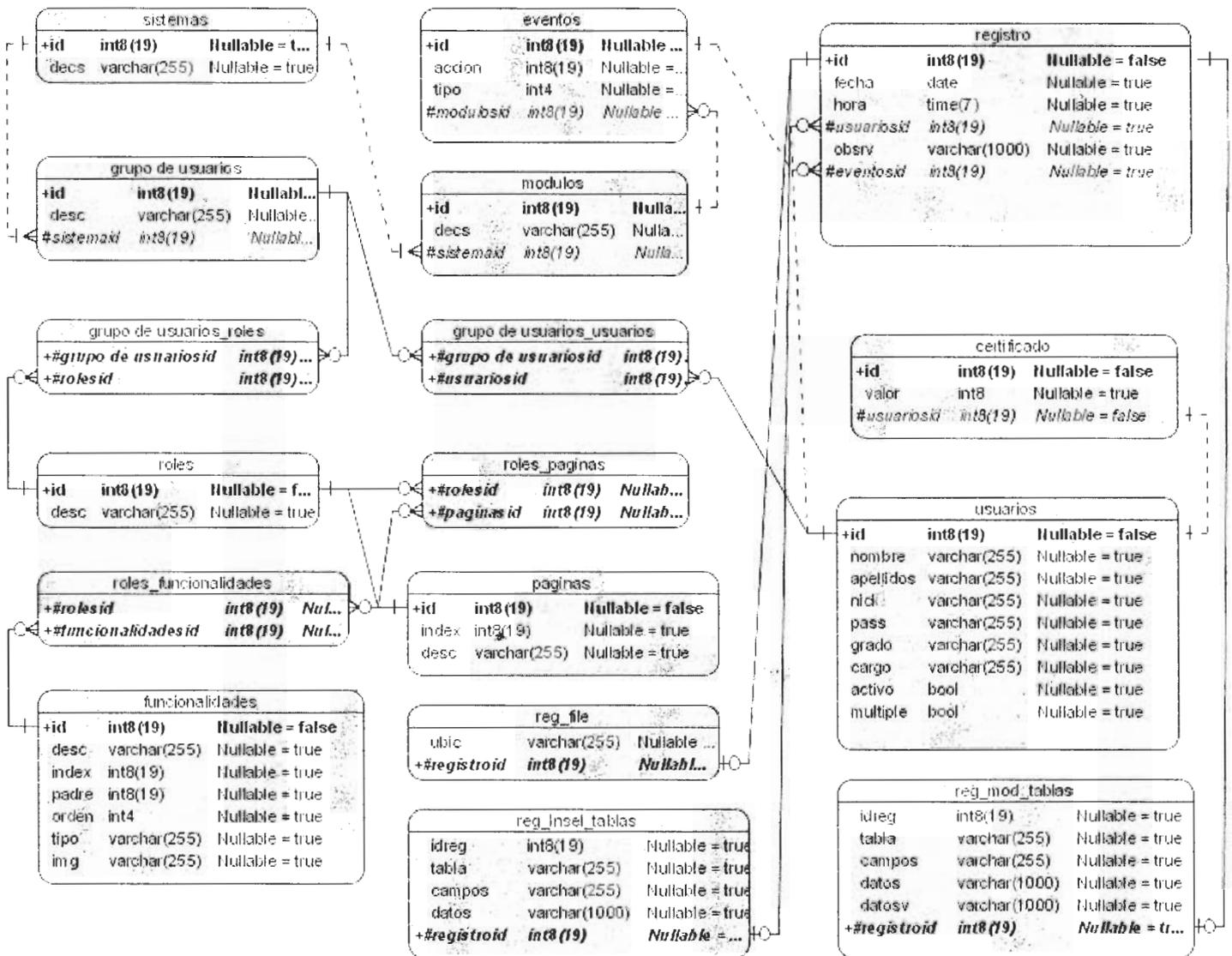


Diagrama de clases persistentes

4.8.2 Modelo físico de datos



4.9 Diagrama de despliegue

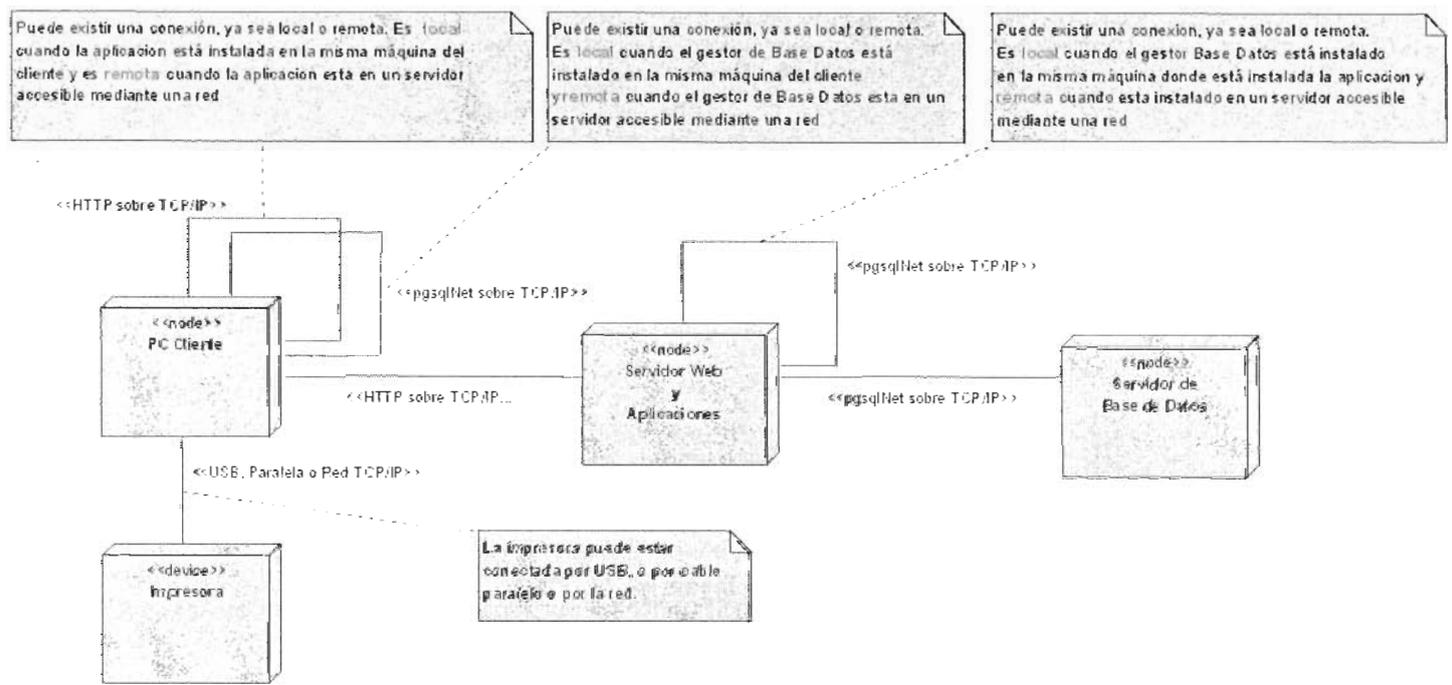


Figura 20. Diagrama de despliegue

4.10 Conclusiones

En el presente capítulo se hizo una descripción detallada de la solución propuesta, el modelo del sistema, la infraestructura donde se pondrá en marcha la aplicación y la estructura de la base de datos.

Una vez terminado el análisis realizado en este capítulo ya se tienen todos los elementos para comenzar la fase siguiente del proceso de desarrollo de software.

5 Estudio de factibilidad

5.1 Introducción

El análisis del costo de un proyecto es imprescindible a la hora de acometer una tarea, es la forma que se tiene de saber si la realización del mismo es factible o no. En este capítulo hacemos un análisis del costo y beneficios que tendría la realización del proyecto utilizando la planificación basada en casos de uso.

