

Implementación de una Plataforma para la Protección de Datos Personales en Soluciones Master Data Management de Gobierno Electrónico

Informe de Proyecto de Grado

Mayo, 2016

Autores

Acuña Ignacio

Serra Diego

Tutores

MSc. Ing. González Laura

Ing. Piedrabuena Federico

Resumen

La evolución de los sistemas de información ha generado que los datos electrónicos almacenados sean esenciales para la operativa de casi cualquier organización. Además, hay una creciente necesidad de compartir estos datos entre organizaciones, permitiendo la toma de decisiones o ejecución de operaciones de forma conjunta.

Actualmente, la orientación a servicios es el enfoque preferido para que distintas organizaciones interactúen con el fin de compartir datos y funcionalidades de negocio. Estas interacciones son soportadas por las denominadas plataformas de integración, las cuales son infraestructuras especializadas que facilitan la provisión, descubrimiento e invocación de servicios interoperables. Algunos ejemplos de sistemas soportados por éstas se pueden encontrar en infraestructuras de gobierno electrónico. El Estado uruguayo, en particular, cuenta con la Plataforma de Gobierno Electrónico Uruguay que incluye una Plataforma de Interoperabilidad.

A su vez, en estos ambientes de gestión integrada y colaborativa es muy importante que los datos compartidos sean válidos, actualizados y de fuentes confiables. Esto motivó la adopción de sistemas de *Master Data Management (MDM)*, con el fin de asegurar la uniformidad, exactitud y calidad de la información. Por otro lado, uno de los principales desafíos en el intercambio de información entre organizaciones es la aplicación de normativas asociadas a la protección de datos personales. Por ejemplo, en Uruguay se promulgó la Ley Nº 18.331 de Protección de Datos Personales y Acción de "Habeas Data".

En este contexto, surge la necesidad de que las plataformas de integración brinden mecanismos para monitorear y hacer cumplir las normas de protección de datos personales en las interacciones que se realizan en el marco de soluciones MDM inter-organizacionales.

En este proyecto se realizó un diseño detallado y una implementación de referencia de la solución propuesta en el artículo "*Enforcing Data Protection Regulations within e-Government Master Data Management Systems*", en el contexto de la realidad uruguaya. Para esto, se utilizaron tecnologías de *middleware* y soluciones de MDM concretas, incluyendo el estándar *eXtensible Access Control Markup Language (XACML)* como mecanismo de control de acceso.

En primer lugar, se comprendió el marco de trabajo presentado en el artículo, relevando los conceptos y estándares asociados a MDM, Enterprise Service Bus (ESB), XACML y las leyes de protección de datos. En segundo lugar, se realizó un diseño detallado enmarcado en una plataforma de gobierno electrónico, haciendo uso de tecnologías de MDM, para asegurar la protección de datos personales.

Por último, se implementó un prototipo acorde al diseño detallado, utilizando los productos *Talend MDM*, *MuleSoft ESB* y *WSO2 Identity Server*. Además, se desarrollaron casos de estudio basados en la realidad del Estado uruguayo que permitieron tener una primera validación de la solución.

Palabras Claves: Protección de datos personales, Master Data Management, XACML, Gobierno electrónico, Plataformas de Integración, Enterprise Service Bus.

Índice

1.	Introducción	8
1.1.	Descripción del problema	8
1.2.	Objetivos	9
1.3.	Resultados y aportes del proyecto	10
1.4.	Organización del informe	10
2.	Marco Conceptual	11
2.1.	Seguridad.....	11
2.1.1.	Conceptos básicos	11
2.1.2.	Protocolos	11
2.1.3.	Estándares de seguridad XML	11
2.2.	SOA – Arquitectura Orientada a Servicios y Servicios Web	12
2.2.1.	WSDL - Web Services Description Language	12
2.2.2.	SOAP - Simple Object Access Protocol	13
2.2.3.	WS-Security	14
2.2.4.	WS-Addressing	14
2.3.	Master Data Management (MDM)	15
2.4.	ESB - Enterprise Service Bus	18
2.4.1.	Principales capacidades.....	19
2.4.2.	Patrón de Modelo de Datos Canónicos.....	20
2.5.	XACML - eXtensible Access Control Markup Language.....	20
2.5.1.	Arquitectura de referencia.....	21
2.5.2.	Formato de políticas.....	23
2.6.	Plataforma de Gobierno Electrónico del Estado Uruguay	28
2.7.	Ley de protección de datos personales en Uruguay	28
3.	Análisis de requerimientos.....	31
3.1.	Marco de trabajo.....	31
3.2.	Interpretación de la ley de protección de datos de Uruguay	32
3.3.	Especificación de requerimientos	34
3.3.1.	Gestión de consentimientos	35
3.3.2.	Administración de datos personales	35
3.3.3.	Gestión de orígenes de datos personales	36
3.3.4.	Administrar propósitos y acciones	36
3.3.5.	Registrar bases de datos de organizaciones	36

3.3.6.	Gestión de categorías de datos personales	36
3.3.7.	Gestión de modelo de datos personales.....	36
3.3.8.	Validar interacción MDM	36
3.4.	Selección de estilo de solución de MDM	36
3.5.	Análisis de interacciones MDM	37
3.6.	Análisis de consentimientos sobre la Agencia Supervisora	39
3.7.	Resumen y conclusiones del análisis.....	39
4.	Propuesta de solución.....	41
4.1.	Descripción general.....	41
4.2.	Componentes de la arquitectura de la solución	42
4.2.1.	Master Data Management	43
4.2.2.	Administrador y evaluador de políticas XACML.....	44
4.2.3.	Sistema de Manejo de Consentimientos (SMC).....	45
4.2.4.	Enterprise Service Bus (ESB).....	45
4.2.5.	Sistema de Consulta de Datos Personales	48
4.2.6.	Organismos.....	49
4.3.	Arquitectura de la solución	50
4.4.	Interacciones MDM.....	50
4.5.	Diseño y evaluación de políticas XACML.....	52
4.5.1.	Consentimiento como política	52
4.5.2.	Dominio de categorías y atributos	55
4.5.3.	Estructura general de políticas.....	55
4.5.4.	Casos de solicitudes de datos personales	57
4.6.	Interacción de componentes de la arquitectura.....	60
4.6.1.	Solicitudes a partir del identificador/es del ciudadano	60
4.6.2.	Solicitudes que no incluyen identificador/es del ciudadano y consultan a partir de atributos sensibles	61
5.	Implementación	63
5.1.	Selección de productos	63
5.1.1.	Master Data Management	63
5.1.2.	Administrador y Evaluador de políticas XACML.....	67
5.1.3.	Enterprise Service Bus.....	69
5.2.	Detalles de implementación	70
5.2.1.	Componente MDM.....	71

5.2.2.	Componente Evaluador.....	73
5.2.3.	Organismos.....	77
5.2.4.	Componente ESB.....	77
5.2.5.	Vista de distribución.....	80
6.	Caso de estudio	83
6.1.	Objetivo	83
6.2.	Descripción general del caso de estudio.....	83
6.3.	Consulta de datos personales	89
6.4.	Acceso a datos sensibles con excepción	92
6.5.	Actualización de datos personales	94
6.6.	Conclusiones del caso de estudio.....	95
7.	Conclusiones y trabajo futuro	96
7.1.	Conclusiones.....	96
7.2.	Trabajo a futuro	97
8.	Referencias.....	99
Apéndice 1.	Políticas de ejemplo utilizadas	103
Apéndice 2.	Mensajes XACML.....	110
Apéndice 3.	Glosario	114

1. Introducción

En este informe se presenta el proyecto de grado *“Implementación de una Plataforma para la Protección de Datos Personales en Soluciones Master Data Management de Gobierno Electrónico”*. En el presente capítulo se describen la problemática abordada, los objetivos planteados, los resultados del proyecto y la organización del documento.

1.1. Descripción del problema

La evolución de los sistemas de información ha generado que los datos electrónicos almacenados sean esenciales para la operativa de casi cualquier organización. Además, hay una creciente necesidad de compartir estos datos entre organizaciones. Esto sucede ya que los datos que posee una única organización pueden no ser suficientes para la toma de decisiones o ejecución de operaciones [1].

Actualmente, la orientación a servicios es el enfoque preferido para que distintas organizaciones interactúen con el fin de compartir datos y funcionalidades de negocio [2]. Estas interacciones son soportadas por las denominadas plataformas de integración, las cuales son infraestructuras especializadas que incluyen una capa de procesamiento que facilita la provisión, descubrimiento e invocación de servicios interoperables. Dicha capa de procesamiento permite realizar validaciones y monitoreo de invocaciones, además de resolver problemas de integración. Las plataformas de integración se basan en tecnologías de *middleware*, dentro de las que se destacan los *Enterprise Service Bus (ESB)* [3].

Algunos ejemplos de sistemas soportados por plataformas de integración se pueden encontrar en infraestructuras de gobierno electrónico. El Estado uruguayo, en particular, cuenta con la Plataforma de Gobierno Electrónico Uruguay que incluye una Plataforma de Interoperabilidad [4].

A su vez, en estos ambientes de gestión integrada y colaborativa es muy importante que los datos compartidos sean válidos, actualizados y de fuentes confiables. Esto motivó la adopción de sistemas de Gestión de Datos Maestros (*Master Data Management, MDM*) como un componente clave en estos ambientes, con el fin de asegurar la uniformidad, exactitud y la aplicación de prácticas rigurosas y sistemáticas de gestión de la información [5].

Por otro lado, uno de los principales desafíos en el intercambio de información entre organizaciones es la aplicación de normativas asociadas a la protección de datos personales [1]. Por ejemplo, en Uruguay se promulgó la Ley Nº 18.331 de Protección de Datos Personales y Acción de "Habeas Data". Esta ley clasifica a los datos de los ciudadanos como públicos o sensibles y establece que toda organización que desee compartir algún dato sensible, debe contar con un consentimiento explícito otorgado por el titular del dato.

En este contexto, surge la necesidad de que las plataformas de integración brinden mecanismos para monitorear y hacer cumplir las normas de protección de datos personales en las interacciones que se realizan en el marco de soluciones MDM inter-organizacionales. En particular, el artículo *“Enforcing Data Protection Regulations within e-Government Master Data Management Systems”* [1] propone soluciones conceptuales para abordar esta problemática en un contexto de gobierno electrónico, utilizando el estándar *eXtensible Access Control Markup Language* (XACML) como mecanismo de control de acceso.

No obstante, las soluciones propuestas en el artículo se encuentran planteadas con un alto nivel de abstracción, no contando hasta el momento con una implementación de referencia que lleve a cabo la propuesta y utilice tecnologías y productos concretos.

1.2. Objetivos

El objetivo general de este proyecto es realizar un diseño detallado y una implementación de referencia de la solución propuesta en el artículo *“Enforcing Data Protection Regulations within e-Government Master Data Management Systems”*[1] (de aquí en más el artículo) en el contexto de la realidad uruguaya, con tecnologías de *middleware* y soluciones de MDM concretas.

El cumplimiento del objetivo general se puede desglosar en los siguientes objetivos específicos:

- Relevar conceptos y estándares asociados a Master Data Management (MDM), Enterprise Service Bus (ESB), eXtensible Access Control Markup Language (XACML), seguridad de la información y Arquitecturas Orientadas a Servicios (SOA).
- Comprender el marco de trabajo del cual parte el proyecto, analizando la Plataforma de Interoperabilidad de Gobierno Electrónico en Uruguay en combinación con una solución de Master Data Management.
- Relevar los requerimientos de acuerdo a lo especificado en la Ley N° 18.331 de Protección de Datos Personales y Acción de “habeas data”, en el marco de trabajo estudiado.
- Realizar un diseño detallado considerando las soluciones propuestas en el artículo e incorporando tecnologías concretas.
- Evaluar productos y herramientas que implementan los conceptos de MDM, XACML y ESB, seleccionando aquellos que permitan abordar los requisitos presentados en el diseño.
- Construir un prototipo que implemente el diseño planteado.
- Desarrollar un caso de estudio basado particularmente en la realidad del gobierno uruguayo con el propósito de validar la solución planteada.

1.3. Resultados y aportes del proyecto

A continuación, se enumeran los principales aportes del proyecto:

- Relevamiento de los conceptos y estándares asociados a MDM, ESB, XACML, seguridad de la información y SOA relevantes para el desarrollo del proyecto.
- Análisis detallado del marco de trabajo propuesto el cual combina la Plataforma de Interoperabilidad de Gobierno Electrónico en Uruguay con una solución de MDM.
- Relevamiento de los requerimientos de acuerdo a la especificación de la Ley N° 18.331 y el marco de trabajo estudiado.
- Diseño detallado de las soluciones propuestas en el artículo.
- Análisis de productos de MDM, XACML y ESB determinando los más apropiados para emular el contexto del problema y abordar los requisitos diseñados. Para esto se seleccionaron los productos *Talend MDM*, *WSO2 Identity Server* y *MuleSoft ESB* respectivamente.
- Implementación de un prototipo que permitió integrar los conceptos de MDM, ESB y XACML, a partir del diseño realizado y manteniendo un bajo acoplamiento con las herramientas de MDM y XACML seleccionadas.
- Desarrollo de un caso de estudio basado en la realidad uruguaya que permitió validar la solución para los distintos tipos de interacción analizados.

1.4. Organización del informe

El resto del documento se organiza según lo descripto a continuación:

El Capítulo 2 contiene el marco conceptual en el cual se detallan los principales conceptos teóricos que se utilizaron durante el desarrollo del proyecto.

El Capítulo 3 detalla el análisis de requerimientos, incluyendo una sección de análisis y otra de especificación de funcionalidades.

El Capítulo 4 presenta la propuesta de solución, explicando la arquitectura definida y detallando cada uno de los sistemas involucrados.

El Capítulo 5 profundiza en los aspectos relacionados con la implementación. Esto incluye una sección en la que se evalúan productos disponibles y se selecciona los que se consideran más apropiados. Además, se detalla cómo se integran a la solución.

El Capítulo 6 desarrolla un caso de estudio que permite tener una primera validación de la solución propuesta.

El Capítulo 7 presenta las conclusiones del proyecto y enumera posibles líneas a realizar como trabajo a futuro.

2. Marco Conceptual

En este capítulo se presentan y explican conceptos teóricos relevantes para el proyecto como ser conceptos básicos de seguridad, Servicios Web, MDM, ESB, XACML, la Plataforma de Gobierno Electrónico del Estado Uruguayo y la Ley de Protección de Datos Personales.

2.1. Seguridad

Esta sección describe conceptos y protocolos de seguridad de interés para el proyecto.

2.1.1. Conceptos básicos

Integridad de datos: Es la propiedad que cumplen los datos determinando que no han sido modificados, perdidos o eliminados de forma no autorizada [6].

Disponibilidad: Es la propiedad de un sistema o recurso de ser usable y accesible a demanda de acuerdo a las especificaciones de desempeño de cada uno. Por ejemplo, un sistema cumple la propiedad si provee los servicios de acuerdo a su diseño siempre que un usuario lo solicite [6].

Confidencialidad: Es la propiedad de la información que determina que ésta no está disponible para individuos, aplicaciones o procesos no autorizados [6].

No repudio: Es la propiedad donde las entidades involucradas en una comunicación no pueden negar haber participado en ella [7].

Codificar: En computación, es el mecanismo que transforma un conjunto de caracteres (letras, números y otros símbolos) en un formato especializado que permite realizar de manera más eficiente su transmisión o almacenamiento [6].

Cifrar: Es el proceso de transformar el contenido de un texto mediante encriptación, logrando que no sea legible o se encuentre directamente disponible [6].

Autenticar: Es el proceso de verificar una identidad reclamada por o para una entidad, sistema, proceso o persona [6].

2.1.2. Protocolos

Autenticación básica: Dentro de un intercambio de mensajes HTTP, es el proceso de intercambio de usuario y contraseña entre el cliente y el servidor para autenticar al cliente. Las credenciales del cliente se envían codificadas en formato base64 [8].

Secure Socket Layer (SSL): Es un protocolo de seguridad que establece un canal seguro entre dos sistemas que se comunican a través de una red, creando una sesión segura. Este protocolo establece el uso de claves criptográficas para que el intercambio de la información durante dicha sesión sea confidencial y no repudiable [9].

2.1.3. Estándares de seguridad XML

eXtensible Access Control Markup Language (XACML): Es un estándar, basado en XML, que permite implementar control de acceso. Se detalla en la sección 2.5 de este capítulo.

XML Signature: Es un estándar diseñado para utilizar en transacciones que involucran datos en formato XML. Además, define un esquema para referenciar el resultado de la operación de

firma digital aplicada a datos arbitrarios, típicamente en formato XML. Esta norma permite la autenticación, integridad y soporte al no repudio de los datos que se firman [10].

Una de las características fundamentales de *XML Signature* es que permite firmar porciones específicas de los datos XML, en lugar de todos los datos; esto es de gran importancia cuando existen pequeñas porciones que son sensibles, ya que disminuyen los costos de procesamiento para realizar la firma [10].

XML Encryption: Es un estándar que especifica cómo realizar el encriptado de datos en formato XML [11]. Los datos pueden ser arbitrarios, desde un documento XML hasta un elemento XML y el resultado de la encriptación es un elemento denominado *XML Encryption EncryptedData* que contiene o identifica los datos cifrados.

Cuando se encripta un elemento XML o su contenido, el elemento *EncryptedData* reemplaza al elemento o su contenido respectivamente en la versión encriptada del documento XML. Por otra parte, cuando se encriptan documentos enteros el elemento *EncryptedData* podría convertirse en la raíz de un nuevo documento XML.

2.2. SOA – Arquitectura Orientada a Servicios y Servicios Web

Una Arquitectura Orientada a Servicios (Service Oriented Architecture, SOA) se puede definir como una arquitectura con bajo nivel de acoplamiento, diseñada para cumplir con los requerimientos de negocio de una organización [12].

En otro enfoque, se puede ver como un estilo arquitectónico para crear aplicaciones de software que utilizan servicios, promoviendo el escalamiento y reutilización de componentes. En cuanto al concepto de servicio, se lo puede definir de forma genérica, como la implementación de una funcionalidad de negocio bien definida, que puede ser consumida por otras aplicaciones o procesos de negocio [13].

Los Servicios Web son una de las tecnologías preferidas para la implementación de este tipo de arquitecturas.

Un Servicio Web es un sistema de software diseñado para soportar la interoperabilidad entre dos aplicaciones a través de una red. Presenta su descripción en un contrato de definición del servicio denominado *Web Service Description Language* (WSDL), donde se detallan sus interfaces, tipos de datos y formas de acceso. Los sistemas interactúan con el servicio según lo especificado por su descripción con mensajes en formato SOAP [14].

2.2.1. WSDL - Web Services Description Language

WSDL es un lenguaje basado en XML que permite describir las interfaces, formas de acceso y ubicación de los Servicios Web. Este lenguaje se divide en dos partes, por un lado la Descripción Abstracta, define de forma general la interfaz del servicio, compuesta por las definiciones abstractas de los tipos de datos (*types*), los parámetros de entrada y salida de las operaciones (*messages*), las operaciones abstractas y una interfaz específica del servicio (*portTypes*). Por otro lado, la Descripción Concreta asocia una dirección de red (*port*) y un protocolo de comunicación (*binding*) a una Descripción Abstracta [15].

En la Figura 1, se puede ver un Servicio Web de ejemplo, descrito mediante un documento WSDL:

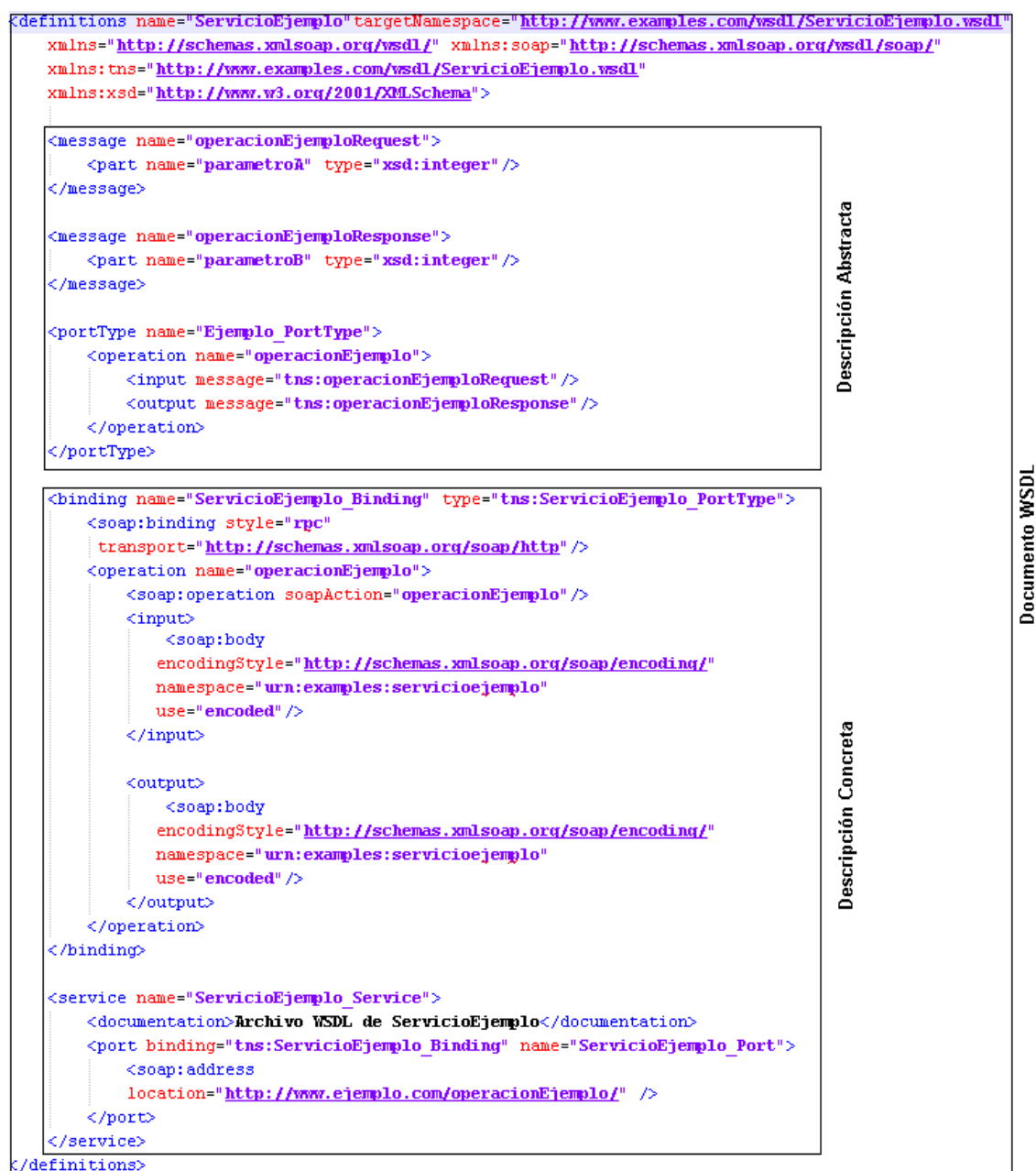


Figura 1 - Ejemplo WSDL

2.2.2. SOAP - Simple Object Access Protocol

SOAP es un protocolo ligero que establece el formato de los mensajes a intercambiar en un entorno descentralizado y distribuido, extendiendo el protocolo HTTP y permitiendo implementar Servicios Web [16].

Un mensaje SOAP es expresado en formato XML y se compone de una sección principal *Envelope* que engloba todo el mensaje, conteniendo los elementos *Header* y *Body*. El primero es opcional y permite extender las características del mensaje, mientras que el segundo contiene los datos que se intercambian entre las aplicaciones.

En la Figura 2 se puede ver un ejemplo de este tipo de mensajes, incluyendo las secciones *Envelope*, *Header* y *Body*.

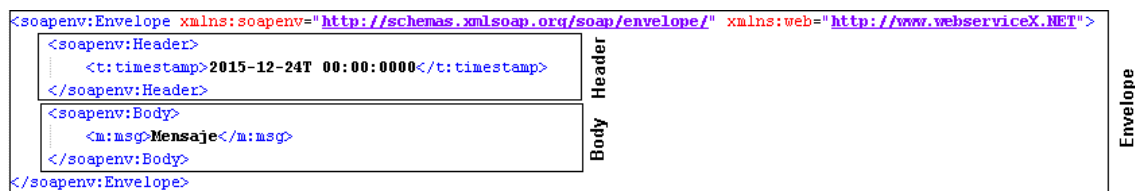


Figura 2 - Ejemplo mensaje SOAP

El protocolo SOAP cuenta con un conjunto de extensiones *WS-** que permiten implementar mecanismos de seguridad, mensajería confiable y transacciones en Servicios Web.

2.2.3. WS-Security

WS-Security es un estándar de la familia *WS-** que permite brindar seguridad en mensajes SOAP. El objetivo de éste es garantizar confidencialidad, integridad y autenticación, lo que se logra utilizando los estándares *XML Encryption* para el cifrado y *XML Signature* para la firma de datos XML. Además, el uso de *tokens* de seguridad permite autenticar al emisor del mensaje [17].

2.2.4. WS-Addressing

WS-Addressing provee un mecanismo estándar para referenciar Servicios Web y direccionar mensajes independientemente del protocolo de transporte utilizado. El estándar especifica un *EndpointReference* como una referencia a un Servicio Web que contiene una dirección, parámetros de referencia y metadatos. Además define las siguientes propiedades:

To: Dirección a la cual se envía el pedido.

Action: Acción que debe realizar el receptor del mensaje.

ReplyTo: Dirección de un *EndpointReference* al que se debe enviar la respuesta de la solicitud.

FaultTo: Dirección de un *EndpointReference* al que se debe enviar la respuesta cuando no se pudo procesar la solicitud.

MessageId: Cadena de caracteres que identifica unívocamente al mensaje.

RelatesTo: Cadena de caracteres que permite correlacionar mensajes.

WS-Addressing permite implementar respuestas desacopladas, es decir, Servicios Web de respuesta asincrónica. Además, permite identificar (*MessageId*) y correlacionar mensajes (*RelatesTo*), para poder determinar a qué petición corresponde una respuesta [18] [19].

2.3. Master Data Management (MDM)

MDM se puede ver como una disciplina organizacional, diseñada para reducir y eventualmente evacuar los errores, las redundancias y las inconsistencias de los datos que se presentan en los sistemas de información de la actualidad. Por lo tanto, implementar Master Data Management en una organización puede ayudar a aumentar su productividad e incentivar el rendimiento operativo al mejorar la precisión de la información y su comunicación en la organización o por fuera de ella [20].

MDM tiene como principal objetivo brindar un único punto de referencia, consistente y preciso de los datos maestros de una organización entre los distintos sistemas y bases de datos que mantiene. Por dato maestro se entiende a todo aquel elemento común a una organización, como pueden ser los clientes, proveedores, activos o productos. Éstos son aquellos datos esenciales para la operativa de la organización.

Sin embargo, no utilizar MDM puede provocar distintas falencias: *“Tener ‘múltiples versiones de datos verdaderos’ conduce con toda probabilidad a operaciones empresariales ineficientes, a la toma de decisiones incorrectas, previsiones poco precisas y a la incapacidad para adherirse a las directrices regulatorias, y otros problemas que pueden obstaculizar el funcionamiento de la empresa, consumir recursos y reducir la ventaja competitiva”* [20].

Una de las grandes ventajas que brinda MDM es que los datos se pueden desacoplar de las aplicaciones, conciliarlos sintácticamente y semánticamente en un *hub*¹ centralizado y desde allí ser distribuidos a las aplicaciones que los utilizaban anteriormente u otras que requieran el consumo de los mismos. Por lo tanto, podría decirse que MDM está orientado al “dato como servicio”, pero además cumple con la estrategia SOA [21].

En resumen, MDM logra gestionar los datos centralizadamente, permitiendo a partir de esto definir políticas, estándares y procesos que determinen el uso, desarrollo y gestión de la información, también llamado *Data Governance*. Una solución de MDM genera una vista consolidada, completa y precisa de los datos maestros.

Algunas de las principales funcionalidades que brinda una solución de MDM son:

- Definición de funciones y derechos de acceso individuales para cada etapa del proceso de gestión de los datos maestros.
- Extracción de los datos de referencia de las distintas fuentes, para cargarlos en el repositorio².
- Servicios de limpieza de los datos, permitiendo por ejemplo detectar duplicados.
- Capacidades de colaboración para coordinar las decisiones de reconciliación y racionalización de los datos maestros.
- Detección de cambios, sincronización y replicación de los datos entre el repositorio y las fuentes de datos afectadas ante cualquier cambio en el repositorio.

¹ Sistema que centraliza el acceso a la información.

² Base de datos principal de la solución de MDM.

- Control de versión y validación de los cambios en el sistema, pudiendo recrear un estado anterior y modelar estados futuros.

Los estilos arquitectónicos de una solución de Master Data Management se pueden encontrar en la literatura con distintos nombres y características. Sin embargo, para el contexto del proyecto y manteniendo la línea propuesta en el artículo “Enforcing Data Protection Regulations within e-Government Master Data Systems”, se considera que una solución de Master Data Management puede ser implementada de acuerdo a los siguientes estilos [1]:

Estilo Registro (Virtual)

El estilo Registro consiste en almacenar en el repositorio la menor cantidad de atributos posibles de los datos maestros, utilizando normalmente aquellos críticos que identifican a la entidad. Por lo tanto, se almacenan referencias a los registros originales de los sistemas fuente que son parte de la entidad. En la Figura 3 se puede ver cómo la solución de MDM de tipo Registro combina las vistas de las fuentes de dos registros en una sola entidad [22].

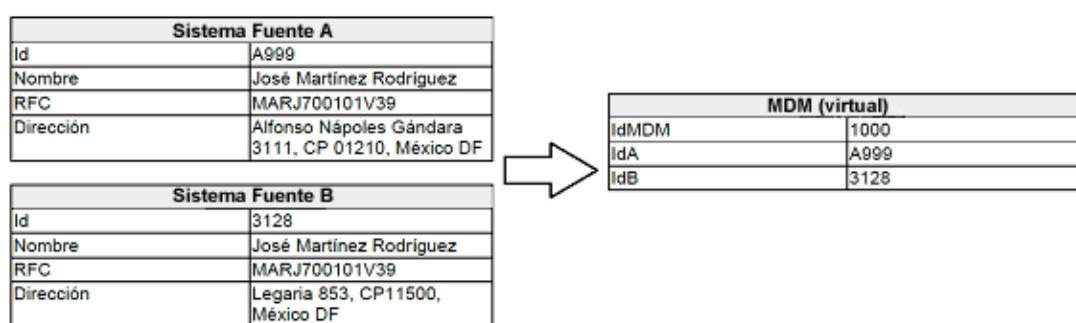


Figura 3 - Ejemplo MDM Registro [Adaptación de [22]]

Para combinar estos registros se debe considerar algún algoritmo que tome el valor de los atributos que crea más conveniente. Sin embargo, siempre es posible acceder a los datos originales mediante las referencias a sus orígenes.

Algunas de las ventajas que brinda esta implementación son:

- La adopción de este estilo no implica cambios en las bases de datos de las fuentes.
- Poca redundancia de los datos.
- Estilo de MDM más sencillo de implementar.

