DESCRIPCIÓN Y GUÍAS DE USO DE LA

Plataforma de Gobierno Electrónico del Estado Uruguayo

PLATAFORMA DE GOBIERNO ELECTRÓNICO

Versión 1.1 – 2011

Tabla de contenidos

Capítulo I

Introducción

Introducción

Como lo indica su misión, la Agencia para el Desarrollo del Gobierno de Gestión Electrónica y la Sociedad de la Información y del Conocimiento (AGESIC) debe "Liderar la estrategia de Gobierno Electrónico del país como base de un Estado eficiente y centrado en el ciudadano". Como parte de las iniciativas tendientes a lograrlo se propuso poner a disposición del Estado una Plataforma de Gobierno Electrónico y asumir un rol de articulador entre dependencias estatales facilitando su coordinación y comunicación para avanzar desde el concepto de Gobierno Electrónico hacia el de Gobierno en Red.

La Plataforma de Gobierno Electrónico (PGE) que ha implementado AGESIC, permite y facilita la integración de los servicios ofrecidos por los organismos, proporcionando el contexto tecnológico y legal que la regula.

La PGE provee a los organismos del Estado:

- Una infraestructura que facilita la implementación de servicios al público así como el acceso a servicios de otros organismos del Estado. A esta infraestructura, conjuntamente con el marco legal y técnico, se le llama **Plataforma de Interoperabilidad**.
- **Servicios Transversales** de valor agregado como lo son hasta el momento, un portal integrador y punto de entrada de trámites y servicios, un sistema de expediente electrónico, un portal de información geográfica y un buscador de información de interés público centrado en contenido gubernamental.

A los efectos de desarrollar el uso de la PGE, AGESIC brinda asesoramiento y apoyo a las diferentes instituciones desde la Dirección Gerencia de Proyectos. En este sentido, se ha elaborado este documento que describe la PGE y presenta una primera versión de la Plataforma de Interoperabilidad.

Organización del Documento

El resto del documento está organizado en 4 capítulos, orientados a distintos perfiles de usuario de la PGE.

A continuación se resume el contenido de cada capítulo con el fin de guiar la lectura del documento.

Capítulo II – Guía Gerencial de la PGE

Este capítulo presenta el concepto de gobierno electrónico, los desafíos de interoperabilidad que éste plantea y cómo las plataformas de gobierno electrónico pueden ayudar a resolverlos. Además, se presenta una visión gerencial de la Plataforma de Gobierno Electrónico del Estado Uruguayo, describiendo sus principales prestaciones y cómo los organismos pueden aprovecharlas. Por último, se describe de qué forma los organismos pueden avanzar y evolucionar al hacer uso de la PGE.

Capítulo III – Descripción Técnica de la PGE

Este capítulo brinda una descripción técnica de la PGE, presentando sus principales componentes. En particular, se profundiza en dos de los componentes de la Plataforma de Interoperabilidad: el Sistema de Seguridad y la Plataforma de Middleware. Para cada uno de ellos, se describen las prestaciones más importantes que brindan y los mecanismos, productos y estándares utilizados para hacerlo.

Capítulo IV – Guía de Programación Java de la PGE

Este capítulo brinda guías de desarrollo para la PGE, utilizando Java. En particular, se describe cómo consumir un servicio publicado en la plataforma.

Capítulo V – Guía para el Alta y Consumo de Servicios

Este capítulo describe, a nivel técnico, los requerimientos y pasos necesarios para que un organismo provea y consuma servicios en la PGE.

Apéndice 1 – Marco Técnico

Este apéndice brinda el marco técnico base para la comprensión de este documento. Concretamente se abordan temas de Seguridad, Web Services y Arquitecturas Orientadas a Servicios.

Apéndice 2 – Formularios para el Alta y Consumo de Servicios

Este apéndice incluye los formularios que deben completar y enviar los organismos para el alta y consumo de servicios de la PGE.

Apéndice 3 – Ejemplos Técnicos

Se incluyen tokens y mensajes SOAP para la comunicación con la PGE.

Capítulo II

Guía Gerencial de la Plataforma de Gobierno Electrónico

Introducción

Este capítulo presenta el concepto de gobierno electrónico, los desafíos de interoperabilidad que éste plantea y cómo las plataformas de gobierno electrónico pueden ayudar a resolverlos. Además, se presenta una visión gerencial de la Plataforma de Gobierno Electrónico del Estado Uruguayo, describiendo sus principales prestaciones y cómo los organismos pueden aprovecharlas. Por último, se describe de qué forma los organismos pueden avanzar y evolucionar al hacer uso de la PGE.

Gobierno Electrónico

El gobierno electrónico suele definirse de muchas formas y cada país le da un nombre diferente a su iniciativa de modernización de la administración pública, a través del uso generalizado de las Tecnologías de la Información y la Comunicación (TIC).

En [1] se define Gobierno Electrónico de la siguiente manera:

"Es el uso de las TICs para reinventar los procedimientos del gobierno, para promover la difusión y disponibilidad de la información y el conocimiento sobre los servicios gubernamentales, y para dotar de oportunidades para interacciones en línea, eliminando entidades intermediarias y generando un poder de cambio en las relaciones entre el gobierno (entidades y agencias estatales) y gobernados (ciudadanos=consumidores, que incluyen organizaciones privadas). Entendido así, el gobierno electrónico es una fuerza de desarrollo y una herramienta para la definición del gobierno".

Lograr una interacción confiable y eficiente entre organismos del estado constituye uno de los objetivos y desafíos principales del gobierno electrónico. Esta interacción resulta evidente para cumplir con los requerimientos de los organismos, pero también es fundamental para atender a los requerimientos de los ciudadanos cuando deben realizar operaciones que involucran múltiples organismos.

Para responder a los objetivos anteriores, los sistemas de gobierno electrónico implementan los mecanismos de interoperabilidad, que permiten que los sistemas informáticos de las instituciones se comuniquen entre ellos para intercambiar datos así como para ordenar la ejecución de funciones.

Desafíos de Interoperabilidad

Los sistemas informáticos de los organismos y agencias de un gobierno son producto de un desarrollo heterogéneo debido a múltiples motivos [2]. En el momento que dos o más organismos necesitan intercambiar información se deben realizar acuerdos entre los mismos para establecer cuáles son los datos que deben fluir de uno a otro y la forma de representarlos e interpretarlos. Se deben definir también esquemas de seguridad para garantizar que en el intercambio no se pierda ni se altere información, y que sólo las personas autorizadas de ambos organismos tengan acceso a la misma.

Si un número N de organismos requieren intercambiar información entre ellos, se deben crear N*(N-1)/2 acuerdos que regulen las relaciones entre ellos. Por cada relación se establece un conjunto de definiciones semánticas, técnicas, operativas y de gobernanza que generalmente reflejan un "acuerdo" que se hace o se firma entre las partes. Esto implica para una entidad cualquiera un número de N-1 acuerdos, con la consecuente dificultad de administración y los desarrollos técnicos y operativos para responder a los mismos.

La Figura 1 presenta como ejemplo la relación entre cinco organismos. Cuando es necesario que todos ellos intercambien información, se establecerán diez relaciones entre organismos, y cada organismo deberá establecer cuatro acuerdos diferentes y posiblemente desarrollar soluciones independientes para cada uno de éstos.

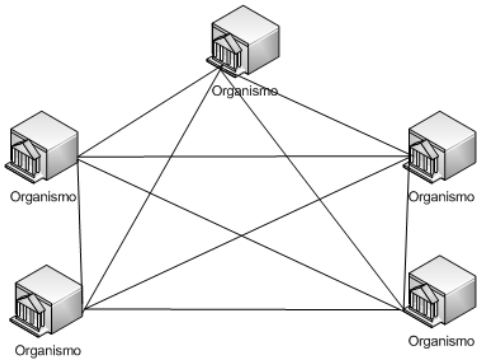

Figura 1 – Relaciones entre Organismos

El desafío de la interoperabilidad entre los N organismos de un gobierno que intercambian información, consiste en encontrar mecanismos que reduzcan drásticamente la complejidad y los costos de definición, implementación y administración de los N-1 acuerdos bipartitos existentes, reemplazándolo por un sólo acuerdo estándar de intercambio. La Figura 2 presenta una solución en donde existe un arco para cada organismo ya que éste debe desarrollar un sólo esquema de intercambio

de información que le servirá para relacionarse con cualquier otra agencia gubernamental.

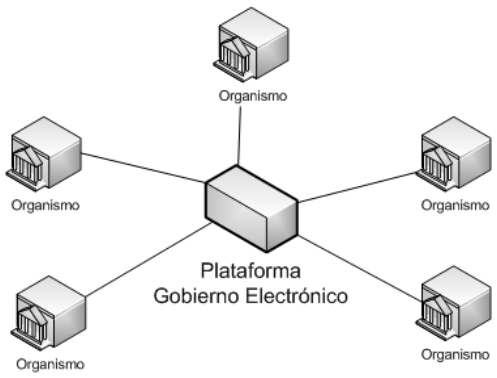

Figura 2 – Relación simplificada por la existencia de un esquema unificado

El esquema unificado señalado responde a un conjunto de estándares de representación de la información, procesos y mecanismos comunes de intercambio de la misma, diseños de seguridad implementados por todas las partes, entre otros factores necesarios para lograr el flujo de información de manera óptima y segura en la administración pública.

En este marco, se facilita que los organismos interactuen en base a una Arquitectura Orientada a Servicios (Service Oriented Architecture, SOA), caracterizada por modelar las aplicaciones como *servicios*, los cuales se utilizan a través de interfaces y protocolos estandarizados. El encare basado en SOA facilita la integración entre organismos, promoviendo la reutilización y el aprovechamiento de los recursos de información y tecnológicos con los que cuentan. Además, contribuye a poder responder de forma ágil ante cambios en requerimientos o regulaciones.

Plataformas de Gobierno Electrónico

La implementación del Gobierno Electrónico se basa en una combinación de tecnologías y especificaciones basadas en modelos que varían según los países.

Las “Plataformas de Gobierno Electrónico” (PGEs) constituyen uno de los tipos de herramienta clave para desarrollar el Gobierno Electrónico, y consisten en infraestructura y servicios que facilitan la conectividad de los sistemas del conjunto de organismos del Estado, ofrecen servicios comunes generando economía de escala y viabilizando el uso de tecnologías avanzadas, y promueven la implementación de servicios multi-organismo.

A continuación se presentan algunas de las características que ofrecen:

- Facilitar la interoperabilidad entre sistemas que implementan servicios públicos y funciones de gobierno en general.
- Aportar servicios comunes (aplicación de leyes y normas, intercambio de información, seguridad, etc).
- Aportar infraestructura común (comunicaciones, sistemas de base) y los servicios de administración.
- Promover, a través de la práctica, la aplicación de buenas prácticas en Gobierno Electrónico en el Estado.
- Viabilizar (facilitar) el desarrollo de servicios de gobierno electrónico (servicios públicos y de gobierno)
- Encaminar la aplicación de buenas prácticas tecnológicas y de informatización del gobierno en las organizaciones.
- Facilitar la integración de los organismos del estado a pesar de diferencias en su desarrollo tecnológico.

Las PGEs consisten en una infraestructura de hardware, software y comunicaciones que permiten integrar sistemas en un marco de Gobierno Electrónico, aportando también funcionalidades y servicios adicionales (por ejemplo seguridad, monitoreo, etc.). Resultan altamente apropiadas para implementar funcionalidades inter-organismo, en especial aquellas que presentan potencial de mayor interacción con otros organismos en desarrollos posteriores.

El uso e integración a una Plataforma de Gobierno Electrónico aporta una serie de ventajas a los organismos y por consiguiente al Estado en su conjunto.

- **Racionalización: evitar costos y complejidades innecesarias** La integración de las instituciones a una PGE evita costos y complejidades asociados a implementar individualmente los mecanismos de interoperabilidad con otras instituciones (aún en un número reducido), realizándose un aporte significativo a la racionalización de recursos tecnológicos en el Estado a través de una solución que genera economía de escala.
- **Promoción y evolución del Gobierno Electrónico** La vinculación entre las instituciones resulta un aspecto clave en la estrategia de Gobierno Electrónico, para ello la PGE constituye un instrumento fundamental facilitando el acceso a servicios e información en forma integrada.
- **Cumplimiento de normas relativas al Gobierno Electrónico** La interacción a través de la PGE, asegura el cumplimiento de las normas relativas al Gobierno Electrónico que se van promulgando, así como de las Normas Técnicas consistentes en

el conjunto de estándares y mejores prácticas que deben seguirse durante el ciclo de vida de la plataforma.

- **Agilidad para ofrecer nuevos servicios a los ciudadanos** Dado que la PGE facilita el diseño y desarrollo de servicios que interoperan entre los distintos organismos incorporados a la PGE, el tiempo requerido para hacer disponibles nuevos servicios a los ciudadanos puede reducirse significativamente.

PGE del Estado Uruguayo

La Plataforma de Gobierno Electrónico (PGE) de AGESIC tiene como objetivo general facilitar y promover la implementación de servicios de Gobierno Electrónico en Uruguay. Para esto, la PGE brinda mecanismos que apuntan a simplificar la integración entre los organismos del Estado y a posibilitar un mejor aprovechamiento de sus activos.

La PGE provee infraestructura (*hardware* y *software*) y servicios utilitarios, que reducen la complejidad de implementar servicios al público y/o accesibles dentro del Estado. Asimismo, la PGE aporta los mecanismos técnicos idóneos para implementar servicios compuestos, basados en los ofrecidos por diferentes organismos, normalizando e integrando la información proveniente de éstos.

La PGE apunta a desarrollar el Gobierno Electrónico, proveyendo una plataforma de integración, junto con un conjunto de utilitarios, que facilitan la implementación y uso de servicios de Gobierno Electrónico. La PGE también promueve el desarrollo técnico de los organismos en forma coordinada, para que los avances institucionales contribuyan al desarrollo del Gobierno Electrónico a nivel nacional.

A nivel tecnológico, la PGE posibilita que los organismos provean sus funcionalidades de negocio a través de servicios de *software* de forma independiente a la plataforma en la que fueron implementados. Esto corresponde a la implementación de una SOA a nivel del Estado, en la cual los servicios ofrecidos por los organismos son descriptos, publicados y descubiertos, invocados y combinados a través de interfaces y protocolos estandarizados.

De esta forma, al facilitarse la reutilización de servicios, se promueve la construcción de nuevos servicios en base a otros ya existentes, reduciéndose los tiempos de implementación de nuevos requerimientos. Por otro lado, el “acoplamiento débil” entre servicios promovido por la SOA, permite la evolución autónoma de los servicios de software implementados en los organismos.

La Figura 3 presenta la idea general de la PGE, donde se puede observar cómo varios organismos ofrecen, buscan y utilizan servicios en la PGE.

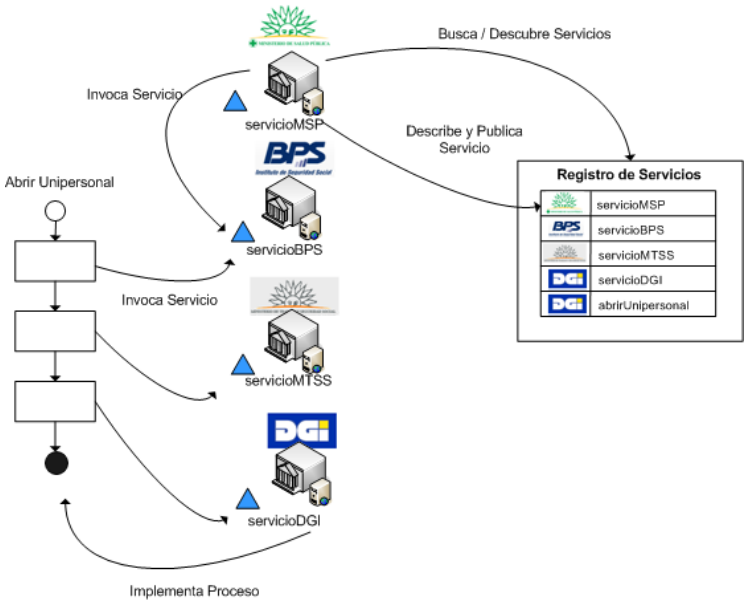

Figura 3 – Visión General de la PGE

La conectividad base entre los organismos es provista por una red de alta velocidad denominada REDuy [3]. Por otro lado, el soporte tecnológico a la PGE está dado por un conjunto de componentes que pueden dividirse en dos grandes grupos:

- Componentes de la Plataforma de Interoperabilidad
- Servicios Transversales

Componentes de la Plataforma de Interoperabilidad

Estos componentes facilitan la provisión, búsqueda e invocación de los servicios informatizados que son brindados por los organismos, así como la interoperabilidad e interacción segura entre los mismos.

La **Plataforma de Middleware** cuenta con mecanismos para facilitar el desarrollo, despliegue e integración de servicios y aplicaciones. Los organismos pueden utilizar esta plataforma para publicar y descubrir servicios, contar con un ambiente de ejecución para servicios o aplicaciones que requieran infraestructura de hardware o software no disponible en los organismos, así como utilizar las diferentes capacidades de mediación, las cuales permiten desacoplar clientes y servicios.

Por otro lado, el **Sistema de Seguridad** constituye un componente esencial de la PGE, dado que provee servicios de seguridad al resto de los componentes. Los organismos pueden utilizar este sistema para controlar

el acceso a los servicios que proveen. Además, el Sistema de Seguridad cuenta con componentes que pueden realizar auditorías de seguridad y facilitar el acceso seguro de los organismos a la PGE.

Por último, el **Sistema de Gestión de Metadatos** provee una especificación de alto nivel de los conceptos relativos a servicios públicos, de forma de evitar, o eventualmente resolver, ambigüedades en el manejo de estos conceptos por parte de los organismos. Los organismos pueden obtener beneficio de este sistema utilizando los conceptos estandarizados y normalizados que provee, lo que asegurará que el intercambio de información con otros organismos se realice sin ambigüedad, no solo con las aplicaciones actuales, sino también con las futuras que se apoyen en estos conceptos.

Servicios Transversales

El **Portal del Estado Uruguayo** es uno de los principales puntos de entrada al Gobierno Electrónico, permitiendo la interacción de los ciudadanos con contenidos, servicios y trámites de interés público. Entre sus principales características se encuentran el soporte a estándares de la industria, su capacidad dinámica de personalización y su cumplimiento con pautas de accesibilidad de sitios Web.

En la Figura 4 se muestra cómo las instituciones podrán integrarse y beneficiarse del Portal exponiendo contenido a través del mismo, desarrollando interfaces para servicios existentes, integrando trámites y enlaces a los mismos, utilizando servicios de estandarización de acceso a la información y, en caso de no contar con un portal, podrán solicitar la creación de uno.

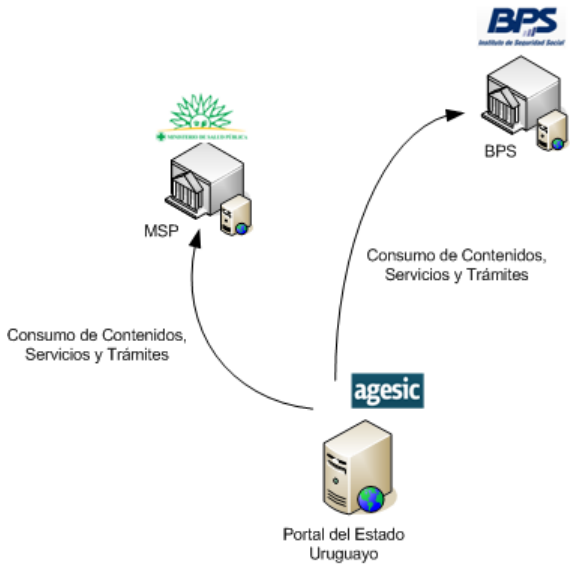

Figura 4 – Portal del Estado Uruguayo

El **Buscador del Estado Uruguayo** tiene como objetivo instrumentar una herramienta de búsqueda orientada a las necesidades de quien realiza búsquedas sobre el Estado Uruguayo. La principal ventaja del buscador, con respecto a otros como Google, es que está específicamente optimizado para dicho fin.

Tal como se aprecia en la Figura 5, el Buscador inicialmente accede e indexa información existente en páginas Web públicas. Posteriormente, los resultados de las búsquedas conducen a los usuarios a los portales estatales o sitios que contienen dicha información.

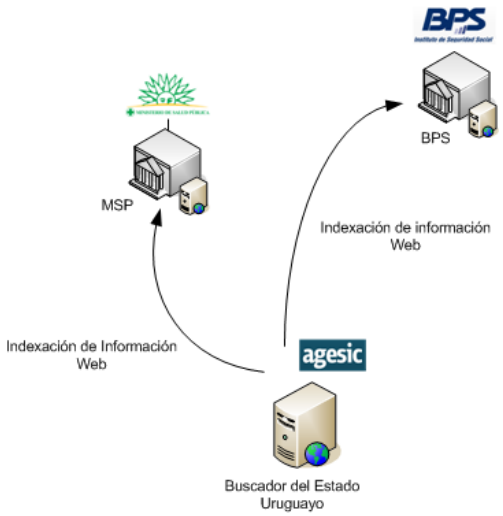

Figura 5 – Buscador del Estado Uruguayo

Las instituciones podrán colaborar con el Buscador a través de la mejora de sus sitios Web de forma de facilitar el acceso y descubrimiento de la información por parte de los usuarios. Otra estrategia de integración es a través de la creación de herramientas o aplicaciones de valor agregado que utilicen los servicios provistos por el buscador. Además, es posible la construcción de buscadores especializados en una temática particular.

El **Sistema de Expediente Electrónico** tiene como objetivo principal informatizar el manejo de Expedientes a nivel del Estado Uruguayo y facilitar la interoperabilidad de los mismos a través de los diferentes organismos. El principal componente del sistema es una aplicación de gestión de expedientes electrónicos, que puede ser utilizada bajo la modalidad de *software* como servicio (Software as a Service, SaaS) o puede ser instalada localmente en los organismos del Estado. Una aplicación Web ofrecerá a ciudadanos y organismos la posibilidad de consultar a través de Internet la trazabilidad de los expedientes en que está involucrado. Finalmente, un módulo de ruteo y trazabilidad permitirá el intercambio y trazabilidad de todas las actuaciones realizadas sobre todos los expedientes.

