Universidad de las Ciencias Informáticas

Facultad 4

.

Título: Desarrollo de un portal único de Autenticación.

Trabajo de Diploma para optar por el título de

Ingeniero en ciencias Informáticas

Autor(es): Boris R. Fernández Justel

Tutor(es): Léster Carballo Pérez

Junio 2007

DECLARACIÓN DE AUTORÍA

Yo Boris Ramón Fernández Justel:

Declaro que soy el único autor de este trabajo y autorizo a la Facultad 4 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los 13 días del mes de junio del año 2007

Boris R Fernández Justel

Tutor

Léster Carballo Pérez

AGRADECIMIENTOS

Le agradezco a todos los que de una forma u otra contribuyeron a la realización de este trabajo, desde mi familia hasta mi tutor y compañeros. A la Revolución por desarrollarme como un intelectual así como a la Universidad de las Ciencias Informáticas.

DEDICATORIA

Le dedico este trabajo en especial a mi familia: mi tía Martha, mi querida abuela Marilis, mi abuelo Aníbal,Maray,Luis Manuel, a la memoria de mi abuela María, a mi papá Fidel, mi tía Magdalena, Eva, a mi novia Yani y en especial a mi querida madre Marilín que es parte de mi cada día. A todos muchas gracias por hacer este sueño realidad.

RESUMEN

El trabajo que a continuación se presenta es el resultado de una investigación para la solución del problema de la autenticación, como prestarle a las aplicaciones un método realmente nuevo para tener un mejor control de los usuarios así como prestarles un mejor servicio de navegación a estos. Con nuestro servicio web las aplicaciones podrán compartir la autenticación así como tener privilegios con sus usuarios. Nuestro objetivo en general es la creación de un servicio que agrupe a varias aplicaciones con una única autenticación. Este trabajo se realizó como tal no es un portal sino un servicio web el cual estará publicado para que las aplicaciones puedan utilizarlo presenta varios métodos los cuales están implementados en el servicio, es decir que el usuario no tendrá contacto directo con el servicio sino las aplicaciones.Nuestro sistema funciona para cualquier aplicación no importa la plataforma en que se desarrolló. Nuestro deseo es poner en las manos de los clientes otra arma para la seguridad de los sistemas.

PALABRAS CLAVE

Servicio Web

Aplicación

Sistema

Software

Plataforma

Seguridad Informática

INDICE

INTRODUCCIÓN

Internet se está convirtiendo cada vez más rápido en el mejor “amigo” de todo el que desea promocionar o vender algo de la manera más eficiente posible. El desarrollo de páginas web en sectores como el turismo, industria, comercio, hostelería, instituciones o agencias de viajes, dan buena prueba de ello. Que su empresa o actividad esté en Internet se hace imprescindible si desea llegar al mayor número de posibles clientes.

La presencia en Internet de las empresas se suele articular en torno a un portal corporativo en el cual se informa y/o se prestan servicios en base a objetivos perfectamente definidos. Estos objetivos responden a una estrategia empresarial y de comunicación fijada previamente por la dirección o el equipo técnico de la compañía. A través del sitio, el usuario tendrá la posibilidad de consumir la información y los servicios que proporciona la Institución y, así mismo, establecer comunicación con ella. Existen elementos que facilitan la realización de las tareas por parte de los usuarios del sitio y que permiten a la empresa cumplir sus objetivos de manera eficaz: un buen diseño gráfico, una arquitectura de la información de calidad y un diseño de interacción que garantice la usabilidad del sitio web.

Uno de los principales problemas que presentan los portales o sitios web tanto en la Universidad de las Ciencias Informáticas como en el mundo, es la aceptación de autenticaciones compartidas por diferentes portales, construidos sobre distintas plataformas, por ejemplo: asp.net Java y PHP. Esto dificulta a los usuarios la navegación en muchos sitios de interés, pues tienen que autentificarse varias veces a medida que cambian de portal. Estos Portales muchas veces están relacionados entre si en cuanto a información, e incluso pertenecen a una misma institución. Este hecho hace que sea engorroso y constituya una la pérdida de tiempo para el usuario, dificultando la obtención de la información. Por otra parte si esta es la primera vez que el usuario visita los portales, tendría que realizar el mismo proceso de registro en todos ellos.

Por todo lo planteado anteriormente se ve evidente la necesidad de tener una aplicación independiente de los mismos que garantice la autenticación, los registros y los roles de los usuarios, para varios portales inscritos a esta.

Problema de investigación

De todo este importante análisis se deriva el siguiente problema ¿Cómo unificar los procesos de autentificación, registro y asignación de roles, a usuarios de varios portales con funcionalidades comunes?

Objetivo General

El objetivo de la investigación en este trabajo se encuentra en la elaboración de un portal que gestione la autenticación, el registro y la asignación de rol a usuarios así como las sesiones de trabajo de estos para un mejoramiento en la seguridad de los sistemas. Dichas funcionalidades tienen que ser utilizadas por otros portales construidos bajo diversas plataformas, garantizando una funcionalidad para las más conocidas asp.net, J2EE y php.

Objetivos Específicos

El presente trabajo presenta además algunos objetivos específicos que se debe de cumplir para su buen funcionamiento de nuestro sitio y dar cumplimiento al objetivo general:

-Verificar la existencia de algún portal semejante en la UCI y establecer un diagnóstico de su funcionamiento (en caso de que exista).

- Estudiar, seleccionar y establecer un sistema comunicación entre el portal y las aplicaciones clientes.
- Establecer las formas de autenticación, registro y asignación de roles a los usuarios.
- Escoger plataformas, entornos de desarrollo, sistemas gestores de base de datos y demás herramientas para construir el software.
- Diseñar y modelar el software y el soporte de base de datos para los usuarios.
- Implementar el software.

- Poner a prueba el producto.

Campo de Acción

Automatización en los registros, asignación de roles y autentificación compartidas de usuarios para los software de usos internos en La Facultad 4 de La Universidad de Ciencias Informáticas

Hipótesis

Si se desarrolla un sitio Web donde este comparta la autenticación, el registro y la asignación de roles de los usuarios, se contribuirá a una ganancia de tiempo en el acceso de los usuarios entre varias aplicaciones, proporcionando mayor calidad en los servicios.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 INTRODUCCION

Cuando Internet inició su funcionamiento como lo conocemos, empezando a tomar popularidad aproximadamente en 1993, solo se podía apreciar texto, imágenes y enlaces. La introducción de Plugins en los navegadores permitió mayor interactividad entre el usuario y el cliente, aunque estaba limitado por la velocidad y la necesidad de tener que bajar e instalar cada plugin que se necesitara, por lo que estos se desarrollaron mayormente en áreas de vídeo, audio y realidad virtual. En este capítulo abordaremos sobre las herramientas importantes para la elaboración del trabajo así como ejemplos de sistemas parecidos no sólo en el mundo sino también en Cuba y exactamente en la Universidad de las Ciencias Informáticas.

1.2 PROTOCOLO HTTP

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

HTTP es un protocolo sin estado, es decir, que no guarda ninguna información sobre conexiones anteriores. Al finalizar la transacción todos los datos se pierden. Por esto se popularizaron las cookies, que son pequeños ficheros guardados en el propio ordenador que puede leer un sitio web al establecer conexión con él, y de esta forma reconocer a un visitante que ya estuvo en ese sitio anteriormente. Gracias a esta identificación, el sitio web puede almacenar gran número de información sobre cada visitante, ofreciéndole así un mejor servicio. La versión actual de HTTP es la 1.1 El protocolo HTTP está basado en el modelo cliente-servidor. Un cliente HTTP abre una conexión y envía su solicitud al servidor, el cual responderá con el recurso solicitado —si está disponible y su acceso es permitido— y la conexión se cierra.

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

Es utilizado principalmente por entidades bancarias, tiendas en línea, y cualquier tipo de servicio que requiera el envío de datos personales o contraseñas. El puerto estándar para este protocolo es el 443.

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

1.3 NAVEGADORES

Un navegador web o explorador web (del inglés, navigator o browser) es una aplicación software que permite al usuario recuperar y visualizar documentos de hipertexto, comúnmente descritos en HTML, desde servidores web de todo el mundo a través de Internet. Esta red de documentos es denominada World Wide Web (WWW). Los navegadores actuales permiten mostrar o ejecutar: gráficos, secuencias de vídeo, sonido, animaciones y programas diversos además del texto y los hipervínculos o enlaces.

La funcionalidad básica de un navegador web es permitir la visualización de documentos de texto, posiblemente con recursos multimedia incrustados. Los documentos pueden estar ubicados en la computadora en donde está el usuario, pero también pueden estar en cualquier otro dispositivo que este conectado a la computadora del usuario o a través de Internet, y que tenga los recursos necesarios para la transmisión de los documentos (un software servidor web). Tales documentos, comúnmente denominados páginas web, poseen hipervínculos que enlazan

una porción de texto o una imagen a otro documento, normalmente relacionado con el texto o la imagen.

1.3.1 FUNCIONAMIENTO DE LOS NAVEGADORES

La comunicación entre el servidor Web y el navegador se realiza mediante el protocolo HTTP, aunque la mayoría de los hojeadores soportan otros protocolos como FTP, Gopher, y HTTPS

La función principal del navegador es descargar documentos HTML y mostrarlos en pantalla. En la actualidad, no solamente descargan este tipo de documentos sino que muestran con el documento sus imágenes, sonidos e incluso vídeos (streaming) en diferentes formatos y protocolos. Además, permiten almacenar la información en el disco o crear marcadores (*bookmarks*) de las páginas más visitadas.

Algunos de los navegadores web más populares se incluyen en lo que se denomina una Suite. Estas Suite disponen de varios programas integrados para leer noticias de Usenet y correo electrónico mediante los protocolos NNTP, IMAP y POP.

Los primeros navegadores web sólo soportaban una versión muy simple de HTML. El rápido desarrollo de los navegadores web propietarios condujo al desarrollo de dialectos no estándares de HTML y a problemas de interoperabilidad en la web. Los más modernos (como Amaya, Mozilla, Opera y versiones recientes de Internet Explorer) soportan los estándares HTML y XHTML (comenzando con HTML 4.01, los cuales deberían visualizarse de la misma manera en todos ellos).

1.4 HTML

El HTML, HyperText Markup Language (lenguaje de marcas hipertextuales),es un lenguaje de marcación diseñado para estructurar textos y presentarlos en forma de hipertexto, el cual es el formato estándar de las páginas web. Gracias a Internet y a los navegadores del tipo Internet

Explorer, Opera, Firefox o Netscape, el HTML se ha convertido en uno de los formatos más populares que existen para la construcción de documentos y también de los más fáciles de aprender.

1.5 PROGRAMACION WEB

Una página Web sólo con texto e imagen se ha convertido, en los escasos años de andanza de la Web, en una excepción. Dejando aparte el hecho de que cada vez más se utilizan páginas generadas dinámicamente en el servidor, vinculadas a sistemas de bases de datos (por ejemplo, sistemas ASP), las páginas Web a las que nos hemos acostumbrado presentan mucha más riqueza que las originales de los primeros años 90 .

1.5.1 Algunas ventajas de las páginas Web dinámicas

Las páginas Web dinámicas presentan una gran ventaja sobre las estáticas por lo que en el trabajo se le da gran prioridad a estos tipos de portales:

- ✓ Acceso on-line: El acceso on-line a los documentos desde cualquier lugar del mundo a través de Internet. Se posibilita su consulta sin colapsos del sistema, eliminando la necesidad de distribuir múltiples copias de un mismo documento. En caso que se necesite distribuir información masiva, contamos también con la posibilidad de volcar toda la información a CD-Roms, DVD\'s o través de una red programada.

- ✓ Rapidez: Los usuarios pueden tener fácil acceso y rápidamente a toda la información disponible con tan solo apuntar y hacer clic y además, pasar de un documento a otro. La información se obtiene al instante y las consultas son inmediatas, mediante distintos tipos de extracción de información (índices de búsqueda, texto dentro del documento, reportes de un sector específico del documento, etc.). Al estar montado íntegramente sobre nuestra

plataforma Web, los procesos brindan una posibilidad de crecimiento hacia nuevos servicios que están marcando claramente el futuro.

- ✓ Seguridad: no todo el mundo tiene acceso a las páginas, las plataforma brindan absolutas garantías de seguridad, gracias a niveles distintos de acceso y restricciones dentro del sistema, así como un proceso de autenticación mediante el cual el sistema sabe el usuario que esta utilizando el servicio de la Web.

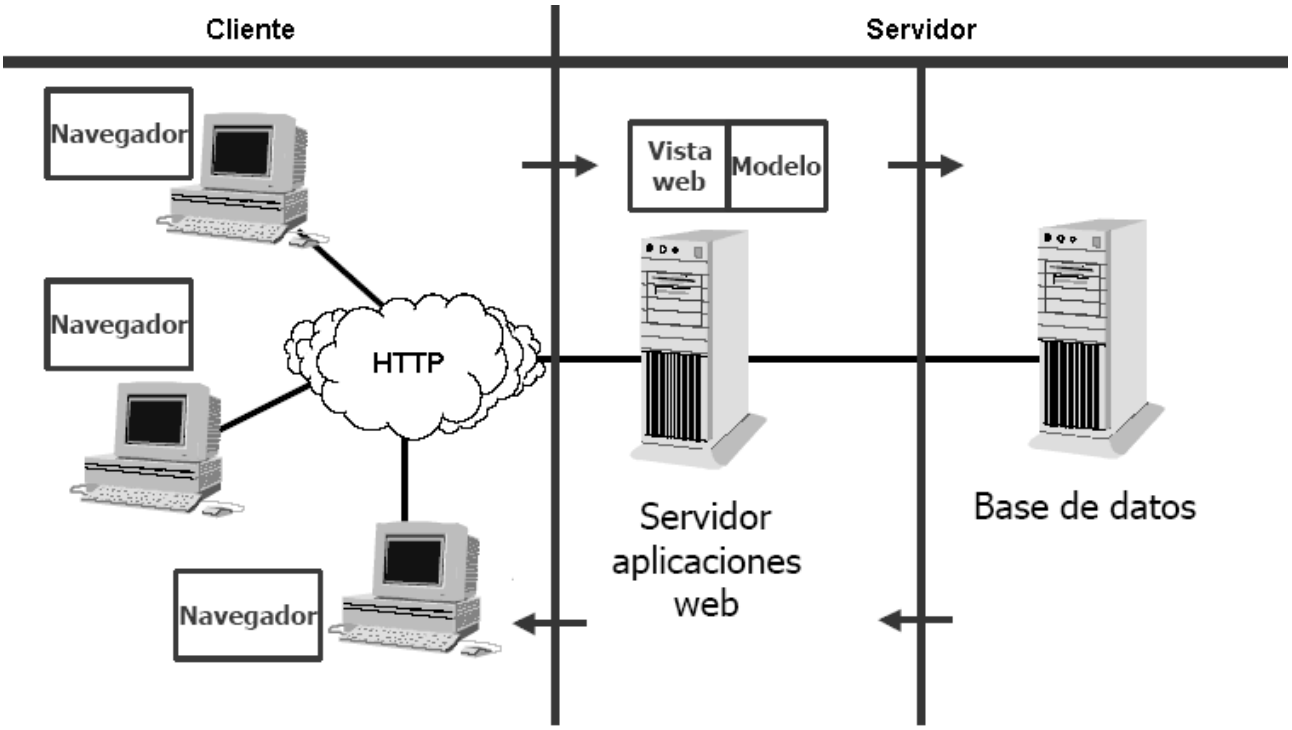


Fig. 1 Descripción del modelo cliente servidor

1.6 Base de datos

En la actualidad, muchas instituciones y empresas se han dado cuenta de la importancia que el Web tiene en el desarrollo de sus potencialidades, ya que con ello pueden lograr una mejor comunicación con personas o instituciones situadas en cualquier lugar del mundo. Gracias a la conexión con la red mundial, poco a poco, cada individuo o institución va teniendo acceso a mayor cantidad de información de las diversas ramas de la ciencia con distintos formatos de almacenamiento.

