Sistema de autenticación, autorización y auditoría (AAA) para aplicaciones basadas en servicios Web XML

Authentication, Authorization and Accounting (AAA) system to applications based on XML Web services

Annia Arencibia Morales, Karel Gomez Velázquez, Danisbel Rojas Rios, Hector Manuel Solis Mulet

Universidad de las Ciencias Informáticas

arencibia@uci.cu

Resumen

La investigación está encaminada a obtener un producto de software que proporcione eficaces procesos de Autenticación, Autorización y Auditoría (AAA) para aplicaciones basadas en Servicios Web XML. El mismo permite una gestión eficiente de usuarios, asignación de roles y privilegios para el acceso a todos los sistemas externos que consuman los servicios proporcionados por este. Por otra parte brinda un eficiente y óptimo proceso de auditoría y trazabilidad, de manera que se lleve un control estricto de las operaciones en que se involucran los usuarios de los sistemas.

Este sistema mediante los procesos de administración proporcionados, permite una gestión eficiente de todos los requerimientos de seguridad para aquellos sistemas externos que consuman sus servicios, lográndose de esta forma la reutilización del código y evitándose que se realicen acciones innecesarias fuera de cada negocio. Esto se logra mediante la publicación de Servicios Web XML debidamente descritos utilizando el Lenguaje de Descripción de Servicios Web (WSDL).

Su desarrollo está basado tecnologías libres, multiplataformas y sobre una arquitectura en capas, utilizando PHP 5 como lenguaje de programación mediante el ramework Symfony quien implementa el patrón de arquitectura Modelo Vista Controlador, PostgreSQL 8.3 como sistema de gestión de bases de datos, metodología AJAX para realizar más eficientemente las peticiones al servidor y la librería YUI para obtener una interfaz visual moderna. Utiliza estándares abiertos como XML lo permite la interoperabilidad entre las aplicaciones. Además utiliza el protocolo de transporte HTTPS que garantiza la confidencialidad de los datos.

Palabras clave: Auditoría, Autenticación, Autorización, Seguridad, Servicios.

Abstartc

The investigation is aimed at obtaining a software product that provides efficient processes of Authentication, Authorization and Audit (AAA) for applications based on XML Web Services. It allows efficient management of users, assigning roles and privileges for access to all external systems that consume services given by this. On the other hand, it provides an efficient and optimal audit and traceability process, so as to keep a strict control of operations which involve the systems users.

Through provided management processes this system allows efficient management of all security required for those external systems that consume their services. Thus, it achieves reusing the code and avoids unnecessary actions taking place outside the business. This is accomplished through the XML web services publication, adequately described, using the language description of Web Services (WSDL).

Its development is based on free technologies, multi-platforms and on a layered architecture, using PHP 5 as a programming language through framework Symfony, which implements the architecture pattern Model View Controller, PostgreSQL 8.3 system as a database management system, AJAX methodology which performs more efficiently requests to the server and the YUI library to get a modern visual interface. It uses open standards like XML allowing interoperability among applications and the transport protocol that permits HTTPS encryption of data.

Key Words: *Audit, Authentication, Authorization, Security, Services.*

Introducción

A través del Programa para Informatizar la Sociedad, el Estado Socialista Cubano, ha experimentado un incremento en la utilización de las tecnologías de la información en los últimos años. Con el fin de alcanzar un mejoramiento en la infraestructura tecnológica para satisfacer las necesidades de almacenamiento y acceso; la utilización de la información tanto en la esfera socioeconómica como política.

La informatización del Sistema Nacional de Salud (SNS) está apoyada en estrategias y políticas trazadas por la dirección del país y el MINSAP, siendo esta una tarea de vital y prioritaria importancia. Con este proceso, se pretende crear una infraestructura informática para el sector, al que se integrarán todos los productos o servicios, respondiendo a una arquitectura Orientada a Servicios y Basada en Componentes (SOA-CBA). Permitiendo que todas las unidades de salud del país alcancen un nivel de informatización elevado en las actividades que realicen, influyendo directamente en el aumento gradual de la eficiencia del personal de salud y en la calidad de los servicios que se brinden a la población

Actualmente la informatización del SNS no ofrece un mecanismo único para la integración. Las instituciones de Salud Pública poseen un conjunto de aplicaciones que brindan solución a determinados problemas, pero estos se comportan como islas de información al no poder interactuar entre si para obtener un flujo lógico y coherente de la información clínica relacionada con los pacientes.

