Universidad de las Ciencias Informáticas

Facultad 1

Propuesta de Guía para el Diseño de los Mecanismos de Autenticación y Autorización en Aplicaciones Web.

Trabajo de Diploma para Optar por el Título de Ingeniero en Ciencias Informáticas

Autores:

Laura Gertrudis Escandell Arada

Tutor: Ing. Yoandy Rodríguez Martínez

"Ciudad de La Habana. Junio, 2012"

Declaración de Autoría

Declaramos que somos los únicos autores de este trabajo y autorizamos al Centro de Identificación y Seguridad Digital de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio. Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

_____

Laura Gertrudis Escandell Arada

_____

Tutor: Ing. Yoandy Rodríguez Martínez

Agradecimientos

Agradecer en primer lugar a nuestro país por permitirme estudiar esta carrera, al tutor por la ayuda brindada.

Agradecer a toda mi familia por el apoyo brindado en todo momento, en especial a mis padres y hermana.

A los profesores que me brindaron su ayuda incondicional, especialmente la profesora Alina.

A todos mis amistades de la Universidad especialmente a mi amiga Zoraimi, Alexander y a mis compañeras de apartamento por su ayuda y apoyo.

A mi prima Belkis por su ayuda incondicional.

Dedicatoria

A mis padres quienes me han brindado amor incondicional y siempre me han impulsado a superarme profesionalmente.

A mi hermana a quien quiero mucho.

A mis abuelos, que aunque algunos ya no están, nunca los voy a olvidar.

Resumen

En el presente trabajo de tesis se propone una Guía para el Diseño de los Mecanismos de Autenticación y Autorización, la cual contiene las actividades fundamentales que se deben seguir para facilitar este diseño en las aplicaciones.

Para alcanzar esta meta, se estudian los diferentes protocolos y modelos que manejan estos aspectos de la seguridad. Se analizan las principales guías para el desarrollo de las aplicaciones web que consideran de forma general los mecanismos a utilizar en la autenticación y autorización en los sistemas de información.

Además la Guía propuesta en esta investigación, favorecerá el incremento de la eficiencia en las aplicaciones web al implementar estos mecanismos, ya que recomienda el uso de los protocolos existentes más usados para elaborar el diseño adecuado de autenticación y autorización en las aplicaciones web.

Abstract

This thesis proposes a guide for the design of authentication mechanisms authorization, which contains a flow of activities that should be taken for a good analysis and design of these mechanisms in applications.

To achieve this goal, we study the different protocols, models that handle these aspects of security. It discusses the main guidelines for the development of web applications that analyze in general the various mechanisms used for authentication and authorization information systems.

The proposal in this research guide, lead to increased efficiency in applications when implementing this mechanism, since in this way the computer systems did not present problems often generated in the authentication and authorization.

Contenido

Índice de Tablas

Índice de Figuras

Introducción

Con la llegada de Internet, el enfoque de desarrollo de aplicaciones informáticas ha cambiado considerablemente, y también el concepto de usuario propuesto para ellas. Actualmente las aplicaciones web (páginas, sitios, portales) permiten a millones de usuarios acceder a la información que contienen, cada una de estas con diferentes objetivos y niveles de conocimiento.

Debido a que el uso de Internet y la Intranet se encuentran en aumento, cada vez más instituciones permiten a sus usuarios acceder a sus sistemas de información. Por tanto, es fundamental saber qué recursos de la institución necesitan ser protegidos para así controlar el acceso al sistema y los derechos de los usuarios del sistema de información. Los mismos procedimientos se aplican cuando se permite el acceso a la compañía a través de Internet.

Un efecto secundario del crecimiento exponencial que ha tenido el Internet y la Intranet es la privacidad de información tanto personal como profesional. En Internet encontramos funcionando a tiendas en línea, negocios que mueven grandes cantidades de dinero, redes de los servicios que habilitan el comercio a nivel internacional, así como sitios de redes sociales que contienen información muy delicada de la vida privada de sus miembros.

Mientras más se conecta el mundo, la necesidad de seguridad en los procedimientos usados para compartir la información va en aumento. Por lo que la importancia de centralizar e integrar la seguridad de estos cobra cada día más fuerza.

Entre los retos que presenta este proceso se encuentra el desarrollo de aplicaciones que hagan uso de los estándares establecidos por la industria para la autenticación y autorización de los usuarios.

Para ello se ha investigado en los métodos que han realizado para fortalecer la seguridad en las aplicaciones web a partir de la manera de acceder a los sistemas. Por lo que varias empresas han desarrollado soluciones para mitigar las vulnerabilidades que frecuentemente reportan las aplicaciones. Generalmente los problemas más identificados en las aplicaciones web se generan a partir del proceso de autenticación y la autorización en los sistemas. Desde

el punto de vista de los recursos y servicios, para los usuarios no debería suponer una dificultad a la hora de acceder a los mismos. En la actualidad, los entornos y aplicaciones de usuario tienden a enmascarar la localización y acceso a los diferentes servicios. En consecuencia, el usuario percibe y trata los recursos como locales, sin importarle ni la localización de los servidores implicados, ni los métodos de acceso a cada uno de ellos, y por supuesto, evitando el tener que conocer diferentes contraseñas para acceder a cada uno de ellos. Del mismo modo, pero desde el punto de vista del cliente, la capacidad de acceder a los servicios y recursos debe ser independiente de la localización del usuario y del método de conexión a Internet que utilice.

Un ejemplo de las soluciones que se han realizado por algunas empresas para prevenir las amenazas comunes a que son expuestos los sistemas, ha sido el desarrollo de las guías de seguridad para la implementación de aplicaciones web seguras.

Luego de realizar un estudio, se encontraron varias empresas que han elaborado guías para el desarrollo de aplicaciones web, entre ellas se podría mencionar la Guía de Campo de Aplicaciones Web, que ha sido realizada por Bert Appward, del equipo de desarrolladores de Google Chrome. También se encuentra la Guía para el Desarrollo de Aplicaciones Web utilizando la tecnología Objeto Relacional de Oracle que muestra una propuesta para el desarrollo de Aplicaciones Web utilizando modelos Objeto Relacional y una alternativa para los desarrolladores que tienen un amplio conocimiento de PL/SQL. El objetivo principal de este proyecto es desarrollar una guía para el desarrollo de aplicaciones web utilizando la tecnología Objeto Relacional de Oracle que oriente a los desarrolladores en la implementación de objetos en la base de datos así como su manipulación, además se proporcionan lineamientos para gestionar el desarrollo de dichas aplicaciones. Entre las guías para el desarrollo de aplicaciones web citadas, se analizaron con mayor profundidad la Guía para Construir Aplicaciones y Servicios Web Seguros de OWASP y ASP.NET: Guía para el Desarrollo de Sitios y Aplicaciones Web Dinámicas. OWASP es un proyecto de código abierto dedicado a determinar y combatir las causas que hacen que el software sea inseguro. Algunos de los aspectos tratados en la guía de OWASP son: consideraciones de arquitectura, mecanismos de autenticación, gestión de sesiones de usuario, mecanismos de autorización, entre otros.

Por otra parte la Guía de ASP.NET es una Guía de Desarrollo de Sitios y Aplicaciones Web Dinámicas. ASP.NET es un modelo de desarrollo web unificado creado por Microsoft para el

desarrollo de Sitios y Aplicaciones Web Dinámicas con un mínimo de código, forma parte de .NET Framework que contiene las librerías necesarias para la codificación.

Luego de realizar un estudio para el conocimiento de lo que proponen estas guías surge la siguiente Situación Problémica:

Las guías existentes para el desarrollo de aplicaciones web como: la Guía para Crear Aplicaciones y Servicios Web Seguros de OWASP y la Guía de ASP.NET para el Desarrollo de Sitios y Aplicaciones Web Dinámicas analizan de forma general los mecanismos de autenticación y autorización para el desarrollo de las aplicaciones web, pero no profundizan en los protocolos que pueden usarse para su implementación: reflejan los elementos a tener en cuenta para implementar estos mecanismos de forma aislada y sin un orden lógico, implicando que sea costoso el análisis para diseñarlos durante el desarrollo de aplicaciones web.

Luego del análisis de las dificultades detectadas surge el siguiente:

Problema Científico: ¿Cómo facilitar el diseño de los mecanismos de autenticación y autorización en aplicaciones web?

Teniendo como **Objeto de Estudio:** el proceso de diseño de los mecanismos de autenticación y autorización.

Objetivo general:

Elaborar una guía que facilite el análisis y diseño de los mecanismos de autenticación y autorización en las aplicaciones web.

Objetivos de la investigación:

- Analizar los referentes teóricos del tema de investigación.
- Elaborar la propuesta de guía de diseño de los mecanismos de autenticación y autorización de aplicaciones web.
- Validar la propuesta de guía elaborada mediante el diseño de un caso de estudio.

Campo de Acción: el proceso de diseño de los mecanismos de autenticación y autorización en las aplicaciones web.

Para el desarrollo de la investigación se aplicó el método científico de investigación, el cual permite abordar la realidad, estudiar la naturaleza, sociedad y el pensamiento, con el propósito de descubrir su esencia y sus relaciones.

Los métodos científicos de investigación plantean un procedimiento para realizar cualquier investigación, ya que éste constituye una mejor organización del trabajo y exige el establecimiento de un conjunto de reglas con objetivos bien precisos. Dentro de las clasificaciones que se le dan a los métodos científicos están: métodos teóricos y métodos empíricos, y aunque difieren en el nombre, estos están dialécticamente relacionados.

Método teórico: Permiten estudiar las características del objeto de investigación que no son observables directamente, proporcionan la construcción de modelos e hipótesis de investigación y crean las condiciones para ir más allá de las características fenomenológicas y superficiales de la realidad, favoreciendo al desarrollo de las teorías científicas y para su cumplimiento se apoyan en el proceso de análisis y síntesis.

Método empírico: Detallan y explican las características fenomenológicas del objeto, representan un nivel de la investigación cuyo contenido procede de la experiencia y es sometido a cierta elaboración racional.

Métodos científicos para el desarrollo de la investigación:

Teóricos:

- **Histórico-Lógico:** Aplicado en el estudio de trabajos realizados sobre el tema para apoyar la base teórica del trabajo, es decir, este método se ve reflejado mediante el estudio de la teoría que permitirá arribar a las conclusiones a partir de los conocimientos adquiridos en el transcurso de la investigación y para analizar la trayectoria por la que atraviesan los elementos que se tratan en el marco teórico, manejándose el avance de dichos elementos y sus conexiones históricas fundamentales. En este método se usará principalmente la búsqueda y las consultas bibliográficas.
- **Analítico-Sintético:** Este método se muestra al hacer un análisis de la situación problémica y obtener la solución. Además se consultarán y estudiarán varias bibliografías para definir una solución informática. Se realizará un resumen de los aspectos más importantes.

Empíricos:

- **Entrevista:** Admitirá información referente al objeto de estudio. Este método es una técnica para obtener información mediante una conversación profesional entre un entrevistado y el entrevistador. La misma se lleva a cabo a través de un diálogo o un cuestionario previamente elaborado. Su uso atribuye conocimiento cualitativo a la investigación, puede ser individual o colectiva. El investigador debe tener bien claro lo que desea preguntar ya que esta puede requerir de mucho tiempo. En la investigación se usa la entrevista para validar la propuesta de solución, se entrevistarán una serie de expertos que darán su criterio, obteniéndose los resultados luego del análisis de sus respuestas.

Estructuración de capítulos

Capítulo 1 Fundamentación Teórica: Se exponen los conceptos fundamentales que sustentan la investigación. Incluye el análisis de la información existente acerca de las tendencias y tecnologías que existen en la actualidad. Se identifican y analizan soluciones que gestionan los mecanismos de autenticación y autorización en las aplicaciones web.

Capítulo 2 Propuesta de solución: Se identifican los pasos a seguir para elaborar el diseño de los mecanismos de autenticación y autorización en las aplicaciones web. Se elaboró la propuesta de guía para el diseño de los mismos basada en sus distintos escenarios de aplicación.

Capítulo 3 Validación de la propuesta de solución: Se realiza el diseño de un caso de estudio en el cual se aplicará la guía propuesta.

Capítulo 1: Fundamentación Teórica

1.1 Introducción

Este capítulo aborda los principales conceptos relacionados con la seguridad informática, en Internet y de las aplicaciones web. Además hace un breve estudio sobre los protocolos de seguridad y los modelos que tratan los aspectos principales para garantizar los mecanismos de autenticación y autorización en las aplicaciones informáticas.

1.2 Marco Conceptual

En la actualidad con el desarrollo de la computación, vinieron avances tecnológicos como es la llegada de Internet, la cual juega un papel fundamental en las comunicaciones, en el mundo de los negocios, la información, el entretenimiento, las finanzas, etc. También con la evolución de esta se desarrollan aplicaciones web que involucran información de carácter confidencial y que requieren mecanismos de seguridad que garanticen que dicha información no será modificada, sustraída o falsificada por personas ajenas. Por tanto es de vital importancia tener en cuenta la **Seguridad Informática**, la cual según fuentes electrónicas es: la disciplina que se ocupa de diseñar las normas, procedimientos, métodos y técnicas, orientadas a promover condiciones seguras y confiables para el procesamiento de datos en sistemas informáticos. Además se puede decir, que es el conjunto de estrategias y herramientas que permiten garantizar la integridad, la disponibilidad y confidencialidad de la información de una entidad. (Garfinkel, 1999)

También en la Seguridad Informática se debe distinguir dos propósitos de protección, la Seguridad de la Información y la Protección de Datos.

Se debe distinguir entre los dos porque forman la base y dan la razón, es decir en la Seguridad de la Información el objetivo de la protección son los datos mismos y trata de impedir su pérdida y modificación no autorizada, mientras que en la protección deben garantizar en primer lugar, la confidencialidad, integridad y disponibilidad de los datos, sin embargo existen otros requisitos como son la autenticidad y el no repudio, entre otros, de los cuales se muestra una breve descripción a continuación. (Informática, 2011)

En el caso de la protección de datos, el propósito no son los datos en sí mismo, sino el contenido de la información sobre personas, para evitar el abuso de ésta.