5.2 Planificación basada en casos de uso

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

$$UAW = S(\text{Factor} * \text{Actores})$$

UAW

5

Tipo de CU	Descripción	Peso	Cantidad de CU	Total
Simple	El caso de uso tiene de 1 a 3 transacciones.	5	7	35
Medio	El caso de uso tiene de 4 a 7 transacciones.	10	8	80
Complejo	El caso de uso tiene más de 8 transacciones.	15	0	0

$$UUCW = \text{Sumatoria}(\text{Factor} * \text{CantCU})$$

UUCW

115

$$UUCP = UAW + UUCW$$

UUCP

120

Factor	Descripción	Peso	Valor asignado	Total
T1	Sistema distribuido	2	5	10
T2	Tiempo de respuesta	1	5	5
T3	Eficiencia del usuario final	1	5	5
T4	Funcionamiento Interno complejo	1	2	2

T5	El código debe ser reutilizable	1	5	5
T6	Facilidad de instalación	0,5	5	2,5
T7	Facilidad de uso	0,5	5	2,5
T8	Portabilidad	2	5	10
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	3	3
			Sumatoria	59
TCF = 0.6 + 0.01 * Sumatoria(Peso * Valor)			TCF	1,19

Factor	Descripción	Peso	Valor asignado	Total
E1	Familiaridad con el modelo de proyecto utilizado	1,5	4	6
E2	Experiencia en la aplicación	0,5	4	2
E3	Experiencia en la orientación a objetivos.	1	4	4
E4	Capacidad del analista líder.	0,5	3	1,5
E5	Motivación.	1	5	5
E6	Estabilidad de requerimientos	2	3	6
E7	Personal Part-Time	-1	0	0
E8	Dificultad del lenguaje de programación	-1	3	-3

Sumatoria 21,5
 EF = 1.4 - 0.03 *
 Sumatoria(Peso * Valor) EF 0,755
 Factor de conversión CF 20
 UCP = UUCP * TCF * EF UCP 107,814
 E = UCP * CF E 2156,28

Actividad	Porcentaje %	Horas-Hombres
Análisis	10	539,07
Diseño	20	1078,14
Implementación	40	2156,28
Pruebas	15	808,605
Sobrecarga (otras actividades)	15	808,605
Total	100	5390,7

Esfuerzo Total (Horas--Hombre) ET1 5390,7
 Esfuerzo Total (Mes--Hombre) ET2 24,9569444

Salario Promedio	SM	50
Cantidad de Hombres	CH	3
Costo Hombre--Mes	CHM	50
Costo Total	Costo	3743,54167

5.3 Beneficios tangibles e intangibles

El desarrollo de la aplicación trae consigo a la entidad una serie de beneficios importantes debido a que primeramente disminuye el costo y el tiempo de producción de otras aplicaciones que se desarrollen en la entidad, puesto que ya no tendrán que tener incorporadas las funcionalidades de control de la seguridad, lo cual implica menos esfuerzo para el desarrollo de nuevas aplicaciones y concentrarlos en los elementos importantes para lo que fueron concebidas. Abre un camino hacia el desarrollo de aplicaciones Open Source, de gran importancia para disminuir considerablemente el costo de producción y los gastos asociados que trae consigo el proceso de desarrollo de software.

5.4 Análisis de costos y beneficios

El desarrollo de la aplicación no constituye un gasto considerable a la entidad ya que se desarrolla con herramientas de programación Web poco costosas. El uso de dichas herramientas disminuye los gastos implicados en el proceso de desarrollo de software.

5.5 Conclusiones

Una vez terminado el estudio de factibilidad del sistema, se estima un tiempo de aproximadamente 8 meses para su construcción por un hombre y su costo asciende a 3743,54 pesos. La herramienta propuesta trae consigo una serie de beneficios para la organización, ya que la misma va a contribuir a mejorar su funcionamiento, lo que indica que es factible implementar la herramienta propuesta.

Conclusiones

Con la implementación del sistema de seguridad se logra controlar de manera centralizada la seguridad de acceso y el seguimiento de las acciones hechas por los diferentes usuarios durante el uso de la aplicación, lo que es de suma importancia por el alto nivel de sensibilidad de la información que se maneja. Además se hace de una manera accesible a todos los sistemas que están y estarán en funcionamiento dentro de la institución, ya que se garantiza interoperabilidad entre las diferentes aplicaciones. La seguridad es el aspecto medular de los sistemas que procesan información confidencial y un estricto control de esta garantiza eficiencia en la tarea para la que fueron destinados.

El desarrollo de esta aplicación constituye un aporte práctico muy importante, debido a que es una novedad tecnológica que marca un proceso de avance en el desarrollo de software multiplataforma e interoperable dentro de la entidad.

Recomendaciones

1-Extender el uso de las aplicaciones basadas en los servicios Web para otras funcionalidades dentro de la entidad.

2-Permitir el uso de la aplicación para el control integrado de la seguridad en otras aplicaciones empresariales fuera del marco de la entidad que le dio origen.

3-Fomentar el uso de aplicaciones distribuidas multiplataforma para la gestión de la información en el ambiente empresarial.

4-Profundizar en las ventajas del uso de los servicios Web como alternativas en software de control de la seguridad.

Bibliografía

1. -Web-services: la siguiente generación de Internet
<http://revista.robotiker.com/articulos/articulo62/página1.jsp>.
2. Uso de la seguridad basada en funciones con Web Services Enhancements 2.0 (artículos técnicos sobre Web Services Enhancements (WSE))
<http://www.microsoft.com/spanish/msdn/articulos/archivo/140305/voices/wserolebasedsec.asp>
3. Software libre http://es.wikipedia.org/wiki/Software_libre
4. El modelo de tres capas. http://www.lagash.com/papers/paper_threetier.html
5. Diseño de Aplicaciones Three tier. <http://www.fpress.com/revista/Num9711/Nov97.htm>

Glosario de términos

PHP (Personal Home Page) Es un lenguaje interpretado de alto nivel impregnado en páginas HTML y ejecutado en el servidor.

SGBD Sistemas de gestión de bases de datos.

Smarty Motor de plantillas para PHP.

SQL (Structured Query Language), Lenguaje de alto nivel, no procedural, normalizado, que permite la consulta y actualización de los datos de base de datos relacionales.

Anexo 1 Descripción expandida de los Casos de Uso

Caso de uso:	Configurar Funcionalidades y menú
Actores:	Administrador
Propósito:	Realizar la configuración de las funcionalidades de los módulos.
Resumen:	El caso de uso se inicia cuando se desean insertar, modificar o eliminar funcionalidades para un modulo determinado. Termina cuando se han insertado, modificado o eliminado dichas funcionalidades y la información está actualizada.
Precondiciones:	El Administrador debe estar validado, caso de uso: Validar usuario, que tiene una relación de inclusión.
Referencias:	R7
Poscondiciones:	Queda actualizada la información de las funcionalidades de un módulo.
Requerimientos Especiales:	

Interfases:

Opciones Administrativas

Elija opcion:

Configurar Sistema

Configurar Módulos

Configurar Funcionalidades y menú

Configurar Roles

Configurar Páginas

Para Usuarios:

