# Gestión de Identidad en la Nube: Un caso usando SAML - Security Assertion Markup Language

**Romero, María Soledad - García Mattío, Mariano**
*Universidad Tecnológica Nacional, Facultad Regional Córdoba*
*Instituto Universitario Aeronáutico*

**Abstract**
*El propósito de este trabajo es realizar una introducción sobre el estado del arte en cuanto a los distintos tipos de autenticación utilizados en aplicaciones Web, más aún en entornos de Cloud Computing y su utilización en un proyecto particular que requiere Single Sign On. La indagación sobre el estado del arte en cuanto a autenticación en sistemas Web forma parte del proyecto de tesis aprobado de la Maestría en Ingeniería en Sistemas de Información de la FRC-UTN. Al momento de abordar el tema se encontró que la documentación es confusa y el análisis exploratorio considerando bibliografía, documentación de proveedores y consultas a expertos, permitió lograr una solución simple. Se pretende reflejar desde lo más simple a lo más complejo el marco teórico conceptual, detallando características de los distintos tipos de autenticación y los proyectos en los cuales se aplica, por qué es una tendencia y los estándares continúan evolucionando ágilmente de manera colaborativa. Se explica también en el presente trabajo la opción tomada para implementar Single Sign On en un entorno específico y llevado a cabo con éxito de acuerdo a las primer prueba de concepto realizada.*

**Palabras Clave**
Autenticación, Single Sign On, Federación, Identity Provider, Gestión de Identidad, Service Provider, Cloud Computing, Open Source, Gestión de Identidad, Security Assertion Markup Language

## Introducción

Las empresas buscan ser competitivas, en ese contexto es cada vez mayor el número de organizaciones que cambian su infraestructura de sistemas. Están mudando de aplicaciones de escritorio a aplicaciones Web incluso hosteando las mismas en entornos de Cloud Computing o Computación en la Nube. En este pasaje uno de los primeros recaudos que se considera es atender la seguridad de la/s aplicación/es. Adicionalmente en un contexto donde una de las premisas es reducir costos y evitar quedar ligados a proveedores, se apunta a lograr una solución del tipo Open Source.

## Elementos del Trabajo y Metodología

Se realizó una revisión bibliográfica de libros y de trabajos existentes , evaluando cada uno de ellos, sus supuestos y sus conclusiones. Con esta revisión examinamos el estado del arte en el que se encuentra la autenticación del tipo SSO dentro de la industria. Además a esta exploración bibliográfica se le ha incorporado consultas en el mercado a expertos que han llevado a cabo implementaciones del tipo SSO. En todos los casos se estableció como premisa buscar casos y tecnología del tipo abierta (Open Source).

Al comenzar la lectura del material el primer hallazgo fue notar que se confunden los conceptos de autenticación y autorización. Por lo que resultó imperioso pasarlos en limpio ya que para poder profundizar en el concepto de Single Sign On (Sánchez, 2011)[1] o inicio de sesión único.

Autenticación: Consiste en un sistema para certificar que el usuario es quien dice ser.

Autorización: Consiste en dar acceso a una serie de recursos a un usuario o un sistema (para ello el usuario o sistema previamente tendrán que haberse autenticado).

Identidad digital: Nombre de usuario y contraseña.

[1] - También puede traducirse como sistema de autenticación reducida.

En general el mecanismo de autenticación usa un método o una combinación de métodos:
- Usuario y contraseña, es decir, algo que el usuario conoce.
- Certificados digitales o tokens, algo que el usuario tiene.
- Identificación biométrica, basado en los propios rasgos físicos del usuario.

A medida que en la industria del software se ha detectado que había maneras de simular la autenticación simple surgieron otros tipos:

Autenticación Doble Factor: Sistema que combina dos métodos de los mencionados.
Por ejemplo, los bancos entregan tarjetas de coordenadas o certificados digitales, que se usan de forma conjunta con las claves tradicionales o pins para el acceso a la banca electrónica por Internet.
Autenticación Multifactor: En sistemas más sofisticados, que requieren un mayor grado de seguridad, pueden llegar a combinarse los tres métodos. Por ejemplo, un sistema que requiera introducir una contraseña, leer un certificado almacenado en una tarjeta inteligente, y escanear la huella digital.
Autenticación Federada: La autenticación federada es un mecanismo de seguridad descentralizado que no está ligado a ningún proveedor determinado, es decir, que un sistema federado permite el acceso mediante una identidad creada en cualquier proveedor basado en un protocolo determinado.