En este sentido, el Problema a resolver por el Sistema de Seguridad es: ¿Cómo fortalecer los procesos de Autenticación, Autorización y Auditoría en los productos desarrollados para la informatización del Sistema Nacional de Salud cubano?

Desarrollo

Objeto de estudio: Proceso de informatización del Sistema Nacional de Salud cubano.

Campo de acción: Proceso de gestión de requerimientos de seguridad en los productos desarrollados para la informatización del Sistema Nacional de Salud cubano.

Para resolver el problema identificado se propone el siguiente

Objetivo general: Desarrollar un Sistema de Seguridad que estandarice los procesos de Autenticación, Autorización y Auditoría en los productos desarrollados para la informatización del Sistema Nacional de Salud cubano.

Metodología

Un sistema o componente con el objetivo de gestionar los requerimientos de seguridad en aplicaciones Web, debe apoyarse en algunos elementos esenciales con el fin de lograr el propósito de su implementación. Estos elementos consisten básicamente en:

El control de acceso de los diferentes usuarios a las aplicaciones, el cual constituye una poderosa herramienta para proteger la entrada a un sistema completo o sólo a ciertos directorios concretos e incluso a ficheros o programas individuales; este control consta generalmente de dos pasos:

Autenticación: es el proceso de verificación de la identidad digital de un remitente de una comunicación que hace una petición para conectarse a un sistema. El remitente puede ser una persona que usa un ordenador u otro medio electrónico, un ordenador por sí mismo o un programa. En otras palabras es un modo de asegurar que los usuarios son realmente quienes dicen ser y que tienen la autorización para realizar funciones en el sistema.

Autorización: proceso por el cual se autoriza al usuario identificado a acceder a determinados recursos del sistema, es decir se comprueba que los usuarios con identidad válida solo tengan acceso a aquellos recursos sobre los cuales tienen privilegios.

Precisamente teniendo en cuenta estas razones el Sistema de Seguridad implementa una fuerte política de Auditoría gracias a la cual quedan registrados todos los accesos y peticiones realizadas por los usuarios, quedando siempre almacenados un conjunto de datos como: usuario, servicio que consume, componente y dirección IP desde el cual accede, fecha, hora, tipo de traza que genera

y una descripción que permite aumentar el nivel de detalles acerca de las acciones de los usuarios, contribuyendo de esta forma a facilitar el proceso de análisis de las trazas.

También es posible realizar búsquedas avanzadas de las mismas. Para ello el sistema brinda la posibilidad de utilizar varios parámetros para ir filtrando la información, estos son: nombre del organismo al cual pertenece el usuario, nombre del componente, usuario, tipos de traza, periodo de tiempo y rangos de direcciones IP, con el objetivo de hacer más flexible y eficiente la búsqueda. Otra funcionalidad es la creación de reportes en formato PDF que permite imprimir la información y facilita el proceso de auditoría del sistema, aún cuando no se dispone de un ordenador.

Las bases de datos de trazas tienden a crecer con mucha rapidez. Como una vía alternativa para solventar esta realidad, el sistema brinda la posibilidad de eliminar las trazas de los usuarios. Cuando se elimina una, pasa a formar parte de otra base de datos donde se registra con el mismo formato pero bajo la categoría de traza histórica. En caso de ser eliminada, nunca se pierde el control sobre la misma, debido a que es guardada en un fichero en el servidor, posibilitando su persistencia futura mediante el uso de dispositivos de almacenamiento externo (CD, DVD o Discos extraíbles). Para complementar este proceso, el Sistema de Seguridad permite recuperar las trazas que una vez fueron eliminadas y que constituyen una porción de información importante en un momento determinado.

Materiales y métodos

Patrones de arquitectura y diseño

Los patrones arquitectónicos y de diseño utilizados para el desarrollo del Sistema de Seguridad son: Modelo-Vista-Controlador (MVC), Arquitectura en tres capas, Arquitectura Orientada a Servicios y Basada en Componentes.

Tecnología Servicios Web XML

Teniendo en cuenta la heterogeneidad tecnológica y la diferente distribución física de los sistemas se definió implementar la arquitectura SOA mediante Servicios Web XML. Los Servicios Web usan SOAP (Simple Object Access Protocol) como protocolo para invocar llamadas remotas debido a su simplicidad, se puede identificar un mensaje SOAP como un documento XML conformado por una envoltura obligatoria, un encabezamiento opcional y un cuerpo también obligatorio. Este permite la comunicación entre aplicaciones heterogéneas, de modo que clientes de diferentes plataformas o lenguajes de programación pueden comunicarse entre sí de manera satisfactoria.