Insertar Usuario

Actualizar Usuario

Para Grupos de Usuarios

Insertar Grupo

Modificar Grupo

Eliminar Grupo

Para Eventos a Auditar

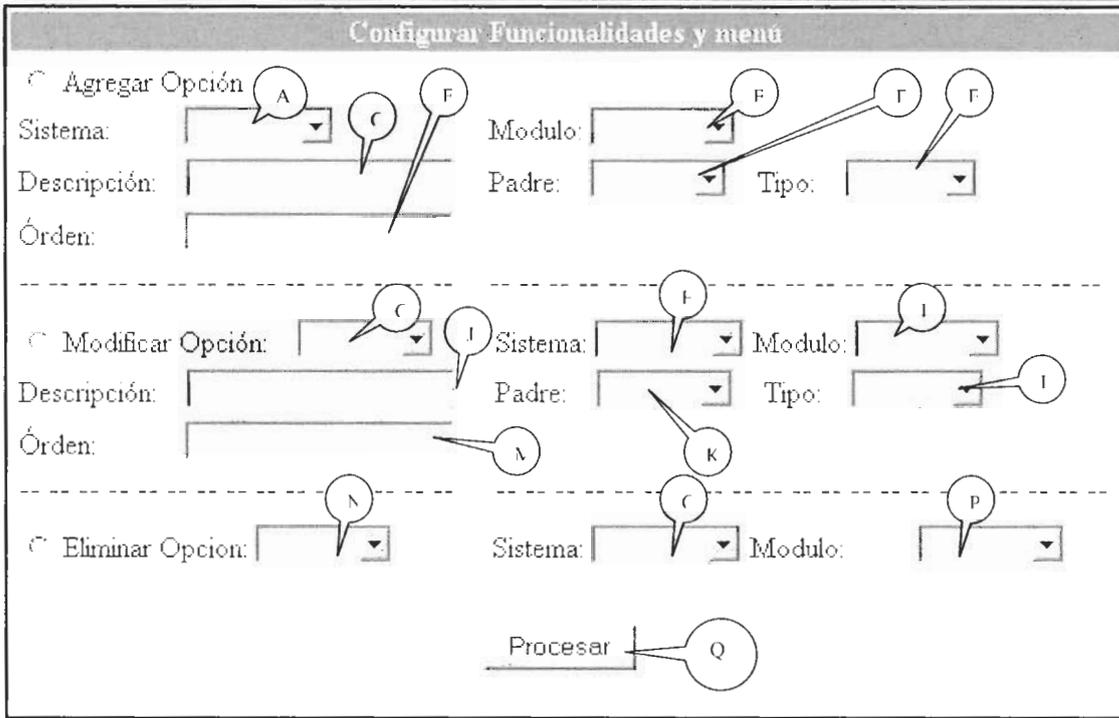
Insertar Evento a Auditar

Modificar Evento a Auditar

Eliminar Evento a Auditar

*

Configurar Funcionalidades



Curso normal de eventos para el caso de uso:

Acción del actor	Respuesta del sistema
<p>1-Selecciona la opción de Configurar Funcionalidades y menú.</p> <p>3-Selecciona:</p> <ul style="list-style-type: none"> • Agregar Opción • Modificar Opción • Eliminar Opción 	<p>2-Muestra interfaz de Configurar Funcionalidades y menú.</p> <p>4- Si:</p> <ul style="list-style-type: none"> • El usuario seleccionó Agregar Opción, ver sección: "Agregar Opción" • El usuario seleccionó Modificar Opción, ver sección: "Modificar Opción" • El usuario seleccionó Eliminar Opción, ver sección: "Eliminar Opción"

Sección Añadir opción

Acción del actor

1-Especifica el Sistema (A) y el módulo (B) al que pertenece la opción, luego escribe la opción en la descripción (C) y en caso de ser un submenú, debe especificar el padre en (D). En caso de no ser un submenú no debe llenar la opción padre. Luego especifica el tipo del menú, es decir, si es un menú, si es un botón, un vínculo o una funcionalidad del modulo en (E). En orden (F) se debe especificar la posición de la opción (orden en que aparece). Presiona luego el botón (Q).

Respuesta del sistema

2-Verifica integridad de la información

3-Procesa la información.

4-Muestra mensaje de operación realizada.

Sección: Modificar opción

Acción del actor

2-Especifica el sistema (H) y el modulo (I) y la opción (G) que desea modificar

4-Modifica el(los) dato(s) de el(los) campo(s) que desea actualizar. Presiona luego el botón (Q).

Respuesta del sistema

1-Busca el la BD los sistemas, módulos y funcionalidad existentes.

3-Carga en los campos: descripción (J), padre (K), tipo (L), orden (M), valores hasta el momento para dicha opción.

5-Verifica integridad de los datos.

6-Procesa información

7-Muestra mensaje de operación realizada.

Sección Eliminar opción

Acción del actor	Respuesta del sistema
1-Especifica Sistema (O), el Modulo (P) y la opción (N). Presiona el botón (Q).	2- Verifica la integridad de los datos. 3-Procesa la información. 4-Muestra mensaje de operación realizada.

Caso de uso:	Configurar Roles
Actores:	Administrador
Propósito: Almacenar la información referente a los roles que jugarán los usuarios dentro de un sistema determinado.	
Resumen: El Caso de Uso se inicia cuando el Administrador desea configurar un rol, el sistema brinda la posibilidad de insertar, modificar o eliminar, según lo que se seleccione el sistema procede mostrando las <i>interfaces</i> para la interacción con el administrador, el caso de uso termina cuando se actualizan las tablas de la base de datos.	
Precondiciones:	El Administrador debe estar validado, caso de uso: Validar usuario, que tiene una relación de inclusión.
Referencias:	R4
Poscondiciones:	Se actualiza en la BD la información referente a Roles.
Requerimientos	-
Especiales:	

Interfaces:

Pantalla: Opciones Administrativas

Elija opción:

Configurar Sistema

Configurar Módulos

Configurar Funcionalidades y menú

Configurar Roles

Configurar Páginas

Para Usuarios:

Insertar Usuario

Actualizar Usuario

Para Grupos de Usuarios

Insertar Grupo

Modificar Grupo

Eliminar Grupo

Para Eventos a Auditar

Insertar Evento a Auditar

Modificar Evento a Auditar

Eliminar Evento a Auditar

Pantalla: Configurar Roles

Configurar Roles

Agregar Rol Sistema:

Descripción:

Funcionalidades que accede:

- Agregar Usuario
- Agregar Módulo
- Agregar Sistema
- Insertar Usuario
- Eliminar Usuario

Páginas que accede:

- insertar_user.php
- index.php
- ag_sistema.php
- conf_711.php
- ver_bitacora.php

Modificar Rol: Sistema:

Descripción:

Funcionalidades que accede:

- Agregar Usuario
- Agregar Módulo
- Agregar Sistema
- Insertar Usuario
- Eliminar Usuario

Páginas que accede:

- insertar_user.php
- index.php
- ag_sistema.php
- conf_711.php
- ver_bitacora.php

Eliminar Rol: Sistema:

Curso normal de eventos para el caso de uso:

Acción del actor	Respuesta del sistema
1-Desea Configurar Roles 3-Selecciona Configurar Roles 5-Selecciona: Agregar Roles Modificar Roles Eliminar Roles Luego oprime el botón aceptar.	2-Muestra la interfaz de Opciones Administrativas 4-Muestra interfaz de Configurar Roles. 6-Si el usuario selecciona: Agregar Roles, ver sección "Agregar Roles"

	<p>Modificar Roles, ver sección "Modificar Roles"</p> <p>Eliminar Roles, ver sección "Eliminar Roles"</p>
Sección: Agregar Rol	
Acción del actor	Respuesta del sistema
1-Selecciona el sistema (A), especifica el nombre del rol (B), selecciona las funcionalidades (C) y la páginas (D) a las cuales tiene acceso y presiona el botón procesar (L).	<p>2- Verifica la integridad de la información.</p> <p>3-Procesa la información.</p> <p>4-Muestra mensaje de operación realizada</p>
Sección: Modificar Rol	
Acción del actor	Respuesta del sistema
1-De los valores modifica el que desee y deja los demás campos como están, presiona el botón procesar	<p>2- Verifica la integridad de la información.</p> <p>3-Procesa la información.</p> <p>4-Muestra mensaje de operación realizada</p>
Sección: Eliminar Rol	
Acción del actor	Respuesta del sistema
1-Selecciona el Sistema (K), el Rol a eliminar (J) y presiona el botón procesar (L).	<p>2-Verifica la integridad de los datos brindados por el usuario.</p> <p>3-Procesa los datos.</p> <p>4-Muestra mensaje de operación realizada.</p>
Cursos alternos:	
Sección: Agregar Rol	
Línea 2: Si existe algún problema en la información, muestra mensaje de error. Luego devuelve el control a la línea 1.	

Sección: Modificar Rol

Línea 2: Si existe algún problema en la información, muestra mensaje de error. Luego devuelve el control a la línea 1.