Sin embargo, genera algunas desventajas:

- El mantenimiento de los datos se debe llevar a cabo en las fuentes.
- Para poder visualizar datos no almacenados en el repositorio de MDM las fuentes deben proveer una interfaz para obtener los atributos faltantes a través del identificador. Estas operaciones enlentecen las consultas que utilicen estos atributos.
- Los servicios de limpieza deben ser provistos por las fuentes.

Estilo Repositorio (Físico)

El estilo Repositorio almacena localmente el modelo completo de los datos maestros. En la Figura 4 se puede visualizar cómo se consolidan dos registros de orígenes distintos a uno almacenado en una solución de MDM estilo Repositorio [22]:

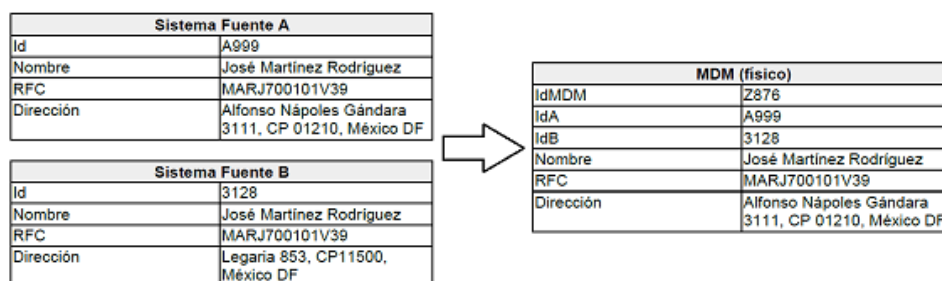


Figura 4 - Ejemplo MDM Repositorio [Adaptación de [22]]

En la Figura 4, se puede ver que a partir de los registros de las fuentes de datos que identifican a un mismo sujeto, se genera un único registro completo en el repositorio, almacenando una referencia a los identificadores en las fuentes correspondientes. Sin embargo, se deben tomar decisiones al momento de almacenar los atributos compartidos y con distintos valores, pudiendo ser un proceso complejo de resolver. Esto sucede con el campo *Dirección* en el ejemplo anterior. Algunos de los criterios a considerar podrían ser la elección del elemento más nuevo o el proveniente de la entidad que asegure la veracidad para el dato en cuestión.

Teniendo el conjunto de datos maestros almacenados en el repositorio, resulta de interés ver cómo se realizan las interacciones con los mismos, identificando los siguientes tres tipos:

- **Consolidado:** Los sistemas impactan sus cambios en las fuentes locales, los cuales se sincronizan luego con el repositorio del MDM.
- **Centralizado:** Los sistemas realizan cambios directamente en el repositorio del MDM, quien luego se encarga de actualizar las fuentes de datos correspondientes.
- **Coexistencia:** Se mantiene una combinación de ambos tipos, donde para algunos datos y algunas aplicaciones los cambios se realizan en las fuentes, mientras que para otras aplicaciones u otros datos los cambios se impactan directamente en el repositorio del MDM.

Como ventajas de este estilo se puede observar que:

- El repositorio del MDM mantiene el *golden record*³, brindando siempre la información más reciente y veraz.
- Se cuenta con la información limpia y sin duplicados.
- A la hora de resolver consultas, es un sistema autónomo, dado que la información almacenada es completa.

³ Es el registro más reciente, completo y válido del dominio.

Sin embargo, se presentan algunas desventajas:

- Implementación compleja, requiriendo un buen análisis de la gobernabilidad de los datos maestros y necesitando mayor tiempo para la implementación que en una solución de MDM estilo Registro.
- Normalmente su aplicación implica cambios en las aplicaciones y procesos con los que ya cuentan las organizaciones.

Estilo Híbrido

El estilo Híbrido surge como una combinación de los estilos Registro y Repositorio tomando algunas ventajas de ellos. Este estilo mantiene un subconjunto considerable de atributos de datos maestros en el repositorio y sólo las referencias para los atributos restantes.

Por ejemplo, se podrían mantener en el repositorio de datos maestros sólo aquellos atributos que se accedan con más frecuencia.

Como principal desventaja se nota la redundancia de datos que se obtiene en comparación con el estilo Registro.

2.4. ESB - Enterprise Service Bus

ESB refiere a la arquitectura de un *middleware*⁴ de tipo plataforma que permite integrar diferentes aplicaciones en una infraestructura de *bus* basándose en los principios de mensajería, arquitecturas orientadas a servicios (SOA) y arquitecturas orientadas a eventos [23] [24].

El ESB cuenta con capacidades de conectividad, mediación y despliegue que permiten interconectar aplicaciones, bases de datos, sistemas legados o Servicios Web entre sí, como se puede ver en la Figura 5.

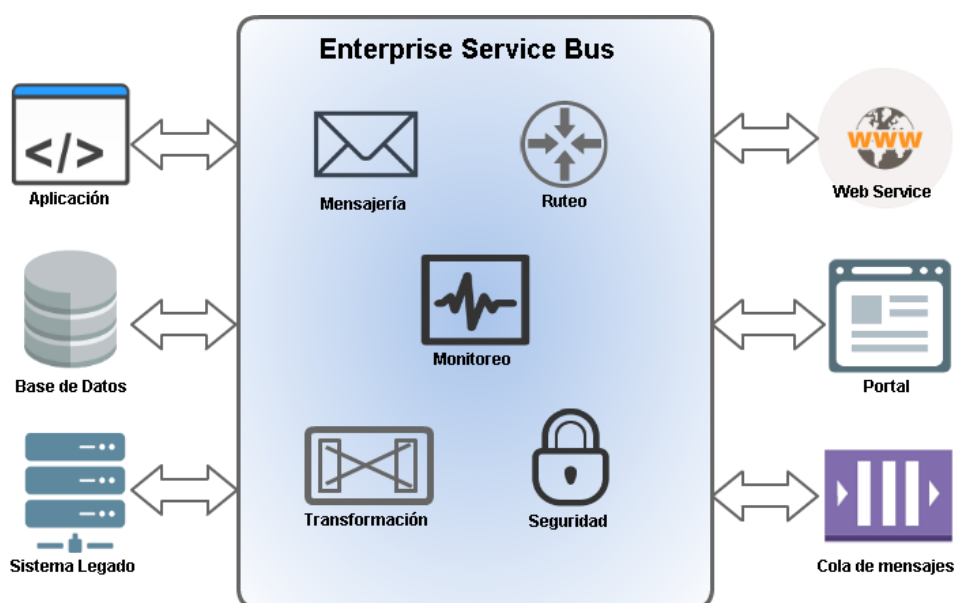


Figura 5 - Capacidades de un ESB

⁴ Sistema o aplicación intermediaria que permite la integración de otros sistemas o aplicaciones.

La conexión de aplicaciones al *bus* se realiza mediante conectores que abstraen la configuración y son brindados por la plataforma. Estos conectores ofrecen distintos protocolos de comunicación como *HTTP*, *JMS* o *JDBC*. Además, promueven el bajo acoplamiento entre las aplicaciones.

Asimismo, las capacidades de conectividad del ESB agilizan el proceso de creación de arquitecturas de tipo SOA, así como también la integración con sistemas legados, que podrían no haber sido diseñados para operar en un entorno orientado a servicios.

Cuando dos aplicaciones se comunican utilizando un ESB, los mensajes fluyen a través del *bus* y son procesados por flujos de mediación que permiten realizar transformaciones, re-direccionamiento y filtrado de mensajes. Además, un ESB provee funcionalidades de *caché*, balanceo de carga y cola de mensajes [23] [24].

2.4.1. Principales capacidades

En esta sección se describen las principales capacidades que hacen a un *middleware* de tipo ESB.

Transformación de mensajes

Las aplicaciones que se integran en un ESB presentan modelos de datos propietarios que podrían ser diferentes al del resto de las aplicaciones integradas al *bus*. Es por esto que el ESB ofrece un componente que permite transformar los mensajes que recibe facilitando la capacidad de integración [25].

Las transformaciones abarcan modificaciones a nivel de modelo o formato de datos, así como también el enriquecimiento o filtrado de contenido. Para su implementación se destaca el uso del estándar eXtensible Stylesheet Language Transformation (XSLT) [26].

Ruteo de mensajes

Los mensajes que recibe un ESB pueden estar destinados para una u otra aplicación dependiendo de varios factores. Es por esto que el ESB ofrece un componente que permite direccionar mensajes a distintos destinatarios a partir de su contenido, el contexto dado por sus propiedades, el entorno en el cual se recibe, estrategias de balanceo de carga o según el itinerario del mensaje [25].

Virtualización de servicios

La virtualización de servicios es una capacidad del ESB que ayuda a aislar la definición de un servicio existente como la dirección, parámetros, versionado y manejo de información, a los consumidores del servicio.

Esta capacidad permite, por ejemplo, exponer servicios virtuales que implementan mecanismos de seguridad incorporando una capa de seguridad al servicio ya existente sin la necesidad de modificar su implementación o requerir un nuevo despliegue [27].

En la Figura 6 se muestra un ejemplo genérico de virtualización de servicios donde el Solicitante accede al servicio del Proveedor a través del Proxy brindado por el ESB. Esto representa el Patrón de proxy de servicio simple.



Figura 6 - Patrón de proxy de servicio simple [28]

2.4.2. Patrón de Modelo de Datos Canónicos

El patrón Modelo de Datos Canónicos (MDC) establece una forma de comunicación entre aplicaciones que manejan distintos esquemas de datos, mediante el uso de un modelo de datos común a todos los sistemas.

Este patrón provee un nivel de indireccionamiento adicional entre los modelos de datos utilizados por cada aplicación. De esta forma, si una nueva aplicación es incorporada en la solución de integración, sólo se debe configurar una nueva transformación entre el modelo de datos de la aplicación y el MDC. Esto es independiente de la cantidad de aplicaciones que ya estén integradas en la solución [25].

2.5. XACML - eXtensible Access Control Markup Language

XACML es un estándar basado en XML, que define un formato de políticas de seguridad y una arquitectura de referencia para la evaluación de las políticas, con el objetivo de establecer un lenguaje común para implementar control de acceso, especialmente de tipo ABAC (*Attribute Based Access Control* - Control de Acceso Basado en Atributos) [29].

Los principales requerimientos de un lenguaje de políticas para expresar políticas de seguridad de sistemas de información son:

- Definir un formato de *reglas* que se aplican sobre *sujetos* que realizan *acciones* sobre *recursos* en un *entorno*.
- Permitir la asignación de reglas a políticas.
- Proveer un mecanismo de combinación de *reglas* y *políticas* en un único conjunto de políticas.
- Especificar una forma de basar las decisiones de autorización en atributos del sujeto, recurso y entorno.
- Definir un conjunto de operadores lógicos que se pueden aplicar sobre atributos del sujeto, recurso y entorno.

- Contar con un método para identificar qué política se aplica a una acción a partir de los atributos del sujeto, recurso y entorno.

2.5.1. Arquitectura de referencia

En esta sección se describe la arquitectura de referencia para la evaluación de políticas que propone el protocolo XACML.

Componentes XACML

A continuación se detallan brevemente cada uno de los componentes XACML:

Policy Enforcement Point (PEP): Se encarga del control de acceso, realizando pedidos de decisión y haciendo cumplir las decisiones de autorización.

Policy Decision Point (PDP): Se encarga de evaluar políticas bien formadas y retornar el permiso, denegación o la no aplicabilidad de la misma.

Policy Information Point (PIP): Se encarga de completar atributos requeridos para evaluar la solicitud, pero que no se encuentren presentes en ella.

Policy Administration Point (PAP): Permite gestionar los atributos necesarios del *XACML Request*, las reglas, políticas y conjunto de políticas a utilizar en las validaciones.

El protocolo sugiere el uso del siguiente conjunto de categorías de atributos:

- **Subject:** Identificación del sujeto que realiza el pedido.
- **Resource:** Recurso sobre el cuál se solicita el acceso.
- **Action:** Acción a realizar sobre el recurso solicitado.
- **Environment:** Provee información adicional necesaria para solicitar la autorización, por ejemplo *timestamp*.

A su vez, las categorías de atributos están compuestas por atributos.

Cabe destacar que XACML v3.0 brinda la posibilidad de definir nuevas categorías de atributos, así como también agregar o modificar atributos dentro de ellas.

XACML Request

Este elemento representa el mensaje de la solicitud que se pretende validar y está compuesto por un conjunto de atributos agrupados en categorías con su valor correspondiente.

XACML Response

La respuesta que se obtiene al evaluar una solicitud se denomina *XACML Response* y contiene la decisión sobre la autorización del pedido realizado y cuyo resultado puede ser:

- **Permit:** Se permite el acceso al pedido evaluado.
- **Deny:** Se deniega el acceso al pedido evaluado.
- **Not Applicable:** No es posible evaluar el pedido.
- **Indeterminate:** No es posible evaluar el pedido.

Se puede ver cómo se utilizan los algoritmos para determinar el resultado de la evaluación de una solicitud en la sección Combinación de algoritmos.

Interacción de componentes XACML

En esta sección se muestran los mensajes que se intercambian entre los componentes del protocolo XACML de forma secuencial, comenzando por un pedido de acceso o *XACML Request*.

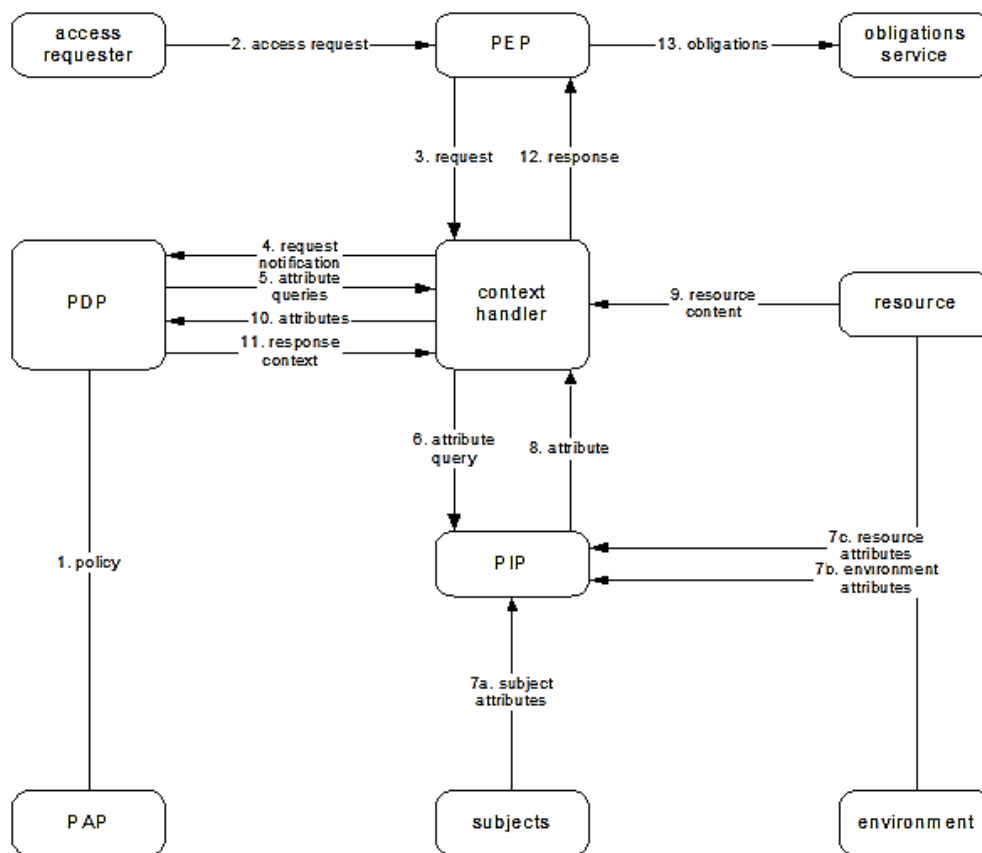


Figura 7 - Diagrama de interacción componentes XACML [29]

A continuación se describen los pasos señalados en la Figura 7:

1. Mediante el PAP se crean las políticas, reglas y conjuntos de políticas, que luego el PDP utilizará para tomar las decisiones.
2. El solicitante del acceso envía un pedido al PEP.
3. El PEP envía el pedido solicitando acceso al manejador de contexto (*context-handler*) en un formato conocido. Opcionalmente puede incluir atributos de los sujetos, recurso, acción, entorno y otras categorías.
4. El manejador de contexto crea un contexto para el pedido XACML, de forma opcional agrega atributos solicitándolos al PIP.
5. El PDP solicita algún atributo adicional de Sujeto, Recurso, Acción, entorno u otras categorías desde el manejador de contexto.
6. El manejador de contexto solicita los atributos al PIP.
7. El PIP obtiene los atributos pedidos.
8. El PIP retorna los atributos pedidos al manejador de contexto.
9. De forma opcional, el manejador de contexto incluye el recurso en el contexto.
10. El manejador de contexto envía los atributos pedidos y opcionalmente el recurso al PDP. El PDP evalúa la política.

11. El PDP retorna el contexto de respuesta incluyendo la decisión de autorización al manejador de contexto.
12. El manejador de contexto traduce el contexto de respuesta al formato de respuesta nativo del PEP. El manejador de contexto retorna la respuesta al PEP.
13. El PEP hace cumplir las obligaciones
14. [No mostrado en la figura] Si el acceso es permitido, entonces el PEP permite el acceso al recurso, en caso contrario deniega el acceso.

2.5.2. Formato de políticas

En esta sección se describe el formato de las políticas de seguridad que propone el protocolo XACML.

Modelo de lenguaje de políticas

La política a aplicar para una solicitud XACML particular podría estar compuesta por múltiples políticas o reglas. Por ejemplo, en una aplicación empresarial, el usuario podría definir políticas personales, mientras que la empresa podría definir sus propias políticas; por lo tanto sería deseable que la decisión de autorización combine estas dos políticas en una política única que aplique a la solicitud.

Es así que XACML define tres niveles anidados de elementos de políticas:

- **<Rule>**: Este elemento contiene una expresión booleana, que puede ser evaluada por sí sola, pero que no será accedida directamente por el PDP. Además, una regla se puede ver como la unidad más elemental de una Política.
- **<Policy>**: Este elemento está formado por un conjunto de elementos de tipo <Rule> y un procedimiento para combinar los resultados de sus evaluaciones. Por lo tanto, se lo puede considerar como la unidad básica a utilizar por el PDP, pudiendo generar una decisión de autorización.
- **<PolicySet>**: Este elemento se compone de un conjunto de <Policy> u otros <PolicySet> y un procedimiento para combinar los resultados de sus evaluaciones. Pudiendo definirlo de manera sencilla como la combinación de políticas independientes en una única política combinada.

En la Figura 8 se puede ver cómo se combinan estos componentes, así como también qué otros elementos los componen.

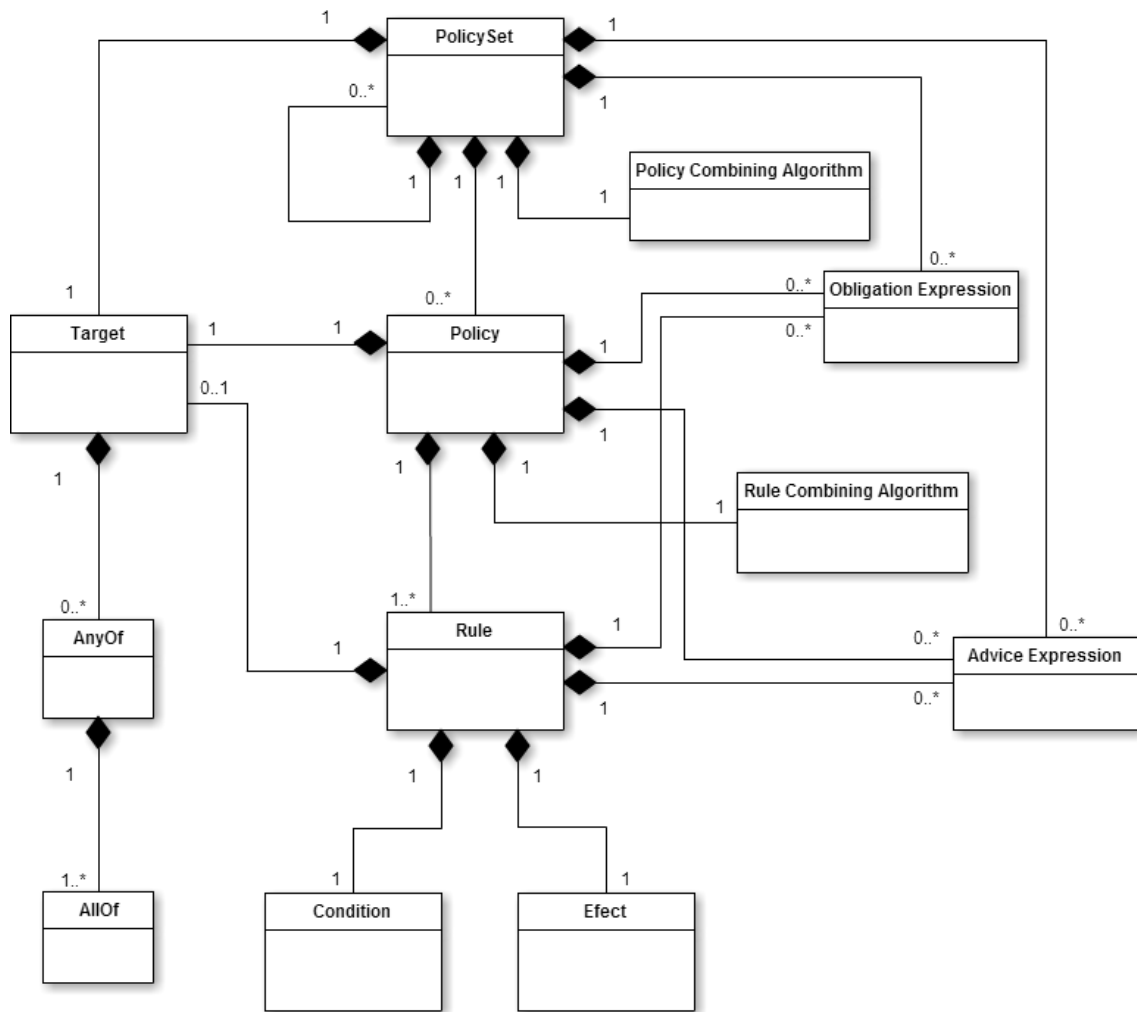


Figura 8 - Modelo de lenguaje de políticas [29]

Una Regla está compuesta principalmente por:

- **Target:** define a qué conjunto de solicitudes va a ser aplicable mediante una expresión lógica basada en atributos de la solicitud. El componente PDP será quien verifique si el target aplica a la solicitud dada. Además, podría darse el caso en que una regla no presente un *target* explícito, en cuyo caso la regla heredaría el target de la política que la contiene.
- **Effect** indica el valor del resultado en caso de que se aplique la regla, pudiendo ser "Permit" o "Deny".
- **Condition:** expresión booleana que complementa al *target* para especificar la aplicabilidad de la regla en la solicitud. La condición no es un elemento obligatorio dentro de la regla.

- **Obligation expressions:** pueden agregarse expresiones de obligación en las reglas, que serán retornadas al PEP cuando el resultado obtenido sea por haber evaluado la regla que las contenía, indicando que cierta condición es requerida.
- **Advice expressions:** pueden agregarse expresiones de sugerencia en las reglas, que serán retornadas al PEP cuando el resultado obtenido sea por haber evaluado la regla que las contenía, indicando un aviso al módulo PEP.

Una política está compuesta por un conjunto de reglas, además de los siguientes principales elementos:

- **Target:** cumple la misma función que el target de una regla, pero a nivel de política.
- **Rule Combining Algorithm:** ver Combinación de algoritmos.
- **Obligation expressions:** análogo a las expresiones de obligación de las reglas, pero a nivel de políticas.
- **Advice expressions:** análogo a las expresiones de sugerencias de las reglas, pero a nivel de políticas.

Un Conjunto de Políticas (*PolicySet*) está compuesto por un conjunto de Políticas u otros *PolicySet*. Además, contiene un **target**, **obligation expressions** y **advice expressions** análogos a los definidos anteriormente y un **Policy Combining Algorithm** detallado en la sección Combinación de algoritmos.

Ejemplo de Política

En la Figura 9 se puede ver un ejemplo de política genérica que incluye los elementos principales descritos anteriormente.

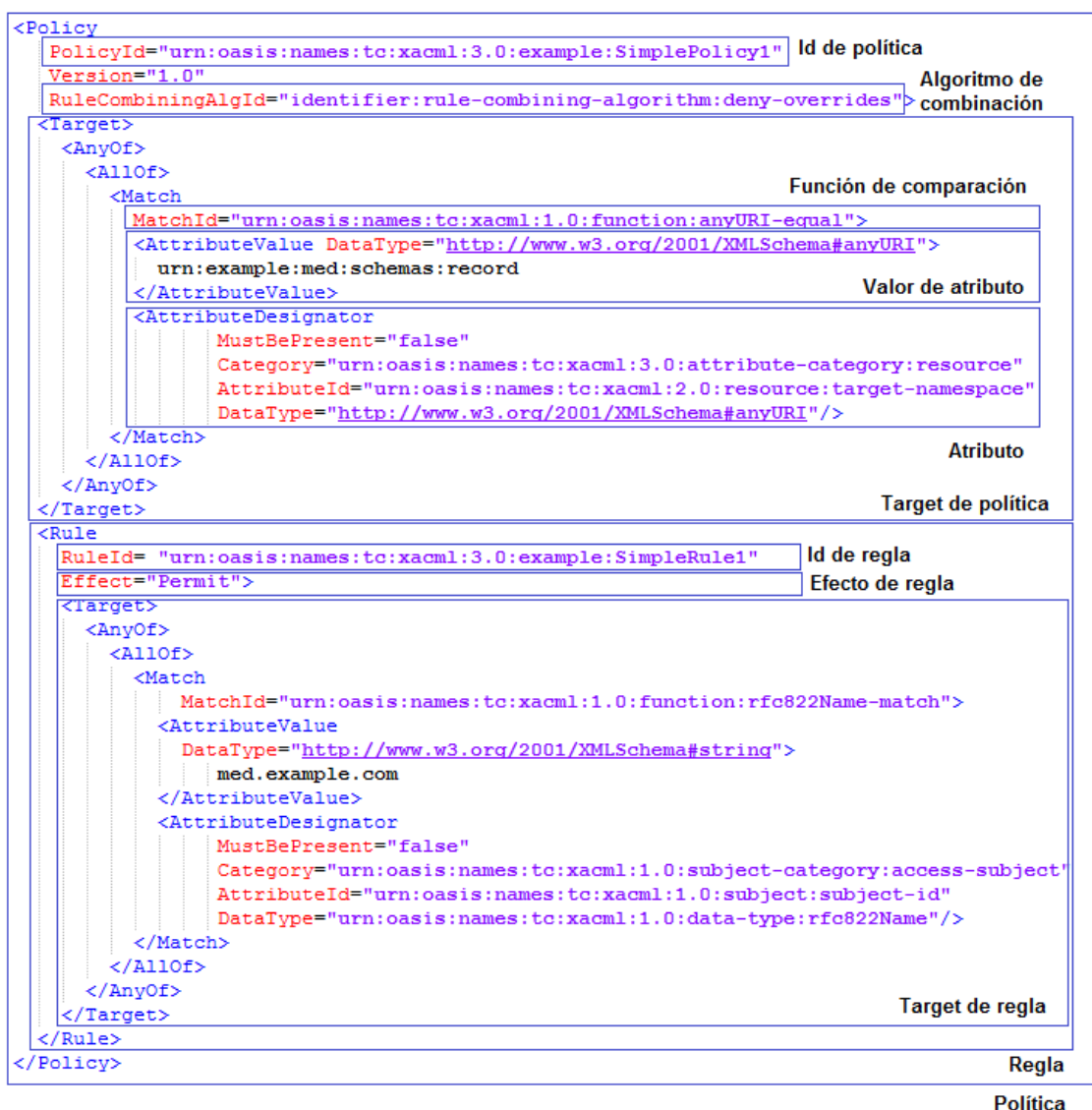


Figura 9 - Ejemplo de política

Las políticas se identifican por su *PolicyId* e indican el algoritmo de combinación de reglas a utilizar para evaluar las reglas asociadas. Luego, se encuentra la sección *Target* que detalla el conjunto de condiciones que se deben cumplir para evaluar la política. En este caso, la política aplica siempre que se solicite un recurso cuyo *namespace* sea igual a *urn:example:med:schemas:record*, de lo contrario se retornará *Not applicable*. Finalmente, la política define una única regla identificada por su *RuleId* y retornará permit en caso de que se solicite *med.example.com* o *Indeterminate* en caso contrario.

Combinación de algoritmos

El protocolo XACML define un conjunto de algoritmos de combinación, tanto para las reglas dentro de una Política (Algoritmo de Combinación de Reglas), como para las políticas dentro de un Conjunto de Políticas (Algoritmo para Combinación de Políticas). El primero de ellos define un procedimiento para obtener una decisión de autorización en base a las evaluaciones individuales del conjunto de Reglas y de manera análoga, el segundo lo hace a partir de evaluaciones de un conjunto de Políticas.

Algunos de los algoritmos definidos son:

- **Deny-overrides:** Si existe una regla o política evaluada como “Deny”, el resultado de la combinación será “Deny”.
- **Permit-overrides:** Si existe una regla o política evaluada como “Permit”, el resultado de la combinación será “Permit”.
- **First-applicable:** El resultado de la combinación será el resultado obtenido al evaluar la primer Regla, Política o Conjunto de Políticas cuyo *target* y *condition* sean aplicables a la solicitud XACML.
- **Only-one-applicable:** En caso de existir una única Política o Conjunto de Políticas que sea aplicable⁵ a la solicitud, se retornará la evaluación de la misma. Por otro lado, si no existiera alguna política aplicable, el resultado será “NotApplicable”, mientras que si se encuentra más de una, el resultado será “Indeterminate”. Cabe destacar que este algoritmo es del tipo Algoritmo para Combinación de Políticas y aplica únicamente a políticas.
- **Deny-unless-permit:** El resultado de la combinación será “Permit” si existe una política aplicable cuya decisión sea “Permit”, de lo contrario se retornará “Deny”.

Además, XACML brinda la posibilidad de definir nuevos algoritmos de combinación si fuese necesario.

⁵ Una Política o Conjunto de Políticas es aplicable a una Solicitud XACML si sus *targets* coinciden.

2.6. Plataforma de Gobierno Electrónico del Estado Uruguayo

La Plataforma de Gobierno Electrónico (PGE) implantada por la Agencia para el Desarrollo del Gobierno de Gestión Electrónica y Sociedad de la Información y del Conocimiento (AGESIC) tiene como objetivo general facilitar y promover la implementación de servicios de Gobierno Electrónico en Uruguay. La plataforma ofrece capacidades de software y hardware, así como también, servicios utilitarios que reducen la complejidad al momento de implementar servicios al público y accesibles dentro del Estado [30].

A nivel tecnológico, la PGE permite que los organismos provean sus funcionalidades de negocio, a través de servicios de software, sin acoplarse a la plataforma en la que fueron desarrollados. Esto corresponde a la implementación de una arquitectura orientada a servicios a nivel del Estado, donde los servicios ofrecidos por los organismos son descriptos, publicados, y descubiertos, además de ser invocados y combinados a través de interfaces y protocolos estandarizados.

Uno de los principales componentes de la PGE es la Plataforma de Interoperabilidad. Ésta cuenta con una Plataforma de Middleware, un Sistema de Seguridad y un Sistema de Gestión de Metadatos.