Alrededor de los Servicios Web existen protocolos y mecanismos adicionales para facilitar tareas como el descubrimiento de servicios distribuidos a lo largo de la red o UDDI (Universal Description, Discovery and Integration), una descripción del contenido de los mensajes o WSLD (Web Services Description Language) el cual describe la forma de comunicación, es decir, los requerimientos del protocolo y los formatos de los mensajes necesarios para interactuar con los servicios listados en su catálogo. Además se utiliza el protocolo http (Hypertext Transfer Protocol) para el transporte de la mensajería, utilizando el puerto 80 ya que el mismo siempre se encuentra accesible.

Single Sign On (SSO)

Los usuarios para identificarse ante varias aplicaciones informáticas son forzados a recordar numerosas contraseñas por lo que en la mayoría de los casos eligen contraseñas sencillas poniendo potencialmente en riesgo la seguridad del sistema. La utilización de una arquitectura SOA tiene como objetivo de dar acceso a los usuarios a múltiples servicios Web y/o aplicaciones. En la mayoría de los casos se encuentra que cada uno de los servicios o aplicaciones cuenta con su propio componente o mecanismo de seguridad, lo que puede comprometer la seguridad de todo el sistema. El nivel de seguridad de todo un sistema es igual al nivel de seguridad del componente más inseguro que lo compone.

No es más que es un proceso de autenticación de sesión/usuario; que permite mediante un solo conjunto de credenciales acceder a diferentes aplicaciones, esto es principalmente utilizado en ambientes distribuidos. El proceso autentica al usuario para todas las aplicaciones a los que les han dado derechos y elimina todos los mensajes de autenticación que se generan cuando se cambia de interfaz durante una sesión particular.

Este es un elemento de mucha importancia que se tuvo en cuenta en el proceso de desarrollo del Sistema de Seguridad; permitiendo el encapsulamiento de la infraestructura de seguridad subyacente, posibilitando que los procesos de implementación, despliegue y mantenimiento sean más fáciles; ninguna de las partes comunicantes en el sistema distribuido, necesita implementar individualmente todos los mecanismos de seguridad. Este tipo de proceso permite a los desarrolladores y organizaciones centrarse en el desarrollo de la lógica de negocio asociada a un componente en particular, obviando los mecanismos de seguridad, debido a estos serán proporcionados de manera automática por el Sistema de Seguridad en el momento de la integración y el despliegue.

Lenguajes utilizados

El sistema se desarrolló utilizando el lenguaje de script PHP 5.2.5, particularmente mediante el ramework Symfony 1.0.8. Para garantizar la validación de los datos del lado del cliente javascript. Así como que para obtener una interfaz visual moderna y de utilización intuitiva la librería de componentes visuales Yahoo User Interface YUI 2.5.0, la cual se integra fácilmente con el conjunto de tecnologías AJAX garantizando de esta manera una rapidez en la obtención de las respuestas originadas desde el servidor.

Otros elementos utilizados

Teniendo en cuenta el volumen de datos que se genera y manipula por un sistema como este se utilizó como sistema de gestión de bases de datos PostgreSQL 8.3. Uno de los elementos más significativos en el desarrollo de este sistema es que está totalmente desarrollado sobre tecnologías no propietarias donde los componentes reutilizados poseen licencia de software BSD, garantizándose que se pueda implantar en cualquier entorno sin costo alguno. El proceso de desarrollo estuvo basado en la metodología RUP y los artefactos generados, fueron modelados visualmente mediante UML 2.0 en la herramienta CASE Enterprise Architect 7.

Resultados

El Sistema de Seguridad contiene un conjunto de definiciones identificadas, las cuales permiten un mayor entendimiento del negocio del mismo, siendo estas:

Administrador General: Actor que en dependencia de su nivel de acceso sobre un organismo cuyos requerimientos de seguridad son gestionados por el Sistema de Seguridad, tiene los permisos necesarios para gestionar la información de los usuarios que en la estructura jerárquica de niveles defina por su organismo, posean un nivel igual o inferior al suyo. Esta gestión incluye crear usuarios, modificar sus datos y privilegios, eliminarlos y realizar búsquedas a partir de diferentes parámetros. Por otra parte puede realizar una auditoría estricta a través de las trazas almacenadas por el sistema, conociendo qué usuario ha participado en cada transacción.