Integridad: Garantiza que la información no será alterada, eliminada o destruida por entidades no autorizadas. Mediante este principio de la seguridad informática se salvaguarda la exactitud e integridad de la información y de los métodos de procesamientos. Esto quiere decir que a través de éste se puede modificar la información sólo mediante el personal autorizado, pues con éste se asegura que los datos sean los que se supone que son. (Sandoval, 2008)

Disponibilidad: Asegurar que los usuarios autorizados tendrán acceso a la información cuando la requieran. Este principio se refiere al debido acceso siempre y cuando se necesite. En otras palabras asegurar que dicha información o datos, estén disponibles, de esto se basa la continuidad operativa. También el objetivo de este es garantizar el acceso a un servicio o a los recursos, que la información del sistema permanezca accesible mediante el personal autorizado y que la información sea estable. (Sandoval, 2008)

Confidencialidad: Permite el acceso a la información o los datos sólo mediante la autorización y de forma controlada. Es decir que con este pilar básico de la seguridad se garantiza que únicamente el personal autorizado tenga acceso a la información y no deberá tener algún tipo de difusión no autorizada. Además se puede decir que consiste en hacer que la información sea ininteligible para aquellos individuos que no estén involucrados en la operación, pues los datos tienen que ser legibles únicamente para los usuarios autorizados. (Sandoval, 2008)

Autenticidad: Este principio permite que el sistema sea capaz de verificar la identidad de los usuarios, y los usuarios la del sistema, ya que mediante ésta se garantiza que en cada una de las partes, su interlocutor es realmente quien dice ser. Además asegura que sólo los individuos autorizados tengan acceso a los recursos. (Mifsud, 2012)

No repudio: Consiste en evitar el repudio de información, pues constituye la garantía de que ninguna de las partes involucradas pueda negar en el futuro una operación realizada, es decir el usuario no debe poder negar las acciones que realizó. (informática, 2011)

Como resumen de las bases de la seguridad informática que se han explicado, se puede decir que la seguridad consiste en conservar el equilibrio adecuado entre estos tres factores. No tiene sentido adquirir la confidencialidad para un archivo si es a costa de que ni tan siquiera el

usuario administrador pueda acceder a él, ya que se está negando la disponibilidad. (Mifsud, 2012)

Pero no siempre se puede mantener el equilibrio entre los pilares fundamentales de la **Seguridad Informática**, ya que esto va en dependencia del entorno de trabajo y de las necesidades que presente la entidad en la cual se pueden dar prioridad a un aspecto de la seguridad o a otro. Por ejemplo en ambientes militares suele ser siempre prioritaria la confidencialidad de la información frente a la disponibilidad. Aunque alguien pueda acceder a ella o incluso pueda eliminarla, no podrá conocer su contenido y reponer dicha información, será tan sencillo como recuperar una copia de seguridad (si las cosas se están haciendo bien). (Mifsud, 2012)

En ambientes bancarios es prioritaria siempre la integridad de la información frente a la confidencialidad o disponibilidad. Se considera menos dañino que un usuario pueda leer el saldo de otro usuario a que pueda modificarlo. (Mifsud, 2012)

Luego de haber mencionado todos estos conceptos es fundamental tener conocimiento que todos ellos son especialmente válidos en el entorno de Internet y particularmente importantes dado el crecimiento explosivo de los servicios y aplicaciones accesibles a través de ésta. Por eso es elemental saber en qué consiste la **Seguridad en Internet**, para así saber qué medidas tomar para erradicar los posibles ataques a los que es expuesta, que según fuentes electrónicas no es más que la implementación de mecanismos de seguridad para que cuando se reciba un mensaje o se realice una transacción por medios electrónicos, se asegure la integridad del contenido y la identidad del remitente y del receptor. Además se puede entender como un conjunto de procedimientos, prácticas y tecnologías para proteger a los servidores, usuarios de la web y las organizaciones que la rodean. La Seguridad es una protección contra el comportamiento inesperado. (Garfinkel, 1999)

Un ejemplo de estos mecanismos son las contraseñas y palabras clave que ya no son un mecanismo suficientemente fiable y seguro, ya que éstas pueden ser interceptadas durante su transmisión, de lo que desgraciadamente es detectado muy tarde o cuando la prensa se hace eco de un caso de estafa electrónica. (Lomáscolo, 2000)

La Seguridad en Internet y las leyes que la protegen, están basadas principalmente en los sistemas de encriptación. Esos sistemas son los que permiten que las informaciones que

circulan por Internet sean indescifrables, ininteligibles, para cualquier persona que no sea aquella a la que va destinada. (Lomáscolo, 2000)

Este es un tema ampliamente debatido y es fácil de entender, sobre todo por las grandes corporaciones y organizaciones empresariales, pues éstas necesitan tener vigente el cumplimiento de los principios de un sistema informático, ya que las organizaciones necesitan proteger la confidencialidad de la información reservada. Por otra parte, los usuarios de a pie también deberían vigilar de cerca todo lo referente a la protección de sus datos y a la identidad de las fuentes y destinatarios de los mismos. (Lomáscolo, 2000)

Evidentemente la seguridad en Internet afecta excesivamente a las empresas que operan con banca electrónica, ya que las cuentas bancarias en Internet no son más que bases de datos y, como tales, están expuestas. En definitiva, la seguridad afecta a todos: a las grandes compañías por ser una tentación y por las consecuencias de una posible filtración, y a los usuarios individuales por su vulnerabilidad. (Lomáscolo, 2000)

Pero existen soluciones seguras. En el plano técnico, ya hay en el mercado mecanismos que pueden asegurar los contenidos y la identidad de las partes que se comunican y realizan transacciones en Internet. (Lomáscolo, 2000)

Un ejemplo de estas soluciones para minimizar los problemas de identidad, autenticación y autorización mayormente generados en las aplicaciones que se acceden a través de la Internet son: el uso de los protocolos ampliamente adoptados en Internet como: OpenID, que es un **protocolo de autenticación federada**, y consiste básicamente en que el usuario selecciona un servidor externo (el "proveedor" de OpenID) que va a ser el que va a validar su identidad en un sistema determinado (el "consumidor" de OpenID). También se encuentra el protocolo abierto OAuth, a diferencia de OpenID, es un **protocolo de autorización**; más exactamente, **de delegación de acceso;** es decir, permite definir cómo un tercero va a acceder a los recursos propios. El propósito de este protocolo es, pues, que un usuario que tiene determinados recursos en un servidor (el "proveedor" de OAuth) pueda dar acceso a un tercero (el "consumidor", usualmente un sitio web) a parte o todos esos recursos, sin necesidad de que ese tercero conozca su usuario y contraseña, ya que con esos datos tendría el control total de la cuenta. Además existen otros protocolos que brindan sistemas de autenticación y autorización en Internet como: SAML (Security Assertions Markup Language), CAS (Servicio de Autenticación Central), PAPI (Punto de Acceso a Proveedores de Información), XACML

(Extensible Authorization Markup Language), los cuales se dará una mayor descripción en el transcurso de la investigación.

Si bien cuando hablamos de la Seguridad de Internet, se debe considerar también la **Seguridad de las Aplicaciones Web,** ya que están estrechamente relacionadas, pues como se mencionó anteriormente muchas de las aplicaciones son accedidas a través de la web. Además en la actualidad, la importancia de la seguridad en Aplicaciones Web es crucial, debido a que las transacciones de índole económica han crecido enormemente. Por eso es esencial garantizar que los aspectos básicos de seguridad, nombrados anteriormente a la hora de desarrollar aplicaciones web se cumplan, ya que si se omitiera alguno de estos puede traer serias consecuencias como por ejemplo: la disponibilidad que se refiere a la continuidad en el servicio prestado, ya que una interrupción en el servicio puede traer un seria afectación, tanto en dinero como en desprestigio de la organización. (Web, 2011)

También cuando hablamos de temas de crucial importancia como: la Seguridad en Internet y la Seguridad de las Aplicaciones Web es inevitable no mencionar la **Seguridad en la Intranet**. Por eso es necesario primeramente tener conocimiento de lo que es una Intranet.

La Intranet es la implantación o integración en una red local o corporativa de tecnologías avanzadas de publicación electrónica basadas en WEB en combinación con servicios de mensajería, con partición de recursos, acceso remoto y toda una serie de facilidades cliente / servidor proporcionadas por la serie de protocolos TCP/IP, diseñado inicialmente para la red global Internet. Su propósito fundamental es optimizar el flujo de información con el objeto de lograr una importante reducción de costes en el manejo de documentos y comunicación interna.

Es una herramienta de gestión que permite una potente difusión de información y mecanismos de colaboración entre el personal. Al igual que en Internet la pieza clave de la Intranet es el World Wide Web (WWW), pero de forma que la información de la empresa esté accesible sólo a los miembros de la organización, quienes, en consecuencia disponen de navegadores WWW para acceder a los datos internos de la empresa.

La Intranet como la red de Internet, tienen en común la tecnología subyacente. Una infraestructura basada en estándares y en tecnologías que soportan el uso compartido de recursos comunes. Por infraestructura se hace referencia a la que crea, administra y permite

compartir el contenido. La única restricción técnica, es que la red física debe estar basada en el protocolo IP (Internet Protocol). Por lo tanto el objetivo de las dos radica en la posibilidad de compartir contenido y recursos. (Magazine, 2011)

Si hubiese que definir qué hace diferente a cada una, se diría que Internet ofrece "teóricamente" acceso a la información a un grupo ilimitado de personas alrededor del mundo, mientras que una Intranet sólo permite el acceso a información privada y recursos de una organización a aquellas personas que pertenecen a la misma o que están estrechamente relacionadas a la organización y que tiene permiso para hacer uso, sin que ello implique de ninguna forma que tienen acceso a toda la información y a todos los recursos. (Magazine, 2011)

Uno de los aspectos más importantes entre sus características, a la hora de establecer una Intranet, es el de la seguridad. Para que los miembros de una organización, y sólo ellos, puedan acceder a la información, cualquier conexión que no tenga una autorización debe ser automáticamente bloqueada, para evitar accesos indeseados e incluso fuga de información importante. (Internet, 2011)

Por eso en la actualidad las empresas que establecen Intranet son conscientes de la gran importancia que tiene para el desarrollo de sus actividades el hecho de proteger de forma adecuada la información que poseen y especialmente aquella que les sirve para realizar correctamente su actividad de negocio. El poder gestionar bien la seguridad de la información que manejan le permitirá garantizar, de cara a la propia organización, que sus recursos están protegidos, asegurando la confidencialidad, integridad y disponibilidad que son los pilares básicos para implantar una Intranet en cualquier institución. (Internet, 2011)

Otra de las características que presentan las Intranets es que proporciona una plataforma excelente para poner en manos de la dirección toda la información relevante de la empresa. Se trata de realizar un cuadro de mando hipermedia, desde el que los usuarios actualizados pueden acceder a los indicadores clave de la compañía simplemente navegando a través de la Página Web. (Internet, 2011)

Además las compañías están encontrando que las Intranets son herramientas poderosas para automatizar procesos, incrementando la posibilidad de acceso a sistemas críticos e información importante, compartiendo las mejores prácticas, colaborando para solucionar problemas a

clientes y permitiendo un alto nivel de interacción entre sus integrantes. Los beneficios de una Intranet bien implementada incluyen el incremento de la flexibilidad de la empresa, la habilidad para responder más rápidamente a las condiciones cambiantes del mercado y la habilidad de servir mejor a sus clientes. (Magazine, 2011)

También otros de los factores que influyen poderosamente en el establecimiento de una Intranet pueden resumirse como se muestra a continuación:

- Costes asequibles, tanto de su puesta en marcha como de uso. Es una forma muy eficiente y económica de distribuir la información interna, sustituyendo los medios clásicos.
- Fácil adaptación y configuración a la infraestructura tecnológica de la organización, así como gestión y manipulación. Disponible en todas las plataformas informáticas.
- Adaptación a las necesidades de diferentes niveles: empresa, departamento, área de negocio, etc. Centraliza el acceso a la información actualizada de la organización, al mismo tiempo que puede servir para organizar y acceder a la información de la competencia dispuesta en Internet.
- Sencilla integración de multimedia.
- Posibilidad de integración con las bases de datos internas de la organización.
- Rápida formación del personal.
- Acceso a Internet, tanto al exterior, como al interior, por parte de usuarios registrados con control de acceso.
- Utilización de estándares públicos y abiertos, independientes de empresas externas, como pueda ser TCP/IP o HTML.

Luego de exponer toda esta información referente a las Intranets, se arriba a la conclusión de que uno de los mayores problemas de la información interna de las empresas es la variedad de plataformas y sistemas informáticos existentes en cualquier organización, y los problemas para compartir información entre ellos. Una de las grandes ventajas del Internet, que explica su éxito internacional es que da cabida a todo tipo de equipos, (Mac, PC, Unix, Vax, etc.) fabricantes,

redes, tecnología y medio físico de transmisión. Con esta premisa, una idea muy interesante es utilizar las tecnologías de Internet dentro de una organización. En ello se basan las llamadas Intranet, es decir, se aprovechan de las herramientas de Internet para su utilización interna dentro de las redes corporativas de una empresa.

Después de realizar un breve estudio sobre la Seguridad Informática y de conocer la estrecha relación que mantiene con las tecnologías que vinieron con el desarrollo de la informática como: la Internet y las Aplicaciones Web, en cuanto a los aspectos de seguridad de la información se refiere, se puede afirmar que la seguridad es un aspecto importante para proteger la integridad y privacidad de los datos y recursos de la aplicación web.

Además debido a las amenazas constantes a las aplicaciones web son razones de peso para pensar en orientar los esfuerzos a ser programadores más "seguros".

Programar de forma segura en la web, amerita una serie de conocimientos acerca de gran cantidad de vulnerabilidades que pueden ser explotadas por los atacantes y cómo prevenirlas, por lo que si eres programador, diseñador web o estás de alguna manera involucrado con el desarrollo de aplicaciones web, es muy posible que ya hayas enfrentado algún problema de seguridad en alguna de las aplicaciones web que has desarrollado. Lógicamente de este tipo de experiencia podrás deducir que el aprendizaje por ensayo y error no es una opción.

Por tanto, es necesario para evitar que las aplicaciones sean fácilmente vulneradas y proporcionarles a los clientes la seguridad que necesitan, replantear algunos aspectos acerca de la forma en que se van a crear dichas aplicaciones de ahora en adelante.

Por lo que algunas instituciones han realizado guías para mejorar la manera en que desarrollan las aplicaciones desde el punto de vista de la seguridad. Un ejemplo de esto es: La **Open Web Application Security Project** (OWASP), que en español quiere decir Proyecto de Seguridad de Aplicaciones Web Abiertas, es una organización a nivel mundial centrada en proveer asistencia para mejorar la seguridad de las aplicaciones. Su principal misión es difundir información sobre vulnerabilidades y fallos en su guía para que las organizaciones y los desarrolladores puedan aplicarla para evitar riesgos reales de seguridad. Todo el material que generan se encuentra disponible bajo una licencia Open Source. (OWASP, 2011)

Además el proyecto OWASP (Open Web Application Security Project) tiene como objetivo ofrecer una metodología, de libre acceso y utilización, que pueda ser utilizada como material de

referencia por parte de los arquitectos de software, desarrolladores, fabricantes y profesionales de la seguridad involucrados en el diseño, desarrollo, despliegue y verificación de la seguridad de las aplicaciones y servicios web. (OWASP, 2011)

Al tratarse de una organización que no está vinculada con empresas de servicios informáticos ni de software sus publicaciones son absolutamente independientes y no aceptan presiones de ningún tipo. Fundamentalmente su proyecto se basa en estos productos: Guía de Desarrollo, Guía de Pruebas de Aplicaciones y la Guía de Revisión de Código, así como un ranking de las vulnerabilidades más activas durante un año, pero la presente investigación sólo realizará un breve estudio de la Guía de Desarrollo de esta organización, ya que es la que más se aproxima al tema del vigente trabajo y de la cual se mostrará un breve resumen a continuación en el siguiente epígrafe. (OWASP, 2011)

1.3 Guías de seguridad analizadas

Se realizó un análisis de las guías existentes para el desarrollo de aplicaciones web entre las que se encontró: la Guía para Construir Aplicaciones y Servicios Web Seguros de OWASP. Esta guía es un compendio de buenas costumbres y de sugerencias a implantar en el proceso de codificación de aplicaciones web para poder generar aplicaciones de calidad en cuanto a seguridad se refiere. Son directrices que nos evitan malos hábitos contándonos como se deben codificar diferentes patrones y soluciones teniendo siempre en mente la posibilidad de un ataque de alta capacidad técnica.

En ésta explican de forma muy directa y detallada las directrices, las cuales se clasifican, en cada una de las secciones, en “Deberías hacer esto” y en “Hazlo”, así como una explicación de los puntos débiles que se van a encontrar y las implicaciones de administración del sistema que se deberían contemplar. A continuación se muestra un breve resumen de los objetivos de la autenticación y autorización que tratan a continuación. (OWASP, 2011)

1.3.1 Guía para Construir Aplicaciones y Servicios Web Seguros de OWASP

La guía de OWASP propone como objetivos para la Autenticación

Proveer servicios de autenticación segura a las aplicaciones Web:

- Vinculando una unidad del sistema a un usuario individual mediante el uso de una credencial.