La Figura 6 presenta diferentes ejemplos de interacción de los organismos con el Sistema de Expediente Electrónico. En la misma es posible apreciar que uno de los organismos utiliza dicha aplicación en la modalidad de SaaS y otra en cambio la instala local en su infraestructura propia. Los organismos que ya han adquirido una solución de expediente electrónico podrán realizar el ruteo de expedientes a través del Estado, integrándola al módulo de Ruteo y Trazabilidad de Expedientes.

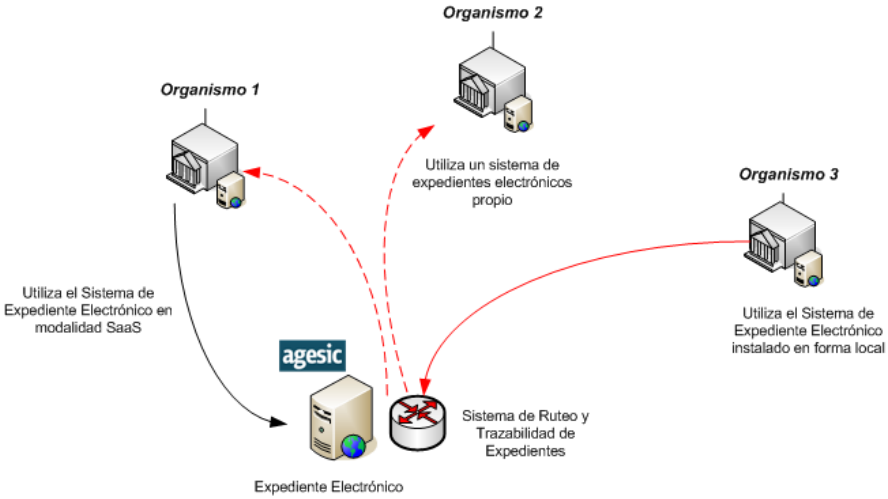

Figura 6 – Sistema de Expediente Electrónico

Por último, el **Geoportal** es un portal de información geográfica que permite la consulta y análisis vía Web de la información geográfica proveniente de los organismos. El Geoportal, se encuentra enmarcado en el proyecto IDE (Infraestructura de Datos Espaciales), el cual tiene como objetivo principal crear un servicio en red para acceder y compartir datos geográficos entre los Organismos del Estado.

Como se aprecia en la Figura 7, el Geoportal opera instalado en un servidor central administrado por AGESIC. Las capas de información geográfica que presenta se obtienen del propio servidor y de varios servidores periféricos que pertenecen a instituciones gubernamentales que forman parte de la Infraestructura de Datos Espaciales.

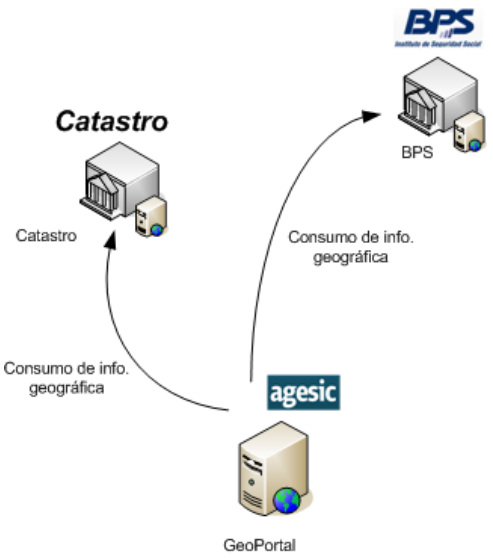

Figura 7 – GeoPortal

Resumen de Prestaciones de la PGE

Los distintos componentes de la PGE permiten que los organismos cuenten con las siguientes prestaciones:

- **Conectividad**
  La REDuy brinda la infraestructura de conectividad base para que los organismos se conecten entre sí, y con la PGE. Por otro lado, el Portal permite la interacción de los ciudadanos con contenidos, servicios y trámites de interés público. A su vez, la Plataforma de Middleware, el Sistema de Gestión de Metadatos y el Sistema de Expediente Electrónico posibilitan la integración e interoperabilidad entre los organismos.
- **Confiabilidad**
  La confiabilidad está dada principalmente por los mecanismos que brinda el Sistema de Seguridad, en particular, el Sistema de Control de Acceso y el de Auditoría.
- **Disponibilidad**
  La infraestructura de hardware y software disponible en la PGE posibilitan la implementación de mecanismos de Redundancia y Contingencia, que permiten garantizar un mayor nivel de disponibilidad.

Avanzando junto a la PGE

Los organismos del Estado pueden evolucionar mediante el uso de la Plataforma de Gobierno Electrónico y a través del empleo de las pautas y estándares que ha promovido AGESIC.

Uso de la Plataforma de Gobierno Electrónico

La PGE puede ser utilizada por los organismos de diferentes formas. En primer lugar, los organismos pueden participar consumiendo servicios de otras instituciones, publicando servicios propios (en su infraestructura propia o de la PGE), o participando en servicios multi-institucionales. Además, las instituciones pueden utilizar diferentes opciones tecnológicas basadas en las pautas de la AGESIC.

El uso de la PGE implica que las instituciones cumplan las siguientes etapas:

1. Informarse sobre las pautas básicas de la AGESIC y comprender el alcance de sus servicios.

2. Contactar a la AGESIC para definir la forma de participación en la PGE, acordando una "hoja de ruta". Generar acuerdos y proyectos coordinados (fondos concursables, planes directores informáticos, etc).

3. Realizar las actividades de formación y pruebas correspondientes.

4. Definir, junto a la AGESIC, un plan de uso de la PGE en producción, ya sea consumiendo o publicando servicios.

5. Realizar el seguimiento a las pautas de la AGESIC, en especial las relativas al uso de la PGE.

Mejora de la calidad de los servicios

El uso de la PGE y la aplicación de las pautas de la AGESIC apoyan a las instituciones en la mejora en la calidad de sus servicios, especialmente los basados en las TIC, y por lo tanto la mejora permanente de las mismas.

Marco de Interoperabilidad

La AGESIC plantea estándares, recomendaciones y procedimientos para que las instituciones utilicen la PGE, y avancen en la informatización de sus servicios en forma compatible e interoperable con otras organizaciones públicas y privadas.

Marco Legal

El Estado Uruguayo ha fomentado y creado un marco jurídico y normativo necesario para impulsar el desarrollo del gobierno en red. Este marco incluye, por ejemplo, el Decreto 450/09 [4] que establece los principios y líneas estratégicas del Gobierno Electrónico y del Gobierno en Red, la Ley N° 18.331 [5] de Protección de Datos Personales y Acción de Habeas Data, la Ley N° 18.381 [6] sobre Acceso a la Información Pública, la Ley N° 18.600 [7] sobre Documento Electrónico y Firma Electrónica y las Leyes N° 9.739 [8] y N° 17.616 [9] sobre Derechos de Autor.

Referencias

[1] RUBINO- HALLMAN, Silvana. E-government in Latin America and The Caribbean. Reinventing governance in the Information Age. P.12.

[2] CEPAL. Libro blanco de interoperabilidad de gobierno electrónico para América Latina y el Caribe. Versión 3.0. 2007. http://www.cepal.org/socinfo/noticias/noticias/2/32222/Libro_blanco_de_interoperabilidad.pdf

[3] AGESIC – REDuy. http://www.agesic.gub.uy/innovaportal/v/759/1/agesic/REDuy.html [Accedida en Abril de 2010]

[4] Decreto 450/09 Principios y líneas estratégicas para el gobierno en red. http://www.presidencia.gub.uy/_web/decretos/2009/09/cm827.pdf

[5] Ley de Protección de Datos Personales y Acción de "Habeas Data". http://www.parlamento.gub.uy/leyes/AccesoTextoLey.asp?Ley=18331

[6] Ley de Derecho de Acceso a la Información Pública. http://www.parlamento.gub.uy/leyes/AccesoTextoLey.asp?Ley=18381

[7] Ley de Documento Electrónico y Firma Electrónica. http://www.parlamento.gub.uy/leyes/AccesoTextoLey.asp?Ley=18600

[8] Ley de Propiedad Literaria y Artística. http://www.parlamento.gub.uy/leyes/AccesoTextoLey.asp?Ley=09739

[9] Ley de Derechos de Autor y Derechos Conexos. http://www.parlamento.gub.uy/leyes/AccesoTextoLey.asp?Ley=17616

[10] Applied SOA: Service-Oriented Architecture and Design Strategies. Rosen, Lublinsky, Smith and Balcer. Wiley Publishing, Inc. 2008.

[11] Enterprise Service Oriented Architectures: Concepts, Challenges, Recommendations. McGovern, Sims, Jain and Little. Springer. 2006.

Capítulo III

Descripción Técnica de la Plataforma de Gobierno Electrónico

Introducción

Este capítulo brinda una descripción técnica de la PGE, presentando sus principales componentes. En particular, se profundiza en dos de los componentes de la Plataforma de Interoperabilidad: el Sistema de Seguridad y la Plataforma de Middleware. Para cada uno de ellos, se describen las prestaciones más importantes que brindan y los mecanismos, productos y estándares utilizados para hacerlo.

Descripción General de la PGE

La Plataforma de Gobierno Electrónico (PGE) del Estado Uruguayo tiene como objetivo general facilitar y promover la implementación de servicios de Gobierno Electrónico en Uruguay. Para esto, la PGE brinda mecanismos que apuntan a simplificar la integración entre los organismos del Estado y a posibilitar un mejor aprovechamiento de sus activos.

A nivel tecnológico, se implementó una Arquitectura Orientada a Servicios (Service Oriented Architecture, SOA) a nivel del Estado, la cual se apoya fuertemente en la tecnología de Web Services. De esta forma, los organismos proveen sus funcionalidades de negocio a través de servicios de Software que son descriptos, publicados, descubiertos, invocados y combinados de forma independiente a la plataforma tecnológica en la que fueron implementados. Esto facilita la integración entre los organismos, promoviendo la reutilización y el aprovechamiento de los recursos de información y tecnológicos con los que cuentan. Además, contribuye a poder responder de forma ágil ante cambios en requerimientos o regulaciones.

El soporte tecnológico a la PGE está dado por un conjunto de componentes que dan soporte a la Plataforma de Interoperabilidad y proveen Servicios Transversales. Estos componentes brindan mecanismos para implementar la SOA, garantizar la interacción segura entre los servicios y aplicaciones, e interactuar con los ciudadanos. La Figura 8 presenta una visión general de los componentes de la PGE y de los distintos actores involucrados.

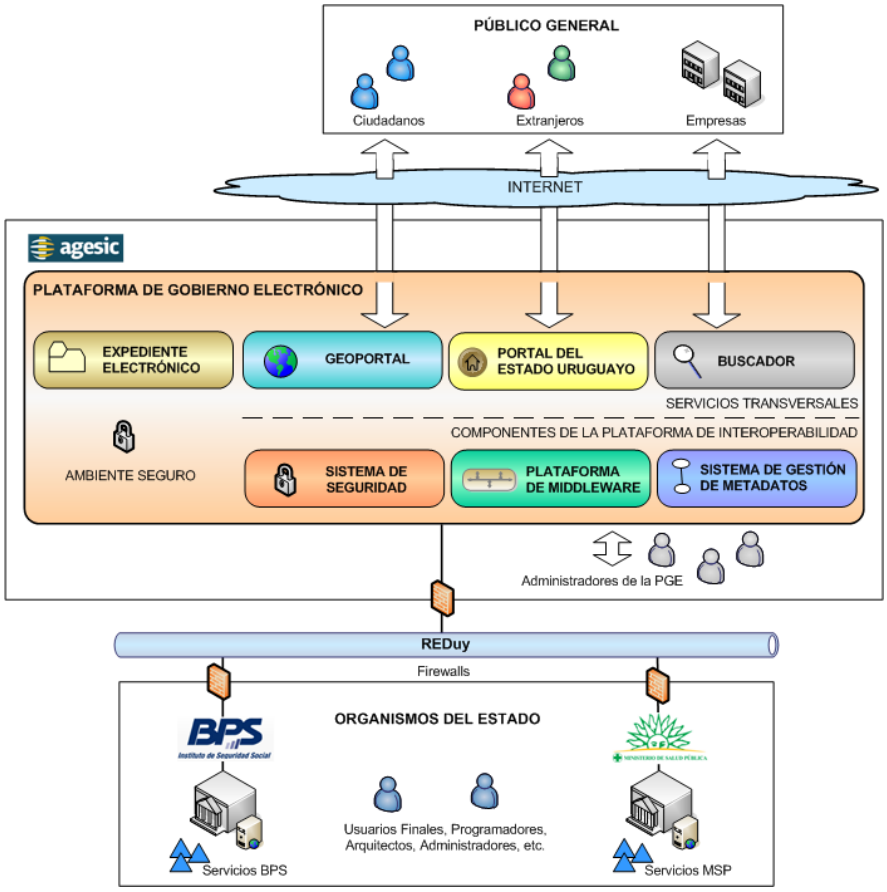

Figura 8 – Principales Componentes y Actores de la PGE

Los componentes de la Plataforma de Interoperabilidad son el Sistema de Seguridad, el Sistema de Gestión de Metadatos y la Plataforma de Middleware. En particular, el Sistema de Seguridad provee los mecanismos para que los componentes y servicios de la PGE se ejecuten en un ambiente seguro. Las Servicios Transversales proveen funcionalidades específicas y actualmente consisten en el Portal y el Buscador del Estado Uruguayo, el Sistema de Expediente Electrónico y el Geoportal.

Por otro lado, los principales actores de la PGE son el Público en General, los Organismos del Estado y el personal de AGESIC. El Público en General, que incluye ciudadanos, extranjeros y empresas, accede a los servicios de la PGE a través de Internet utilizando, por ejemplo, el Portal del Estado Uruguayo. Los usuarios y Sistemas de Software en los Organismos del Estado, se apoyan en la infraestructura de conectividad provista por la REDuy para acceder a los servicios y componentes de la PGE. También utilizan esta infraestructura de conectividad para proveer, a través de la PGE, sus servicios al resto de los organismos. Por último, el personal de AGESIC se encarga de la administración de la plataforma.

Infraestructura de Conectividad: REDuy

La REDuy [1] es una red de alta velocidad que provee la infraestructura de conectividad necesaria para que los organismos se interconecten, entre ellos y con la PGE, de manera segura y con adecuados niveles de servicio y seguridad informática. La REDuy está implementada sobre la red MPLS (MultiProtocol Label Switching) de AntelData y cuenta con velocidades de acceso mínimas de 10Mbps y máximas de 100Mbps. Cuenta además con un centro de soporte gestionado por AGESIC y es considerada un activo de información crítico del Estado, por lo que tiene especial atención del CERTuy [2].

La REDuy es una red metropolitana y actualmente conecta a varios organismos de Montevideo. Además, se planea extender su alcance para llegar también a los organismos del interior del país.

Como se presenta en la Figura 9, la conexión de los organismos a la REDuy está protegida por *firewalls* que controlan el tráfico de red de los organismos, desde y hacia la REDuy. La configuración y administración de estos *firewalls* está a cargo del equipo de soporte de AGESIC.

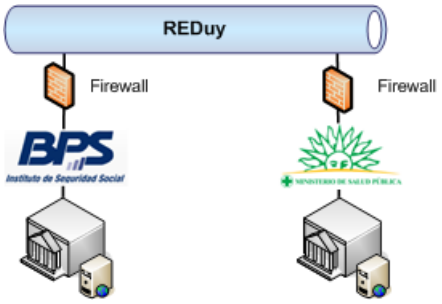

Figura 9 - *Firewalls* REDuy

El tráfico desde los organismos hacia la REDuy está permitido libremente. Sin embargo, el tráfico hacia los organismos desde la REDuy, debe ser habilitado en los *firewalls* por el equipo de soporte de AGESIC.

Servicios provistos por Organismos

Los Organismos del Estado proveen sus funcionalidades de negocio a través de la tecnología de Web Services. Generalmente, los sistemas informáticos correspondientes, accesibles a través de Web Services, se alojan en los servidores de los propios organismos, aunque también es posible alojarlos en la PGE, en caso que se tengan requerimientos especiales que no se puedan satisfacer en los organismos.

Los Web Services se pueden implementar utilizando distintas tecnologías como Java EE, la plataforma .NET y PHP, entre otras. Para mejorar el nivel de interoperabilidad, se requiere que las implementaciones de los Web Services se ajusten a los perfiles Basic Profile [3] y Basic Security Profile [4] definidos por la organización Web Services Interoperability (WS-I).

Cabe recalcar que los Web Services pueden exponer las funcionalidades de negocio de Sistemas Legados alojados en los organismos, aprovechando de esta forma los recursos de información y tecnológicos existentes.

Componentes de la Plataforma de Interoperabilidad

Los componentes de la Plataforma de Interoperabilidad de la PGE [5] son la Plataforma de Middleware, el Sistema de Seguridad y el Sistema de Gestión de Metadatos.

La **Plataforma de Middleware** provee mecanismos que facilitan el desarrollo, despliegue e integración de servicios y aplicaciones. Además, cuenta con los componentes necesarios para la implementación de la SOA a nivel del Estado. En la sección “Plataforma de Middleware” se brinda una descripción más detallada de la misma, junto con los productos y tecnologías que la implementan.

El **Sistema de Seguridad** constituye un componente esencial de la PGE, dado que provee servicios de seguridad al resto de los componentes. Este sistema brinda mecanismos que permiten realizar auditorías de seguridad en la PGE, aplicar políticas de acceso asociadas a los servicios publicados en la plataforma, y facilitar el acceso seguro de los organismos a la PGE. En la sección “Sistema de Seguridad” se brinda una descripción más detallada del mismo, junto con los productos y tecnologías que lo implementan.

Por último, el **Sistema de Gestión de Metadatos** provee una especificación de alto nivel de los conceptos relativos a servicios públicos, de forma de evitar, o eventualmente resolver, ambigüedades en el manejo de estos conceptos por parte de los organismos. El Conocimiento en este sistema se maneja a través de ontologías, utilizando OWL [6] (Web Ontology Languge) como lenguaje de especificación y Protégé [7] como herramienta de modelado. El Sistema de Gestión de Metadatos expone interfaces, a través de Web Services, para que otros sistemas puedan interactuar con él.

Servicios Transversales de la PGE

Los Servicios Transversales de la PGE son actualmente el Portal y el Buscador del Estado Uruguayo, el Sistema de Expediente Electrónico y el Geoportal.

El **Portal del Estado Uruguayo** es uno de los principales puntos de entrada al Gobierno Electrónico, permitiendo la interacción de los ciudadanos con contenidos, servicios y trámites de interés público. Desde el punto de vista tecnológico, el portal está basado en la herramienta WebSphere Portal [8] de IBM, complementada con un manejador de contenido y herramientas de estadística. Entre sus principales características se encuentran el soporte a estándares de la industria, como las especificaciones de Portlets Java [9][10] y Web Services for Remote Portlets (WSRP) [11], su capacidad dinámica de personalización y su cumplimiento con pautas de accesibilidad de sitios Web.

El **Buscador del Estado Uruguayo** tiene como objetivo instrumentar una herramienta de búsqueda orientada a las necesidades del gobierno electrónico en Uruguay. La principal ventaja del buscador, con respecto a otros como Google, es que está específicamente optimizado para realizar búsquedas de información del Estado Uruguayo. A nivel tecnológico, está implementado utilizando el producto Google Search Appliance [12] complementado con indexación de texto, búsqueda por palabras claves, detección de errores de digitación y errores ortográficos, glosarios y taxonomías.

El **Sistema de Expediente Electrónico** tiene como objetivo principal informatizar el manejo de Expedientes a nivel del Estado Uruguayo y facilitar la interoperabilidad de los mismos a través de los diferentes organismos. El principal componente del sistema es una aplicación de gestión de expedientes electrónicos, que puede ser utilizada bajo la modalidad de *software* como servicio (Software as a Service, SaaS) o puede ser instalada localmente en los organismos del Estado. Una aplicación Web ofrecerá a ciudadanos y organismos la posibilidad de consultar a través de Internet la trazabilidad de los expedientes en que está involucrado. Finalmente, un módulo de ruteo y trazabilidad permitirá el intercambio y trazabilidad de todas las actuaciones realizadas sobre todos los expedientes.

Por último, el **Geoportal** es un portal de información geográfica que permite la consulta y análisis vía Web de la información geográfica proveniente de los organismos. El Geoportal, se encuentra enmarcado en el proyecto IDE (Infraestructura de Datos Espaciales), el cual tiene como

objetivo principal crear un servicio en red para acceder y compartir datos geográficos entre los Organismos del Estado.

Plataforma de Middleware

El objetivo de la Plataforma de Middleware es fomentar la interoperabilidad entre los diferentes Organismos del Estado proveyendo los mecanismos necesarios para facilitar el desarrollo, despliegue e integración de servicios y aplicaciones. Estos mecanismos brindan a su vez, la infraestructura base para la implementación de la SOA a nivel del Estado.

A continuación, se brinda un ejemplo de funcionamiento de la Plataforma de Middleware, para luego describir sus principales componentes.

Ejemplo de Uso de la Plataforma de Middleware

El ejemplo de uso de la Plataforma de Middleware consta de un servicio "Cédula de Identidad" (CI), que permite, a partir de un número de cédula, obtener información pública de una persona. El servicio es provisto por los organismos A y B, pero con las diferencias que se presentan en la Tabla 1.

Característica	Servicio (del organismo) A	Servicio (del organismo) B
Información	Oficial	Réplica de datos provenientes del organismo A. El proceso de réplica se ejecuta de forma mensual.
Cantidad máxima de pedidos concurrentes	100	500
Formato de datos de entrada	Número de CI sin puntos, ni dígito verificador. P. ej: 25694581	Número de CI con puntos y dígito verificador. P ej: 2.569.458-1

Tabla 1 – Diferencias entre servicio A y el servicio B

Los principales consumidores del servicio de CI son los funcionarios del organismo C, los cuales acceden al mismo a través de la PGE por intermedio de aplicaciones de escritorio y Web. Asimismo, los Ciudadanos Uruguayos son otros potenciales consumidores, los cuales acceden al servicio utilizando el Portal del Estado Uruguayo.