Plataforma de Middleware: brinda mecanismos para realizar el desarrollo, despliegue e integración de servicios y aplicaciones de manera sencilla. Los organismos hacen uso de esta plataforma para publicar o descubrir servicios, contar con un entorno de ejecución y poder utilizar capacidades de mediación permitiendo desacoplar a los clientes de los mismos.

Sistema de Seguridad: provee servicios de seguridad al resto de los componentes, permitiendo realizar auditorías y facilitando el acceso seguro de los organismos a la plataforma. Los organismos pueden hacer uso de este sistema para controlar el acceso a los servicios que proveen.

Sistema de Gestión de Metadatos: especifica en alto nivel los conceptos relativos a servicios públicos, evitando ambigüedades en el manejo de estos conceptos por parte de los organismos. Esto permite que el intercambio de información se realice de forma estándar.

2.7. Ley de protección de datos personales en Uruguay

En esta sección se presenta la Ley Nº 18.331 - Protección de datos personales y acción "Habeas Data" [31]. Dicha ley estipula que el derecho a la protección de datos es inherente a la persona humana y será extensible a personas jurídicas cuando corresponda. Aplicará a todo conjunto de datos almacenados que lo haga susceptible de tratamiento o posterior divulgación pública o privada, exceptuando aquellas de uso personal, de preservación de la seguridad pública o aquellas creadas por leyes especiales.

La actuación por parte de los responsables de las bases de datos, que contengan datos de terceros (no propios) deberá ajustarse a los siguientes principios:

- **Legalidad:** deberá estar debidamente inscripta, observando que se cumplan los principios y reglamentaciones de la ley antedicha. Además, su finalidad no podrá violar los Derechos Humanos o contrarias a leyes existentes.

- **Veracidad:** los datos que se utilicen deberán ser veraces, adecuados, ecuanímenes y no excesivos sobre la finalidad por la que se obtuvieron. Asimismo, todo dato que se encuentre falso, inexacto o caducado deberá ser eliminado o actualizado.
- **Finalidad:** no se podrán utilizar los datos para finalidades distintas a las previstas. Además no se podrán compartir los datos con otras bases de datos sin mediación de la ley o previo consentimiento del titular.
- **Previo consentimiento informado:** el tratamiento de datos personales es lícito cuando existe un previo consentimiento del titular, de manera libre, expresa e informada y que debe ser documentada. Este principio tendrá excepciones, destacando si se requieren para ejercicio propio de Poderes del Estado u obligación legal o son listados de datos que no se consideran sensibles como son nombres, apellidos, documento de identidad, nacionalidad, domicilio y fecha de nacimiento para las personas físicas.
- **Seguridad de los datos:** el responsable o usuario de los datos deberá garantizar la seguridad y la confidencialidad de los datos personales.
- **Reserva:** quienes obtuvieron legítimamente información proveniente de una base de datos bajo la normativa de protección de datos, deberán utilizarla con total reserva y con la única finalidad para la que se lo fue concedida.
- **Responsabilidad:** el responsable de la base de datos es el responsable de la violación de lo que se dispone en la ley.

Además, esta ley enumera un conjunto de derechos que tiene el titular de los datos, donde se destacan los siguientes:

- **Derecho de acceso:** todo titular de datos, previamente identificado, tendrá derecho a obtener toda su información que se encuentre en las bases de datos tanto públicas como privadas. La información deberá ser brindada en el transcurso de los siguientes cinco días hábiles desde su solicitud y vencido el plazo o si se denegara por razones no justificables, quedará habilitada la acción "Habeas Data". La información brindada debe ser el registro completo del solicitante, de manera clara y entendible, por medio escrito, electrónico, telefónico, de imagen u otro que se considere idóneo para tal fin.
- **Derecho de rectificación, actualización, inclusión o supresión:** todo titular tendrá la posibilidad de modificar, agregar o eliminar alguno de los datos que tenga posesión, cuando se detecte un error, falsedad o exclusión de información.
- **Derechos referentes a la comunicación de datos:** los datos personales podrán ser comunicados únicamente para el cumplimiento de los fines legítimos del emisor y destinatario y con el consentimiento previo del titular. Dicho consentimiento es revocable y no será necesario en las excepciones antes descriptas.

Por otro lado, se definen datos especialmente protegidos:

- **Datos sensibles:** ninguna persona podrá ser obligada a brindar datos sensibles, éstos únicamente se podrán brindar con el consentimiento expreso y escrito del titular. Los datos sensibles sólo pueden ser recolectados y utilizados cuando haya sido autorizado o exista mandato legal que lo autorice. Además, no se podrán formar bases de datos que revelen datos sensibles. Los datos relativos a infracciones penales, civiles o administrativas sólo pueden tratarse por las autoridades públicas competentes, según las leyes y reglamentaciones pertinentes.

Asimismo, se estipula que toda creación, modificación o supresión de bases de datos, tanto de titularidad pública como privada, que almacene datos personales, deberá inscribirse en el registro que establezca el Órgano de Control. Para lo cual deberá principalmente brindar la siguiente información:

- Identificación y responsable de la base de datos.
- Naturaleza de los datos personales que contiene.
- Cómo se obtendrán los datos y qué tratamiento se le dará.
- Descripción técnica de la base de datos, incluyendo medidas de seguridad.
- Cómo se protegerán los datos personales y cómo se permitirán ejercer los derechos.
- Destino de los datos y personas a los que se les puede transmitir información.
- Cuánto tiempo se almacenarán los datos.
- Cómo las personas deberán hacer para ejercer sus derechos (acceder a los datos propios, realizar rectificaciones o actualizaciones) y bajo qué condiciones.

Continuando con las normas de la ley, se definen los siguientes conceptos:

- **Órgano de Control:** se crea la Unidad Reguladora y de Control de Datos Personales, la cual será un órgano independiente de la Agencia para el Desarrollo del Gobierno de Gestión Electrónica y la Sociedad de la Información (AGESIC), conteniendo la más amplia autonomía técnica.
El órgano de control deberá realizar las acciones necesarias para cumplir con lo estipulado con la ley Nº 18.331, para lo cual deberá, entre otras cosas, mantener el registro de las bases de datos abarcadas en esta ley y controlar el cumplimiento de las normas, pudiendo realizar fiscalizaciones e inspecciones que se crean convenientes.
- **Acción de protección de datos personales - Habeas Data:** Habeas Data es la acción judicial, la cual toda persona tiene derecho a ejecutar en caso de estar frente a incumplimientos antes mencionados.

3. Análisis de requerimientos

En esta sección, se establece el marco de trabajo sobre el cual se basa el proyecto. Además, se realiza una interpretación de la ley de protección de datos personales y el análisis de requerimientos correspondiente.

3.1. Marco de trabajo

Como punto de partida del análisis, se toma el contexto establecido en el artículo. Éste establece una arquitectura de referencia combinando una plataforma de gobierno electrónico con una solución de MDM.

La plataforma de gobierno electrónico presenta las siguientes características principales:

- Brinda a los organismos la posibilidad de interactuar entre ellos, permitiendo realizar operaciones de manera conjunta.
- Permite a los ciudadanos acceder a sus datos personales.
- Acuerda estructuras comunes para compartir la información con los organismos.
- Cuenta con cuerpos regulatorios, como pueden ser agencias supervisoras.
- Utilizan plataformas de integración.
- Autentica e identifica el origen de las interacciones.

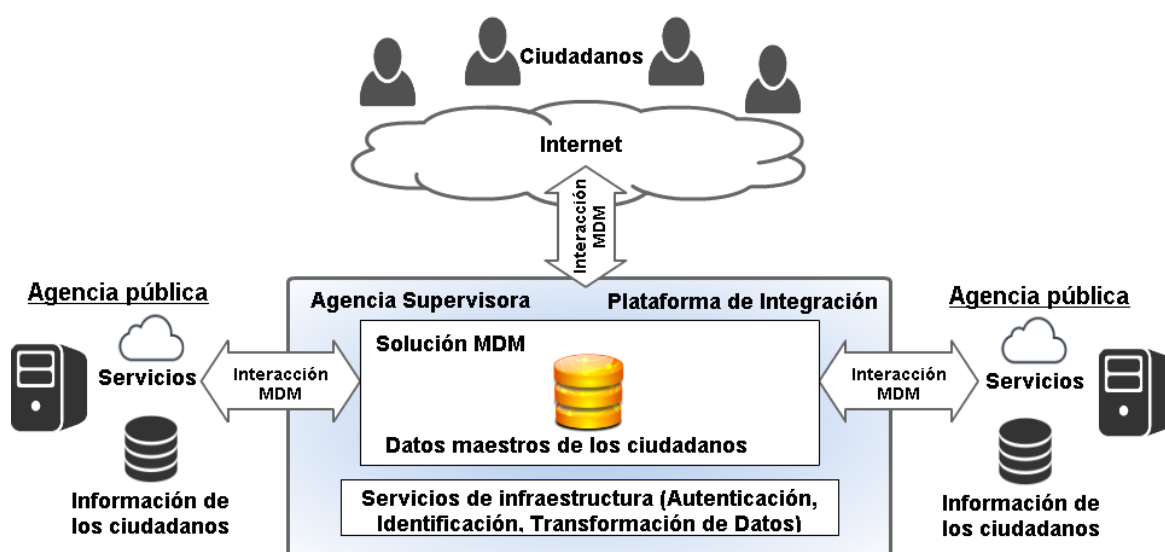


Figura 10 - Contexto de MDM en gobierno electrónico [Adaptación de [1]]

Como se puede observar en la Figura 10, los organismos y los ciudadanos se comunican a través de interacciones MDM con la solución de Master Data Management.

Todas las interacciones entre organismos deben pasar a través de la plataforma de integración, la cual tiene capacidades de:

- **Mediación:** Permite filtrar, modificar y direccionar las interacciones según corresponda.
- **Transformación de datos:** Permite transformar las interacciones recibidas, logrando implementar el patrón Modelo de Datos Canónico.

- **Seguridad:** Autentica e identifica el origen de las interacciones, así como también verificando su integridad y no repudio.

Además, es responsable de alojar la solución de MDM, permitiendo que el acceso a los datos de los ciudadanos se realice bajo las normas de regulación correspondientes.

Las interacciones MDM entre la plataforma de integración y los organismos varían según el estilo de MDM implementado. Éstas se identifican como operaciones de mantenimiento, sincronización y solicitud de datos maestros.

MDM provee una vista actualizada de los datos maestros a través de servicios. Estos servicios pueden ser utilizados, tanto por la plataforma de integración, como por los organismos. Además, se basan en mecanismos estándares para el intercambio de mensajes, como son el protocolo SOAP y el conjunto de estándares WS-*. Asimismo, la solución de MDM puede requerir de servicios expuestos por los organismos para brindar el registro completo de los datos solicitados.



Figura 11 - Ejemplo emisión pasaporte 1

A modo de ejemplo, como se puede ver en la Figura 11, la Dirección Nacional de Identificación Civil (DNIC) podría tener la necesidad de consultar a la plataforma de integración para obtener los antecedentes de un ciudadano que esté tramitando su pasaporte. Esto requiere que la DNIC inicie una interacción MDM, respetando el acuerdo en las estructuras con la Agencia Supervisora. La plataforma se encarga de autenticar e identificar el origen de la interacción y solicitar al sistema de MDM la información correspondiente, quien debe obtener los antecedentes actualizados del ciudadano. Considerando que el dueño de estos datos es la Dirección Nacional de Policía Científica (DNPC), el sistema de MDM deberá generar una nueva interacción invocando a un servicio expuesto por dicho organismo que retornará los antecedentes solicitados.

3.2. Interpretación de la ley de protección de datos de Uruguay

En esta sección se realiza una interpretación y aplicación al proyecto de la ley descrita en la sección 2.7.

En primer lugar, se define como principal elemento al **ciudadano**, utilizando, para el presente proyecto, los derechos establecidos únicamente para la persona física. Posteriormente, se puede extender a personas jurídicas, como señala la ley.

El ciudadano es dueño de un conjunto de **datos personales** que lo describen e identifican, pudiendo ser éstos de carácter tanto **público** como **sensible**. Los primeros, tienen la peculiaridad de que toda persona, debidamente identificada podrá acceder a ellos, mientras que los últimos no. Es así que, todo **organismo** u organización, que desee almacenar o consultar algún dato sensible, debe contar con un **consentimiento** explícito para almacenar o compartir, otorgado por el ciudadano dueño del dato.

Asimismo, a partir de lo establecido en el artículo, se desprende la necesidad de utilizar un **propósito** para el cual se brinda el consentimiento. Éste representa la finalidad para la que se utilizará el dato. Además, para especificar el tipo de consentimiento (almacenar o compartir) se incorpora la **acción**, permitiendo así soportar estos y otros tipos de consentimientos que puedan surgir.

Por otro lado, existen excepciones a la hora de consultar datos sensibles que no requieren un consentimiento previo. Tal es el caso de obligaciones legales o tareas de Poderes del Estado.

En la Figura 12 se presenta un diagrama con las relaciones entre los conceptos principales obtenidos a partir de la ley.

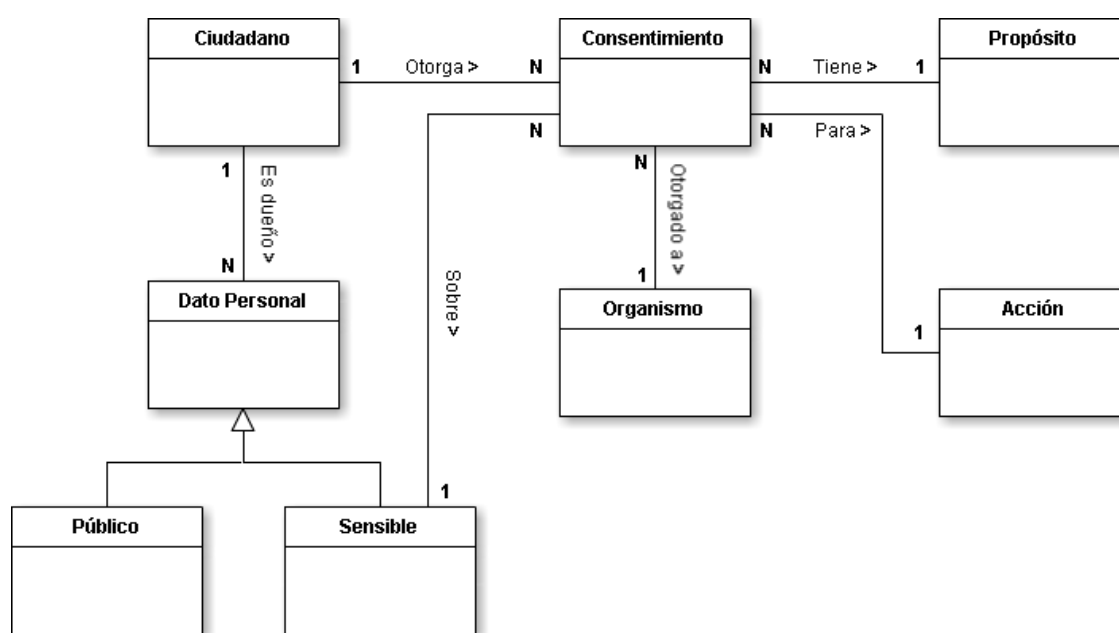


Figura 12 - Diagrama de consentimientos

Cabe destacar que los organismos a los que se hace referencia deben estar previamente inscriptos en el Órgano de Control, siendo en este caso la Agencia Supervisora, que en el caso de Uruguay es AGESIC.

Todo ciudadano podrá solicitar ver los datos de los que es dueño y rectificarlos en caso de que se detecten irregularidades. Además, los consentimientos pueden ser consultados y revocados siempre que se considere necesario.

Por otro lado, en circunstancias de incumplimiento, un organismo puede ser objeto de la acción “Habeas Data”. Esto implica, circunstancialmente, la penalización de su base de datos.

Volviendo al ejemplo de emisión de pasaporte e identificando a los antecedentes del ciudadano como dato sensible, para que la DNIC pueda acceder a la información solicitada, debe contar con un consentimiento. El mismo debe ser explícito, informado, otorgado por el individuo y registrado en la plataforma. Dicho consentimiento debe habilitar al organismo antes mencionado el acceso a los antecedentes cuando la Acción es *consultar* y el Propósito es *emitir pasaporte*.

3.3. Especificación de requerimientos

A partir del marco de trabajo definido y mediante la inclusión de las normativas establecidas en la Ley Nº 18.331, se identificaron los siguientes requerimientos generales que debe cumplir la plataforma:

- Monitorear y validar todas las interacciones MDM que se realicen a través de ella, por ejemplo, cuando se mantienen, sincronizan o solicitan datos maestros.
- Permitir gestionar los consentimientos provistos por los ciudadanos.
- Brindar a los ciudadanos una interfaz para administrar sus datos personales.
- Mantener el registro de las bases de datos abarcadas en la ley de protección de datos.
- Encargarse de proteger los datos de los ciudadanos que gestione, no revelando información a terceros no autorizados.

La plataforma de integración debe ser completa, es decir, los componentes requeridos para la validación de interacciones deben ubicarse dentro de la plataforma.

A su vez, se identifican los principales actores del sistema:

- **Administrador:** Es el encargado de las configuraciones generales del sistema.
- **Ciudadano:** Es quien brinda consentimientos y administra sus datos personales.
- **Organismo:** Es quien administra datos personales.

De esta forma, se establecen las funcionalidades que les deben ser provistas a los distintos actores del sistema, como se muestra en la Figura 13.

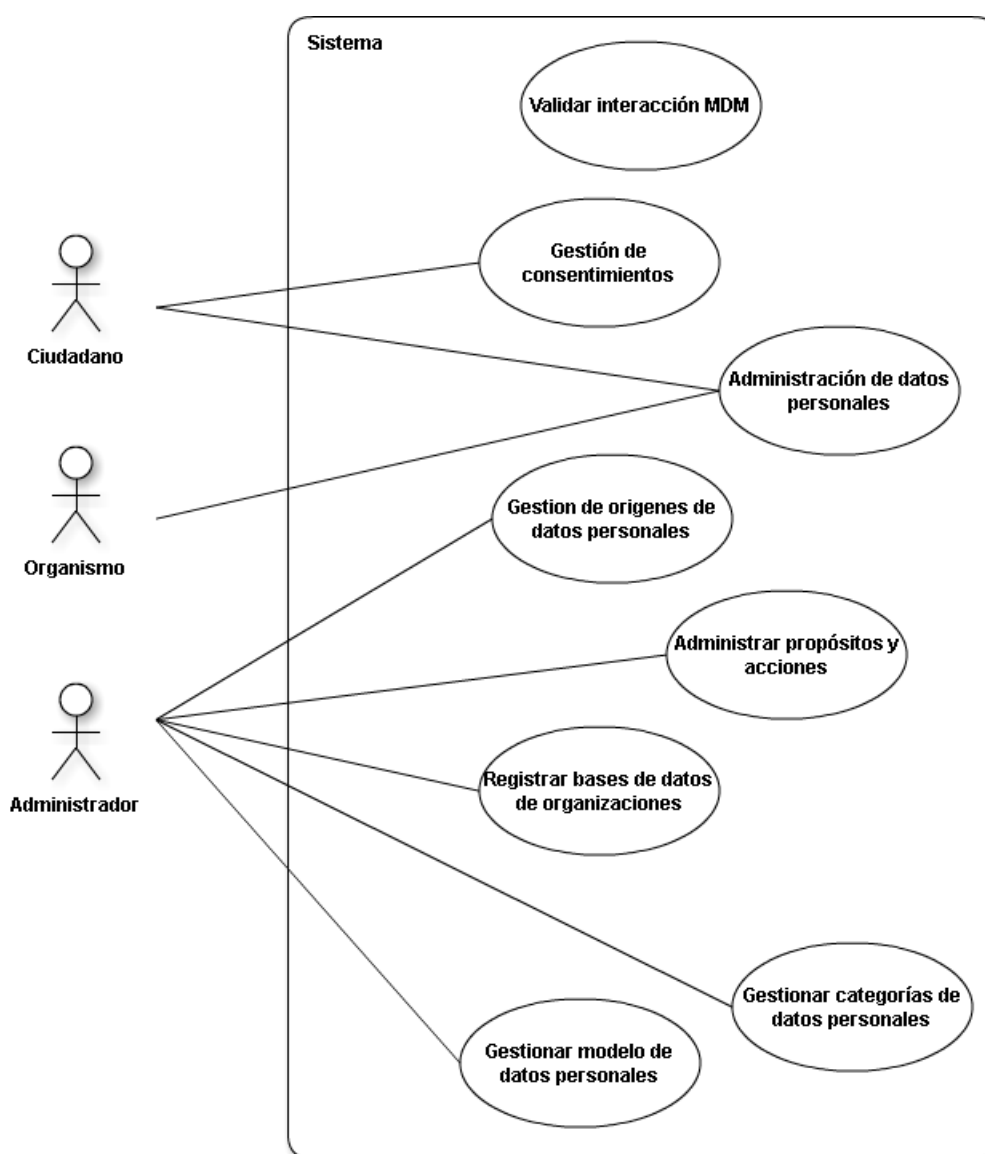


Figura 13 - Diagrama de casos de uso

3.3.1. Gestión de consentimientos

El sistema debe permitir que todo ciudadano pueda crear un consentimiento para un determinado Propósito, Sujeto (solicitante), Acción y Entorno sobre un conjunto de datos de su propiedad (recursos).

Además, se debe permitir al ciudadano administrar (consultar, modificar y revocar) sus consentimientos.

3.3.2. Administración de datos personales

El sistema debe permitir a los ciudadanos consultar, insertar, modificar y suprimir sus datos personales.

3.3.3. Gestión de orígenes de datos personales

El sistema debe permitir registrar, modificar y eliminar orígenes proveedores de datos personales.

3.3.4. Administrar propósitos y acciones

El sistema debe permitir crear, modificar y eliminar tanto propósitos como acciones.

3.3.5. Registrar bases de datos de organizaciones

El sistema debe permitir registrar bases de datos de organizaciones habilitadas para almacenar datos de los ciudadanos, según lo estipulado en la ley.

3.3.6. Gestión de categorías de datos personales

El sistema debe permitir indicar cuáles son los datos personales sensibles y cuáles de acceso público.

3.3.7. Gestión de modelo de datos personales

El sistema debe permitir modificar el modelo de datos de persona que represente al modelo de datos canónico.

3.3.8. Validar interacción MDM

El sistema debe interceptar todas las interacciones MDM y verificarlas a partir de los consentimientos definidos por los ciudadanos y las normas establecidas en la Ley N° 18.331.

3.4. Selección de estilo de solución de MDM

Analizando el marco de trabajo y la Ley N° 18.331, se decidió utilizar el estilo de solución de MDM Híbrido, considerando los siguientes puntos:

1. El conjunto de datos críticos de los ciudadanos debe incluirse dentro de una solución de MDM estilo Registro, siendo administrados por los organismos responsables. Esto evita almacenar duplicados de información sensible.
2. El conjunto de datos de acceso público debe incluirse dentro de una solución de MDM estilo Repositorio, siendo gestionados por la plataforma de integración. Esto permite brindar la información pública de los ciudadanos de manera rápida y eficaz.
3. Se debe abarcar la mayor cantidad de escenarios posibles para mantener la generalidad de la solución.
4. Los estilos de solución de MDM Repositorio y Registro son casos particulares del estilo Híbrido.

3.5. Análisis de interacciones MDM

A partir del artículo se obtienen los siguientes propósitos según la intención de la interacción MDM que se realiza:

- **MDM-Maintain:** siempre que la interacción sea para realizar mantenimiento de datos maestros.
- **MDM-Synchronize:** siempre que la interacción sea para sincronizar datos maestros entre el repositorio y las fuentes de datos externas.
- **MDM-BusinessOriented:** siempre que la interacción sea para consultar datos maestros.

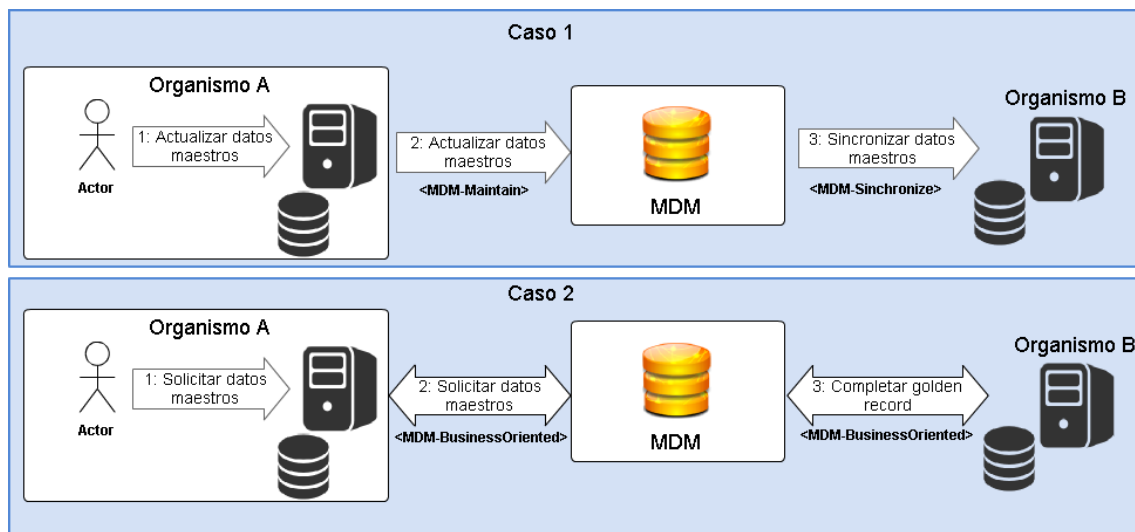


Figura 14 - Interacciones MDM

En la Figura 14 se muestran dos ejemplos que permiten ilustrar los distintos tipos de interacción MDM utilizando el estilo Híbrido como la combinación de los estilos Repositorio Centralizado y Registro.

En el Caso 1, un actor del Organismo A utiliza una aplicación para actualizar los datos maestros. Aplicando el estilo Repositorio Centralizado, la actualización se realiza en el repositorio de datos maestros, mediante una interacción con propósito *MDM-Maintain*. Luego, el sistema de MDM se encarga de sincronizar la nueva información con el Organismo B mediante una interacción con propósito *MDM-Synchronize*.

Cabe destacar que si se utilizara el estilo Repositorio Consolidado, la actualización de datos maestros representaría una sincronización, realizando una interacción con propósito *MDM-Synchronize*.

En el Caso 2, un actor del Organismo A utiliza una aplicación para solicitar datos maestros. Para esto, realiza la solicitud al repositorio de datos maestros mediante una interacción con propósito *MDM-BusinessOriented*. Aplicando el estilo Registro, el sistema de MDM solicita la información al Organismo B mediante una interacción con propósito *MDM-BusinessOriented*.

En resumen, se identifican las siguientes interacciones MDM con sus respectivos propósitos:

- Interacciones de entrada⁶ desde organizaciones o ciudadanos
 - *MDM_Maintain* (Inserción, Actualización o Eliminación): Toda aplicación que intente agregar, modificar o eliminar datos, se comunica con el sistema de MDM, quien realiza los cambios en el repositorio y luego los propaga a los orígenes correspondientes.
 - *MDM_BusinessOriented* (Consulta): Toda aplicación que requiera consultar datos actualizados y completos se comunicará con el sistema de MDM, quien resuelve este pedido completando los datos faltantes con información de los orígenes convenientes.
- Interacciones de entrada para sincronización ante cambios detectados en base de datos de organizaciones:
 - *MDM_Synchronize*: Toda aplicación que ejecute operaciones de adición, modificación o eliminación en la base de datos de una organización, sobre elementos sincronizados con el repositorio, provoca una replicación de los cambios hacia el sistema de MDM.
- Interacciones de salida⁷
 - *MDM_Synchronize*: Luego de haber recibido una interacción con propósito *MDM_Maintain* o *MDM_Synchronize*, el sistema de Master Data Management envía los cambios correspondientes para actualizar las bases de datos de las organizaciones registradas.
 - *MDM_BusinessOriented* (Consulta): Cuando se requieren datos que no se encuentran en el repositorio central, se genera una interacción con propósito *Business Oriented* para consultarlos en la fuente de información adecuada.

Dentro del marco de la consulta de antecedentes de un ciudadano, la interacción inicial generada por la DNIC se clasifica como una interacción de entrada de tipo *MDM_Business Oriented*, mientras que la interacción generada por el sistema de MDM hacia la DNPC se corresponde con una interacción de salida del mismo tipo. Ahora bien, si se considera que la Dirección General de Información e Inteligencia (DGII) también almacena los antecedentes de los ciudadanos, es de esperar que si existe un consentimiento para compartir los datos, éstos sean sincronizados por el sistema de MDM con la DNPC. De esta forma, cuando se registra un nuevo antecedente en la DGII, se genera una interacción de entrada de tipo *MDM_Synchronize* y posteriormente otra de salida del mismo tipo hacia la DNPC.

⁶ Interacciones con destino al sistema de MDM

⁷ Interacciones con origen al sistema de MDM

3.6. Análisis de consentimientos sobre la Agencia Supervisora

La Agencia Supervisora no deja de ser una organización, por lo cual, debe contar con los consentimientos correspondientes para almacenar o distribuir los datos personales. Es así que se analizan dos posibilidades:

1. A la Agencia Supervisora se la asignan permisos totales sobre los datos, es decir, existe un consentimiento implícito para el uso o distribución de los datos personales de cualquier ciudadano para con la Agencia.
2. A la Agencia Supervisora se la trata como una organización más, lo que implica que se debe validar que exista un consentimiento para todo atributo involucrado en una interacción MDM que pase por ésta. En caso de no existir dicho consentimiento, los datos involucrados deben estar encriptados.

No obstante, no existe una manera de corroborar programáticamente que un texto se encuentra encriptado. Por lo tanto, no es posible garantizar el cumplimiento de los consentimientos de forma completa en el segundo caso.

3.7. Resumen y conclusiones del análisis

El análisis de requerimientos se centró en los siguientes elementos de la Ley N° 18.331:

- Datos especialmente protegidos
 - **Datos sensibles:** se considera que todo ciudadano tiene un conjunto de datos personales clasificados como sensibles, éstos únicamente se podrán brindar con el previo consentimiento del titular. Este elemento determinó la incorporación de la Gestión de categorías de datos personales y la Gestión de consentimientos.
- Derechos del titular de los datos
 - **Derecho de acceso:** los ciudadanos pueden solicitar obtener toda su información que se encuentre en las bases de datos de los organismos.
 - **Derecho de rectificación, actualización, inclusión o supresión:** los ciudadanos pueden solicitar estas acciones en caso de detectar irregularidades en alguno de los datos que le pertenecen. Este derecho, en conjunto con el anterior, determinaron la incorporación de la Administración de datos personales para los ciudadanos.
 - **Derechos referentes a la comunicación de datos:** los datos podrán ser comunicados únicamente para el cumplimiento del fin indicado por el emisor y con el consentimiento previo del titular. Estos derechos implicaron la incorporación de la Administración de propósitos y acciones así como también la Validación de todas las interacciones MDM que se realicen.

A su vez, se considera la existencia de penalizaciones a organismos que limitan sus interacciones con la plataforma a partir de la ejecución de la Acción de protección de datos personales – Habeas Data.