- Proveyendo controles de autenticación razonables de acuerdo al riesgo de la aplicación.
- Denegando el acceso a atacantes que usan varios métodos para atacar el sistema de autenticación.

La Guía de OWASP propone como objetivos para la Autorización:

- Asegurar que únicamente usuarios autorizados puedan realizar acciones permitidas con su correspondiente nivel de privilegio.
- Controlar el acceso a recursos protegidos mediante decisiones basadas en el rol o el nivel de privilegio.
- Prevenir ataques de escalada de privilegios, como por ejemplo utilizar funciones administrativas siendo un usuario anónimo o incluso un usuario autenticado. (OWASP, 2011)

La Guía de OWASP comienza estableciendo los principios básicos de seguridad que cualquier aplicación debe cumplir:

- Validación de la entrada y salida de información: siempre debe verificarse que cualquier dato entrante o saliente es apropiado y en el formato que se espera. Las características de estos datos deben estar predefinidas y debe verificarse en todas las ocasiones.
- Diseños simples: los mecanismos de seguridad deben diseñarse para que sean los más sencillos posibles, huyendo de sofisticaciones que compliquen excesivamente la vida a los usuarios.
- Utilización y reutilización de componentes de confianza: debe evitarse reinventar la rueda constantemente, es decir buscar nuevos componentes a utilizar. Por tanto, cuando exista un componente que resuelva un problema de forma correcta, lo más inteligente es utilizarlo.
- Defensa en profundidad: nunca confiar en que un componente realizará su función de forma permanente y ante cualquier situación. Se debe de disponer de los mecanismos de seguridad suficientes para que cuando un componente del sistema falle ante un determinado evento, otros sean capaces de detectarlo.
- Tan seguros como en el eslabón más débil: la frase "garantizamos la seguridad, ya que se utiliza SSL" es realmente muy popular, pero también es muy inexacta. La utilización de SSL garantiza que el tráfico en tránsito entre el servidor y el cliente se encuentra

cifrado, pero no garantiza nada acerca de los mecanismos de seguridad existentes. Por tanto, no se deben fiar únicamente de los mecanismos de seguridad "exteriores", sino que es preciso identificar cuáles son los puntos precisos en los que deben establecerse las medidas de seguridad. Si no se hace este trabajo, seguro que los atacantes si lo harán.

- La "seguridad gracias al desconocimiento" no funciona: el simple hecho de ocultar algo no impide que, a medio o largo plazo, llegue a ser descubierto. Tampoco es ninguna garantía de que tampoco será descubierto a corto plazo.
- Verificación de privilegios: los sistemas deben diseñarse para que funcionen con los menos privilegios posibles. Igualmente, es importante que los procesos únicamente dispongan de los privilegios necesarios para desarrollar su función, de forma que queden compartimentados.
- Ofrecer la mínima información: ante una situación de error o una validación negativa, los mecanismos de seguridad deben diseñarse para que faciliten la mínima información posible. De la misma forma, estos mecanismos deben estar diseñados para que una vez denegada una operación, cualquier operación posterior sea igualmente denegada. (OWASP, 2011)

Además en esta guía se tratan otros aspectos como son:

- Consideraciones de arquitectura.
- Mecanismos de autenticación.
- Mecanismos de autorización.
- Gestión de sesiones de usuario.
- Control de acceso.
- Registro de actividad.
- Prevención de problemas comunes.
- Consideraciones de privacidad.
- Consideraciones de criptografía.

Después de haber realizado un breve análisis de los temas que trata la Guía de Desarrollo de OWASP, se puede decir que la guía de desarrolladores cubre la mayoría de los controles de seguridad que los desarrolladores de software deben utilizar. Estas son las protecciones “positivas” que los desarrolladores deben construir en sus aplicaciones. Aunque hay miles de tipos de vulnerabilidades en software, ellos pueden ser generalmente evitados con un

conveniente conjunto de controles de seguridad fuertes. Pero no hacen énfasis en los protocolos que pueden utilizarse para su desarrollo. Además muestran los aspectos a tener en cuenta para la implementación de los mecanismos de autenticación y autorización de manera aislada, lo que trae como consecuencia que sea más costoso el análisis para diseñarlos.

1.3.2 Guía para crear aplicaciones ASP.NET seguras

Otra de las guías analizadas además de la Guía de Desarrollo de OWASP como se muestra anteriormente fue la ASP.NET: Guía de Desarrollo de Sitios y Aplicaciones Web Dinámicas.

ASP.NET es un modelo de desarrollo web unificado creado por Microsoft para el desarrollo de sitios y aplicaciones web dinámicas con un mínimo de código, forma parte de .NET Framework que contiene las librerías necesarias para la codificación.

La Guía ASP.NET proporciona los requerimientos mínimos para programar, desarrollar una primera aplicación, ejecutar código Java script, aprender sobre los controladores de servidos y eventos. Además, en ella se muestra la utilización de estilos (CSS), manejo de WebForms, la estructura de clases y objetos. También, en ésta se hace uso de Master Pages, Ajax.

Esta guía se concentra en la autenticación para identificar a los clientes de la aplicación, autorización para ofrecer controles de acceso para los clientes y la comunicación segura para garantizar que los mensajes mantienen su privacidad y no son manipulados por personas no autorizadas.

Luego de exponer los aspectos de seguridad en los que se basa esta guía es fundamental aclarar porque se centra en ellos nada más. Primeramente tener en cuenta, que la seguridad es un tema muy amplio. Por lo que según los estudios, al diseñar al principio la autenticación y autorización se elimina un alto porcentaje de vulnerabilidades de las aplicaciones. La comunicación segura es una parte esencial de la protección de la aplicación distribuida para garantizar la confidencialidad de los datos (incluidas las credenciales) que se transmiten a la aplicación y desde ella, y entre niveles de la aplicación. (ASP, 2011)

La información de esta guía tiene como objetivo mostrar cómo:

- Aumentar la seguridad de la aplicación.
- Identificar dónde y cómo es necesario realizar la autenticación.

- Identificar dónde y cómo es necesario realizar la autorización.
- Identificar riesgos comunes y estrategias para evitarlos.
- Identificar los riesgos principales y su mitigación en relación con la autenticación y la autorización.
- Evitar comprometer la seguridad solamente para que todo funcione bien.
- Identificar no sólo cómo, sino también cuándo utilizar varias características de seguridad.
- Eliminar TID (temor, incertidumbre y dudas).
- Promover prácticas recomendadas y resultados predecibles. (ASP, 2011)

Después de mostrar los objetivos en que se basa la guía, es muy importante analizar de forma breve el tratamiento que se le da en esta guía a la autenticación y autorización, pues son dos de los aspectos que más prioridad se le brinda en la guía y que pueden proporcionar gran ayuda en el desarrollo de la presente investigación.

La guía dedica uno de sus capítulos a la autenticación y autorización en el cual transmite instrucciones para ayudar a diseñar una estrategia de autorización adecuada a su escenario de aplicación específico. Además servirá de ayuda para seleccionar la técnica de autenticación y autorización más apropiada y a aplicarla en los puntos adecuados de la aplicación.

Además según la Guía de ASP.NET el diseño de una estrategia de autenticación y autorización para aplicaciones web distribuidas constituye un verdadero desafío. Afortunadamente, el disponer de un diseño adecuado de la autenticación y la autorización durante las primeras fases del desarrollo de la aplicación ayuda a mitigar muchos de los peores riesgos de seguridad. (ASP, 2011)

Por esas razones la guía le enseña a diseñar una estrategia de autenticación y autorización adecuada para la aplicación y a responder a las siguientes preguntas clave para la realización de éstas como son: ¿Dónde debo realizar la autorización y qué mecanismos debo utilizar?, ¿Qué mecanismo de autenticación debo utilizar?

Pero antes de dar respuesta a estas preguntas primero se debe tener conocimiento que cuando se considere la autorización, se deberá también considerar la autenticación.

En primer lugar, cualquier directiva de autorización coherente requiere la existencia de usuarios autenticados. En segundo lugar, el modo de autenticación de los usuarios (y sobre todo el modo de representación de la identidad de usuarios autenticados en la aplicación) determina los guardianes de los que podrá disponer.

Algunos guardianes, como la autorización de archivos de ASP.NET, las funciones de la aplicación de Servicios Empresariales (COM+) y las listas de control de acceso (ACL) de Windows, requieren una identidad de Windows autenticada, en forma de objeto WindowsIdentity que encapsula un testigo de acceso de Windows, el cual define el contexto de seguridad del llamador. Otros guardianes, como la autorización de direcciones URL de ASP.NET y las funciones de .NET no la requieren. Requieren solamente una identidad autenticada, que no tiene por qué estar representada forzosamente por un testigo de acceso de Windows.

Para dar respuesta a estas preguntas la guía te enuncia algunos de los enfoques, estrategias y mecanismos de autorización, además de los diferentes mecanismos de autenticación que se pueden utilizar, de los cuales se mostrará una breve descripción a continuación.

Las dos estrategias básicas de autorización son:

- **Basada en funciones:** El acceso a las operaciones (normalmente métodos) se protege en función de la pertenencia a funciones del llamador. Las funciones sirven para dividir la base de usuarios de la aplicación en conjuntos de usuarios que comparten los mismos privilegios de seguridad en la aplicación, como por ejemplo, Directivos superiores, Directores y Empleados. Los usuarios se asignan a funciones y, si el usuario está autorizado a realizar la operación solicitada, la aplicación utiliza identidades fijas para obtener acceso a los recursos. Estas identidades tienen la confianza de los administradores de recursos respectivos (por ejemplo, las bases de datos, el sistema de archivos, etc.).

- **Basada en recursos:** Los recursos individuales se protegen mediante listas de control de acceso (ACL) de Windows. La aplicación suplanta al llamador antes de obtener acceso a los recursos, lo que permite al sistema operativo realizar controles

estándar de acceso. Todo acceso a recursos se realiza mediante el contexto de seguridad del llamador original. Este enfoque de suplantación tiene un fuerte impacto en la escalabilidad de la aplicación, puesto que no permite utilizar la agrupación de conexiones de forma eficaz en el nivel medio de la aplicación.

Los dos enfoques básicos de la autorización:

- **Basada en funciones:** Los usuarios se dividen en funciones lógicas definidas por la aplicación. Los miembros de una función determinada comparten los mismos privilegios en la aplicación. El acceso a las operaciones (normalmente llamadas a métodos) se autoriza en función de la pertenencia a funciones del llamador.

- **Basada en recursos:** Los recursos individuales se protegen mediante ACL de Windows. La ACL determina los usuarios a los que se permite el acceso al recurso junto con los tipos de operación que puede realizar cada usuario (leer, escribir, eliminar, etc.).

Una vez mencionado los distintos enfoques y estrategias de autorización que muestra la guía, es fundamental, también conocer los mecanismos de autorización que expone esta. Como por ejemplo:

- **Autorización basada en funciones:** La mayoría de las aplicaciones web .NET utilizan un enfoque de autorización basado en funciones. Deberá considerar los distintos tipos de funciones y seleccionar los más adecuados a su escenario de aplicaciones. Dispone de las siguientes opciones:

    - Funciones de .NET.
    - Funciones de la aplicación de Servicios Empresariales (COM+).
    - Funciones de base de datos definidas por el usuario de SQL Server.
    - Funciones de la aplicación de SQL Server.

A continuación se dará una breve descripción de los diferentes tipos de funciones mostradas anteriormente para tener conocimiento en lo que consisten.

- **Funciones de .NET:** Las funciones de .NET son muy flexibles y se basan en objetos IPrincipal que contienen la lista de funciones a las que pertenece una identidad

autenticada. Las funciones de .NET pueden utilizarse en aplicaciones web, servicios web o componentes remotos alojados en ASP.NET

La autorización con funciones de .NET puede realizarse de forma declarativa, mediante llamadas a la clase PrincipalPermission, o a través de un programa, mediante llamadas imperativas a esta clase o utilizando el método IPrincipal.IsInRole.

- **Funciones de la aplicación de Servicios Empresariales (COM+):** El uso de funciones de la aplicación de Servicios Empresariales (COM+) mueve los controles de acceso al nivel medio y le permite utilizar la agrupación de conexiones de bases de datos al establecer la conexión con bases de datos de servidor. No obstante, para una autorización basada en funciones de la aplicación de Servicios Empresariales (COM+) coherente, su aplicación web cliente deberá suplantar y transmitir la identidad del llamador original (mediante un testigo de acceso de Windows) a la aplicación de Servicios Empresariales.

- **Funciones de base de datos definidas por el usuario de SQL Server:** En este enfoque, se crean funciones en la base de datos, se asignan permisos basados en las funciones y se asignan cuentas de grupo y de usuario de Windows a las funciones. Este enfoque requiere que transmita la identidad del llamador al servidor (si utiliza la autenticación de Windows recomendada para SQL Server).

- **Funciones de la aplicación de SQL Server:** En este enfoque, los permisos se conceden a las funciones de la base de datos, pero las funciones de la aplicación de SQL Server no contienen cuentas de usuario ni de grupo.

Con las funciones de aplicación, se autoriza el acceso a una aplicación específica (en lugar de a un conjunto de usuarios). La aplicación activa la función mediante un procedimiento almacenado integrado que acepta un nombre de función y una contraseña. Una de las principales desventajas de este enfoque es que requiere que la aplicación administre las credenciales (el nombre de función y la contraseña correspondiente) de forma segura.

Además como se mencionó anteriormente la guía contiene instrucciones diseñadas para ayudarle a seleccionar un mecanismo de autenticación adecuado a escenarios de aplicaciones habituales. En primer lugar, deberá considerar los siguientes aspectos:

- **Identidades:** El mecanismo de autenticación de Windows sólo es adecuado si los usuarios de la aplicación tienen cuentas de Windows que pueda autenticar una autoridad de confianza a la que tenga acceso el servidor web de la aplicación.

- **Administración de credenciales:** Una de las ventajas clave de la autenticación de Windows reside en que le permite dejar que el sistema operativo se encargue de la administración de credenciales. En otros enfoques distintos de Windows, como la autenticación mediante Formularios, deberá considerar detenidamente la ubicación y el modo de almacenamiento de las credenciales de usuario.

- **Transmisión de la identidad:** ¿Necesita implementar un modelo de suplantación/delegación y transmitir el contexto de seguridad del llamador original por los niveles en el sistema operativo? Por ejemplo, para admitir la auditoría o la autorización por usuario (granular). En caso afirmativo, deberá poder suplantar al llamador y delegar su contexto de seguridad al subsistema siguiente de nivel inferior.

También en la guía se muestran un conjunto de métodos de autenticación que pueden utilizar en el momento de desarrollar una aplicación utilizando ASP.NET que son los siguientes:

Autenticación basada en Windows: Este tipo de autenticación, ha pasado a llamarse Autenticación Integrada de Windows y constituye una variante de la autenticación mediante resúmenes criptográficos. Se trata igualmente una forma segura de autenticación en la medida en que no se envían ni la contraseña ni el nombre de usuario a través de la Red. En su lugar, el navegador tiene que demostrarle al servidor que conoce la clave por medio de un corto intercambio de datos, pero sin revelar nunca la clave. No obstante, debido a los detalles de implantación, resulta incompatible con la autenticación por resúmenes. (ASP, 2011)

- **Autenticación básica:** Cuando el usuario accede a un recurso del servidor web protegido mediante autenticación básica, tiene lugar el siguiente proceso:

    1. El navegador presenta al usuario la ventana de autenticación, para que introduzca su nombre y contraseña.

    2. El navegador intenta establecer una conexión con el servidor utilizando esta información.

3. Si el servidor rechaza la información de autenticación, el navegador le presenta nuevamente la ventana al usuario hasta que éste introduce por fin una contraseña válida o cierra la ventana.