Es muy importante aclarar la importancia del uso de Base de Datos en las páginas y sitios dinámicos debido a la constante relación de la información con lo usuarios, las actualizaciones y un mejor control es parte del trabajo de estos sistemas.

Una Base de Datos es un conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su posterior uso. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta.

En la actualidad, y gracias al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos tienen formato electrónico, que ofrece un amplio rango de soluciones al problema de almacenar datos. En informática existen los sistemas gestores de bases de datos (SGBD), que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Las propiedades de los sistemas gestores de bases de datos se estudian en informática. Las aplicaciones más usuales son para la gestión de empresas e instituciones públicas. También son ampliamente utilizadas en entornos científicos con el objeto de almacenar la información experimental.

1.6.1 Sistemas Gestores de Base de Datos (SGBD)

Los **Sistemas de gestión de base de datos** son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.

Existen distintos objetivos que deben cumplir los SGBD:

- **Abstracción de la información**. Los SGBD ahorran a los usuarios detalles acerca del almacenamiento físico de los datos. Da lo mismo si una base de datos ocupa uno o cientos de archivos, este hecho se hace transparente al usuario. Así, se definen varios *niveles de abstracción*.
- **Independencia**. La independencia de los datos consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.
- **Redundancia mínima**. Un buen diseño de una base de datos logrará evitar la aparición de información repetida o redundante. De entrada, lo ideal es lograr una redundancia nula; no obstante, en algunos casos la complejidad de los cálculos hace necesaria la aparición de redundancias.
- **Consistencia**. En aquellos casos en los que no se ha logrado esta redundancia nula, será necesario vigilar que aquella información que aparece repetida se actualice de forma coherente, es decir, que todos los datos repetidos se actualicen de forma simultánea.
- **Seguridad**. La información almacenada en una base de datos puede llegar a tener un gran valor. Los SGBD deben garantizar que esta información se encuentra asegurada frente a usuarios malintencionados, que intenten leer información privilegiada; frente a ataques que deseen manipular o destruir la información; o simplemente ante las torpezas de algún usuario autorizado pero despistado. Normalmente, los SGBD disponen de un complejo sistema de permisos a usuarios y grupos de usuarios, que permiten otorgar diversas categorías de permisos.

- **Integridad**. Se trata de adoptar las medidas necesarias para garantizar la validez de los datos almacenados. Es decir, se trata de proteger los datos ante fallos de hardware, datos introducidos por usuarios descuidados, o cualquier otra circunstancia capaz de corromper la información almacenada.
- **Respaldo y recuperación**. Los SGBD deben proporcionar una forma eficiente de realizar copias de seguridad de la información almacenada en ellos, y de restaurar a partir de estas copias los datos que se hayan podido perder.
- **Control de la concurrencia**. En la mayoría de entornos (excepto quizás el doméstico), lo más habitual es que sean muchas las personas que acceden a una base de datos, bien para recuperar información, bien para almacenarla. Y es también frecuente que dichos accesos se realicen de forma simultánea. Así pues, un SGBD debe controlar este acceso concurrente a la información, que podría derivar en inconsistencias.
- **Tiempo de respuesta**. Lógicamente, es deseable minimizar el tiempo que el SGBD tarda en darnos la información solicitada y en almacenar los cambios realizados.

1.6.2 Ventajas de los gestores de base de datos

1. Facilidad de manejo de grandes volúmenes de información.
2. Gran velocidad en muy poco tiempo.
3. Independencia del tratamiento de información.
4. Seguridad de la información (acceso a usuarios autorizados), protección de información, de modificaciones, inclusiones, consulta.
5. No hay duplicidad de información, comprobación de información en el momento de introducir la misma.
6. Integridad referencial el terminar los registros.

1.7 Sistemas Gestores de Base de Datos libres

17.1 Postgre SQL:

PostgreSQL es un motor de base de datos, es servidor de base de datos relacional libre, liberado bajo la licencia BSD.

Algunas de sus principales características son:

Alta concurrencia

Mediante un sistema denominado MVCC (Acceso concurrente multiversión) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo *commit*. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos.

Amplia variedad de tipos nativos

PostgreSQL provee nativamente soporte para:

- Números de precision arbitraria
- Texto de largo ilimitado.
- Figuras geométricas (con una variedad de funciones asociadas)
- Claves ajenas también denominadas Llaves ajenas o Llaves Foráneas (*foreign keys*).
- Disparadores (*triggers*).
- Vistas.
- Integridad transaccional.
- Herencia de tablas.
- Tipos de datos y operaciones geométricas.

1.7.2 My SQL

MySQL es un sistema de gestión de base de datos, multihilo y multiusuario con más de seis millones de instalaciones. MySQLAB desarrolla MySQL como software libre en un esquema de licenciamiento dual. Por un lado lo ofrece bajo la GNU GPL, pero, empresas que quieran incorporarlo en productos privativos pueden comprar a la empresa una licencia que les permita ese uso.

MySQL es muy utilizado en aplicaciones web como Media Wiki o Drupal, en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python), y por herramientas de seguimiento de errores como Bugzilla. Su popularidad como aplicación web está muy ligada a PHP, que a menudo aparece en combinación con MySQL. MySQL es una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones Web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones.

1.7.2.1 Características distintivas

Las siguientes características son implementadas únicamente por MySQL:

- Múltiples motores de almacenamiento (MyISAM, Merge, InnoDB, BDB, Memory/heap, MySQL Cluster, Federated, Archive, CSV, Blackhole y Example en 5.x), permitiendo al usuario escoger la que sea más adecuada para cada tabla de la base de datos.
- Agrupación de transacciones, reuniendo múltiples transacciones de varias conexiones para incrementar el número de transacciones por segundo.
- Un amplio subconjunto de ANSI SQL 99, y varias extensiones.
- Soporte a multiplataforma
- Procedimientos almacenados
- Tiggers
- Cursors

- Vistas actualizables
- Soporte a VARCHAR
- INFORMATION_SCHEMA

1.7.3 SQL Server

Microsoft SQL Server es un sistema de gestión de bases de datos relacionales (SGBD) basada en el lenguaje SQL, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea. Así de tener unas ventajas que a continuación se pueden describir.

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

1.8 La seguridad en los sitios Web

Todos conocemos el ímpetu que ha tenido en la actualidad el desarrollo de la programación Web y los portales digitales, la pagina Web es una forma dinámica y creativa de interactuar tanto con otros usuarios como de tiendas virtuales y un mundo de información que nos ayudan desarrollarnos como buenos intelectuales .Pero existen situaciones que dificultan que atrasan este proceso.

La seguridad es un factor clave para los arquitectos y los desarrolladores de aplicaciones. Las aplicaciones que almacenan información confidencial deben adoptar medidas de protección frente a ataques malintencionados y rivales que intenten apropiarse de información o propiedad

intelectual. A la hora de diseñar un modelo de seguridad para una aplicación, es necesario considerar los requisitos de seguridad desde una perspectiva empresarial y las implicaciones que el modelo seleccionado pueda tener en el rendimiento, la escalabilidad y la distribución.

1.9 La autenticación

Se trata del proceso de aceptación de las credenciales de un usuario y la validación de las mismas frente a una autoridad designada. La identidad del usuario (o potencialmente, la de una aplicación o equipo) se denomina principal de seguridad. El cliente debe proporcionar las credenciales para permitir que el servidor compruebe la identidad del principal. Una vez conocida la identidad, la aplicación podrá autorizar al principal para que tenga acceso a los recursos del sistema

El robo de identidades en Internet, el phishing y los fraudes financieros en línea impulsan cada vez más la implementación de autenticación fuerte en las aplicaciones que utilizan los consumidores es por eso que es de vital importancia una buena autenticación para saber cual o cuales son los usuarios que están en nuestro sistema. Existen varios tipos de autenticaciones entre ellas se encuentran:

1.9 La autenticación

Se trata del proceso de aceptación de las credenciales de un usuario y la validación de las mismas frente a una autoridad designada. La identidad del usuario (o potencialmente, la de una aplicación o equipo) se denomina principal de seguridad. El cliente debe proporcionar las credenciales para permitir que el servidor compruebe la identidad del principal. Una vez conocida la identidad, la aplicación podrá autorizar al principal para que tenga acceso a los recursos del sistema

El robo de identidades en Internet, el phishing y los fraudes financieros en línea impulsan cada vez más la implementación de autenticación fuerte en las aplicaciones que utilizan los consumidores es por eso que es de vital importancia una buena autenticación para saber cual o

cuales son los usuarios que están en nuestro sistema. Existen varios tipos de autenticaciones entre ellas se encuentran:

1.9.1 Windows

La contraseña de Windows se basa en el juego de caracteres Unicode. Distingue mayúscula de minúscula y esta contraseña puede tener longitud máxima de 128 caracteres. La versión OWF de esta contraseña es también la contraseña OWF de Windows. Esta contraseña se calcula utilizando el algoritmo de cifrado RSA MD-4. Este algoritmo calcula una síntesis de 16 bytes de una cadena de bytes de contraseña de longitud variable de texto sin cifrar.

1.9.2 Formularios

Normalmente, la autenticación mediante formularios hace referencia a un sistema que redirige las solicitudes no autenticadas a un formulario HTML mediante la redirección del cliente HTTP. La autenticación mediante formularios es la elección correcta si la aplicación necesita recopilar sus propias credenciales del usuario en el inicio de sesión a través de formularios HTML. El usuario proporciona las credenciales y envía el formulario. Si la aplicación autentica la solicitud, el sistema emite una cookie que contiene las credenciales o una clave para readquirir la identidad. Las solicitudes posteriores se emiten con la cookie en los encabezados de la solicitud. Las solicitudes se autentican y autorizan mediante un controlador de eventos ASP.NET que utiliza el método de validación especificado en la aplicación.

Tenga en cuenta que la autenticación mediante formularios se utiliza a menudo para personalizar, cuando el contenido se ha personalizado para un usuario conocido. En algunos casos, el problema está en la identificación y no en la autenticación, por lo que basta con almacenar el nombre de usuario en una cookie duradera y utilizar ésta para tener acceso a la información de personalización del usuario.

Esta autenticación hace referencia a un componente de interfaz de usuario personalizado que acepta las credenciales del mismo; por ejemplo, un nombre de usuario y una contraseña. Un gran

número de aplicaciones de Internet utilizadas actualmente presentan este tipo de formularios para que los usuarios inicien la sesión. Es importante tener en cuenta que el formulario no realiza la autenticación por sí mismo, sino que sólo se ofrece como un modo de obtener las credenciales de usuario. La autenticación se realiza cuando se obtiene acceso al nombre y a la contraseña de usuario empleando código personalizado.

Cuando el usuario se autentica, el servidor suele ofrecer al cliente varios medios para indicar que ya ha sido autenticado para las siguientes solicitudes. Si es necesario, se puede obligar al cliente a autenticarse en cada solicitud, aunque ello podría afectar al rendimiento y a la escalabilidad. Para identificar a un cliente que ya ha iniciado la sesión con anterioridad existen dos enfoques básicos que se deben tener en cuenta:

- **Cookies**. Un cookie es una pequeña porción de datos que el servidor presenta inicialmente al cliente. Posteriormente, el cliente vuelve a mostrarla al servidor con cada solicitud HTTP. Se puede utilizar como una forma de indicar que el cliente ya se ha autenticado. ASP .NET proporciona una mecanismo para que se utilicen cookies en la autenticación de formularios en el módulo CookieAuthenticationProvider. Dichos cookies son compatibles con la mayor parte de los exploradores Web, incluidos Internet Explorer y Netscape Navigator.
- **Personalización**. Se puede implementar un mecanismo personalizado propio para identificar el cliente en el servidor. Si los clientes tienen deshabilitada la función de cookies, se debe considerar el almacenamiento de un identificador único en cada cadena de consulta URL. Asimismo, se pueden utilizar campos de formulario ocultos, almacenados en un nivel superior permanente o marco no visible. En cualquier caso, es necesario asegurarse de que ningún intruso pueda simular la autenticación en la aplicación mediante programación.

Los sitios Web que implementan la autenticación de formularios emplean generalmente cookies. La primera versión de .NET admitirá únicamente la autenticación de formularios mediante cookies.

1.9.2.1 Escenarios típicos de uso

Se debe considerar el uso de la autenticación de formularios cuando:

- Los nombres de usuario y contraseñas se encuentran almacenados en otras ubicaciones distintas de las cuentas de Windows. Se debe tener en cuenta que la autenticación de formularios no se puede utilizar con cuentas de Windows.
- Se está distribuyendo la aplicación a través de Internet.
- Es necesario admitir todos los sistemas operativos de los exploradores y del cliente.
- Se desea proporcionar un formulario de interfaz de usuario propio como página de inicio de sesión.

No se debe considerar el uso de la autenticación de formularios cuando:

- Se está distribuyendo una aplicación en una intranet corporativa y se puede aprovechar la autenticación de Windows integrada.
- No se puede efectuar el acceso mediante programación para comprobar el nombre de usuario y la contraseña.

1.9.2.2 Establecimiento de la seguridad en la autenticación de formularios

Si los usuarios envían contraseñas a través de la página de inicio de sesión, se puede proteger el canal utilizando SSL para evitar el robo de las mismas. Si se utilizan cookies para mantener la identidad del usuario entre las solicitudes, se debe tener en cuenta el riesgo potencial de que un intruso intente "robar" el cookie del usuario mediante un programa de supervisión de red. La única forma de proteger realmente el sitio cuando se empleen cookies, consiste en utilizar SSL en todas las comunicaciones. Para la mayoría de los sitios de comercio, esta opción resulta poco práctica debido a que implica una considerable reducción del rendimiento. Con ASP .NET el servidor puede volver a generar cookies en intervalos temporales. Esta directiva de caducidad de cookies está diseñada para evitar que otros usuarios obtengan acceso al sitio con un cookie robado.

1.9.2.3 Rendimiento y escalabilidad

A la hora de diseñar un sitio Web de gran volumen es necesario tener en cuenta las implicaciones de rendimiento de la autenticación de los usuarios. Si se espera que un gran número de usuarios inicie la sesión de forma simultánea, es preciso agilizar todo lo posible el proceso de comprobación de credenciales.