En respuesta a estos conceptos las tecnologías de autenticación ofrecen soluciones al problema de autenticarse en múltiples sistemas, mientras que las de autorización resuelven el problema de tener datos y recursos personales distribuidos en distintos sistemas. Como consecuencia de esto han surgido y evolucionado los protocolos de autenticación y autorización en la Web, los mismos tienen un importante impacto en la computación en la nube.

Estándares de autenticación y autorización en la nube.
Se ha realizado una exploración de los estándares de mayor uso relacionados con autenticación y autorización en la nube.
Los mismos son:
OpenId, OpenID OAuth Hybrid, Protocol, Facebook Connect y OpenID Connect, OAuth, OpenSocial, Google Friend Connect (Yang Zhamg, Yaaang: 2010)[2].
En la evolución algunos han sido absorbidos por otros, se detallan los más destacados y con mayor cantidad de implementaciones.
OpenID: Es un protocolo estándar abierto de identificación digital (Open, Id) que propone una solución de la categoría abierta (Open Source) y es en consecuencia gratuito. Utiliza autenticación federada. Permite la autenticación de usuario y el control del acceso a una aplicación publicada en Internet o a un sitio Web y a partir de éste punto de ingreso a otros (múltiples servicios) con la misma identidad digital. Este protocolo está orientado sobretodo a los servicios de consumidor ofrecidos por empresas como Google e eBay. Desde el 2011 usando OpenID se puede entrar en Facebook con una cuenta de Google. Fue creado por la fundación OpenID Foundation[3] para apoyar el modelo de código abierto y actuar como apoyo de este framework y su propiedad intelectual. La fundación es una organización sin fines de lucro establecida en los Estados Unidos y que se formó principalmente para proteger el protocolo de autenticación. Fue desarrollado en mayo de 2005 por Brad Fitzpatrick. Es un sistema de identificación digital descentralizado que permite proteger el acceso a aplicaciones

[2] Yang Zhamg, Yaaang - Enero 2010 en su blog del MIT (yz.mit.edu). "Making Sense of OpenID, OAuth, Open Social, Google Friend Connect, Facebook Connect, and more".
[3] http://openid.net/ Fundación OpenID: Accedido junio 2013

Web, o un conjunto reducido de funcionalidades de las mismas. La principal característica es que, como norma general, cualquier aplicación o servicio protegido por esta tecnología está disponible para cualquier usuario que tenga un identificador OpenID, cuyo formato es una URL. El Servicio de Identidad de RedIRIS (SIR)[4] proporciona dicho identificador OpenID a todos los usuarios de las instituciones usuarias del servicio siempre y cuando cumplan con una serie de determinados requerimientos técnicos.

OAuth: Las siglas de este protocolo hacen referencia a Open authorization. A diferencia de OpenID, es un protocolo de autorización, o más precisamente de delegación de acceso; es decir, permite definir cómo un tercero va a acceder a los recursos propios (Sánchez, 2011) [1]. Empezó a definirse en 2006 ante las carencias del protocolo OpenID, y en 2007 se publicó la primera versión oficial. Es un protocolo de identificación abierto que utiliza un estándar Twitter que facilita la autorización para el acceso a microblogs. El sistema funciona de modo tal que en lugar de dar el nombre de usuario y la contraseña, se introduce la cuenta OAuth (oauth.net). Es el protocolo que utiliza Twitter desde el año 2010 para ofrecer un modo rápido de identificarse en algunos servicios y otras redes sociales. También es un estándar abierto para autenticación Permite a los usuarios compartir sus recursos privados (fotos, videos, listas de contactos) almacenados en un sitio con cualquier otro sitio sin tener que introducir a mano su identificación digital (nombre de usuario y contraseña).

OAuth 2.0: Es una versión revisada y simplificada de OAuth, que ya ha sido aprobada y oficialmente ha sido adoptada por grandes compañías como Facebook, Twitter, Yahoo, Google y Microsoft. Respecto de la versión anterior presenta una mayor facilidad de implementación y una arquitectura más robusta que da soporte a mayor número de plataformas.

OpenID y OAuth son protocolos con objetivos distintos pero complementarios. "El protocolo híbrido OpenID Auth combina ambos sistemas, integrándolos en una interfaz única; de este modo, el usuario se autentica en un servidor externo utilizando su proveedor de OpenID y al mismo tiempo autoriza al servidor externo a que acceda a determinados recursos de proveedor. Sin este protocolo híbrido, la utilización de ambos protocolos implicaría que el usuario debería realizar dos acciones: la autenticación primero y la autorización después".[2]

Single Sign On: Sus siglas provienen de inicio de sesión único. También se traduce como sistemas de autenticación reducida.