4. Cuando el servidor web verifica con éxito los datos de autenticación, se establece la conexión de acceso al recurso protegido.

- **Autenticación implícita:** Envía la información en un hash codificado, requiere Internet Explorer 5 o superior y además solicita Active Directory.

- **Autenticación basada en formularios:** Las solicitudes no autenticadas son redirigidas a un formulario HTML. Además el usuario proporciona credenciales, envía el formulario HTML y una vez verificadas las credenciales, se suministra una cookie de autenticación.

- **Autenticación mediante Microsoft Passport:** Servicio de autenticación centralizado que ofrece una única opción de inicio de sesión. Microsoft Passport es un Servicio Web XML. (ASP, 2011)

Luego de exponer algunos de los aspectos que toca la Guía de ASP.NET en cuanto a la autenticación y autorización se refiere se arriba a la conclusión de que aunque cuenta con una fuente de información muy amplia en cuanto a la seguridad de las aplicaciones, se orienta nada más al tema específico de la utilización de un modelo de desarrollo web como ASP.NET. Además esta guía de seguridad no brindan una vista completa de los protocolos de autenticación y autorización como por ejemplo: SAML, OpenID y otros que se analizan en la investigación.

Después de haber analizado las guías de desarrollo de ASP.NET y de OWASP se ha arribado a la conclusión de que uno de los aspectos más importantes en el desarrollo de las aplicaciones es la autenticación. Por lo que es necesario profundizar un poco más en este tema de modo general, para así tener conocimiento de los métodos y sistemas de autenticación existentes, entre otros elementos que se comentarán a continuación.

1.4 Autenticación

Uno de los aspectos principales que se debe tener en cuenta para garantizar la seguridad en las aplicaciones web es la autenticación, la cual es el proceso que debe seguir un usuario para tener acceso a los recursos de un sistema o de una red de computadores. Este proceso implica

identificación (decirle al sistema quién es) y autenticación (demostrar que el usuario es quién dice ser). La autenticación por sí sola no verifica derechos de acceso del usuario; estos se confirman en el proceso de autorización. (Info@citel, 2011)

Para llevar a cabo este proceso existen tres clasificaciones que se muestran a continuación:

Tipos de autenticación

Se puede efectuar autenticación usando uno o varios de los siguientes métodos:

- **Autenticación por conocimientos:** Basada en información que sólo conoce el usuario, es decir algo que sólo este sabe, como por ejemplo: contraseña, código, frases de paso.
- **Autenticación por pertenencia:** Basada en algo que posee el usuario, es decir algo que tiene el usuario, como por ejemplo: tarjeta de identidad, tarjeta inteligente (smartcard), dispositivo USB (tokens).
- **Autenticación por características:** Basada en alguna característica física del usuario, es decir cualquier característica física del usuario, como por ejemplo: verificación de voz, escritura, huellas, patrones oculares, etc. (Info@citel, 2011)

De lo anterior se deduce que la autenticación involucra aspectos físicos y lógicos relacionados con el acceso, la utilización y la modificación de los recursos de la red o sistema, es decir que se pueden clasificar en dos tipos de autenticación.

- **Autenticación física:** La autenticación física se basa en algún objeto físico que posee el usuario, o en alguna característica física del usuario; en tal caso utiliza algún tipo de mecanismo biométrico. La información capturada en el proceso de autenticación, pasa al proceso de autorización realizado por personas, dispositivos electrónicos de seguridad o sistemas de seguridad informática.
- **Autenticación lógica:** La autenticación lógica puede utilizarse para identificar personas o sistemas y se basa en información que sólo conoce el usuario. La autenticación y autorización las realiza el software especializado. (Info@citel, 2011)

En función del número de factores en los que se base el sistema de autenticación se puede hablar de:

- **Autenticación unimodal:** Basado en algún elemento conocido, algo poseído o alguna característica física (por ejemplo la contraseña asociada a un usuario para entrar en un portal web o un token criptográfico o una huella dactilar).
- **Autenticación múltiple:** Si se combinan dos o más métodos de autenticación, ésta se denomina autenticación múltiple (multi-factor authentication) y es una autenticación más segura. Por ejemplo, autenticación doble si el usuario debe presentar dos tipos de identificación, una física (una tarjeta) y el otro algo que el usuario ha memorizado como una clave de seguridad o un número de identificación personal (PIN—Personal Identification Number). Este es el caso de una tarjeta bancaria que se utiliza con un cajero automático (ATM—Automatic Teller Machine). Más aún, algunos sistemas utilizan autenticación triple (con tres factores): un objeto físico, una contraseña y algún dato biométrico como la huella digital. Este método también puede llamarse autenticación multimodal, ya que es una combinación de varios métodos de autenticación distintos en las que intervienen un elemento que el usuario sabe, como se mencionó anteriormente y otro que el usuario posee o cualquier combinación posible. Por ejemplo el DNI (Documento Nacional de Identidad) es un soporte seguro (que el usuario posee) y además para autenticarse en un sistema necesitará un pin (algo que el usuario conoce). (Info@citel, 2011)

Dentro de todos estos grupos los sistemas más utilizados o reconocidos son los siguientes:

- **Autentificación biométrica:** Consiste en la verificación de identidad de un sujeto, basándose en ciertos elementos morfológicos, que le son inherentes y que sólo se dan en ese sujeto. Es decir rasgos distintivos en una persona (su voz, huella dactilar, etc.) para más tarde ser capaz de comparar esa muestra con otra original, y poder averiguar si son iguales o no. Puede ser unimodal o multimodal.
- **Autenticación basada en tokens:** Son dispositivos para autenticación de usuarios que permiten la portabilidad de certificados. Se conectan al computador mediante USB, lo cual los hace compatible con cualquier sistema operativo y ordenador. Los certificados van protegidos con claves, lo que para su uso hace imprescindible estar en posesión del token y conocer la contraseña. Multimodal basado en algo que el usuario posee y conoce.
- **Autenticación SSO (del inglés Single Sign-On):** Es un procedimiento de autenticación multimodal que habilita al usuario para acceder a varios sistemas con una

sola instancia de identificación. Hay cinco tipos principales de SSO que se muestran a continuación:

- E-SSO (Enterprise SSO)
- Kerberos
- OpenID
- Web-SSO
- Identidad Federada

En el caso de autenticación a una aplicación web, que puede efectuarse desde cualquier dispositivo incluyendo dispositivos móviles, las diferentes pruebas se pueden utilizar con más o menos facilidades. La forma más sencilla es preguntar a la persona algo que ella debe saber. En este caso, puede responder con el teclado y no es necesario utilizar otros aparatos como en el caso de las tomas de huellas. Es el modo que se utiliza en este proyecto, se pregunta al usuario su identificador y una contraseña que únicamente él sabe y permite autentificarle. (Info@citel, 2011)

Luego de realizar un breve estudio de forma general sobre uno de los aspectos más importantes en la seguridad de las aplicaciones como lo es la autenticación, es elemental reconocer que dentro de los sistemas de autenticación mencionados es inevitable no realizar un análisis más profundo de la autenticación única, por las ventajas que presenta este proceso, por lo que forma parte de la presente investigación, además de los diferentes protocolos que se pueden utilizar para llevarlo a cabo como por ejemplo: SAML, OpenID, CAS y PAPI, que además participan en la federación de identidades, exceptuando CAS que se basa única y exclusivamente en la autenticación única. A continuación se muestra una explicación más detallada de los procesos de autenticación única y federada y de los protocolos que la realizan.

1.4.1 Modelo de Autenticación Única

El inicio de sesión única más conocida por Single Sign-On (SSO) por sus siglas en inglés consiste en la autenticación única por parte del usuario para acceder a sus recursos.

El objetivo de este proceso es introducir una sola vez el nombre de usuario y contraseña sin necesidad de volver a introducirlo en el momento de acceder a nuevos recursos en los que aún no se había autenticado.

El inicio de sesión única web (Web-SSO) trabaja únicamente con aplicaciones y recursos vía web. Los accesos son interceptados con la ayuda de un servidor proxy o de un componente instalado en el servidor web destino. Los usuarios no autenticados que intentan acceder son redirigidos a un servidor de autenticación o como también puede llamarse proveedor de identidad, que se trata de un sistema que expide información sobre un sujeto determinado. Un Proveedor de Identidad puede asegurar, entre otras cosas, que un usuario ha sido autentificado y que posee determinados atributos, como por ejemplo una dirección de email, un puesto determinado dentro de una organización y regresan sólo después de haber logrado un acceso exitoso.

También otro componente fundamental para llevar a cabo este proceso es el proveedor de servicios que es un sistema donde residen los recursos a los que el usuario desea acceder y que confía en la información que le proporciona el Proveedor de Identidad. Sobre él recae la responsabilidad última de la decisión de acceso a un recurso propio. Por lo tanto, aunque depende de la información que le llega desde el Proveedor de Identidad, es autónomo en cuanto a la decisión de acceso.

Otras de las características que posee este modelo es que es multiplataforma, ya que facilita las tareas de inicio de sesión y de acceso a recursos de red desde diversas plataformas. El acceso a los recursos de sistemas lo realiza de manera transparente al usuario dado a la automatización del inicio de sesión.

También brinda facilidades de uso, pues el usuario se autentica una sola vez y el sistema le permite acceder a los recursos para los cuales está autorizado. De esta manera se evita las interrupciones realizadas por la solicitud de usuario y contraseña para el acceso a diversos recursos. Además proporciona una gestión sencilla, ya que la utilización de SSO recomienda la sincronización de contraseñas e información de los usuarios. Esto trae consigo la simplificación de la gestión de los recursos por parte de los administradores.

El control de acceso no muestra afectaciones por la aplicación de este sistema, pues SSO implica modificar los mecanismos de autenticación del cliente y/o servidor, pero no cambia los permisos de los recursos y en la seguridad depende de la arquitectura utilizada, pero en todos los casos la información viaja cifrada por la red (mediante SSL, certificados). (UPC, 2010)

1.4.2 Modelo Autenticación Federada

Las características básicas que conforman un modelo federado son: la identidad y la privacidad. El modelo federado consiste en que se puedan asociar identidades a usuarios, recursos o componentes del sistema, y que estas identidades sean reconocidas por los diversos componentes que constituyen parte de la federación y valgan para poder identificar de manera unívoca a estos componentes. Por ejemplo, a un usuario mediante un proceso de autenticación se le asocia una identidad y dicha identidad le servirá para ser identificados frente a recursos y otros componentes de la federación, que no es más que el conjunto de tecnologías y normas que permiten a organizaciones asociadas (círculo de confianza) compartir de forma segura información de las identidades digitales que custodian y gestionan en sus diferentes dominios, asegurando la privacidad de los usuarios.

Es importante conocer que dentro de la federación, las organizaciones pueden realizar dos roles que se muestran a continuación:

- Proveedor de identidad: Es el caso de una organización que se encarga de la autenticación de sus usuarios. También puede conservar un repositorio con información complementaria sobre el usuario, que en ocasiones resulta ser válido para el acceso a recursos que su política de filtrado solicita dicha información.

- Proveedor de servicios: Se trata de organizaciones que brindan servicios a usuarios que pertenecen a las organizaciones de la federación. Dichos servicios, generalmente, presentan una política de acceso o filtrado que se implementa mediante un conjunto de reglas de acceso.

La identidad dentro de la federación debe ser válida dentro de ésta y no obligatoriamente hay que asociar a un usuario una identidad que quebrante su privacidad. Dentro de una federación pueden tratarse identificadores que resultan válidos como una identidad dentro de la federación, pero que sólo el proveedor de identidad puede asociar al usuario real. Así de esta manera se asegura la privacidad del usuario y al mismo tiempo, ante un caso de abuso en un servicio, se puede trazar el usuario que realizó dicho acceso.

Otras de las características básicas que debe de haber en un sistema federado son las relaciones de confianza a varios niveles como:

- Entre organizaciones: Debe de haber confianza en el momento de enviar datos sobre usuarios (por ejemplo que sean necesarios para desarrollar una determinada regla de control de acceso), y acerca del uso que la otra organización le dará a ellos. Así mismo debe existir confianza a la hora de recibir información desde otra organización, en el sentido de que la información recibida se considere cierta.

- Entre elementos del sistema: En el sentido de que un elemento debe confiar en que cuando implanta una comunicación con un segundo elemento, éste es quien dice ser y además la información intercambiada es cierta y completa.

- Coherencia de atributos: Tiene que existir una coordinación entre las diversas organizaciones que forman la federación para obtener que la información que una organización presenta sobre un usuario es coherente al nivel sintáctico y semántico con las políticas de acceso que implementan los proveedores de servicios dentro de la federación.

- Criterio de buen uso: Es aconsejable para conservar la confianza entre los usuarios, proveedores de identidad y los proveedores de servicios, que exista una mínima serie de reglas referidas a la utilización de la federación.

- Criterio de incorporación: Es recomendable que se defina un conjunto de criterios y procedimientos para la agregación de nuevos miembros a la federación, tanto nuevos proveedores de identidad, como nuevos proveedores de servicio. Esto permite mejorar la confianza entre las distintas entidades de la federación, y facilita la gestión de la federación. (Rojo, 2010).

1.4.3 Protocolos de autenticación

1.4.3.1 SAML

SAML (Security Assertions Markup Language) es un estándar abierto basado en XML que sirve para la comunicación de la autenticación de usuarios, autorización y la información de atributos. También permite a una entidad empresarial intercambiar aserciones sobre la identidad, atributos y autorización de un usuario (una entidad que generalmente es un humano). Especialmente este intercambio sucede entre el proveedor de identidad (que es el que genera

las aserciones) y el proveedor de servicios (que es quién consume las aserciones). (Pérez, 2011)

SAML se convirtió en estándar de OASIS en noviembre de 2002 y actualmente su última versión es la SAML 2.0. Este estándar ha tenido un potencial éxito en la industria informática. También ha sido utilizado por todos los proveedores de web más reconocidos en la gestión de acceso. Además es compatible con los primordiales productos de servidor de aplicaciones y soporte de SAML. Es muy frecuente entre las web de gestión de servicios y proveedores de seguridad. (C. Gutiérrez, 2005)

Es importante saber que tanto el proveedor de identidad como el proveedor de servicios son entidades fundamentales que forman parte del protocolo SAML y que trabajan simultáneamente para efectuar el mecanismo de autenticación de un usuario. Por lo que es necesario conocer la función que realiza cada una de estas entidades para lograr este mecanismo y que se muestran a continuación:

- Proveedor de identidad: Sistema que envía información sobre un sujeto específico. Además puede garantizar que un usuario ha sido autenticado y que presenta determinados atributos, como por ejemplo una dirección de email, un puesto específico dentro de una organización, etc.
- Proveedor de servicios: Sistema donde se encuentran los recursos a los que el usuario quiere acceder y que confía en la información que le brinda el proveedor de identidad. Este es el responsable de decidir si el usuario accede o no a un recurso propio. Por lo que aunque depende de la información que le envía el proveedor de identidad, es libre de tomar la decisión de acceso.

Este estándar presenta dos escenarios que son: el Control de Acceso Único (SSO) en navegadores web, que es uno de los problemas más importantes que trata de resolver y el otro es la Federación de Identidades. (GIL, 2011).

SAML no establece cómo definir políticas de acceso, pero dentro del protocolo que define contempla la posibilidad de intercambiar mensajes de autorización, los cuales podrían contener estructuras basadas en XACML.

Además un protocolo SAML describe cómo ciertos elementos SAML (incluyendo afirmaciones) son empaquetados en solicitud SAML y elementos de respuesta, y da las reglas de

procesamiento que las entidades SAML deben seguir cuando se producen y consumen estos elementos. En su mayor parte, un protocolo SAML es un simple protocolo de petición-respuesta.