Sección: Eliminar Rol

Línea 2: Si existe algún problema en la información, muestra mensaje de error. Luego devuelve el control a la línea 1.

Caso de uso:	Configurar Grupos de Usuarios.
Actores:	Administrador del sistema(Inicia)
Propósito: Insertar, modificar o eliminar grupos de usuarios.	
Resumen: El caso de uso se inicia cuando el actor decide insertar, modificar o eliminar grupos de usuarios. En caso de insertar el sistema solicita la información para el nuevo grupo. Si se decide modificar el sistema permite seleccionar el grupo y muestra las características con posibilidad de ser modificadas. Si se opta por eliminar el sistema permite seleccionar el grupo. El caso de uso termina cuando el sistema inserta, modifica o elimina el grupo y se actualiza la tabla de los grupos de usuarios. Luego se pasa al caso de uso Registrar Evento.	
Precondiciones:	El Administrador debe estar validado, caso de uso: Validar usuario, que tiene una relación de inclusión.
Referencias:	R5
Poscondiciones:	Se inserta, modifica o elimina en la BD un grupo de usuarios.
Requerimientos Especiales:	-

Interfaz Opciones Administrativas

Elija opción:

- Configurar Sistema
- Configurar Módulos
- Configurar Funcionalidades y menú
- Configurar Roles
- Configurar Páginas

Para Usuarios:

- Insertar Usuario
- Actualizar Usuario

Para Grupos de Usuarios

- Insertar Grupo
- Modificar Grupo
- Eliminar Grupo

Para Eventos a Auditar

- Insertar Evento a Auditar
- Modificar Evento a Auditar
- Eliminar Evento a Auditar

Interfaz Conf. Grupos de Usuarios

Configurar Grupo de Usuarios

Agregar Grupo A B

Nombre: Sistema:

Roles que juega:

- Actualizador C
- Visualizador
- Administrativo
- Avanzado
- Operador de Copia
- Común

Modificar Grupo: D E

Nombre: Sistema:

Roles que juega: F

- Actualizador G
- Visualizador
- Administrativo
- Avanzado
- Operador de Copia
- Común

Eliminar Grupo: H I

J

Curso normal de eventos para el caso de uso:

Acción del actor	Respuesta del sistema
1. Desea Configurar Grupos de Usuarios.	2. Muestra Interfaz de Opciones Administrativas.
3. Selecciona Configurar Grupos de Usuarios.	4. Muestra Interfaz de configuración de Grupos de Usuarios consultando la BD y seleccionando los sistemas, grupos y roles existentes con sus

5. El administrador selecciona:

- Insertar Grupo de Usuarios.
- Modificar Grupo de Usuarios.
- Eliminar Grupo de Usuarios.

características.

6. si selecciona:

- Insertar Grupo de Usuarios ver sección "**Insertar Grupo de Usuarios**".
- Modificar Grupo de Usuarios ver sección "**Modificar Grupo de Usuarios**".
- Eliminar Grupo de Usuarios ver sección "**Eliminar Grupo de Usuarios**".

Sección: Insertar Grupo de Usuarios

Acción del actor

1. Especifica el Nombre (A) del grupo, el sistema (B) a que pertenece y los roles (C) que juega dentro de ese sistema. Luego presione Procesar.

Respuesta del sistema

2. El sistema verifica que no existan campos vacíos.
3. El sistema verifica que el grupo de usuarios no exista.
4. El sistema guarda la información del grupo de usuarios.
5. Muestra mensaje de Operación realizada.

Sección: Modificar Grupo de Usuarios

1. Selecciona el sistema (E) al que pertenece el grupo a modificar, selecciona el grupo (D), y especifica el nombre (F) y los roles (G) que juega. Luego presiona Procesar.

2. Verifica la integridad de la información e inserta los datos.
3. Muestra mensaje de Operación realizada.

Sección: Eliminar Grupo de Usuarios

1. Selecciona el sistema (I) al que pertenece el grupo a modificar y selecciona el grupo (H) y presiona Procesar.

2. Verifica la integridad de la información e inserta los datos.
3. Muestra mensaje de Operación Realizada

Cursos alternos:

Sección "Insertar Grupos de Usuarios"

Línea 2

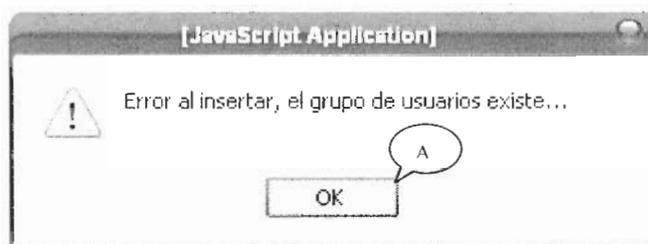
Si el sistema verifica que existen campos en blanco, entonces el sistema genera un mensaje de alerta, el administrador acepta en **A** del mensaje y el sistema devuelve el control a la *línea 1* de la sección "Insertar Grupos de Usuarios".



Mensaje de error datos en blanco

Línea 2

Si el sistema verifica que el Grupo de Usuarios existe, genera mensaje de error, el administrador acepta en **A** y el sistema devuelve el control a la *línea 1* de la sección "Insertar Grupos de Usuarios".



Mensaje de error de duplicado

Sección "Modificar Grupos de Usuarios"

Línea 2

Si el sistema verifica que faltan datos, entonces el sistema genera mensaje de alerta de datos en blanco, el administrador acepta en **A** y el sistema devuelve el control a la *línea 1* de la sección "Modificar Grupos de Usuarios".

Caso de uso:	Configurar Páginas
Actores:	Administrador
Propósito:	Almacenar toda la información de las páginas que conforman el sitio.
Resumen:	El caso de uso se inicia cuando el administrador desea incorporar toda la información de las páginas que conforman el sitio para el control de acceso a ellas.
Precondiciones:	El Administrador debe estar validado, caso de uso: Validar usuario, que tiene una relación de inclusión.
Referencias:	R2
Poscondiciones:	Se tendrá almacenada en la BD toda la información de las páginas que tiene el sitio.
Requerimientos Especiales:	

Interfase: Opciones Administrativas

Elija opción:

Configurar Sistema

Configurar Módulos

Configurar Funcionalidades y menú

Configurar Roles

Configurar Páginas

Para Usuarios:

Insertar Usuario

Actualizar Usuario

Para Grupos de Usuarios

Insertar Grupo

Modificar Grupo

Eliminar Grupo

Para Eventos a Auditar

Insertar Evento a Auditar

Modificar Evento a Auditar

Eliminar Evento a Auditar

Curso normal de eventos para el caso de uso:

Acción del actor	Respuesta del sistema
1. Desea configurar las páginas del sistema.	2. Muestra interfaz de Opciones Administrativas.
3. Selecciona Opción de configurar páginas	4. Muestra interfaz de Configurar Páginas.
5. El usuario selecciona: <ul style="list-style-type: none">• Agregar Página.• Modificar Página.• Eliminar Página. Y presiona aceptar.	6. El sistema decide si: <ul style="list-style-type: none">• Se selecciona Agregar Página, ver

sección: "Agregar Página".

- Se selecciona Modificar Página, ver sección: "Modificar Página".
- Se selecciona Eliminar Página, ver sección: "Eliminar Página".

Interfase: Configurar Páginas

Configurar Páginas

The screenshot shows a web interface titled "Configurar Páginas" with three sections separated by dashed lines:

- Agregar Página:** Includes a radio button, a "Nombre:" text input field (callout B), a "Número de Índice:" text input field (callout C), and a "Sistema:" dropdown menu (callout A).
- Modificar Página:** Includes a radio button, a "Nombre:" text input field (callout F), a "Número de Índice:" text input field (callout G), and a "Sistema:" dropdown menu (callout E).
- Eliminar Página:** Includes a radio button, a "Nombre:" text input field (callout H), a "Número de Índice:" text input field (callout I), a "Sistema:" dropdown menu (callout I), and a "Procesar" button (callout J).