Si se utiliza SSL, existe un marcado descenso de rendimiento debido a los pasos de cifrado adicionales que se deben realizar. Con el fin de obtener los requisitos de rendimiento adecuados, puede que sea necesario separar los servidores que ejecutan el inicio de sesión de los servidores de contenido en una granja Web.

1.9.2.4 Comprobación de la presencia de cookies

Si se utiliza .NET, el proceso de comprobación de la existencia de cookies se realiza automáticamente. Sin embargo, si no se dispone de esta aplicación, existen dos enfoques básicos para este procedimiento:

- Se puede implementar un filtro ISAPI que confirme la presencia de un cookie en una solicitud del cliente, lo que constituye una prueba de que dicho cliente se ha autenticado. Si existe el cookie, se podrá autorizar que la solicitud siga su curso. Si, por el contrario, no está presente, se redirigirá al cliente a la página de inicio de sesión. Un filtro ISAPI como el descrito anteriormente se implementa mediante Microsoft® Commerce Server 2000.
- Se puede escribir código al principio de cada página Web que compruebe la existencia del cookie o de algún otro valor personalizado que pase esta página. Si el token no está presente, el código redirigirá al usuario a la página de inicio de sesión. Se trata de una implementación simple; no obstante, es probable que no se puedan proteger recursos que no sean las páginas ASP. Así, por ejemplo, los archivos de imagen pueden seguir siendo inaccesibles.

1.10 Servicios Web

Los Web Services o servicios Web son aplicaciones que nos permiten el intercambio de datos por XML, es una colección de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Es una aplicación de software desarrollada en lenguajes de programación diferente y ejecutada sobre cualquier plataforma pueden utilizar los servicios Web para intercambiar datos en redes de ordenadores como Internet La interoperatividad se consigue mediante la adopción de estándares abiertos.La virtud de estos radica en que cualquier aplicación puede consumirlos, es decir si el servicio Web esta construido bajo .NET, Java, PHP, CFML, cualquier cliente que use estas tecnologías puede consumirlo lo que lo hace multiplataforma y sobre todo muy portable. Los servicios Web son la revolución informática de la nueva generación de aplicaciones que trabajan colaborativamente en las cuales el software esta distribuido en diferentes servidores.

La industria de la tecnología de la información ha utilizado los servicios Web durante más de tres años. En numerosos documentos se describen las ventajas técnicas y empresariales que aportan los servicios Web. Un gran número de compañías hace uso de estos servicios en sus entornos de producción. Estos escenarios de clientes demuestran que se han conseguido aplicar en la práctica los objetivos de los servicios Web.

Los clientes, analistas del sector industrial, proveedores de sistemas así como las publicaciones periódicas del sector comercial coinciden en identificar un área clave que aún no se ha tratado: la entrega de mensajes confiable. La entrega confiable de mensajes se considera un aspecto crucial para que los servicios Web se conviertan en la infraestructura principal de la interconexión heterogénea de los procesos, sistemas y productos de las empresas.

Un sistema de mensajería confiable resulta vital para los servicios Web. No es posible resolver muchos problemas empresariales si los participantes no pueden estar seguros de que se realizarán los intercambios de mensajes. Sin un estándar de los servicios Web que proporcione una entrega de mensajes confiable, las aplicaciones implementarán la función necesaria en su lógica empresarial. Este requisito representa una carga para los desarrolladores de lógica

empresarial, pero lo que es aún más importante: impide la interoperabilidad debido a que se aportan soluciones incoherentes y diferentes a un problema común.
Por último, un estándar de entrega de mensajes confiable mejorará la eficacia de otros estándares de servicios Web como, por ejemplo, la seguridad, las transacciones y los procesos empresariales. Estas mejoras únicamente podrán producirse si la entrega de mensajes confiable es un estándar, en lugar de estar incrustada en la lógica empresarial. La entrega de mensajes confiable garantiza que la arquitectura, los protocolos y las interfaces de los servicios Web ofrecerán unas soluciones seguras, interoperables, transaccionales y sólidas.

1.10.1 ¿Qué es el SOAP?

Es un protocolo que define el formato XML para los mensajes de intercambio en el uso de un Web Service. Para aquellos programadores que solían utilizar llamadas del tipo RPC, SOAP también las soporta. Adicionalmente es posible mediante SOAP definir un mensaje HTTP y este punto es de especial interés puesto que el protocolo imprescindible para Internet es HTTP.

1.10.2 WSDL

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

Así, WSDL se usa a menudo en combinación con SOAP y XML Schema. Un programa cliente que se conecta a un servicio web puede leer el WSDL para determinar que funciones están disponibles en el servidor. Los tipos de datos especiales se incluyen en el archivo WSDL en forma de XML Schema. El cliente puede usar SOAP para hacer la llamada a una de las funciones listadas en el WSDL.

1.10.3 UDDI

UDDI son las siglas del catálogo de negocios de Internet denominado *Universal Description, Discovery and Integration*. El registro en el catálogo se hace en XML. UDDI es una iniciativa industrial abierta (sufragada por la OASIS) entroncada en el contexto de los servicios Web. El registro de un negocio en UDDI tiene tres partes:

- Páginas blancas - dirección, contacto y otros identificadores conocidos.
- Páginas amarillas - categorización industrial basada en taxonomías.
- Páginas verdes - información técnica sobre los servicios que aportan las propias empresas.

UDDI es uno de los estándares básicos de los servicios Web cuyo objetivo es ser accedido por los mensajes SOAP y dar paso a documentos WSDL, en los que se describen los requisitos del protocolo y los formatos del mensaje solicitado para interactuar con los servicios Web del catálogo de registros.

1.11 Arquitectura y standards

El funcionamiento de los Web services está basado en un conjunto de standards que permiten su creación, registración, ubicación y utilización. Los Web services pueden comunicarse entre sí, ya que están basados en XML que puede ser interpretado por cualquier aplicación, también es necesario otro tipo de información para su uso.

Las tecnologías detalladas a continuación han emergido como standards de facto para los web services*:*

- El Simple Object Access Protocol (SOAP) define un protocolo de comunicaciones Standard para Web services
- El Web Service Description Language (WSDL) define un mecanismo Standard para describir un web service
- El Universal Description, Discovery and Integration (UDDI) provee la forma de registrar y descubrir web services.

La siguiente figura muestra cómo se relacionan estas tecnologías entre sí. Cuando un proveedor de web services quiere poner un servicio a disposición de consumidores del mismo, lo describe a través del WSDL y lo registra en un repositorio de tipo UDDI.

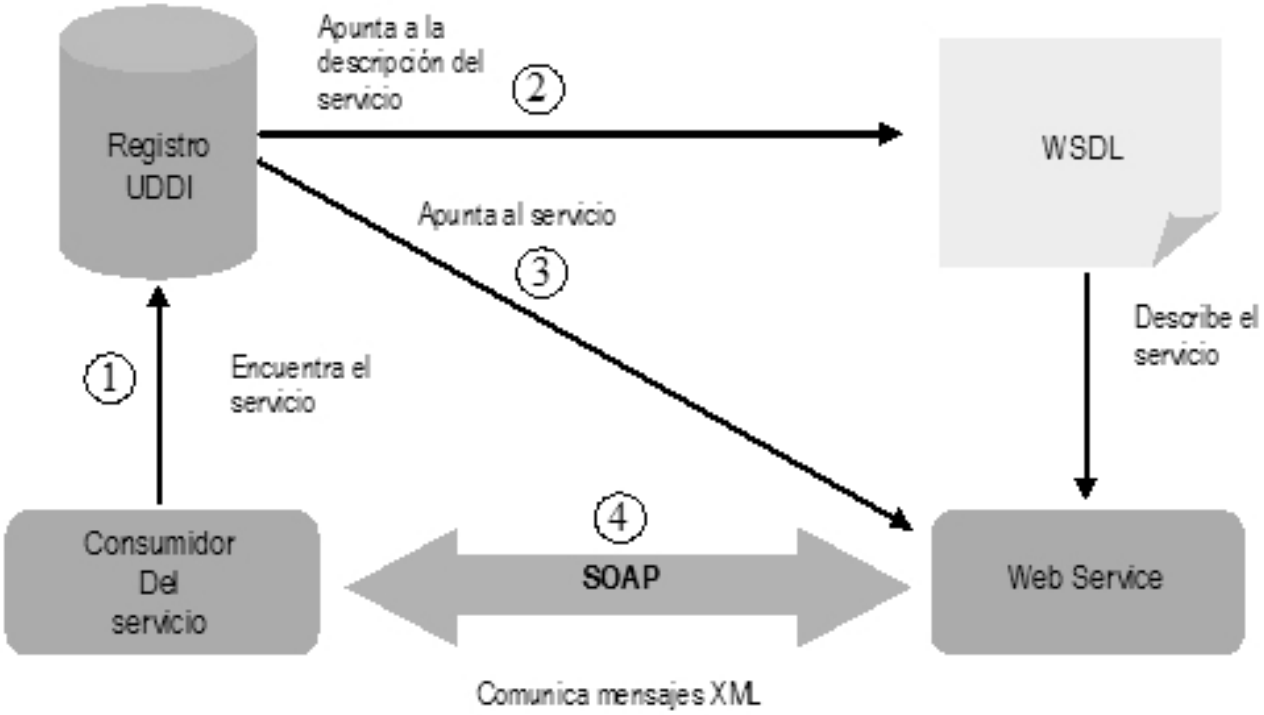


Fig. 2 Las tecnologías centrales de los web services son UDDI, WSDL y SOAP

Cuando el consumidor del servicio necesita un web service, primero lo busca en repositorios UDDI (1), obtiene información del mismo a través del WSDL (2) y un punto de acceso al mismo (3). Luego construye un mensaje tipo SOAP a través del cual comunicarse y utilizar el servicio (4).

1.12 Modelo de acceso a Web Service

Existe una serie de actividades esenciales que deben cumplirse para brindar un entorno de web services que permita explotar sus ventajas:

- El web service debe ser creado y sus interfaces y métodos de invocación deben ser definidos.
- Debe ser publicado en repositorios de intranet o de Internet para que los potenciales usuarios los localicen.
- Debe ser localizado para poder ser invocado.
- Debe ser invocado por el usuario cliente.
- Un web service puede ser dado de baja del repositorio si ya no está más publicado.

Por lo tanto una arquitectura de web services requiere tres operaciones fundamentales: publicar, encontrar y utilizar.

Los usuarios encuentran los web services publicados y luego los utilizan. Este esquema se muestra en la siguiente figura

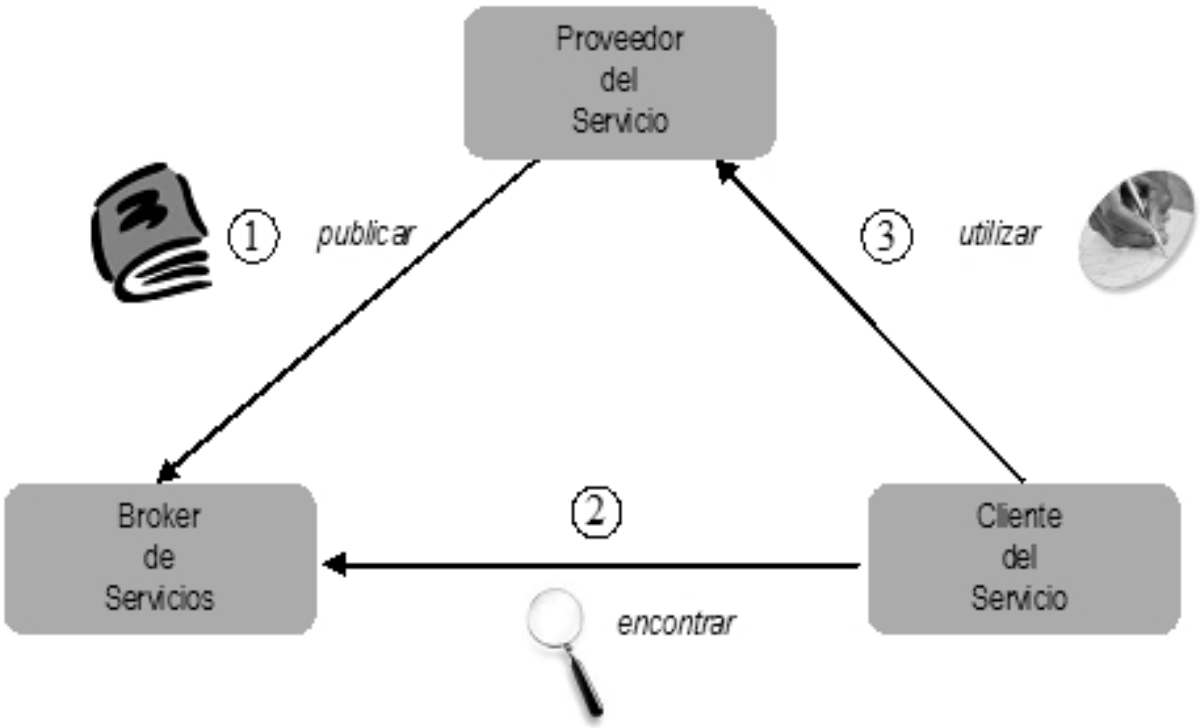


Fig. 3 Diagrama cliente proveedor

El proveedor del servicio publica los servicios disponibles en la red a través de un broker de servicios. El papel del broker reviste la característica de un nivel intermedio de concentración de la oferta de web services. Una vez encontrado el servicio deseado el usuario accede al proveedor del mismo a través de un mecanismo de utilización denominado bind.