La ley establece que toda base de datos que contenga información personal de ciudadanos debe ser registrada en el Órgano de control. Esto implicó la incorporación del Registro de bases de datos de organizaciones.

Asimismo, para poder construir la solución de MDM, permitiendo la comunicación con los sistemas que mantienen información de los ciudadanos, resultó necesaria la Gestión de orígenes de datos personales.

La plataforma de gobierno electrónico permite alojar la solución MDM, autenticar el origen de las interacciones e implementar el patrón Modelo de Datos Canónico.

Por otro lado, se determinó que la solución de MDM estilo Híbrido resulta ser la más adecuada para el desarrollo del proyecto, ya que cuando se considera este estilo, se están considerando también los estilos Registro, para almacenar atributos sensibles y Repositorio, para almacenar atributos públicos. Esto permite evitar duplicación de información sensible y obtener la información pública de los ciudadanos de manera rápida y eficaz. Además, se abarcan la mayor cantidad de escenarios posibles manteniendo la generalidad de la solución. Por lo tanto, la etapa de diseño se basa en el estilo Híbrido.

4. Propuesta de solución

En este capítulo se describe la solución propuesta brindando una especificación de los componentes de la arquitectura, así como también de sus responsabilidades e interacciones. Los principales desafíos que se tuvieron que enfrentar al realizar este diseño fueron: la intercepción de las interacciones MDM y el diseño de la estructura que permite almacenar y evaluar los consentimientos de los ciudadanos.

El capítulo se organiza en 4.1 Descripción general donde se describe la solución propuesta en alto nivel, 4.2 Componentes de la arquitectura de la solución donde se presenta en detalle los distintos componentes diseñados. Luego, en 4.3 Arquitectura de la solución se muestra la combinación de los componentes antes presentados, en 4.4 Interacciones MDM se detalla el formato de las interacciones MDM. Finalmente, en 4.5 Diseño y evaluación de políticas se describe cómo se utiliza el protocolo XACML en la solución y en 4.6 Interacción de componentes de la arquitectura se describen las principales interacciones entre los componentes de la solución.

4.1. Descripción general

En la Figura 15 se presenta un diagrama con la arquitectura en alto nivel de la solución propuesta, donde se incluyen los elementos agregados al marco de trabajo presentado resaltados con (*).

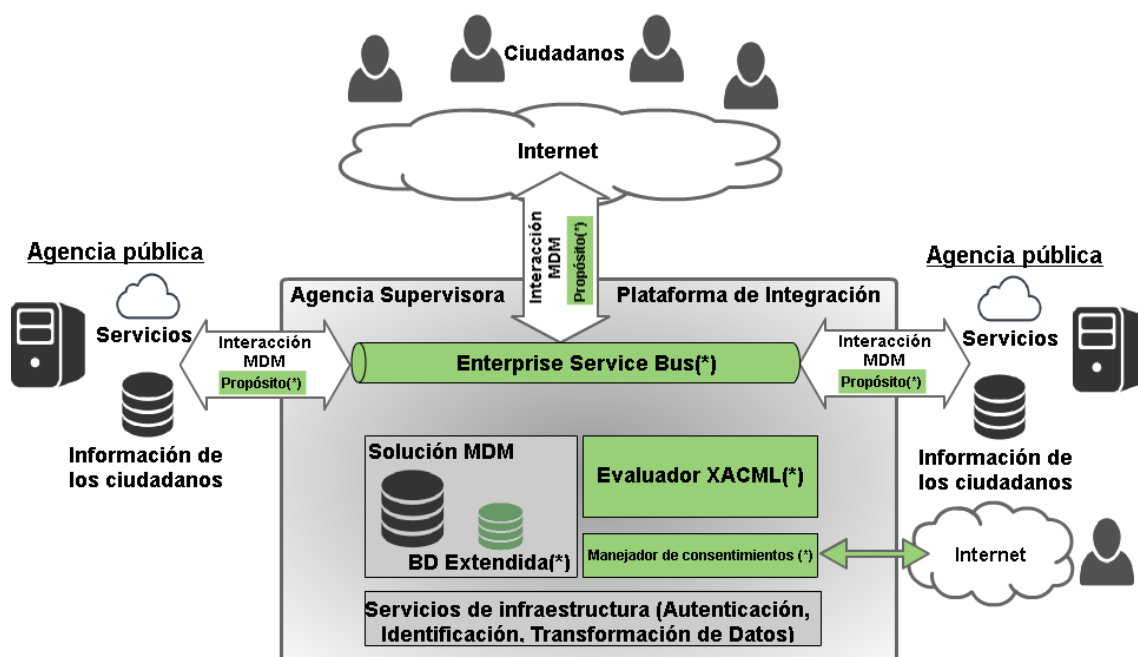


Figura 15 - Diagrama de componentes incorporados en la solución [Adaptación [1]]

- Para interceptar y validar todas las interacciones MDM que pasan a través de la plataforma se incorpora un *middleware* Enterprise Service Bus. Además, estas interacciones deben incluir el propósito que indica la intención de la interacción.
- Para gestionar los consentimientos de los ciudadanos se incorpora el Sistema de Manejo de Consentimientos.
- Para evaluar las interacciones MDM y almacenar los consentimientos de los ciudadanos se incorpora el sistema Evaluador XACML.

- Para mantener el registro de las bases de datos que almacenan datos personales y todo aquel dato que no pertenezca al esquema de datos maestros, se incorpora la BD Extendida.

Se asume que los Servicios de Infraestructura son brindados por la Plataforma de Integración. Estos servicios son los encargados de controlar los aspectos relacionados con la seguridad, como ser, la identificación de los solicitantes, integridad del mensaje, el no repudio y la confidencialidad.

4.2. Componentes de la arquitectura de la solución

En esta sección se detallan los componentes que se incorporan en la arquitectura.

Con el objetivo de prevenir el acceso no autorizado, el ESB brinda servicios virtuales intermediarios para controlar el uso de las interfaces del sistema de MDM. Éstos se comunican con el sistema Evaluador, quien valida las interacciones MDM utilizando las políticas definidas a partir de los consentimientos de los ciudadanos.

Con el objetivo de considerar el conjunto de interacciones MDM definidas, el repositorio de datos maestros cuenta al menos con dos interfaces: una de sincronización y otra de consulta.

En primer lugar, en la Figura 16, se muestra el detalle de los elementos necesarios para implementar la funcionalidad de consulta de datos maestros.

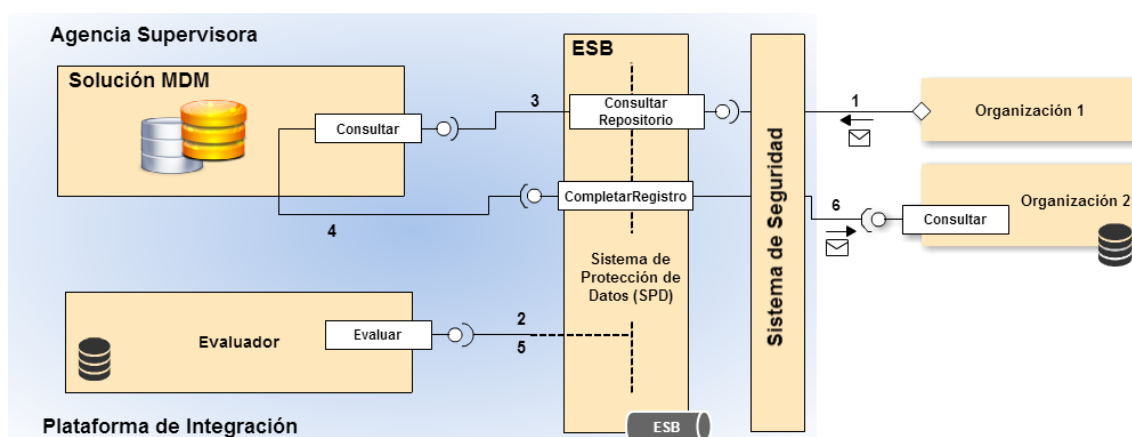


Figura 16 - Arquitectura con servicios de consulta

Cuando una organización (Organización 1) realiza una consulta de datos maestros (1), ésta es interceptada por el servicio virtual Consultar Repositorio provisto por el ESB (SPD). Éste transforma el pedido en un *XACML Request* y lo envía al Evaluador (2). El componente Evaluador realiza la validación del pedido según las reglas definidas y retorna al SPD el *XACML Response* incluyendo los atributos admitidos y los denegados. A continuación, el SPD filtra el pedido original según la respuesta obtenida y redirige su curso al servicio Consultar de la solución de MDM (3).

Si la solución de MDM requiere información adicional para completar el *golden record*, debe solicitarla al organismo correspondiente (Organización 2). Esta solicitud es interceptada por el servicio virtual Completar Registro (4), donde el ESB solicita al Evaluador la validación de la interacción (5). Luego, habiendo tomado las acciones correspondientes, redirige su curso al servicio Consultar de la Organización 2 (6). Finalmente, la solución de MDM retorna la información solicitada.

Por otro lado, en la Figura 17, se muestra el detalle de los elementos necesarios para implementar la funcionalidad de actualización de datos maestros.

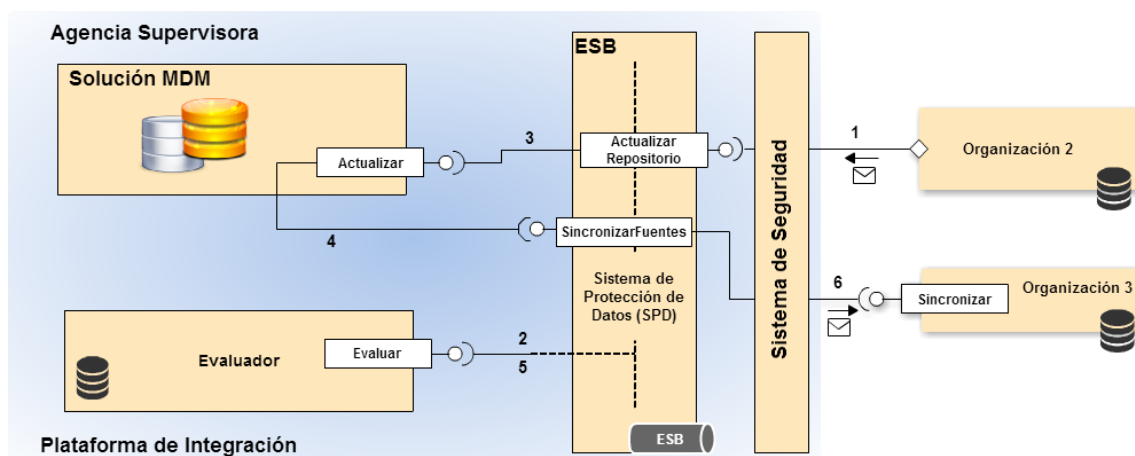


Figura 17 - Arquitectura con servicios de actualización

Cuando una organización (Organización 2) realiza una actualización de datos maestros (1), ésta es interceptada por el servicio virtual Actualizar Repositorio provisto por el ESB (SPD). Éste se comunica con el servicio Evaluar del Evaluador (2) de forma análoga al caso anterior. A continuación, el SPD filtra el pedido original según la respuesta obtenida y redirige su curso al servicio Actualizar de la solución de MDM (3).

Al existir una organización (Organización 3) cuya base de datos está en sincronía con la solución de MDM, se propaga la actualización a dicha organización. Ésta es interceptada por el servicio virtual Sincronizar Fuentes (4), donde el ESB solicita al Evaluador la validación de la interacción (5). Luego, habiendo tomado las acciones correspondientes, redirige su curso al servicio Sincronizar de la Organización 3 (6).

4.2.1. Master Data Management

Se asume que el sistema de gestión de datos maestros es brindado por el marco de trabajo. Éste es el responsable de la administración de los datos maestros que contienen información de los ciudadanos, así como también de las bases de datos registradas de los organismos. A su vez, es posible realizar la configuración del esquema de datos necesario en el repositorio de datos maestros.

El sistema de MDM expone servicios SOAP que permiten realizar consultas de los datos maestros. Para ello, puede comunicarse con servicios de consulta brindados por los organismos, logrando obtener el *golden record*.

Además, cuenta con servicios que implementan las operaciones de mantenimiento de los datos maestros. Éstos pueden requerir actualizar las bases de datos de los organismos, a través de servicios adicionales brindados por ellos.

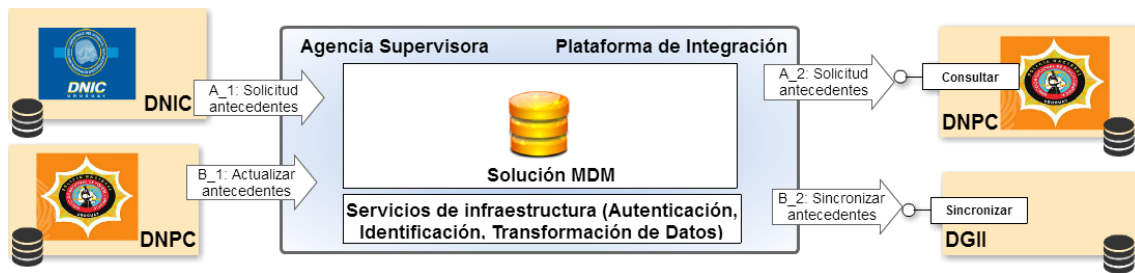


Figura 18 - Ejemplo emisión pasaporte 2

En el ejemplo de la consulta de antecedentes para la emisión de pasaporte, la base de datos de la DNPC se encuentra registrada en el sistema de MDM y, como lo muestra la Figura 18, ofrece un Servicio Web de consulta (Consultar) que brinda los antecedentes de un ciudadano. Ahora bien, si se pretende realizar la sincronización con la DGII, su base de datos debe encontrarse registrada en el sistema de MDM y deberá brindar un servicio de actualización (Sincronizar) de los datos considerados.

4.2.2. Administrador y evaluador de políticas XACML

Los módulos del Evaluador siguen el protocolo XACML definido anteriormente. En la Figura 19 se especifican las interacciones entre sus módulos y la integración del componente en la solución.

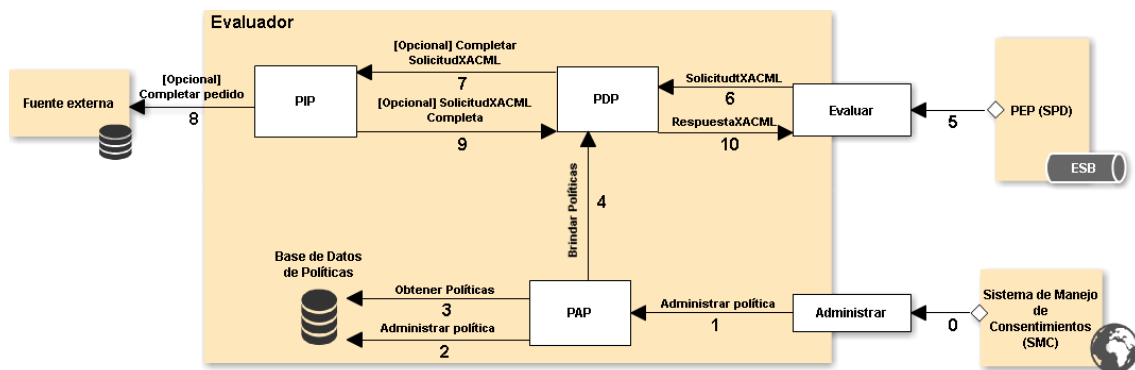


Figura 19 - Componente Evaluador

Administración de políticas

El Sistema de Manejo de Consentimientos (SMC) permite gestionar las políticas de los ciudadanos a través de la interfaz Administrar (0) y enviarlas al módulo PAP (1).

El módulo PAP implementa la lógica necesaria para administrar las políticas recibidas y almacenarlas en su Base de Datos de Políticas (2). Además, es responsable de brindar las políticas almacenadas al módulo PDP (3)(4).

Evaluación de solicitudes

El módulo PEP envía las solicitudes XACML al Evaluador a través de la interfaz Evaluar (5). Este módulo es implementado siguiendo el patrón ESB y es descrito en la sección Enterprise Service Bus (ESB).

El módulo PDP se encarga de procesar las solicitudes XACML recibidas por el módulo PEP (6). Para ello, valida la solicitud a partir de las políticas definidas. En caso de requerir información extra para completar el contexto de la solicitud se invoca al módulo PIP (7).

El módulo PIP se conecta con sistemas de información externos (8), logrando así completar el contexto de la solicitud XACML (9).

Finalmente, el módulo PDP retorna la Respuesta XACML (10).

4.2.3. Sistema de Manejo de Consentimientos (SMC)

Este componente consta de una interfaz gráfica destinada a los ciudadanos para permitir que administren sus consentimientos. Éstos son traducidos a un conjunto de reglas y políticas, que luego se incluyen dentro del componente Evaluador utilizando la interfaz Administrar.

El SMC debe determinar los posibles consentimientos que puede brindar un ciudadano considerando:

- El conjunto de Acciones y Propósitos definidos.
- El modelo de datos personales.
- Los organismos registrados.

4.2.4. Enterprise Service Bus (ESB)

Este componente es un *middleware* implementado siguiendo el patrón ESB. El mismo se encarga de interceptar las interacciones MDM, para transformarlas en solicitudes XACML y enviarlas al componente Evaluador. Luego, filtra el mensaje original, mediante la respuesta obtenida, para finalmente transmitirlo a su destino.

En la Figura 20 se presentan los servicios diseñados para este componente.

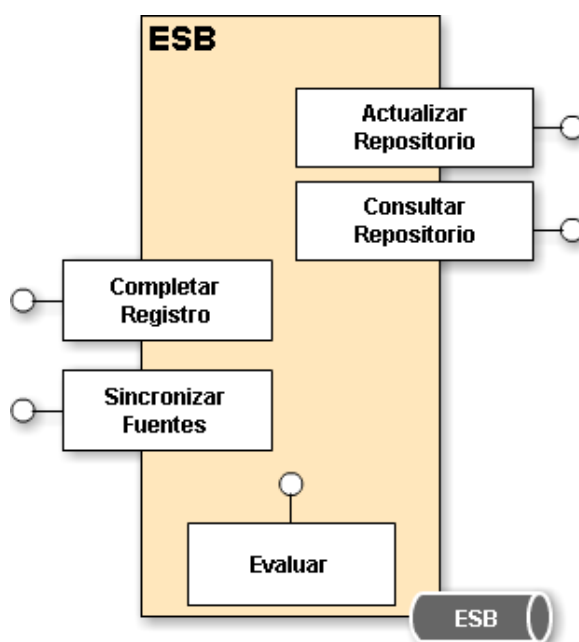


Figura 20 - Servicios del componente ESB

Los servicios de Actualizar y Consultar Repositorio permiten a los organismos acceder a los datos maestros del sistema de MDM. Asimismo, los servicios de Completar Registro y Sincronizar Fuentes permiten que el sistema de MDM acceda a servicios de organismos para enviar o recibir datos maestros. Además, el servicio Evaluar permite invocar al componente Evaluador para evaluar una solicitud XACML.

Servicio Evaluar

El servicio de evaluación de solicitudes XACML es utilizado por el resto de los servicios del ESB y permite validar una interacción MDM de acuerdo al protocolo XACML. Para ello, transforma la interacción en una solicitud XACML (1) e invoca el servicio Evaluar del componente Evaluador (2) como lo muestra la Figura 21.

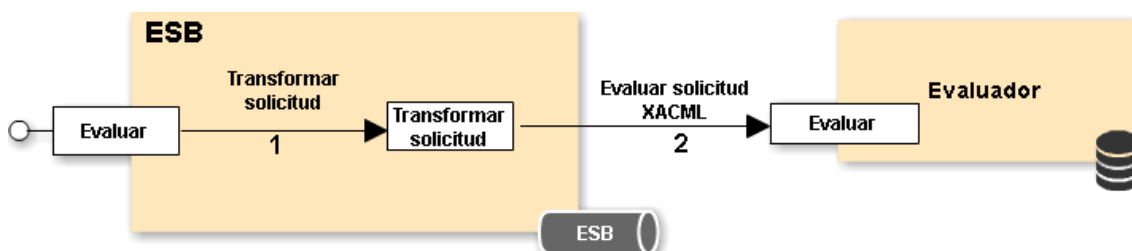


Figura 21 - Servicio Evaluar

Servicio Consultar Repositorio

El servicio es utilizado por aplicaciones de organismos y mediante el envío de una interacción MDM permite consultar datos maestros. Para ello, verifica si el organismo que consulta tiene consentimientos para consultar los datos solicitados (1), filtra la interacción eliminando los atributos de la solicitud que no fueron admitidos, es decir, de aquellos que no se obtuvo como respuesta del servicio Evaluar una decisión *Permit* (2). Finalmente, envía la interacción filtrada al servicio Consultar del sistema de MDM (3) como lo muestra la Figura 22.

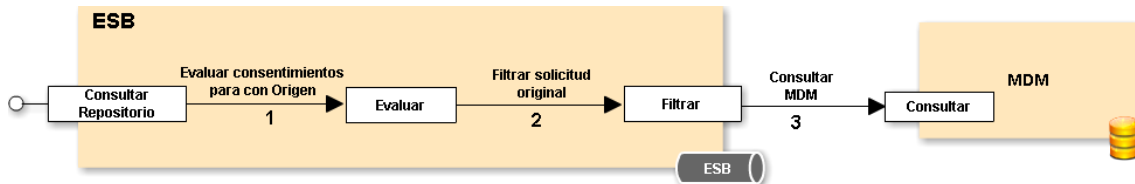


Figura 22 - Servicio Consultar Repositorio

Servicio Completar Registro

El servicio es utilizado por el sistema de MDM y permite consultar datos maestros de un organismo. Para ello, se verifica que el organismo destino cuente con los consentimientos necesarios para compartir los datos de la interacción (1). En caso de que no cuente con dichos consentimientos, se descarta la solicitud. De lo contrario, se verifica que la Agencia cuente con los consentimientos necesarios para visualizar los datos consultados (3). Esta verificación determina si la Agencia puede manejar en texto claro los datos solicitados. Es así que, si la Agencia no cuenta con los consentimientos, se solicita que la respuesta sea encriptada (4). Finalmente, se invoca el servicio Consultar del organismo (5) como lo muestra la Figura 23.

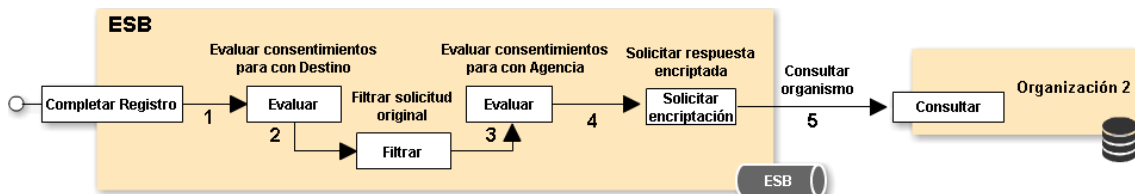


Figura 23 - Servicio Completar Registro

Servicio Actualizar Repositorio

El servicio es utilizado por aplicaciones de organismos y permite actualizar datos maestros en el sistema de MDM. Para ello, verifica que el organismo tenga consentimientos para actualizar los datos invocando al servicio Evaluar del Evaluador (1) y filtrando aquellos atributos no consentidos de la interacción (2). Luego, se verifica que la agencia cuente con consentimientos para visualizar los datos de la interacción (3). En caso de que la agencia no cuente con los consentimientos necesarios, se verifica que estos atributos se encuentren encriptados. En caso de no estarlo se rechaza la actualización. Por último, se invoca el servicio Actualizar del sistema MDM (4) como lo muestra la Figura 24.

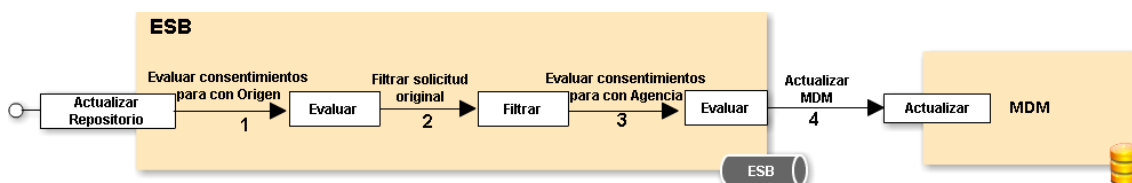


Figura 24 - Servicio Actualizar Repositorio

Servicio Sincronizar Fuentes

El servicio es utilizado por el sistema de MDM y mediante una interacción MDM permite enviar datos maestros a organismos. Para ello, verifica que el organismo que consulta tenga consentimientos para almacenar los datos (1), filtra la interacción (2) y luego la envía al servicio Sincronizar del organismo (3) como lo muestra la Figura 25.

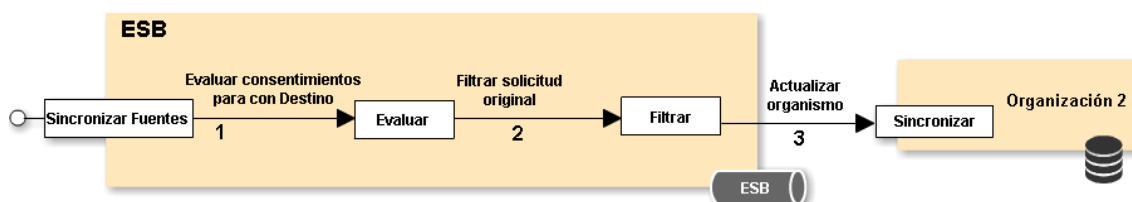


Figura 25 - Servicio Sincronizar Fuentes

4.2.5. Sistema de Consulta de Datos Personales

Este componente permite a los ciudadanos, a través de una interfaz gráfica, consultar sus datos personales invocando el servicio Consultar del ESB.

4.2.6. Organismos

En el marco del proyecto, los organismos que interesan son aquellos que tienen su base de datos registrada dentro del sistema de MDM. Cualquier organización, tanto pública como privada, que se encuentre previamente identificada, puede realizar consultas de datos maestros a la plataforma.

Por otro lado, aquellos sistemas de información que almacenen datos de los ciudadanos pueden brindar servicios, dejándolos disponibles al sistema de MDM tanto para la consulta como la sincronización de registros. En ambos casos se deben registrar los servicios en el sistema de MDM.

Los organismos que brinden servicios de consulta, son registrados en la plataforma como *orígenes de datos personales*. El servicio brinda al sistema de MDM información adicional para complementar el *golden record*, permitiendo de esta forma la implementación de atributos de tipo Registro⁸.

Además, los organismos que brinden servicios de sincronización, son registrados en la plataforma como *sincronizables*. El servicio permite al sistema de MDM insertar o actualizar atributos que se definan como *sincronizables* en dichos organismos.

Cabe destacar que la Agencia Supervisora es la encargada de establecer la interfaz que deben implementar los organismos para brindar los servicios de sincronización y consulta de datos personales.

⁸ Los atributos implementados siguiendo el estilo de solución MDM Registro, son denominados atributos tipo Registro.

4.3. Arquitectura de la solución

En la Figura 26 se muestra un diagrama con la arquitectura de la solución incluyendo la combinación de los componentes antes presentados.

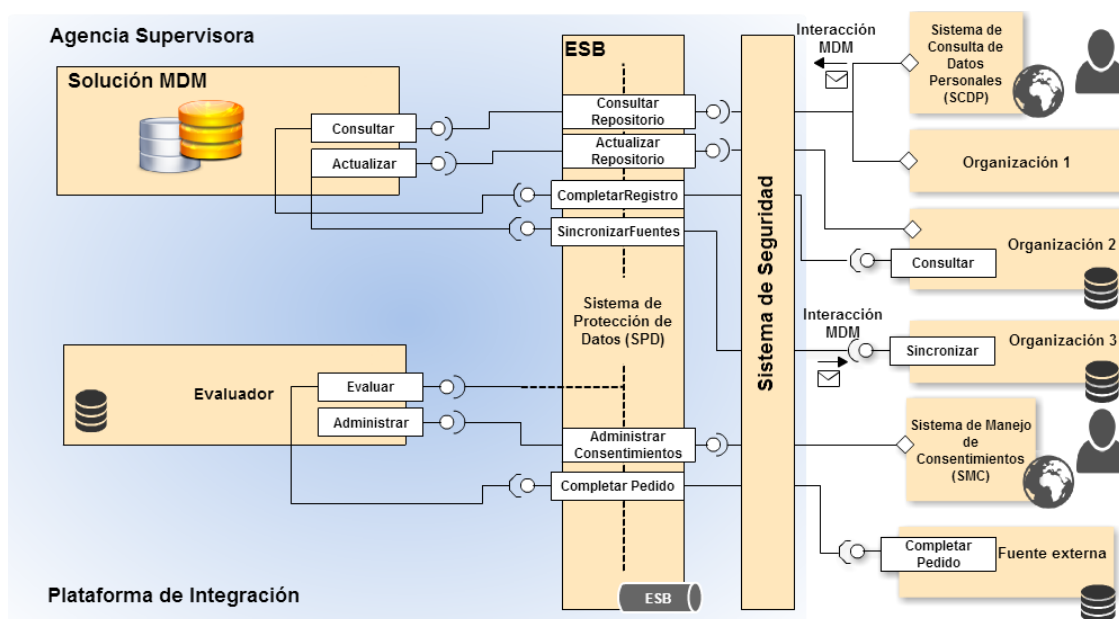


Figura 26 - Diagrama de arquitectura de la solución

Cabe destacar que tanto los ciudadanos como los organismos inician solicitudes de uso de datos mediante la interfaz Consultar, a través de interacciones MDM.

4.4. Interacciones MDM

El acceso a los datos maestros que se encuentran en el sistema de MDM se realiza mediante la exposición de servicios. Para este proyecto se decide utilizar Servicios Web que implementen el protocolo SOAP. Algunos de los motivos son:

- Es un protocolo estándar.
- Permite el cifrado de porciones del mensaje.
- Cumple con propiedades de seguridad pertinentes en la comunicación entre organizaciones que involucra datos sensibles, como ser autenticación, integridad y no repudio.
- Es el protocolo utilizado para implementar servicios en las plataformas de gobierno electrónico en general.

Por otro lado, a partir de los tipos de interacción MDM definidos y los datos necesarios para tomar una decisión de autorización, se diseña la estructura de interacción MDM, como se puede ver en Figura 27.

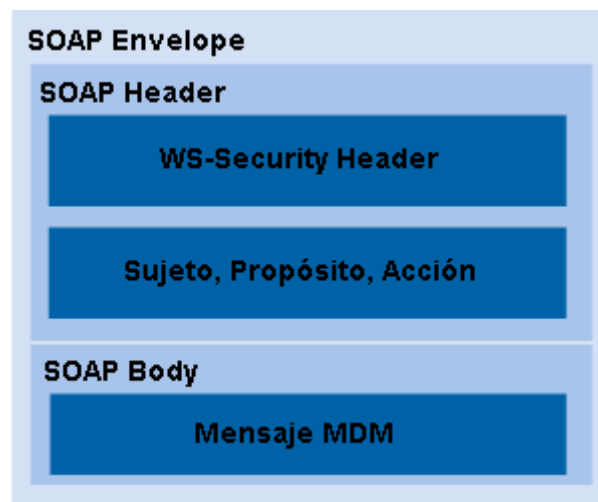


Figura 27 - Interacción MDM

En el cabezal del mensaje SOAP, se especifican los aspectos de seguridad pertinentes para brindar autenticación, integridad, confidencialidad y no repudio según el estándar *WS-Security* (detallado en 2.2.3 *WS-Security*). Además se incorporan el Sujeto (Organismo), el Propósito y la Acción necesarios para realizar la validación de políticas. Por último, el cuerpo del mensaje se crea a partir del mensaje MDM original.