Sección: Agregar página

Acción del actor

1-Selecciona el sistema (A), especifica el nombre de la página (B) (no la dirección http), un número de índice (C), que será el identificador de la página en el sitio y presiona el botón J.

Respuesta del sistema

2. Verifica que no existan errores en la información y guarda la información en la BD.
3. Muestra mensaje de operación realizada.

Sección: Modificar página**Acción del actor**

1-Selecciona el sistema (E), la página a modificar (D), especifica el nombre de la página (F) (no la dirección http), un número de índice (G), que será el identificador de la página en el sitio y presiona el botón J.

Respuesta del sistema

2. Verifica que no existan errores en la información y guarda la información en la BD.
3. Muestra mensaje de operación realizada.

Sección: Eliminar página**Acción del actor**

1-Selecciona el sistema (E), la página a eliminar (D), luego presiona el botón J.

Respuesta del sistema

2. Elimina la página.
3. Muestra mensaje de operación realizada.

Cursos alternos:**Sección: Agregar página**

Línea 2:

Si existen errores en los datos muestra mensaje de error. Retorna el control a la línea 1.

Sección: Modificar página

Línea 2:

Si existen errores en los datos muestra mensaje de error. Retorna el control a la línea 1.

Caso de uso:	Actualizar usuario
Actores:	Administrador del sistema(Inicia)
Propósito:	Que el administrador modifique o elimine los usuarios existentes.
Resumen:	El caso de uso se inicia cuando el administrador decide modificar o eliminar un usuario existente. Si se va a modificar el sistema muestra las características del usuario con posibilidad de ser modificadas, luego se actualizará el usuario. Si se va a eliminar se elimina el usuario seleccionado, el caso de uso termina cuando se actualiza la tabla de los usuarios. Luego se pasa a los casos de uso Registrar Evento y Crear Perfil. Si el usuario no es eliminado
Precondiciones:	El Administrador debe estar validado, caso de uso: Validar usuario, que tiene una relación de inclusión.
Referencias:	R1
Poscondiciones:	El administrador modifica o elimina usuarios.
Requerimientos Especiales:	
Interfaz Opciones Administrativas	

Elija opción:Configurar SistemaConfigurar MódulosConfigurar Funcionalidades y menúConfigurar RolesConfigurar Páginas**Para Usuarios:**Insertar UsuarioActualizar Usuario**Para Grupos de Usuarios**Insertar GrupoModificar GrupoEliminar Grupo**Para Eventos a Auditar**Insertar Evento a AuditarModificar Evento a AuditarEliminar Evento a Auditar**Curso normal de eventos para el caso de uso:****Acción del actor**

2. Selecciona Opción Modificar usuario.
4. Selecciona el sistema (A), el usuario (B) y:
Chequea la opción Cambiar Contraseña (C).
Chequea la opción eliminar (D).
No chequea ninguna opción.
- Y presiona el botón (P).

Respuesta del sistema

1. Muestra interfaz de opciones Administrativas.
3. Muestra Interfaz de Modificar y eliminar usuario.
5. decide si:
El usuario chequea la opción Cambiar Contraseña, ver sección "Cambiar Contraseña".
El usuario chequea la opción eliminar,

ver sección "Eliminar Usuario".

No chasquea ninguna opción, ver sección "Modificar usuario"

Interfaz Modificar _ eliminar Usuario

The screenshot shows a web form titled "Modificar Usuario". At the top, there are two radio buttons: "Cambiar contraseña" (callout C) and "Eliminar" (callout D). Below these are several input fields: "Nombre Completo:" (callout F), "Cargo:" (callout G), "Nombre de Usuario:" (callout I), "Confirmar Clave de Acceso:" (callout K), "Especialidad:" (callout F), "Grado:" (callout H), "Clave de Acceso:" (callout I), and "Grupo de Usuarios:" (callout I) which is a dropdown menu currently showing "Administradores". There is a section for "Permisos Especiales" with a list of checkboxes: "insertar_user.php" (callout M), "index.php", "ag_sistema.php", "conf_711.php", and "ver_bitacora.php". Below this are two checkboxes: "Acceso Múltiple" (callout N) and "Activo" (callout O). At the bottom center is a "Procesar" button (callout P). The top left has a "Sistema:" dropdown (callout A) and a "Usuario:" input field (callout B).

Sección: Modificar usuario

Acción del actor

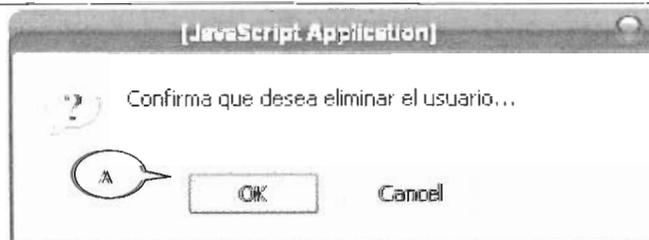
1. Especifica de los datos del formulario el que desee cambiar (desde E-O) excepto los campos de contraseña que serán desactivados, los demás campos no deben ser cambiados. Luego presiona el botón (P).

Respuesta del sistema

2. Verifica la información e inserta datos en la BD.
3. Muestra mensaje de operación realizada.

Sección: Eliminar usuario

Mensaje de confirmación



Acción del actor	Respuesta del sistema
1. Presiona el botón Procesar (P).	2. Elimina el usuario de la BD. 3. Muestra mensaje de Operación realizada.

Sección: Cambiar Contraseña:

Acción del actor	Respuesta del sistema
1. Especifica de los datos del formulario el que desee cambiar (desde E-O), los demás campos no deben ser cambiados. Luego presiona el botón (P).	2. Verifica la información e inserta datos en la BD. 3. Muestra mensaje de operación realizada.

Cursos alternos:



Mensaje de datos incompletos

Sección: Modificar Usuario.

Línea 2: Si falta algún dato muestra mensaje de error. Devuelve el control a la línea 1.

Sección: Cambiar Contraseña.

Línea 2: Si falta algún dato muestra mensaje de error. Si los campos de contraseña no coinciden muestra mensaje de error. Devuelve el control a la línea 1.

Caso de uso:	Crear perfil.
Actores:	Administrador del Sistema
Propósito:	Crear un perfil para un determinado usuario.
Resumen:	El caso de uso se inicia cuando el administrador inserta un nuevo usuario, a su vez se crea su perfil, esto va a ser un xml se utiliza para mandar toda la información del usuario a los otros módulos de servicios, se guarda como fichero donde se almacena la información de las funcionalidades de los módulos de servicio a las cuales el usuario puede acceder, en dependencia de su rol, finalmente se crea el perfil y termina el caso de uso.
Precondiciones:	El usuario debe haberse insertado y jugar un rol.
Referencias:	R1, R2
Poscondiciones:	Se almacena en un xml toda la información del perfil de usuario creado.
Requerimientos Especiales:	-
Curso normal de eventos para el caso de uso:	
Acción del actor	Respuesta del sistema
1. Una vez que inserta un usuario, se crea automáticamente su perfil (xml).	2. El sistema consulta los grupos de usuarios para ver que roles juega el usuario.
	3. Verifica las funcionalidades a las que puede acceder (Descripción: (Insertar, Usuario... etc.), orden: la posición en el menú, el padre: id del padre y el tipo: (menú, vinculo, funcionalidad, botón... etc).
	4. Almacena los datos correspondientes en un xml, que se guarda como fichero. 5. Crea el perfil del usuario.