1.13 Modelo de seguridad para los servicios Web

La seguridad de los servicios Web puede aplicarse en tres niveles distintos:

1.13.1Seguridad (de punto a punto) para plataformas/transporte

Puede utilizarse el canal de transporte entre dos puntos finales (cliente de servicios Web y servicios Web) para garantizar la seguridad de punto a punto

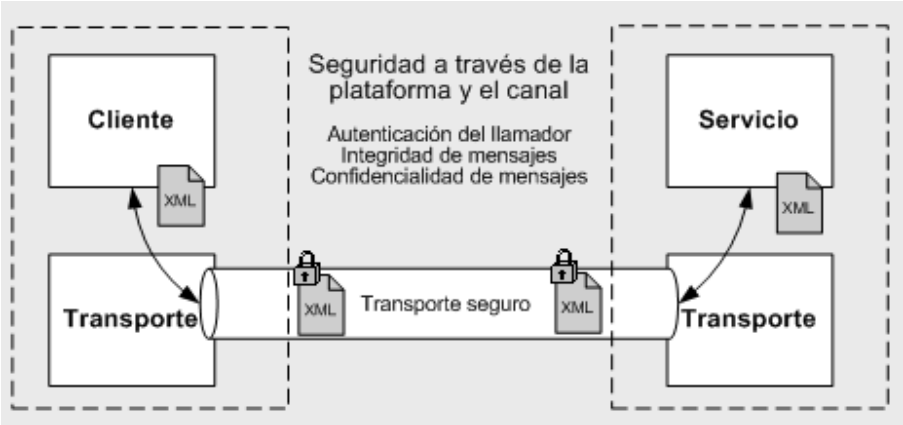


Fig. 4

1.13.1.2 Seguridad (de extremo a extremo) para mensajería

Se trata del enfoque más flexible y eficaz y el que utiliza la iniciativa GXA, en especial para la especificación de seguridad WS-Security

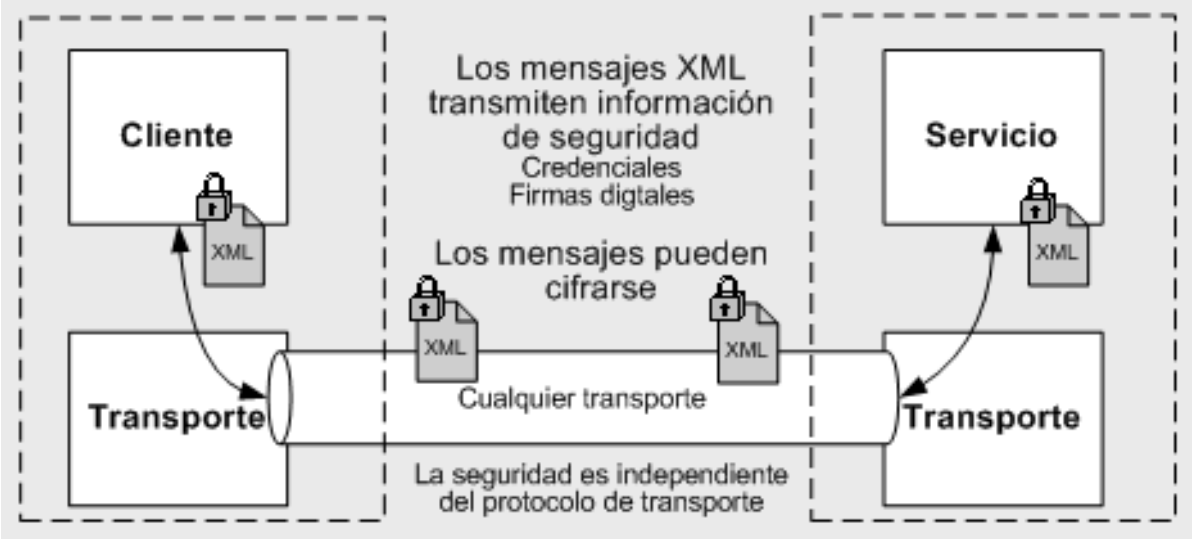


Fig. 5

1.14 Arquitectura de seguridad para plataformas/transporte

Muestra los mecanismos de autenticación y autorización que incluyen los servicios Web de ASP.NET. Cuando un cliente llama a un servicio Web, se desencadena la siguiente secuencia de eventos de autenticación y autorización:

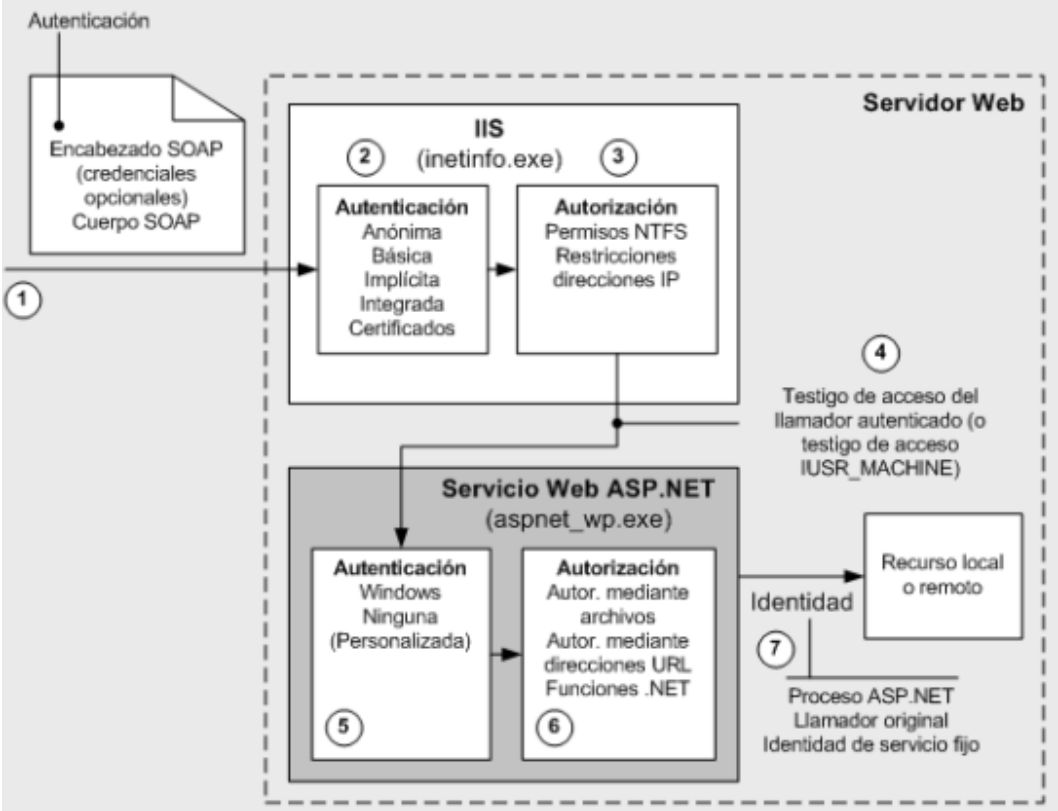


Fig. 6

1.15 El Futuro

Existen organizaciones de standards y difusión de los web services a nivel mundial, tales como World Wide Web Consortium (W3C: www.w3.org) , WS-I (www.ws-i.org) y OASIS (www.oasis-open.org) , que continúan trabajando para seguir robusteciendo el modelo de web services con características de seguridad, disponibilidad, workflow y capacidades transaccionales.

Desde el punto de vista tecnológico, los web services podrían jugar un papel muy importante a la hora de proveer beneficios de convergencia e interoperabilidad a un conjunto de tecnologías alternativas ya existentes.

Los años 2002 y 2003, han sido claves para la consolidación de los web services como tecnologías. Por otro lado, se han transformado en una tecnología de vanguardia sobre la cual

están puestas las miradas de la mayor parte de las organizaciones y proveedores para el desarrollo de aplicaciones de negocios.
Los web services ofrecen una visión neutral, en cuanto a plataformas, pueden ser utilizados para integrar sistemas diversos, han logrado formar su masa crítica mucho más rápido que cualquier tecnología previa de aplicaciones distribuidas y definitivamente serán el foco de la atención de los desarrolladores de este tipo de aplicaciones.

1.16 Ventajas de los servicios Web

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

1.17 Razones para crear servicios Web

La principal razón para usar servicios Web es que se basan en http sobre TCP (Transmission Control Protocol) en el puerto 80. Dado que las organizaciones protegen sus redes mediante *firewalls* -que filtran y bloquean gran parte del tráfico de Internet-, cierran casi todos los puertos TCP salvo el 80, que es, precisamente, el que usan los navegadores. Los servicios Web se vehiculan por este puerto, por la simple razón de que no resultan bloqueados.

-Otra razón por la que los servicios Web son muy prácticos es que pueden aportar gran independencia entre la aplicación que usa el servicio Web y el propio servicio. De esta forma, los cambios a lo largo del tiempo en uno no deben afectar al otro. Esta flexibilidad será cada vez más

importante, dado que la tendencia a construir grandes aplicaciones a partir de componentes distribuidos más pequeños es cada día más acusada.

1.18 Ultimas versiones de Servicios Web

1.18.1 Web Services Security

La especificación WS-Security, que ha sido recientemente ratificada como estándar por OASIS, describe la forma de asegurar los servicios Web en el nivel de los mensajes, en lugar de en el del protocolo de transferencia o en el de la conexión. Las soluciones en el nivel de transporte actuales, como SSL/TLS, proporcionan un sólido cifrado y autenticación de datos punto a punto, aunque presentan limitaciones cuando un servicio intermedio debe procesar o examinar un mensaje. Por ejemplo, un gran número de organizaciones implementan un firewall que realiza un filtrado en el nivel de la aplicación para examinar el tráfico antes de pasarlo a una red interna.

Si un mensaje debe pasar a través de varios puntos para llegar a su destino, cada punto intermedio debe reenviarlo a través de una nueva conexión SSL (consulte la figura 1). En este modelo, el mensaje original del cliente no está protegido mediante cifrado puesto que atraviesa servidores intermedios y para cada nueva conexión SSL que se establece se realizan operaciones de cifrado adicionales que requieren una gran cantidad de programación.

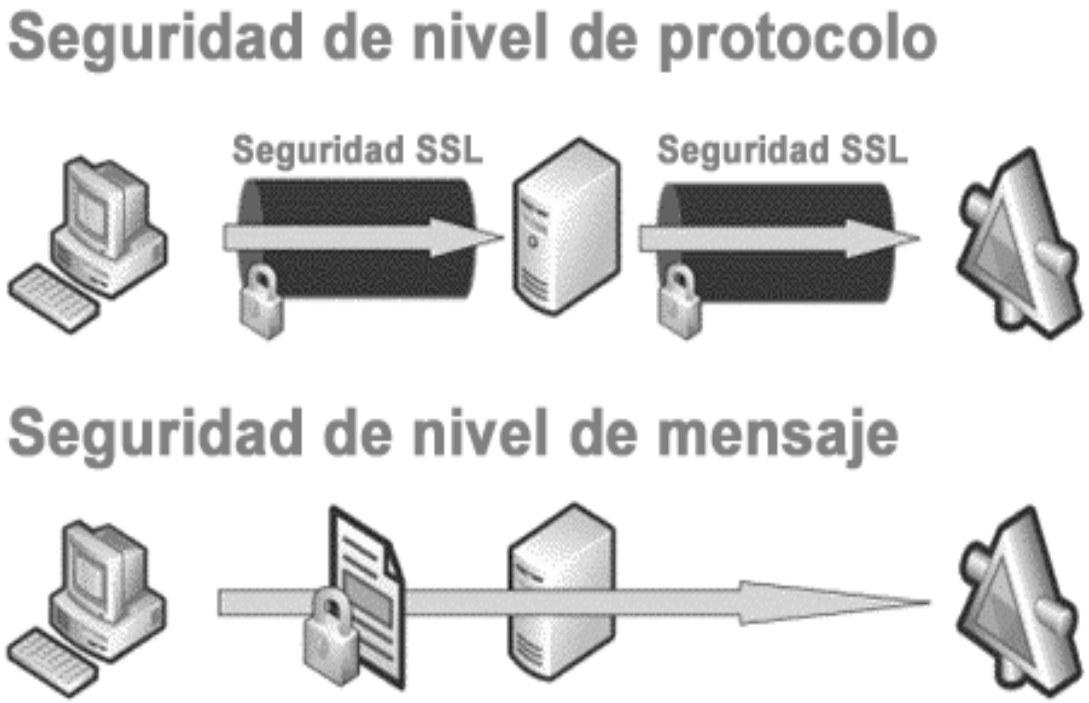


Fig. 7 Seguridad en el nivel del protocolo frente a seguridad en el nivel de los mensajes

Los servicios Web ofrecen comunicación para sistemas con diferentes sistemas operativos y plataformas de desarrollo. Para conseguir este objetivo, se basan en una familia de especificaciones de protocolos industriales para los servicios Web, que generalmente se denomina WS-*. WSE 2.0 proporciona implementaciones de muchas de estas especificaciones para aquellos programadores que deseen ser los primeros en utilizar la tecnología de última generación para los servicios Web. WSE, que se puede descargar gratuitamente, es totalmente compatible y amplía la compatibilidad actual de los servicios Web de .NET Framework. Aunque no se ofrecen garantías de que la versión 2.0 vaya a ser compatible en cuanto a la conexión y el modelo de objeto con las principales futuras versiones de WSE (p. ej., 3.0), la compatibilidad de lado a lado está garantizada. Se esperan unas actualizaciones sencillas y mecánicas para cada nueva versión, así como para "Indigo": la infraestructura de la nueva

generación de mensajería para la plataforma Microsoft Windows. Asimismo, se espera que las versiones de WSE que se lancen antes, o al mismo tiempo que Indigo, sean interoperables con éste en el nivel de la conexión.

Los clientes envían los mensajes a un único servicio de enrutamiento, que está visible públicamente. Este servicio utiliza el direccionamiento y enrutamiento de WSE para inspeccionar los encabezados del mensaje con el fin de determinar a qué servicio de pago interno se debe enviar. Al disponer de un servicio Web que enruta los mensajes SOAP de este modo, se puede "virtualizar" la red física, implementar el equilibrio de carga o permitir que los servicios se quiten o sustituyan sin que esto afecte a los clientes.

1.19 LDAP

LDAP (*Lightweight Directory Access Protocol*) es un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red. LDAP también es considerado una base de datos (aunque su sistema de almacenamiento puede ser diferente) al que pueden realizarse consultas.

Habitualmente, almacena la información de login (usuario y contraseña) y es utilizado para autenticarse aunque es posible almacenar otra información (datos de contacto del usuario, ubicación de diversos recursos de la red, permisos, certificados, etc.).En conclusión, LDAP es un protocolo de acceso unificado a un conjunto de información sobre una red.Trabaja sobre TCP/IP.

1.19.1 Tipo de información que este puede almacenar y como se almacena

El modelo de información de LDAP está basado en entradas. Una entrada es una colección de atributos que tienen un único y global Nombre Distinguido (DN). El DN se utiliza para referirse a una entrada sin ambigüedades. Cada atributo de una entrada posee un tipo y uno o más valores En LDAP, las entradas están organizadas en una estructura jerárquica en árbol. Tradicionalmente, esta estructura reflejaba los límites geográficos y organizacionales. Las entradas que representan países aparecen en la parte superior del árbol. Debajo de ellos, están las entradas que representan los estados y las organizaciones nacionales. Debajo de estás,

pueden estar las entradas que representan las unidades organizacionales, empleados, impresoras, documentos o todo aquello que pueda imaginarse

1.19.2 ¿Cómo se accede a la información en LDAP?

LDAP define operaciones para interrogar y actualizar el directorio. Provee operaciones para añadir y borrar entradas del directorio, modificar una entrada existente y cambiar el nombre de una entrada. La mayor parte del tiempo, sin embargo, LDAP se utiliza para buscar información almacenada en el directorio. Las operaciones de búsqueda de LDAP permiten buscar entradas que concuerdan con algún criterio especificado por un filtro de búsqueda. La información puede ser solicitada desde cada entrada que concuerda con dicho criterio.

1.19.3 ¿Cómo se protege la información en LDAP?

Algunos servicios de directorio no proveen protección, permitiendo a cualquier persona acceder a la información. LDAP provee un mecanismo de autentificación para los clientes, o la confirmación de identidad en un servidor de directorio, facilitando el camino para un control de acceso que proteja la información que el servidor posee. LDAP también soporta los servicios de privacidad e integridad.