La invocación de los servicios de la PGE se hace mediante el envío de mensajes. Por lo que, para consumir el servicio de CI, los funcionarios y ciudadanos deben enviar, a través de la aplicación en uso, un mensaje a la PGE con los datos de su solicitud.

Una vez que el mensaje llega a la PGE, y pasa los controles de seguridad necesarios, es reenviado a la Plataforma de Middleware que se encarga de realizar las siguientes acciones:

1. Verificación sintáctica: Se realizan validaciones de integridad como la verificación de nulos, estructuras de datos incompletas o errores en tipos de datos. En caso de encontrar errores, el mensaje es rechazado y se notifica al cliente los motivos.

2. Verificación de políticas de seguridad: Se realizan validaciones para determinar si el mensaje satisface las restricciones de seguridad definidas por la Ley 18.331 de Protección de Datos Personales y acción de Habeas Actas [13], y otras políticas definidas por la PGE. En casos de encontrar errores, el mensaje es rechazado y se notifica al cliente los motivos.

3. Elección del destino del mensaje: Se elige el mejor destino del mensaje. En este ejemplo, existen dos posibles destinos: 1) el servicio A y 2) el servicio B. La política de direccionamiento de mensajes de la PGE define que "*Siempre se enviará el mensaje al servicio del organismo A si hay menos de 100 pedidos concurrentes pendientes. En caso contrario, se redirigirán los pedidos al servicio del organismo B.*"

4. Transformación de datos: En casos donde se haya optado por dirigir el mensaje al servicio B, es necesario transformar el pedido, ingresando la información faltante (puntos y guiones a la CI).

5. Envío del mensaje al servicio: Se envía el mensaje al servicio destino.

La Figura 10 presenta de forma gráfica el ejemplo descrito anteriormente.

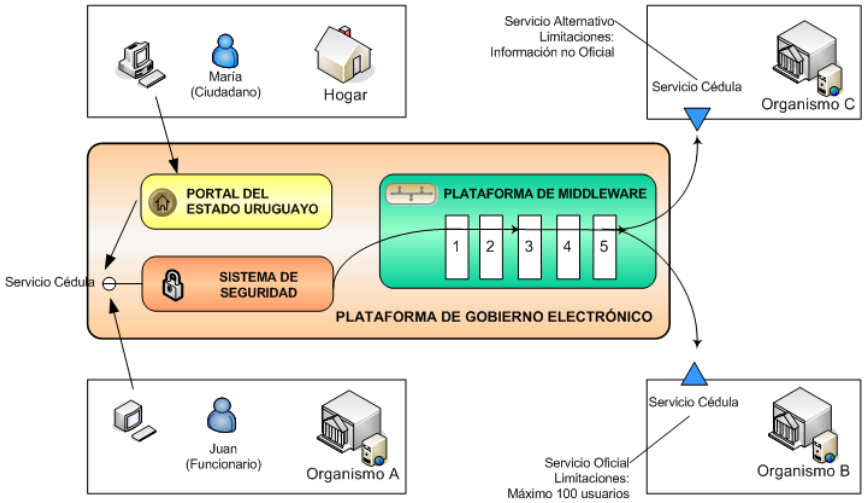

Figura 10 – Ejemplo de Funcionamiento de la Plataforma de Middleware

La Tabla 2 presenta los principales beneficios que obtienen consumidores y productores de servicios del ejemplo al utilizar la PGE.

Beneficios	Consumidor	Productor
Transparencia en la ubicación de los servicios	Los consumidores no conocen la ubicación real del servicio. La PGE identifica el mejor destino y se encarga de manejar posibles errores (caída del servicio, etc).	No tienen que hacer pública la ubicación real de sus servidores, brindando un mayor nivel de seguridad para los mismos.
Balanceo de Carga	Dado que los servicios no se saturan, los tiempos de respuesta al consumidor no se ven degradados.	La PGE realiza un balanceo de carga de los pedidos de acuerdo a capacidades de los servicios, evitando su saturación.
Transformación de formatos	Los clientes no necesitan reconfigurarse debido a (algunos) cambios en el formato de los mensajes recibidos por el servicio.	La PGE puede aprovechar servicios con formatos legados haciendo transparente la transformación de formatos a los clientes.
Verificación de datos		Los servicios sólo procesan pedidos válidos. Procesamiento dedicado exclusivamente al negocio.

Tabla 2 – Beneficios del uso de la PGE (en el ejemplo) para Consumidores y Productores

Componentes de la Plataforma de Middleware

La Figura 11 presenta una visión general de la Plataforma de Middleware, en la que se distinguen tres grandes bloques: entornos de

ejecución para Aplicaciones y Servicios, un Registro de Servicios y productos de tipo Enterprise Service Bus.

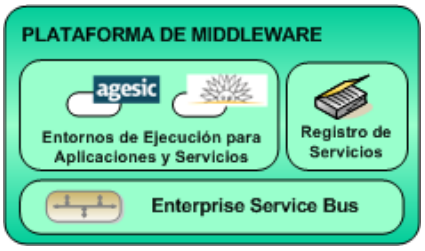

Figura 11 - Plataforma de Middleware

Entornos de Ejecución

Si bien en general las aplicaciones y servicios de la PGE se alojan en los propios organismos, la Plataforma de Middleware provee entornos de ejecución para alojar aplicaciones y servicios en la propia PGE. Estos entornos se basan en tecnologías de Middleware tales como Servidores de Aplicaciones, entre otros.

Los organismos pueden aprovechar estos entornos para alojar en la PGE servicios o aplicaciones que requieran infraestructura de *hardware* o *software* avanzada, no disponible en los mismos. Esta infraestructura puede ser necesaria para garantizar determinados niveles de calidad de servicio en relación, por ejemplo, a tiempos de respuesta y disponibilidad.

Por otro lado, los entornos de ejecución se utilizan también para servicios, componentes o aplicaciones que brindan funcionalidades comunes o utilitarias. A modo de ejemplo, existe actualmente en la PGE un servicio "Timestamp" provisto por AGESIC, el cual provee la fecha y hora actual. La Figura 12 presenta gráficamente este servicio[1].

Figura 12 - Servicio de Timestamp

La Plataforma de Middleware proporciona entornos de ejecución a través de dos de las principales plataformas para el desarrollo de aplicaciones empresariales: la plataforma .NET [14] de Microsoft y la plataforma Java

[1] [ip-srv-pge] corresponde a la dirección IP del servidor donde está alojado el servicio

Enterprise Edition (Java EE) [15]. Esta última se provee a través del JBoss Enterprise SOA Platform [16]. Además de estos dos entornos de ejecución, la plataforma cuenta con otros componentes que también permiten la implementación y ejecución de lógica de negocio. A modo de ejemplo, se cuenta con motores para la ejecución de procesos y reglas de negocio, como motores WS-BPEL.

Registro de Servicios

El Registro de Servicios de la PGE provee funcionalidades para que los organismos publiquen, describan, busquen y descubran servicios en la PGE.

La Figura 13 presenta, por ejemplo, dos de los servicios publicados en el Registro de Servicios. En primer lugar, se encuentra el servicio "Certificado de Nacidos Vivos" provisto por el Ministerio de Salud Pública (MSP). En segundo lugar, se encuentra el servicio "Timestamp", descripto previamente y provisto por AGESIC.

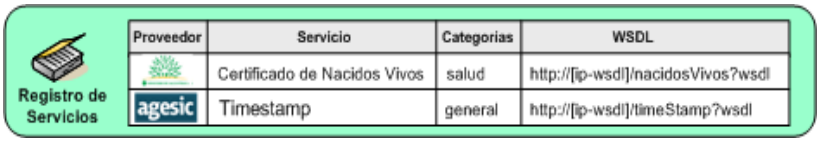

Figura 13 - Directorio de Servicios[2]

Además de los nombres, proveedores y categorías de los servicios, es posible acceder a la descripción de los mismos, especificada en WSDL, que brinda los datos necesarios para que una aplicación cliente pueda invocarlos.

Si bien actualmente el Registro de Servicios se maneja de forma interna a AGESIC, se planea brindar un registro UDDI [17] a través del cual los organismos podrán buscar y descubrir servicios de acuerdo a distintos criterios. A modo de ejemplo, la Figura 14 presenta dos búsquedas que se podrían realizar en un registro UDDI y sus resultados. En el primer caso, el usuario Juan busca servicios cuyo proveedor es "AGESIC"; la búsqueda retorna entonces el servicio "Timestamp". En el segundo caso, la usuaria Ana busca servicios en la categoría "salud"; la búsqueda retorna entonces el servicio "Certificado de Nacidos Vivos". El resultado de una búsqueda retorna la información necesaria para poder invocar a los servicios que se retornan.

[2] [ip-wsdl] corresponde a la dirección IP del servidor donde están alojadas las descripciones de los servicios

Figura 14 - Búsqueda de Servicios

Productos Enterprise Service Bus

Los productos de ESB de la Plataforma de Middleware proveen mecanismos que pueden ser utilizados por los organismos para el consumo y provisión de servicios. La Plataforma de Middleware cuenta con dos productos de tipo ESB: JBoss ESB [16] y Microsoft Biztalk Server [18] complementado con el Biztalk ESB Toolkit [19].

A continuación se describen algunos de los principales mecanismos que brindan estos productos en el marco de la PGE.

Transparencia de Ubicación

Los productos de ESB proveen mecanismos que permiten a la PGE brindar transparencia en la ubicación de los servicios que se acceden a través de ella, esto es, las aplicaciones cliente no conocen la dirección real (física) de los servicios que invocan. Cuando una aplicación cliente quiere invocar un servicio, debe enviar un pedido a la PGE especificando, a través de una dirección lógica, el servicio que se quiere invocar. Esta dirección lógica identifica al servicio en la plataforma.

El mapeo entre direcciones lógicas y físicas es gestionado en los ESBs. De esta forma, si la ubicación de un servicio cambia, las aplicaciones cliente no deben ser modificadas dado que basta con configurar la nueva dirección en el ESB. Por otro lado, los proveedores de servicios se benefician de este mecanismo, ya que no tienen que hacer pública la ubicación real de sus servidores, brindando un mayor nivel de seguridad para los mismos.

Para especificar la dirección lógica del servicio que se quiere invocar, una aplicación cliente debe utilizar el estándar WS-Addressing [20]. En la sección “Consumo de un Servicio en la PGE” se describe con más detalle cómo debe especificarse esta dirección en el mensaje SOAP utilizado en la invocación del servicio.

Mecanismos de Mensajería Confiable

Los productos de ESB de la PGE permiten brindar mecanismos de mensajería confiable utilizando los modelos *point-to-point* y *publish-and-subscribe* [21].

En particular, el modelo *publish-and-subscribe* se basa en una comunicación de tipo *broadcast*, donde un emisor/productor envía un mensaje que reciben varios receptores/consumidores. En este modelo, los consumidores se suscriben a un determinado evento/tópico de información y cada vez que los productores generan un mensaje sobre un tópico/evento determinado, éste es automáticamente redirigido a los consumidores suscritos al mismo.

El modelo *publish-and-subscribe* puede ser aplicado, por ejemplo, a la modificación de padrones del territorio nacional, donde existen varios interesados en esta información. En este escenario, el evento/tópico es "modificación de padrón", el productor es la Dirección Nacional de Catastro (DNC) y los consumidores pueden ser la Dirección General de Registro (DGR) y la Intendencia Municipal de Montevideo (IMM). Estos últimos, se suscribirán al tópico "modificación de padrón" y cada vez que DNC genere un mensaje para este tópico, el sistema de mensajería reenviará dicho mensaje a la IMM y DNC. En caso que éstos no estén activos, podrán posteriormente consultar el almacén de mensajes en busca de notificaciones perdidas.

Transformación y Enriquecimiento de Mensajes

Los productos de ESB de la PGE proveen varios mecanismos para transformar y enriquecer mensajes que fluyen entre clientes y servicios, por ejemplo, a través del estándar XSLT [22]. Estos mecanismos de transformación pueden utilizarse para abordar distintos requerimientos, como la resolución de discrepancias entre los formatos de datos intercambiados. A modo de ejemplo, si un cliente maneja datos en formato XML y un servicio espera datos en formato CSV[3], se podría utilizar XSLT para transformar los datos XML en datos CSV.

Además, las transformaciones podrían utilizarse para minimizar el impacto, en las aplicaciones cliente, ante cambios en los servicios. Por ejemplo, si un servicio cambiara su interfaz funcional, en ciertas ocasiones se podrían utilizar transformaciones para que los mensajes enviados por clientes se ajusten a la nueva interfaz. Dado que esto se resuelve en el ESB, los cambios en los servicios no tendrían impacto en las aplicaciones cliente.

[3] CSV – Comma-separated values (Valores Separados por Coma)

Ruteo Basado en Contenido

Otra funcionalidad que proveen los productos de ESB de la PGE, es la posibilidad de direccionar mensajes de acuerdo a su contenido. Los Content Based Routing (CBR) Services son servicios especializados que pueden ser introducidos entre cliente y servicio con el fin de inspeccionar el mensaje y a partir de su contenido redirigirlo a un determinado servicio.

Un ejemplo de un servicio CBR es el servicio de Ruteo de Mensajes de la PGE, el cual examina el mensaje para determinar su destino. En particular, se consulta el cabezal WS-Addressing "To", el cual contiene el servicio que se quiere invocar.

Monitoreo

Los productos de ESB cuentan con varias funcionalidades nativas que permiten el monitoreo de distintos tipos de información como tiempos de respuesta de los servicios, contenido de los mensajes, cantidad de invocaciones a los servicios, etc.

Otros Posibles Mecanismos

Dado el rol mediador de los ESBs, es de esperar que se implementen otros mecanismos que podrían ser utilizados por los organismos, tanto consumidores como proveedores de servicios. Algunos ejemplos de estos mecanismos son:

- **Tolerancia a Fallas**

  Este mecanismo permitiría que en caso de que un servicio no esté disponible o falle por algún motivo, se pueda invocar a otro equivalente de forma transparente al cliente que efectúa la invocación.

- **Control y Balanceo de Carga**

  En este caso, un organismo podría solicitar a AGESIC que se controle la cantidad de invocaciones a un determinado servicio, para que reciba como máximo un número dado de invocaciones por período de tiempo.

- **Validaciones**

  Un organismo que sabe que gran parte de las invocaciones a sus servicios no son válidas, podría solicitar realizar validaciones en los productos de ESB para no saturar a sus servidores con estas invocaciones. Esto además, reduciría el tráfico de red.

- **Aplicación de Políticas**

  AGESIC podría implementar controles, por ejemplo asociados a la ley de Protección de Datos Personales, en base a las funcionalidades de los productos de ESB.

Sistema de Seguridad

El Sistema de Seguridad de la PGE provee un conjunto de mecanismos que facilitan la implementación de requerimientos de seguridad a aplicaciones, servicios o componentes en el marco de la PGE. En particular, permite que los organismos deleguen a la PGE la tarea de controlar el acceso a los servicios que proveen.

A continuación, se describe el funcionamiento, componentes y prestaciones del Sistema de Seguridad.

Ejemplo de Uso del Sistema de Seguridad

En esta sección se brinda un ejemplo de uso del Sistema de Seguridad, con el fin de proveer una idea general de su funcionamiento para controlar el acceso a los servicios de la PGE.

El control de acceso en la plataforma se realiza a nivel de métodos, por lo que cuando un organismo publica un servicio debe especificar quién tiene acceso a cada método del mismo. En este ejemplo, como se presenta en la Figura 15, el organismo B provee el servicio de CI que tiene dos métodos: getNombre y getFechaNacimiento. El organismo B brinda acceso a los funcionarios del organismo A para invocar únicamente al método getNombre. De esta forma, si un funcionario del organismo A intenta invocar el método getFechaNacimiento, la PGE le negará el acceso al mismo.

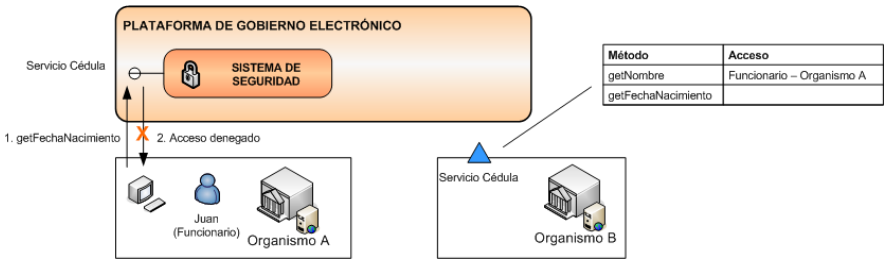

Figura 15 – Ejemplo de Funcionamiento del Sistema de Seguridad

Como se presenta en el ejemplo, el organismo proveedor del servicio puede delegar al Sistema de Seguridad de la PGE el control de acceso a

los servicios que provee, por lo que no debe preocuparse ni invertir recursos en esta tarea.

Componentes del Sistema de Seguridad

Como se puede observar en la Figura 16, el Sistema de Seguridad se puede dividir en tres grandes bloques: Sistema de Auditoría, Sistema de Control de Acceso y Servicio Periféricos de Seguridad.

Figura 16 - Sistema de Seguridad

Sistema de Auditoría

El Sistema de Auditoría provee las herramientas necesarias para realizar auditorías de seguridad sobre la PGE. Este sistema recolecta información y realiza análisis y reportes de auditoría. El Sistema de Auditoría está implementado por el producto Tivoli Compliance Insight Manager (TCIM)[23].

Servicios Periféricos de Seguridad

Los Servicios Periféricos de Seguridad tienen la finalidad de brindar los mecanismos necesarios para facilitar a los organismos el acceso seguro a la PGE. Como se observa en la Figura 17, en este componente existen dos servicios principales: Autoridad Certificadora (Certification Authority, CA) y Servicio de Directorio.

Figura 17 - Servicios Periféricos de Seguridad

La **CA** tiene como cometido emitir y gestionar los certificados de propósito general que se utilicen en la PGE. Por ejemplo, la CA tiene a cargo la emisión de los certificados que deben utilizar los servidores de los organismos para establecer conexiones seguras con la PGE. La CA es provista por el producto Windows 2003 Server.

El **Servicio de Directorio** provee servicios de directorio a través del protocolo LDAP, y tiene cuatro funciones principales [25]:

- replicar automáticamente las estructuras de directorio de los organismos que cuenten con este servicio
- proveer servicio de directorio a aplicaciones de la PGE
- proveer servicio de directorio a organismos que no cuenten con este servicio
- brindar una visión unificada de las estructuras de directorio de los organismos y la PGE

El Servicio de Directorio está implementado principalmente por los productos IBM Directory Server, Tivoli Identity Manager (TIM) [36] y Tivoli Directory Integrator (TDI) [37].

La Figura 18 presenta la estructura base del árbol de directorio manejado por la PGE, y a partir de dónde se integran los árboles de directorio de los organismos.

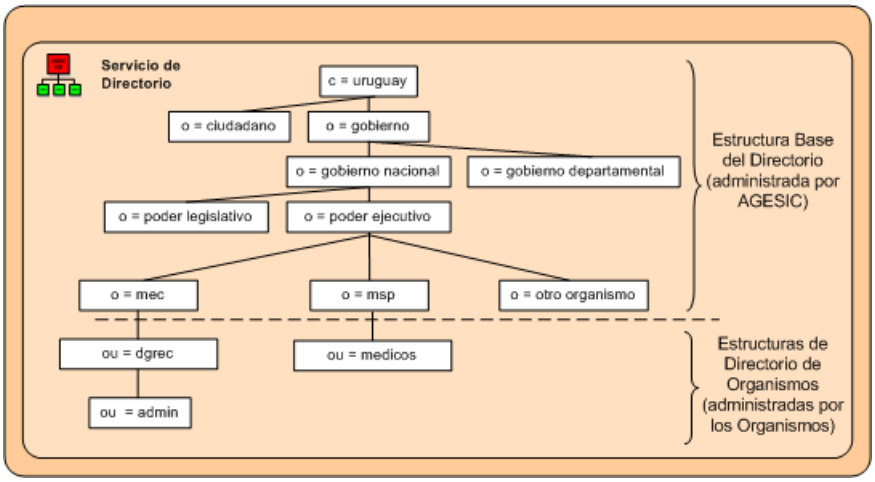

Figura 18 - Estructura Base del Directorio

A modo de ejemplo, se puede observar que hay definida una Organización "mec" que corresponde al Ministerio de Educación y Cultura (o =mec), la cual tiene una Unidad Organizacional denominada "dgrec" (ou = dgrec) que corresponde a la Dirección General de Registro del Estado, en la cual existen una entrada denominada "admin" que corresponde a un rol existente en la unidad organizacional.

Sistema de Control de Acceso

La finalidad del Sistema de Control de Acceso de la PGE es brindar los mecanismos para aplicar políticas de control de acceso sobre los servicios publicados y las aplicaciones disponibles en la PGE. El control de acceso

en la PGE se realiza siguiendo un esquema RBAC, utilizando el rol del usuario que quiere acceder al servicio o aplicación, y las políticas de acceso definidas en la PGE. En esta sección se describen las principales características, los componentes y el funcionamiento de este sistema, para realizar el control de acceso sobre los servicios.

Como se presenta en la Figura 19, el Sistema de Control de Acceso para Servicios consiste de tres componentes: un Servicio de Tokens de Seguridad, un Administrador de Políticas de Seguridad y un Firewall XML.

Figura 19 – Sistema de Control de Acceso para Servicios

El **Servicio de Tokens de Seguridad (Security Token Service, STS)** tiene la responsabilidad de emitir los *tokens* de seguridad necesarios para que las aplicaciones cliente puedan invocar a los servicios publicados en la PGE. Este componente soporta el estándar WS-Trust v1.3[24] y es implementado por el producto Tivoli Federated Identity Manager (TFIM) [25].

Para emitir los *tokens* de seguridad la PGE confía en las autenticaciones realizadas en los sistemas de los organismos, verificando la autenticidad de las solicitudes mediante el uso de Firma Electrónica [31].