Caso de uso:	Eliminar perfil.
Actores:	Administrador del Sistema
Propósito: Eliminar un perfil de un determinado usuario.	
Resumen: El caso de uso se inicia cuando el administrador decide eliminar un perfil de un determinado usuario el sistema muestra una ventana con todos los perfiles que hay creados, el administrador selecciona el solicitado y el sistema finalmente lo elimina.	
Precondiciones:	Se valida a un administrador para realizar esta acción.
Referencias:	R1
Poscondiciones:	Se actualiza el sistema.
Requerimientos Especiales:	-
Curso normal de eventos para el caso de uso:	
Acción del actor	Respuesta del sistema
1. El caso de uso se inicia cuando el actor decide eliminar perfil.	2. El sistema muestra una interfaz con todos los perfiles que hay creados.
3. El administrador busca el perfil a eliminar, lo selecciona.	
4. El administrador elimina el perfil de dicho usuario.	
	5. Elimina el xml

Caso de uso:	Configurar Eventos Auditables
Actores	Administrador
Propósito	Configurar todos los eventos auditables en cada uno de los sistemas.
Resumen: El caso de uso se inicia cuando el administrador desea configurar los	

eventos auditables de los sistemas, ya sea agregar, modificar o eliminar existentes. El caso de uso termina cuando se ha actualizado toda la información referente a los eventos auditables.

Precondiciones El Administrador debe estar validado, caso de uso: Validar usuario, que tiene una relación de inclusión.

Referencias R6

Poscondiciones Se actualiza toda la información referente a los eventos auditables en la BD.

Requerimientos especiales: -

Interfaz Opciones Administrativas

<p>Elija opción:</p> <p><u>Configurar Sistema</u></p> <p><u>Configurar Módulos</u></p> <p><u>Configurar Funcionalidades y menú</u></p> <p><u>Configurar Roles</u></p> <p><u>Configurar Páginas</u></p> <p>Para Usuarios:</p> <p><u>Insertar Usuario</u></p> <p><u>Actualizar Usuario</u></p> <p>Para Grupos de Usuarios</p> <p><u>Insertar Grupo</u></p> <p><u>Modificar Grupo</u></p> <p><u>Eliminar Grupo</u></p> <p>Para Eventos a Auditar</p> <p><u>Insertar Evento a Auditar</u></p> <p><u>Modificar Evento a Auditar</u></p> <p><u>Eliminar Evento a Auditar</u></p>
--

Curso normal de eventos

Acción del actor	Respuesta del sistema
1. Desea configurar Eventos Auditables	2. Muestra Interfaz de Opciones

3. Selecciona Configurar Eventos a auditar y presiona aceptar.

5. Selecciona:

- Agregar Evento
- Modificar Evento
- Eliminar Evento

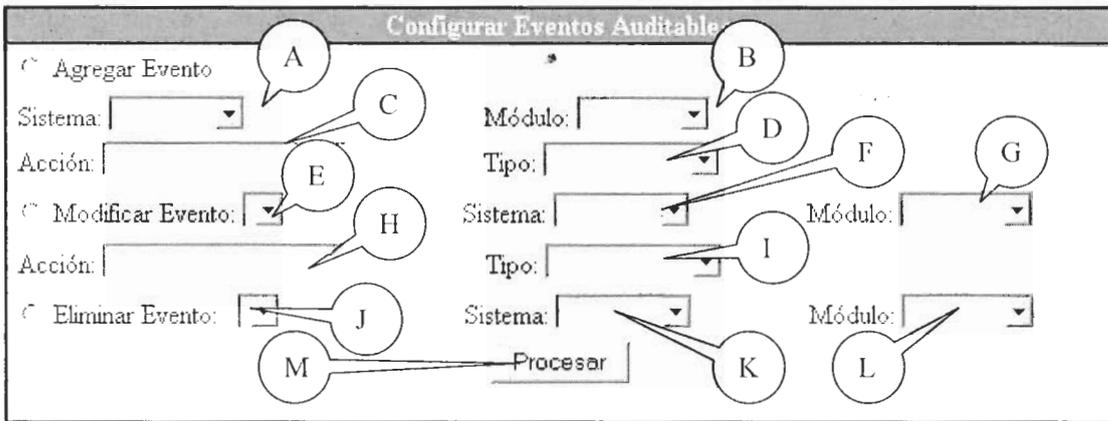
Administrativas

4. Muestra interfaz de Configurar Eventos.

6. Decide si el usuario selecciona:

- Agregar evento, ver sección "Agregar Evento".
- Modificar Evento, ver sección "Modificar Evento".
- Eliminar Evento, ver sección "Eliminar Evento".

Interfaz Configurar Eventos



Sección: Agregar Evento

Acción del actor	Respuesta del sistema
<p>1. Especifica el sistema (A), el módulo (B), la descripción de la acción (C), el tipo (E) y presiona el botón procesar (M).</p>	<p>2. Verifica la integridad de la información.</p> <p>3. Guarda los datos y muestra mensaje</p>

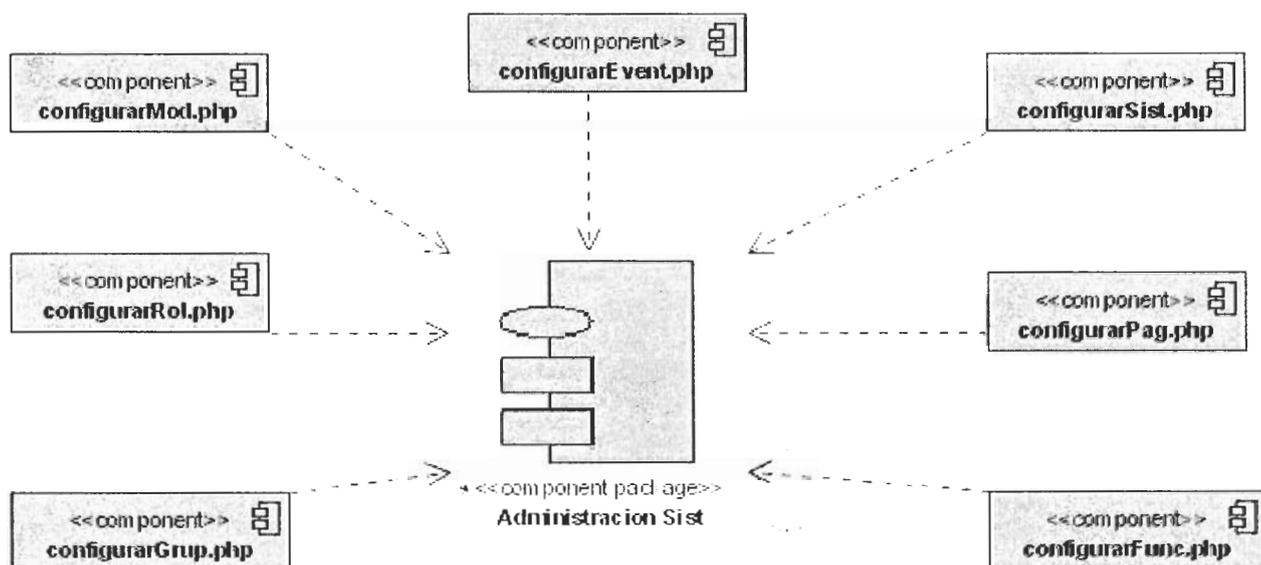
	de operación realizada.
Sección: Modificar Evento	
Acción del actor	Respuesta del sistema
1. Especifica el sistema (F), el módulo (G) y el evento a modificar (E), luego cambia de los datos H e I el que desee modificar y presiona el botón procesar (M).	2. Verifica la integridad de la información. 3. Guarda los datos y muestra mensaje de operación realizada.
Sección: Eliminar Evento	
Acción del actor	Respuesta del sistema
1. Especifica el sistema (K), el módulo (L) y el evento a eliminar (J), luego presiona el botón M.	2. Borra el evento seleccionado y muestra mensaje de operación realizada.
Cursos Alternos:	

Caso de uso:		Registrar Evento
Actores:	Generadores de sucesos (Inicia)	
Propósito:		
Registrar los eventos que tienen lugar en los sistemas y que están configurados para ser auditados.		
Resumen:		
El caso de uso se inicia cuando el generador de sucesos solicita registrar uno de los eventos configurados para ser auditados. El sistema guarda la información referente al usuario, el evento, la fecha y la hora en que se accedió. El caso de uso termina cuando se actualiza la tabla de Eventos.		
Precondiciones:	Un actor interactúa con un sistema generando un evento auditable.	
Referencias:	R8, R9	
Poscondiciones:	Se registra un Evento auditable.	
Curso normal de eventos para el caso de uso:		
Acción del actor	Respuesta del sistema	
1. El actor envía los datos para registrar la acción de uno de los eventos configurados para ser auditados.	2. Se valida el sistema y el evento. 3. El sistema guarda la información en la base de datos.	
Cursos alternos:		
<i>Línea 2</i>		
Si se verifica que el sistema o el evento no son validos concluye el caso de uso.		