1.20 Servicios Web en la Universidad de las Ciencias Informáticas

En nuestra Universidad se hace cada vez más útiles la presencia de los servicios web debido a las ventajas que estos representan y además de ser estos un gran salto de avance en la programación web.Existen una gran cantidad de servicios web que son utilizados en la UCI entre ellos se encuentran:

1.20.1 Akademos

- ✓ ObtenerGrupos
- ✓ ObtenerFacultades

- ObtenerAsignaturasDadoIdPersona
  - ✓ ObtenerGruposDadoFacultad
  - ✓ ObtenerEstudianteDadoIdPersona

1.20.2 Trabajadores

- ✓ ObtenerAreaDadoIdPersona
- ✓ ObtenerTrabajadorDadoIdPersona
- ✓ ObtenerAreaDadoIdArea
- ✓ ObtenerCargoDadoIdTrabajador
- ✓ ObtenerTrabajadoresDadoIdArea

1.20.3 Identificación

- ✓ ObtenerSolapinDadoIdPersona
- ✓ ObtenerCodigoBarraDadoIdPersona
- ✓ **O**btenerIdPersonaDadoSolapin
- ✓ ObtenerIdPersonaDadoCodigoBarra

1.20.4 Guía telefónica

- ✓ ObtenerTelefonoDadoApto
- ✓ ObtenerPropietariosDadoPT
- ✓ ObtenerUbicacionDadoPT
- ✓ ObtenerAreaDadoPT

1.20.5 Telemáticos

- ✓ ObtenerIdPersonaDadoUsuario
- ✓ ObtenerUsuarioDadoIdPersona

1.20.6 Ciudadano

- ✓ ObtenerPersonaDadoId
- ✓ ObtenerPersonaDadoSolapin
- ✓ ObtenerPersonaDadoCI

Todos estos servicios están publicados para que acceda todo el que desee utilizarlos.

1.21 Conclusiones

Con este capítulo tratamos de mostrarles algunas herramientas que se utilizarán para la construcción de nuestro sistema, cada uno de ellas es pieza clave para que se desarrolle con eficiencia y calidad el software, elaborado con la mas actual tecnología y puesto el las manos de ustedes para que sea usado para el desarrollo armónico no solo de la Universidad sino también de la sociedad en general.

CAPITULO 2: CARACTERÍSTICAS DEL SISTEMA

2.1 Introducción

Todos conocemos el ímpetu de los servicios web en la actualidad, lo que nos llevo a realizar nuestro proyecto por esta vía la cual nos brindara rapidez y eficiencia no solo en la gestión de la información sino también en el control de los usuarios y las aplicaciones en sentido general.
Tras una larga investigación en la cual hemos abordado anteriormente en cuanto a las herramientas que utilizaremos en la aplicación nos dimos a la tarea de desarrollar nuestro sistema. Es importante desglosar el trabajo para una mejor comprensión asi como sus características debido a que este será puesto a consideración de los usuarios asi como de la Universidad.
El sistema como tal no tiene un negocio definido, no existe un flujo para desarrollar negocio debido a que no tiene precedentes y se llegó a resolver el problema existente debido a una ardua investigación en la cual llegamos a la conclusión de que era evidente una exitosa solución utilizando un servicio Web.

2.2 Funcionalidad

Como funciona:
Nuestro Web Service presenta varias páginas clientes las cuales pretenden que nuestro software les preste servicio. Ellas para su seguridad presentan un login para tener un control del usuario así como un registro en el cual puede crear una cuenta para su futuro uso en el sistema. Cuando un usuario entra por primera vez a cualquier sitio este le va a pedir un registro que servirá para tener un mejor control de quien es el que esta en el sistema. Este registro irá a la base de datos y será registrado debido a que cuando este mismo usuario acceda a otro portal cliente este sepa quien es el que accedió así como el rol que este desempeña. Este proceso es muy importante

debido a que cada sitio gestiona la seguridad utilizando nuestro servicio web, es fácil acceder a este y proporciona una gran interoperabilidad entre las aplicaciones.
Presentamos a continuación las características del sistema para una comprensión exacta de cómo va a funcionar la aplicación.

2.3 Requerimientos Funcionales

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir, los cuales son la base de la funcionalidad del sistema:

1. Autenticar Usuario.
2. Registrar Usuario.
3. Asignar permiso de navegación en todas las páginas clientes a los usuarios autenticados en al menos una.
4. Mostrar listado de usuarios en línea.
5. Devolver existencia de un usuario
6. Devolver si usuario está autenticado
7. Permitir cerrar sesión.

2.4 Requerimientos no Funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. A continuación se presentan los que posee el caso de estudio actual, a los que se tratará de dar cumplimiento con la aplicación.

1. Usabilidad: Esfuerzo necesario para aprender, operar, preparar entradas e interpretar la salida de un programa. La herramienta será utilizada por operadores que no necesariamente tendrán experiencia en el uso de la computadora, a estos les será trasparente el funcionamiento de la aplicación.

2. Seguro: La información manejada por el sistema debe estar protegida de acceso no autorizado y divulgación. Deberán estar encriptadas las contraseñas que se asociarán a

los distintos usuarios durante el proceso de autenticación de los clientes. La información manejada por el sistema debe ser objeto de cuidadosa protección contra la corrupción y estados inconsistentes. La aplicación deberá estar disponible en todo momento para aquellas personas con acceso a la información y los mecanismos utilizados para lograr la seguridad no deben ser un obstáculo a los usuarios.

3. Interoperabilidad: Esfuerzo requerido para acoplar un sistema con otro. El servicio Web mediara entre las páginas clientes.
4. Reusabilidad: Grado en el que un programa se puede utilizar en otras aplicaciones.
5. Para que la aplicación funcione de manera adecuada deberá correrse en una computadora Pentium 4, con 40 GB de disco duro y 512MB de RAM como mínimo, con Sistema Operativo Windows Server 2003 o Windows XP, debido a que el servicio que presta va a ser utilizado por varias paginas clientes a la misma vez .
6. Fiabilidad. La aplicación debe realizar sus funciones en un tiempo prudencial y sin margen a errores. La fiabilidad de un programa se logra cuando realiza su función con una precisión requerida.
7. Compatibilidad: Es una aplicación multiplataforma, debido a que va a ser usada por páginas clientes programadas en diversos lenguajes, por ejemplo .NET, PHP, entre otras.
8. Como gestor de base de datos se usará SQLServer y en caso de los clientes de páginas PHP, se usará un servidor Apache.
9. Integridad: La integridad es el grado en el que se controla el acceso al programa o los datos por usuarios no autorizados. Esto se controla en la aplicación mediante la distribución de permisos y accesos de los usuarios.
10. Disponibilidad: Es el grado en que un programa está disponible para ser usado por cualquier otro programa. Va a ser utilizado por la aplicación que desee utilizar el servicio web, puesto que va a estar publicado.
11. Eficiencia: Cantidad de recursos y código requeridos por un programa para realizar una función.

12. Portabilidad: Esfuerzo requerido para transferir un programa de una configuración hardware o entorno software a otro.

13. La herramienta propuesta podrá ser usada bajo cualquier sistema operativo; preferiblemente Windows server 2003 o Windows XP así como una PC con 512 de RAM y 40 GB de disco duro como mínimo debido a que el Web service estará en un servidor del cual se conectaran varias aplicaciones a consumir a la vez. Esta aplicación se desarrollo utilizando Visual Estudio 2005.

14. Corrección: Grado en el que un programa satisface las especificaciones y cumple los objetivos del usuario.

2.5 Diagrama de casos de uso del sistema

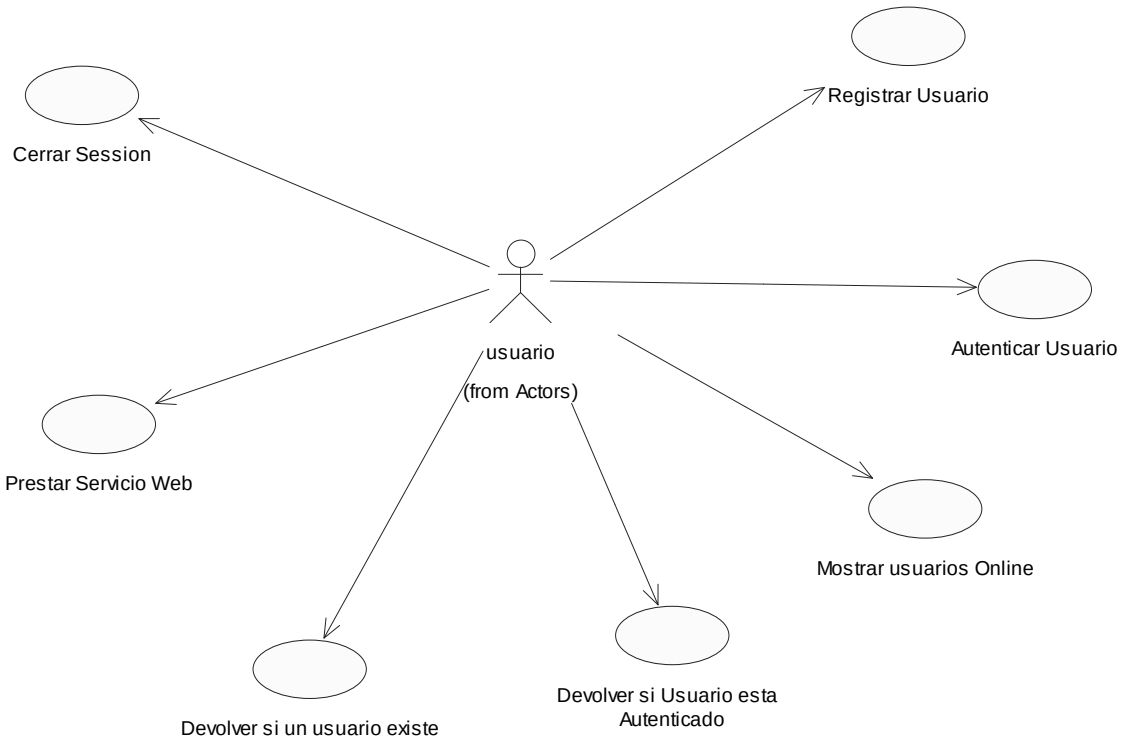


Fig. 8 Diagrama de casos de uso del sistema

2.6 Descripción de los casos de uso del sistema

Por la importancia que esto proporciona a la aplicación es necesario la descripción de los casos de uso del sistema para tener claro el funcionamiento de este y tratar de ser claros a la hora de solicitar el servicio Web.

2.6.1 Autenticar Usuario

Caso de Uso:	**Autenticar Usuario**	
Actor(es):	Usuario (inicia)	
Propósito:	Permitir al usuario acceder a cualquiera de las páginas que utilizan el servicio web.	
Resumen:	El caso de uso se inicia cuando un usuario desea consultar una página que utiliza el servicio.	
Referencias:	RF1, RF4	
Precondiciones:	El usuario se haya registrado.	
Acción del Actor		**Respuesta del Sistema**
1. Autenticación		1.1 Verifica si el usuario y contraseñas son correctos. 1.2 el sistema le da el acceso a la navegación
2. Accede a la navegación		

2.6.2 Registrar Usuario

Caso de Uso:	**Registrar Usuario**
Actor(es):	Usuario (inicia)
Propósito:	Permitir al usuario acceder a cualquiera de las páginas que utilizan el servicio web.
Resumen:	El caso de uso se inicia cuando un usuario desea consultar una página que utiliza el servicio y no se ha registrado previamente.
Referencias:	RF1, RF3
Precondiciones:	

Acción del Actor	**Respuesta del Sistema**
1. Registrarse	1.1 presenta un formulario para que el usuario lo llene.
2. Llena el formulario de Registro	2.1 Guarda la información del formulario en la base de datos.
3. Obtiene una cuenta para navegar a través de las páginas que utilizan el servicio web.	

2.6.3 Prestar servicio a las aplicaciones utilizadas por los usuarios

Caso de Uso:	**Prestar servicio web**
Actor(es):	Usuario (inicia)
Propósito:	Permitir al usuario acceder a cualquiera de las páginas que utilizan el servicio web, sin necesidad de autenticarse en cada una de las paginas que utilizan el servicio.
Resumen:	El caso de uso se inicia cuando un usuario desea consultar una página que utiliza el servicio y no se ha registrado previamente.
Referencias:	RF1, RF3
Precondiciones:	El usuario este autenticado
Acción del Actor	**Respuesta del Sistema**
Después que el usuario se autentica podrá acceder a cada una de las páginas que utilizan el servicio web	

2.6.4 Devolver si un usuario existe

Caso de Uso:	**Prestar servicio web**
Actor(es):	Usuario (inicia)
Propósito:	Dado un login y un password devolver si el usuario existe
Resumen:	El caso de uso se inicia cuando un usuario desea consultar una página que utiliza el servicio y no se ha registrado previamente.
Referencias:	RF1,
Precondiciones:	
Acción del Actor	**Respuesta del Sistema**
Este caso de uso comienza cuando una aplicación desea verificar si el usuario que presenta sus credenciales son correctas y además existe. El servicio web le devuelve verdadero si	

existe y falso si no, en este caso deberá registrarse para poder acceder a la navegación.

2.6.5 Devolver si un usuario esta autenticado

Caso de Uso:	**Prestar servicio web**
Actor(es):	Usuario (inicia)
Propósito:	Dado un login y un password devolver si el usuario esta autenticado
Resumen:	
Referencias:	RF1,
Precondiciones:	
Acción del Actor	**Respuesta del Sistema**
Este es un servicio que presta la aplicación, cada cierto tiempo hace una verificación para saber si el usuario esta autenticado, es parte del control de seguridad para si en caso de que sea verdadera entonces autenticarlo	

2.6.6 Cerrar sesión

Caso de Uso:	**Prestar servicio web**
Actor(es):	Usuario (inicia)
Propósito:	Que el usuario salga del sistema una vez haber terminado su navegación
Resumen:	
Referencias:	RF1,
Precondiciones:	
Acción del Actor	**Respuesta del Sistema**
El usuario una vez terminada su navegación se desloguea informando al sistema del mismo, esto es importante para tener un mejor control de los usuarios	

2.6.7 Mostrar usuarios online

Caso de Uso:	**Prestar servicio web**
Actor(es):	Usuario (inicia)
Propósito:	Después que los usuarios están utilizando es servicio web tener un control de los que están online en un momento determinado
Resumen:	
Referencias:	RF1 RF3
Precondiciones:	
Acción del Actor	**Respuesta del Sistema**
Este proceso se realiza para tener un conocimiento de quienes son los usuarios que están utilizando mi servicio, de ahí se pueden sacar en gran medida información importante para promover futuros servicios. Comienza cuando el usuario se autentica y realiza su navegación, el servicio le muestra quienes son los usuarios que están online en ese momento.	