Administrador Configuración: Actor que tiene permiso total sobre la configuración del sistema. Único encargado de la gestión de organismos, niveles, ubicaciones, componentes, servicios y roles.

Usuario: Actor que interactúa directamente con el Sistema de Seguridad una vez que se hayan superado las fases de desarrollo correspondientes, con el objeto de consultar, modificar o eliminar la información gestionada por el mismo.

Componente: Es una unidad ejecutable que representa el núcleo de la aplicación, la cual puede ser implantada independientemente

y ser a la vez sujeto de composición de terceras partes, es decir, se puede tomar el componente y agregarlo a otro componente en desarrollo o simplemente consumir algunos de los servicios que brinda.

Certificado: Conjunto de datos y privilegios de un usuario determinado que se crea de forma automática durante los procesos de autenticación y autorización. El mismo posee un identificador único de treinta y dos caracteres que se genera de manera aleatoria, contiene el nivel de acceso del usuario, el identificador del nivel de acceso y un listado de los componentes a los que tiene derecho de acceso, así como los privilegios de ejecución correspondientes en estos componentes.

Nivel de Actividad: Clasificación del usuario que permite diferenciar entre un usuario activo (que está autorizado a utilizar el sistema dentro de un período de tiempo determinado) y uno de tipo inactivo (contrarresta la definición anterior).

Traza: Historial donde se almacenan todos los eventos realizados por el usuario al interactuar con la información perteneciente al sistema.

Rol: Papel que cumple un usuario dentro de un componente y que limita el conjunto de funcionalidades que puede desempeñar en ese ámbito dentro del sistema.

Servicio: Operación proporcionada por un componente determinado.

Organismo: Conjunto de dependencias, oficinas o empleos que cumplen con determinadas leyes, usos y costumbres y que a la vez forman una institución social.

Nivel: Concepto asociado a las diferentes instancias de dirección administrativa de un organismo determinado.

Ubicación: Recoge la ubicación exacta del usuario en cada uno de los niveles en que se desempeñe.

Grupo de Nivel: Especifica los diferentes niveles que tiene cada uno de los organismos.

Grupo de Rol: Se recogen todos los roles que pertenecen a cada organismo.

Período de Actividad: Es el tiempo durante el cual el usuario va a estar en estado activo dentro de la aplicación.

Historial de Traza: Se especifica los tipos de trazas definidos que pueden dejar los usuarios.

Discusión

Seguridad en las comunicaciones: Desde la concepción inicial del Sistema de Seguridad se identificó la necesidad de contar con un canal seguro de comunicación, debido a la importancia de la información que el mismo gestiona y a las consecuencias nefastas que se podrían originar en caso de no tomar las precauciones necesarias para garantizar la seguridad de la información que viaja entre el servidor Web y el usuario. En caso de no tener presente este aspecto, el sistema podría ser victima de un ataque electrónico que permitiría interceptar el contenido de las comunicaciones TCP/IP comprometiendo la seguridad de las aplicaciones Web que utilicen los servicios que brinda el Sistema de Seguridad.

Por esta razón es necesaria la utilización de protocolos seguros de comunicación como el HTTPS. Este protocolo no forma parte del proceso de elaboración propio del Sistema de Seguridad, sino que debe configurarse en el servidor Web donde va a ser instalado el sistema una vez que haya rebasado sus fases de construcción.

Integración con servidores LDAP: Generalmente los organismos o instituciones tienen su propia red informática, para la comunicación de los diferentes servicios que pudieran brindar o consumir, así como sus propios usuarios ya establecidos con una estructura jerárquica definida. Para acceder a los datos de los usuarios utilizan un servidor LDAP, que puede ser utilizado desde

distintas plataformas, debido a que su implementación esta basada en estándares internacionales y hace que los procesos de búsqueda, y de autenticación sean mucho mas rápidos y eficientes que un SGBD convencional.

Teniendo en cuenta estas características, el Sistema de Seguridad, para la gestión de la información de sus usuarios, brinda de forma adicional la posibilidad de conectarse a servidores LDAP, posibilitando la reutilización de la información y beneficiándose de las ventajas que proporciona el uso del protocolo LDAP.