En la Figura 28 se muestra un ejemplo de interacción MDM.

```

<soapenv:Envelope>
  <soapenv:Header>
    <SubjectId>Organismo</SubjectId>
    <PurposeId>Propósito</PurposeId>
    <ActionId>Acción</ActionId>
  </soapenv:Header>
  <soapenv:Body>
    <item>
      <Persona>
        <PersonaId>NroDocumento</PersonaId>
        <Atributo1></Atributo1>
        <Atributo2></Atributo2>
        ...
        <AtributoN></AtributoN>
      </Persona>
    </item>
  </soapenv:Body>
</soapenv:Envelope>
  
```

Figura 28 - Formato de interacción MDM

4.5. Diseño y evaluación de políticas XACML

En esta sección se describe la incorporación de los consentimientos de los ciudadanos como políticas XACML, los atributos y categorías de XACML utilizados, la estructura general de políticas diseñada, así como también los distintos casos de solicitud de datos personales.

4.5.1. Consentimiento como política

Se identifica que todo ciudadano brinda un conjunto de consentimientos para cada Acción y Propósito asociados a una organización y el conjunto de datos personales que habilita, como lo muestra la Figura 29.

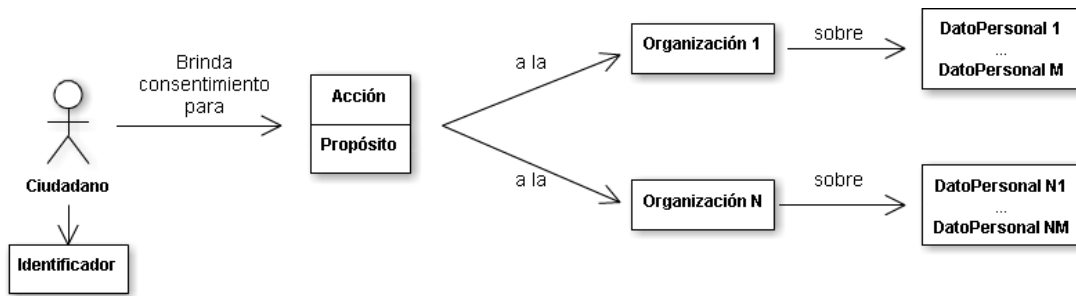


Figura 29 - Abstracción de consentimientos

Por lo tanto, se puede encontrar una correspondencia entre un consentimiento y una política XACML. A continuación se realiza la agrupación de elementos XACML de manera general, como se muestra en la Figura 30.

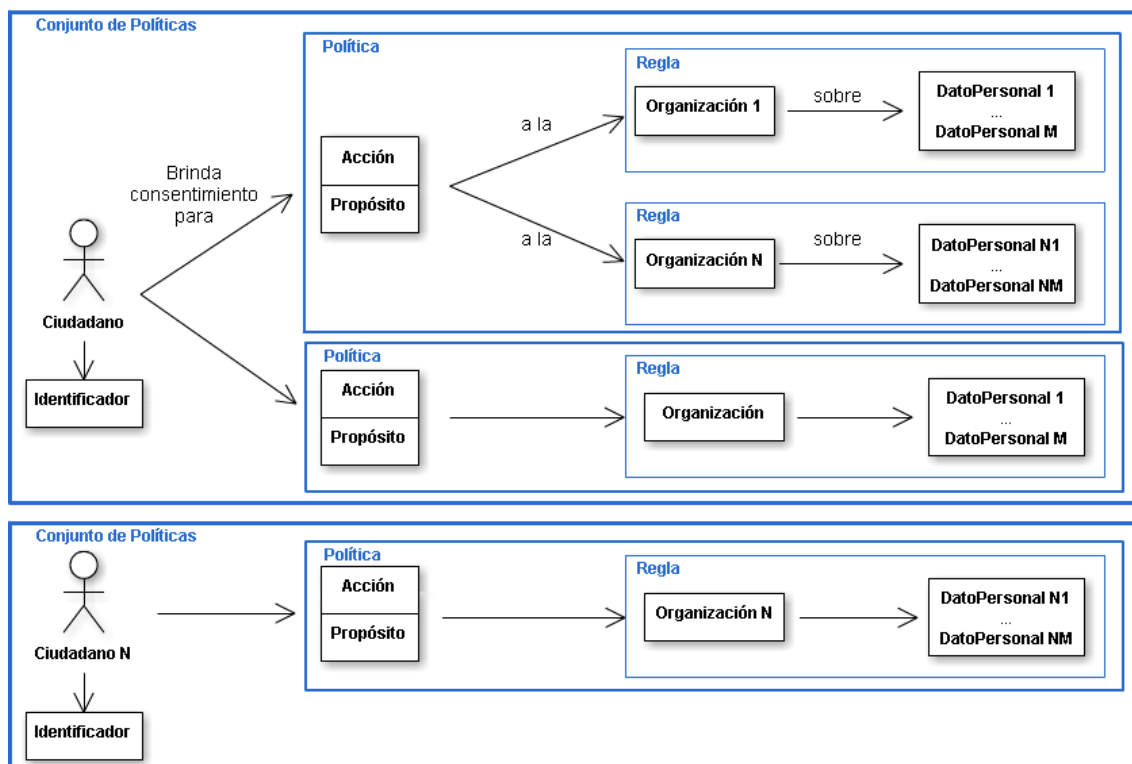


Figura 30 - Consentimientos como elementos XACML

Por ejemplo, se considera que el ciudadano cuya CI es 12345678 brinda un consentimiento para la Acción *Select_for_use* y el Propósito *MDM_BusinessOriented* a la Organización *DGI* sobre los atributos *Ascendencia* y *EstadoCivil*, como se puede ver en la Figura 31.

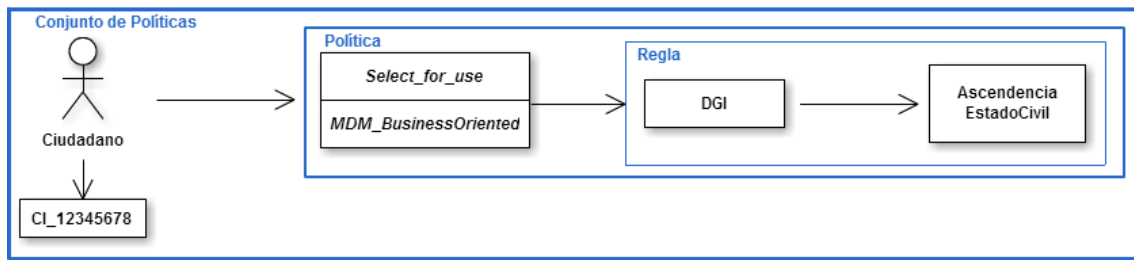


Figura 31 - Ejemplo de consentimientos como elementos XACML

Acorde a la agrupación realizada, el consentimiento de este ciudadano se corresponde entonces con una estructura XACML como se describe en la Figura 32.

```

<PolicySet PolicySetId="PS1" PolicyCombiningAlgId="deny-unless-permit">
  <Target>
    <AnyOf>
      <Allof>
        <Match>
          <AttributeValue>12345678</AttributeValue>
          <AttributeDesignator AttributeId="resource:NumeroDocumento"
            Category="attribute-category:resource"></AttributeDesignator>
        </Match>
        <Match>
          <AttributeValue>CI</AttributeValue>
          <AttributeDesignator AttributeId="resource:TipoDocumento"
            Category="attribute-category:resource"></AttributeDesignator>
        </Match>
      </Allof>
    </AnyOf>
  </Target>
  <Policy PolicyId="P1" RuleCombiningAlgId="first-applicable">
    <Target>
      <AnyOf>
        <Allof>
          <Match>
            <AttributeValue>Select_for_use</AttributeValue>
            <AttributeDesignator AttributeId="action:action-id"
              Category="attribute-category:action"></AttributeDesignator>
          </Match>
          <Match>
            <AttributeValue>MDM_BusinessOriented</AttributeValue>
            <AttributeDesignator AttributeId="purpose:purpose-id"
              Category="attribute-category:purpose"></AttributeDesignator>
          </Match>
        </Allof>
      </AnyOf>
    </Target>
    <Rule Effect="Permit" RuleId="Rule0">
      <Target>
        <AnyOf>
          <Allof>
            <Match>
              <AttributeValue>DGI</AttributeValue>
              <AttributeDesignator AttributeId="subject:subject-id"
                Category="subject-category:access-subject"></AttributeDesignator>
            </Match>
          </Allof>
        </AnyOf>
      </Target>
      <Condition>
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:or">
          <Apply>
            <AttributeValue>Ascendencia</AttributeValue>
            <AttributeDesignator AttributeId="resource:Target"
              Category="attribute-category:resource"></AttributeDesignator>
          </Apply>
          <Apply>
            <AttributeValue>EstadoCivil</AttributeValue>
            <AttributeDesignator AttributeId="resource:Target"
              Category="attribute-category:resource"></AttributeDesignator>
          </Apply>
        </Apply>
      </Condition>
    </Rule>
  </Policy>
</PolicySet>

```

Identificador del ciudadano

Acción

Propósito

Organización

Datos Personales

Regla

Política

Conjunto de políticas

Figura 32 - Ejemplo de política a partir de consentimiento

En resumen, la estructura de políticas que representan a los consentimientos se diseña en tres niveles. El primero de ellos es un Policy Set que agrupa a las políticas de cada ciudadano y cuyo Target es el identificador del mismo. Luego, en el segundo nivel, se definen políticas por cada par Acción y Propósito válidos. Éstas contienen el conjunto de reglas que tienen como Target a la organización y como Condición a aquellos atributos autorizados, retornando la decisión *Permit*.

4.5.2. Dominio de categorías y atributos

Para incorporar el Propósito a la solución XACML, se crea la categoría *Propósito* con su respectivo atributo identificador. Además, se crean nuevos atributos en la categoría Recurso: *Target*, *NumeroDocumento* y *TipoDocumento*. Éstos permiten identificar al recurso que se desea acceder.

De acuerdo a las posibles interacciones MDM se define el conjunto de valores de atributos válidos:

- **Acción**
 - **Action-id:** *Insert, Update, Delete, Select_for_use, Select_for_share*
- **Propósito**
 - **Purpose-id:** *MDM_Maintain, MDM_Synchronize, MDM_BusinessOriented.*
- **Recurso**
 - **Target:** Dato personal al que se pretende acceder.
 - **NumeroDocumento:** Número de documento que se utiliza para identificar al ciudadano.
 - **TipoDocumento:** Tipo de documento que se utiliza para identificar al ciudadano.
- **Sujeto**
 - **Subject-id:** Identificador de la organización.

4.5.3. Estructura general de políticas

Las solicitudes de datos personales se realizan a partir del identificador del ciudadano pudiendo solicitar un conjunto de atributos, para lo cual se hacen las siguientes distinciones:

- Al realizar una consulta:
 - Todos los atributos públicos son permitidos en las interacciones sin importar el identificador del ciudadano o el organismo solicitante.
 - Todos los atributos sensibles son permitidos en las interacciones en los siguientes casos:
 - a) Existe un consentimiento para el Propósito, Acción y Sujeto (organismo), brindado por el ciudadano identificado.
 - b) Existe una política de excepción que lo habilita. Por ejemplo, una investigación policial como lo indica la ley.
- Al realizar una actualización, tanto los atributos públicos como sensibles, son permitidos en las interacciones de forma análoga a la consulta sobre atributos sensibles.

Por lo tanto, la estructura de políticas se puede representar como una estructura de árbol como lo muestra la Figura 33.

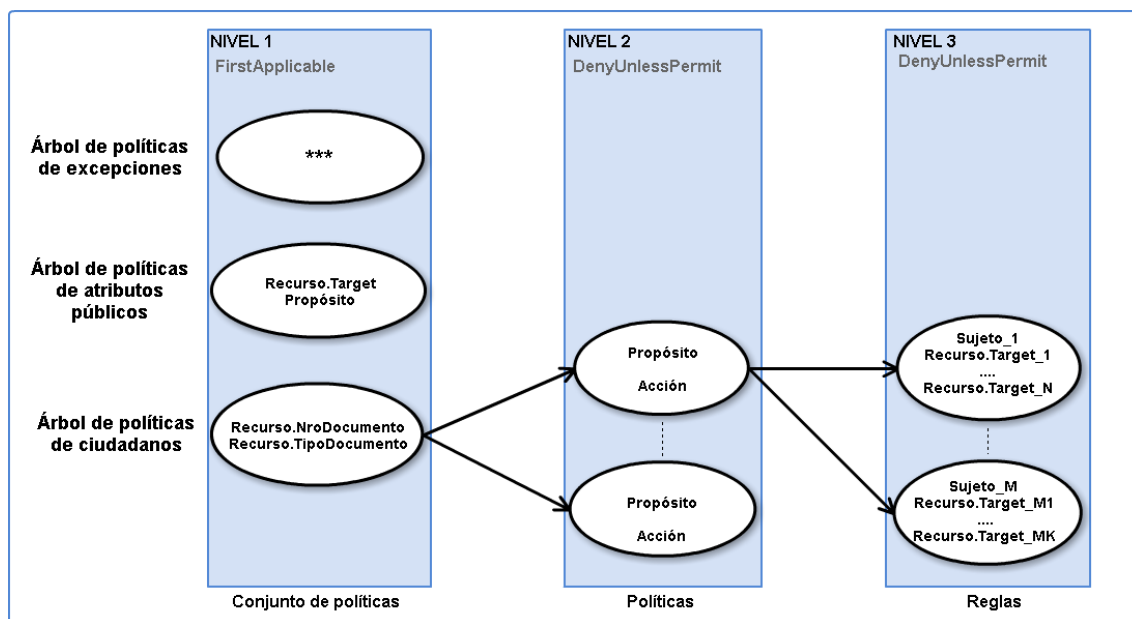


Figura 33 - Estructura de políticas

Se diseñan entonces tres tipos de árbol: Árbol de políticas de ciudadanos, Árbol de políticas de atributos públicos y Árbol de políticas de Excepciones.

Dependiendo de la solicitud XACML realizada, se evalúa un tipo de árbol diferente:

- Si la solicitud es una consulta sobre un atributo público, se evalúa el Árbol de políticas de atributos públicos y de acuerdo a lo establecido por la ley (analizado en 2.7 Ley de protección de datos personales en Uruguay) se admite toda solicitud para cualquier organización, siempre que el Propósito sea *MDM_BusinessOriented*.
- Si la solicitud es una consulta o actualización sobre un atributo sensible o una actualización sobre un atributo público, se evalúa el Árbol de políticas de ciudadanos. En dicho caso, se considera el conjunto de políticas del segundo nivel según el Propósito y Acción de la solicitud. Luego de encontrada la política correspondiente, se chequea su conjunto de reglas, conformadas por los Sujetos (organizaciones) y Recurso.Target (atributos). Esto permite determinar si la solicitud es permitida o denegada.

Además, el Árbol de políticas de Excepciones, es utilizado para todos los casos. Si el resultado de evaluar una solicitud sobre este árbol tiene un efecto de permiso o denegación, la decisión tiene prioridad frente al resto de los árboles. Cuando se esté frente a casos excepcionales donde se requiera información adicional que avale la excepción, se designará al Policy Information Point (PIP) como responsable de su obtención. Cabe destacar que el PIP debe contar con una configuración que indique las fuentes de datos con los registros de excepciones.

Por ejemplo, si el Poder Judicial (PJ) quisiese obtener los antecedentes de un ciudadano y existiera una investigación policial que lo habilite, no deberá ser necesario el previo consentimiento para acceder a estos datos. Por lo tanto, al momento de validar la interacción entre el PJ y el sistema de MDM, el módulo de evaluación de políticas identificará que se trata de una excepción incurriendo en el Árbol de Excepciones. Luego, solicitará al PIP que verifique su existencia para finalmente autorizar la solicitud.

Finalmente, para representar el orden de prioridad de los árboles, en el nivel uno se utiliza el algoritmo de combinación *First-Applicable*. Luego, para los otros niveles, se aplica *Deny-Unless-Permit* exigiendo una autorización explícita para autorizar una solicitud (ambos algoritmos son detallados en Combinación de algoritmos).

4.5.4. Casos de solicitudes de datos personales

Los consentimientos de los ciudadanos son personales, por lo tanto si una solicitud se realiza a partir de un atributo que no identifica al ciudadano resulta complejo determinar qué atributos son admitidos para cada ciudadano en particular.

Por lo tanto, se realiza la siguiente clasificación de solicitudes:

- Solicitudes a partir del identificador/es del ciudadano. Por ejemplo, solicitar datos personales del ciudadano cuya CI sea 12345678.
- Solicitudes que no incluyen identificador/es del ciudadano y consultan a partir de atributos públicos (condición de la solicitud). Por ejemplo, solicitar datos personales de los ciudadanos cuyo Nombre sea Juan.
- Solicitudes que no incluyen identificador/es del ciudadano y consultan a partir de atributos sensibles (condición de la solicitud). Por ejemplo, solicitar datos personales de los ciudadanos cuyo EstadoCivil sea Soltero/a.

En la Figura 34 se muestra el proceso de evaluación de cada solicitud dependiendo de su clasificación.

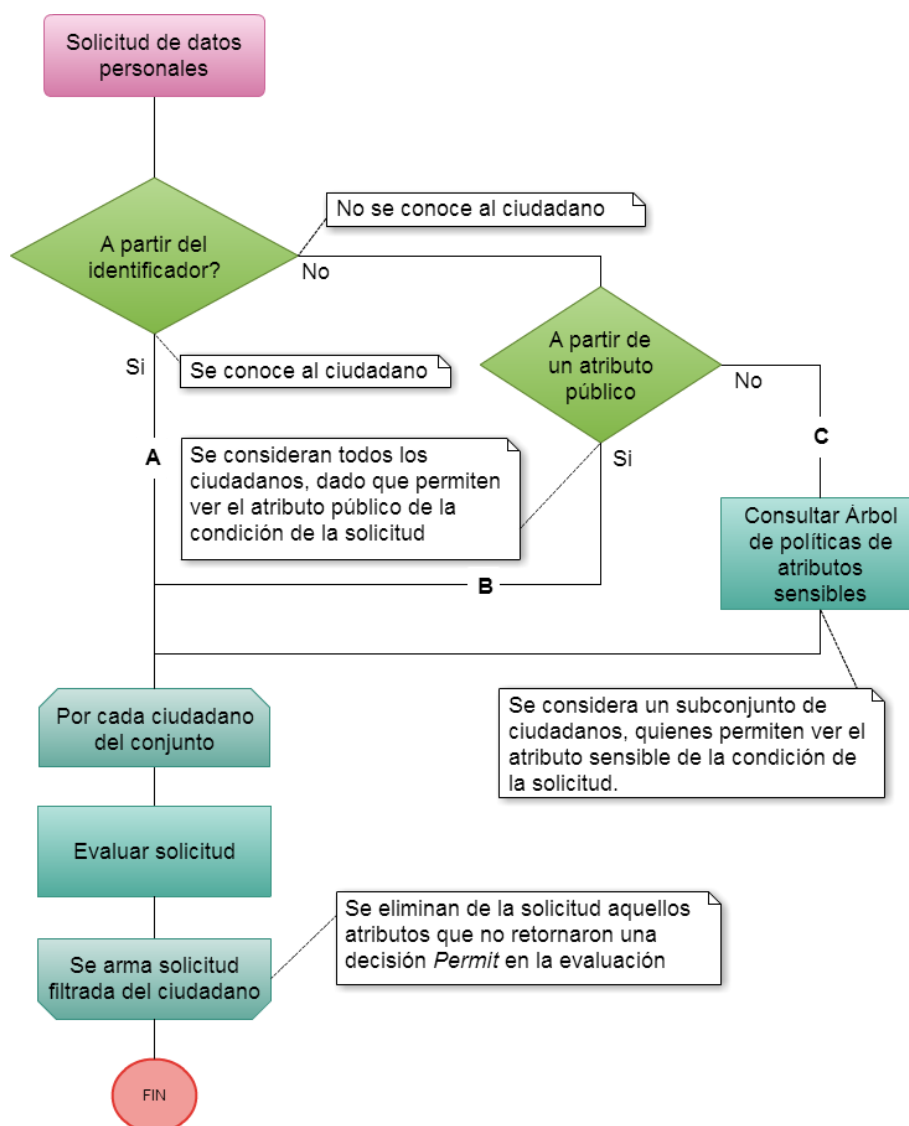


Figura 34 - Proceso de evaluación de solicitud

A partir de la Figura 34 se identifica que para poder verificar que se cumplan las leyes de protección de datos en las solicitudes, es necesario conocer previamente el conjunto de ciudadanos sobre los cuales se van a evaluar los consentimientos. Por lo tanto, se detallan a continuación los casos A, B y C identificados:

- A. Como se conoce el identificador del ciudadano se evalúa directamente la solicitud.
- B. El conjunto está formado por todos los ciudadanos. De esta forma, se deben evaluar los consentimientos de cada uno de ellos. Además, visto que cada ciudadano autoriza el acceso a un conjunto de datos distinto, es necesario crear una solicitud filtrada especialmente para cada uno de ellos. Por lo tanto, como el conjunto de ciudadanos es potencialmente grande, este tipo de solicitudes resulta muy costosa.

C. Se debe encontrar el conjunto de ciudadanos que permiten ver el atributo sensible de la condición de la solicitud. Para esto, se analizan dos posibilidades:

1. Evaluar para todo el conjunto de ciudadanos quiénes permiten su acceso. Al igual que el caso B, como el conjunto de ciudadanos es potencialmente grande, esta solución resulta muy costosa a nivel de desempeño.
2. Mantener una estructura auxiliar que permita brindar este conjunto de ciudadanos. Para lo cual se crea el Árbol de políticas de atributos sensibles.

De esta forma, se deben evaluar los consentimientos de cada ciudadano del conjunto pudiendo, en algunos casos, manejar cantidades potencialmente grandes que implican un alto costo de desempeño.

Árbol de políticas de atributos sensibles

Es necesario que la evaluación del Árbol de políticas de atributos sensibles retorne el conjunto de identificadores de ciudadanos que admiten el acceso a un `Recurso.Target` (atributo sensible) para determinados `Acción`, `Propósito` y `Sujeto` (organismo). Para lo cual, se utiliza el elemento `advice` del protocolo XACML como se muestra en la Figura 35.

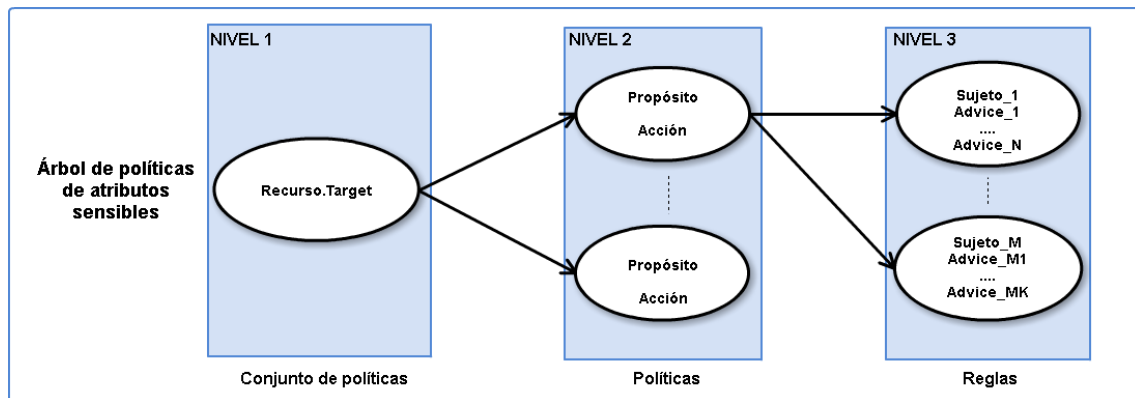


Figura 35 - Árbol de políticas de atributos sensibles

4.6. Interacción de componentes de la arquitectura

En esta sección se detallan las principales interacciones de los componentes en la arquitectura propuesta.

4.6.1. Solicitudes a partir del identificador/es del ciudadano

Al momento de recibir una consulta para obtener los datos personales de un ciudadano a partir de su identificador, se desencadena una serie de operaciones como se muestra en Figura 36.

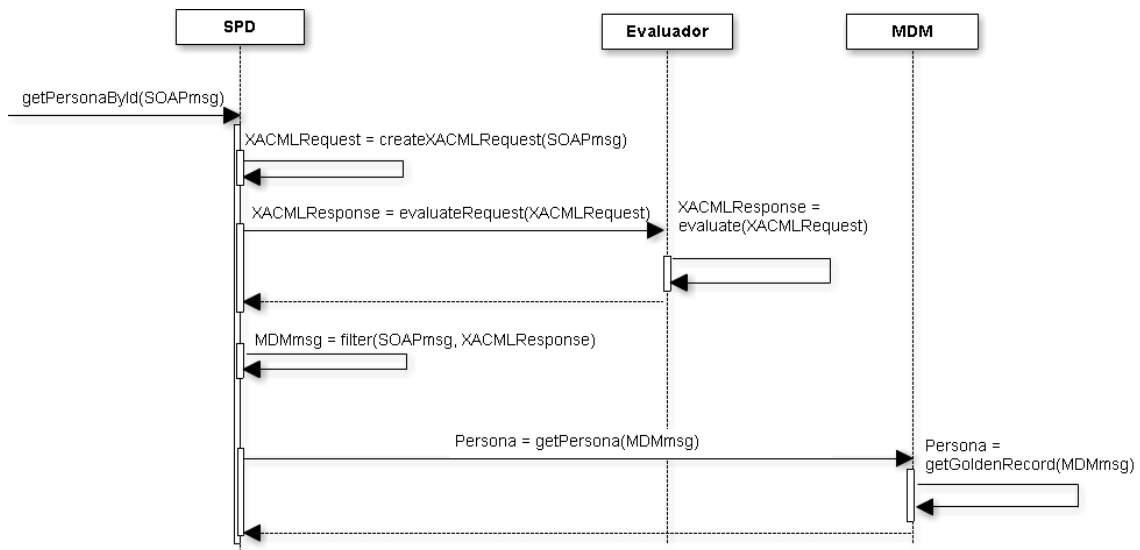


Figura 36 - DSS Solicitud de ciudadano por identificador

1. El SPD recibe una solicitud para ejecutar la operación. Ésta contiene una interacción MDM, cuyo mensaje presenta el identificador de persona y el conjunto de atributos a los cuales solicita acceder.
2. El SPD crea una solicitud XACML (*XACML Request*) a partir de la interacción recibida.
3. El SPD envía la solicitud al Evaluador para saber qué atributos son admitidos para la interacción.
4. El Evaluador evalúa la solicitud y envía al SPD la respuesta XACML (*XACML Response*) que contiene, para cada atributo solicitado, si se permite o deniega el acceso.
5. El SPD filtra el mensaje MDM original a partir del *XACML Response* creando un mensaje MDM.
6. El SPD envía el pedido al sistema de MDM solicitando la persona.
7. El sistema de MDM responde al SPD con la persona solicitada (*).
8. El SPD finaliza la operación retornando la persona solicitada.

(*) En el punto 7, en caso de ser necesario, el sistema de MDM puede requerir datos de fuentes externas para completar el *golden record*. Por lo tanto, se debe interceptar esta comunicación para validar los permisos que tengan la fuente y la plataforma sobre estos datos. En caso de que la fuente no tenga permiso para compartir, el mensaje es filtrado retornando un valor vacío; de lo contrario, se procede con la validación sobre la plataforma. Si ésta puede acceder al dato, el mensaje continúa su curso, de otro modo, se debe solicitar a la fuente externa que envíe los datos involucrados encriptados mediante una clave acordada entre el organismo solicitante y la fuente externa. Esto es requerido para asegurar que la plataforma no acceda ni manipule datos de forma no consentida.

Cabe destacar, que el proceso de evaluación de la solicitud anterior se puede extender, sin perder generalidad, a solicitudes de inserción, eliminación o modificación de registros, siempre que se cuente con el identificador del ciudadano.

4.6.2. Solicitudes que no incluyen identificador/es del ciudadano y consultan a partir de atributos sensibles

Al momento de consultar por un conjunto de ciudadanos a partir del valor de un atributo sensible, se desencadena una serie de operaciones como se puede ver en Figura 37.

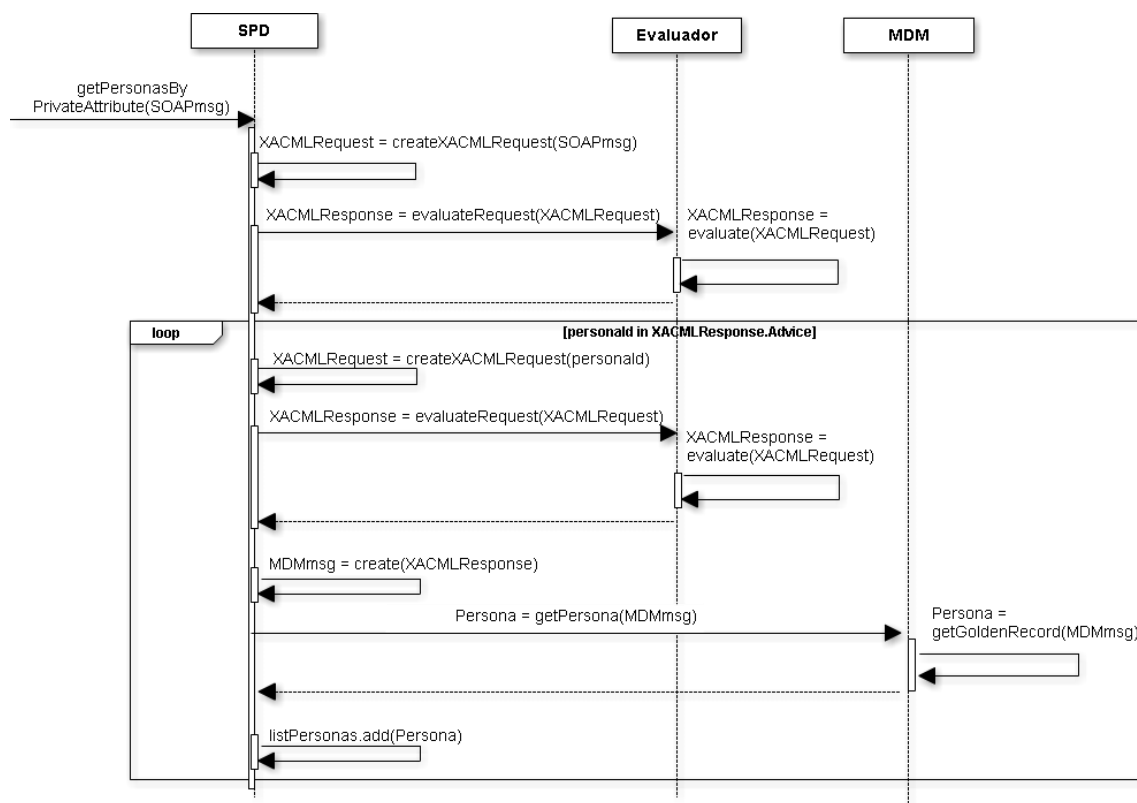


Figura 37 – DSS Solicitud de ciudadanos filtrando por atributo sensible

1. El SPD recibe una solicitud para ejecutar la operación que contiene una interacción MDM, cuyo mensaje presenta el identificador del atributo sensible y el valor por el cual se quiere filtrar la búsqueda.
2. El SPD crea una solicitud XACML (*XACML Request*) y la envía al Evaluador.
3. El Evaluador evalúa la solicitud y crea una respuesta XACML (*XACML Response*). Ésta contiene el conjunto de identificadores de ciudadano que permiten acceder al atributo por el cual se filtra, según el Sujeto (organización), Acción y Propósito dados. Los identificadores se retornan mediante *Advice expressions*.
4. El SPD recibe la respuesta y para cada identificador recibido:
 1. El SPD crea una solicitud XACML para saber qué atributos son admitidos para la interacción sobre los datos personales del ciudadano que se está considerando.
 2. El SPD envía las solicitudes al Evaluador.
 3. El Evaluador evalúa la solicitud y envía al SPD una respuesta XACML que contiene, para cada atributo solicitado, si se permite o deniega el acceso.
 4. El SPD recibe la respuesta, crea un mensaje MDM para la persona considerada con los atributos admitidos y lo envía al sistema de MDM.
 5. El sistema de MDM responde al SPD con la información del ciudadano solicitada. De igual forma que en la solicitud de ciudadano por identificador.
 6. El SPD recibe la persona y la agrega a la lista del resultado
5. El SPD retorna la lista generada.