También en el protocolo que define SAML presenta varias implementaciones como son: SimpleSAMLphp, Shibboleth (OpenSAML), Lasso, OpenSSO, Ping Identity, ZXID y el Authentic de los cuales se enunciará una breve descripción de los mismos a continuación.

- SimpleSAMLphp: Es una aplicación simple escrita en PHP que da solución al problema de la autenticación. Además soporta diversos escenarios de federación, mecanismos de autenticación y puede ser utilizada tanto para autenticación local, como proveedor de servicios (SP) o proveedor de identidad (IdP).

- Shibboleth (OpenSAML): Primeramente aclarar que el sistema Shibboleth es un paquete software basado en estándares y de código abierto que permite implantar el Control de Acceso Único. Además permite a los sitios tomar decisiones informadas acerca de la autenticación de accesos a usuarios a los recursos protegidos de una manera que mantiene la privacidad.

Luego de enunciar una breve descripción del sistema Shibboleth en OpenSAML que no es más que un componente básico de éste, una serie de librerías Java y C++ de código abierto que pueden ser aplicadas para crear, transportar y analizar mensajes SAML. OpenSAML es capaz de guardar individualmente los campos de información los campos de información que conforman un mensaje SAML.

Brinda además soporte adicional para la implementación de sistemas web de Control de Acceso Único, que apliquen perfiles SAML que impliquen la utilización de un browser, redirecciones y mensajes Post.

OpenSAML está diseñado para ser extensible y poder constituir una amplia gama de modelos de confianza y requisitos de seguridad, aunque por ahora se enfoca principalmente a transacciones protegidas mediante Infraestructura de Llave Pública.

- Lasso: Se trata de una librería desarrollada en C y que proporciona bindings para Java, Perl, Python y PHP. Además esta librería soporta Liberty ID-FF 1.2, ID-WSF y SAML versión 2.0, pero no se ha utilizado debido a malas experiencias anteriores y se ha utilizado la librería ZXID por ésta.

- OpenSSO: consiste en una serie de librerías de “Open Federation” e implementaciones de “SP Interfaces”. Además ofrece una lista de interfaces para el proveedor de servicios en cada aplicación estándar para satisfacer diversas necesidades de despliegue y también define la integración con la autenticación existente, la configuración, la sesión, el registro y la infraestructura de almacenamiento de datos.

- Ping Identity: es una integra plataforma de identidad de seguridad diseñada para cumplir con cualquier gestión de identidad en una organización estipulada.

- ZXID: Primordialmente implementa un SP en SAML 2.0. Es software libre y está implementado en C, pero soporta Perl, PHP, y Java. Además soporta otros protocolos como: D-FF, ID-WSF y WS-Fed.

- Authentic: es un proveedor de identidad, el cual trata una extensa gama de necesidades, desde simples configuraciones hasta tareas más complejas. Realmente es un software libre que implementa un proveedor de identidad. Aplica la librería de Lasso y está certificado por el consorcio Liberty Alliance. ZXID recomienda utilizar este proveedor de identidad. (EstandaresyTecnologiasDeFederacionDeIdentidades, 2011)

1.4.3.2 OPENID

OPENID es un protocolo abierto que su primera versión fue definida en el año 2005 para su utilización en el sitio web LiveJournal.

OPENID es una tecnología de código abierto que brinda un protocolo para que los usuarios de la web puedan registrar su identidad en un proveedor de identidad y después utilizar esa identidad en cualquier sitio de la web que soporte este protocolo. Esto quiere decir que, como usuario, no tiene que continuar creando y recordando nombres de usuarios y contraseñas en cada sitio que quiera acceder. Lo único que necesita es un identificador creado en un servidor que identifique OPENID, llamado proveedor de identidad.

Con el uso de OPENID se puede poner en marcha dentro de una organización la solución de Control de Acceso Único, el cual es uno de los escenarios de este protocolo, lo cual quiere decir que el usuario se autentica una vez en su proveedor de identidad y luego tiene acceso a todas las herramientas que soportan una autenticación con OPENID y otro de sus escenarios es la identificación en sistemas federados para el cual se recomienda más, el uso de éste.

El esquema OPENID lo conforman un conjunto de componentes que trabajan conjuntamente para realizar el proceso de autenticación que se detallará más adelante en la propuesta de guía de la presente investigación, pero antes se debe conocer qué función realizan estos componentes que se muestran a continuación:

- Propietario del recurso: Representa al recurso web a proteger. Este componente solicita del esquema OPENID una identificación del usuario que está tratando de acceder al recurso.

- Web de identificación: Esta web se corresponde con la URL que el usuario le proporciona al propietario del recurso para que, desde entonces comience los pasos que concluyan en una exitosa identificación del usuario.

- Proveedor OPENID: Es el servidor responsable de verificar si el usuario está correctamente autenticado y de informar de forma fiable este suceso al propietario del recurso. (ROJO, 2010)

Este protocolo también presenta varias implementaciones, las cuales son:

- PhpMyID: Es una aplicación que realiza la funcionalidad de un servidor OpenID, es fácil de instalar y de configurar. Sólo hay que editar unas pocas líneas en su fichero de configuración y adicionar todos los datos del perfil de usuario en el propio código y ya terminó la configuración, nada más faltaría acceder a la URL para que funcione. El primordial problema, es que sólo se puede aplicar para un sólo usuario y presenta poca seguridad debido a que la utilización de la autenticación es mediante http.

- Clamshell: Proveedor OpenID que permite la gestión de varias identidades OpenID desde una sola página web. Procede de un conjunto de un conjunto de fuentes, sobre todo PhpMyID, las bibliotecas OpenID JainRain y la utilización de Drupal 6.

  Los requisitos del servidor son:

  - Servidor http como Apache (recomendado)

  - PHP 4.3

- WSO2: WSO2 Identity Server, es un servidor de administración de código abierto. Es compatible con tarjetas de información y con autenticación OpenID. Provee una

completa solución de identidad que se integra fácilmente en los servicios de usuarios existentes, tales como LDAP o Active Directory y admite la autenticación de múltiples factores y mucho más.

En la versión se incluye OpenID y tarjetas de información, optimizando aún más la solución de identidad WSO2 para atender a un público más extenso para la autenticación basada en la web. OpenID es un elemento clave en la descentralización de Control de Acceso Único, muy favorecido por muchos usuarios. (GIL, 2011)

1.4.3.3 PAPI

PAPI (Punto de Acceso a Proveedores de Información) es una solución de seguridad concebida, desde un principio, para dar respuesta a este tipo de sistemas abiertos. Su principal característica es su arquitectura distribuida que le permite posicionar sus elementos y procesos en diferentes puntos de la red independientemente de su localización. Esta característica le da una gran capacidad de integración en entornos multi-organización, ya que permite distribuir sus funciones, y simplifica la gestión global de la infraestructura de seguridad y potencia su escalado.

PAPI es una tecnología que te permite desplegar una infraestructura de autenticación y autorización y un sistema de SSO fácilmente.

Su primordial objetivo es conservar la autenticación como un asunto local a la organización a la que pertenece el usuario, mientras que los recursos efectuarán el proceso de autorizar los accesos de dichos usuarios.

El protocolo que implementa el sistema PAPI con su última versión 1.0 constituye dos fases: autenticación y control de acceso. La fase de autenticación inicia cada vez que el usuario accede al servidor de autenticación (AS) para obtener las claves temporales. Luego de obtenidas las claves no es necesario regresar a esta fase siempre que las claves sean válidas. A no ser en determinados casos que el usuario debe autenticarse de forma prematura. Por otro lado el esquema de control de acceso es claro para el usuario, compatible con la generalidad de navegadores y no solicita ningún hardware o software adicional. El sistema PAPI cuenta con cuatro componentes distintos:

- Servidor de autenticación (AS): También conocido como proveedor de identidad es el responsable del proceso de autenticar al usuario, validando su identidad y

relacionándole un conjunto de atributos para que cuente con ellos en los recursos protegidos.

- Punto de acceso (PoA): Este componente, llamado como proveedor de servicio, efectúa la autorización del acceso a un recurso protegido verificando la autenticación del usuario y sus atributos.
- Grupo de puntos de acceso (GPoA): Permite concentrar las políticas de autorización de una organización en un solo punto, al cual preguntarán los proveedores de servicio de esta organización.
- Proxy PAPI: Es un proxy http con re-escritura de los enlaces con respecto a un recurso web externo. De esta manera los recursos que sólo permiten autenticación por IP pueden integrarse también en una infraestructura de autenticación y autorización.

Las primordiales características del protocolo PAPI son:

- Infraestructura completa para desplegar un sistema de SSO dentro de una organización o entre distintas organizaciones, proveyendo la tecnología necesaria para una federación de identidad digital.
- Proxy web con re-escritura de enlaces http, admitiendo el acceso controlado a recursos externos que nada más cuentan con mecanismos de autorización básicos como el control por IP.
- Protocolo abierto, ligero y documentado.
- Fácilmente interoperable con otros protocolos de autenticación y autorización como: SAML 1.1, SAML 2, OpenID y OAuth.
- Software abierto y disponible en diversos lenguajes de programación como: Perl, PHP, Java, ASP.NET, etc.
- Multitud de conectores que se cuentan para proveedores de servicio: MediaWiki, DokuWiki, Moodle, etc.

Dentro de las implementaciones que este protocolo presenta se encuentra: phpPoA que es una implementación muy ligera del protocolo de federación PAPI que nos brinda la posibilidad de conectar la gestión de identidad en aplicaciones escritas en el lenguaje PHP de forma sencilla y descentralizada.

En concreto, PAPI es el protocolo que se aplica en la federación española SIR (Servicio de Identidad de RedIRIS). Mediante la librería phpPoA conectar una aplicación con la federación se reduce a algo tan simple como un par de líneas de código PHP.

Otra de las implementaciones actuales de este protocolo es: PHP easyGPoA: es un GPoA implementado completamente en PHP y muy simple. El objetivo es proporcionar un componente de este tipo de funcionalidad para aquellos casos que no solicitan una complejidad mayor.

Sus características fundamentales son:

- GPoA completamente funcional implementado en PHP.
- Define que Proveedores de identidad PAPI permite.
- Filtro de atributos por PoA, permitiendo definir qué atributos se deben enviar a cada uno de los PoAs en los que confía.

La implementación PHP icGPoA: es una re-implementación completa del icGPoA implementado en PHP de forma que tengamos uno modular que sea más fácil de conservar entre todos los usuarios de este componente. De esta manera, las adaptaciones locales están separadas del código principal, por lo que a partir de ahora será más fácil mantener nuestro icGPoA a la última versión oficial. (Rojo, 2011)

1.4.3.4 CAS

CAS (Servicio de Autenticación Central) es un sistema de autenticación elaborado originalmente por la Universidad de Yale para brindar una forma de confianza en una aplicación para autenticar un usuario. CAS se convirtió en un proyecto JASIG en diciembre de 2004.

CAS es un protocolo de inicio de sesión única para la web. Su principal objetivo es permitir al usuario acceder a múltiples aplicaciones al mismo tiempo que va proporcionado sus credenciales (como nombre de usuario y contraseña) sólo una vez. También permite que los usuarios se autentiquen para entrar a las aplicaciones web sin tener que acceder a las credenciales de seguridad del usuario, como por ejemplo una contraseña.

El protocolo CAS implica al menos tres partes: un navegador web del cliente, la autenticación de aplicaciones web solicitante y el servidor CAS. También puede consistir en un servicio de

back-end, tales como un servidor de base de datos, que no tiene su propia interfaz HTTP, pero se comunica con una aplicación web.

Cuando el cliente entra en una aplicación que desee autenticarse en ella, la solicitud se re-direcciona a CAS. CAS ratifica la autenticidad del cliente, por lo general chequeando un nombre de usuario y la contraseña en una base de datos (por ejemplo, Kerberos o Active Directory).

Si la autenticación se realiza correctamente, CAS devuelve el cliente a la aplicación, pasando a lo largo de una ficha de seguridad. La aplicación valida el billete en contacto con CAS mediante una conexión segura y proporcionar su identificador propio, servicio y el boleto. CAS da la información de aplicaciones de confianza acerca de si un usuario se ha autenticado correctamente.

Además CAS es un protocolo abierto y bien documentado. Presenta una biblioteca de clientes de Java, .Net, PHP, Perl, Apache, U portal, entre otros. También tiene una amplia comunidad de adoptantes y se integra con uPortal, Bluesocket, TikiWiki, Mule, Liferay, Moodle y otros.

Nota: Algo que tiene que quedar muy claro es que CAS se encarga única y exclusivamente de la autenticación es decir, de comprobar contra una fuente de datos específica si el usuario y contraseña facilitados existen, no se encarga de la autorización, que sería la gestión de lo que puede o no puede hacer ese usuario en función de sus roles. (Community, 2009)

1.5 Autorización

Otro de los aspectos fundamentales en el desarrollo de aplicaciones web es la autorización, hecho de dejar una entidad, tener acceso o no a una funcionalidad o un servicio. En el caso de una persona, se basa en la identidad de una persona autenticada. La autorización es un elemento imprescindible a tener en cuenta en la implementación de los sistemas de información. Por eso la industria de la información ha desarrollado una serie de protocolos y modelos de autorización para garantizar la seguridad en estos mecanismos y se muestran alguno de ellos en los siguientes epígrafes.

1.5.1 Autorización centralizada

Las organizaciones necesitan políticas de seguridad flexibles que puedan utilizarse fácilmente en varios servicios y aplicaciones. Deben implementar un único servicio de seguridad

compartido para simplificar la administración y la elaboración de informes relativos al cumplimiento, y reducir la carga asociada a la seguridad de los desarrolladores de aplicaciones. Por tanto, sin sistemas de gestión de accesos a la web, los desarrolladores de sistemas deberían implementar una lógica de seguridad completa en sus aplicaciones. Esto conllevaría riesgos y complicaciones para el cumplimiento. Por ejemplo, las técnicas de desarrollo de seguridad necesarias dependerían del tipo de servidor Web, sistema operativo y lenguaje de programación utilizados para la aplicación.

Para ello se centraliza la gestión de las atribuciones de usuario de los clientes, partners, los cuales serían en este caso los que actuarían como proveedores de identidad y empleados en todas las aplicaciones Web mediante un servicio compartido. La autorización centralizada reduce en gran medida los costes de desarrollo, lo que permite a los desarrolladores centrarse en la lógica empresarial de la aplicación en lugar de en las políticas de seguridad de programación. Además, logra brindar la posibilidad de aplicar políticas de seguridad en toda la empresa, lo que elimina la necesidad de directorios redundantes de usuarios y lógicas de seguridad específicas de cada aplicación. (SiteMinder®, 2011)

1.5.2 XACML

XACML **(**Extensible Authorization Markup Language**)** lo que en español quiere decir control de lenguaje extensible de marcado de acceso se define como un lenguaje, basado en XML para definir políticas de control de acceso. Lo que implica que brinda una sintaxis para gestionar el acceso a los recursos por parte de determinados sujetos que quieren efectuar ciertas operaciones sobre ellos.

El objetivo de la especificación XACML es promover un mecanismo unificado de control de acceso precisando un lenguaje que sea capaz de expresar información de autorización de forma flexible y extensible de forma tal que pueda aplicarse a una extensa variedad de sistemas y dispositivos.

Este lenguaje que define XACML basado en XML es utilizado para los mensajes de autorización, el cual establece la estructura de los mensajes de pedido de acceso así como la respuesta a dichos mensajes.

Además se puede decir que XACML es un estándar que detalla tanto un lenguaje de políticas como un protocolo de petición/respuesta para el control de las decisiones de autorización.