Para la conexión a servidores LDAP, se especifican una serie de parámetros de entrada, que permiten la comunicación de manera estándar, independientemente del lugar y estructura del servidor que provee la información. Estos parámetros son:

- Dirección IP donde se encuentra el servidor.
- Puerto por el cual se va a establecer la conexión.
- Versión del LDAP.
- Base DN (Nombre Distinguido), estructura en forma de árbol jerárquico que define la concatenación de los DNS (Sistema de Nombres de Dominio) relativos de las entradas "padre" hasta llegar a la entrada "raíz" del árbol. Un ejemplo de DN pudiera ser (cn=Pedro Pérez, ou= UCI Domain Users, o=uci ,c=cu)
- Usuario y Contraseña registrados en el LDAP para establecer la conexión.
- Filtro de conexión: filtro que permite restringir la búsqueda de una persona en el directorio.

Luego de haber establecido la conexión con el servidor LDAP, se obtienen una serie de parámetros de salida estándares, lo cuales pueden ser utilizados en dependencia de las necesidades del usuario o sistema que consulta el Servidor LDAP. Algunos de estos parámetros son:

- givenname: Devuelve el nombre del usuario.
- sn: Apellidos del usuario.
- cn: Nombre completo del usuario.
- mail: Dirección de correo electrónico del usuario.

Mailnickname: Nombre de usuario

Publicación de Servicios Web: La implementación de Servicios Web juega un papel protagónico en el Sistema de Seguridad, debido a que brinda la posibilidad a componentes externos, de consumir algunas de sus principales funcionalidades independientemente de la plataforma o lenguaje en el que hayan sido desarrollados.

Para publicar un Servicio Web se agrupan las funcionalidades que se deseen brindar como servicios en una o varias clases contenedoras, posteriormente se especifica la descripción para cada tipo de datos, tanto de entrada como de salida de cada función. Luego utilizando el IDE de Desarrollo para PHP, ZendStudio, se especifica la dirección del fichero donde se encuentran la clases contenedoras, a continuación se selecciona la clase que contiene las funciones que se desean publicar y finalmente se describe el servicio en un fichero .wsdl que es la interfaz entre el Servicio Web y los sistemas o clientes que necesitan consumir alguno de los servicios publicados.

El Sistema de Seguridad brinda los Servicios Web: Autenticar, Autorizar, Búsqueda de Usuarios, Búsqueda de Componentes, Adicionar Traza, Búsqueda de Traza. Seguidamente se muestran a modo de ejemplo los datos del servicio Autenticar, entre los que se incluyen: una breve descripción, parámetros de entrada y salida, ejemplos sobre cómo consumirlos así como la estructura de la información que devuelven.

Descripción General: Este servicio es el encargado de verificar la identidad digital de un usuario que intenta acceder a un sistema cuyos requerimientos de seguridad son gestionados por el Componente de Seguridad.	
PARÁMETROS DE ENTRADA	
Descripción	**Tipo**
Usuario	String
Contraseña	String
PARÁMETROS DE SALIDA	
Descripción	**Tipo**
IDUsuario	String
Usuario	String
Nombre	String
Correo electrónico	String
Certificado	String
Lista derechos	Array

Tabla 1. Nombre del Servicio Web: *Autenticar*

```
try
{
    $cliente = new SoapClient("http://localhost:5800/Seguridad/modelo/wsdl/servicios_usuario.wsdl");
    $arg=array("user"=>"root","pass"=>"1234567");
    $r= $cliente->__call("autenticar",$arg);
    print_r ($r);
}
catch (SoapFault  $f)
{
    print_r($f);
}
```

Fig. 1. Ejemplo de implementación para consumir el Servicio Web: *Autenticar*.

```
[idusuario] => 0_158hmsolis
[usuario] => hmsolis
[nombre] => Hector Manuel Solis Mulet
[correo] => hmsolis@estudiantes.uci.cu
[certificado] => a4cedf64b4debf70f72b027f934a1b03
[ListaDerechos] => Array
    (
        [3] => stdClass Object
            (
                [modulo] => 113
                [nombremodulo] => docencia
                [url] =>
                [ListaRoles] => Array
                    (
                        [0] => stdClass Object
                            (
                                [rol] => 158
                                [nombrerol] => visualizador
                                [ListaNiveles] => Array
                                    (
                                        [0] => stdClass Object
                                            (
                                                [nivel] => 201
                                                [nombrenivel] => Nacional
                                                [idnivel] =>
                                                [ListaServicios] => Array
                                                    (
                                                        [0] => stdClass Object
                                                            (
                                                                [servicio] => 270
                                                                [nombreservicio] => dar baja estudiante
                                                            )
                                                    )
                                            }
                                    }
                            }
                    }
            }
    }
}
```

Fig. 2. Estructura de la información que devuelve el Servicio Web: *Autenticar*.