2.7 Conclusiones

En este capítulo nos dimos a la tarea de ofrecerles las características del software para tener una visión exacta de su funcionamiento así como cual es el ambiente en cuanto a hardware y sistema operativo además de otras características no propiamente del sistema debatidas anteriormente.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

3.1 Introducción

En este capítulo debemos mostrar como será el análisis de nuestro Web Service, las clases que utilizaremos así como los métodos con los cuales daremos cumplimientos a las principales funcionalidades de la aplicación.

En este caso hemos decidido realizar un diagrama general para mostrarles como funcionará nuestro sistema teniendo en cuanta los requerimientos funcionales del mismo. El tipo de clase que utilizaremos es fundamental ya que el usuario no tendrá contacto directo con nuestro servicio sino las aplicaciones.

Clases del diseño:

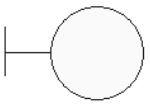
Clase Interfaz:

1. Modelan la interacción Actor – Sistema.
2. Ventanas, Formularios, comunicación con otros sistemas o dispositivos

¿Cómo obtenerlas?

Las identificamos a partir de los actores:

1. Una clase para cada interacción Actor – Caso Uso.
2. Una clase para cada sistema externo.
3. Una clase para cada dispositivo que se utilice



Clase Interfaz

Fig. 9

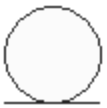
Clase Entidad:

1. Modelan la información del Sistema.
2. Modelan el comportamiento asociado a una información.

¿Cómo obtenerlas?

Las identificamos a partir de:

1. Objetos o Entidades del Negocio
2. Glosario Términos.
3. Conceptos de los modelos conceptuales o Modelo Dominio



Clase Entidad

Fig. 10

Clase Control

1. Coordinan el trabajo de las clases.
2. Encapsulan comportamiento de un CU.
3. Funciones complejas

¿Cómo obtenerlas?

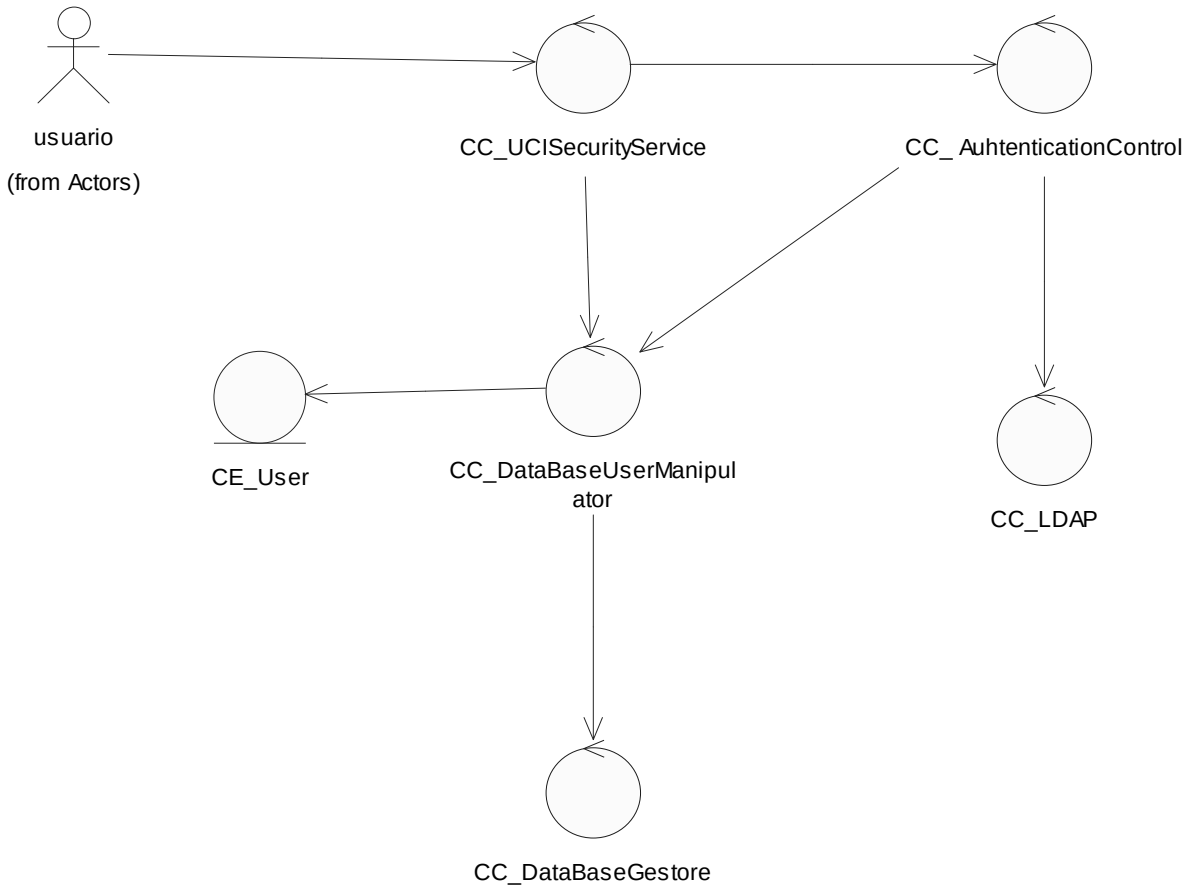
1. En principio se define una clase control por cada Caso de Uso.
2. No usar clase control si el flujo es simple.
3. Crear mas de una cuando algún flujo puede re-usarse en otro CU
4. Una clase control por cada Actor

Clase Controladora

Fig. 11

3.2 Diagrama de las clases de análisis

Fig. 12



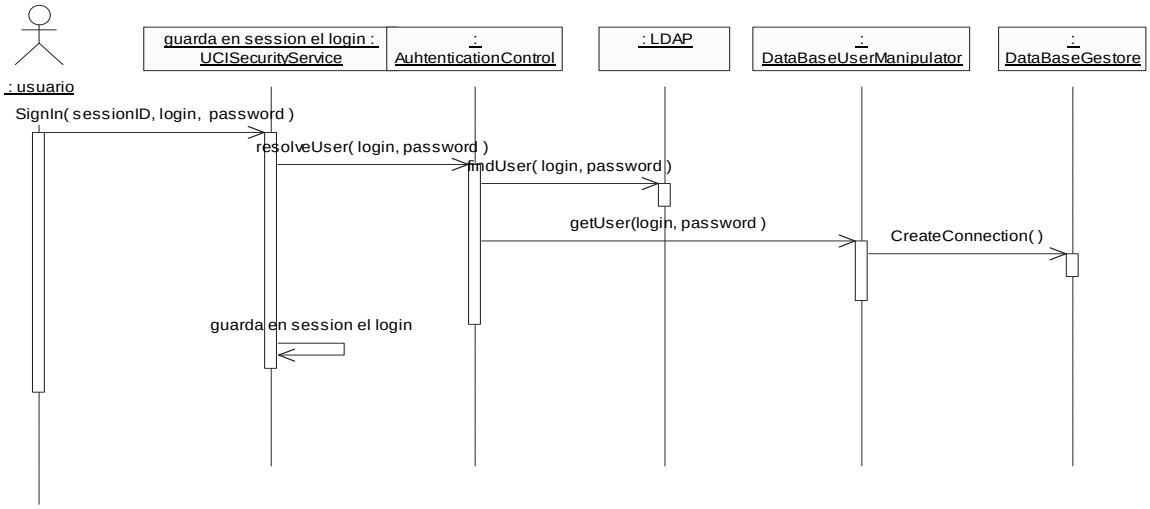
3.3 Diagramas de secuencia

En esta parte se define como su nombre lo indica como serán utilizadas las clases, es decir el orden en que aparecerán en los distintos casos de uso del sistema. En el proyecto existen varios casos de uso pero como tal describiremos los diagramas con los principales, debido a la importancia que estos requieren en la aplicación.

- ✓ Autenticar usuarios
- ✓ Registrar usuarios

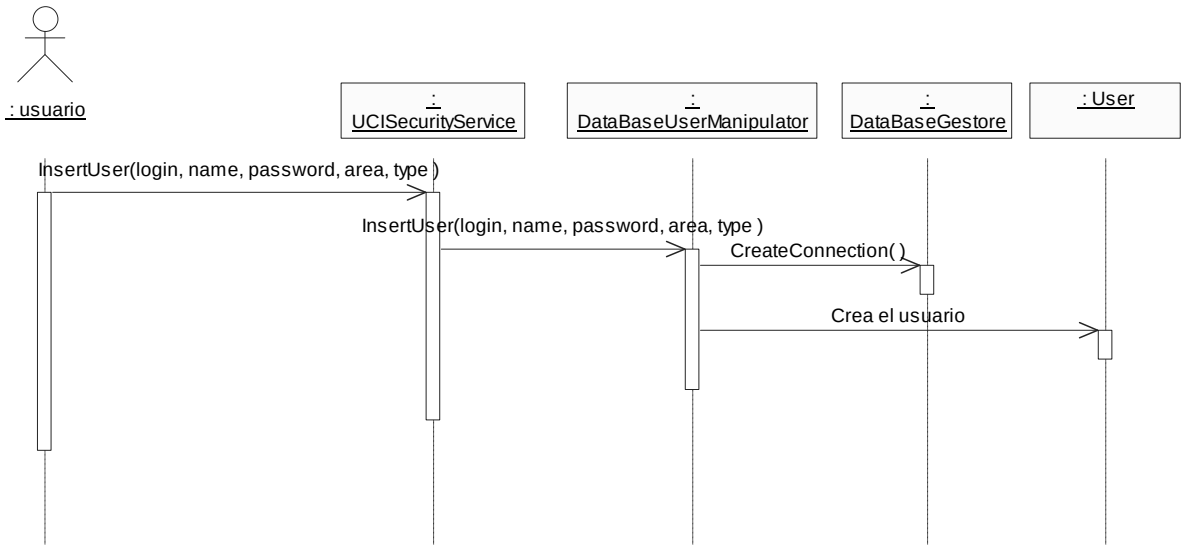
3.3.1 Diagrama de secuencia (Autenticar usuario)

Fig. 13



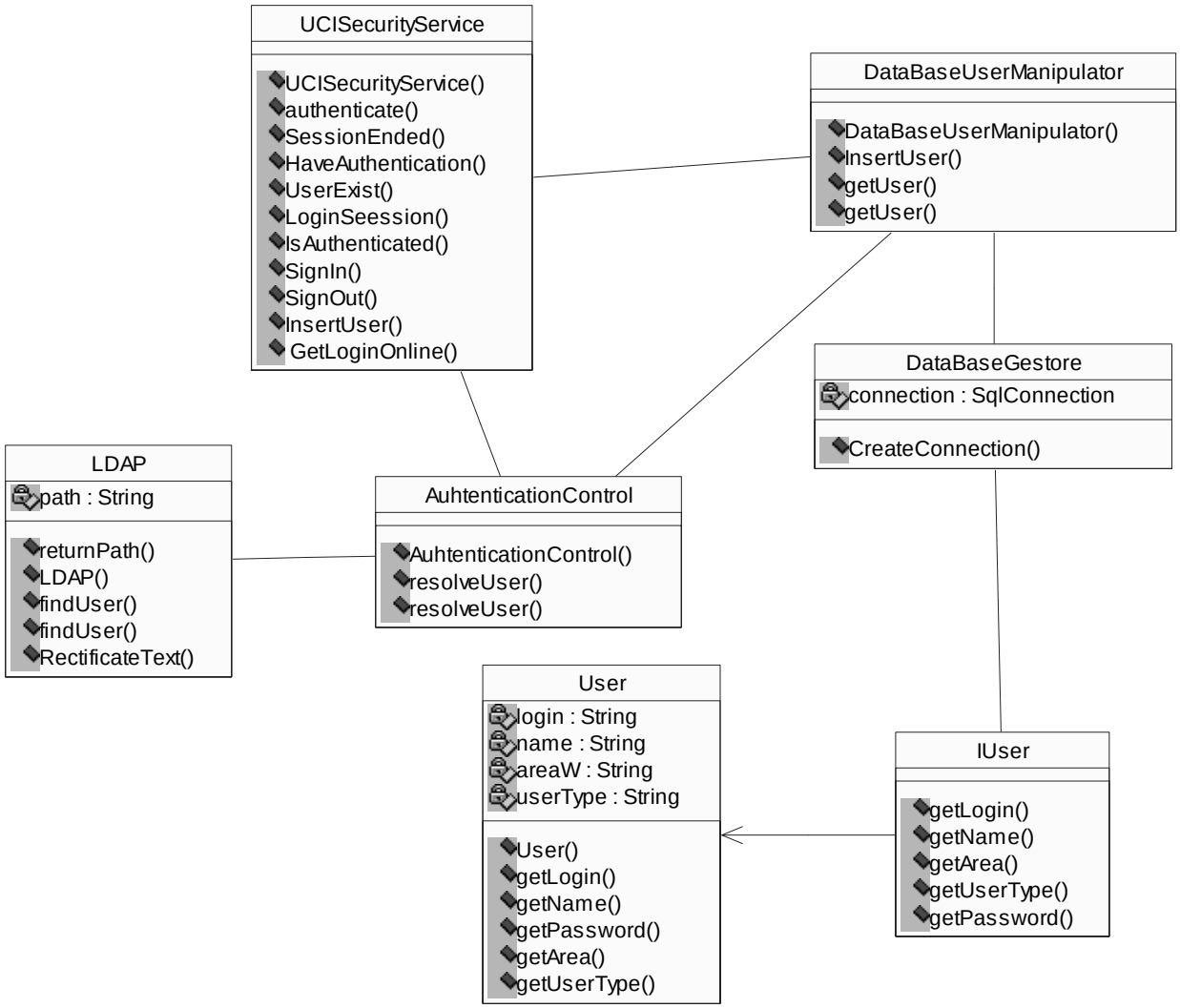
3.3.2 Diagrama de secuencia (Registrar usuario)

FIG. 14



3.4 Diagrama de clases del diseño

FIG.15



3.5 Descripción de clases del diseño

3.5.1 Clase: UCISecurityService

Nombre: UCISecurityService	
Tipo de clase: Controladora	
Para cada responsabilidad:	
Nombre:	**UCISecurityService**
Descripción:	Método constructor de la clase.
Nombre:	**HashTable authenticate()**
Descripción:	Método privado devuelve una tabla hash con llave un string, que identifica la sesión del explorador donde se conectan y valor una instancia de la clase usuario.
Nombre:	**bool HaveAuthentication(string login, string password)**
Descripción:	Método público que devuelve si un usuario dado por el login y password ese usuario está registrado en la aplicación con estos parámetros.
Nombre:	**bool UserExist(string login)**
Descripción:	Método público que devuelve si dado el login un usuario, si este está registrado en la aplicación.
Nombre:	**string LoginSeession(string sessionID)**
Descripción:	Método público que devuelve el login del usuario que esta logueado en ese momento dada su sesión.
Nombre:	**bool IsAuthenticated(string sessionID)**
Descripción:	Método público que devuelve si el usuario esta autenticado dado el identificador de sesión
Nombre:	**bool SignIn(string sessionID, string login, string password)**
Descripción:	Método público que me devuelve si el usuario descrito por estos parámetros

	a sido autenticado en el sistema.
Nombre:	**void SignOut(string sessionID)**
Descripción:	Método público que permite al usuario cerrar sesión.
Nombre:	**bool InsertUser(string login, string name, string password, string area, string type)**
Descripción:	Método público que permite insertar un nuevo usuario al sistema, y devuelve si fue posible lograrlo o no.
Nombre:	**string[] GetLoginOnline()**
Descripción:	Método público que devuelve un listado con todos los usuarios que están autenticados en el sistema en ese momento.