Este enfoque se puede extender de manera sencilla filtrando por varios atributos sensibles. Para lograr esto, el SPD debe considerar únicamente aquellos identificadores que pertenezcan a la intersección de los conjuntos recibidos por cada atributo sensible del filtro.

5. Implementación

En este capítulo, se presentan los distintos productos y herramientas utilizados en la implementación del prototipo. Además, se detalla la solución construida mostrando cómo se utilizan los productos seleccionados.

Los principales desafíos de la etapa de implementación fueron el análisis y elección de los distintos productos a utilizar, así como también su configuración e incorporación a la solución.

5.1. Selección de productos

En esta sección se describe la selección de productos MDM, XACML y ESB así como las principales características de los mismos.

5.1.1. Master Data Management

En esta sección se realiza una comparación sobre los productos analizados. Para seleccionar el producto de MDM a utilizar se comenzó con una preselección en base a la posibilidad de contar con los productos, tanto por ser de libre acceso como por estar comprendidos en convenios de iniciativas académicas entre la Facultad de Ingeniería y los proveedores.

El análisis de los productos se centró en encontrar un producto que:

- Se logre instalar y configurar en el tiempo planificado para esta etapa.
- Permita configurar la mayor variedad de estilos arquitectónicos de MDM.
- Cuento con documentación clara y accesible que permita realizar los puntos anteriores.

En resumen, la finalidad en la evaluación de los productos es encontrar el que mejor se adecúe al contexto del proyecto y permita brindar la mayor variedad de estilos arquitectónicos de MDM posibles. Todo esto considerando los tiempos planificados para esta etapa del proyecto.

Las características que se utilizan para evaluar y comparar los productos se presentan en la Tabla 1. Cada característica puede tomar un valor entre uno y tres, siendo el tres el que brinda mejor posicionamiento en cada característica.

Instalar y configurar en el tiempo planificado	X
Mayor variedad de estilos arquitectónicos de MDM	X
Documentación clara y accesible	X

Tabla 1 - Ejemplo tabla comparativa producto MDM

5.1.1.1. Master Data Services (MDS)

Master Data Services es la solución que plantea Microsoft [32], a través de SQL Server⁹, para la gestión de los datos maestros. MDS permite definir los dominios de datos que se necesiten, además de brindar jerarquías de los datos, seguridad específica, transacciones, versionado de la información y la aplicación de reglas de negocio [34].

⁹ “[...]sistema de administración y análisis de bases de datos relacionales de Microsoft para soluciones de comercio electrónico, línea de negocio y almacenamiento de datos” [33].

Por otro lado, MDS incluye un conjunto de herramientas:

- **MDS Configuration Manager:** para crear y configurar las bases de datos y aplicaciones web necesarias.
- **Master Data Manager:** para realizar tareas administrativas y poder visualizar los datos.
- Una herramienta que permite empaquetar el modelo de objetos para desplegarlo en distintos entornos.
- **MDS WebService:** que permiten realizar soluciones personalizadas para los Servicios de Datos Maestros, mediante la extensión o reemplazo de los mismos.
- **Complemento Excel:** para administrar las entidades y sus atributos.

La instalación del producto se realiza como extensión de la instalación de *SQL Server* señalando que se desea incluir *Master Data Services*. Sin embargo, la configuración del producto resulta poco intuitiva.

Además, se destacan las siguientes limitaciones:

- No brinda todos los estilos arquitectónicos de Master Data Management.
- Alto acoplamiento con tecnologías de *Microsoft*, por ejemplo, a nivel de sistema operativo.
- Los foros de discusión que se encontraron mostraban poca actividad.

En base a estas consideraciones, la evaluación de este producto es la que se presenta en la Tabla 2.

Instalar y configurar en el tiempo planificado	2
Mayor variedad de estilos arquitectónicos de MDM	1
Documentación clara y accesible	1

Tabla 2 - Puntaje de características MDS

5.1.1.2. Infosphere MDM

IBM InfoSphere Master Data Management es el producto desarrollado por *IBM* para la gestión de datos maestros y brinda principalmente las siguientes características [35]:

- Alto rendimiento y escalabilidad.
- Facilita la administración y el gobierno de datos, ofreciendo información fiable.
- Se integra con otras aplicaciones y sistemas.

Este producto, en su versión 11, cuenta con cuatro ediciones: *Standard Edition*, *Advanced Edition*, *Collaborative Edition* y *Enterprise Edition*.

IBM InfoSphere Master Data Management Standard Edition ofrece vistas precisas y en tiempo real de los datos maestros a toda la organización. Esta versión brinda soporte únicamente a implementaciones del estilo virtual o también llamado registro. Este producto permite de manera sencilla implementar un MDM [36].

IBM InfoSphere Master Data Management Advanced Edition brinda un conjunto completo de funcionalidades para la gestión de datos maestros [37]. Este sistema permite generar una solución de MDM tanto del estilo virtual como físico, pudiendo utilizarse ambos en conjunto, logrando el estilo híbrido [38].

IBM InfoSphere Master Data Management Collaborative Edition es una implementación basada en *Product Information Management* (PIM). Esto permite la creación de un repositorio único y actualizado. Además, brinda la posibilidad de actualizar el repositorio a partir de sistemas externos, asegurar los procesos de negocio y la precisión de los datos, así como también la sincronización de la información hacia fuentes de terceros [38].

IBM InfoSphere Master Data Management Enterprise Edition ofrece vistas confiables y seguras de los datos maestros a través de la organización. Esta solución soporta todos los dominios y estilos arquitectónicos de MDM [39].

Para la evaluación del producto se contó con *Infosphere MDM – Collaborative Edition* (v11.4) e *Infosphere MDM – Advanced Edition* (v11.4), las cuales requieren otro conjunto de productos auxiliares para la instalación, como el *Rational*¹⁰, *Websphere Application Server*¹¹, entre otros. Es así, que a la hora de realizar la instalación se generaron un conjunto de inconvenientes:

- Elevados requerimientos de hardware.
- Manuales de instalación complejos.
- Procesos de instalación complejos y poco intuitivos.
- La edición colaborativa funciona únicamente en un conjunto reducido de sistemas operativos.
- Se requiere un nivel de comprensión avanzado de todos los productos involucrados.

Luego de sortear la gran mayoría de los inconvenientes en la instalación de la edición Avanzada, se configura una solución de MDM estilo Repositorio. Sin embargo, no se logra configurar otros estilos ni completar la instalación de la edición colaborativa.

En base a estas consideraciones, la evaluación de este producto es la que se presenta en la Tabla 3.

Instalar y configurar en el tiempo planificado	1
Mayor variedad de estilos arquitectónicos de MDM	2
Documentación clara y accesible	1

Tabla 3 - Puntaje de características InfoSphere MDM

5.1.1.3. Talend MDM

Talend es una empresa fundada en 2006, cuyos trabajadores de distintas partes del mundo desarrollan soluciones empresariales [42].

¹⁰ "Permite el desarrollo flexible de software a través de una mejor colaboración en equipo y un mayor control de riesgos y cambios" [40].

¹¹ "Ofrece opciones para un entorno de ejecución de servidor de aplicaciones Java más rápido y flexible. (...)" [41].

Talend Master Data Management es una herramienta implementada por *Talend* que permite unificar grandes cantidades de datos entre clientes, productos y proveedores. Esto se puede realizar desde varias líneas de negocio y múltiples dominios, creando un repositorio con datos maestros, conteniendo el conjunto de los registros confiables o *golden records*.

Talend cuenta con tres distribuciones con diferentes características para MDM: *Open Studio for MDM*, *Platform for MDM* y *Platform for MDM with Big Data*.

La distribución *Open Studio* es de código abierto y se puede descargar gratis [43]. El resto de las distribuciones tiene costo según la cantidad de usuarios y características de la plataforma que se desee.

Se analizará únicamente la versión *Open Studio for MDM*, dado que es la única gratuita.

“Talend Master Data Management (MDM) unifies any amount of data—from customers to products to suppliers and beyond—into a single, actionable ‘version of the truth.’” [43]. Como lo señala *Talend* en su página, éste pretende unificar grandes cantidades de información en una única procesable *“versión de la verdad”*.

Talend se ha transformado en el proveedor líder de sistemas de integración de código abierto, mediante el producto *Open Studio for MDM*. Esta solución es simple de usar y presenta las siguientes características [43]:

- Entorno de desarrollo basado en Eclipse¹², presentando de manera sencilla la construcción de modelos de datos, la definición de procesos (*Jobs*) de integración de datos y el manejo de los flujos de administración de los mismos.
- Más de 900 conectores para establecer flujos y transformaciones entre distintas bases de datos, tipos de archivos y sistemas legados.
- La posibilidad de diseñar un modelo de datos flexible.
- Herramientas de ETL¹³ robustas.
- Validaciones de los datos y funcionalidades para incrementar los mismos, para mejorar su calidad.
- Servidor Web, donde expone Servicios Web permitiendo implementar servicios orientados a datos.

Un *Job* es toda unidad básica ejecutable que compone a los flujos creados en *Talend*. Éstos pueden ser desplegados como Servicios Web SOAP en el servidor web de *Talend* o ser compilados en una librería Java, permitiendo su ejecución en otros entornos.

La instalación y configuración de *Talend MDM – Open Studio* resulta sencilla, amigable y multiplataforma. Por otro lado, si bien este producto no brinda de manera nativa la implementación de los estilos de MDM, mediante *Jobs* y flujos de administración se logra configurar los estilos Registro, Repositorio Centralizado e Híbrido.

¹² Entorno de desarrollo y plataformas para trabajar con lenguajes de programación como Java, C, C++, entre otros. [44]

¹³ Extracción, transformación y carga de datos desde un sistema origen a un sistema destino [45].

En base a estas consideraciones, la evaluación de este producto es la que se presenta en la Tabla 4.

Instalar y configurar en el tiempo planificado	3
Mayor variedad de estilos arquitectónicos de MDM	2
Documentación clara y accesible	3

Tabla 4 - Puntaje de características Talend MDM

5.1.1.4. Resumen y selección de producto MDM

A modo de resumen se presenta la siguiente tabla comparativa:

Características	MDS	InfoSphere MDM	Talend MDM
Instalar y configurar	2	1	3
Estilos arquitectónicos de MDM	1	2	2
Documentación	1	1	3

Tabla 5 - Comparación productos MDM

A partir de los productos analizados y dada su simplicidad de configuración y uso, así como también la posibilidad de obtener la más amplia cantidad de escenarios posibles, se decide utilizar *Talend MDM* como producto MDM para la implementación del prototipo.

Cabe destacar, que el objetivo fue encontrar el producto de MDM que permita implementar la mayor variedad de estilos arquitectónicos posibles, considerando los tiempos planificados para esta etapa del proyecto.

5.1.2. Administrador y Evaluador de políticas XACML

Para seleccionar un Administrador y Evaluador de políticas XACML se analiza una librería de código abierto y un producto que hace uso de la misma, provistos por *WSO2*, para implementar el protocolo XACML.

WSO2 es una compañía estadounidense que desarrolla aplicaciones de software libre cuyo objetivo es proveer una arquitectura basada en servicios (SOA) para desarrolladores profesionales, empresas multinacionales o usuarios [46]. Además, la empresa se especializa en tecnologías de *middleware*.

5.1.2.1. Balana XACML

Balana es una librería de código abierto, desarrollada en lenguaje Java, que implementa los principales conceptos del protocolo XACML 3.0 [47].

El proyecto *Balana* cuenta con los siguientes componentes:

- **Balana Core:** contiene el código fuente.
- **Balana Samples:** conjunto de ejemplos de ejecución.
- **Balana Utils:** métodos útiles del *Core*, por ejemplo, funciones disponibles para el proceso de edición de políticas.
- **Balana Documentation:** documentación sobre la implementación.
- **Balana Distribution:** programa compilado.

El proyecto se compila generando una librería Java, que se puede integrar con otros sistemas implementados en el mismo lenguaje.

5.1.2.2. WSO2 Identity Server

WSO2 Identity Server es un software desarrollado por WSO2 que permite el manejo seguro de identidades en aplicaciones web empresariales y servicios. A su vez, se encarga de gestionar los derechos de los usuarios de un sistema de forma ágil y eficaz. Para esto, utiliza como base distintos protocolos como XACML [48].

Las principales capacidades del software en cuanto al soporte del protocolo XACML incluyen [49]:

- Interfaz web para la gestión de políticas, permitiendo crear atributos, reglas, políticas y conjuntos de políticas.
- Mecanismos de evaluación de prueba de políticas.
- Soporte de múltiples módulos PIP.
- Mecanismos de Caché para, por ejemplo, obtener políticas que se utilizan de manera frecuente.
- Notificaciones sobre cambios en políticas.

Para realizar la gestión de las políticas XACML se debe configurar previamente el módulo PAP y el PDP indicando el conjunto de atributos válidos a utilizar en las políticas XACML. Por otro lado, el ciclo de vida de las políticas se desarrolla a través de las siguientes etapas:

1. **Creación:** En esta etapa se crean las políticas a partir de un editor de texto. Estas deben seguir el formato establecido por el protocolo XACML y la configuración dada [50].
2. **Prueba:** Una vez que la política fue creada, se pueden realizar solicitudes para evaluarla en un ambiente de prueba, validando así su funcionamiento.
3. **Edición:** Dentro del PAP se pueden editar las políticas, generando un historial de versiones.
4. **Publicación:** En esta etapa las políticas son publicadas al PDP. Dentro de la administración del PDP se puede ver el conjunto de políticas publicadas y se puede ajustar el orden de su evaluación [51].
5. **Habilitación:** Esta etapa permite la habilitación de las políticas publicadas, pasando a ser consideradas en la evaluación de solicitudes [52].
6. **Prueba en producción:** El conjunto de políticas comprendidas en el PDP pueden ser evaluadas utilizando la herramienta de prueba, simulando ejecutar la solicitud en el ambiente de producción.
7. **Eliminación:** Es posible eliminar las políticas comprendidas dentro del PDP o el PAP.

Por otro lado, WSO2 IS permite la incorporación de módulos PIP en el servidor de identidades. Estos deben implementar una interfaz definida por la herramienta para poder integrarse de manera correcta.

Otros aspectos a destacar de *WSO2 Identity Server* que hacen a la usabilidad del producto y capacidad de integración son:

- Interfaz de usuario amigable.
- Conjunto de Servicios Web para la integración con sistemas externos, brindando las funcionalidades principales del sistema.
- Documentación en línea actualizada, con ejemplos de configuración y uso, tanto para usuarios administradores como usuarios finales.

5.1.2.3. Resumen y selección de producto XACML

Luego de enumerar las características de ambos productos, se decidió utilizar *WSO2 IS* debido a que extiende las funcionalidades disponibles en *WSO2 Balana*, adicionando una interfaz gráfica para el manejo de políticas, servicios para integración con sistemas externos, integración con componente PIP, entre otras.

5.1.3. Enterprise Service Bus

En esta sección se detalla el producto de *MuleSoft* utilizado como *middleware* de tipo Enterprise Service Bus. La decisión de incorporar este producto se fundamenta en que es de código abierto, permite definir flujos de mediación de manera sencilla y es compatible con múltiples sistemas operativos. Además, el equipo del proyecto contaba con experiencia sobre el producto.

MuleSoft es una empresa de software, especializada en conectividad e integración de aplicaciones, que evolucionó a partir de la creación de un proyecto de software libre, *Mule Project*, en el año 2003 [53].

Mule ESB es un *middleware* de tipo Enterprise Service Bus liviano, implementado en Java, que facilita la interconexión de distintas aplicaciones [54].

Las principales funcionalidades que ofrece la plataforma son de creación y despliegue de servicios, servicios de mediación, ruteo de mensajes y transformación de datos.

El sistema está basado en el IDE Eclipse y adiciona un complemento que permite crear flujos de mediación mediante una interfaz gráfica.

Un flujo es la combinación de elementos que logran gestionar la recepción, procesamiento y ruteo de mensajes. A su vez, varios flujos pueden ser conectados entre sí permitiendo el despliegue de soluciones más complejas [55].

Los elementos que pueden estar comprendidos en el flujo del ESB son [56]:

- **Enrutador:** direcciona mensajes dentro de un flujo.
- **Filtro:** evalúa si un mensaje es válido siguiendo un criterio definido.
- **Transformador:** altera el contenido o propiedades de un mensaje.
- **Conector:** permite al flujo enviar o recibir mensajes de un sistema externo.
- **Componente:** procesador de mensajes que ejecuta lógica de negocio.
- **Estrategia de excepción:** flujo que se realiza para manejar excepciones.

5.2. Detalles de implementación

En esta sección se describe la implementación de los distintos componentes de la solución propuesta.

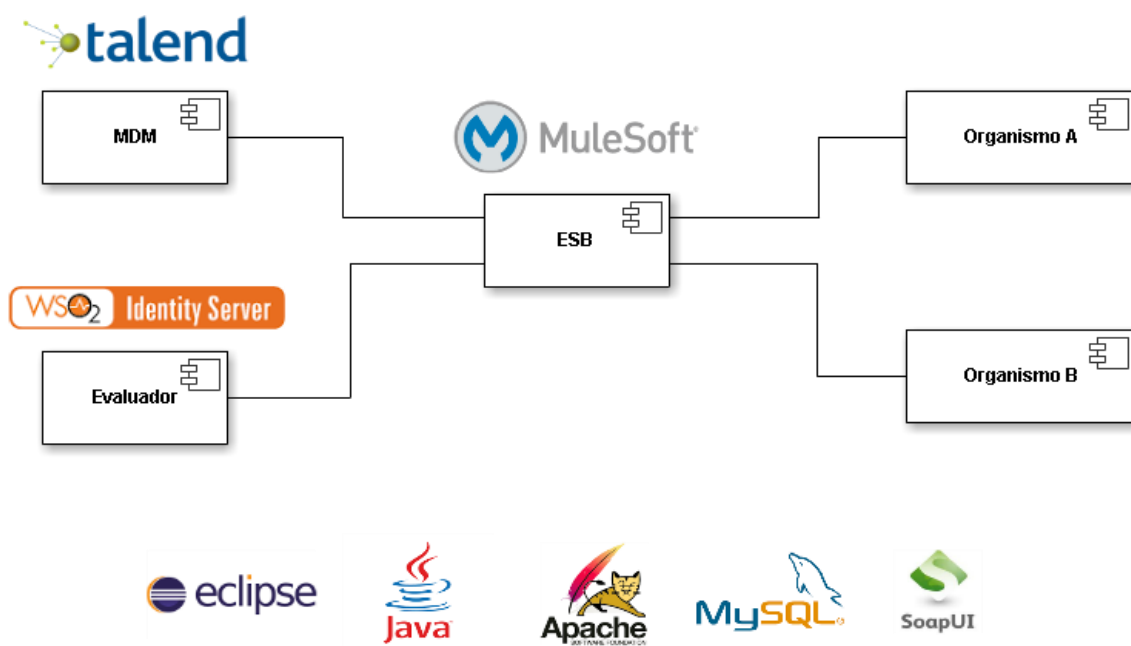


Figura 38 – Herramientas utilizadas

A partir de la Figura 38 se pueden destacar los siguientes puntos:

- Para implementar la solución de MDM se utiliza el producto de *Talend MDM - Open Studio for Master Data Management*.
- El componente Evaluador se implementa mediante la herramienta de *WSO2 Identity Server*.
- El desarrollo del módulo ESB se realiza a través de *MuleSoft Enterprise Service Bus*.
- Para el desarrollo de los Servicios Web de los organismos se utiliza Java como lenguaje de desarrollo, *Eclipse* como entorno de desarrollo y *Apache Tomcat*¹⁴ como servidor web.
- Para simular las solicitudes de los organismos se utiliza *SOAP UI*¹⁵.
- Como sistema de gestión de bases de datos en los distintos componentes se utiliza *MySQL*¹⁶.

En las siguientes secciones se detallan los módulos principales de la solución.

¹⁴ "Implementación de código abierto de tecnologías Java Servlet, JavaServer Pages, Java Expression Language y Java WebSocket" [57].

¹⁵ "SoapUI es una solución para testing funcional de código abierto y multiplataforma." [58].

¹⁶ Sistema de gestión de bases de datos relacionales [59].

5.2.1. Componente MDM

Luego de la instalación y configuración de *Talend MDM*, se debe crear el modelo de datos a utilizar. Este producto brinda una interfaz de usuario, permitiendo lograrlo de manera sencilla.

Por otro lado, se desarrollan tres *Jobs* (procesos):

- **Consulta:** recibe una interacción MDM cuyos parámetros determinan la entidad que debe retornar. Además, debe considerar aquellos atributos de tipo Registro, resolver qué fuente de información contiene el dato y luego solicitarlo. Finalmente, retorna la información requerida.
- **Actualización:** recibe una interacción MDM, determinando la entidad a insertar o actualizar con los valores de los atributos correspondientes.
- **Sincronización:** se ejecuta luego de una actualización y será encargado de actualizar las bases de datos de organizaciones que hayan sido configuradas como fuente de datos maestros, invocando un Servicio Web expuesto por el organismo, enviando los atributos pertinentes.

De esta forma, combinando los distintos *Jobs* implementados, se logran los siguientes estilos arquitectónicos de MDM:

- **Registro:** al configurar la fuente original de cada dato y mediante el *job* de consulta.
- **Repositorio – Centralizado:** el tipo repositorio es brindado de manera natural por el producto. Adicionando los *Jobs* de actualización y sincronización, se logra la exposición de servicios de edición y la sincronización con fuentes externas de datos maestros.
- **Híbrido:** resultado de la aplicación de los estilos anteriores.

En la Figura 39 se puede ver un ejemplo genérico de invocación original a los servicios MDM cuya interfaz fue implementada para acceder a los atributos que se solicitan dentro del cuerpo del mensaje.

```
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:tal="http://talend.org">
  <soapenv:Header/>
  <soapenv:Body>
    <tal:args>
      <item>
        <![CDATA[
          <Entidad>
            <AtributoA></AtributoA>
            <AtributoB></AtributoB>
            <AtributoC></AtributoC>
          </Entidad>
        ]]>
      </item>
    </tal:args>

    </soapenv:Body>
  </soapenv:Envelope>
```

Dato maestro

Interacción MDM

Figura 39 - Ejemplo interacción MDM original

En la Figura 40 se puede ver un ejemplo genérico de la invocación a los servicios del MDM. A la interacción original se agregan los *Metadatos XACML* que incluyen el Propósito para el cual el sujeto realiza la Acción sobre los datos maestros indicados.

```
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:tal="http://talend.org">
  <soapenv:Header >
    <SubjectId></SubjectId>
    <PurposeId></PurposeId>
    <ActionId></ActionId>
  </soapenv:Header>
  <soapenv:Body>
    <tal:args>
      <item>
        <![CDATA[
          <Entidad>
            <AtributoA></AtributoA>
            <AtributoB></AtributoB>
            <AtributoC></AtributoC>
          </Entidad>
        ]]>
      </item>
    </tal:args>
  </soapenv:Body>
</soapenv:Envelope>
```

Metadatos XACML

Interacción MDM

Figura 40 - Ejemplo interacción MDM

5.2.2. Componente Evaluador

Con el objetivo de incorporar a la solución el protocolo XACML se hace uso del producto *WSO2 Identity Server (WSO2 IS)*. A continuación, se describen las configuraciones necesarias a realizar en cada módulo, como se muestra en la Figura 41.

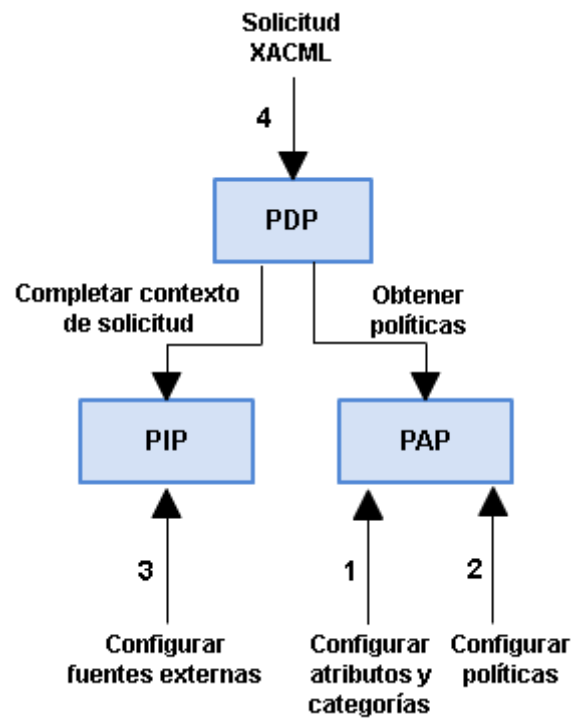


Figura 41 - Comunicaciones componente Evaluador

Configuración general

En el módulo PAP se configuran las categorías y atributos que extienden a los sugeridos por el protocolo XACML (1), como se puede ver en la Figura 42. Esta configuración es utilizada por la herramienta para validar sintácticamente las políticas.

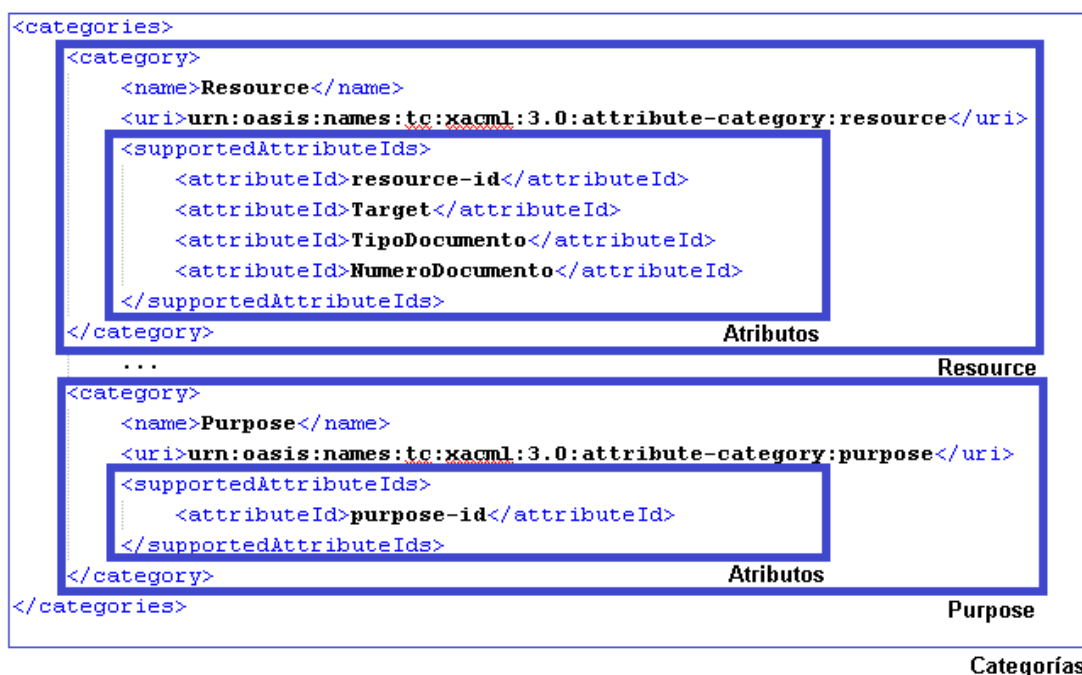


Figura 42 - Configuración de nuevos atributos XACML

Por otro lado, para evaluar solicitudes XACML (4) se utiliza el servicio *EntitlementService* brindado por *WSO2 IS* como una interfaz para acceder al módulo PDP. De éste se utiliza la operación *getDecision* que recibe como parámetro una solicitud XACML y retorna una Respuesta XACML siguiendo el estándar y las políticas habilitadas en el PDP [60]. Resulta necesario entonces habilitar el servicio *EntitlementService* en la configuración de *WSO2 IS*. El servicio utiliza el protocolo SOAP e implementa mecanismos de seguridad bajo el protocolo SSL. Por lo tanto, para establecer la comunicación se debe contar con las credenciales y el certificado correspondiente [61].

Finalmente, para configurar el módulo PIP se implementa el método *getAttributeValues*. En éste se implementa la lógica necesaria para completar el contexto de las solicitudes, incluyendo el acceso a las fuentes de datos externas requeridas (3).

Cabe destacar que el módulo PIP no manipula datos personales, por lo tanto, no es necesario realizar la validación de las leyes de protección de datos cuando accede a información de fuentes externas.

Gestión de políticas

Para configurar las políticas en el PAP (2), se presenta un ejemplo genérico de conjunto de políticas (*Policy Set*) que se corresponde con el primer nivel del árbol de políticas de un ciudadano, cuyo tipo de documento es CI y número 1.234.567-8. El conjunto de políticas se evalúa según el algoritmo *Deny-unless-permit* e incluye las políticas asociadas al ciudadano, como lo muestra la Figura 43.