Ambos, lenguaje y protocolo, están definidos en XML. El lenguaje de políticas XACML se usa para detallar requisitos de control de acceso generales. Este estándar también presenta unos puntos de extensión estándar que aprueban la definición de nuevas funciones, tipos de datos, combinaciones lógicas, etc. El protocolo petición/respuesta nos permite conformar una petición para requerir si determinada acción debería, o no, ser permitida, además de proporcionarnos maneras de interpretar las decisiones de control. La respuesta siempre contiene una contestación que muestra si la petición debería ser o no admitida, usando uno de los siguientes valores: Permit, Deny, Indeterminate (no se pudo tomar una decisión porque ocurrió un error o fue omitido algún valor solicitado) o NotAplicable (la petición no puede ser respondida por este servicio). Estos valores están definidos por la propia especificación. (C. Gutiérrez, 2005)

Los escenarios típicos que se ven mayormente en la autorización aplicando el protocolo XACML son: la implementación de políticas de autorización para sistemas existentes y subsistemas con lenguajes de autorización incompatibles.

Una de las implementaciones que presenta el protocolo de autorización XACML es Sun Microsystems como miembro activo del conjunto de estándares de OASIS, decidió hacer su propia implementación de XACML con el fin de apresurar su adopción. La implementación fue realizada en código abierto en el lenguaje de programación Java. (Esponda, 2007)

1.5.3 OAUTH

OAuth (Open Authorization) es un protocolo abierto y estándar que comenzó a utilizarse en noviembre del 2006 y que permite a un sitio web A acceder de un modo seguro, previa autorización del usuario, a datos de acceso limitado de dicho usuario almacenados en otro sitio web B mediante una API que soporta OAuth y que B pone a disposición de A.

La versión más actual de este protocolo es 2.0 y no es compatible con OAuth 1.0. OAuth 2.0 se centra en la simplicidad del cliente desarrollador mientras que provee flujos de autorización específicos para las aplicaciones web, aplicaciones de escritorio, teléfonos móviles, dispositivos, etc.

El aspecto más importante en cuanto a la seguridad que brinda OAuth es que el sitio web A (OAuth Consumer) no almacena los credenciales de acceso (usuario y contraseña) que el usuario utiliza en su cuenta en el sitio web B (Service Provider).

También se puede decir que OAuth es una forma encilla, segura y rápida de publicar y acceder a datos protegidos (fotos, videos, lista de contactos). Se trata de un modelo de autorización abierta basada primordialmente en las normas vigentes que aseguran las credenciales de seguridad pueden ser proporcionados y comprobados por las plataformas de software diferentes.

En otras palabras, OAuth permite compartir los recursos privados almacenados en un sitio con otro sitio sin tener que repartir su nombre de usuario y una contraseña.

Para poder hablar de OAuth es fundamental presentar tres elementos que son: un servidor o proveedor de servicios, usuarios y un consumidor y de los cuales se muestra un breve resumen del funcionamiento de éstos a continuación:

- OAuth Service Provider o proveedor de servicios OAuth: son sitios o servicios web que almacenan información de usuarios cuyo acceso es limitado. Algunos de los más conocidos son Facebook, Twitter o Youtube. Estos proveedores ponen a disposición de los desarrolladores una API que soporta el protocolo de autenticación OAuth.
- Usuarios: Sin los usuarios, no existiría OAuth. Por usuario se entiende cualquier persona que tiene una cuenta de usuario en un Service Provider.
- OAuth Consumer o consumidor OAuth: es cualquier sitio o aplicación web, móviles o de escritorio que requiere el permiso de un usuario para acceder a sus datos de acceso limitado que almacena un Service Provider. El usuario puede autorizar o denegar el acceso del consumer a sus datos. Es necesario que el consumer soporte el protocolo HTTP y que utilice la versión de OAuth que el Service Provider haya implementado. (YDN, 2012)

Una de las implementaciones que éste proporciona es la extensión PECL OAuth de PHP. Además OAuth posee un conjunto de librerías entre los que se encuentra: OAuth-PHP, Zend Framework y Extensión PECL para PHP, pero la más aplicada es la librería oauth-php. (Hammer-Lahav, 2007)

1.6 Conclusiones

El énfasis de este capítulo ha estado en profundizar los conceptos fundamentales relacionados con el dominio del problema como fueron: la Seguridad Informática, ya que presenta dentro de sus objetivos garantizar la autenticación y la autorización en los sistemas de información. Además de los aspectos relacionados con la Seguridad en Internet, que también deben cumplir los mismos objetivos que poseen la Seguridad Informática y la Seguridad en las Aplicaciones Web, ya que este tema es especialmente válido en el entorno de Internet, como bien se mencionó anteriormente debido a que la mayoría de las aplicaciones web son accedidas a través de Internet.

Por estas razones, este estudio de los protocolos existentes más usados que tratan directamente los temas de la autenticación y la autorización, puede contribuir al desarrollo de la propuesta de solución de la presente investigación, la que consiste en proponer una guía para el diseño de los mecanismos de autenticación y autorización en las aplicaciones web, para potenciar en gran medida la seguridad de la implementación de las aplicaciones.

También a partir del estudio bibliográfico se identificaron algunos de los modelos de autenticación, que sirven de gran ayuda para la realización de la propuesta, dada a las amplias posibilidades que brindan, que contribuyen al diseño de los diferentes mecanismos.

Capítulo 2: Propuesta de Solución

2.1 Introducción

La mayoría de los problemas que se generan en las aplicaciones web desarrolladas actualmente se ven en las técnicas de autenticación y autorización utilizadas en los sistemas. De ahí que gran parte de las vulnerabilidades a que son expuestos los sistemas de información son originadas generalmente por estos aspectos. Por eso es fundamental garantizar un óptimo diseño de los mecanismos de autenticación y autorización que velen por la seguridad de los sistemas.

De ahí la importancia de poder contar con una guía, que explique detalladamente las actividades que se deben efectuar para llevar a cabo estos mecanismos.

En el presente capitulo se propone una guía con todo el flujo de actividades que se deben cumplir para realizar el diseño de los mecanismos de autenticación y autorización en las aplicaciones web.

2.2 Mapa de la Guía

A continuación se muestra el flujo de actividades que se deben cumplir para realizar el diseño de los mecanismos de autenticación y autorización:

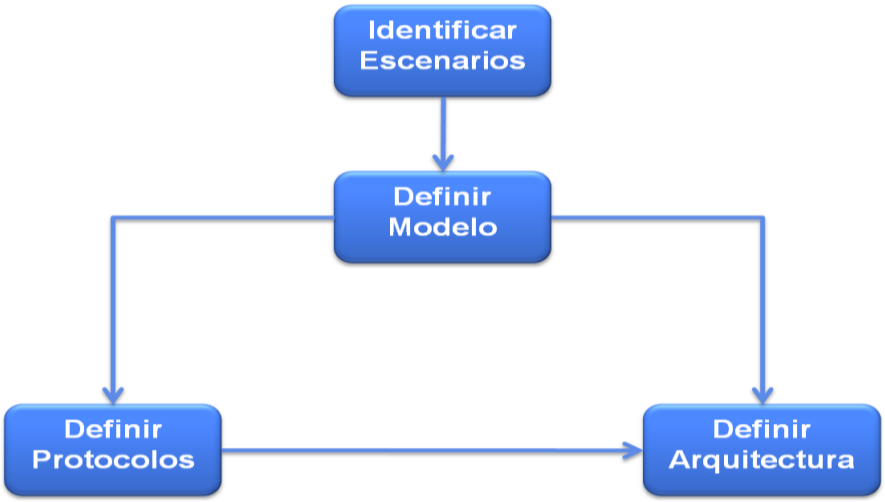

Figura 1: Flujo de actividades a seguir para el diseño del mecanismo de autenticación y autorización.

2.3 Descripción del proceso

En el presente trabajo se define una guía para la realización del diseño de los mecanismos de autenticación y autorización basados en los protocolos más usados de seguridad que tratan estos temas, los cuales se fundamentaron en el capítulo anterior.

El modelo que se propone surge de la necesidad de contar con una guía que te ayude a mostrar el proceso del diseño de los mecanismos de autenticación y autorización, y para ello se tuvo en cuenta el análisis del funcionamiento de los protocolos que tratan la autenticación y la autorización para elaborar estos mecanismos. A partir de aquí se definen los pasos a seguir en la realización de las actividades mencionadas anteriormente para la elaboración del diseño del mecanismo propuesto.

Como se observa en la figura 1, para diseñar los mecanismos de autenticación y autorización que propone la presente guía, hay un diagrama de flujo de cuatro actividades fundamentales que siguen un orden lógico como se muestra a continuación.

Primeramente se realiza la Actividad 1 Identificar escenarios, la cual depende de los requisitos de autenticación y autorización que presente el sistema donde se va aplicar la guía, los que deben ajustarse a las condiciones de alguno de los escenarios existentes que se exponen más

adelante en la explicación de la Actividad 1. Seguidamente una vez culminada la Actividad 1, se procede a definir el modelo que es la Actividad 2, la cual es dependiente del escenario identificado, es decir, si el escenario que se identificó es la Autenticación Única, el modelo a definir será el Modelo de Autenticación Única y así si se identificará un nuevo escenario. Luego seguiría la Actividad 3 Definir Protocolo, ya en esta actividad hay que tener en cuenta que según el modelo definido anteriormente, qué protocolos de los analizados en la presente investigación participan en ese modelo. Como es el caso, por ejemplo del modelo de autenticación única que puede realizar todos los protocolos analizados en el presente trabajo y se definirá el protocolo que más se adapte a los requerimientos que presente el sistema donde se va aplicar la guía. Por último sigue la Actividad 4 Definir Arquitectura, la cual depende del modelo y protocolo definido con anterioridad, ya que teniendo conocimiento de cuáles fueron éstos, se podrá saber que componentes va a utilizar la arquitectura definida.

En la actividad Identificar escenarios es donde se presenta más peso, porque es la actividad fundamental de la cual se derivan las demás en un orden consecutivo, ya que en dependencia de la identificación del escenario, se tendrán en cuenta el modelo y el protocolo a definir, lo que proporcionará como resultado la arquitectura definitiva del sistema de aplicación.

2.4 Descripción de las actividades de la guía para realizar el proceso

2.4.1 Actividad 1: Identificar escenarios.

Actividad 1 Identificar escenarios, la cual depende de los requisitos de autenticación y autorización que presente el sistema donde se va aplicar la guía, los cuales deben ajustarse a las condiciones de alguno de los escenarios existentes que se muestran a continuación con los pasos más detallados.

Lo primero que se debe tener en cuenta para realizar esta actividad es:

- Identificar los requerimientos de autenticación y autorización de la aplicación donde se va a aplicar el mecanismo. Pues una vez que estos son identificados, se debe tener en cuenta si los requisitos identificados se adaptan a algunas de las condiciones que presentan los siguientes escenarios, de ser así se identificará el escenario que cumpla con esas condiciones para que este sea aplicable.

- Identificar los escenarios de autenticación que son:

- Autenticación Única : Este escenario es aplicable cuando:

    1. Se desea integrar una aplicación a un esquema de autenticación único existente.

    2. Se desea desarrollar un conjunto de aplicaciones que compartan el mismo mecanismo de autenticación.

    3. Se desea actualizar aplicaciones legadas para que formen parte de un esquema de autenticación única.

- Autenticación federada: Este escenario es aplicable cuando:

    1. Se desea integrar aplicaciones de dominios de seguridad diferentes.

    2. Se desea permitir a los usuarios del dominio de seguridad interno la posibilidad de utilizar sus credenciales en dominios de seguridad externos.

➢ Definir los escenarios de autorización que son:

- Autorización no centralizada: Este escenario es aplicable cuando:

    1. La aplicación maneja sus propias políticas y reglas de control de acceso.

- Autorización centralizada: Este escenario es aplicable cuando:

    1. Se desea definir las políticas de seguridad para varias aplicaciones desde un único punto centralizado.

2.4.2 Actividad 2: Definir modelo

Seguidamente una vez culminada la Actividad 1, se procede a definir el modelo que es la Actividad 2, la cual es dependiente del escenario identificado, es decir si el escenario que se identificó es la Autenticación Única, el modelo a definir será el Modelo de Autenticación Única y así sucederá lo mismo con otro escenario.

Para realizar esta actividad debe haberse culminado la actividad anterior, ya que todas las tareas son consecuentes una de otras, por lo que el paso fundamental es:

- Definir el modelo a utilizar en correspondencia al escenario identificado en la actividad 1 como se muestra a continuación en la siguiente tabla:

Escenarios	Modelos
Autenticación Única	Modelo de Autenticación Única
Autenticación Federada	Modelo de Autenticación Federada
Autorización Centralizada	Modelo de Autorización Centralizada
Autorización No Centralizada	Modelo de Autorización No Centralizada

Tabla 1: Definición de Modelos

2.4.3 Actividad 3: Definir protocolo

Actividad 3 Definir Protocolo, ya en esta actividad hay que tener en cuenta, según el modelo definido anteriormente, qué protocolos de los analizados en la presente investigación participan en ese modelo. Como es el caso, por ejemplo del modelo de autenticación única que lo puede realizar todos los protocolos analizados que traten la autenticación única en el presente trabajo, por lo que se debe definir el protocolo que según sus características y ventajas se adapte más a los requerimientos que presente el sistema donde se va aplicar la guía. Además para la definición de éste es necesario tener en cuenta una serie de características que contribuirán en el momento de seleccionar el protocolo a utilizar para el diseño del mecanismo, como son: las implementaciones en software libres que presentan los protocolos y que facilitan el uso de estos, qué organización lo soporta y este aspecto es muy importante, pues hay organizaciones renombradas, que se dedican a desarrollar soluciones como por ejemplo: OASIS (Organization for the Advancement of Structured Information Standards), la integración de un determinado protocolo con otros protocolos de seguridad que se centren en el mismo problema a resolver, como se muestra en la tabla más adelante, soporte comercial que ha tenido el protocolo, es decir la empresas que trabajen con la implantación de estos esquemas o que vendan productos que los utilicen, para de esta forma conocer la aceptación que ha tenido por los usuarios. El análisis de todas estas características pueden ayudar a definir qué protocolo utilizar para el diseño del mecanismo.

- Definir el protocolo a utilizar en correspondencia al modelo definido en la actividad 2 como se muestra en la siguiente tabla:

Modelos	Protocolos
Modelo de Autenticación Única	SAML
Modelo de Autenticación Federada	
Modelo de Autenticación Única	PAPI
Modelo de Autenticación Federada	
Modelo de Autenticación Única	CAS
Modelo de Autenticación Única	OPENID
Modelo de Autenticación Federada	
Modelo de Autorización Centralizada	XACML
Modelo de Autorización No Centralizada	OAUTH

Tabla 2: Definición de Protocolos

2.4.4 Actividad 4: Definir arquitectura

Actividad 4 Definir Arquitectura, la cual depende del modelo y protocolo definido con anterioridad, ya que teniendo conocimiento de cuales fueron estos, se podrá saber que componentes va a utilizar la arquitectura definida. Además en esta actividad es muy importante tener bien definido el protocolo a utilizar, para así tener conocimiento de los componentes que se van a utilizar en la arquitectura que se obtiene como resultado del cumplimiento de todas las actividades que conforman la Guía.