Fig. 2. Página de Autenticación del Sistema de Autenticación, Autorización y Auditoría.

En sentido general se puede destacar que el desarrollo del Sistema de Seguridad proporciona un grupo de beneficios para cualquier aplicación que utilice la arquitectura anteriormente descrita. Entre ellos se pueden mencionar los siguientes:

- Gestión eficiente de requerimientos de seguridad para todos los sistemas informáticos que consuman sus servicios de Autenticación, Autorización y Auditoría. Proporciona facilidades de mantenimiento a estos sistemas y permite la agilización del proceso de construcción de las aplicaciones. Los desarrolladores podrán obviar los mecanismos de seguridad que serán proporcionados de manera automática por el Sistema de Seguridad en el momento de la integración.

- Aumento de los niveles de integración entre los diferentes sistemas, evitando que cada uno posea de manera aislada la administración de sus usuarios, esta información será almacenada y gestionada centralizadamente. De igual manera, permitirá organizar los módulos por grupos permitiendo la unificación de componentes heterogéneos de acuerdo a su negocio.

- Perfeccionamiento de los procesos de gestión de usuarios y asignación de privilegios.

- Gestión eficiente del proceso de Auditoría de los productos integrados a él, permitiendo hacer reportes de las trazas históricas de acuerdo a diferentes parámetros y controlando la acumulación de información de esta naturaleza en las bases de datos.

- Permite la integración con servidores LDAP, posibilitando la reutilización de los datos de usuarios de un organismo determinado, evitando de esta forma la duplicación de la información en la base de datos.

- Está desarrollado en forma de producto. Su funcionamiento no está limitado algún ambiente de desarrollo en específico, sino que puede ser perfectamente escalable ante cualquier conjunto de tecnologías y requerimientos de software.

Conclusiones

1. Ha sido realizado un estudio de las tendencias y tecnologías actuales para el desarrollo del sistema, implementado mediante una arquitectura Orientada a Servicios y Basada en Componentes.
2. Se han analizado los procesos de negocio actuales de Autenticación, Autorización y Auditoria (AAA) de varios sistemas ya desarrollados y con algún tiempo de explotación.
3. Se desarrollaron los Servicios Web XML públicos: Autenticar, Autorizar, Adicionar Traza, Buscar Traza, Búsqueda de Usuarios, Búsqueda de Componentes para los componentes desarrollados por el Área Temática con sus respectivas descripciones (WSDL).
4. La información gestionada será objeto de cuidadosa protección contra estados corruptos e inconsistentes, teniendo acceso a ella solo el personal autorizado.

Referencias Bibliográficas

- *Ataques más comunes sobre aplicaciones Web*, 2 de mayo de 2006. Disponible en: http://www.hispasec.com/corporate/noticias/94

- Aruquipa Chambi Marcelo G., Márquez Granado Edwin P. *Desarrollo de Software Basado en Componentes*. Universidad

-Espinosa Eduardo. *Seguridad en la Web. Ing. En Computación*. Disponible en: http://www.espina.info/papers/seguridadenlaWeb.pdf.

- Gudiño Fleites, Pedro. Tutorial de Sistemas Distribuidos I. Departamento de Sistemas y Computación. Instituto Tecnológico de Colima. México. 2004. Disponible en: http://www.itcolima.edu.mx/profesores/tutoriales/sistemas_distribuidos_I/sd_u1_1.htm

- Jacobson, Ivar; Booch, Grady; Rumbaugh, James. *El Proceso Unificado de Desarrollo de Software*. La Habana. Cuba. Editorial Félix Varela. 2004. Pág. 4, 5, 6 y 7

- Mayor de San Andrés. La Paz Bolivia, agosto de 2007. Disponible en:http://pgi.umsa.bo/enlaces/investigacion/pdf/INGSW3_23.pdf

- Navarro Franco, Ángel José. *UML en acción. Modelando Aplicaciones Web*. Instituto Superior Politécnico José Antonio Echeverría. La Habana, Cuba, Mayo 2005.

- Pressman, RS. *Ingeniería de Software, un enfoque práctico. Parte 1*. Roger Pressman, La Habana, Cuba: Editorial Félix Varela. 2005.

- *¿Qué es LDAP?*, 10 de diciembre de 2004. Disponible en: http://www.ldap-es.org/contenido/04/12/1.-%C2%BFque-es-ldap%3F

- Reynoso, Carlos; Kicillof, Nicolás. *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft*. Universidad de Buenos Aires 2004.