Consiste en un sistema centralizado de autenticación y autorización. Es un proceso de autenticación de sesión / usuario que permite a un usuario que introduzca un nombre y una contraseña acceder a varias aplicaciones.

Kevin Roebock lo describe como "es una propiedad de control de acceso relacionado múltiple, pero a sistemas de software independientes. Con esta propiedad un usuario inicia sesión una vez y se accede a todos los sistemas sin que se le pida que entre de nuevo en cada uno de ellos. Single sign-off es la propiedad inversa mediante el cual una sola acción de cerrar la sesión finaliza el acceso a múltiples sistemas de software.

Como diferentes aplicaciones y recursos de apoyo a los diferentes mecanismos de autenticación, un registro único tiene que traducir internamente y almacenar credenciales diferentes en comparación con lo que se utiliza para la autenticación inicial." (Roebock,2012) [3]

[4] http://www.rediris.es/sir/howto-openid.html

**Estándares y Fuentes Abiertas Colaborativas:**

Debido a que los estándares de la industria de software crecen día a día en el año 2005 se creó la Cloud Security Alliance (CSA), cuya foco principal son las mejores prácticas para la seguridad en la Nube. Acompañando el evolución del software y del middleware libre o gratuito, han surgido nuevas asociaciones que tienen como objetivo fomentar la creación de estándares abiertos en la nube bajo el concepto de software abierto (open source). La organización de estándares en la nube[5], a través de una normativa y con la partipación muy activa del NIST[6] promueve la documentación y la actividad colaborativa de las organizaciones que se encuentran produciendo nuevos estándares o actualizándolos de acuerdo a las necesidades de integración. El objetivo primordial es unir un conjunto estándares de código abierto, requisitos y servicios. Se basa en dos principios fundamentales: interfases abiertas y formatos abiertos.

**Secure Socket Layer**: Secure Socket Layer (SSL) es un protocolo criptográfico para dar seguridad a la transmisión de datos. Este protocolo permite abrir conexiones seguras a través de una red, cifrando los datos intercambiados entre cliente y servidor mediante un algoritmo de cifrado simétrico. Para poder intercambiarse la clave de sesión utilizada entre cliente y servidor, se utiliza un algoritmo de cifrado de clave pública, típicamente RSA. Como algoritmo de hash se utilizan SHA-256, SHA1, MD5, etc.

Para cada transacción de envío de datos se genera una clave de sesión distinta. La principal área de aplicación de las conexiones SSL es asegurar la comunicación entre el navegador y el servidor web. Pero SSL también se utiliza frecuentemente para asegurar la comunicación entre servidores.

SSL tiene como objetivos:

-Seguridad: mediante el cifrado de datos se garantiza que terceros no podrán tener acceso a la información cuando es enviada a través de la red.

-Integridad de los datos: la información enviada mediante una conexión puede ser validada en los extremos para comprobar que no ha sido alterada durante el camino.

-Autenticidad: nadie puede hacerse pasar por un sitio porque gracias a los algoritmos de encriptación se comprueba que los datos realmente han llegado al servidor que el cliente espera. La autenticidad permite evitar fraudes y ataques.

-Certificados digitales: Las conexiones SSL requieren que el servidor disponga de un certificado digital, el cual consiste en un archivo que identifica de modo único tanto a individuos como a servidores. El servidor puede autenticarse antes de establecer la sesión SSL.

PKI X.509 es un estándar UIT-T para infraestructuras de claves públicas. X.509 especifica, entre otras cosas, formatos estándar para certificados de claves públicas y un algoritmo de validación de la ruta de certificación. A partir de este estándar, se han desarrollado nuevos estándares que utilizan certificados X.509 en conexiones de correo seguro, comercio electrónico, etc.

Los certificados X.509 contienen información relacionada con el usuario, la entidad emisora y el certificado en sí mismo, además de la firma digital del emisor. Los certificados digitales son generalmente firmados por una autoridad de certificación (CA), la cual es fiable y permite garantizar la validez del certificado. Ejemplos: Thawtee, VeriSign.