Cuando una aplicación cliente de un organismo quiere consumir un servicio publicado en la plataforma, debe solicitar un *token* de seguridad al STS de la PGE utilizando el estándar WS-Trust. En esta solicitud se debe incluir otro *token* de seguridad que incluya, entre otros datos, el rol de usuario con el que se quiere acceder al servicio. Este *token* debe especificarse utilizando el estándar SAML v1.1 o v2.0 y debe además estar firmado electrónicamente por el organismo cliente.

Cuando la PGE recibe un pedido para el STS, verifica la firma digital del *token* de seguridad incluido en el pedido, de forma de corroborar que se trata de un consumidor en el que se confía. Además, se verifica que el rol del usuario, incluido en el *token* de seguridad, exista en el Directorio LDAP. Si la firma digital es verificada y el rol de usuario existe, el STS emite un *token* de seguridad firmado por la PGE. Este *token* de seguridad

se especifica utilizando el estándar SAML v1.1 e incluye, entre otros datos, el rol del usuario.

La comunicación entre las aplicaciones cliente y el STS de la PGE se debe realizar a través de HTTPS. En la sección "Conexiones SSL con la PGE" se brindan más detalles sobre las conexiones SSL que se deben establecer, entre los organismos y la PGE, para cumplir con este requerimiento.

La Figura 20 presenta un resumen de los pasos que debe seguir una aplicación cliente para solicitar un *token* de seguridad al STS de la PGE.

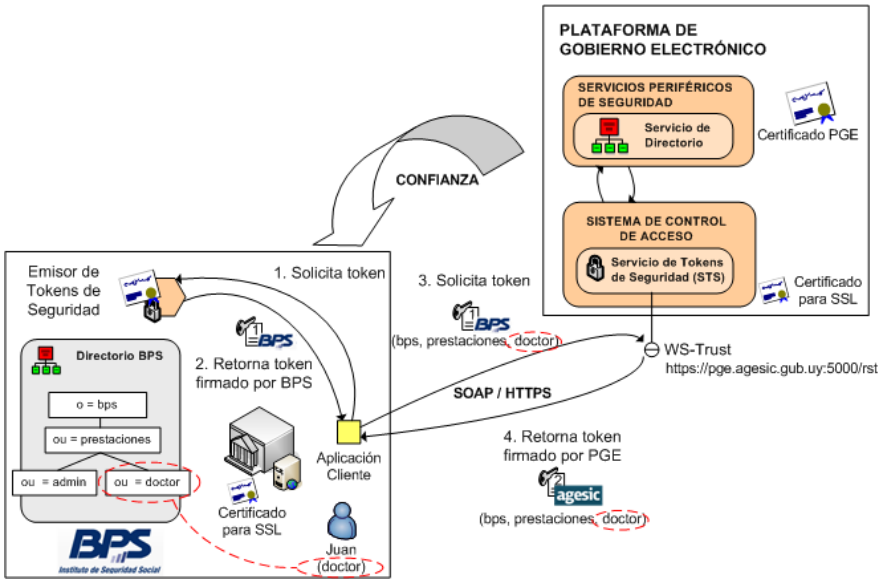

Figura 20 - Security Token Service

Primero, en los pasos 1 y 2, la aplicación cliente obtiene un *token* de seguridad firmado por el organismo, en este caso BPS. Para esto, se puede utilizar el estándar WS-Trust o cualquier otro mecanismo interno al organismo. Luego, en el paso 3, la aplicación cliente envía una solicitud de *token* de seguridad al STS de la PGE. Para esto se debe utilizar el estándar WS-Trust e incluir el *token* previamente obtenido. Finalmente, en el paso 4, si la firma del *token* enviado es verificada y el rol de usuario especificado en el *token* existe en el directorio LDAP, el STS emite un *token* de seguridad firmado por la PGE.

En la sección "Consumo de un Servicio en la PGE" se especifica con más detalle los datos que se deben incluir en la solicitud del *token* al STS de la PGE, y los datos que se devuelven en la respuesta.

El **Administrador de Políticas de Seguridad** actúa como Punto de Decisión de Políticas (Policy Decision Point, PDP) siendo responsable por tomar la decisión de autorizar, o no, los pedidos de invocación a

servicios de la PGE. Este componente es implementado por el producto Tivoli Security Policy Manager (TSPM) [35].

En este componente se especifica qué roles tienen acceso a los métodos de los servicios de la PGE. Para esto, es necesario definir los Perfiles de Usuario que accederán a cada servicio, los métodos a los que tienen acceso estos perfiles, y con qué roles (de los organismos) se corresponden.

A modo de ejemplo, la Figura 21 presenta las políticas que se pueden definir para el servicio "Certificado de Nacidos Vivos".

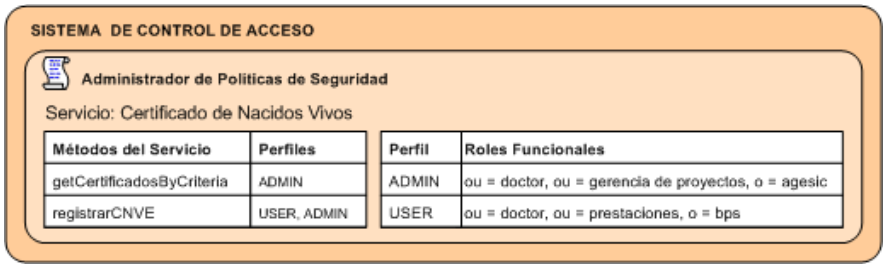

Figura 21 - Administrador de Políticas de Seguridad

En este caso se definen dos Perfiles de Usuario: ADMIN y USER. El Perfil de Usuario ADMIN tiene acceso al método "getCertificadosByCriteria" y está asociado al rol funcional "ou=doctor, ou=gerencia de proyectos, o=agesic". De forma similar, el Perfil de Usuario USER tiene acceso al método "registrarCNVE" y está asociado al rol funcional "ou=doctor, ou=prestaciones, o=bps".

Para que el Administrador de Políticas pueda tomar la decisión de autorizar, o no, la invocación a un método de un servicio, el cliente debe especificar el servicio y método que se quiere invocar. Para esto se utiliza el estándar WS-Addressing como se detalla en la sección "Consumo de un Servicio en la PGE".

Además, como se menciona anteriormente, en la invocación del servicio se incluye también el rol funcional del usuario con el que se quiere realizar la invocación.

De esta forma, cuando la PGE recibe un pedido de invocación para a un método de un servicio, el Administrador de Políticas de Seguridad cuenta con toda la información necesaria para permitir o negar el acceso.

Por último, el **Firewall XML** actúa como Punto de Aplicación de Políticas (Policy Enforcement Point, PEP) de acuerdo a lo que decida el

Administrador de Políticas de Seguridad. Este componente está implementado por el producto IBM Websphere Datapower Xi50 [29].

Para que el Firewall XML pueda actuar como PEP, para cada servicio de la PGE se despliega un Servicio Proxy en dicho Firewall. A modo de ejemplo, la Figura 22 presenta los Servicios Proxy de los servicios "Timestamp" y "Certificado de Nacidos Vivos". Una aplicación que quiera invocar a estos servicios debe hacerlo a través de sus Servicios Proxy. En caso de que se permita el acceso, el Firewall XML redirige el pedido a la Plataforma de Middleware, la cual envía la solicitud al servicio real.

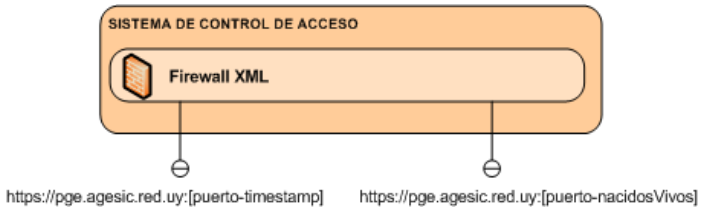

Figura 22 - Firewall XML

La comunicación entre las aplicaciones cliente y Servicios Proxy, así como la comunicación entre la PGE y los servicios en los organismos se realiza a través de HTTPS. En la sección "Conexiones SSL con la PGE" se brindan más detalles sobre las conexiones SSL que se deben establecer, entre los organismos y la PGE, para cumplir con este requerimiento.

A modo de resumen, la Figura 23 presenta los pasos que debe realizar una aplicación cliente para acceder a un servicio de la PGE.

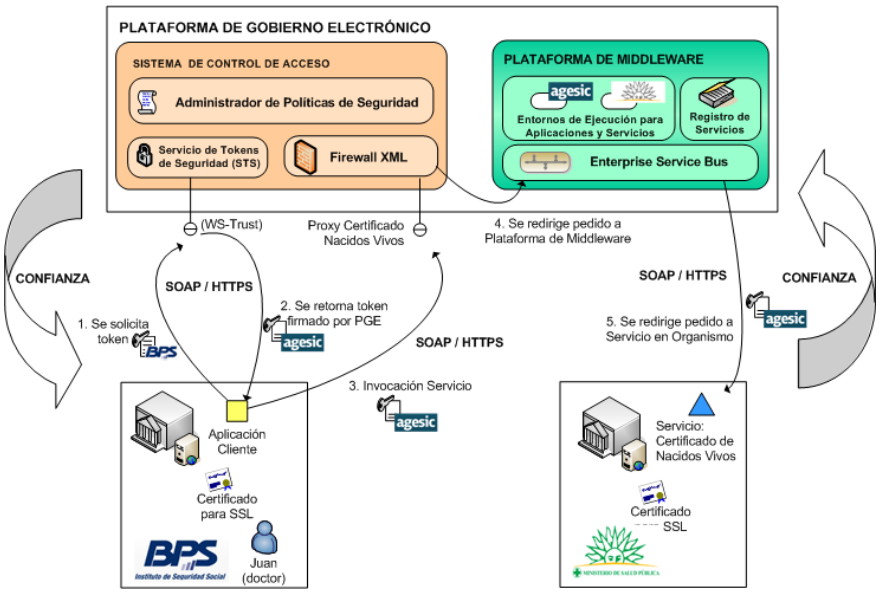

Figura 23 –Acceso a un Servicio de la PGE

En el paso 1 la aplicación cliente solicita un *token* de seguridad al STS[4] de la PGE, incluyendo en la solicitud un *token* firmado por el organismo, en este caso BPS, que contiene el rol del usuario. Si la firma del *token* es verificada y el rol del usuario existe en el Directorio LDAP, el STS, en el paso 2, devuelve a la aplicación un *token* de seguridad firmado por la PGE. Luego en el paso 3, la aplicación cliente invoca al servicio, a través de su Servicio Proxy, incluyendo en la invocación el *token* de seguridad firmado por la PGE. El Firewall XML permite o niega el acceso al servicio invocado, basándose en la decisión que tome el Administrador de Políticas de Seguridad. En caso de que el acceso sea permitido, el Firewall XML redirige el pedido, en el paso 4, a la Plataforma de Middleware la cual finalmente, en el paso 5, redirige el pedido al servicio5.

En la sección "Consumo de un Servicio en la PGE" se explica en detalle este procedimiento especificando la información que se debe enviar en la invocación al servicio.

Conectividad con la PGE

Para que un organismo pueda comunicarse con la PGE, ya sea para proveer o consumir servicios, es necesario que:

- el organismo esté conectado a la REDuy
- los *firewalls* de REDuy estén configurados para habilitar el tráfico de red requerido
- se puedan establecer conexiones SSL entre el organismo y la PGE

En esta sección se describen cada uno de estos requerimientos, especificando cómo debe proceder un organismo para cumplirlos.

[4] El acceso al STS se efectúa también a través del Firewall XML pero se omite para simplificar los diagramas.

[5] La comunicación entre la Plataforma de Middleware y el servicio también pasa a través del Firewall XML, pero se omite para simplificar los diagramas.

Conexión con REDuy

Como se menciona previamente en este documento, la REDuy provee la infraestructura de conectividad necesaria para que los organismos se interconecten entre sí, y con la PGE.

Para que un organismo provea un servicio en la PGE es necesario, entonces, que el organismo en el que se aloja el servicio esté conectado a la REDuy.

De forma similar, para que un organismo pueda consumir un servicio de la PGE, es necesario que el organismo en el que se aloja la aplicación cliente esté conectado a la REDuy.

En caso de no contar con conexión a la REDuy, un organismo puede solicitarla enviando un correo electrónico a soporte@agesic.gub.uy, especificando en el asunto del correo "[Conexión a REDuy] *Nombre del Organismo Solicitante*".

Configuración de Firewalls de REDuy

Como se menciona previamente en este documento, la conexión de los organismos a la REDuy está protegida por *firewalls* que controlan el tráfico de red de los organismos, desde y hacia la REDuy.

Para que un organismo provea un servicio en la PGE es necesario, entonces, que estos *firewalls* estén configurados para habilitar el tráfico de red desde la PGE hacia el servidor donde se aloja el servicio.

La configuración de los firewalls está a cargo del equipo de soporte de AGESIC y se realiza una vez que se recibe y aprueba la petición de publicación de un servicio. Esta solicitud se realiza mediante el "**¡Error! No se encuentra el origen de la referencia.**" que se encuentra en el Apéndice 2.

Por otro lado, para que un organismo consuma servicios en la PGE, no es necesario realizar ninguna configuración adicional en los *firewalls*, dado que el tráfico hacia la REDuy se habilita al momento de instalarlos.

Conexiones SSL con la PGE

Como se menciona en secciones anteriores, al consumir servicios de la PGE, la comunicación entre las aplicaciones cliente y la PGE, así como la comunicación entre la PGE y los servicios, se realiza a través de mensajes SOAP sobre HTTPS. Para esto es necesario establecer conexiones SSL entre la PGE y los organismos.

Para poder establecer estas conexiones, un organismo debe seguir los siguientes pasos:

1. Solicitar a AGESIC un certificado digital emitido por la CA de la PGE. La solicitud debe realizarse a través de un pedido de certificado en formato PKCS#10, el cual debe enviarse por correo electrónico a soporte@agesic.gub.uy junto con el "**¡Error! No se encuentra el origen de la referencia.**" o el "**¡Error! No se encuentra el origen de la referencia.**" (que se encuentran en el Apéndice 2), según el caso.

2. Instalar el certificado raíz de la CA de la PGE en los servidores o computadores del organismo, en donde se encuentran los servicios o aplicaciones que interactuarán con la PGE.

Por otro lado, las conexiones SSL que se establecen entre los organismos y la PGE deben cumplir los siguientes requisitos:

- Ser compatible con SSL v3.0
- Utilizar client_authentication con Certificados Digitales X.509 v3
- Utilizar certificados digitales emitidos por la CA de la PGE
- Soportar algunas de las CipherSuite especificadas en la Tabla 3

RC4_MD5_EXPORT Cipher
RC4_MD5_US Cipher
RC4_SHA_US Cipher
RC4_56_SHA_EXPORT1024 Cipher
TRIPLE_DES_SHA_US Cipher
TLS_RSA_WITH_AES_128_CBC_SHA Cipher
TLS_RSA_WITH_AES_256_CBC_SHA Cipher
AES_SHA_US Cipher

Tabla 3 - CipherSuites para SSL

Referencias

[1] AGESIC – REDuy. http://www.agesic.gub.uy/innovaportal/v/759/1/agesic/REDuy.html [Accedida en Abril de 2010]

[2] Centro Nacional de Respuesta a Incidentes en Seguridad Informática (CERTuy) http://www.cert.uy/ [Accedida en Abril de 2010]

[3] Basic Profile. http://www.ws-i.org/Profiles/BasicProfile-1.1.html [Accedida en Mayo de 2010]

[4] Basic Security Profile. http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0.html [Accedida en Mayo de 2010]

[5] AGESIC – Plataforma de Gobierno Electrónico. http://www.agesic.gub.uy/innovaportal/v/771/1/agesic/Plataforma-de-Gobierno-Electrónico.html [Accedida en Abril de 2010]

[6] Web Ontology Language (OWL) http://www.w3.org/2004/OWL/ [Accedida en Mayo de 2010]

[7] Protégé. http://protege.stanford.edu/ [Accedida en Mayo de 2010]

[8] WebSphere Portal. http://www-01.ibm.com/software/websphere/portal/ [Accedida en Mayo de 2010]

[9] JSR 168: Portlet Specification. http://jcp.org/en/jsr/detail?id=168 [Accedida en Mayo de 2010]

[10] JSR 286: Portlet Specification 2.0. http://jcp.org/en/jsr/detail?id=286, [Accedida en Mayo de 2010]

[11] Web Services for Remote Portlets (WSRP). http://www.oasis-open.org/committees/wsrp/ [Accedida en Mayo de 2010]

[12] Google Search Appliance. http://www.google.com/enterprise/gsa/ [Accedida en Mayo de 2010]

[13] Ley N° 18.331 de Protección de Datos Personales y Acción de Habeas Data. http://www.protecciondedatos.gub.uy/sitio/Leyes/Ley-18.331.pdf [Accedida en Junio de 2010]

[14] Microsoft .NET Framework. http://www.microsoft.com/net/ [Accedida en Mayo de 2010]

[15] Java Enterprise Edition. http://java.sun.com/javaee/ [Accedida en Mayo de 2010]

[16] JBoss Enterprise SOA Platform. http://www.jboss.com/products/platforms/soa/ [Accedida en Noviembre de 2010]

[17] OASIS UDDI Specification TC http://www.oasis-open.org/committees/uddi-spec/ [Accedida en Agosto de 2010]

[18] Microsoft Biztalk Server. http://www.microsoft.com/biztalk/ [Accedida en Mayo de 2010]

[19] Biztalk ESB Toolkit. http://msdn.microsoft.com/en-us/biztalk/dd876606.aspx [Accedida en Mayo de 2010]

[20] Web Services Addressing Working Group. http://www.w3.org/2002/ws/addr/ [Accedida en Junio de 2010]

[21] Dave Chappell. Enterprise Service Bus. O'Reilly. 2004.

[22] XSL Transformations (XSLT). http://www.w3.org/TR/xslt [Accedida en Mayo de 2010]

[23] Tivoli Compliance Insight Manager. http://www-01.ibm.com/software/tivoli/products/compliance-insight-mgr/ [Accedida en Abril de 2010]

[24] WS-Trust 1.3. http://docs.oasis-open.org/ws-sx/ws-trust/200512/ws-trust-1.3-os.html [Accedida en Junio de 2010]

[25] Sistema de Seguridad de la Plataforma de Gobierno Electrónico. Presentación. 2009. http://www.agesic.gub.uy/innovaportal/file/758/1/seguridad.pdf [Accedida en Abril de 2010]

[26] Tivoli Federated Identity Manager. http://www-01.ibm.com/software/tivoli/products/federated-identity-mgr/ [Accedida en Abril de 2010]

[27] Tivoli Access Manager. http://www-01.ibm.com/software/tivoli/products/access-mgr-productline/ [Accedida en Abril de 2010]

[28] Tivoli Security Policy Manager. http://www-01.ibm.com/software/tivoli/products/security-policy-mgr/ [Accedida en Abril de 2010]

[29] WebSphere DataPower Integration Appliance XI50. http://www-01.ibm.com/software/integration/datapower/xi50/ [Accedida en Abril de 2010]

[30] JXplorer – Java LDAP Browser. http://jxplorer.org/ [Accedida en Mayo de 2010]

[31] Ley Nº 18.600. Documento Electrónico y Firma Electrónica http://www.parlamento.gub.uy/leyes/AccesoTextoLey.asp?Ley=18600 [Accedida en Mayo de 2010]

[32] Basic Profile. http://www.ws-i.org/Profiles/BasicProfile-1.1.html [Accedida en Mayo de 2010]

[33] Basic Security Profile. http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0.html [Accedida en Mayo de 2010]

[34] AGESIC. Guía de Programación Java para la Plataforma de Gobierno Electrónico. Junio 2010.

[35] Tivoli Security Policy Manager. http://www-01.ibm.com/software/tivoli/products/security-policy-mgr/ [Accedida en Mayo de 2010]

[36] Tivoli Identity Manager.
http://www-01.ibm.com/software/tivoli/products/identity-mgr/
[Accedida en Mayo de 2010]

[37] Tivoli Directory Integrator.
http://www-01.ibm.com/software/tivoli/products/directory-integrator/
[Accedida en Mayo de 2010]

Capítulo IV

Guía de Programación Java para la Plataforma de Gobierno Electrónico

Introducción

Este capítulo brinda guías de desarrollo para la implementación de servicios y aplicaciones cliente de la PGE, utilizando Java.

En primera instancia se describe cómo utilizar una Librearía de Ejemplo, desarrollada por AGESIC, que muestra cómo obtener un token de seguridad firmado digitalmente por la PGE.

Luego se provee una guía, paso a paso, para el desarrollo de una aplicación Java de escritorio (Aplicación Cliente) que consuma un servicio de la PGE. Para esto se utiliza Eclipse [1] y JBoss Tools [2] como entorno de desarrollo, y JBossWS - Native [3] como stack de Web Services. Asimismo, se utiliza la Librearía de Ejemplo desarrollada por AGESIC.

Se recomienda leer previamente la Descripción Técnica de la PGE en el Capítulo III.

Librería de Ejemplo AGESIC

AGESIC desarrolló una librería Java para brindar un ejemplo de cómo obtener un *token* de seguridad firmado digitalmente por la PGE. En esta sección se describen las principales características de la librería y cómo utilizarla.

Importante: La librería fue desarrollada como prueba de concepto, por lo cual no está garantizada la ausencia de errores, ni fallas de seguridad. No se recomienda entonces utilizarla en producción, sin los resguardos apropiados según las políticas de *testing* y seguridad de cada organismo.

Descripción General

La Librería de Ejemplo resuelve las siguientes dos tareas:

- la solicitud y obtención de un *token* de seguridad SAML firmado digitalmente por el organismo
- la solicitud y obtención de un *token* de seguridad SAML firmado digitalmente por la PGE

Para realizar estas tareas, la librería provee la clase PGEClient, en el paquete uy.gub.agesic.sts.client. Dicha clase cuenta con el método

requestSecurityToken que es el encargado de invocar al STS de la PGE y retornar el *token* de seguridad emitido por éste.