- Definir la arquitectura en correspondencia al escenario, modelo y protocolo definido en las actividades 1,2 y 3 respectivamente.
- Definir en la arquitectura los componentes a utilizar en esta como se muestra a continuación en la tabla:

<table>
<tr><th>Escenarios</th><th>Protocolos</th><th>Modelos</th><th>Arquitectura</th></tr>
<tr><td>Autenticación Única</td><td rowspan="2">SAML</td><td>Modelo de Autenticación Única</td><td rowspan="7">Proveedor de servicios
Proveedor de identidad
Directorio de usuario
Implementaciones del protocolo</td></tr>
<tr><td>Autenticación Federada</td><td>Modelo de Autenticación Federada</td></tr>
<tr><td>Autenticación Única</td><td rowspan="2">PAPI</td><td>Modelo de Autenticación Única</td></tr>
<tr><td>Autenticación Federada</td><td>Modelo de Autenticación Federada</td></tr>
<tr><td>Autenticación Única</td><td>CAS</td><td>Modelo de Autenticación Única</td></tr>
<tr><td>Autenticación Única</td><td rowspan="2">OPENID</td><td>Modelo de Autenticación Única</td></tr>
<tr><td>Autenticación Federada</td><td>Modelo de Autenticación Federada</td></tr>
<tr><td>Autorización Centralizada</td><td>XACML</td><td>Modelo de Autorización Centralizada</td><td>Proveedor de servicios
Implementaciones del protocolo
Directorio de Políticas
Servidor de Políticas</td></tr>
<tr><td>Autorización No Centralizada</td><td>OAUTH</td><td>Modelo de Autorización No Centralizada</td><td>Proveedor de servicios
Implementaciones del protocolo</td></tr>
</table>

Tabla 3: Definición de la arquitectura

A continuación se detallaran 2 entidades fundamentales, que son necesarias tener conocimiento de la función de cada una, para así lograr un mejor entendimiento de las posibles arquitecturas que se obtengan como resultado de la aplicación de la Guía.

Proveedor de servicios (SP): Sistema o dominio administrativo donde residen los recursos a los que el usuario desea acceder y que depende de la información que le proporciona el Proveedor de Identidad. Confiar o no en las afirmaciones que el IdP realiza sobre un usuario es una decisión propia del Proveedor de Servicios, si bien SAML define mecanismos que permiten otorgar dicha confianza. Hay que resaltar que, el hecho de que un SP confíe en las afirmaciones de un Proveedor de Identidad sobre la identidad o los atributos de un determinado sujeto, no significa necesariamente que dicho sujeto pueda acceder a los recursos de dicho Proveedor de Servicios, siendo las políticas locales de acceso sobre las que recae, en último término, la responsabilidad de esta decisión.

Proveedor de identidad (IdP): Sistema o dominio administrativo que expide información sobre un sujeto determinado (generalmente un usuario). Un Proveedor de Identidad puede asegurar, entre otras cosas, que un usuario ha sido autentificado y que posee determinados atributos. Por ejemplo, un IdP podría afirmar que el usuario se llama Paco Teleco, tiene como dirección

de email paco.teleco@telecosunidos.com y que se autentificó mediante la introducción de una contraseña.

Servidor de directorio: Un servicio de directorio (SD) es una aplicación o un conjunto de aplicaciones que almacena y organiza la información sobre los usuarios de una red de ordenadores, sobre recursos de red, y permite a los administradores gestionar el acceso de usuarios a los recursos sobre dicha red. Además, los servicios de directorio actúan como una capa de abstracción entre los usuarios y los recursos compartidos.

Las arquitecturas que se obtienen como resultado de la realización de las actividades en correspondencia con el escenario, protocolo y modelo definido son:

En la presente arquitectura de autenticación única de la figura 2 el usuario intenta acceder a un recurso o aplicación que se encuentra en el proveedor de servicios, que es la entidad encargada de dar acceso al usuario a un recurso o aplicación como se explica anteriormente en la fundamentación teórica de la investigación. Este se comunica con el proveedor de identidad, el cual es el responsable de la autenticación del usuario para validar el acceso de dicho usuario, el cual a su vez se conecta con el servidor de directorio, donde se encuentran registrados los usuarios para verificar la identidad del usuario.

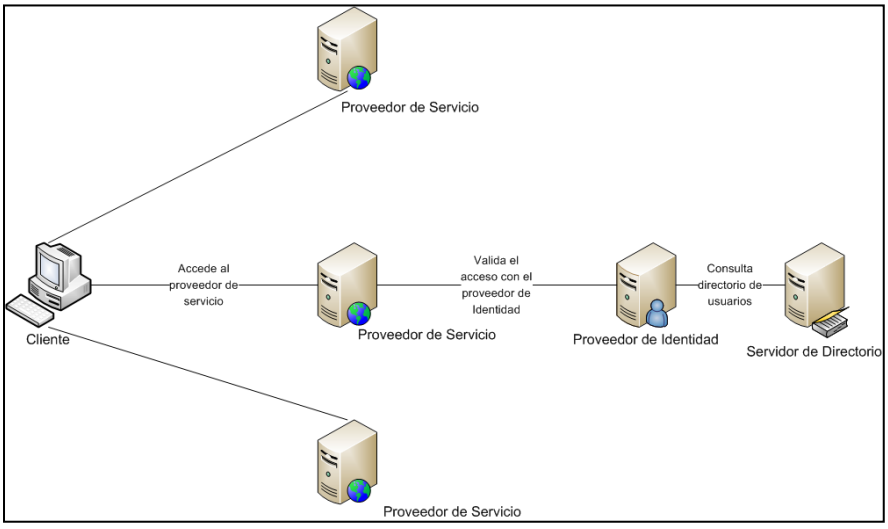

Figura 2: Arquitectura de autenticación única

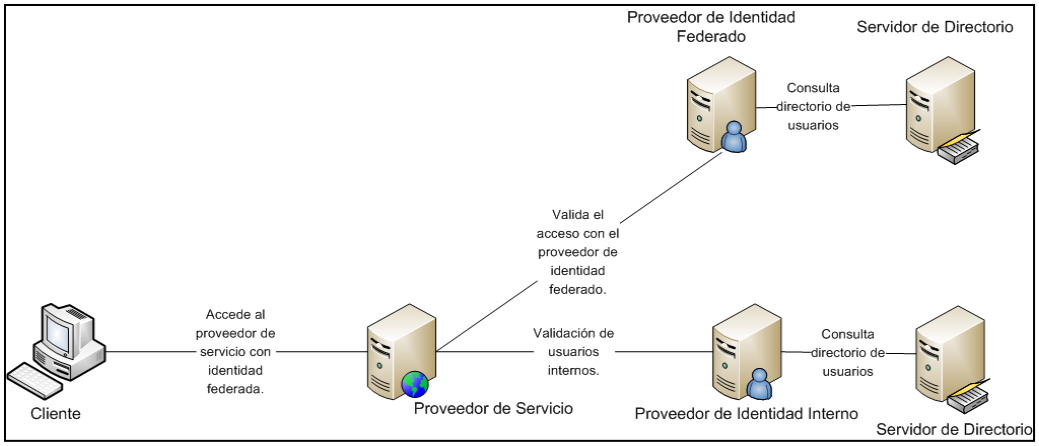

Figura 3: Arquitectura de autenticación federada

En la presente arquitectura de autenticación federada de la figura 3 el usuario de dominio externo que puede ser cualquier usuario de una entidad externa, con un dominio de seguridad diferente al de la entidad, intenta acceder a la aplicación o recurso alojada en el proveedor de servicio, que es la entidad encargada de dar acceso al usuario a un recurso o aplicación, explicado en la arquitectura anterior, el cual valida el acceso con el proveedor de identidad federado también explicado en la arquitectura anterior que a su vez consulta el servidor de directorio de usuarios para comprobar la identidad del usuario.

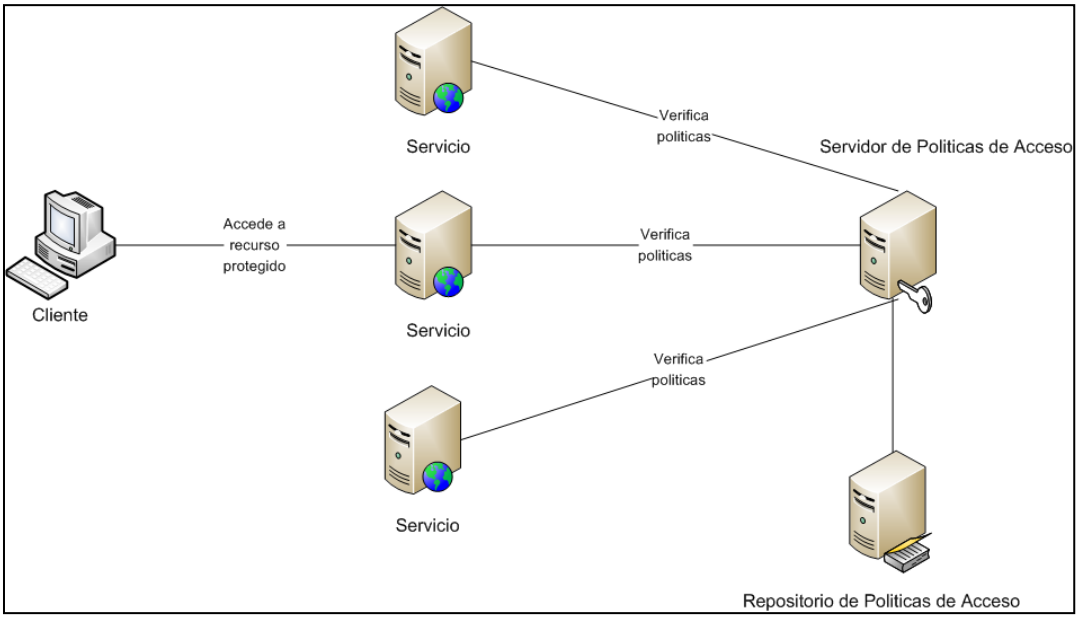

Figura 4: Arquitectura de autorización centralizada

En la arquitectura de autorización centralizada que se muestra en la figura 4 el usuario intenta acceder a un recurso protegido que se encuentra en el proveedor de servicios, donde se encuentra alojado el recurso o la aplicación que quiere acceder, el cual a su vez verifica las políticas con el servidor de políticas de acceso que es el responsable de tomar la decisión de si acepta o no el pedido de autorizo.

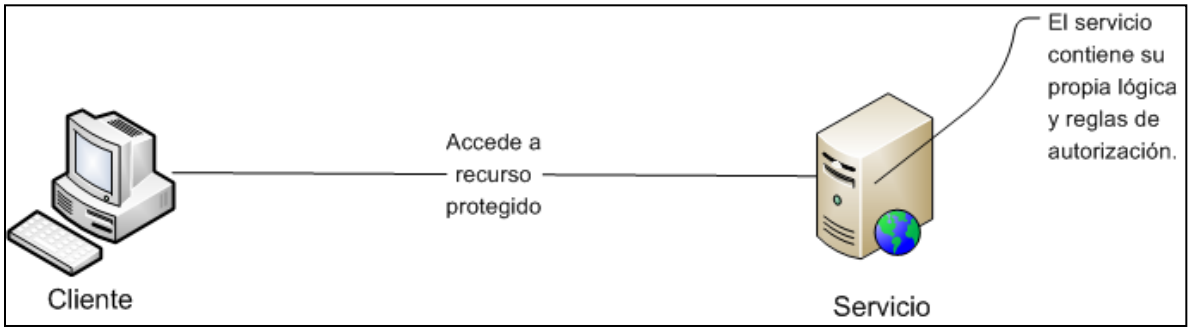

Figura 5: Arquitectura de autorización no centralizada

En la arquitectura de autorización no centralizada el usuario intenta acceder al recurso protegido o la aplicación que se encuentra alojado en el proveedor de servicio, donde se encuentra alojado el recurso o la aplicación que quiere acceder, el cual contiene sus propias políticas y reglas de acceso.

2.5 Conclusiones

Al finalizar este capítulo se ha arribado a la conclusión que la propuesta de guía presentada contribuirá al mejoramiento del diseño de los mecanismos de autenticación y autorización para el desarrollo de aplicaciones web, logrando una aplicación más segura y controlada en cuanto al acceso de la misma, lo que trae consigo la reducción de la mayoría de los problemas generados en los sistemas.

Capítulo 3: Validación de la propuesta de Guía mediante el diseño de un caso de estudio.

3.1 Introducción

En el presente capitulo, se abordará la validación de la propuesta de Guía para el diseño de mecanismos de autenticación y autorización de aplicaciones web. Esta validación será basada en un caso de estudio, que muestra el funcionamiento del protocolo de autenticación SAML, aplicando el modelo de autenticación única SSO.

3.2 Descripción del caso de estudio

La Universidad de Oriente, desea desarrollar la intranet del centro escolar, donde todos los usuarios de la red, encuentren los servicios automatizados que se han implementado para de esta forma lograr la centralización de toda la información que circula y ocurre en la Universidad. Este portal tendrá un boletín informativo, que podrá descargarse en cualquier formato, noticias de lo que acontece en el país y el resto del mundo e información actual de lo que ocurre en la Universidad. Entre otras funciones el usuario tendrá acceso también a todas las aplicaciones que brindan las áreas de la Universidad como el área de Formación, Investigación, Extensión Universitaria, Producción y Organizaciones entre ellas podemos mencionar, el directorio telefónico, el servicio de correo electrónico, eventos, charlas interactivas, sistema de gestión académica, sistema de contabilidad , sistema de gestión de pase, sistema de alimentación, sistema de reservación, entre otros que podrán surgir luego de un levantamiento de requisitos para informatizar la comunidad universitaria. Para la implementación de este portal, hemos realizado el levantamiento de requisitos funcionales y no funcionales, los cuales listamos a continuación:

Requerimientos Funcionales:

- RF1 Autenticar usuario

- RF2 Gestionar sistema contable
- RF3 Gestionar sistema académico
- RF4 Gestionar directorio telefónico

Como requerimientos no funcionales se define:

Requerimiento No Funcional de Seguridad:

- RnF1 La aplicación debe permitir que una vez que los usuarios se autentiquen en la intranet institucional no requieran autenticarse de nuevo en la aplicación.
- RnF2 La aplicación debe ser desarrollada en el lenguaje de programación Java.
- RnF3 El acceso a cualquier manipulación del sistema, tanto entrada como análisis de datos debe estar sometido a un proceso de autentificación por el personal autorizado.
- RnF4 Cada usuario va a tener asignado un rol en el sistema.
- RnF5 Cada rol definido tendrá niveles de acceso al SW.
- Rnf6 Los datos almacenados en la BD solo podrán ser modificados por administradores y profesores con previa autenticación de los mismos.

A continuación se procederá a utilizar las actividades de la guía propuesta una vez obtenidos los requisitos, para ver los resultados que arroja la misma según el caso de estudio enunciado anteriormente. Solo se aplicarán estas actividades al RnF1, a modo de ejemplificar como quedaría el resultado del caso de estudio para este requisito.

Actividad 1: Identificar el escenario según las características del requisito.

Requisitos	**Escenarios**
RnF1	Autenticación Única

Tabla 4: Identificación de escenarios

Al aplicar la Actividad 1 de la guía en el presente caso de estudio, según lo que plantea el requisito no funcional uno (RnF1) se puede identificar el escenario de autenticación única, ya que este plantea una de las condiciones principales para que sea aplicable este escenario,

pues permite como bien menciona el requisito introducir una sola vez la contraseña y puedes acceder a nuevos recursos sin necesidad de volverlo a hacer.

Actividad 2: Definir el modelo a utilizar según el escenario identificado. Seguidamente una vez culminada la Actividad 1, se procede a definir el modelo que es la Actividad 2, la cual es dependiente del escenario identificado, es decir si el escenario que se identificó es la Autenticación Única, el modelo a definir será el Modelo de Autenticación Única y así sucederá lo mismo con otro escenario.

Escenarios	**Modelos**
Autenticación Única	Autenticación Única

Tabla 5: Definición del modelo

Actividad 3: Definir el protocolo de seguridad según el modelo definido en la actividad anterior.