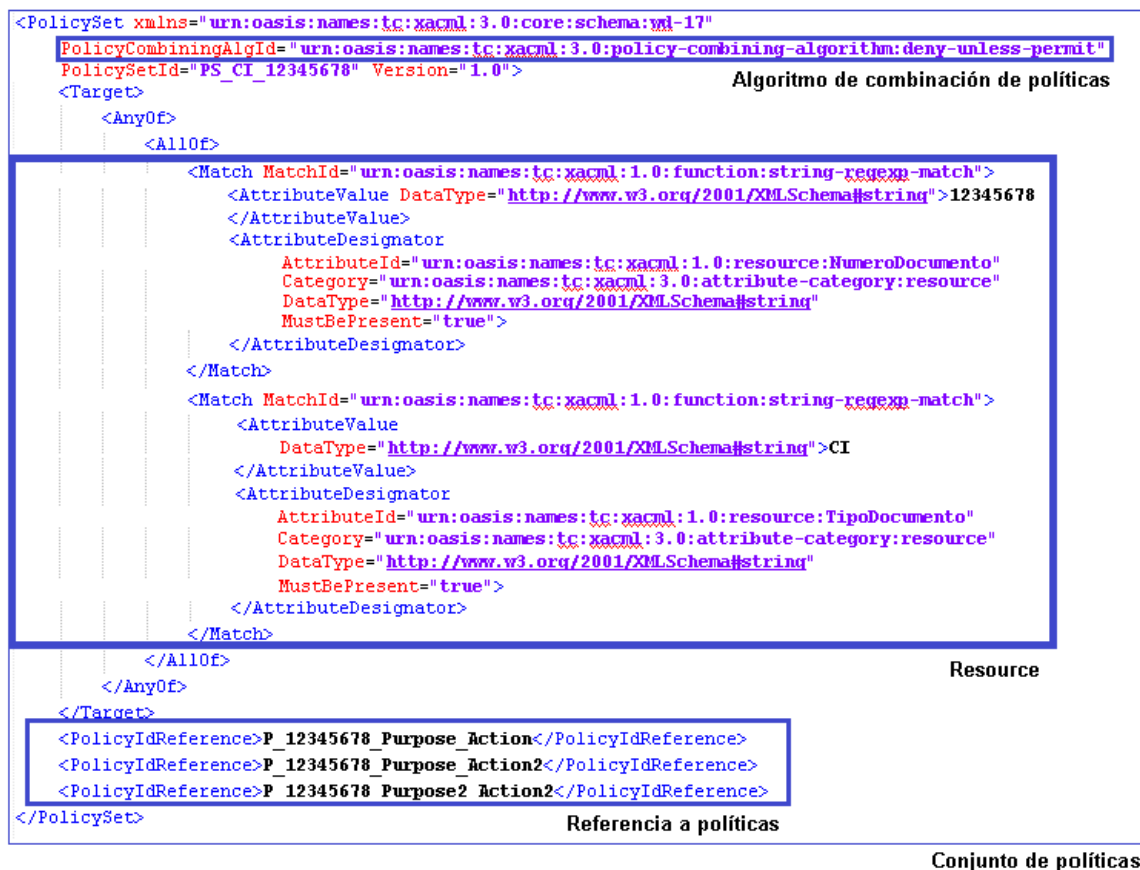


Figura 43 - Ejemplo de Policy Set

Solicitud XACML

En la Figura 44 se muestra un ejemplo genérico de solicitud XACML. Ésta indica la Acción y el Propósito que pretende realizar el Sujeto sobre el Recurso *Atributo_A* del ciudadano, cuyo tipo de documento es CI y número de documento es 12345678.

```
<Request CombinedDecision="false" ReturnPolicyIdList="false">
  <Attributes
    Category="urn:oasis:names:tc:xacml:3.0:attribute-category:action">
      <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
        IncludeInResult="false">
        <AttributeValue
          DataType="http://www.w3.org/2001/XMLSchema#string">Acción
        </AttributeValue>
        </Attribute>
      </Attributes>
      Acción
    <Attributes
      Category="urn:oasis:names:tc:xacml:3.0:attribute-category:purpose">
        <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:purpose:purpose-id"
          IncludeInResult="false">
          <AttributeValue
            DataType="http://www.w3.org/2001/XMLSchema#string">Propósito
          </AttributeValue>
          </Attribute>
        </Attributes>
        Propósito
      <Attributes
        Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject">
          <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
            IncludeInResult="false">
            <AttributeValue
              DataType="http://www.w3.org/2001/XMLSchema#string">Organismo
            </AttributeValue>
            </Attribute>
          </Attributes>
          Sujeto
        <Attributes
          Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource">
            <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:NumeroDocumento">
              IncludeInResult="true">
                <AttributeValue
                  DataType="http://www.w3.org/2001/XMLSchema#string">12345678
                </AttributeValue>
              </Attribute>
            <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:TipoDocumento">
              IncludeInResult="true">
                <AttributeValue
                  DataType="http://www.w3.org/2001/XMLSchema#string">CI
                </AttributeValue>
              </Attribute>
            </Attributes>
            Ciudadano
          <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:Target"
            IncludeInResult="true">
              <AttributeValue
                DataType="http://www.w3.org/2001/XMLSchema#string">Atributo_A
              </AttributeValue>
            </Attribute>
          </Attributes>
          Atributo solicitado
        </Attributes>
      </Request>
```

Solicitud XACML

Figura 44 - Ejemplo de solicitud XACML

En el Apéndice 2 se puede ver un ejemplo de solicitud y respuesta XACML utilizado dentro de los casos de estudio.

5.2.3. Organismos

En el contexto del proyecto, existe un conjunto de organismos que se comunica con la plataforma. Algunos de los organismos están registrados en la plataforma como orígenes de datos personales. Éstos deben contar con un Servicio Web SOAP de consulta de datos maestros. El servicio brinda al sistema de MDM información adicional para complementar el *golden record*, permitiendo de esta forma la implementación de atributos de tipo Registro. Asimismo, estos organismos deben implementar una funcionalidad de encriptación, donde todo dato solicitado que contenga el atributo *Require Encryption*, debe retornarse encriptado y con la etiqueta *Encrypted*¹⁷.

Por otro lado, existen organismos registrados como *sincronizables*. Estos deben proveer un Servicio Web SOAP de sincronización de datos maestros. El servicio permite al sistema de MDM insertar o actualizar datos definidos como *sincronizables* en dichos organismos.

Cabe destacar que la simulación de las aplicaciones de los organismos que generan solicitudes de datos personales se realiza a través de Soap UI.

5.2.4. Componente ESB

Dada la necesidad de controlar todas las interacciones MDM entre organizaciones y ciudadanos que pasen por la plataforma de integración, se hace uso de las capacidades de un ESB.

Las interacciones MDM se realizan a través de Servicios Web, brindados tanto por la plataforma como por los organismos. Es así que, para poder asegurar las leyes de protección de datos, la comunicación con los servicios debe ser mediada por un ESB, donde cada servicio tenga su correspondiente servicio virtual en la plataforma.

De esta forma se identifican cuatro tipos de servicios virtuales a implementar a través de flujos de mediación que se detallan a continuación.

Cabe destacar que el flujo de búsqueda por atributo sensible incluido en la sección 4.6.2 se deja fuera del alcance del prototipo considerando el tiempo disponible para la etapa de implementación.

¹⁷ Se asume que las claves de encriptación se acuerdan y manejan entre los organismos correspondientes.

Flujo de consulta al sistema MDM

Este flujo intercepta las invocaciones al servicio de consulta brindado por el sistema de MDM.

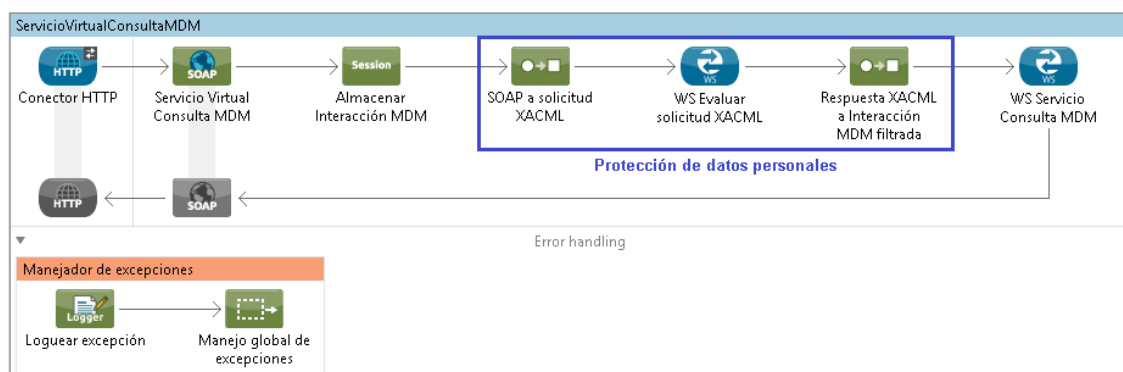


Figura 45 - Flujo de consulta MDM¹⁸

Como lo muestra la Figura 45, el flujo comienza con un conector de tipo HTTP, indicando en qué dirección y puerto está escuchando el servicio. Luego, el conector SOAP permite generar un servicio virtual que simula ser el servicio original de consulta del sistema de MDM. La configuración se realiza mediante el WSDL brindado por el servicio original.

A continuación, se almacena la interacción recibida, se genera la solicitud XACML a través de transformaciones XSLT y se invoca al servicio WS Evaluar solicitud XACML. Posteriormente, a partir de la interacción almacenada y la respuesta XACML obtenida, se crea la interacción MDM filtrada, eliminando los atributos que no hayan obtenido el permiso de acceso. Queda así definido el proceso titulado como “Protección de datos Personales”. Finalmente, se invoca al servicio WS Servicio Consulta MDM para obtener los datos consultados.

Si se produce alguna excepción en el transcurso del flujo, ésta es capturada por el “Manejador de excepciones”. Éste genera una nueva entrada en los registros de Log e invoca al flujo titulado como “Manejo global de excepciones”.

Flujo para completar el golden record

Este flujo intercepta las invocaciones al servicio de consulta brindado por los organismos.

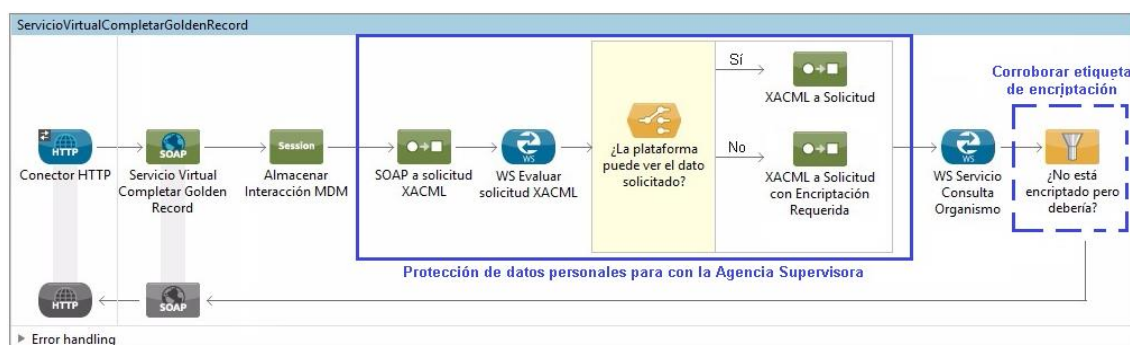


Figura 46 - Flujo para completar el golden record¹⁸

¹⁸ El flujo ha sido simplificado para lograr una mejor comprensión.

Como se puede ver en la Figura 46, el flujo comienza de manera análoga al anterior. En éste, se evalúa si la fuente externa cuenta con consentimientos para compartir la información solicitada. Si ella no cuenta con los consentimientos necesarios, se retorna una respuesta vacía. De lo contrario, se valida la protección de datos para con la Agencia Supervisora. Se genera entonces, mediante XSLT, la solicitud XACML correspondiente que es luego enviada al servicio WS Evaluar Solicitud XACML. Esta solicitud tiene como Sujeto a la propia Agencia Supervisora y la respuesta obtenida determina si puede manejar en texto claro los datos solicitados. Es así que, si la Agencia no cuenta con los permisos necesarios para visualizar dicho dato, se transforma la interacción, agregando el atributo *Require Encryption* (Encriptación Requerida).

A continuación, se realiza la invocación al servicio de consulta del Organismo. Al obtener la respuesta con la información solicitada, se valida que todo atributo requerido como encriptado, contenga la etiqueta *Encrypted* (Encriptado). En caso de que lo anterior no se cumpla, el mensaje es descartado.

Cabe destacar que, este flujo sólo es necesario cuando la Agencia Supervisora no cuenta con permisos totales sobre los datos.

Flujo de actualización del sistema MDM

Este flujo intercepta las invocaciones al servicio de actualización de datos maestros brindado por el sistema de MDM.

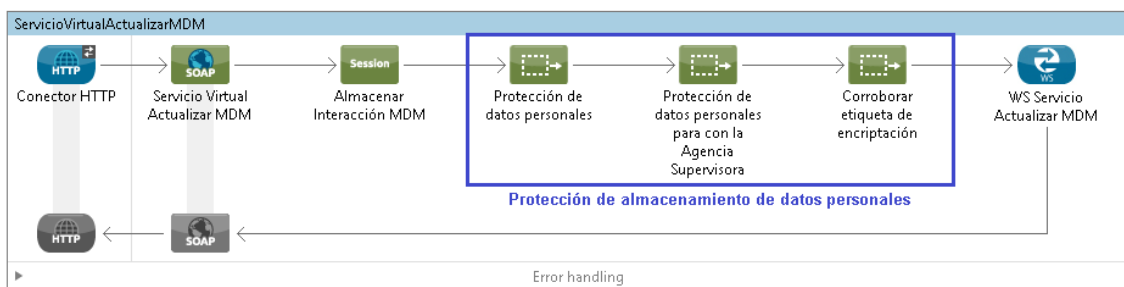


Figura 47 - Flujo de actualización del MDM¹⁸

Como se muestra en la Figura 47, el flujo comienza de manera análoga a los casos anteriores. Luego, se procede con la “Protección de almacenamiento de datos personales”, donde se hace referencia a tres sub flujos antes definidos. Primero, se invoca “Protección de datos personales” validando los permisos que tenga el emisor para realizar la acción dada y filtrando aquellos atributos sobre los cuales no se tiene permiso de acceso. Luego, se valida que la Agencia Supervisora tenga el consentimiento para almacenar los datos recibidos. Se corrobora que aquellos atributos donde la Agencia no tiene permisos, contengan la etiqueta *Encrypted*, descartando aquellos que no. Finalmente, se invoca al servicio WS Servicio Actualizar MDM.

Es importante aclarar, que los flujos “Protección de datos personales para con la Agencia Supervisora” y “Corroborar etiqueta de encriptación” son necesarios cuando la Agencia no cuenta con permisos totales sobre los datos. Además, podría ser necesario almacenar ciertos metadatos que permitan luego la des-encriptación de los atributos por los organismos pertinentes, sin embargo, esto se mantiene fuera del alcance del proyecto.

Flujo de sincronización de organismos

Este flujo intercepta las invocaciones al servicio de sincronización de datos maestros brindado por los organismos.

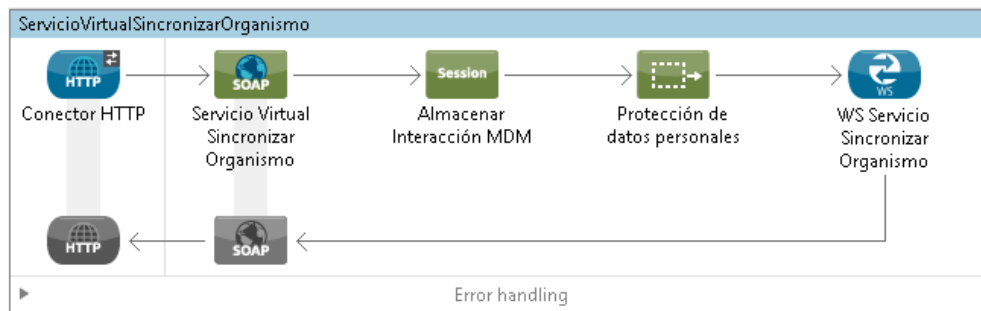


Figura 48 - Flujo sincronización organismos¹⁸

Como muestra la Figura 48, este flujo realiza la virtualización del servicio de sincronización de los organismos mediante la utilización del sub flujo “Protección de datos personales” y posterior invocación al servicio de sincronización expuesto por el organismo correspondiente.

5.2.5. Vista de distribución

En esta sección se muestran los servicios necesarios incorporados en la solución, así como también los sistemas utilizados y desarrollados mediante su vista de distribución.

En la Figura 49, se pueden visualizar los módulos ubicados en cada componente.

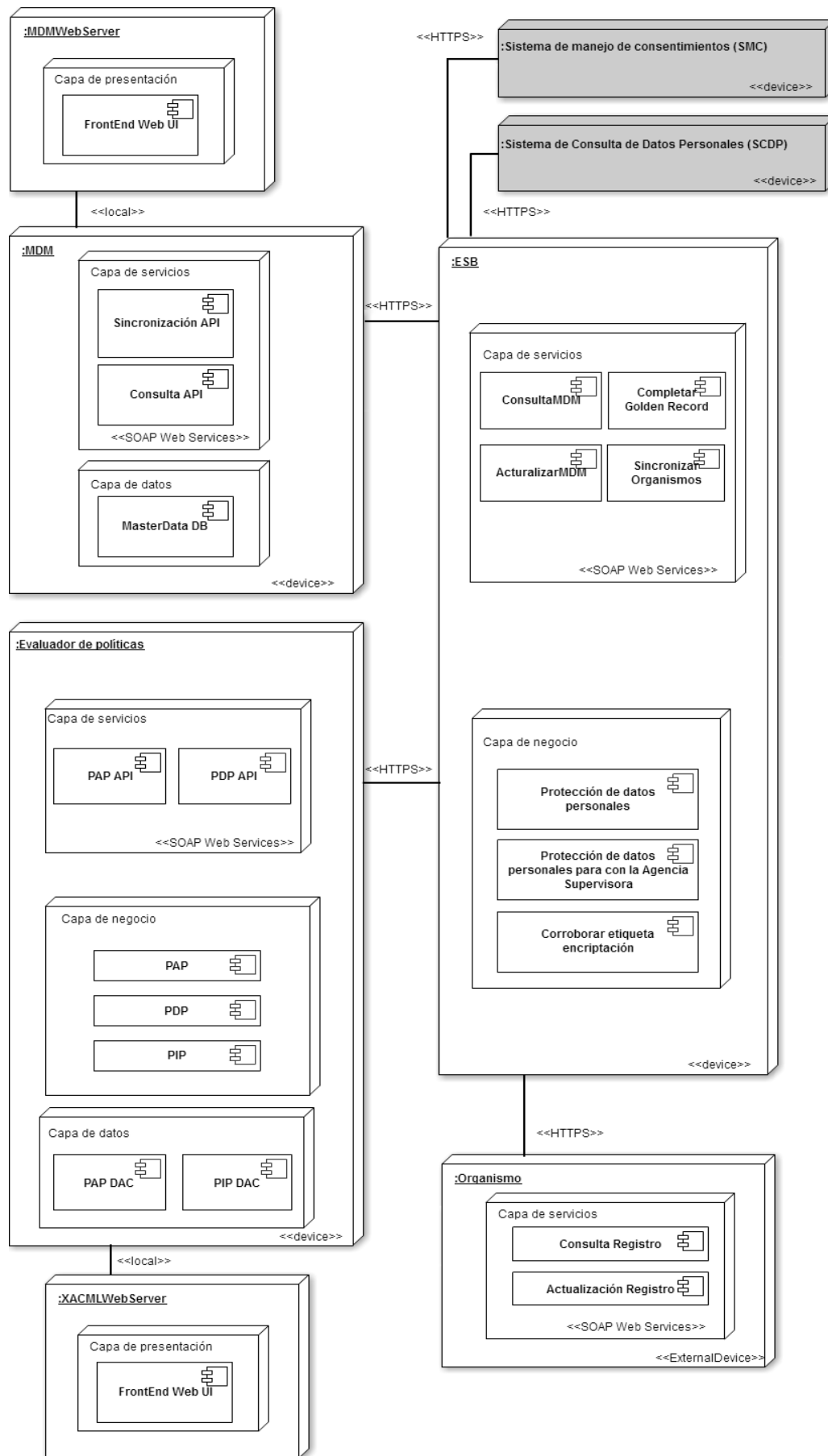


Figura 49 - Vista de distribución

El componente XACMLWebServer se comunica localmente con el componente Evaluador de políticas, permitiendo administrar la configuración del mismo.

El componente MDMWebServer se comunica localmente con el componente MDM, permitiendo administrar el modelo de datos maestros y realizar solicitudes de datos.

Los componentes SMC y SCDP se comunican con el componente ESB para administrar consentimientos y consultar datos personales respectivamente. Esta comunicación utiliza el protocolo HTTPS.

El componente ESB, mediante HTTPS, se comunica con el componente MDM para consultar datos personales, con el componente Evaluador para evaluar solicitudes y con los organismos para consultar y actualizar datos personales de éstos.

Cabe destacar que no se realiza la implementación del Sistema de Manejo de Consentimientos y el Sistema de Consulta de Datos Personales. Esta decisión se basa en que el desarrollo se centra en los aspectos relacionados con la intercepción, evaluación y filtrado de solicitudes.

6. Caso de estudio

En esta sección se presenta un caso de estudio que permite visualizar el funcionamiento del prototipo implementado.

6.1. Objetivo

El objetivo del caso de estudio es validar la solución planteada en un escenario derivado de una simplificación de la realidad actual del Estado uruguayo. Este caso permite verificar que el intercambio de datos personales entre organismos se realiza de acuerdo a los consentimientos de los ciudadanos y los aspectos considerados de la ley Nº 18.331.

En el primer ejemplo, se valida un flujo normal de acceso a datos personales. En el segundo, se analiza la participación del módulo de evaluación de políticas frente a excepciones circunstanciales. Finalmente, en el tercero se verifica el flujo de mantenimiento y sincronización de datos personales.

6.2. Descripción general caso de estudio

Para desarrollar el caso de estudio es necesario configurar una infraestructura que represente el contexto sobre el cual se desarrolla el proyecto. Es así, que se realiza una reducción del modelo de referencia de persona que propone AGESIC [62], como lo muestra la Figura 50. El esquema planteado permite representar los distintos estilos arquitectónicos de MDM presentados en Componente MDM.

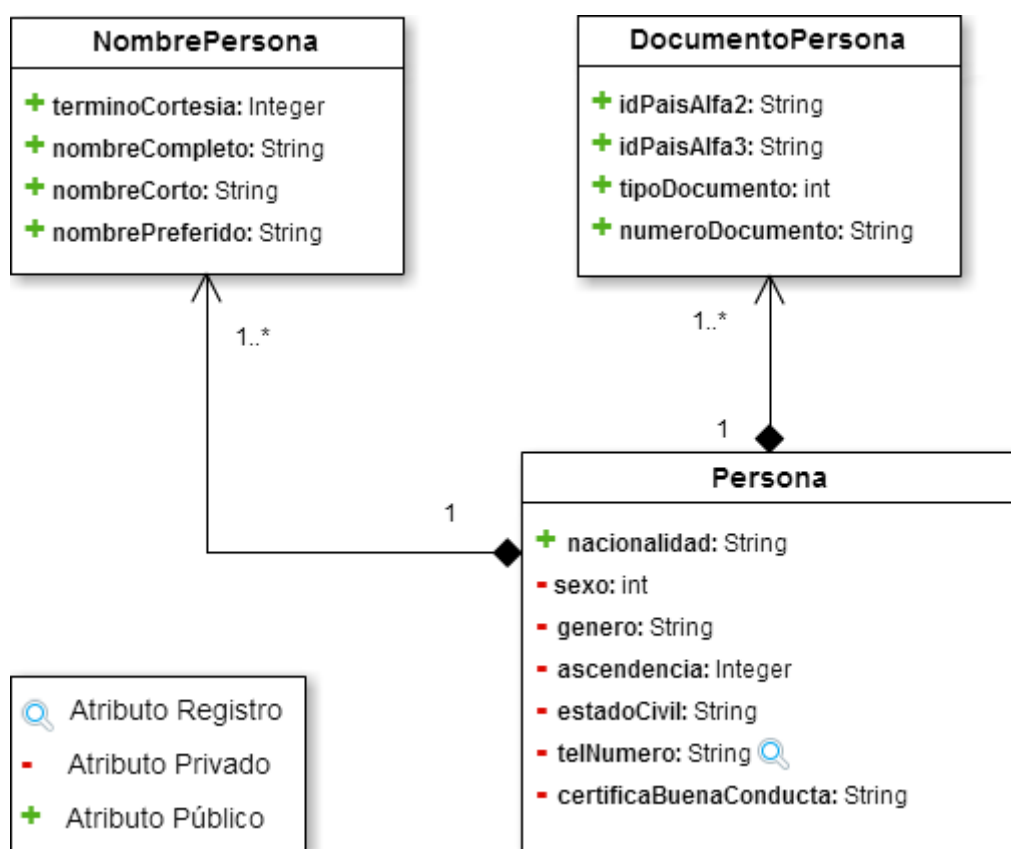


Figura 50 – Modelo persona utilizado [Adaptación modelo persona AGESIC [62]]

Los organismos registrados en la plataforma son Dirección General Impositiva (DGI), Dirección Nacional de Identificación Civil (DNIC) y Dirección Nacional de Policía Científica (DNPC).

Se asume a la DNIC como el único ente responsable del almacenamiento del atributo sensible *telNumero*, implementando este atributo como tipo Registro. Por lo tanto, este organismo se identifica como *proveedor de datos personales*, debiendo exponer un Servicio Web SOAP de consulta.

A su vez, se considera que la DNPC solicita estar en sincronización con el repositorio de datos maestros. Por lo tanto, es registrada como *sincronizable*, debiendo exponer un Servicio Web SOAP de actualización.

El modelo que se muestra en la Figura 50, en conjunto con las consideraciones antes mencionadas, se implementa en el repositorio del sistema de MDM. Para lograr esto, se utiliza Talend Open Studio como se puede ver en la Figura 51.

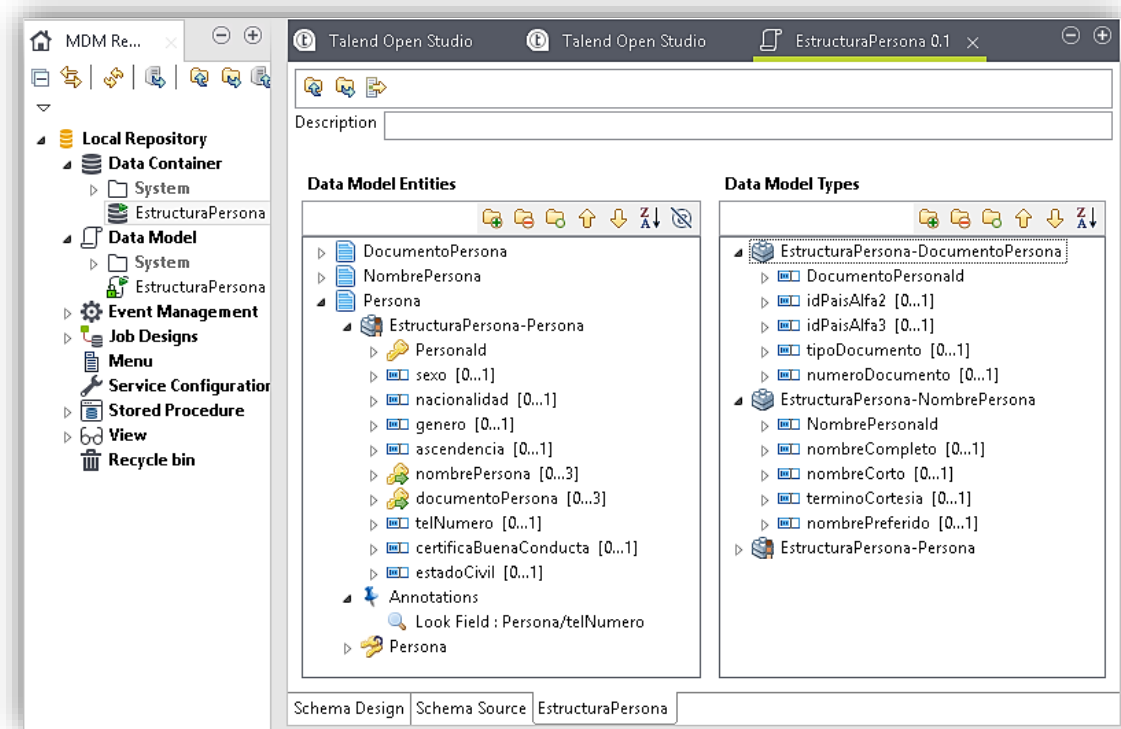


Figura 51 - Implementación del modelo de persona utilizado

Finalmente, el ciudadano cuyo identificador es 12345678 (de aquí en más el ciudadano), brinda los siguientes consentimientos:

- A la DGI, para la acción *Select_for_use* y el propósito *MDM_BusinessOriented* sobre los atributos *Genero*, *Sexo* y *Ascendencia*. Cabe destacar que *EstadoCivil* y *CertificaBuenaConducta* no cuentan con dicho consentimiento.
- A la DNIC, para la acción *Update* y el propósito *MDM_Maintain* sobre los atributos *Nacionalidad* y *Ascendencia*.