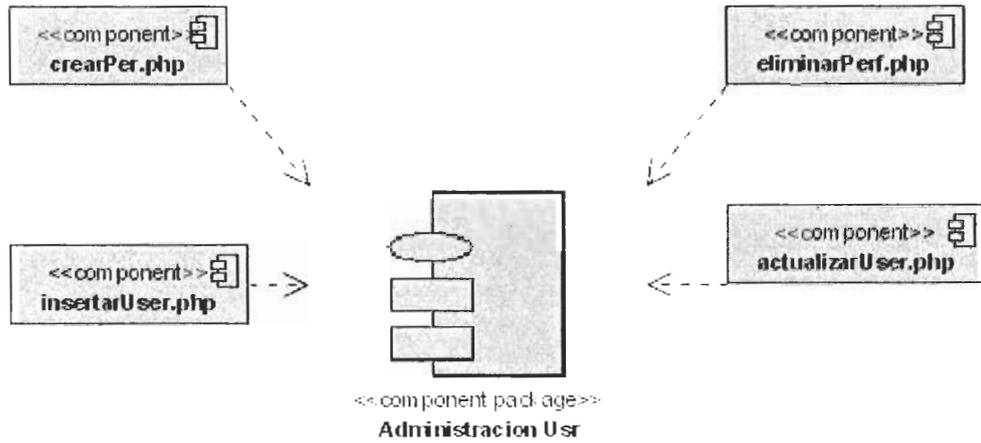
Anexo 2 Diagramas de Componentes

A continuación se muestran los diagramas de componentes por cada uno de los paquetes que forman el sistema:

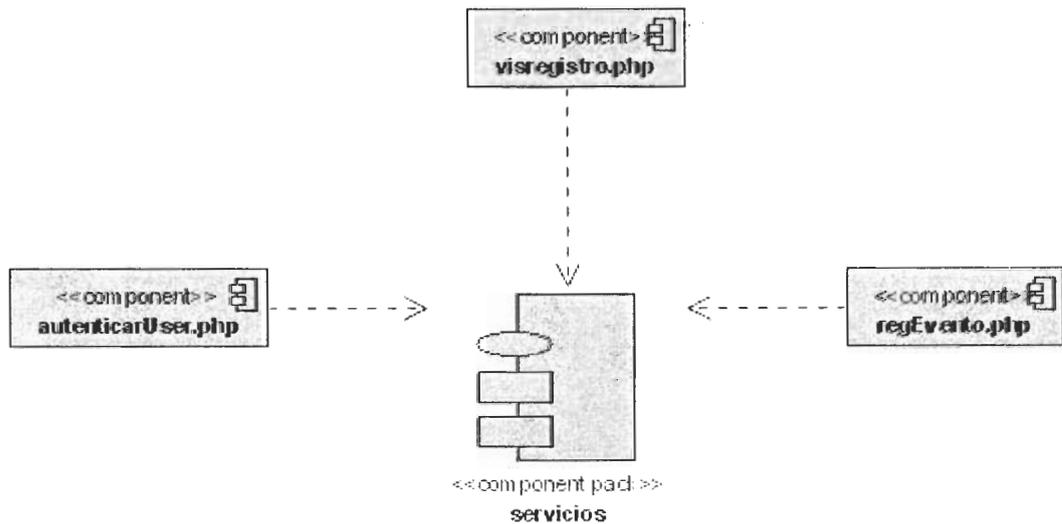
El Paquete de componentes de Administración de sistemas contiene las páginas relacionadas con las operaciones vinculadas a la configuración para el funcionamiento de control de la seguridad en un sistema determinado.



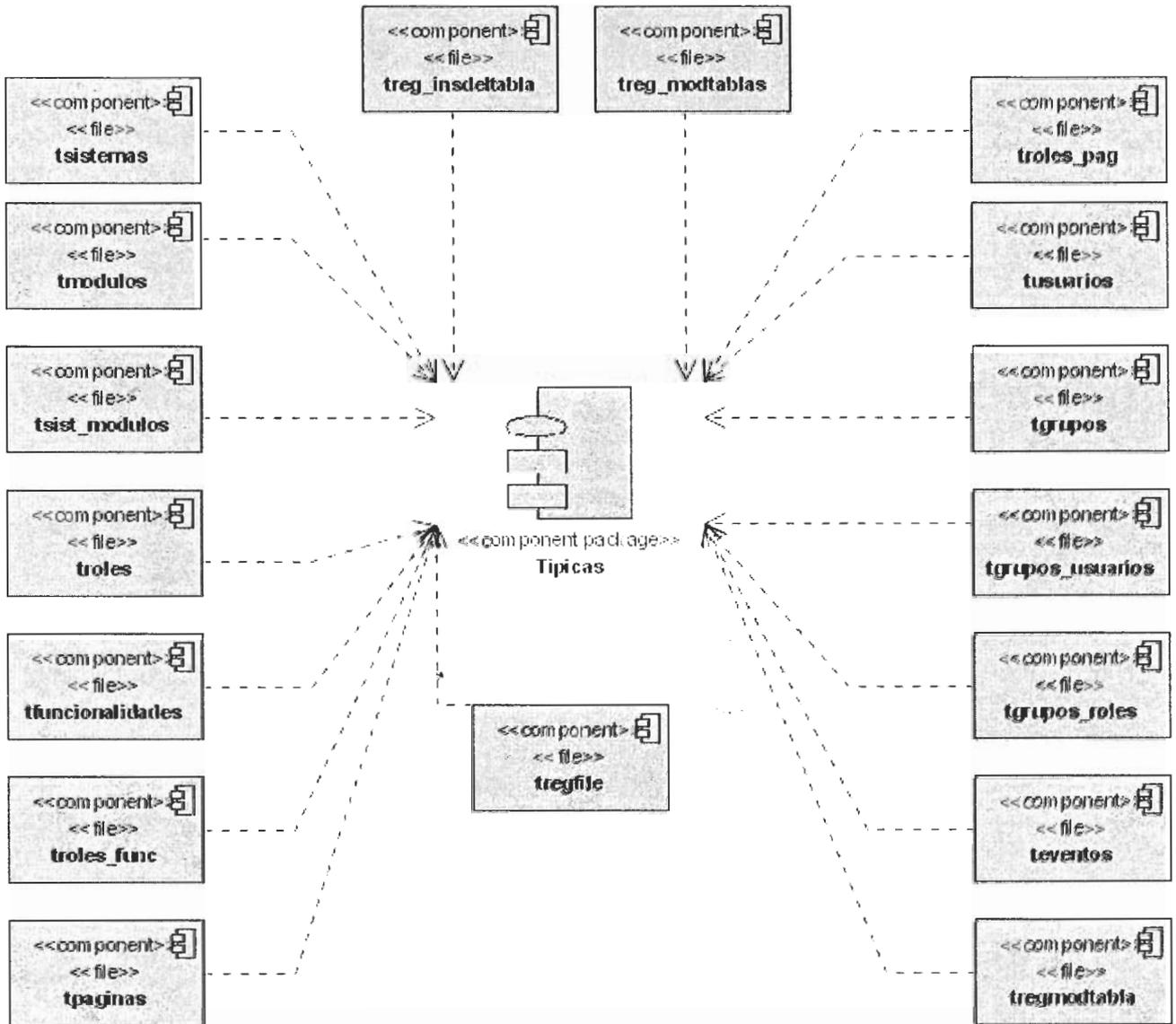
El paquete de Administración de Usuarios brinda la posibilidad de manejar toda la información referente a los usuarios que integran los sistemas, los roles que juegan, grupos a los que pertenecen, y a la vez administrar la información de estos últimos. Así se garantiza la operabilidad del sistema para el control estricto de la seguridad.