3.5.2 Clase: AuhtenticationControl

Nombre: AuthenticationControl	
Tipo de clase: Controladora	
Para cada responsabilidad:	
Nombre:	**AuthenticationControl()**
Descripción:	Método constructor de la clase.
Nombre:	**IUser Resolve user(login,password)**
Descripción:	Método que me devuelve o me permite verificar si el usuario existe, con esta contraseña en LDAP o en caso de no existir lo verifica en la Base de datos.
Nombre:	**IUser Resolve user(login)**
Descripción:	Método que me devuelve o me permite verificar si el usuario existe en LDAP o

	en caso de no existir lo verifica en la Base de datos.

3.5.3 Clase: DataBaseUserManipulator

Nombre: DataBaseUserManipulator	
Tipo de clase: Controladora	
Para cada responsabilidad:	
Nombre:	**DataBaseUserManipulator()**
Descripción:	Método constructor de la clase.
Nombre:	**bool InsertUser(usr)**
Descripción:	Método que inserta en la Base de Datos el usuario previamente registrado
Nombre:	**IUser getUser(login, password)**
Descripción:	Método que me devuelve el usuario con una consulta a la Base Datos, que contenga un login y un password específico.
Nombre:	**IUser getUser(login)**
Descripción:	Método que me devuelve el usuario con una consulta a la Base Datos, que contenga un login específico.

3.5.4 Clase: DataBaseGestore

Nombre: DataBaseGestore	
Tipo de clase: Controladora	
Atributo connection	**Tipo** SQL connection
Para cada responsabilidad:	
Nombre:	**CreateConnection()**

Descripción:	Método que crea la conexión con la Base de Datos para realizar cualquier operación.

3.5.5 Clase: User

Nombre: User	
Tipo de clase: Entidad	
Atributo login name areaW userType	**Tipo** string string string string
Para cada responsabilidad:	
Nombre:	**User(login,name,password,area,userType)**
Descripción:	Método constructor de la clase.
Nombre:	**string getLogin()**
Descripción:	Método que devuelve el login.
Nombre:	**string getName()**
Descripción:	Método que devuelve el nombre.
Nombre:	**string getPassword()**

Descripción:	Método que devuelve el password.
Nombre:	**string getArea()**
Descripción:	Método que devuelve el área del usuario
Nombre:	**string getUserType()**
Descripción:	Método que devuelve el tipo de usuario que es en el sistema

3.5.6 Clase: LDAP

Nombre: LDAP	
Tipo de clase: controladora	
Atributo phat	**Tipo** string
Para cada responsabilidad:	
Nombre:	**LDAP()**
Descripción:	Método constructor de la clase
Nombre:	**String ReturnPath()**
Descripción:	Método que devuelve el string de conexión con el servidor LDAP, configurado en un xml.
Nombre:	**FindUser(login,pass)**

Descripción:	Método que busca si un usuario descrito por su login y password, está registrado en el servidor LDAP.
Nombre:	**findUser(login)**
Descripción:	Método que busca si un usuario descrito por su login, está registrado en el servidor LDAP.
Nombre:	**RectificateText(TextoARectific)**
Descripción:	Método que obtiene un substring del texto a rectificar.

3.6 Diseño de la Base de Datos

En servicio web está bien relacionado con la base de datos debido a que este estará en contacto con varias aplicaciones a la vez, esto provoca que se estén registrando usuarios constantemente de los cuales debe de tener un control. En sí la base de datos no es grande debido a que presenta una tabla en la cual entran los datos que el usuario lleno previamente en el registro.

3.6.1 Descripción de la tabla

Nombre: Security		
Descripción: esta tabla almacena usuarios además de información importante de cada uno de ellos previamente elaborado en el registro de usuario		
Atributo	**Tipo**	**Descripción**
login	string	El nombre del usuario
name	string	Nombre del usuario en el sistema
password	string	Contraseña del usuario en el sistema
area	string	Área a la cual representa el usuario

type	string	Tipo de usuario que es en el sistema

3.7 Conclusiones

En este capítulo tratamos lo relacionado con el análisis y diseño de nuestro sistema el cual es aspecto importante para la explicación de su funcionamiento en general. Además vimos la tabla de nuestra base de datos y el tipo de datos que almacenarán, también las clases que participan desglosando sus descripciones para un mejor entendimiento no solo de las funcionalidades principales sino de todo nuestra aplicación.

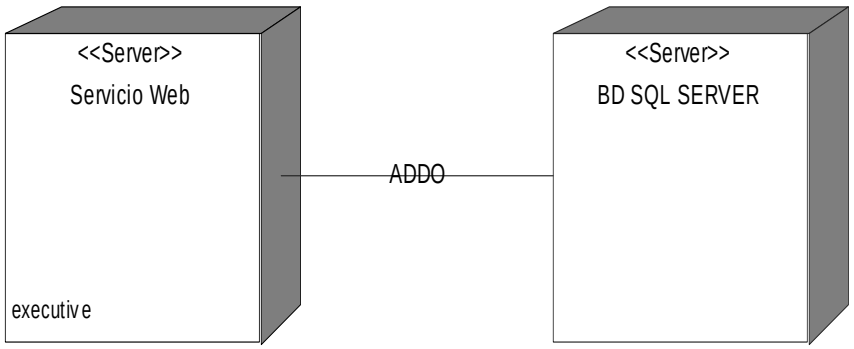
CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

4.1 Introducción

En este capítulo trataremos todo lo relacionado con la implementación de la aplicación, es decir, todos los componentes utilizados así como los dispositivos involucrados en sus respectivos diagramas.Es importante observar que en caso de las pruebas se excluyen del sistema ya que este es un servicio web el cual estará publicado pero sólo para las aplicaciones, es decir que no tendrá una interfaz directa con el usuario, pero para darle solución a esto trataremos las pruebas directamente a los servicios que presta nuestro servicio web.

4.2 Diagrama de despliegue

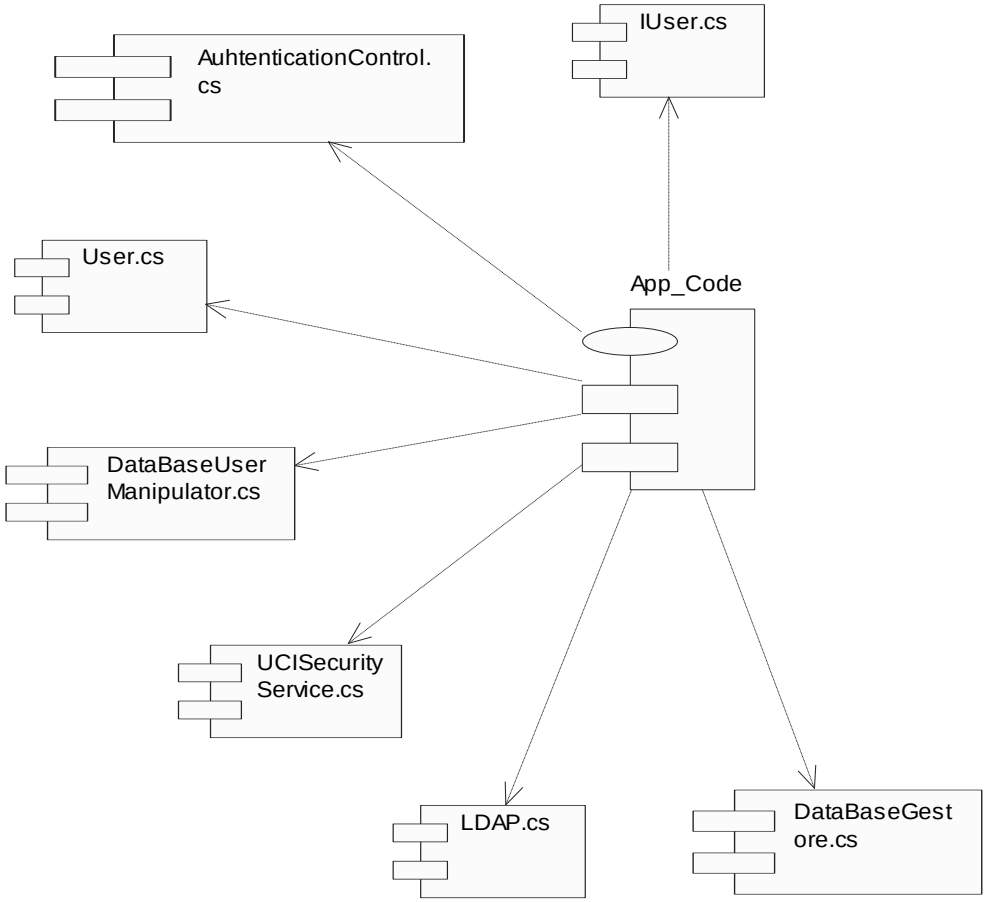
Fig.16



En el diagramase muestra los dispositivos que hacen falta para implementar el servicio web. Como todos conocemos después de publicar el servicio podrán acceder a él cualquier aplicación con el objetivo de tener una buena seguridad la cual es evidente que nuestra aplicación brinda. Es importante ya que sabemos que el web service estará en una computadora tener la base de datos en otra lo cual nos brindará rapidez y a su vez hacer una navegación segura que fue para lo cual fue diseñado el sistema.

4.3 Diagrama de componentes

Fig.17



Este diagrama muestra las clases en el app.code las cuales son parte fundamental del funcionamiento del web service

4.4 Modelo de pruebas

Nombre del caso de uso: Autenticar usuario

Entrada	**Resultados**	**Condiciones**
El usuario accede al sitio con login y password	Nuestro sistema aclara al portal que el usuario tiene acceso a utilizar el sitio	Usuario este registrado en el sistema

Nombre del caso de uso: Registrar usuario

Entrada	**Resultados**	**Condiciones**
El usuario se registra llenando el formulario	El usuario posee una cuenta para utilizar nuestro sistema	Acceder y conocer nuestro servicio así como las páginas clientes

En general estos dos son los casos de uso más importantes pues los demás son servicios que nuestro portal realiza directamente con las aplicaciones clientes como son:

- ✓ Mostrar usuarios online
- ✓ Devolver si un usuario existe
- ✓ Devolver si un usuario está autenticado
- ✓ Cerrar sesión
- ✓ Prestar el servicio de navegación

4.5 Conclusiones

En este capítulo tratamos los componentes que utilizaremos además de su descripción, además de las pruebas que le hacemos a los métodos para su buen funcionamiento. Esto es de mucha importancia para que nuestro software tenga la calidad necesaria.

CONCLUSIONES GENERALES

Tras una intensa investigación en la cual nos dimos a la tarea de recopilar información de sistemas semejantes a este pero con distintas funcionalidades, nos dimos cuenta que esta era la vía correcta para la solución inmediata del problema existente: la autenticación compartida. La realización de este proyecto trajo consigo que una aplicación sólo conociendo la dirección de nuestro servicio web pueda configurar con este en gran medida su seguridad, teniendo en cuenta quienes son los usuarios que están no sólo en ese momento sino quienes la han visitado. Ha sido un trabajo que se pone en las manos de todos los que consideran que la seguridad es parte fundamental del mundo de la informática hoy en día.

RECOMENDACIONES

Todo está en constante desarrollo, las herramientas de trabajo cada vez se hacen más fuertes por lo cual es indudable la futura renovación de esta aplicación. En nuestro caso este trabajo fue diseñado para ayudar a mejorar la seguridad de las aplicaciones teniendo un control de quienes son los usuarios que navegan en ellas.

Al servicio web se le pueden agregar otras funciones para satisfacer a los clientes como por ejemplo:

- ✓ Tener un control del portal que el usuario accedió por primera vez.
- ✓ La cantidad de aplicaciones que utilizan el servicio Web.
- ✓ El portal en el cual el usuario está en un momento determinado.

Son aspectos en los cuales se podrían profundizar los cuales ayudarán a mejorar la seguridad de las páginas y con esto que los usuarios encuentren en este trabajo una herramienta más para perfeccionar la seguridad en sus sistemas.

BIBLIOGRAFÍA

Citada

Libros

- ✓ C# a fondo autor: McGraw-HillAnteramerican páginas: 352 España 2001
- ✓ Object Oriented Programming in C# autores: Robert J. Oberg Howard Lee Harkness

  Páginas: 621 2002
- ✓ C# .N ET Web Developer's Guide autores: Adrian Turtschi DotThatCom.com Jason Werry

  Greg Hack Joseph Albahari

  Páginas: 781 2002

Consultada

Páginas web

1- http://www.mcgraw-hi1l.es

2- http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/art51.asp

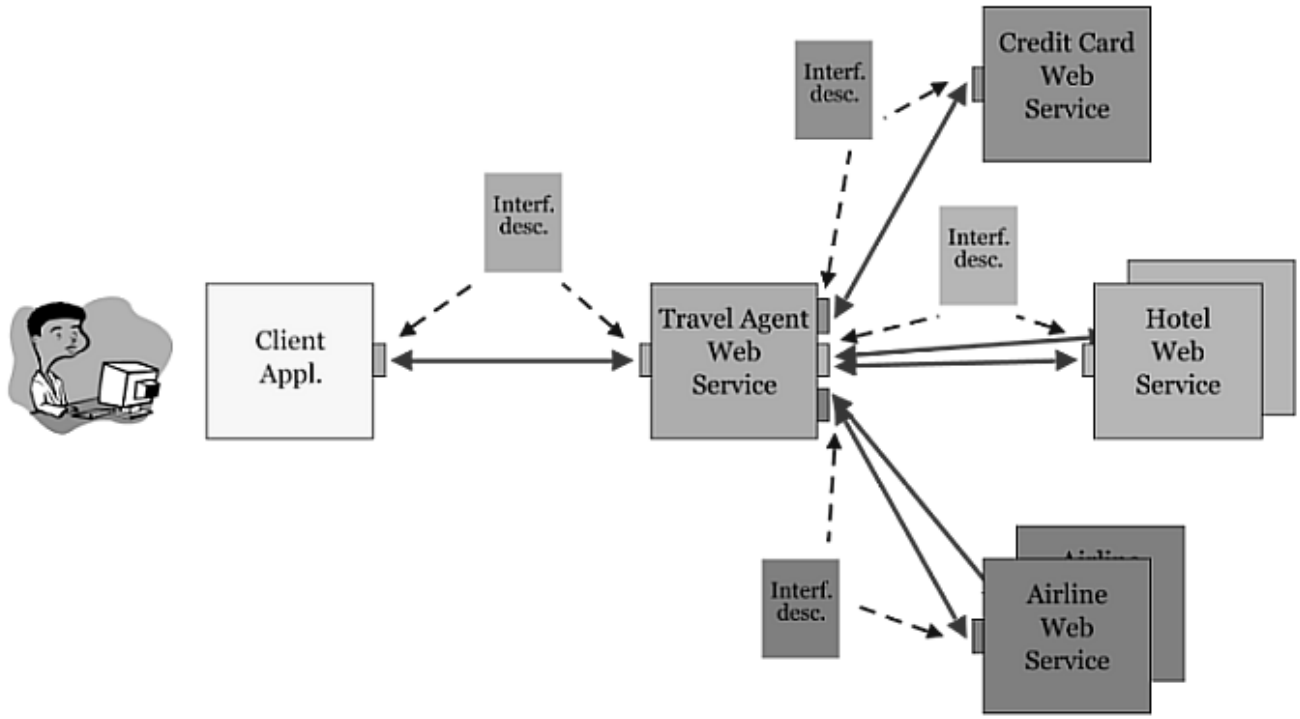
3- http://www.microsoft.com/spanish/msdn/articulos/archivo/141103/voices/wssecdrill.asp