Internamente, el método realiza las siguientes tareas:

- emite un *token* SAML firmado por el Organismo Cliente
- genera y envía al STS de la PGE un mensaje RequestSecurityToken (RST), siguiendo el estándar WS-Trust, en el que se incluye el *token* emitido y otros datos requeridos
- al recibir el mensaje de respuesta (RequestSecurityTokenResponse, RSTR), se comprueba que el *token* SAML recibido fue emitido por la PGE mediante la verificación de su firma digital

Obtención del token SAML firmado por la PGE

Para obtener un token SAML firmado por la PGE se debe crear una instancia de la clase PGEClient e invocar el método requestSecurityToken, el cual recibe tres parámetros: bean, keyStore y trustStore.

El parámetro bean (uy.gub.agesic.beans.RSTBean) aloja los datos que se utilizan para emitir el token firmado por el Organismo Cliente y para construir el mensaje RST a enviar al STS. Concretamente estos datos son: nombre de usuario, rol del usuario, servicio, issuer y policy name.

El parámetro keystore (uy.gub.agesic.beans.StoreBean) aloja los datos para acceder a la keystore donde se almacenan las claves y certificados digitales del organismo. Estos se utilizan para firmar el token de seguridad emitido y para establecer la conexión SSL. La Tabla 4 describe los datos que se deben especificar en este parámetro y los métodos de la clase StoreBean a utilizar para este fin.

Nombre	**Método**	**Descripción**
Alias	setAlias	Alias de la entidad en la *keystore*.
Ruta	setStoreFilePath	Ruta del archivo de la *keystore*
Contraseña	setStorePwd	Contraseña para acceder a la *keystore*.

Tabla 4 – Datos a Especificar en el parámetro keyStore

De forma similar, el parámetro trustStore (uy.gub.agesic.beans.StoreBean) aloja los datos para acceder a la trustStore donde se almacenan los certificados de la PGE. Estos certificados se utilizan para verificar la firma del token emitido por el STS y para establecer la conexión SSL. La Tabla 5 describe los datos que se deben especificar en este parámetro y los métodos de la clase StoreBean a utilizar para este fin.

Nombre	Método	Descripción
Ruta	setStoreFilePath	Ruta del archivo de la *trustStore*
Contraseña	setStorePwd	Contraseña para acceder a la *trustStore*.

Tabla 5 – Datos a Especificar en el parámetro trustStore

La Figura 24 presenta un ejemplo completo donde se invoca el método requestSecurityToken y se obtiene un uy.gub.agesic.beans.SAMLAssertion con el token de seguridad emitido por la PGE.

```
RSTBean bean = new RSTBean();
bean.setIssuer(issuer);
bean.setPolicyName(policyName);
bean.setRole(role);
bean.setUserName(userName);
bean.setService(service);

StoreBean keyStore = new StoreBean();
keyStore.setAlias(alias);
keyStore.setStoreFilePath(keyStoreFilePath);
keyStore.setStorePwd(keyStorePwd);

StoreBean trustStore = new StoreBean();
trustStore.setStoreFilePath(trustStoreFilePath);
trustStore.setStorePwd(trustStorePwd);

PGEClient client = new PGEClient();
SAMLAssertion assertionResponse =
client.requestSecurityToken(bean, keyStore,
trustStore);
```

Figura 24 – Obtener un *token* firmado por la PGE

La documentación Java de la librería se puede acceder en [4].

Tutorial: Consumir un Servicio de la PGE

Uno de los principales escenarios de uso de la PGE, es el consumo de servicios. En esta sección se presenta un tutorial que explica cómo consumir un servicio de la PGE utilizando Java.

Objetivo

El objetivo de este tutorial es proveer una guía, paso a paso, para el desarrollo de una aplicación Java de escritorio (Aplicación Cliente) que consuma un servicio de la PGE. Para esto se utiliza Eclipse y JBoss Tools como entorno de desarrollo, y JBoss WS – Native como *stack* de Web Services.

Prerrequisitos

Para implementar y ejecutar la Aplicación Cliente se debe cumplir con los prerrequisitos que se describen en esta sección.

Conocimientos Requeridos

Para comprender el tutorial se requiere que el lector esté familiarizado con el desarrollo de aplicaciones Java EE, haya desarrollado Web Services con tecnología Java y cuente con conocimientos de seguridad informática. Concretamente, se asume que el lector conoce los estándares WS-Addressing, WS-Security, WS-Trust y SAML, y tiene experiencia en el uso de certificados digitales.

Si no se cuenta con estos conocimientos, se puede consultar el marco técnico que se presenta en el Apéndice 1 y la bibliografía asociada.

Conectividad con la PGE

Para poder ejecutar en un organismo la Aplicación Cliente se requiere que:

- el organismo esté conectado a la REDuy
- los firewalls de REDuy estén configurados para habilitar el tráfico de red requerido

- se puedan establecer conexiones SSL entre el organismo y la PGE

Si no se cumple con alguno de estos prerrequisitos consultar la sección "Conectividad con la PGE" del Capítulo III.

Requerimientos de Software

La Tabla 6 presenta las herramientas y productos de *software* requeridos para desarrollar y ejecutar la Aplicación Cliente.

Producto	Versión
Java Developer Kit (JDK)	5.0, Update 22
JBoss Application Server	5.1
JBoss Web Services	3.2.2.GA
Eclipse	3.5 /Galileo
JBossWS Tools	3.1 GA
OpenSAML	2.3.1

Tabla 6 – Requerimientos de Software

Almacenes de Claves y Certificados

La ejecución del escenario requiere una *keyStore* y una *trustStore* que almacene los certificados y claves necesarias para establecer la conexión SSL, firmar los tokens SAML emitidos y verificar la firma de los *tokens* de seguridad emitidos por la PGE.

En la *keystore* se deben alojar las claves y certificados para:

- Firmar los *tokens* SAML emitidos.
- Llevar a cabo la comunicación SSL.

Por otro lado, la *truststore* necesita tener la siguiente información:

- Certificado de la CA de la PGE, utilizada para llevar a cabo la comunicación SSL.
- Certificado de la PGE para verificar la firma de los *tokens* SAML emitidos por la misma.

En [4] se pueden encontrar certificados, *truststores* y *keystores* de ejemplo. Por más información acerca de cómo generar un *keystore* y *truststore* en Java, ver [5]. Por información acerca de cómo importar un archivo pfx a la *keystore*, ver [6].

Descripción del Escenario

La Figura 25 presenta el escenario de ejemplo que se utiliza en este tutorial, en el cual intervienen dos organismos: el Banco de Previsión Social (BPS) (Organismo Cliente) y el Ministerio de Salud Pública (MSP) (Organismo Proveedor).

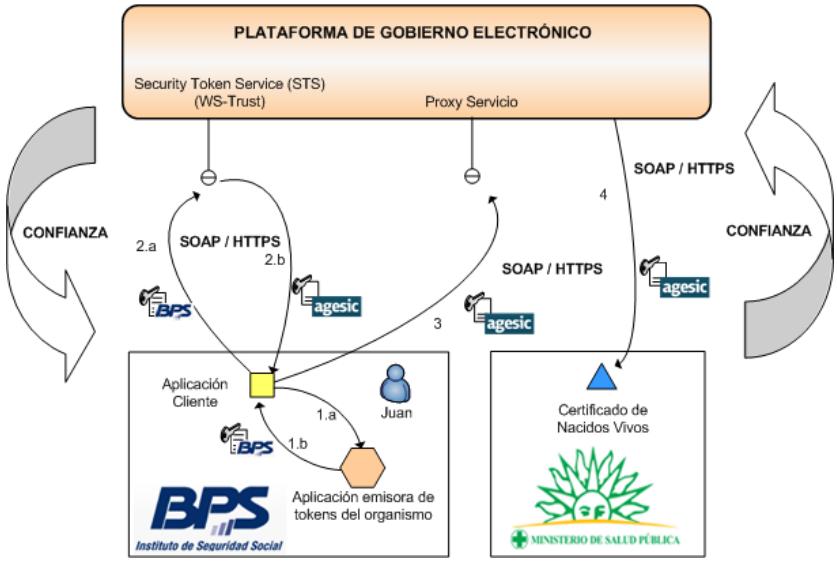

Figura 25 – Escenario de Uso de la PGE

El MSP provee el servicio "Certificado de Nacidos Vivos" el cual tiene dos métodos: "getCertificadosByCriteria" y "registrarCNEV". Cuando se registró el servicio en la PGE, se desplegó un Servicio Proxy en ella para que las Aplicaciones Cliente accedieran al servicio a través de él. Además, mediante la configuración de políticas de control de acceso, el MSP autorizó a los usuarios con rol "doctor" de la sección "prestaciones" del BPS a consumir el método "registrarCNEV" de dicho servicio.

Por otro lado, en el BPS hay una Aplicación Cliente que está siendo utilizada por el usuario Juan que tiene el rol "doctor" en la sección "prestaciones". La aplicación necesita acceder al servicio del MSP para lo cual, utilizando las credenciales del usuario Juan y a través de una Aplicación Emisora de Tokens interna al BPS, obtiene un *token* de seguridad SAML firmado por el BPS (pasos 1.a y 1.b).

Luego con dicho *token* obtiene del STS de la PGE, a través del estándar WS-Trust, otro *token* de seguridad firmado por la plataforma (pasos 2.a y 2.b). Para emitir este *token* la PGE verifica la firma digital del *token* enviado por la aplicación y la existencia del rol "ou=doctor, ou=prestaciones, o=bps".

Por último, la Aplicación Cliente invoca al Servicio del MSP mediante su Servicio Proxy. En la invocación se incluye el *token* firmado por la PGE y se especifican el servicio (Certificado de Nacidos Vivos) y método (registrarCNEV) a invocar. Dado que el usuario Juan está autorizado a utilizar el método del servicio, la invocación se efectúa de forma exitosa.

La Tabla 7 especifica algunos de los datos a utilizar en la implementación del escenario.

Dato	Valor
Nombre de Usuario	Juan
Rol de Usuario	ou=doctor, ou=prestaciones, o=bps
Dirección Lógica del Servicio	http://192.168.40.190:9000/Servicio
Método del Servicio	http://xml.cnve.msp.gub.uy/wsdl/certificadoCNVEWSDL/certificadoCNVEWSDLPortType/registrarCNVE
PolicyName[6]	urn:simpletoken
Tipo de *Token*[7]	http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLV1.1

Tabla 7 – Datos para la Implementación del Escenario

Los datos de negocio a incluir en la invocación, están especificados en la descripción del servicio (WSDL). En esta descripción también se incluye la dirección del Servicio Proxy a donde el cliente debe enviar los mensajes SOAP para invocar al servicio.

Implementación Escenario

En esta sección se describe, paso a paso, la implementación de una Aplicación Cliente Java de escritorio según el escenario descrito previamente.

La implementación del escenario comprende tres etapas:

- Creación del Proyecto Eclipse y Configuración del Entorno

[6] Es la política de autenticación utilizada por AGESIC para la verificación de solicitudes del cliente. Actualmente el único valor posible es "urn:simpletoken)".

[7] Actualmente la PGE acepta la emisión de *tokens* SAML versión 1.1.

- Obtención del *token* de Seguridad emitido por la PGE
- Invocación al Servicio

En las siguientes sub-secciones se describen en detalle estas etapas.

Creación y Configuración del Proyecto Eclipse

La primera etapa consta de la creación de un proyecto Eclipse y su configuración, así como la del entorno de desarrollo. Concretamente en esta etapa se debe: crear un proyecto Eclipse, crear un Runtime para JBossAS e incluir librerías y otros archivos necesarios en el proyecto creado.

Creación del Proyecto Eclipse

Una vez iniciado Eclipse, crear un proyecto de tipo "Faceted Project"[8] incluyendo los facets Java 5.0, JBoss Web Service Core 3.0 y Dynamic Web Module.

Nota: La aplicación Java a implementar no es una aplicación Web, sin embargo, se incluye el *facet* Dynamic Web Module porque es requerido por el *facet* JBoss Web Service Core 3.0.

Los pasos a seguir para realizar esta tarea son:

1. Seleccionar *File* → *New* → *Other* → *General* → *Faceted Project*, crear un nuevo proyecto con el nombre PGEClientTutorial y los facets Java 5.0, JBoss Web Service Core 3.0 y Dynamic Web Module 2.4 según la Figura 26 y Figura 27.

   Nota: Para cada *facet* seleccionado se deben configurar algunos valores que se describen en los siguientes pasos.

[8] Los *Faceted Projects* de Eclipse son proyectos que aceptan unidades de funcionalidad (*facets*) que pueden ser fácilmente agregadas por los usuarios.

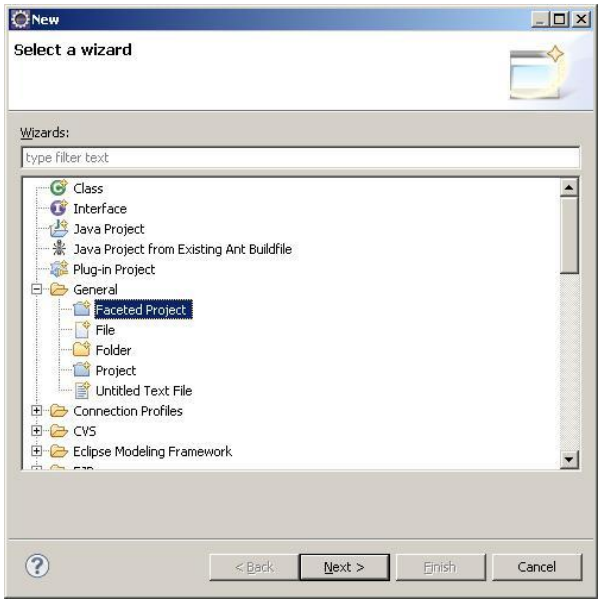

Figura 26 – Crear proyecto Faceted

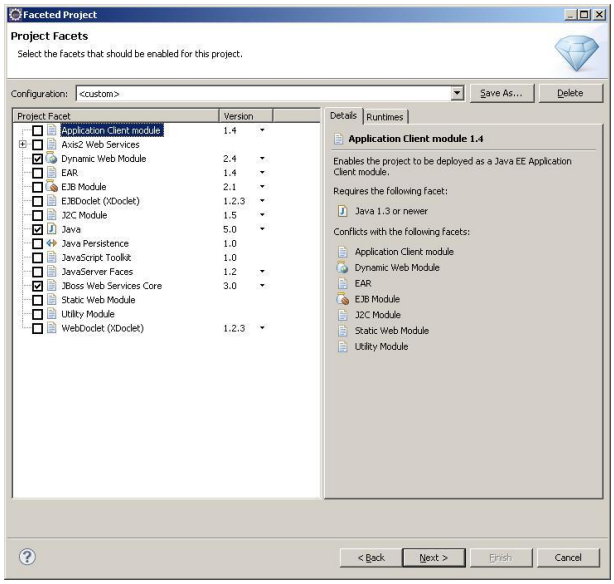

Figura 27 – Opciones Faceted

2. Configurar el *facet* Java. Se debe especificar la carpeta destino del código fuente (src) y compilado (build). Dejar valores por defecto.

3. Configurar el *facet* Dynamic Web Module. Se debe especificar el directorio de contenido Web. Dejar valores por defecto.

4. Configurar el *facet* JBossWS 3.0. Se debe especificar un Web Service Runtime. Para esto seleccionar "*New*" y completar con datos similares a los de la Figura 28. Presionar el botón "*Finish*" y luego seleccionar el *runtime* como se muestra en la Figura 29.

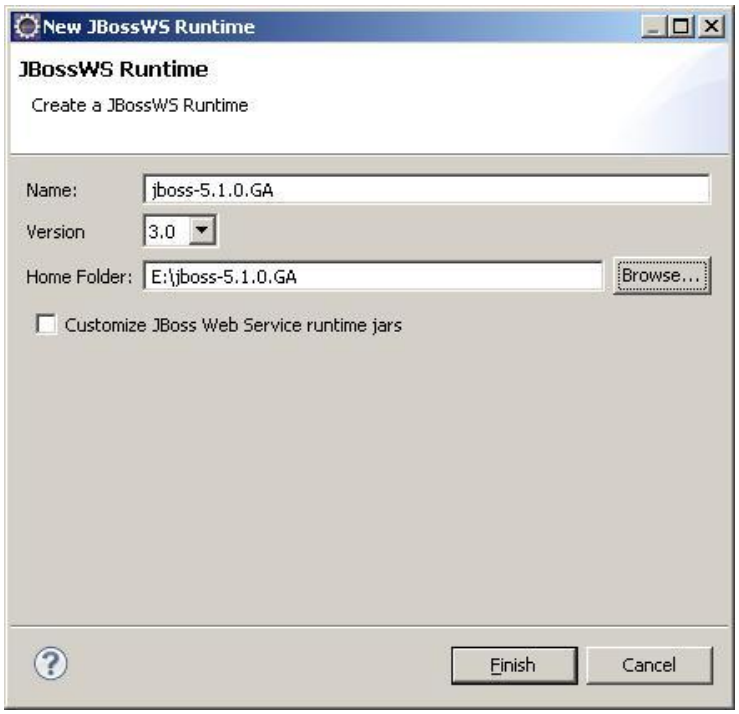

Figura 28 – Crear JBossWS Runtime

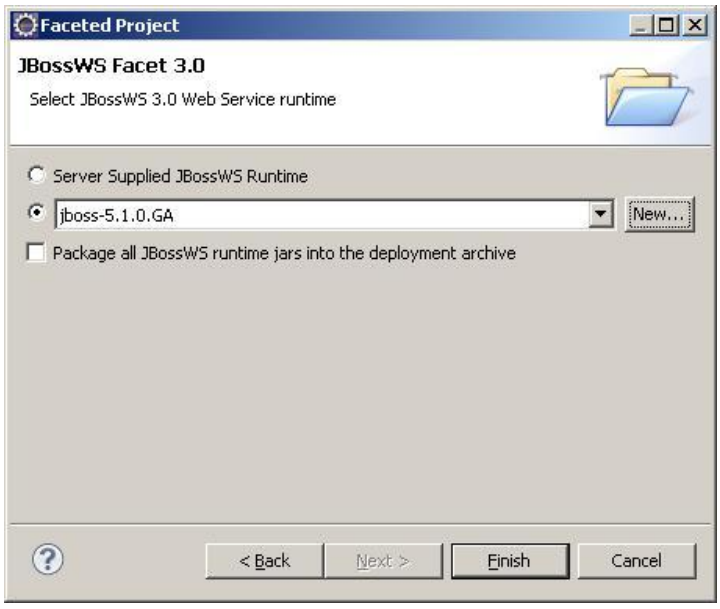

Figura 29 – Seleccionar JBossWS 3.0 Web Service Runtime

Definir un *Server Runtime* para JBoss AS

Esta configuración es necesaria para que funcione correctamente la herramienta de generación de código automática.

Los pasos a seguir para realizar esta tarea son:

1. Seleccionar del menú de Eclipse la opción Windows → Preferences.

2. Buscar la opción Server →Runtime Environments como se muestra en la Figura 30.

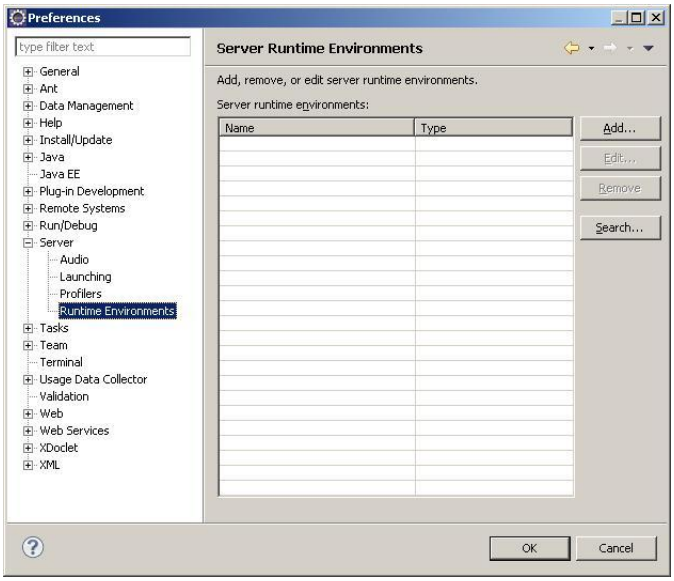

Figura 30 – Configurar JBoss Runtime

3. Seleccionar el botón *Add*... y luego la opción *JBoss Community* → *JBoss 5.1 Runtime* como se muestra en la Figura 31. Se debe alcanzar un resultado similar al de la Figura 32.

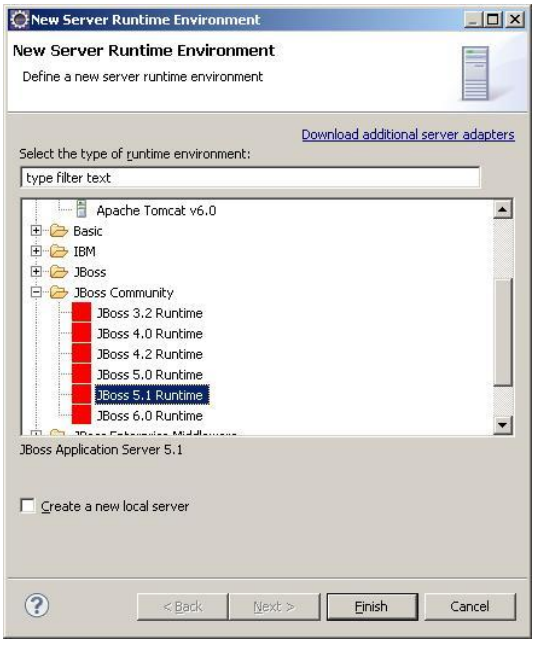

Figura 31 – Configurar JBoss Runtime parte 2

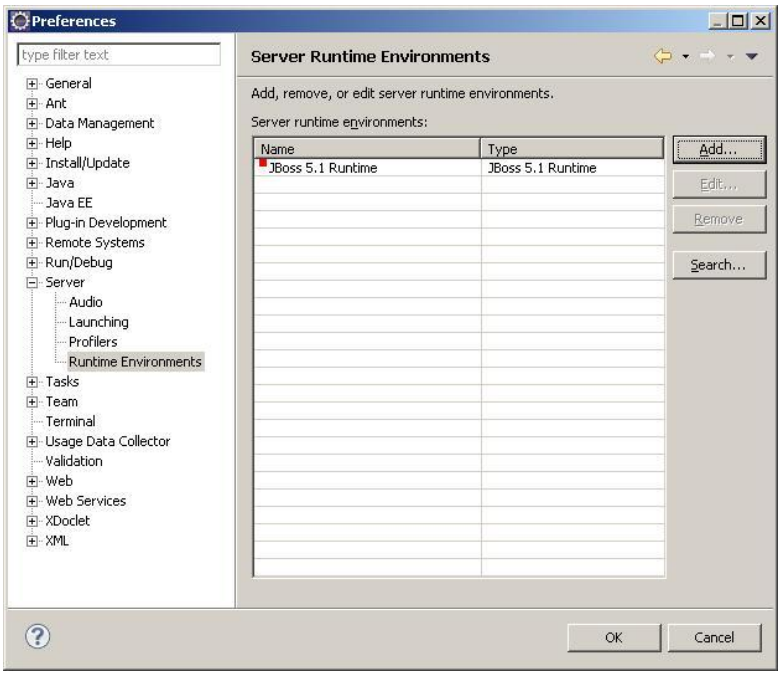

Figura 32 – Configuración completa del JBoss Runtime

Incluir Librerías y Otros Archivos Necesarios

La Aplicación Cliente requiere librerías de JBossWS y OpenSAML para su implementación. Además, requiere la Librería de Ejemplo implementada por AGESIC. Por último, es necesario también incluir en el proyecto el WSDL del servicio Certificado de Nacidos Vivos.