El paquete de Componentes de Opciones de Servicio Contiene el núcleo de las funcionalidades básicas del sistema, es mediante el cual los sistemas externos que usan el servicio se comunican con este sistema.

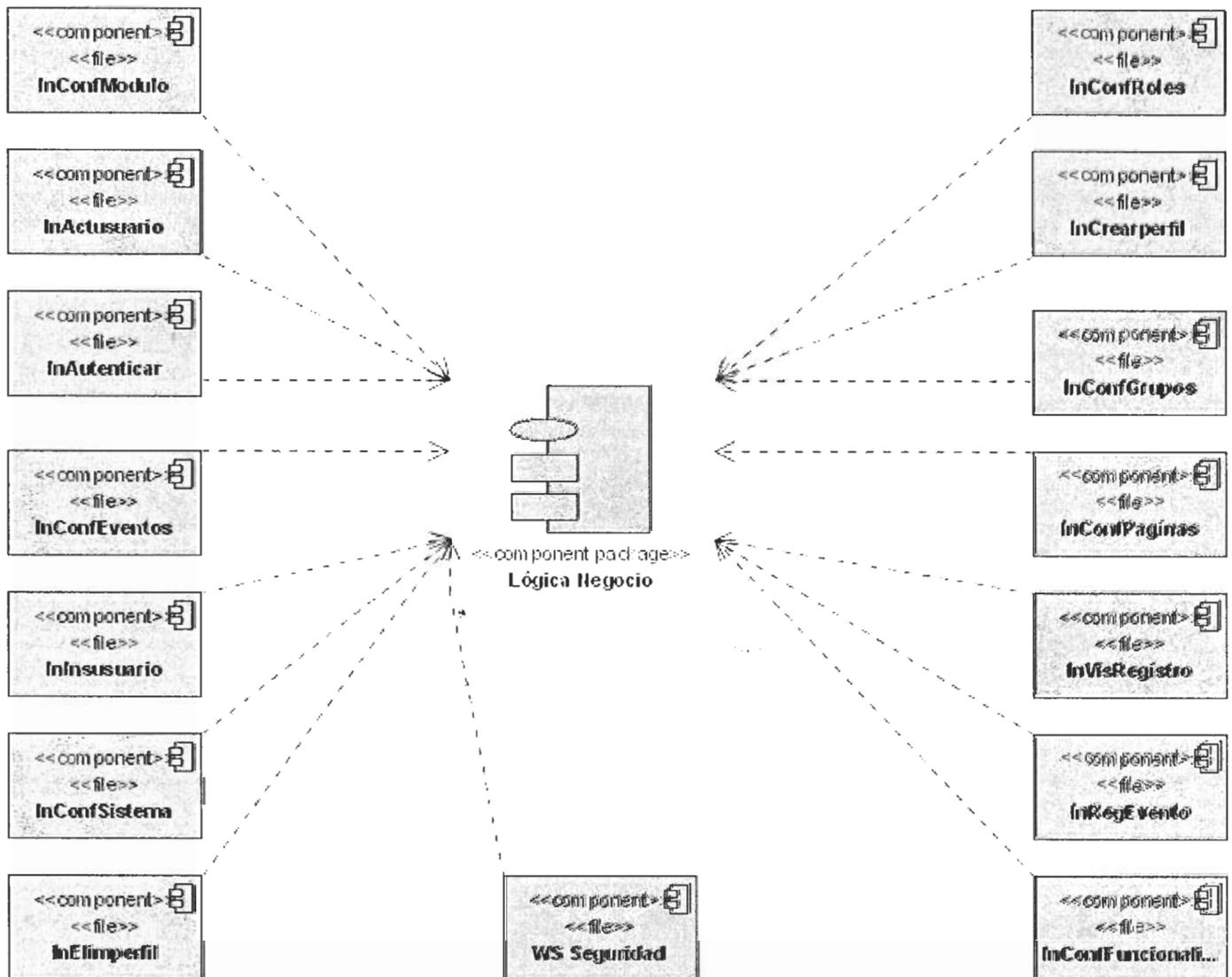


El diagrama de componentes de Típicas contiene todas las clases típicas relacionadas con cada una de las tablas de la Base de datos.

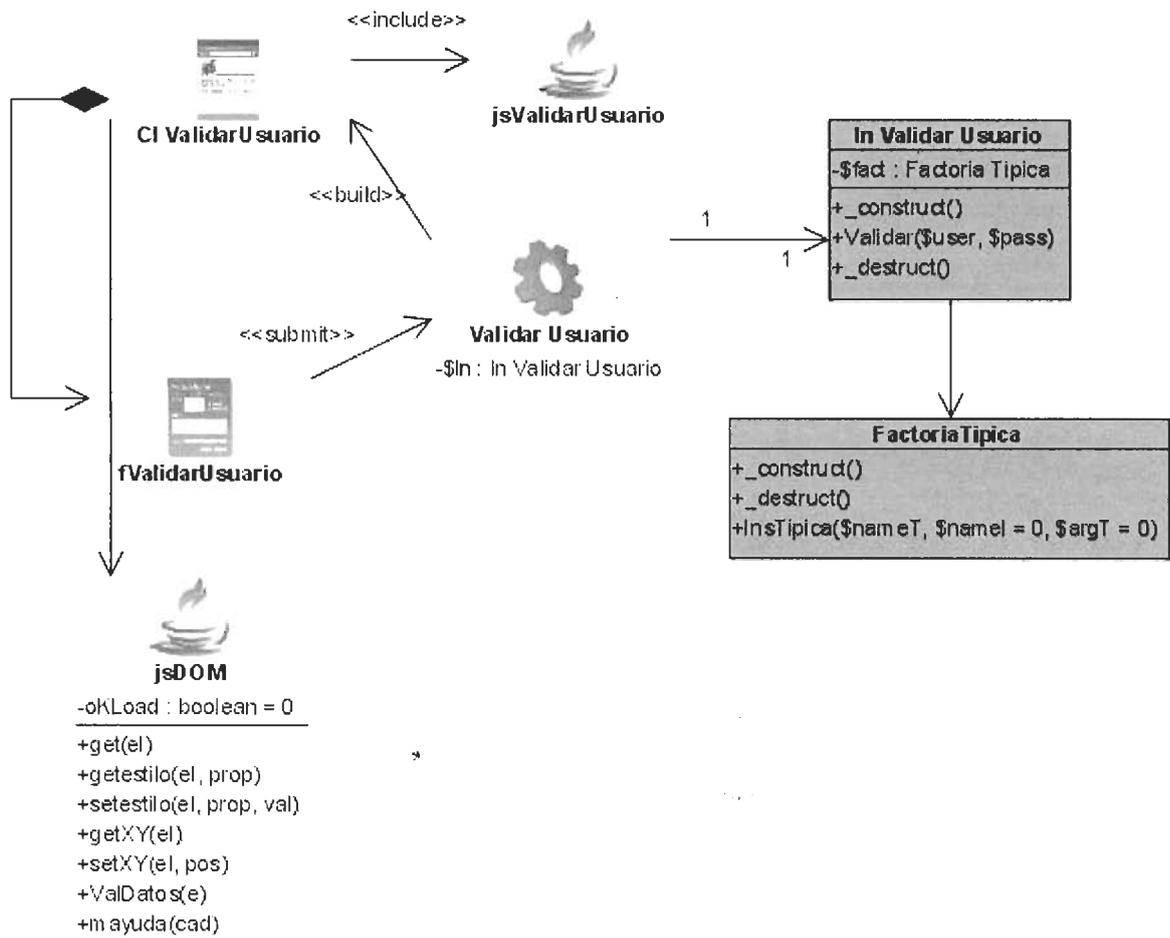


Paquete de Componentes Típicas

El paquete de componentes de lógica de negocio contiene los componentes de lógica de negocio, se muestra a continuación:



Anexo 3 Diagrama de Clases del Diseño Validar Usuario



Corrección figura 19