4- http://msdn2.microsoft.com/es-es/library/system.web.sessionstate.httpsessionstate.sessionid(VS.80).aspx

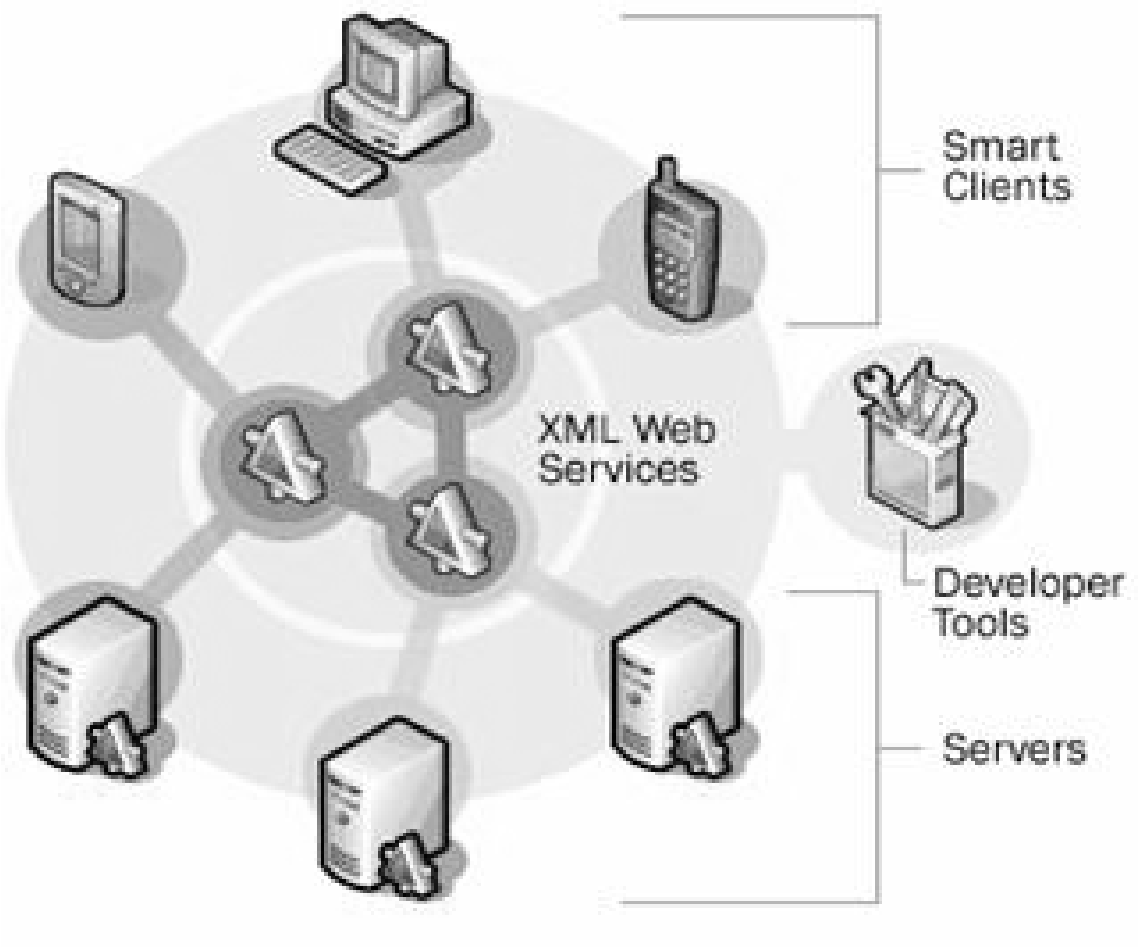
5- http://msdn2.microsoft.com/es-es/library/system.web.sessionstate.httpsessionstate_properties(vs.80).aspx

ANEXOS

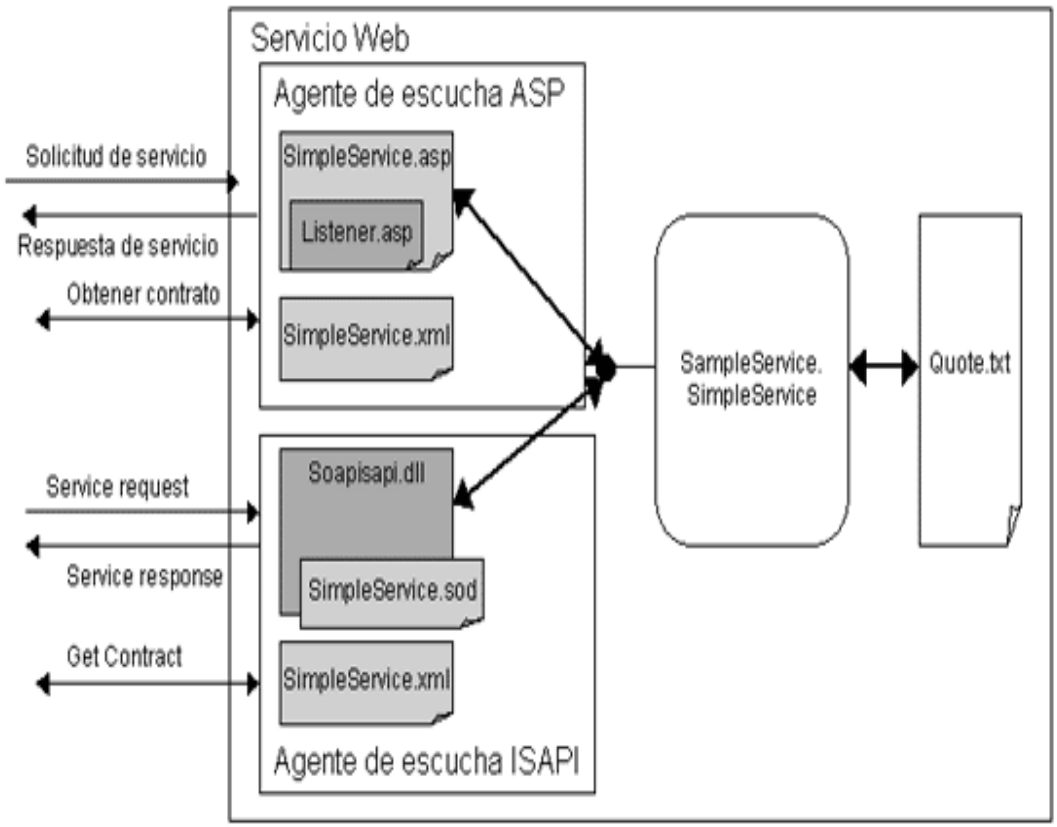
Ejemplos de servicios web:



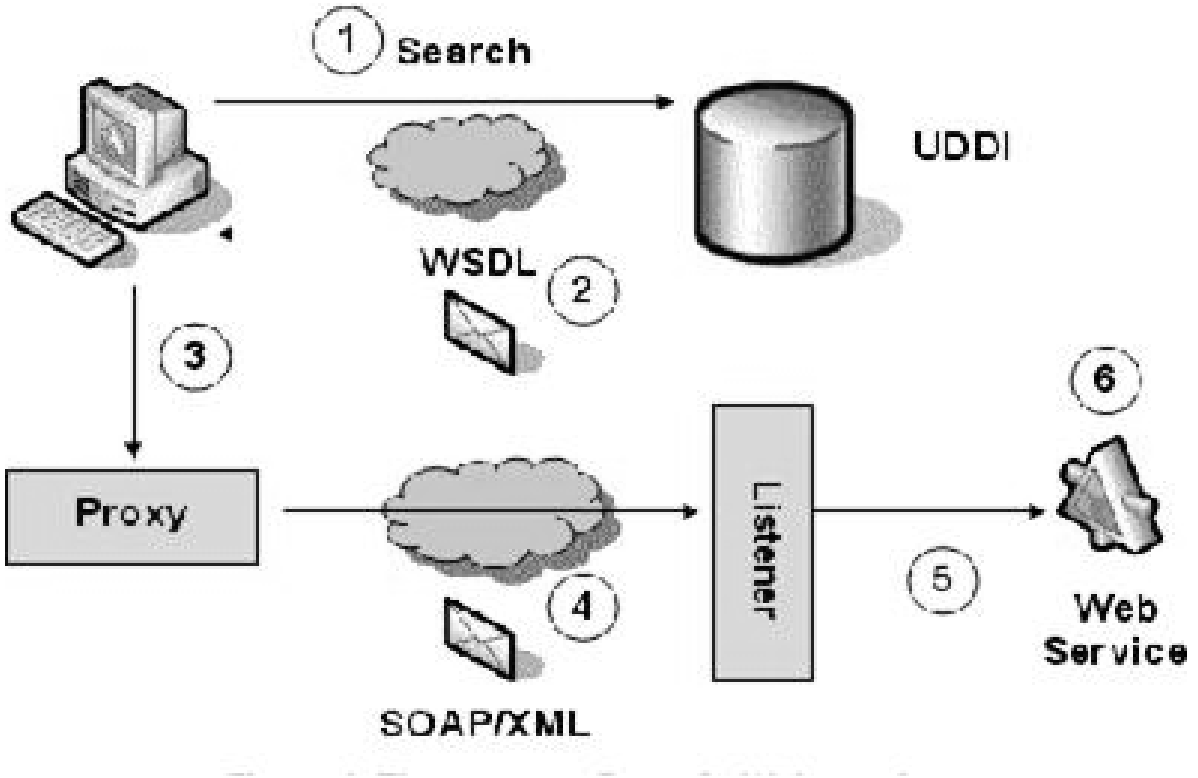
Ejemplo de servicios en el mundo actual



Como al servicio web pueden acceder las aplicaciones y como está definida la trasmisión mediante XML

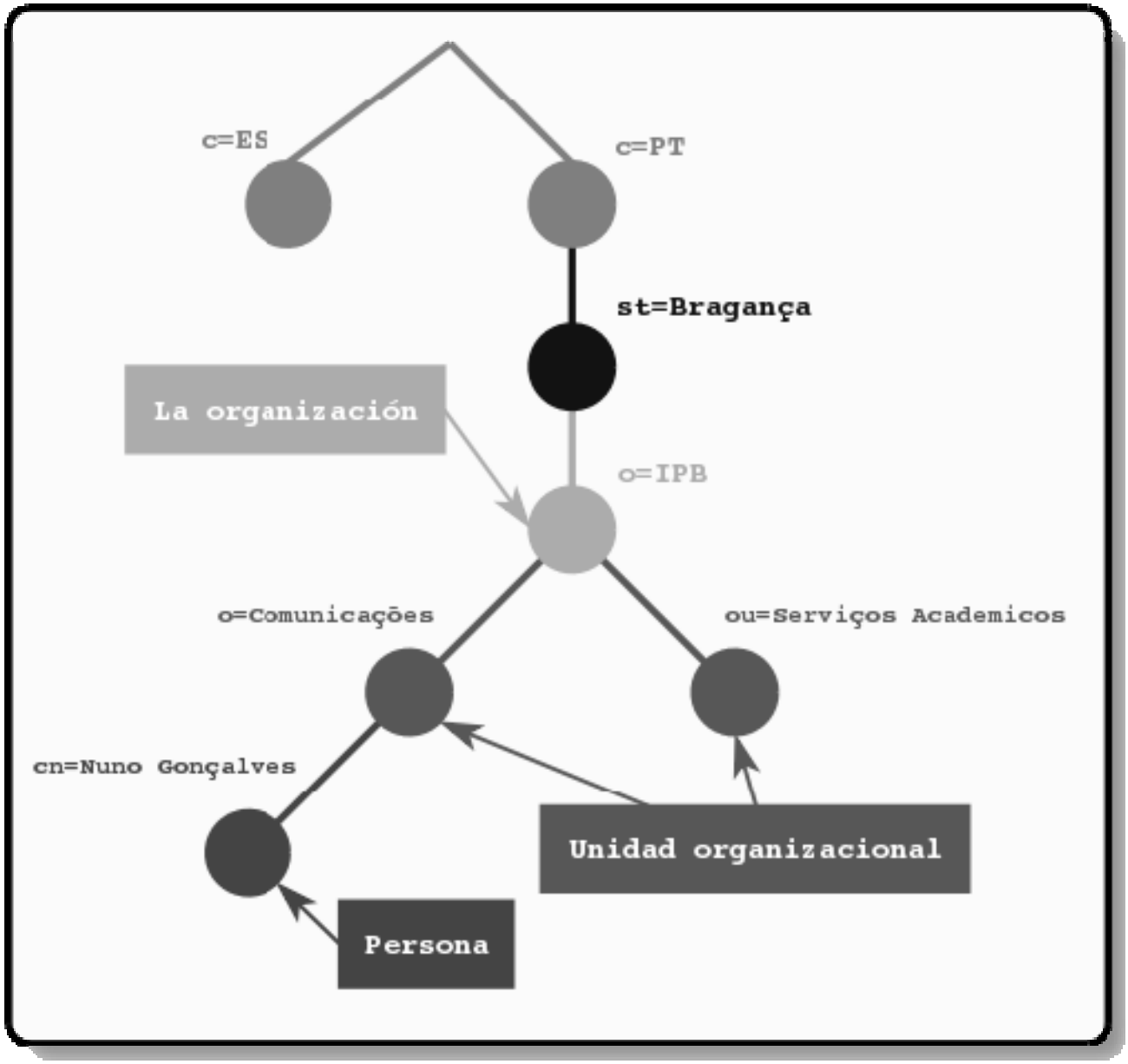


Ejemplo de cómo el servicio web atiende y responde a una solicitud



Como está publicado el servicio y los procesos para poder acceder a él

Funcionamiento de LDAP



GLOSARIO DE TERMINOS

Trataremos los términos no comunes:

Web service - servicio web

Método - líneas de código que realizan una determinada función

Portal digital - página web

Security - seguridad

Aplicación - programa realizado en un determinado lenguaje de programación la cual puede ser aplicación web o de escritorio.

CC - Clase controladora

CE - Clase Entidad

CI - Clase Interfaz

1