Los pasos a seguir para incluir estos archivos en el proyecto son:

1. Crear la carpeta lib y agregar las bibliotecas de JBossWS - Native, de OpenSAML y de AGESIC.

2. Agregar estas bibliotecas al Java Build Path del proyecto, haciendo clic derecho sobre el mismo y seleccionando *Properties* → *Java Build Path* → *Libraries* → *Add JARs…*

3. Crear la carpeta wsdl y agregar el WSDL del servicio a consumir. Este archivo debe tener extensión .wsdl para ser interpretado correctamente por Eclipse.

Nota: Las librearías requeridas y el WSDL del servicio se pueden obtener en [4].

Obtención del *token* de Seguridad emitido por la PGE

Para realizar esta tarea, se utiliza la Librearía de Ejemplo desarrollada por AGESIC. Los pasos a seguir son los siguientes:

1. Crear el *package* test.

2. Crear la clase PGEClientTest en el *package* test de forma tal que contenga un método *main* como se presenta en la Figura 33.

```
package test;

public class PGEClientTest {

	public static void main(String[] args) {

	}
}
```

Figura 33 – Clase PGEClientTest

Crear en el *main* un RSTBean especificando los datos para enviar el pedido al STS de la PGE, como se muestra en la Figura 34.

```
String userName = "Juan";
String role = "Doctor";
String service = "http://192.168.40.190:9000/Servicio";
String policyName = "urn:tokensimple";
String issuer = "BPS";

RSTBean bean = new RSTBean();
bean.setUserName(userName);
bean.setRole(role);
bean.setService(service);
bean.setPolicyName(policyName);
bean.setIssuer(issuer);
```

Figura 34 – Clase PGEClientTest

3. Como se presenta en la Figura 35, crear dos StoreBeans para almacenar los datos para acceder a los almacenes de claves que contienen los certificados y claves requeridas.

```
String alias = "alias";
String keyStoreFilePath = "c:/...";
String keyStorePwd = "password";

String trustStoreFilePath = "c:/...";
String trustStorePwd = "password";


StoreBean keyStore = new StoreBean();
keyStore.setAlias(alias);
keyStore.setStoreFilePath(keyStoreFilePath);
keyStore.setStorePwd(keyStorePwd);

StoreBean trustStore = new StoreBean();
trustStore.setStoreFilePath(trustStoreFilePath);
trustStore.setStorePwd(trustStorePwd);
```

Figura 35 – Keystore y Truststore

4. Por último, crear un PGEClient e invocar el método requestSecurityToken para obtener el *token* SAML firmado por la PGE, como se muestra en la Figura 36.

```
PGEClient client = new PGEClient();
SAMLAssertion assertionResponse =
client.requestSecurityToken(bean, keyStore, trustStore);
```

Figura 36 – Obtención del *token* SAML firmado por la PGE

Invocación al Servicio

Una vez obtenido un *token* SAML firmado por la PGE, es posible consumir el servicio. Para ello, se envía un mensaje SOAP al Servicio Proxy del servicio Certificado de Nacidos Vivos, que incluya:

- información de negocio según el WSDL del servicio
- servicio y método a invocar (especificados a través de WS-Addressing)
- *token* SAML firmado por la PGE (incluido a través de WS-Security)

En este ejemplo, la invocación al servicio consta de cuatro pasos:

1. Crear las clases para consumir el servicio. A través de estas clases se crea el mensaje SOAP con la información de negocio.

2. Adjuntar en el mensaje SOAP el servicio y método a invocar.

3. Adjuntar en el mensaje SOAP el *token* SAML firmado por la PGE.

4. Consumir el servicio

Crear las clases para consumir el servicio

Para esta tarea se utiliza la herramienta de generación de clientes de Web Services provista por el entorno de desarrollo. Los pasos a seguir son los siguientes:

1. Hacer clic derecho en el archivo wsdl del servicio ubicado en la carpeta wsdl y seleccionar *Web Service* → *Generate Client* como se muestra en la Figura 37.

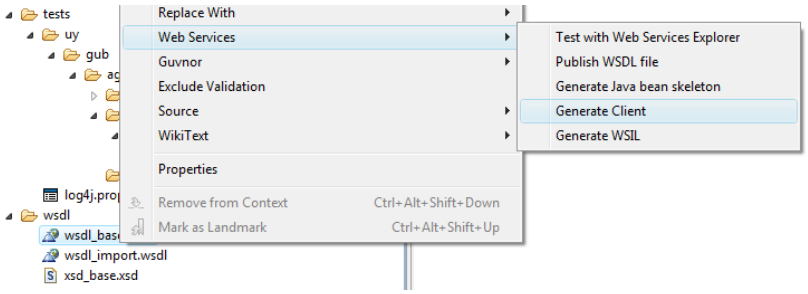

Figura 37 – Generar Clases para Consumir Web Service

2. Seleccionar JBossWS como Web Service Runtime y seleccionar el nivel de generación del cliente como "*Develop Client*", según se muestra en la Figura 38.

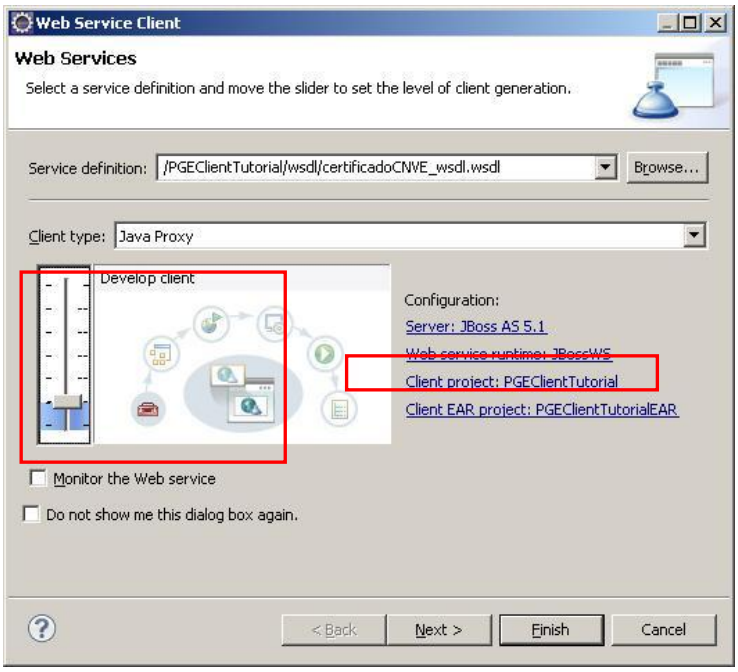

Figura 38 – Crear clases Proxy del servicio

3. Presionar "*Next*" y si se desea, modificar el nombre del paquete donde se colocan las clases generadas.

Al finalizar estos pasos, JBoss Tools genera un conjunto de clases Java, que se muestran en la Figura 39, para llevar a cabo la comunicación con el servicio de la PGE.

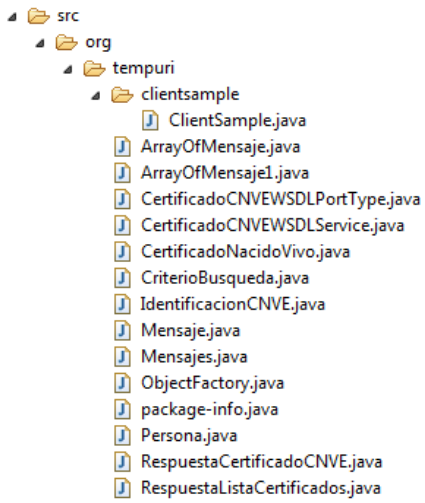

Figura 39 – Clases Generadas para Invocar al Web Service

En particular, como se muestra en la Figura 40, se genera una clase denominada ClientSample que brinda un ejemplo de invocación a los métodos del Web Service.

```
System.out.println("Create Web Service Client..");
CertificadoCNVEWSDLService service1 = new
CertificadoCNVEWSDLService();

System.out.println("Create Web Service...");
CertificadoCNVEWSDLPortType port1 =
service1.getCustomBindingCertificadoCNVEWSDLPortType();

System.out.println("Call Web Service Operation...");
System.out.println("Server said: " +
port1.registrarCNVE(null));
//Please input the parameters instead of 'null' for the
upper method!

System.out.println("Server said: " +
port1.getCertificadosByCriteria(null));
//Please input the parameters instead of 'null' for the
upper method!
```

Figura 40 – Clase ClientSample

Adjuntar en el mensaje SOAP el servicio y método a invocar.

Como se menciona previamente, la PGE requiere que en la invocación al servicio se especifique el servicio y método a invocar, utilizando los cabezales de WS-Addressing "To" y "Action", respectivamente.

La plataforma JBoss (plataforma utilizada para el desarrollo de este tutorial) implementa las especificaciones WS-* siguiendo un diseño basado en *'pipes & filters'* donde un mensaje SOAP pasa por una serie de *pipes* (o *handlers*) y filtros previo a su envío. Además, provee *handlers* prefabricados para poder utilizar las especificaciones WS-* minimizando al máximo las tareas de programación. Concretamente, cuenta con la clase *WSAddressingClientHandler* la cual se encarga de adjuntar los cabezales WS-Addressing al mensaje permitiendo especificarlos a través de las clases *AddressingBuilder* y *SOAPAddressingProperties*.

La Figura 41 presenta cómo utilizar este mecanismo para adjuntar los cabezales WS-Addressing requeridos por la PGE.

```
//Build handler chain
List<Handler> customHandlerChain = new
ArrayList<Handler>();
customHandlerChain.add(new WSAddressingClientHandler());


//Build addressing properties
AddressingBuilder addrBuilder =
SOAPAddressingBuilder.getAddressingBuilder();
SOAPAddressingProperties addrProps =
(SOAPAddressingProperties)addrBuilder.newAddressingProper
ties();

AttributedURI to = new AttributedURIImpl(service);
AttributedURI action = new AttributedURIImpl(method);

addrProps.setTo(to);
addrProps.setAction(action);

//Add bindings to the soap call
CertificadoCNVEWSDLService cnveService = new
CertificadoCNVEWSDLService();
CertificadoCNVEWSDLPortType port =
cnveService.getCustomBindingCertificadoCNVEWSDLPortType()
;

BindingProvider bindingProvider = (BindingProvider)port;
bindingProvider.getRequestContext().put(JAXWSAConstants.C
LIENT_ADDRESSING_PROPERTIES, addrProps);
bindingProvider.getBinding().setHandlerChain(customHandle
rChain);
```

Figura 41 – Agregar los cabezales WS-Addressing al mensaje

Adjuntar en el mensaje SOAP el *token* SAML firmado por la PGE.

Para adjuntar el *token* SAML utilizando WS-Security se procede de forma similar que para adjuntar los cabezales WS-Addressing. Sin embargo, en este caso AGESIC provee un *handler* específico (SAMLHandler) para adjuntar el *token* SAML al mensaje, dado que la plataforma JBoss no provee ninguno prefabricado. La Figura 42 presenta cómo utilizar este mecanismo para adjuntar el *token* SAML requerido por la PGE.

```
//Build handler chain
...
customHandlerChain.add(new SAMLHandler());

...
//Add bindings to the soap call
...
bindingProvider.getRequestContext().put(AgesicConstants.S
AML1_PROPERTY, assertionResponse);
bindingProvider.getBinding().setHandlerChain(customHandle
rChain);
```

Figura 42 –Agregar *token* SAML al mensaje usando WS-Security

El *handler* desarrollado por AGESIC para adjuntar el *token* SAML se puede obtener en [4].

Consumir el Servicio

Por último, se debe consumir el servicio. Para esto se utiliza las clases generadas en el paso "Crear las clases para consumir el servicio", como se muestra en la Figura 43.

```
//Create input
IdentificacionCNVE idCNVE = new IdentificacionCNVE();
Persona mother = new Persona();
mother.setPrimerNombre("Marta");

CertificadoNacidoVivo solicitudCNVE = new
CertificadoNacidoVivo();
solicitudCNVE.setUsuario(userName);
solicitudCNVE.setNumeroCertificado(idCNVE);
solicitudCNVE.setDatosMadre(mother);

//Call the web service
RespuestaCertificadoCNVE response =
port.registrarCNVE(solicitudCNVE);
String code = response.getCodigoRespuesta();

System.out.println("Response code: "+code);
```

Figura 43 –Consumir el Servicio

Para ejecutar el cliente implementado, seleccionar la clase PGEClientTest, hacer clic derecho y ejecutar *Run as* → *Java Application*. La consola debería mostrar un mensaje similar al presentado en la Figura 44.

```
Codigo de respuesta: 01
```

Figura 44 – Mensaje de respuesta del servicio

Referencias

[1] Eclipse. http://www.eclipse.org/ [Accedida en Junio de 2010]

[2] JBoss Tools. http://www.jboss.org/tools [Accedida en Junio de 2010]

[3] JBoss Web Services. http://www.jboss.org/jbossws [Accedida en Junio de 2010]

[4] AGESIC. Recursos para el desarrollo de los tutoriales.

[5] keytool - Key and Certificate Management Tool. http://java.sun.com/j2se/1.5.0/docs/tooldocs/solaris/keytool.html [Accedida en Junio de 2010]

[6] Useful WS-Security Command-Line Tools. http://java.sun.com/webservices/docs/1.6/tutorial/doc/XWS-SecurityIntro6.html#wp526882 [Accedida en Junio de 2010]

Capítulo V

Alta y Consumo de Servicios

Introducción

Este capítulo describe, a nivel técnico, los requerimientos y pasos necesarios para que un organismo provea y consuma servicios en la PGE.

Alta de un Servicio en la PGE

Prerrequisitos

Para que un organismo pueda proveer un servicio en la PGE es necesario que esté conectado a la REDuy. Si no se cuenta con conexión a la REDuy consultar la sección “Conexión con REDuy”.

Implementación, Despliegue y Ejecución del Servicio

Los servicios que proveen los organismos se despliegan y ejecutan principalmente en sus servidores. Si se quiere alojar un servicio en la PGE consultar la sección “Ejecución de un Servicio en la PGE”.

La implementación de los servicios puede realizarse utilizando diferentes tecnologías y plataformas como Java EE, .NET, PHP u otras. Cualquiera sea la tecnología que se utilice, los servicios a publicarse en la PGE deben cumplir con los siguientes requerimientos:

1. deben poder ser invocados mediante el envío, vía HTTPS, de mensajes que se ajusten al estándar SOAP (versión 1.1)

2. deben describirse utilizando el estándar WSDL (versión 1.1)

3. deben ajustarse a los lineamientos especificados en el Basic Profile (versión 1.1) [1] y Basic Security Profile (versión 1.0) [2] de la organización WS-I

4. el WSDL que describe al servicio debe incluir documentación general del servicio y documentación detallada de sus operaciones y parámetros de entrada y salida

Es importante mencionar que los servicios pueden implementarse completamente desde cero o apoyarse en Sistemas Legados existentes, permitiendo así su re-utilización y aprovechamiento.

Completar y Enviar Formulario “Alta de un Servicio”

Para exponer un servicio en la PGE, los organismos deben completar y enviar a AGESIC el “**¡Error! No se encuentra el origen de la referencia.**” que se encuentra en el Apéndice 2. En este formulario se debe especificar información general, ubicación, descripción, e información técnica y de seguridad del servicio. El formulario debe ser enviado por correo electrónico a la dirección soporte@agesic.gub.uy, especificando en el asunto del correo “[Alta de un Servicio en la PGE] *Nombre del Organismo Proveedor*”.

Una vez aprobada la solicitud de alta de servicio, el equipo de soporte de AGESIC configura los *firewalls* de REDuy para habilitar el tráfico de red desde la PGE hacia el servicio, como se explica en la sección “Configuración de Firewalls de REDuy”.

En las próximas sub-secciones se describe la información que se debe incluir en el “**¡Error! No se encuentra el origen de la referencia.**”.

Información General de la Solicitud

En primer lugar, el organismo debe brindarle a AGESIC información general sobre la solicitud de alta del servicio. Es necesario especificar los datos que se presentan y describen en la Tabla 8.

Dato	Descripción
Nombre del Organismo Solicitante	Nombre del organismo que solicita la publicación del servicio en la PGE (ej. AGESIC, BPS, etc.).
Dependencia	Si es una dependencia de un organismo, indicar su nombre.
Fecha de Solicitud	Fecha en que se realiza la solicitud de publicación del servicio.
Nombre del Solicitante	Nombre completo del funcionario que realiza la solicitud.
Correo Electrónico del Solicitante	Correo electrónico de contacto del funcionario que realiza la solicitud.
Nombre del Responsable Técnico	Nombre del responsable técnico del servicio a publicar.
Correo Electrónico del Responsable Técnico	Correo electrónico del responsable técnico del servicio a publicar.

Tabla 8 – Información General de la Solicitud de Alta de Servicio

Información de Conexión

En esta sección del formulario, el organismo debe proporcionar la información necesaria para la conexión del servidor a la PGE. Se deben especificar los datos que se presentan y describen en la Tabla 9.

Dato	Descripción
Nombre del Nodo de REDuy	Nombre otorgado por AGESIC en el momento de la conexión del organismo a REDuy (ej. AGESIC, BPS, etc.).
Dirección IP Interna del Servidor	Dirección IP interna del servidor del organismo que aloja el servicio a publicar. Es el equipo que recibe los pedidos del firewall de REDuy.
Puerto TCP	Puerto TCP del servidor, en donde el servicio atenderá las peticiones de los clientes.

Tabla 9 – Datos de Conexión del Servidor donde está el Servicio

Estos datos son utilizados por AGESIC para habilitar, en los firewalls de REDuy, el tráfico hacia el servidor donde se encuentra desplegado el servicio.

Solicitud de Certificado para Conexión SSL

En esta sección del formulario, el organismo debe especificar si posee el certificado digital necesario para la conexión SSL y en caso de no poseerlo debe realizar el pedido correspondiente, según lo descrito en la sección “Conexiones SSL con la PGE”. Concretamente, se deben indicar los datos que se presentan y describen en la Tabla 10.

Dato	Descripción
¿Posee un certificado digital para dicho servidor?	Indicar si ya fue otorgado un certificado digital para la conexión SSL con el servidor.
Pedido del certificado digital (PKCS#10)	En caso de no poseer certificado, indicar el pedido en formato PKCS#10.

Tabla 10 – Datos de Solicitud del Certificado para la Conexión SSL

Descripción del Servicio

Esta sección del formulario agrupa información relativa al servicio a publicar. Se deben especificar los datos que se presentan y describen en la Tabla 11.

Dato	Descripción
Nombre del Servicio	Nombre del servicio a publicar.
Versión del Servicio	Versión del servicio a publicar.
Nombre del Archivo WSDL	Nombre del archivo descriptor del servicio.
Descripción del Servicio	Descripción general del servicio.
Categorías de Servicio	Categorías del servicio (mínimo 3). Por ejemplo: salud, social, trabajo, general, etc.

Tabla 11 – Datos de Descripción del Servicio

Políticas de Acceso al Servicio

El servicio a publicar por el organismo en la PGE debe especificar las políticas de acceso que define. Las mismas incluyen los perfiles de usuarios que admite, las operaciones a las que cada perfil tiene permitido invocar y cómo se relacionan los perfiles con los roles definidos por los organismos que consumirán dicho servicio.

Perfiles de Usuario

En esta sección del formulario se incluyen los perfiles de usuarios a los que se les permitirá el acceso al servicio. Estos perfiles deben ser definidos por el organismo proveedor del servicio. Se deben especificar los datos que se presentan y describen en la Tabla 12.

Dato	Descripción
Perfil de Usuario	Nombre del perfil de usuario.
Descripción	Descripción del perfil de usuario.

Tabla 12 – Datos de los Perfiles de Usuario Definidos

En la Tabla 13 se presenta un ejemplo de perfiles de usuario definidos para el servicio “Certificados de Nacidos Vivos”.

Perfil de Usuario	Descripción
Admin	Administradores del servicio.
User	Usuarios del servicio.

Tabla 13 – Ejemplo de Definición de Perfiles

Métodos Autorizados por Perfil

Esta sección presenta los métodos del servicio autorizados para cada perfil definido en el punto anterior. Se deben especificar los datos que se presentan y describen en la Tabla 14.

Dato	Descripción
Método del Servicio	Nombre de cada método del servicio
Perfiles autorizados	Perfiles de usuarios autorizados para invocar el método en cuestión

Tabla 14 – Datos de los Métodos Autorizados por Perfil

En la Tabla 15 se presenta un ejemplo de métodos autorizados para los perfiles de usuario definidos en el servicio "Certificados de Nacidos Vivos".