**Resultados**

De los distintos modelos analizados para atender la necesidad de SSO, la Identidad Federada es el se ha impuesto en la

[5] http://cloud-standards.org Wiki -Estándares Colaborativos de la Nube

[6] http://www.nist.gov National Institute of Standars and Technology.

industria y han sido varias las tecnologías que han surgido a lo largo del tiempo hasta desembocar en el estándar propuesto por OASIS SAML, Security Assertion Markup Language, que con la versión 2.0 parece que se impone claramente sobre el resto de propuestas que seguían este modelo.

**SAML**

Las identidad Federada que propone SAML permite crear un círculo de confianza entre distintas organizaciones para que puedan emplear entre ellas una misma autentificación de usuario de forma nativa, sin necesidad de tener que compartir la misma tecnología de directorio, modelo de seguridad y mecanismos de autentificación o recurrir a pasarelas de conversión que les permita entenderse entre ellos. Permite una gestión de identidad eficiente, puesto que permite la sincronización de datos identificativos entre las entidades federadas.[4]

**Cómo funciona SAML**: Cuando un usuario acude a iniciar sesión en un sitio web que exige autentificación y que opera en un dominio externo al suyo, el servidor web, al no disponer de ninguna información que valide el acceso del usuario, redirecciona su petición a un servidor local de federación de identidad. Este servidor, tampoco dispone de información para verificar las credenciales del usuario por lo que tiene que reenviar la petición de inicio de sesión a un servidor remoto de la federación, capaz de autenticar al usuario. Será este último servidor quien se encargue de recoger el login de usuario, normalmente nombre y contraseña aunque puede ser cualquier otro mecanismo de comprobación de identidad, como los “tokens”, hardware o software. Una vez que el servidor de federación remoto ha verificado la identidad del usuario, normalmente a través de un servidor LDAP de su dominio local, se genera un ticket, una “aserción” en la nomenclatura de SAML, un mensaje XML en la práctica, que el usuario validado integra en su navegador y que presenta al servidor de federación del sitio web al que quiere acceder. Este servidor, extrae el mensaje XML y lo sustituye por una cookie de sesión que el sitio web reconoce como autenticación validada y podrá utilizar para el control de acceso al contenido albergado.

Como se desprende de la descripción, el proceso seguido por las entidades implicadas para la autenticación es trasparente para el usuario, quién sólo ha de hacer un único inicio de sesión, frente al proveedor del servicio, la entidad que le proporciona el recurso al que quiere acceder. Su identidad es verificada por el proveedor de identidad, cliente, que es quien verdaderamente se responsabiliza de toda la gestión de identidad del usuario. Ambos proveedores aunque son organizaciones o, si se prefiere, dominios diferentes, pueden utilizar una firma única y un único login de acceso gracias al mensaje que intercambian en el documento XML que, de forma estandarizada, contiene información sobre el método de autenticación utilizado y los atributos de identidad que puedan ser necesarios para el control de acceso a los recursos y aplicaciones. El intercambio de estos mensajes puede ser iniciado por cualquiera de las dos entidades, utilizando diferentes medios como HTTP, HTTPS, SOA ( Service Oriented Architecture, Arquitectura Orientada a Servicios de cliente). Sea cual sea el iniciador de los mensajes, el proveedor de servicio no dará acceso a un recurso o aplicación hasta que no reciba del proveedor de identidad la “aserción” que permita el acceso o, en su caso, la denegación Cuando un usuario acude a iniciar sesión en un sitio web que exige autentificación y que opera en un dominio externo al suyo, el servidor web, al no disponer de ninguna información que valide el acceso del usuario, redirecciona su petición a un servidor local de federación de identidad. Este servidor, tampoco dispone de información para verificar las

credenciales del usuario por lo que tiene que reenviar la petición de inicio de sesión a un servidor remoto de la federación, capaz de autenticar al usuario. Será este último servidor quien se encargue de recoger el login de usuario, normalmente nombre y contraseña aunque puede ser cualquier otro mecanismo de comprobación de identidad, como los tokens, hardware o software. Una vez que el servidor de federación remoto ha verificado la identidad del usuario, normalmente a través de un servidor Active Directory de su dominio local, se genera una aserción según la nomenclatura de SAML, en realidad en la práctica es un mensaje XML, que el usuario validado integra en su navegador y que presenta al servidor de federación del sitio web al que quiere acceder. Este servidor, extrae el mensaje XML y lo sustituye por una cookie de sesión que el sitio web reconoce como autenticación validada y podrá utilizar para el control de acceso al contenido de la aplicación Web.