Modelos	**Protocolos**
Autenticación Única	SAML

Tabla 6: Definición del protocolo

En este caso al cumplir la Actividad 3 Definir el protocolo, se definió utilizar el protocolo SAML, porque aunque los otros protocolos estudiados en la investigación realizan este modelo, éste es uno de los protocolos más usados en las instituciones, ya que el principal problema que este se centra en resolver es la autenticación única, además de las numerosas ventajas que presenta. También es un protocolo soportado por una de las organizaciones más conocidas en el mundo del desarrollo de los diferentes estándares y protocolos de seguridad como es OASIS (Organization for the Advancement of Structured Information Standards), lo trabajan una cantidad significativa de empresas, como son: que implantan estos esquemas o venden productos que lo hagan, se integra con otros protocolos que presentan los mismos objetivos de éste, que es la autenticación única, posee un conjunto de implementaciones que facilitan el diseño del mecanismo a realizar, entre otros.

La arquitectura que se obtiene como resultado de la realización de estas actividades en correspondencia con el escenario, protocolo y modelo definido es:

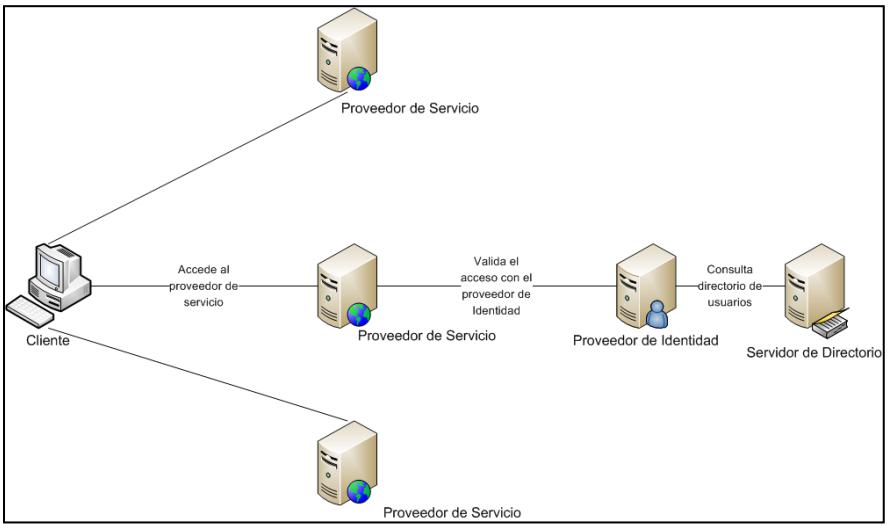

Figura 6: Arquitectura de autenticación única

3.3 Conclusiones

Al culminar este capítulo, en el cual se desarrolló un caso de estudio donde se aplica la propuesta de guía, se ha llegado a la conclusión que la puesta en práctica de esta traerá consigo muchas ventajas como evitar el ingreso constante de nombre de usuario y contraseña (autenticación única) cada vez que intente acceder a una nueva aplicación mediante el uso de los distintos mecanismos de autenticación y autorización que te brindan los protocolos de autenticación analizados.

Conclusiones generales

Al finalizar este trabajo se ha arribado a la conclusión de que los referentes teóricos del tema de investigación que fueron analizados, como son los protocolos de autenticación y autorización y los modelos de estos respectivamente brindan diversos mecanismos de autenticación y autorización que proporcionan un nivel de seguridad mayor para las aplicaciones web.

Además con la propuesta de guía presentada para el diseño de los mecanismos de autenticación y autorización en aplicaciones web mejorará considerablemente las técnicas de autenticación y autorización utilizadas actualmente con el objetivo de evitar los frecuentes problemas que presentan los sistemas en estos aspectos de la seguridad.

El desarrollo del diseño de un caso de estudio, donde se muestra el funcionamiento de uno de los mecanismos de autenticación que brinda el protocolo SAML en el escenario de autenticación única, traerá como ventajas evitar el reingreso de credenciales cada vez que intente acceder a un nuevo sitio o recurso.

Bibliografía

Aguilera. 2010. Definiciones Conceptuales. [Online] 2010. https://sites.google.com.

ASP. 2011. Guia completa ASP.NET. [Online] 2011. www.es-asp.net/Foro/general/4559/guia-completa-asp-net.aspx.

Ayala, Alejandro Cartas. 2008. Politicas de privacidad en P3P y XACML. [Online] 2008. www.ipv6.itam.mx/publicacion/xacml.pdf.

C. Gutiérrez, E. Fernández-Medina, M. Piattini. 2005. *Seguridad en Servicios Web.* 2005.

Community, Protocol CAS | Jasig. 2009. Protocol CAS | Jasig Community. [Online] 2009. www.jasig.org › Projects.

Esponda, Sebastián. 2007. Tesina Esponda2. [Online] 2007. www.ub.edu.ar/investigaciones/tesinas/135_esponda.pdf.

ESQUIVEL SUAZO MARÍA LUISA. 2011 . Definiciones Conceptuales. [Online] 02 1, 2011 . https://sites.google.com.

EstandaresyTecnologiasDeFederacionDeIdentidades. 2011. Estandares y Tecnologias de Federacion De Identidades. [Online] 2011. estandaresytecnologiasdefi.wordpress.com/.

Garfinkel. 1999. Capitulo 1.Seguridad informatica:Conceptos Basicos. [Online] 1999. catarina.udlap.mx.

GIL, ALBERTO BERMEJO. 2011. BIBLIOTECA PARA PORTABILIDAD DE PERFILE EN ENTORNOS SSO (SINGLE SIGN-ON). [Online] 2011. e-archivo.uc3m.es/bitstream/10016/.../PFC_Alberto_Bermejo_Gil.

Gil, Alberto Bermejo. 2011. BIBLIOTECA PARA PORTABILIDAD DE PERFILES EN ENTORNOS DE SSO (SINGLE SIGN-ON). [Online] 2011. e-archivo.uc3m.es/bitstream/10016/.../PFC_Alberto_Bermejo_Gil.pdf.

Hammer-Lahav, Eran. 2007. Introduction — OAuthn - OAuth. [Online] 2007. oauth.net/about/ .

Info@citel. 2011. Definicion de autenticacion. [Online] 2011. www.oas.org/en/citel/infocitel/2006/junio/seguridad_e.asp.

informática, Definicion de seguridad. 2011. Definicion de seguridad informática. [Online] 2011. www.definicion.de.

Informática, Gestion de Riesgo en la Seguridad. 2011. Gestion de Riesgo en la Seguridad Informática. [Online] 2011. www.wordpress.com.

Internet, Extranet,Intranet. 2011. Internet, Extranet,Intranet. [Online] 2011. internet-ipdd.blogspot.com/ .

JUAN, CARLES DE HARO. 2009. MODELING AND DEVELOPING ACCESS CONTROL POLICIES USING SEMANTIC TECHNOLOGIES. [Online] 2009. upcommons.upc.edu.

Lomascolo, Rodolfo. 1999. Capítulo 1. Conceptos Básicos. [Online] 1999. catarina.udlap.mx.

Lomáscolo, Rodolfo. 2000. La seguridad en internet es posible. [Online] 2000. marketingycomercio.com.

Magazine, Estr@tegia. 2011. GestioPolis. [Online] 2011. www.gestiopolis.com.

Martin, Tubal. 2010. OAuth 1.0a: Introducción e implementación utilizando PHP, PECL OAuth y Twitter. [Online] 2010. blog.margenn.com/.../oauth-introduccion-implementacion-ph... - España.

Mifsud, Elvira. 2012. MONOGRÁFICO: Introducción a la seguridad informática - Seguridad de la información / Seguridad informática. [Online] 2012. www.monografias.com.

OWASP. 2011. OWASP Guide Project - OWASP. [Online] 2011. www.OWASP.org.

Pérez, Sergio Serrano. 2011. Uso de servicios de seguridad y confianza desde la nube:Trustedx y la plataforma Azure de Microsoft. [Online] 2011. upcommons.upc.edu/pfc/bitstream/2099.1/12401/1/74186.pdf.

Perigaud, Benoît. 2010. Resolución de los problemas DE AUTENTICACION A UNA APLICACION WEB. [Online] 2010. upcommons.upc.edu/pfc/bitstream/2099.1/10776/1/PFC.pdf.

Rojo, Rodrigo Castro. 2011. documentacion:protocolo [PAPI]. [Online] 2011. www.papisoftware.net/doku.php?id=documentacion:protocolo.

ROJO, RODRIGO CASTRO. 2010. PAPI COMO INFRAESTRUCTURA DE SEGURIDAD DISTRIBUIDA APLICADA A ENTORNOS DE FUSION TERMONUCLEAR. [Online] 2010. e-spacio.uned.es/fez/view.php?pid=tesisuned:IngInf-Rcastro.

Rojo, Rodrigo Castro. 2010. Papi como infraestructura de seguridad distribuida aplicada a entornos de fusión termonuclear. [Online] 2010. e-spacio.uned.es.

Sandoval, Alejandro Núñez. 2008. Estándares de seguridad en la información. [Online] 2008. www.enter@te.com.

SiteMinder®, CA. 2011. CA SiteMinder®. [Online] 2011. http://www.ca.com/~/media/Files/TechnologyBriefs/CA-SiteMinder-Technology-Brief-ESN.pdf.

Suazo, Maria Luisa Esquivel. 2011. Definiciones Conceptuales. [Online] 2011. https://sites.google.com.

UPC, Single Sign-On -. 2010. Single Sign-On - UPC. [Online] 2010. docencia.ac.upc.es/FIB/CASO/seminaris/1q0304/T7.ppt.

Web, Seguridad en desarrollo de aplicaciones. 2011. Seguridad en desarrollo de aplicaciones Web. [Online] 2011. www.monografias.com.

YDN, OAuth Authorization Model -. 2012. OAuth Authorization Model - YDN. [Online] 2012. developer.yahoo.com/oauth/.

Glosario de Términos

ASP.NET: Es un modelo de desarrollo web unificado creado por Microsoft para el desarrollo de sitios y aplicaciones web dinámicas con un mínimo de código, forma parte de .NET Framework que contiene las librerías necesarias para la codificación.

AJAX: Siglas de Asíncronos Java Script and XML, es un término que describe un nuevo acercamiento a usar un conjunto de tecnologías existentes juntas, incluyendo las siguientes: HTML o XHTML, hojas de estilo (Cascading Style Sheets o css), Java script, el DOM (Document Object Model), XML, XSLT, y el objeto XMLHttpRequest.

Back-End: El back-end es la "memoria" de las aplicaciones; es la parte del sistema que se encarga de recuperar y registrar la información central del negocio.

CSS: Es un lenguaje de estilo que define la presentación de los documentos HTML. Por ejemplo, CSS abarca cuestiones relativas a fuentes, colores, márgenes, líneas, altura, anchura, imágenes de fondo, posicionamiento avanzado y muchos otros temas.

Cookies: Las cookies son un conocido mecanismo que almacena información sobre un usuario de internet en su propio ordenador, y se suelen emplear para asignar a los visitantes de un sitio de internet un número de identificación individual para su reconocimiento subsiguiente.

HASH: Se refiere a una función o método para generar claves o llaves que representen de manera casi unívoca a un documento, registro, archivo, etc., resumir o identificar un dato a través de la probabilidad, utilizando una función hash o algoritmo hash.

HTML: HTML es el lenguaje con el que se definen las páginas web. Básicamente se trata de un conjunto de etiquetas que sirven para definir el texto y otros elementos que compondrán una página web.

IP: (Internet Protocol), lo que en español quiere decir protocolo de internet que es un protocolo utilizado para la comunicación de datos a través de una red de paquetes combinados.

Internet Explorer: Internet Explorer o IE es un navegador web desarrollado por Microsoft. Funciona en el sistema operativo Windows. Permite la búsqueda de cualquier cosa a través de su navegador.

Java Script: JAVA Script es un lenguaje interpretado, multiplataforma, orientado a eventos con manejo de objetos, cuyo código se incluye directamente en el mismo documento, usado para el desarrollo de aplicaciones cliente-servidor en páginas HTML.

Master Pages: Una Master Page o Página Principal es una estructura base para un conjunto de páginas pertenecientes a un mismo sitio Web. Este esqueleto base se almacena en un archivo independiente y luego es heredado por otras páginas que requieren esa estructura base.

.NET: Es un framework de Microsoft que hace un énfasis en la transparencia de redes, con independencia de plataforma de hardware y que permita un rápido desarrollo de aplicaciones. Basado en ella, la empresa intenta desarrollar una estrategia horizontal que integre todos sus productos, desde el sistema operativo hasta las herramientas de mercado.

.Net Framework: Es un conjunto de rutinas, librerías y componentes necesarios para las nuevas aplicaciones de Microsoft que utilicen la tecnología .Net Framework.

Oracle: Oracle es básicamente una herramienta cliente/servidor para la gestión de Bases de Datos. Es un producto vendido a nivel mundial, aunque la gran potencia que tiene y su elevado precio hacen que sólo se vea en empresas muy grandes y multinacionales, por norma general. En el desarrollo de páginas web pasa lo mismo: como es un sistema muy caro no está tan extendido como otras bases de datos, por ejemplo, Access, MySQL, SQL Server, etc.

PL/SQL: PL/SQL es el lenguaje de programación que proporciona Oracle para extender el SQL estándar con otro tipo de instrucciones.

Proxy: Un proxy, en una red informática, es un programa o dispositivo que realiza una acción en representación de otro, esto es, si una hipotética máquina A solicita un recurso a una C, lo hará mediante una petición a B; C entonces no sabrá que la petición procedió originalmente de A. Esta situación estratégica de punto intermedio suele ser aprovechada para soportar una serie de funcionalidades: proporcionar caché, control de acceso, registro del tráfico, prohibir cierto tipo de tráfico etcétera.

SSL: Son las siglas en inglés de Secure Socket Layer (en español capa de conexión segura). Es un protocolo criptográfico (un conjunto de reglas a seguir relacionadas a seguridad,

aplicando criptografía) empleado para realizar conexiones seguras entre un cliente (como lo es un navegador de Internet) y un servidor (como lo son las computadoras con páginas web).

SQL Server: Es un conjunto de objetos eficientemente almacenados. Los objetos donde se almacena la información se denominan tablas, y éstas a su vez están compuestas de filas y columnas.

TCP/IP: El TCP / IP es la base del Internet que sirve para enlazar computadoras que utilizan diferentes sistemas operativos, incluyendo PC, minicomputadoras y computadoras centrales sobre redes de área local y área extensa. El nombre TCP / IP Proviene de dos protocolos importantes de la familia, el Transmisión Control Protocol (TCP) y el Internet Protocol (IP). Todos juntos llegan a ser más de 100 protocolos diferentes definidos en este conjunto.

Unix: Es un sistema operativo de tiempo compartido, controla los recursos de una computadora y los asigna entre los usuarios. Permite a los usuarios correr sus programas. Controla los dispositivos de periféricos conectados a la máquina.

URL: Significa **Uniform Resource Locator**, es decir, localizador uniforme de recurso y se refiere a la dirección única que identifica a una página web en Internet.

Webforms: Se trata de la simulación para la web de la visión de Rapid Application Development sostenida por Microsoft, puntualmente, de su visión del desarrollo de aplicaciones para sus sistemas operativos.

XML: Son las siglas de **Extensible Markup Language**, una especificación/lenguaje de programación desarrollada por el W3C. XML es una versión de Standard Generalized Markup Language o Lenguaje de Señalización General Normalizado (SGML), diseñado especialmente para los documentos de la web. Permite que los diseñadores creen sus propias etiquetas, permitiendo la definición, transmisión, validación e interpretación de datos entre aplicaciones y entre organizaciones.