Método del servicio	Perfiles de usuario autorizados
getCertificadosByCriteria	Admin
registrarCNVE	User, Admin

Tabla 15 – Ejemplo de definición de perfiles

Mapeo entre Perfiles de Usuario y Roles Funcionales

En esta sección del formulario se incluyen los métodos del servicio autorizados para cada perfil definido en el punto anterior. Se deben especificar los datos que se presentan y describen en la Tabla 16.

Dato	Descripción
Perfil	Nombre del perfil del usuario definido.
Roles funcionales	Roles funcionales de los clientes, asociados al perfil definido por el servicio.

Tabla 16 – Datos de los Métodos Autorizados por Perfil

En la Tabla 17 se presenta un ejemplo de mapeo entre perfiles de usuario y roles funcionales de los organismos clientes del servicio "Certificados de Nacidos Vivos".

Perfil	Roles Funcionales
Admin	ou=doctor, ou=gerencia de proyectos, o=agesic
User	ou=doctor , ou=prestaciones, o=bps

Tabla 17 – Ejemplo de Mapeo entre Perfiles de Usuario y Roles Funcionales

Configuración Conexión SSL

Como se explica en la sección "Conexiones SSL con la PGE", se debe instalar el certificado raíz de la CA de la PGE en el servidor del organismo donde se aloja el servicio. Además, se deben realizar las tareas de configuración necesarias en el servidor, para posibilitar el establecimiento de la conexión SSL.

Manejo de Invocaciones al Servicio

Como se explica previamente, todas las invocaciones que reciba un servicio de la PGE fueron previamente autenticadas y autorizadas. De esta forma, el único control de acceso que debe realizar el servicio es el de validar el origen de los pedidos.

Asimismo, como se menciona en secciones anteriores, en toda invocación se adjunta un *token* SAML firmado por la PGE, cuya firma es aconsejable que el proveedor valide. Además, el proveedor puede obtener los datos del *token* SAML (por ejemplo, el rol del usuario) para lo que considere necesario.

Comentarios Adicionales

Tecnologías de Implementación

Para seleccionar la tecnología de implementación de los servicios, se pueden tomar en cuenta muchos factores como el ambiente objetivo de instalación y ejecución, experiencia del personal, costos y apoyo de las herramientas de desarrollo. El último punto es importante ya que muchas de las tareas de implementación descriptas en este documento se pueden ver simplificadas dependiendo de la elección realizada.

Ejecución de un Servicio en la PGE

Como se menciona en secciones anteriores, existe la posibilidad de alojar y ejecutar los servicios de los organismos en la PGE, si éstos tienen algún requerimiento que no es posible cumplir en los servidores de los organismos.

En el caso que un organismo quiera solicitar alojamiento en la PGE para un servicio, debe enviar un correo electrónico a la dirección soporte@agesic.gub.uy, especificando en el asunto del correo "[Ejecución de un Servicio en la PGE] *Nombre del Organismo Proveedor*". Se debe brindar además información sobre el servicio a publicar y sobre los motivos por los cuales se requiere su ejecución en la PGE. Una vez recibido el formulario se coordinará una entrevista con el personal de AGESIC.

Los servicios a ejecutarse en la PGE tienen el requerimiento extra de que deben poder desplegarse y ejecutarse en alguno de los entornos de ejecución provistos por la misma.

Consumo de un Servicio en la PGE

Prerrequisitos

Para que un organismo pueda consumir un servicio de la PGE es necesario que tenga conexión a la REDuy.

A su vez, es necesario que se den de alta en el directorio de la PGE, los roles del organismo que consumirán los servicios. Como se explicó previamente cada organismo administrará su rama del árbol de roles.

Completar y Enviar Formulario para el "Consumo de Servicios"

El organismo que quiera consumir servicios de la PGE debe completar y enviar a AGESIC el "**¡Error! No se encuentra el origen de la referencia.**". Dicho formulario requiere información general e información para la conexión del organismo consumidor con la PGE. El formulario debe ser enviado por correo electrónico a la dirección soporte@agesic.gub.uy, especificando en el asunto del correo "[Consumo de Servicios de la PGE] Nombre del Organismo Cliente". En las

próximas sub-secciones se describe la información que se debe incluir en este formulario.

Información General de la Solicitud

En primer lugar, el organismo debe brindarle a AGESIC información general sobre la solicitud para el consumo de servicios. Es necesario especificar los datos que se presentan y describen en la Tabla 18.

Dato	Descripción
Nombre del Organismo Solicitante	Nombre del organismo que solicita el consumo de servicios en la PGE (ej. AGESIC, BPS, etc.).
Dependencia	Si es una dependencia de un organismo, indicar su nombre.
Fecha de Solicitud	Fecha en que se realiza la solicitud de consumo de servicios.
Nombre del Solicitante	Nombre completo del funcionario que realiza la solicitud.
Correo Electrónico del Solicitante	Correo electrónico de contacto del funcionario que realiza la solicitud.
Nombre del Responsable Técnico	Nombre del responsable técnico de la aplicación cliente.
Correo Electrónico del Responsable Técnico	Correo electrónico del responsable técnico de la aplicación cliente.

Tabla 18 – Información General de la Solicitud de Consumo de Servicios

Información de Conexión

En esta sección del formulario, el organismo debe proporcionar la información necesaria para la conexión con la PGE. Se deben especificar los datos que se presentan y describen en la Tabla 19.

Dato	Descripción
Nombre del Nodo de REDuy	Nombre otorgado por AGESIC en el momento de la conexión del organismo a REDuy (ej. AGESIC, BPS, etc.).
Dirección IP interna del cliente	Dirección IP interna de la aplicación cliente.

Tabla 19 – Datos de Conexión para el Cliente

Solicitud de Certificado para Conexión SSL

En esta sección del formulario, el organismo debe especificar si posee el certificado digital necesario para la conexión SSL y en caso de no poseerlo debe realizar el pedido correspondiente, según lo descrito en la

sección “Conexiones SSL con la PGE”. Concretamente, se deben indicar los datos que se presentan y describen en la Tabla 20.

Dato	Descripción
¿Posee un certificado digital para dicho cliente?	Indicar si ya fue otorgado un certificado digital para la conexión SSL.
Pedido del certificado digital (PKCS#10)	En caso de no poseer certificado, indicar el pedido en formato PKCS#10.

Tabla 20 – Datos de Solicitud del Certificado para la Conexión SSL

Configuración Conexión SSL

Como se explica en la sección “Conexiones SSL con la PGE”, se debe instalar el certificado raíz de la CA de la PGE en el servidor o computador donde se aloja la aplicación cliente. Además, se deben realizar las tareas de configuración necesarias para posibilitar el establecimiento de la conexión SSL.

Obtener Descripción del Servicio

La PGE proveerá un Registro UDDI para la búsqueda y descubrimiento de servicios. El resultado de las búsquedas brindará información general de los servicios y la ubicación del WSDL que los describe.

Implementar Aplicación Cliente

Una aplicación cliente en un organismo debe realizar tres pasos para consumir un servicio de la PGE:

1. obtener un *token* de seguridad firmado por el organismo
2. obtener un *token* de seguridad firmado por la PGE
3. invocar al servicio

En las próximas sub-secciones se describen estos pasos, especificando los estándares que se utilizan y la información que se envía y recibe en cada uno de ellos.

Obtener *token* de Seguridad SAML firmado por el Organismo

El primer paso que debe realizar una aplicación cliente para invocar un servicio de la PGE, es obtener un token de seguridad SAML (v 1.1 o 2.0) firmado digitalmente por el organismo. En casos en que el organismo no cuente con una aplicación emisora de tokens, puede utilizar la Librería de Ejemplo implementada por AGESIC para desarrollar una.

Importante: La librería fue desarrollada como prueba de concepto, por lo cual no está garantizada la ausencia de errores, ni fallas de seguridad. No se recomienda entonces utilizarla en producción, sin los resguardos apropiados según las políticas de testing y seguridad de cada organismo.

La emisión de un token SAML por el organismo, implica que el usuario está autenticado y que su información enviada en el token es válida. La firma del token es el mecanismo utilizado para garantizar la autenticidad del pedido e integridad de la información presentada. El token SAML debe incluir la información descrita en la Tabla 21.

Dato	Descripción
Rol	Rol del cliente dentro del organismo. Se debe especificar a través del DN de la entrada en el directorio LDAP del organismo. Por ej.: “ou=medicos, o=msp”.
Usuario	Nombre de usuario que está ejecutando la aplicación. Este atributo es utilizado con fines de auditoría y no participa en los procesos de autenticación y autorización de la PGE. Queda a criterio del organismo el valor a utilizar.

Tabla 21 – Datos a del *token* de Seguridad SAML a emitir por el Organismo Cliente

En el Apéndice 3 se presenta un ejemplo simplificado de un token SAML generado y firmado por un organismo, mostrando cómo ubicar los elementos presentados en la Tabla 21.

Obtener un *token* de Seguridad SAML firmado por la PGE

El segundo paso que debe realizar una aplicación cliente, es obtener un token de seguridad firmado por la PGE. Para ello debe realizar una solicitud al STS de la PGE utilizando el estándar WS-Trust (versión 1.3). La Tabla 22 describe los datos que debe incluir la solicitud.

Dato	Descripción
Token SAML	*Token* de seguridad SAML (versión 1.1 o 2.0) con información del cliente y firmado por el organismo.

PolicyName	Política de autenticación utilizada por AGESIC para la verificación de solicitudes del cliente. Los posibles valores para este atributo son definidos por la AGESIC. Actualmente el único valor posible es "urn:simpletoken)".
Tipo de *token* a solicitar	Este valor indica el tipo de *token* que se solicita. Actualmente la PGE acepta la emisión de *tokens* SAML versión 1.1.
Servicio	Dirección lógica del servicio de la PGE a consumir.

Tabla 22 – Datos a incluir en el RST

En el Apéndice 3 se presenta un ejemplo simplificado de un mensaje RST, mostrando cómo ubicar los elementos presentados en la Tabla 22 mediante el uso del estándar WS-Trust.

El STS verifica la firma del token SAML y la existencia del rol en la PGE y emite un token de seguridad SAML firmado por la PGE con los datos presentados en la Tabla 23.

Dato	Descripción
Rol	Rol del cliente dentro de la PGE. Se debe especificar a través del Distinguished Name (DN) de la entrada en el directorio LDAP de la PGE. Por ejemplo: "ou=medicos, o=msp, ……,c=uruguay.
Usuario	Nombre de usuario. Este atributo es utilizado con fines de auditoría y no participa en los procesos de autenticación y autorización de la PGE.
PolicyName	Política de autenticación utilizada por AGESIC para la verificación de solicitudes del cliente.
Servicio	Dirección lógica del servicio de la PGE a consumir.

Tabla 23 – Datos incluidos en el token emitido por el STS

En el Apéndice 3 se presenta una versión simplificada de este token, mostrando la ubicación de los elementos de la Tabla 23.

Invocar al servicio

Por último, para invocar un servicio de la PGE, la aplicación cliente debe enviar un mensaje SOAP al Servicio Proxy del servicio con la siguiente información:

1. *token* SAML emitido por el STS de la PGE, especificado a través del estándar WS-Security, versión 1.1.

2. servicio y método a consumir, especificados a través de estándar WS-Addressing versión 1.0.

3. información de negocio de acuerdo al WSDL del servicio

En el Apéndice 3 se presenta un ejemplo simplificado de un mensaje SOAP para consumir el servicio "Certificado de Nacidos Vivos" utilizando los estándares WS-Addressing y WS-Security con la información mencionada anteriormente.

Comentarios Adicionales

Tecnologías de Implementación

Muchas de las tareas de implementación para el consumo de servicios se pueden ver simplificadas dependiendo de las herramientas de desarrollo que se utilicen.

Referencias

[1] Basic Profile. http://www.ws-i.org/Profiles/BasicProfile-1.1.html [Accedida en Mayo de 2010]

[2] Basic Security Profile. http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0.tml [Accedida en Mayo de 2010]

Apéndice I

Marco Técnico

Introducción

En esta sección se brindan algunos conceptos necesarios para la comprensión de este documento, concretamente en los temas Seguridad, Web Services y Arquitecturas Orientadas a Servicios.

Seguridad

Conceptos Básicos

Autenticación

Es el proceso de verificar una identidad [1]. Existen tres tipos principales de evidencias que una entidad puede presentar para probar su identidad [2]:

- Algo que conoce (nombre de usuario, contraseña, respuestas a preguntas, etc.)
- Algo que tiene (*token* físico, etc.)
- Algo que es (huella dactilar, escaneo de retina, etc.)

Confidencialidad de datos

Es la propiedad de que la información sensible no es revelada a individuos, entidades o procesos no autorizados. [3]

Los datos intercambiados a través de una red deben ser protegidos para evitar que usuarios no autorizados puedan tener acceso a los mismos. La técnica estándar para asegurar la confidencialidad de los datos intercambiados es la encriptación. [2]

Integridad de datos

Es la propiedad de que los datos no han sido alterados de manera no autorizada. La integridad de datos refiere tanto a los datos que están almacenados, o que están siendo procesados o transmitidos. [3]

No Repudio

Esta característica permite asegurar que el emisor no puede repudiar, o negar, que ha enviado determinada información o mensaje. [2]

Autorización

Una vez que un usuario ha sido autenticado, una aplicación necesita determinar si está autorizado a acceder a la funcionalidad que está solicitando. A la Autorización se le conoce comúnmente también como Control de Acceso. La decisión de otorgar acceso puede depender de múltiples criterios, como la acción solicitada, el recurso sobre el que se aplica y los grupos o roles a los que pertenece el usuario. [2]

Roled Based Access Control (RBAC)

Control de Acceso Basado en Roles (Role Based Access Control, RBAC) [4][5] es un modelo para el control de acceso en el cual los permisos son asociados a roles y los usuarios adquieren permisos al pertenecer a esos roles. La principal motivación de RBAC es facilitar las tareas administrativas.

Claims y Token de Seguridad

Una *claim* es una afirmación acerca de un sujeto hecha por él o por otro sujeto. Toda *claim* tiene un tipo y un valor. Por ejemplo, el tipo de una *claim* puede ser “Nombre” y el valor “Juan”. Las *claims* se empaquetan en *tokens* de seguridad que son distribuidos por su emisor. [1]

Criptografía

La criptografía es la disciplina que engloba los principios, medios y técnicas para la transformación de datos con el fin de ocultar su contenido, impedir su modificación sin ser detectada e impedir su uso no autorizado. [6]

La criptografía permite cifrar (encriptar) y descifrar (desencriptar) mensajes, permitiendo que sólo usuarios autorizados puedan leerlos. La encriptación es el proceso en el cual el mensaje original (texto plano) se transforma en un mensaje codificado (texto cifrado o encriptado) a través de un algoritmo de codificación y una clave de encriptado. [7]

En la criptografía simétrica, o de clave secreta, tanto emisor como receptor utilizan la misma clave. Sin embargo, en la criptografía asimétrica, o de clave pública, se utiliza una pareja de claves (pública y privada). La clave pública se distribuye a cualquier usuario que la requiera, mientras que la clave privada debe conocerla únicamente su dueño.

Firma Digital

Una Firma Digital es una Firma Electrónica que permite validar la identidad del emisor de una transmisión digital y asegurar que la información transmitida no ha sido alterada por personas no autorizadas [9]. Las firmas digitales están basadas en criptografía de clave pública.

Certificados Digitales

Un Certificado Digital es una tecnología similar a una tarjeta de identificación, que certifica la identidad y autenticidad de su dueño y por lo tanto puede ser utilizado para verificar la identidad de una entidad (individuo, servicio, sistema, compañía, etc.) con la cual se está estableciendo una comunicación electrónica. [9]

Los certificados son emitidos y firmados digitalmente por Autoridades Certificadoras (CAs). Además de los datos de identificación y una copia de la clave pública del dueño, el certificado contiene un número serial, la identidad y firma digital de la CA emisora y la fecha de expiración.

Infraestructura de Clave Pública (Public Key Infraestructure, PKI)

Infraestructura que permite la utilización de criptografía de clave pública en un entorno corporativo o público. Consiste esencialmente en un grupo de servicios que posibilita la generación, almacenamiento y transmisión segura de parejas de claves.

Estos servicios coordinados generalmente incluyen una autoridad certificadora de confianza (CA), una autoridad de registro para aceptar pedidos de certificados digitales, un almacén de certificados en donde los usuarios pueden acceder a las claves públicas de otros usuarios, un certificado digital y un sistema para generar, almacenar y transmitir de forma segura certificados y parejas de claves a las entidades que los solicitan. [9]

Public Key Cryptography Standards (PCKS)

Public Key Cryptography Standards (PCKS) es un conjunto de estándares para criptografía de clave pública. Fueron desarrollados por un consorcio de la industria liderados por RSA Laboratories y especifican cómo un

sistema criptográfico de clave pública debería ser implementado y operado. [9]

PKCS #10 (Certification Request Syntax Standard)

Define una sintaxis para solicitar certificados digitales. El pedido es un archivo de texto con un formato definido por el estándar, codificado en base 64.

PKCS #12 (Personal Information Exchange Syntax Standard)

Especifica un formato portable para almacenar y transportar certificados y claves privadas, entre otros datos.

Secure Socket Layer

Secure Sockets Layer (SSL) es un protocolo de seguridad a nivel de capa de transporte utilizado en Internet para asegurar comunicaciones. Provee autenticación, confidencialidad e integridad. [9]

SSL emplea encriptación de clave pública para asegurar la autenticación y de clave simétrica para la encriptación de información transmitida. La integridad del mensaje es garantizada al incorporar un mecanismo de chequeo de integridad llamado Código de Autenticación de Mensaje (Message Autenthication Code, MAC)

HTTPS

Hypertext Transfer Protocol Secure (HTTPS) es una versión segura del protocolo HTTP. Es esencialmente una combinación de HTTP y el protocolo SSL. [9]

HTTPS está basado en el sistema criptográfico de clave pública y permite cifrar información transmitida a través de Internet. Para ejecutar HTTPS en un servidor Web, es necesario instalar un certificado digital en el mismo.

Servicios de Directorio LDAP

Lightweight Directory Access (LDAP) es un protocolo estándar para acceder a información almacenada en un directorio. Un cliente se puede conectar a un servicio de directorio compatible con LDAP (o X.500) para agregar, eliminar, modificar y buscar información, siempre y cuando tenga los suficientes privilegios para hacerlo. [9]

LDAP fue desarrollado por la IETF y la versión LDAPv3 está definida en el RFC 2251. Está diseñado para ejecutar sobre el *stack* de TCP/IP operando como un protocolo cliente/servidor.

Security Assertion Markup Language (SAML)

SAML [8] define un *framework* para intercambiar información de autenticación y autorización entre dominios de seguridad[9] en la forma de *assertions*, en lugar de utilizar identidades o *tokens* de seguridad tradicionales. [2]

Una *assertion* es una sentencia de código que un servicio SAML crea en respuesta al término exitoso de un evento. Los tipos más importantes de *assertions* son:

- *Assertion* de Autenticación: El sujeto en cuestión fue autenticado de alguna forma, en determinado momento
- *Assertion* de Atributo: El sujeto en cuestión posee información adicional de atributos que puede ser útil para un pedido
- *Assertion* de Decisión de Autorización: El sujeto en cuestión debería tener acceso a un recurso especificado

Firewalls

Un *firewall* es una infraestructura de seguridad, ubicada entre redes para separar y proteger de forma lógica la privacidad e integridad de las comunicaciones entre las mismas, y como protección contra usos maliciosos. Los *firewalls* examinan el tráfico de mensajes que entran y salen de una organización, bloqueando el acceso de los mensajes no autorizados de acuerdo a las reglas de acceso definidas en ellos. [7]

En particular, los Firewalls XML operan en la capa de aplicación o sobre ella (en el stack TCP/IP tradicional) e inspeccionan el contenido XML antes que los mensagen pasen a la aplicación o clientes. Están diseñados para abordar ataques comunes que pueden ser transportados vía XML. [10]

[9] Un dominio de seguridad es una colección de sistemas que tienen una política de seguridad única y un sólo conjunto de controles de seguridad.

Arquitecturas Orientadas a Servicios

La Computación Orientada a Servicios (Service Oriented Computing, SOC) es un paradigma de computación que utiliza servicios como elementos fundamentales para dar soporte al desarrollo rápido, y de bajo costo, de aplicaciones distribuidas en ambientes heterogéneos. [11]

Los Servicios son entidades de software autónomas, auto-contenidas e independientes de la plataforma. Los Servicios proveen funcionalidades de negocio, tienen una interfaz pública y pueden ser descubiertos, invocados y combinados de forma dinámica.

La puesta en práctica del paradigma SOC requiere la implementación de Arquitecturas Orientadas a Servicios (Service Oriented Architectures, SOAs). Una SOA es una forma lógica de diseñar un Sistema de Software para proveer servicios, a usuarios finales, aplicaciones u otros servicios, a través de interfaces públicas que pueden ser descubiertas. Para esto es necesario contar con tecnologías de middleware que permitan el descubrimiento, utilización y combinación de servicios interoperables, para dar soporte a los procesos de negocio de las empresas. [11][12]

Los tres roles principales que pueden encontrarse en una SOA son: Proveedor de Servicios, Registro de Servicios y Consumidor de Servicios.

El Consumidor de Servicios (también denominado cliente) es una aplicación, módulo de software, o servicio, que busca servicios en el registro y ejecuta las funcionalidades de negocio que estos proveen. Por otro lado, el Proveedor de Servicios es una entidad, accesible en la red, que acepta y ejecuta pedidos de los consumidores. El proveedor publica sus servicios en el registro para que los consumidores puedan descubrirlos e invocarlos. Por último, el Registro de Servicios brinda mecanismos para que los consumidores interesados puedan buscar y descubrir servicios. [13]

Las SOAs facilitan muchas de las tareas del desarrollo de aplicaciones empresariales distribuidas, como su integración, la implementación de procesos de negocios y el aprovechamiento de sistemas legados. Además, las SOAs proveen la flexibilidad y agilidad que requieren los usuarios de negocio, permitiéndoles definir servicios de alta granularidad que pueden ser combinados y reutilizados para abordar necesidades de negocio actuales y futuras.