SAML describe dos roles de federación: el de proveedor del servicio, que es la entidad que da acceso al usuario a un recurso o aplicación; y el de proveedor de identidad, responsable de la autenticación del usuario.

Ambos, proveedor del servicio y proveedor de identidad, intercambian mensajes para permitir la firma única y un único log de acceso. Tal intercambio de mensajes puede ser iniciado por cualquiera de las dos entidades. El proveedor de identidad se responsabiliza de crear y enviar al proveedor de servicio una aserción, que contiene la identidad del usuario. El proveedor de servicio, por su parte, se hace cargo de validar el aserto SAML antes de permitirle acceder a la aplicación.

Un aserto SAML es un documento XML que contiene diversos elementos relativos a la identidad del usuario, como la forma en que el usuario ha sido autenticado y, opcionalmente, atributos sobre su identidad. El intercambio de tales mensajes puede producirse por medios diferentes, ya sea en forma HTTP o en servicios Web. La figura 1 detalla los principales componentes.

De acuerdo a lo descripto, el proceso seguido por las entidades implicadas para la autenticación es trasparente para el usuario, quién sólo ha de hacer un único inicio de sesión, frente al proveedor del servicio, la entidad que le proporciona el recurso al que quiere acceder. Su identidad es verificada por el proveedor de identidad, cliente, que es quien verdaderamente se responsabiliza de toda la gestión de identidad del usuario. Ambos proveedores aunque son organizaciones o, si se prefiere, dominios diferentes, pueden utilizar una firma única y un único login de acceso gracias al mensaje que intercambian en el documento XML que, de forma estandarizada, contiene información sobre el método de autenticación utilizado y los atributos de identidad que puedan ser necesarios para el control de acceso a los recursos y aplicaciones.

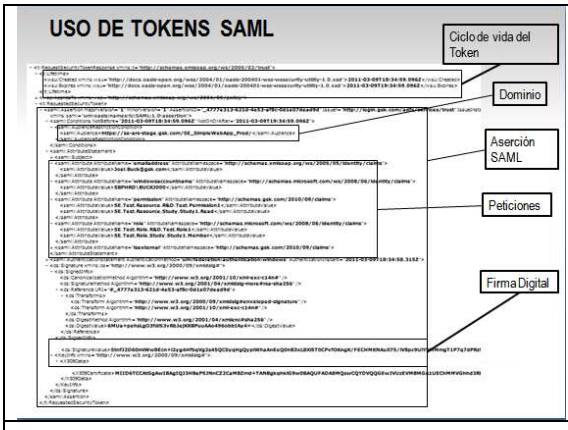


Figura 1

El intercambio de estos mensajes puede ser iniciado por cualquiera de las dos entidades, utilizando diferentes medios convencionales como HTTP en el modo más básico o entornos más complejos, como son los desarrollos SOA, Service Oriented Architecture, Arquitectura Orientada a Servicios de cliente. Es recomedable usar HTTPS a fin de que el canal esté cifrado. Sea cual sea el iniciador de los mensajes, el proveedor de servicio no

dará acceso a un recurso o aplicación hasta que no reciba del proveedor de identidad la "aserción" que permita el acceso o, en su caso, el rechazo.

**Discusión**

Vimos que hoy la aplicación Web objeto de estudio (desarrollada en Java J2EE, publicada mediante servidor de aplicaciones Web Apache Tomcat) autentica a los usuarios a través de un tradicional formulario de acceso o formulario de login donde se ingresa usuario y contraseña. Ambos son validados contra la información almacenada en una tabla de usuarios (base de datos relacional). Una vez validados se crea un sesión única identificada por un id de sesión y un módulo se encarga de administrar la sesión y autorizar el uso de los recursos generales de acuerdo a un archivo de configuración.

En la implementación del aplicativo en la nube para un cliente particular, se planteó la necesidad de modificar el tipo de autenticación usado (autenticación simple) por algún tipo de autenticación multifactor de manera de fortalecer la seguridad. Otra alternativa que se puso en discusión fue la de utilizar procedimientos de inicio de sesión único. Aquí se comparó OpenID y SAML, las ventajas más importantes de SAML se refieren a que cuenta con más detalles que pueden customizarse lo que lo hace más potente y está orientado a uso empresarial, mientras que OpenID es más simple y está orientado a consumidores de servicios. También fue necesario analizar que la empresa cliente ya cuenta con AD FS (Active Directory Federation Services) y en lo posible era aconsejable no innovar.