Web Services

Un Web Service es una aplicación de *software* identificada por una URI, cuyas interfaces y formas de acceso pueden ser definidas, descriptas y

descubiertas como artefactos XML, y soporta la interacción directa con otros componentes de software utilizando mensajes basados en XML que son intercambiados a través de protocolos basados en internet. [14]

Actualmente, los Web Services son el principal mecanismo de integración de aplicaciones tecnológicamente heterogéneas, y constituye la tecnología preferida para la implementación de SOAs.

Los Web Services se apoyan en un conjunto de estándares, varios de los cuales se describen en esta sección.

Estándares Básicos: SOAP, WSDL y UDDI

La tecnología de Web Services estuvo inicialmente basada en tres estándares principales: Simple Object Access Protocol (SOAP) [15], Web Services Description Language (WSDL) [16] y Universal Description Discovery and Integration (UDDI) [17].

Simple Object Access Protocol (SOAP)

SOAP es una especificación de la W3C que define un protocolo de comunicación basado en XML para intercambiar mensajes entre computadores, sin importar su sistema operativo o entorno de programación. SOAP es actualmente el estándar de facto utilizado por los Web Services. [7]

SOAP especifica un formato de mensaje, en el cual los mensajes son definidos como SOAP *Envelopes* y están compuestos por un conjunto de cabezales (*header*) y un cuerpo (*body*). Los cabezales tienen como propósito alojar información referente a infraestructura y direccionamiento, mientras que el cuerpo del mensaje aloja información específica del negocio.

Web Services Description Language (WSDL)

WSDL es una especificación de la W3C que define un lenguaje basado en XML para describir de forma estándar a un Web Service. Esta descripción puede incluir las operaciones que ofrece el Web Service, los protocolos de mensajería XML soportados, información de tipos de datos para los mensajes, información acerca del transporte específico a utilizar e información de direccionamiento para localizar al Web Service. [7]

Los documentos WSDL se componen de dos grandes partes: una descripción abstracta y una descripción concreta. La descripción abstracta representa la interfaz del Web Service y no hace referencia a tecnología o protocolo. Por otro lado, la descripción concreta consiste en la definición

de protocolos de comunicación y transporte que permiten la ejecución de las operaciones definidas en la interfaz del servicio.

La principal ventaja de esta separación, abstracta y concreta, es la posibilidad de definir la interfaz de un servicio con sus operaciones y mensajes de forma independiente de los protocolos de transporte, ubicaciones y tecnologías, los cuales podrían cambiar más frecuentemente a lo largo del tiempo.

Universal Description, Discovery and Integration (UDDI)

UDDI (Universal Description, Discovery and Integration) es una especificación de la OASIS que tiene como propósito principal, facilitar la categorización, descubrimiento y recuperación de Web Services. Los registros UDDI permiten categorizar los servicios de acuerdo al tipo de negocio, al tipo de servicio o a las relaciones que tienen con otros servicios. A su vez, define una API basada en Web Services que permite la búsqueda, recuperación y publicación de servicios.

WS-Addressing

WS-Addressing [18] es un conjunto de especificaciones recomendadas por la W3C, que definen un mecanismo independiente del medio de transporte para referenciar Web Services y "direccionar" mensajes. Dentro de este conjunto de especificaciones se encuentran: WS-Addressing - Core, WS-Addressing - SOAP Binding y WS-Addressing - Metadata.

WS-Addressing Core es la principal de las especificaciones ya que define qué es un *Endpoint Reference* (EPR), qué son los *Messages Headers* y cómo pueden ser usados para referenciar Web Services o ser útiles para el direccionamiento de mensajes. Una de las principales características de esta especificación es que sus definiciones son totalmente abstractas e independientes del medio de transporte, por lo cual es posible utilizarlas en conjunto con HTTP, SMTP o cualquier otro protocolo.

Un EPR está compuesto por una dirección, parámetros secundarios y *metadata*. Por otro lado, los *Message Headers* o *Message Addressing Properties* son un conjunto de propiedades que permiten el direccionamiento de mensajes dando soporte a varios patrones de comunicación para Web Services. Algunas de las propiedades definidas por WS-Addressing Core son:

- To: Es una IRI absoluta que representa la dirección del receptor del mensaje.

- From: Es un EPR que indica quién envió el mensaje.
- ReplyTo: Es un EPR que indica a dónde se deben enviar las respuestas del mensaje.
- FaultTo: Es un EPR que indica a dónde se deben enviar los errores relacionados al mensaje enviado.
- Action: Es una IRI absoluta que identifica de forma unívoca la semántica del mensaje.
- MessageID: Es una IRI absoluta que identifica de forma unívoca el mensaje.

Por otro lado, la especificación WS-Addresssing – SOAP Binding define cómo utilizar las propiedades, definidas de forma abstracta en el Core, en mensajes SOAP.

Por último, WS-Addressing – Metadata, indica cómo describir las propiedades utilizando un WSDL, cómo incluir *metadata* del WSDL en un *Endpoint Reference* y cómo utilizar WS-Policy para indicar que un Web Service soporta WS-Addressing.

WS-Security

WS-Security [23] es un estándar de la OASIS que propone un conjunto estándar de extensiones a SOAP que pueden ser utilizadas para enviar *tokens* de seguridad como partes del mensaje SOAP y proveer integridad y confidencialidad del contenido de los mismos [7]. WS-Security describe principalmente cómo incluir *tokens* de seguridad en mensajes SOAP y cómo utilizar XML Encryption y XML Signature para cifrar y firmar esos *tokens*, así como otras partes del mensaje.

WS-Security define un conjunto de *security tokens* y el mecanismo para adjuntarlos, identificarlos y referenciarlos dentro de un mensaje. El *UserNameToken*, por ejemplo, permite especificar un nombre de usuario y opcionalmente una contraseña. A su vez, el *BinarySecurityToken* permite incluir certificados X.509 o tickets Kerberos. Finalmente, a través de los *XMLTokens*, es posible adjuntar *security tokens* basados en XML utilizando distintos formatos como Security Assertion Markup Language (SAML) y eXtensible Rights Markup Language (XrML).

WS-Security provee un modelo de seguridad extensible que brinda soporte a múltiples *tokens* de seguridad, dominios de confianza, firmas digitales y algoritmos de encriptación, que junto con un modelo de seguridad específico (PKI, Kerberos, SAML, etc), permite lograr una solución completa y segura de punta a punta.

WS-Security versus SSL

Dado que HTTP es el principal medio utilizado para el transporte de mensajes SOAP, el uso de HTTPS (HTTP sobre SSL) es una clara alternativa para proveer propiedades de seguridad al intercambio de dichos mensajes. Sin embargo, SSL por sí sólo no permite proveer la protección necesaria para algunos escenarios de uso de Web Services.

Concretamente, la tecnología de Web Services prevé la existencia de intermediarios, que se sitúan entre el cliente y el Web Service destino. Los intermediarios pueden realizar tareas de ruteo, auditoría, transformación y validación de mensajes. SSL protege el mensaje únicamente cuando está siendo transmitido, pero no cuando llega a la capa de aplicación. De esta forma, un servicio o aplicación intermediaria podría examinar o modificar el mensaje antes de transmitirlo al destinatario final. Por otro lado, SSL no permite cifrar sólo partes del mensaje SOAP, como lo permite WS-Security.

WS-Trust

WS-Trust [24] es un estándar de la OASIS que extiende WS-Security y define un modelo de seguridad para Web Services basado en relaciones de confianza y mecanismos de emisión, renovación y validación de *tokens* de seguridad.

El modelo de seguridad definido por WS-Trust se basa en un proceso por el cual los servicios requieren que todo mensaje recibido contenga un conjunto de *claims* para ser aceptado. Todo mensaje que no las tenga, es rechazado. Los servicios generalmente utilizan WS-Policy para especificar qué *claims* requieren.

En casos en los cuales los clientes no tengan posesión de las *claims* para consumir un servicio, pueden contactar a autoridades apropiadas para así obtenerlas. Dentro de la especificación WS-Trust, se conoce a estas autoridades como los *Security Token Services* (STS). Los clientes pueden contactar a los STS para que éstos les emitan los *claims* y así poder consumir el servicio. En determinados casos puede ocurrir que los STS tengan definidas políticas de seguridad, siendo necesario presentar un conjunto de *claims* ante ellos.

Enterprise Service Bus

Un Enterprise Service Bus (ESB) es una plataforma de integración basada en estándares que combina mensajería, Web Services, transformaciones de datos y ruteo inteligente para conectar y coordinar,

de forma confiable, la interacción de diversas aplicaciones con integridad transaccional. [21]

Si bien la tecnología de Web Services constituye una base sólida para la implementación de SOAs, existen algunos aspectos, como su naturaleza punto a punto, que pueden afectar la flexibilidad y escalabilidad de soluciones que usan exclusivamente esta tecnología.

En este contexto, el ESB brinda una capa de integración intermedia que provee lógica de integración y comunicación reutilizable, para posibilitar la interacción entre clientes y servicios en una SOA. El ESB acepta pedidos en la forma de mensajes, sobre los cuales se pueden realizar distintas operaciones de mediación [20][21] para solucionar heterogeneidades entre clientes y servicios. A modo de ejemplo, el ESB puede interactuar con servicios a través de protocolos de comunicación no soportados por los clientes, o puede transformar los mensajes enviados por el cliente para que se ajusten al formato que espera el servicio.

La utilización de un ESB promueve el bajo acoplamiento entre clientes y servicios, dividiendo la lógica de comunicación e integración en pequeñas piezas de software administrables. Permite además tener una clara separación entre la lógica de integración y comunicación, y la lógica de negocio implementada por los servicios.

Las principales funcionalidades que generalmente proveen los productos de tipo ESB son: la transformación de protocolos de comunicación, la transformación de mensajes, el ruteo de mensajes, el enriquecimiento de mensajes y mecanismos de mensajería confiable. Asimismo, en general también cuentan con funcionalidades que permiten garantizar propiedades de calidad de servicio, como performance, disponibilidad y seguridad.[19][22][21]

Referencias

[1] D. Baier, V. Bertocci, K. Brown, M. Woloski, and E. Pace, A Guide to Claims-Based Identity and Access Control, Microsoft Press, 2010.

[2] Ramarao Kanneganti, Prasad Chodavarapu. SOA Security. Manning. 2008.

[3] NIST. Glossary of Key Information Security Terms. http://csrc.nist.gov/publications/nistir/NISTIR-7298_Glossary_Key_Infor_Security_Terms.pdf [Accedida en Junio de 2010]

[4] R.S. Sandhu, E.J. Coyne, H.L. Feinstein, and C.E. Youman, “Role-Based Access Control Models,” IEEE COMPUTER, vol. 29, 1996, pp. 38--47.

[5] R. Sandhu, D. Ferraiolo, and R. Kuhn, “The NIST Model for Role-Based Access Control: Towards A Unified Standard,” IN PROCEEDINGS OF THE FIFTH ACM WORKSHOP ON ROLE-BASED ACCESS CONTROL, 2000, pp. 47--63.

[6] W. Fumy and J. Sauerbrey, Enterprise security: IT security solutions : concepts, practical experiences, technologies, Wiley-VCH, 2006.

[7] M. Papazoglou, Web Services: Principles and Technology, Prentice Hall, 2007.

[8] SAML Specifications. http://saml.xml.org/saml-specifications [Accedida en Junio de 2010]

[9] I. Tulloch, Microsoft Encyclopedia of Networking, Microsoft Press, 2002.

[10] A. Belapurkar, A. Chakrabarti, H. Ponnapalli, N. Varadarajan, and S. Padmanabhuni, Distributed systems security: issues, processes, and solutions, John Wiley and Sons, 2009.

[11] Dimitrios Georgakopoulos, M. P. Papazoglou. Service-Oriented Computing. The MIT Press. 2009

[12] M. P. Papazoglou, P. Traverso, S. Dustdar, F. Leymann. "Service-oriented computing: State of the art and research challenges". 2007.

[13] Mark Endrei, Jenny Ang, Ali Arsanjani, Sook Chua, Philippe Comte, Pal Krogdahl, Min Luo, Tony Newling. Patterns: Service-Oriented Architecture and Web Services. IBM Redbooks. 2004.

[14] W3C. Web Services Description Requirements. http://www.w3.org/TR/ws-desc-reqs/#definitions [Accedida en Junio de 2010]

[15] Simple Object Access Protocol. http://www.w3.org/TR/soap/ [Accedida en Junio de 2010]

[16] Web Services Description Language. http://www.w3.org/TR/wsdl [Accedida en Junio de 2010]

[17] Universal Description Discovery and Integration. http://www.oasis-open.org/committees/wsrp/specifications/version1/wsrp-pfb-uddi-tn-1.0.pdf [Accedida en Junio de 2010]

[18] Web Services Addressing Working Group. http://www.w3.org/2002/ws/addr/ [Accedida en Junio de 2010]

[19] T. Rademakers and J. Dirksen, Open-Source ESBs in Action. Manning Publications, October 2008.

[20] Colombe Hérault, Gael Thomas, and Philippe Lalanda. Mediation and Enterprise Service Bus: A position paper. 2005.

[21] M. T. Schmidt, B. Hutchison, P. Lambros, R. Phippen. "The enterprise service bus: making service-oriented architecture real" IBM Syst. J., vol. 44, no. 4, pp. 781-797, 2005.

[22] W. Roshen, SOA-Based Enterprise Integration: A Step-by-Step Guide to Services-based Application, 1st ed. McGraw-Hill Osborne Media. 2009.

[23] WS-Security 1.1. http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf [Accedida en Mayo de 2010]

[24] WS-Trust 1.3. http://docs.oasis-open.org/ws-sx/ws-trust/200512/ws-trust-1.3-os.html [Accedida en Junio de 2010]

[25] Dave Chappell. Enterprise Service Bus. O'Reilly. 2004.

Apéndice II

Formularios de Alta y Consumo de Servicios

Solicitud para un uso particular de la PGE (v.1.0)

Fecha Solicitud:		
Nombre Organismo solicitante:		
Contacto en Organismo:	Nombre:	
	Mail:	
	Teléfono:	
Tipo de uso solicitado:		
1) Publicación de servicio: ☐		
2) Consumo de servicio: ☐		
3) Otro uso: ☐		
Información General		
1) Publicación de Servicio		
Nombre del Servicio		
Descripción del Servicio		
Dirección Física (url)		
WSDL		
Descripción de Operaciones		
Roles Autorizados a consumir métodos	Método 1	
	Roles 1	
	Método 2	
	Roles 2	
Contactos para Soporte del servicio:		
2) Consumo de servicio:		

Nombre del servicio a consumir:	
3) Otro uso	
Uso específico:	
Prestaciones de la Plataforma	
Middleware	
Mecanismos de comunicación	Sincrónico/Asincrónico
Confiabilidad de mensajes	Si/No
Transformación de datos	Si/No
Balanceo de Carga	Si/No
Transferencia de datos binarios	Si/No (tamaño promedio datos)
Control de Acceso	
Tipo de token	Estándar./Larga Duración
Autorización	El sistema utiliza la PGE como complemento de los sistemas de seguridad existentes (definición de roles) / No Aplica
Cifrado de la información	Transporte SSL : Si/No Mensaje WS-Security: Si/No
Integridad de la información	Transporte SSL : Si/No Mensaje WS-Security: Si/No
Inclusión de fecha de creación del mensaje	Si/No
Metadatos	
Usa Metadatos específicos de PGE	Si (cuales ?) / No
Usa Metadatos no definidos en la PGE	Si (cuales ?) / No
Resolución (reservado para AGESIC, no llenar):	Fecha: ..../..../..... Afirmativa: Negativa: Causas:.....

Apéndice III

Ejemplos

Ejemplo de Token de Organismo

La Figura 45 presenta un ejemplo simplificado de un *token* SAML generado y firmado por un organismo, mostrando cómo ubicar los datos necesarios mediante el uso de *AuthenticationStatements* y *AtributeStatements*.

```
<saml1:Assertion xmlns:saml1="..." AssertionID="..."
       IssueInstant="2010-04-22T21:21:10.156Z"
       Issuer="Agesic" MajorVersion="1" MinorVersion="1">
       <saml1:AuthenticationStatement
              AuthenticationInstant="2010-04-22T21:21:10.062Z"
              AuthenticationMethod="...">
             <saml1:Subject>
                    <saml1:NameIdentifier Format="…">          Rol del Usuario
                                 o=agesic, ou=certUy, ou=admin
                    </saml1:NameIdentifier>
                     ...
       </saml1:AuthenticationStatement>

       <saml1:AttributeStatement>
              <saml1:Subject>
                    <saml1:NameIdentifier Format="...">        Rol del Usuario
                          o=agesic, ou=certUy, ou=admin
                     </saml1:NameIdentifier>
                    ...
              </saml1:Subject>
              <saml1:Attribute AttributeName="User"
                    AttributeNamespace="urn:simpletoken">      Policy Name
                    <saml1:AttributeValue ...                  Nombre de Usuario
                                 xsi:type="xs:string">
                                Juan
                     </saml1:AttributeValue>
             </saml1:Attribute>
       </saml1: AttributeStatement>
       <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
            ...
       </ds:Signature>
```

Figura 45 – Ejemplo de un *token* SAML generado y firmado por un organismo

Ejemplo de RST

La Figura 46 presenta un ejemplo simplificado de un mensaje RST, mostrando cómo ubicar los elementos requeridos para la solicitud, utilizando WS-Trust.

```
<s:Envelope xmlns:a="..." xmlns:s="...">
  <s:Header>
      ...
  </s:Header>
  <s:Body>
     <RequestSecurityToken
        xmlns="http://schemas.xmlsoap.org/ws/2005/02/trust">
        <TokenType>                                              Tipo de token
         http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLV1.1
        </TokenType>
        <AppliesTo xmlns="http://schemas.xmlsoap.org/ws/2004/09/policy">
                  <a:EndpointReference>                          Servicio
                     <a:Address>http://192.168.40.190:9000/Servicio</a:Address>
                  </a:EndpointReference>
        </AppliesTo>
        <RequestType>
                http://schemas.xmlsoap.org/ws/2005/02/trust/Issue
        </RequestType>
        <Issuer>                                                 Policy Name
               <a:Address>urn:nac</a:Address>
        </Issuer>
        <Base>
             <saml1:Assertion xmlns:saml1="..." AssertionID="..."
                   IssueInstant="2010-04-22T21:21:10.156Z" Issuer="Agesic" ... >
                    ...                                          Token SAML emitido por
             </saml1:Assertion>                                  organismo
        </Base>
        <SecondaryParameters>                                    Rol del Usuario
                  <Rol>Doctor</Rol>
        </SecondaryParameters>
      </RequestSecurityToken>
  </s:Body>
</s:Envelope>
```

Figura 46 – Ejemplo de un RST

Ejemplo Token emitido por PGE

En la Figura 47 se presenta una versión simplificada de un *token* de seguridad emitido por la PGE.

```
<saml:Assertion AssertionID="..." IssueInstant="..." Issuer="..."
      MajorVersion="1" MinorVersion="1" xmlns:saml="...">
      <saml:Conditions NotBefore="..." NotOnOrAfter="...">
            <saml:AudienceRestrictionCondition>
                  <saml:Audience>                                    Servicio
                        http://192.168.40.190:9000/Servicio
                  </saml:Audience>
            </saml:AudienceRestrictionCondition>
      </saml:Conditions>
      <saml:AuthenticationStatement
            AuthenticationInstant="..." AuthenticationMethod="...">
            <saml:Subject>
                  <saml:NameIdentifier>                    Rol del usuario en
                        uid=rolPruebaDoctor,cn=agesic          la PGE
                  </saml:NameIdentifier>
            </saml:Subject>
      </saml:AuthenticationStatement>
      <saml:AttributeStatement>
            <saml:Subject>
                  <saml:NameIdentifier>                    Rol del usuario en
                        uid=rolPruebaDoctor,cn=agesic          la PGE
                  </saml:NameIdentifier>
            </saml:Subject>                                      Policy Name
            <saml:Attribute AttributeName="User" AttributeNamespace="urn:nac">
                  <saml:AttributeValue>Juan</saml:AttributeValue>  Nombre de
                                                                   Usuario
            </saml:Attribute>
      </saml:AttributeStatement>
      <ds:Signature Id="uuid2765608c-0128-1428-9ce4-8913af9af38d"
                  xmlns:ds="http://www.w3.org/2000/09/xmldsig#">...
      </ds:Signature>
```

Figura 47 – Ejemplo de un *token* SAML emitido por el STS de la PGE

Ejemplo de Invocación

La Figura 48 presenta un ejemplo simplificado de un mensaje SOAP para consumir el servicio "Certificado de Nacidos Vivos" utilizando los estándares WS-Addressing y WS-Security con la información mencionada anteriormente.

```
<env:Envelope xmlns:env='…'>
    <env:Header xmlns:wsa='…'>
        <wsse:Security env:mustUnderstand='1'
              xmlns:ds='...'   xmlns:wsse='...' xmlns:wsu='...'>
              <saml:Assertion AssertionID='...' IssueInstant='...'
                    Issuer='...' MajorVersion='1' MinorVersion='1'
                    xmlns:saml='urn:oasis:names:tc:SAML:1.0:assertion'>
                   ...
              </saml:Assertion>
        </wsse:Security>          Cabezales WS-Security: token SAML emitido por la PGE

        <wsa:To>http://192.168.40.190:9000/Servicio</wsa:To>
        <wsa:Action>
http://xml.cnve.msp.gub.uy/.../certificadoCNVEWSDLPortType/registrarCNVE
        </wsa:Action>          Cabezales WS-Addressing: To (Servicio) y Action (Método)

    </env:Header>
    <env:Body>
       <ns1:solicitudCNVE usuario='Juan'
            xmlns='http://xml.msp.gub.uy/schema/cnveSchema'
            xmlns:ns1='http://xml.msp.gub.uy/schema/cnveSchema'>
            <numeroCertificado />
            <datosMadre>
                      <primerNombre>Marta</primerNombre>
            </datosMadre>
       </ns1:solicitudCNVE>                    Información de Negocio
    </env:Body>
</env:Envelope>
```

Figura 48 – Ejemplo de Mensaje SOAP para Consumir un Servicio de la PGE