**Conclusión**

Al adoptar la plataforma que proporciona SAML, el esfuerzo de administración que trae aparejado la gestión de usuarios se simplifica, que ya no es necesario mantener en el sistema propio las cuentas de los usuarios externos. La empresa ya cuenta con AD FS (Active Directory Federation Services) que es el responsable de crear y enviar al proveedor de servicio una aserción (SAML - Security Assertion Markup Language), que contiene la identidad del usuario. Es decir, que el cliente ya cuenta con un proveedor de identidad que puede acepta o rechaza el pedido.

En la aplicación Web bajo estudio, entonces, la propuesta consiste en reemplazar el formulario de login de usuario como funcionaba desde su origen y se incorpora SAML (aserción en formato XML para el intercambio de información entre los proveedores (proveedor de servicio - proveedor de identidad). Ya que la aplicación Web donde se necesita cambiar el modo de autenticación está desarrollada en lenguaje Java, el módulo que implementa SAML se planteó desarrollarlo también usando lenguaje Java.

La figura 2 muestra cómo interactúan los componentes en la prueba de concepto.

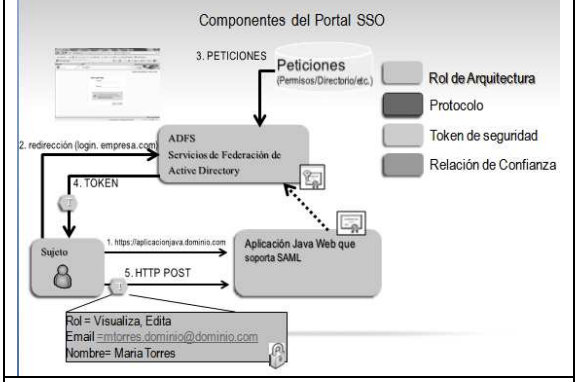


Figura 2

El dominio local del usuario es quien se encarga de la gestión de validación y a través de esta tecnología federativa se normaliza y elimina la comunicación de la validación, entre los federados sin necesidad de recurrir a soluciones complejas y confusas, que suelen ser costosas. Cabe destacar que en la

implementación en producción se utilizará HTTPS a fin de asegurar los canales.

**Referencias.**


[1] Sánchez, Jordi - "Sistemas de autenticación y autorización en internet" - Trabajo de investigación realizado en el Master "Interacción Persona Ordenador" de la Universidad Española de Lleida - Junio 2011
http://jordisan.net/Proyectos/Autent_y_auth-J_Sanchez.pdf (Accedido: Julio 2013)

[2] Aguilar, Luis Joyanes - "Computación en la Nube Estrategias de Cloud Computing en las empresas" - Editorial Alfaomega - 2012 - ISBN: 978-607-707-468-7

[3] Roebuck, Kevin -"Single Sign-on (SSO): High-impact Strategies" - Editorial Emereo Pty Limited - 2011 - ISBN: 9781743044957

[4] "SAML Toolkit for Java" - https://onelogin.zendesk.com/entries/512080-saml-toolkit-for-java - Accedido: Julio 2013
http://www.rediris.es/sir/howto-openid.html

[5] "Authentication and Authorization"
http://httpd.apache.org/docs/2.2/howto/auth.html
Accedido: Julio 2013

[6] Moskowitz , Robert - "Authentication Types" Draft version 1.2 - 2003 -
http://www.ieee802.org/1/files/.../Authentication%20Types%20v%201-2.doc
Accedido: Febrero 2013

[7] "Authentication and Authorization in the Google Data Protocol"
https://developers.google.com/gdata/docs/auth/overview#OAuth2
Accedido: Febrero 2013

[8] "A Survey of Middleware", 18th International Conference on Computers and Their Applications, March 26-28, 2003, Honolulu, Hawaii.
http://triton.towson.edu/~karne/research/middlew/surveym.pdf
Autores: Toni A. Bishop y Ramesh K. Karne
Accedido: Julio 2012

[9] Richardson, Leonard - "RESTful Web Services" - Editor: O' Reilly Media - 2007

[10] Kalin, Martin - "Java Web Services: Up and Running" - Editor: O'Reilly & Associates Inc. ISBN: 978-0-596-52112-7


**Datos de Contacto:**


*Ing. María Soledad Romero*
*Universidad Tecnológica Nacional – Facultad Regional Córdoba*
*romeroma.soledad@gmail.com*
*Ing. Mariano García Mattío*
*Instituto Universitario Aeronáutico*
*magm3333@gmail.com*