De esta forma, dentro del conjunto de políticas generadas para el ciudadano, se destacan las políticas *P_12345678_Select_Business* y *P_12345678_Update_Maintain* que se muestran en la Figura 52 y en la Figura 53 respectivamente. Éstas se corresponden con los consentimientos detallados anteriormente. El resto de las políticas definidas se incluyen en el Apéndice 1.

```

<Policy xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
  PolicyId="P_12345678_Select_Business"
  RuleCombiningAlgId="urn:oasis:names:tc:xacml:3.0:rule-combining-algorithm:deny-unless-permit"
  Version="1.0">
  <Target>
    <AnyOf>
      <AllOf>
        <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          <AttributeValue
            DataType="http://www.w3.org/2001/XMLSchema#string">Select_for_use
          </AttributeValue>
          <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
            Category="urn:oasis:names:tc:xacml:3.0:attribute-category:action"
            DataType="http://www.w3.org/2001/XMLSchema#string"
            MustBePresent="true"/>
        </Match>
        <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
            MDM_BusinessOriented</AttributeValue>
          <AttributeDesignator
            AttributeId="urn:oasis:names:tc:xacml:1.0:purpose:purpose-id"
            Category="urn:oasis:names:tc:xacml:3.0:attribute-category:purpose"
            DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true">
          </AttributeDesignator>
        </Match>
      </AllOf>
    </AnyOf>
  </Target>
  <Rule Effect="Permit" RuleId="Rule_DGI">
    <Target>
      <AnyOf>
        <AllOf>
          <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
              DataType="http://www.w3.org/2001/XMLSchema#string">DGI</AttributeValue>
            <AttributeDesignator
              AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
              Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
              DataType="http://www.w3.org/2001/XMLSchema#string"
              MustBePresent="true"/>
            </Match>
          </AllOf>
        </AnyOf>
      </Target>
      <Condition>
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:or">
          <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-is-in">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
              EstructuraPersona.Persona_Genero</AttributeValue>
            <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:resource:Target"
              Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
              DataType="http://www.w3.org/2001/XMLSchema#string"
              MustBePresent="true"/>
            </Apply>
          <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-is-in">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
              EstructuraPersona.Persona_Sexo</AttributeValue>
            <AttributeDesignator
              AttributeId="urn:oasis:names:tc:xacml:1.0:resource:Target"
              Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
              DataType="http://www.w3.org/2001/XMLSchema#string"
              MustBePresent="true"/>
            </Apply>
          <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-is-in">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
              EstructuraPersona.Persona_Ascendencia</AttributeValue>
            <AttributeDesignator
              AttributeId="urn:oasis:names:tc:xacml:1.0:resource:Target"
              Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
              DataType="http://www.w3.org/2001/XMLSchema#string"
              MustBePresent="true"/>
            </Apply>
          <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-is-in">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
              EstructuraPersona.Persona_TelNumero</AttributeValue>
            <AttributeDesignator
              AttributeId="urn:oasis:names:tc:xacml:1.0:resource:Target"
              Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
              DataType="http://www.w3.org/2001/XMLSchema#string"
              MustBePresent="true"/>
            </Apply>
          </Apply>
        </Condition>
      </Rule>
    </Policy>

```

Acción y Propósito

Organismo

Datos personales con consentimiento

Figura 52 - Política P_12345678_Select_Business del ciudadano

```

<Policy xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
  PolicyId="P_12345678_Update_Maintain"
  RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:first-applicable"
  Version="1.0">
  <Target>
    <AnyOf>
      <AllOf>
        <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          <AttributeValue
            DataType="http://www.w3.org/2001/XMLSchema#string">Update
          </AttributeValue>
          <AttributeDesignator
            AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
            Category="urn:oasis:names:tc:xacml:3.0:attribute-category:action"
            DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true">
          </AttributeDesignator>
        </Match>
        <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          <AttributeValue
            DataType="http://www.w3.org/2001/XMLSchema#string">MDM_Maintain
          </AttributeValue>
          <AttributeDesignator
            AttributeId="urn:oasis:names:tc:xacml:1.0:purpose:purpose-id"
            Category="urn:oasis:names:tc:xacml:3.0:attribute-category:purpose"
            DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true">
          </AttributeDesignator>
        </Match>
      </AllOf>
    </AnyOf>
  </Target>
  <Rule Effect="Permit" RuleId="Rule_DNIC">
    <Target>
      <AnyOf>
        <AllOf>
          <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
              DataType="http://www.w3.org/2001/XMLSchema#string">DNIC</AttributeValue>
            <AttributeDesignator
              AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
              Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
              DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true">
            </AttributeDesignator>
          </Match>
        </AllOf>
      </AnyOf>
    </Target>
    <Condition>
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:or">
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-is-in">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
            EstructuraPersona.Persona_Ascendencia</AttributeValue>
          <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:resource:Target"
            Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
            DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true">
          </AttributeDesignator>
        </Apply>
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-is-in">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
            EstructuraPersona.Persona_Nacionalidad</AttributeValue>
          <AttributeDesignator
            AttributeId="urn:oasis:names:tc:xacml:1.0:resource:Target"
            Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
            DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true">
          </AttributeDesignator>
        </Apply>
      </Apply>
    </Condition>
  </Rule>
</Policy>

```

Acción y propósito

Organismo

Datos personales con consentimiento

Figura 53 - Política P_12345678_Update_Maintain del ciudadano

Además, se crea una política dentro del Árbol de Excepciones como se muestra en la Figura 54. Ésta permite a la DNIC acceder al atributo sensible *CertificaBuenaConducta* para el Propósito

MDM_BusinessOriented y la Acción *Select_for_use*, siempre que exista un expediente sobre el ciudadano en cuestión.

```

<Policy xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
  PolicyId="E_Select_Business"
  RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:first-applicable"
  Version="1.0">
  <Target>
    <AnyOf>
      <AllOf>
        <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          <AttributeValue
            DataType="http://www.w3.org/2001/XMLSchema#string">Select_for_use
          </AttributeValue>
          <AttributeDesignator
            AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
            Category="urn:oasis:names:tc:xacml:3.0:attribute-category:action"
            DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true">
          </AttributeDesignator>
        </Match>
        <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          <AttributeValue
            DataType="http://www.w3.org/2001/XMLSchema#string">MDM_BusinessOriented
          </AttributeValue>
          <AttributeDesignator
            AttributeId="urn:oasis:names:tc:xacml:1.0:purpose:purpose-id"
            Category="urn:oasis:names:tc:xacml:3.0:attribute-category:purpose"
            DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true">
          </AttributeDesignator>
        </Match>
      </AllOf>
    </AnyOf>
  </Target>
  <Rule Effect="Permit" RuleId="Rule0">
    <Target>
      <AnyOf>
        <AllOf>
          <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
              DataType="http://www.w3.org/2001/XMLSchema#string">DGI
            </AttributeValue>
            <AttributeDesignator
              AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
              Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
              DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true">
            </AttributeDesignator>
          </Match>
          <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
              DataType="http://www.w3.org/2001/XMLSchema#string">EstructuraPersona.Persona_CertificaBuenaConducta
            </AttributeValue>
            <AttributeDesignator
              AttributeId="urn:oasis:names:tc:xacml:1.0:resource:Target"
              Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
              DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true">
            </AttributeDesignator>
          </Match>
        </AllOf>
      </AnyOf>
    </Target>
    <Condition>
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of">
        <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          <AttributeValue
            DataType="http://www.w3.org/2001/XMLSchema#string">ExisteExpediente
          </AttributeValue>
          <AttributeDesignator
            Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
            AttributeId="http://fing.pdpinmdm.com/id/group"
            DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="false">
          </AttributeDesignator>
        </Apply>
      </Condition>
    </Rule>
  </Policy>

```

Acción y Propósito

Organismo y Datos personal

Atributo evaluado por el PIP

Figura 54 - Política de excepción

Además, existe un expediente sobre el ciudadano con identificador 12345678 ingresado en la fuente de datos de excepciones como se puede ver en la Figura 55.

idExpendiente	personaId	fechaCreación	materia	estado
50012	12345678	2016-03-13	Obligación de dar suma de dinero por pagaré adeudado	en_curso
50020	44888789	2016-03-13	Obligación de dar suma de dinero por pagaré adeudado	en_curso
50332	12345678	2016-03-01	Obligación de dar suma de dinero por pagaré adeudado	en_curso
50351	12345678	2016-03-13	Obligación de dar suma de dinero por pagaré adeudado	en_curso
50450	12345678	2016-03-13	Obligación de dar suma de dinero por pagaré adeudado	en_curso

Figura 55 - Ejemplo de bases de datos de expedientes

6.3. Consulta de datos personales

Cuando la DGI realiza una consulta de datos personales, se desencadenan las interacciones que se describen en la Figura 56.

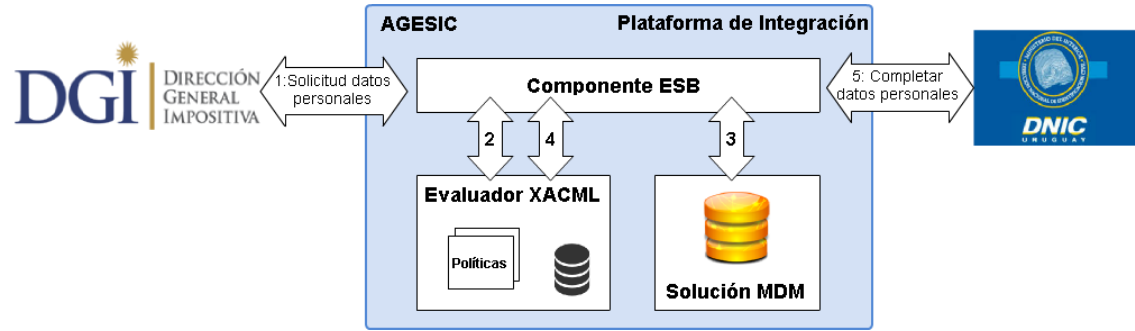


Figura 56 - Consulta de datos personales

Es así que, para realizar la consulta de datos personales, la DGI solicita información (1) del ciudadano con el identificador 12345678 para la Acción *Select_for_use*, con el Propósito *MDM_BusinessOriented* como se puede ver en la Figura 57.



Figura 57 - Solicitud SOAP de consulta de datos personales

Para validar el acceso a los atributos sensibles (2), se evalúa el conjunto de políticas del ciudadano, que admite a la DGI el acceso a *Sexo*, *Ascendencia*, *Genero* y *TelNumero*, excluyendo *EstadoCivil* (3). Además, al ser el dato sensible *TelNumero* de tipo Registro, es solicitado a la DNIC. Por lo tanto, se valida que la DNIC tenga el consentimiento para el Propósito *MDM_BusinessOriented* y la Acción *Select_for_share* sobre el atributo *TelNumero* (4). Se asume que dicho consentimiento ha sido otorgado. Sin embargo, al no existir una política que permita a la Agencia Supervisora visualizar dicho dato, se lo solicita de manera encriptada (5).

Por otro lado, todo atributo público solicitado con el propósito *MDM_BusinessOriented* queda admitido por la política *P_PublicAttributes* detallada en el Apéndice 1.

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <runJobReturn xsi:type="nsl:runJobReturn"
      xmlns="http://talend.org" xmlns:nsl="http://talend.org">
      <nsl:item xsi:type="nsl:ArrayOf_xsd_string">
        <nsl:item xsi:type="xsd:string">
          <![CDATA[<?xml version="1.0" encoding="ISO-8859-15"?>
            <Persona>
              <PersonaId>12345678</PersonaId>
              <Sexo>1</Sexo>
              <Ascendencia>Otra</Ascendencia>
              <Genero>Masculino</Genero>
              <Nacionalidad>URY</Nacionalidad>
              <TelNumero Encrypted="True"
                EncryptedBy="DNIC">OuRajKsK9Zb4Fx/D1ORd0A==
              </TelNumero>
              <NombrePersona>
                <NombreCompleto>Juan Carlos Navarro Montoya</NombreCompleto>
                <NombrePersonaId>[12345678]</NombrePersonaId>
                <NombreCorto>Juan Navarro</NombreCorto>
                <TerminoCortesía>1</TerminoCortesía>
                <NombrePreferido>Juan</NombrePreferido>
              </NombrePersona>
              <DocumentoPersona>
                <DocumentoPersonaId>[12345678]</DocumentoPersonaId>
                <IdPaisAlfa2>UY</IdPaisAlfa2>
                <IdPaisAlfa3>URY</IdPaisAlfa3>
                <TipoDocumento>68909</TipoDocumento>
                <NumeroDocumento>12345678</NumeroDocumento>
              </DocumentoPersona>
            </Persona>]]>
          </nsl:item>
        </nsl:item>
      </runJobReturn>
    </soap:Body>
  </soap:Envelope>

```

**Atributos
sensibles**

**Atributo
tipo registro**

Figura 58 - Respuesta SOAP a consulta de datos personales

En la Figura 58 se puede ver la respuesta que recibe la DGI. En ésta se encuentran los atributos públicos y los sensibles que fueron consentidos. Cabe destacar, que el atributo *EstadoCivil* fue filtrado al no existir un consentimiento asociado. A su vez, el atributo *TelNumero* presenta las etiquetas *Encrypted* y *EncryptedBy*, que brindan información a la DGI para lograr su descryptado.

6.4. Acceso a datos sensibles con excepción

Cuando la DGI realiza una consulta por un dato personal con excepción, se desencadenan las interacciones que se describen en la Figura 59.

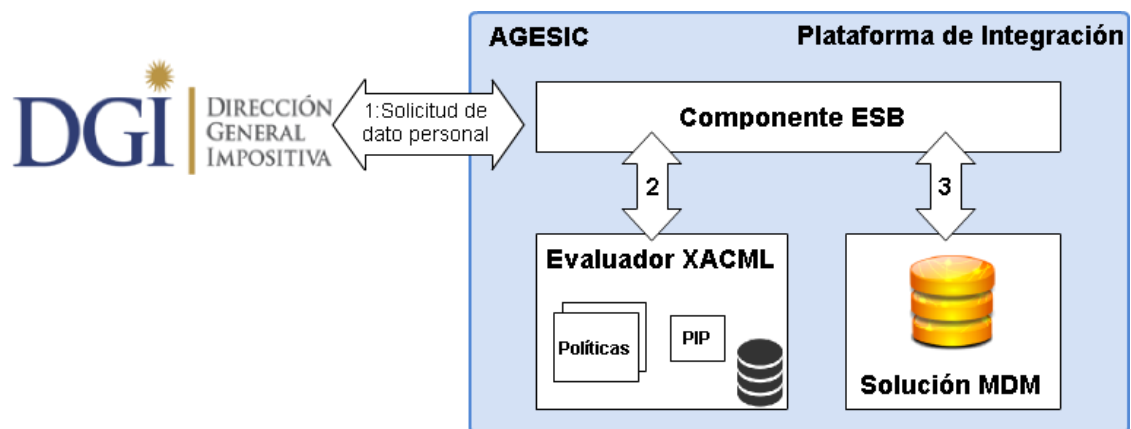


Figura 59 - Acceso a datos sensibles con excepción

En este ejemplo, la DGI solicita el atributo sensible *CertificaBuenaConducta* (1) como se puede ver en la Figura 60.

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap
  <soapenv:Header>
    <SubjectId>DGI</SubjectId>
    <PurposeId>MDM_BusinessOriented</PurposeId>
    <ActionId>Select_for_use</ActionId>
  </soapenv:Header>
  <soapenv:Body>
    <tal:args>
      <item>
        <![CDATA[
          <Persona>
            <PersonaId>12345678</PersonaId>
            <CertificaBuenaConducta/>
          </Persona>
        ]]>
      </item>
    </tal:args>
  </soapenv:Body>
</soapenv:Envelope>

```

Dato personal solicitado

Solicitud de atributo sensible no consentido

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <runJobReturn xsi:type="ns1:runJobReturn" xmlns="http://talend.org">
      <ns1:item xsi:type="ns1:ArrayOf_xsd_string">
        <ns1:item xsi:type="xsd:string">
          <![CDATA[<?xml version="1.0" encoding="ISO-8859-15"?>
            <Persona>
              <PersonaId>12345678</PersonaId>
              <CertificaBuenaConducta>True</CertificaBuenaConducta>
            </Persona>]]>
          </ns1:item>
        </ns1:item>
      </runJobReturn>
    </soap:Body>
  </soap:Envelope>

```

Dato personal obtenido

Respuesta obtenida

Figura 60 - Interacción sobre atributo sensible no consentido

Cabe destacar que el atributo no fue consentido por el ciudadano. Sin embargo, a partir de la Política de Excepciones, el módulo PIP verifica la existencia de un expediente sobre el ciudadano relativo al Propósito y Acción dados (2), permitiendo así el acceso a dicho atributo (3).

6.5. Actualización de datos personales

Cuando la DGI realiza una actualización de datos personales, se desencadenan las interacciones que se describen en la Figura 61.

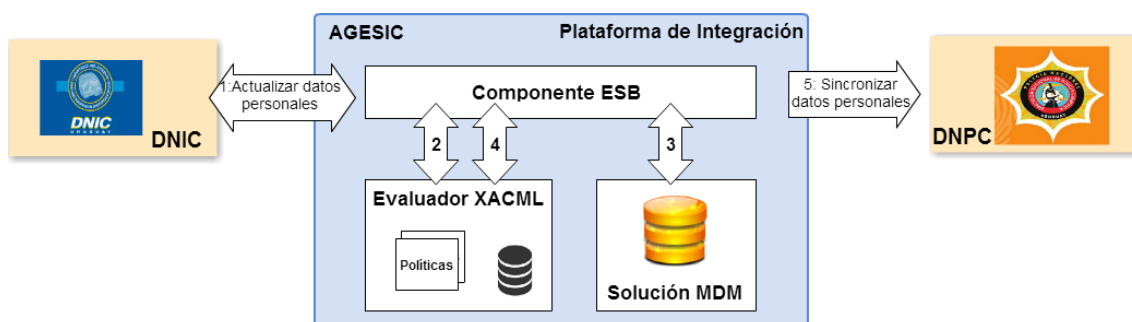


Figura 61 - Actualización de datos personales

En este ejemplo se realiza la actualización de datos personales del ciudadano (1), verificando que se realice la sincronización pertinente en los organismos registrados como *sincronizables*.

Los datos personales del ciudadano almacenados en el repositorio del MDM al momento de realizar la solicitud se pueden ver en la Figura 62.

☐	PersonalId	sexo	nacionalid...	genero	ascendencia	certificaBuenaConducta	estadoCivil
☐	7654321	2	URY	Femenino	Afro	True	Casado
☐	4488555	1	URY	Masculino	Amarilla	True	Soltero
☐	4758598	1	URY	Masculino	Blanca	True	Casado
☐	4488577	1	URY	Masculino	Indigena	False	Soltero
☐	4488579	2	URY	Masculino	Blanco	True	Soltero
☒	12345678	1	URY	Masculino	Otra	True	Soltero

Figura 62 - Estado del repositorio previo a la solicitud

Luego, como se puede ver en la Figura 63, se realiza la solicitud de actualización de datos personales. En ésta, se indican nuevos valores de *Nacionalidad*, *Genero* y *Ascendencia* siendo estos *ARG*, *Femenino* y *Afro* respectivamente.

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  <soapenv:Header>
    <SubjectId>DNIC</SubjectId>
    <PurposeId>MDM_Maintain</PurposeId>
    <ActionId>Update</ActionId>
  </soapenv:Header>
  <soapenv:Body>
    <tal:args>
      <item>
        <![CDATA[
          <Persona>
            <PersonaId>12345678</PersonaId>
            <Nacionalidad>ARG</Nacionalidad>
            <Genero>Femenino</Genero>
            <Ascendencia>Afro</Ascendencia>
          </Persona>
        ]]>
      </item>
    </tal:args>
  </soapenv:Body>
</soapenv:Envelope>

```

Datos a actualizar

Solicitud de actualización

Figura 63 - Solicitud de actualización de datos personales

Luego de verificar las políticas del ciudadano (2), se actualizan únicamente los atributos que tienen consentimiento para la Acción y Propósito dados. En este caso, *Nacionalidad* y *Ascendencia* admiten cambios, mientras que el atributo *Genero* se mantiene sin modificaciones como se puede ver en la Figura 64 (3).

☐	Personald	sexo	nacionalid...	genero	ascendencia	certificaBuenaConducta	estadoCivil
☐	7654321	2	URY	Femenino	Afro	True	Casado
☐	4488555	1	URY	Masculino	Amarilla	True	Soltero
☐	4758598	1	URY	Masculino	Blanca	True	Casado
☐	4488577	1	URY	Masculino	Indigena	False	Soltero
☐	4488579	2	URY	Masculino	Blanco	True	Soltero
☒	12345678	1	ARG	Masculino	Afro	True	Soltero

Figura 64 - Estado del repositorio posterior a la solicitud

Finalmente, como la DNPC ha sido registrada como fuente de datos *sincronizable*, se invoca al servicio de sincronización provisto por el organismo (5). Cabe destacar que la invocación es filtrada según los consentimientos definidos para éste (4).

6.6. Conclusiones del caso de estudio

El caso de estudio permitió verificar que se realiza la validación las interacciones MDM generadas por los organismos cuando se consultan o actualizan datos personales. También aquellas interacciones generadas por el sistema de MDM hacia las fuentes externas para completar registros o sincronizar información. Esto se realiza considerando los consentimientos de los ciudadanos, excepciones definidas por la ley y atributos de carácter público definidos en la misma.

7. Conclusiones y trabajo futuro

En este capítulo se presentan las conclusiones del proyecto y se enumeran posibles mejoras a realizar como trabajo a futuro.

7.1. Conclusiones

En el marco del gobierno electrónico, es necesario que las plataformas de integración brinden mecanismos para monitorear y hacer cumplir las normas de protección de datos personales. Particularmente, en el intercambio de información inter-organizacional en soluciones de MDM y utilizando mecanismos estándares para el control de acceso, como lo es XACML.

Para resolver esta problemática, se analizaron las soluciones propuestas en el artículo “*Enforcing Data Protection Regulations within e-Government Master Data Management Systems*” [1]. Además, se estudió la Ley de Protección de Datos Personales N° 18.331 en el contexto del Estado uruguayo y la Plataforma de Gobierno Electrónico del Uruguay.

El estudio permitió el diseño y desarrollo de un prototipo para asegurar la protección de datos personales según la Ley antes mencionada. El diseño se enmarca en una plataforma de gobierno electrónico, haciendo uso de tecnologías de Master Data Management. La implementación se basa en tecnologías de ESB en combinación con un sistema que implementa el protocolo XACML. Dicha solución permite asegurar las interacciones MDM acorde a las normas establecidas, considerando los consentimientos brindados por los ciudadanos.

Además, se realizaron casos de estudio basados en la realidad del Estado uruguayo que permitieron tener una primera validación de la solución.

En relación al estándar XACML, se constató que es un protocolo que resulta de utilidad cuando se conoce el ciudadano consultado. Sin embargo, cuando no se conoce, la evaluación de consentimientos resulta más compleja, lo que podría afectar la eficiencia de la solución. Por otro lado, se destaca que el protocolo permite definir consentimientos como políticas de forma intuitiva.

En cuanto a la gestión de datos maestros, una solución de MDM resulta ser un componente clave en el contexto de gobierno electrónico, permitiendo hacer un uso adecuado de la información y logrando obtener datos uniformes y exactos en todo momento.

A su vez, se identificó que el estilo arquitectónico de MDM más adecuado para el contexto de la Plataforma de Gobierno Electrónico es el Híbrido. Donde se almacene todos los datos públicos en el repositorio y que cada organismo mantenga los datos sensibles de los que es responsable.

En relación a la tecnología de *middleware* ESB, se identificó que resulta útil al permitir mediar y monitorear las comunicaciones, logrando realizar las validaciones pertinentes.

En cuanto a las herramientas utilizadas, se destaca la facilidad con la que *WSO2 IS* permite implementar el protocolo XACML, así como también la capacidad de integración con otros sistemas, brindando sus funcionalidades a través de Servicios Web. Por otro lado, se identificó que si bien *Talend MDM Open Studio* ofrece un conjunto de operaciones básicas para la

gestión de datos maestros y permite realizar una extensión de ellas para alcanzar distintos estilos arquitectónicos, este proceso puede requerir el desarrollo de *Jobs* complejos resultando en un esfuerzo de gran costo frente a otros productos que brinden éstas capacidades ya implementadas. Finalmente, el entorno de desarrollo de *MuleSoft* ESB permitió la creación y despliegue de flujos de mediación de manera ágil y sencilla.

La solución diseñada presenta un bajo acoplamiento para con el sistema de MDM y el Evaluador de políticas. Esto implica que cualquiera de ellos puede ser reemplazado sin generar un alto impacto.

Si se considera que la Agencia Supervisora cuenta con permisos totales sobre los datos maestros, se logra una solución de mejor rendimiento. Esto es así, porque no se solicitará la encriptación de atributos a los organismos, evitando el uso de mecanismos criptográficos entre ellos.

Por otro lado, dada la generalidad con la que se desarrolló la solución es posible extenderla a otros ámbitos fuera del contexto de la Plataforma de Gobierno Electrónico. Un ejemplo aplicable sería en ámbito de la Salud, para asegurar la protección de datos de los pacientes, haciendo uso de la historia clínica electrónica.

7.2. Trabajo a futuro

En esta sección se detallan posibles líneas a seguir para mejorar o continuar el trabajo realizado.

- **Validar formatos de mensaje**

La implementación realizada no valida el formato de los mensajes interceptados. Sería entonces de interés realizar una validación del formato de los mensajes que se interceptan, asegurando que éstos presenten correctamente los atributos requeridos, como ser *Purpose*, *Action* y *SubjectId*.

- **Desarrollo de Sistema de Manejo de Consentimientos**

Si bien se consideró un sistema de manejo de consentimientos, éste no fue implementado, quedando entonces planteado como trabajo futuro. Dicho sistema permitiría a los ciudadanos la gestión de sus consentimientos de forma interactiva.

- **Desarrollo de Sistema de Consulta de Datos Personales**

Si bien se consideró un sistema de consulta de datos personales, éste no fue implementado, quedando entonces planteado como trabajo futuro. Dicho sistema permitiría a los ciudadanos gestionar sus datos personales e identificar a los responsables de su almacenamiento.

- **Registro de bases de datos de organizaciones**

Actualmente la configuración de las bases de datos de organizaciones se realiza de manera estática, definiendo la información de los organismos registrados en la plataforma en archivos de configuración. Sería de utilidad poder gestionar estos registros de forma interactiva, facilitando así la incorporación o modificación de los mismos.

- **Implementar consultas a partir de atributos que no identifican al ciudadano**

Si bien se consideraron las consultas a partir de atributos sensibles, no se implementó la lógica necesaria para la validación de las mismas. Sería de utilidad incorporar esta lógica para aumentar la funcionalidad del sistema.

- **Sistema de mantenimiento del modelo de datos personales**

Actualmente la configuración del modelo de datos personales utilizado en la solución se realiza de forma independiente en cada uno de los componentes involucrados (Componente MDM, Componente ESB, Componente Evaluador) implicando un mayor costo de mantenimiento. Sería de utilidad poder gestionar el modelo de forma interactiva, impactando en los distintos componentes, disminuyendo así el costo de mantenimiento.

- **Considerar la intercepción de otro tipo de comunicación**

Para este proyecto se utilizó como base la comunicación a través de Servicios Web. Sin embargo, podrían existir otros mecanismos que involucren colas de mensajes o bases de datos, sobre los cuales se debería aplicar la protección de datos personales.

- **Monitoreo y alertas**

Actualmente no se monitorea la actividad del sistema. Sería de interés incorporar a la solución un sistema de monitoreo y alertas que permita entre otras cosas detectar y evitar posibles ataques de seguridad. Dadas ciertas condiciones detectadas se podría determinar la penalización de organismos.

- **Generación de reportes**

Actualmente no se generan reportes para auditar el funcionamiento del sistema. Sería de interés incorporar un generador de reportes, donde éstos pueden estar asociados a un organismo o un ciudadano. En el primer caso, podría resultar de interés conocer la cantidad de accesos permitidos o denegados, el tiempo promedio de respuesta, historial de penalizaciones, entre otros. Para el caso de los ciudadanos, podría ser relevante conocer datos estadísticos sobre solicitudes de rectificación de datos personales.

- **Cambios en la Ley**

La Ley de Protección de Datos Personales podría ser modificada a consecuencia del impulso de nuevas tecnologías de almacenamiento de información o nuevas regulaciones tanto de carácter nacional como internacional.

Un ejemplo válido a analizar, puede ser que el acceso a cualquiera de los datos personales de los ciudadanos menores de edad requiera de consentimiento, es decir, todos sus datos personales pasan a ser de carácter sensible.

8. Referencias

- [1] Federico Piedrabuena, Laura González, y Raúl Ruggia, «Enforcing Data Protection Regulations within e-Government Master Data Management Systems», presentado en 17th International Conference on Enterprise Information Systems, Barcelona, España, 2015.
- [2] M. Mecella, M. Scannapieco, A. Virgillito, R. Baldoni, T. Catarci, y C. Batini, «Managing Data Quality in Cooperative Information Systems», presentado en On the Move to Meaningful Internet Systems, London, UK, UK, 2002, pp. 486-502.
- [3] Laura González y Raúl Ruggia, «Towards a Compliance-Aware Inter-organizational Service Integration Platform», presentado en On the Move to Meaningful Internet Systems: OTM 2014 Workshops, 2014, pp. 8–17.
- [4] Laura González, Raúl Ruggia, J. Abin, Guzmán Llambías, R. Sosa, Bruno Rienzi, D. Bello, y F. Alvarez, «A Service-oriented Integration Platform to Support a Joined-up E-government Approach: The Uruguayan Experience», 2012.
- [5] I. Boydens, «Strategic Issues Relating to Data Quality for E-Government: Learning from an Approach Adopted in Belgium», en *Practical Studies in E-Government: Best Practices from Around the World*, S. Assar, I. Boughzala, y I. Boydens, Eds. New York, NY: Springer New York, 2011, pp. 113-130.
- [6] R. Shirey, «RFC 2828 - Internet Security Glossary». [En línea]. Disponible en: <http://www.ietf.org/rfc/rfc2828.txt>. [Accedido: 07-feb-2016].
- [7] «Glosario de términos de seguridad informática». [En línea]. Disponible en: <http://nvlpubs.nist.gov/nistpubs/ir/2013/NIST.IR.7298r2.pdf>.
- [8] Franks, et al., «RFC 2617 - HTTP Authentication». [En línea]. Disponible en: <https://tools.ietf.org/html/rfc2617#section-2>.
- [9] Alan O. Freier, Philip Karlton, y Paul C. Kocher, «The SSL Protocol Version 3.0», 18-nov-1996. [En línea]. Disponible en: <https://tools.ietf.org/html/draft-ietf-tls-ssl-version3-00>. [Accedido: 12-may-2016].
- [10] Ed Simon, Paul Madsen, y Carlisle Adams, «An Introduction to XML Digital Signatures». [En línea]. Disponible en: <http://www.xml.com/pub/a/2001/08/08/xmlldsig.html>. [Accedido: 07-feb-2016].
- [11] Takeshi Imamura, Blair Dillaway, y Ed Simon, «XML Encryption Syntax and Processing». [En línea]. Disponible en: <https://www.w3.org/TR/2002/REC-xmlenc-core-20021210/Overview.html#sec-Overview>. [Accedido: 07-feb-2016].
- [12] «Chapter 1: Service Oriented Architecture (SOA)». [En línea]. Disponible en: <https://msdn.microsoft.com/en-us/library/bb833022.aspx>. [Accedido: 09-feb-2016].
- [13] «SOA and Web Services». [En línea]. Disponible en: <http://www.oracle.com/technetwork/articles/javase/soa-142870.html>. [Accedido: 09-feb-2016].
- [14] David Booth, et al., «Web Services Architecture». [En línea]. Disponible en: <https://www.w3.org/TR/ws-arch/wsa.pdf>. [Accedido: 07-feb-2016].
- [15] Erik Christensen, Francisco Curbera, Greg Meredith, y Sanjiva Weerawarana, «Web Services Description Language (WSDL) 1.1». [En línea]. Disponible en: <https://www.w3.org/TR/wsdl>.
- [16] Martin Gudgin, et al., «SOAP Specifications». [En línea]. Disponible en: <https://www.w3.org/TR/soap12/>. [Accedido: 07-feb-2016].
- [17] Anthony Nadalin, et al., «Web Services Security: SOAP Message Security 1.1 (WS-Security 2004)». [En línea]. Disponible en: <https://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf>.

- [18] Michael P. Papazoglou, *Web Services and SOA: Principles and Technology (2nd Edition)*, 2.^a ed. PEARSON, 2012.
- [19] Don Box, et al., «Web Services Addressing (WS-Addressing)». [En línea]. Disponible en: <https://www.w3.org/Submission/ws-addressing/>. [Accedido: 07-feb-2016].
- [20] «MDM (Master Data Management)». [En línea]. Disponible en: <http://www.informationbuilders.es/mdm-master-data-management>. [Accedido: 07-feb-2016].
- [21] Adriana Rodriguez y Luis Fernando Ortiz, «Master Data Management - Modelado y Gestión de Información». [En línea]. Disponible en: <http://es.slideshare.net/lfortizr/master-data-management-5689900>. [Accedido: 07-feb-2016].
- [22] «Cómo seleccionar la edición correcta de IBM InfoSphere MDM para un proyecto de Integración de Datos de Cliente». [En línea]. Disponible en: <http://www.ibm.com/developerworks/ssa/library/infosphere-mdm/>.
- [23] Chappell, David., «Enterprise Service Bus: Theory in Practice». .
- [24] Hutchison, Dan Wolfson, y Marcia Stockton, et al., «SOA programming model for implementing Web services, Part 4: An introduction to the IBM Enterprise Service Bus.» .
- [25] Gregor Hohpe y Bobby Woolf, *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*, 1st ed. Addison-Wesley.
- [26] James Clark, «XSL Transformations (XSLT)». [En línea]. Disponible en: <https://www.w3.org/TR/xslt>. [Accedido: 06-mar-2016].
- [27] Satadru Roy y Chris Madrid, «SOA Patterns | Candidate Patterns | Service Virtualization». [En línea]. Disponible en: http://soapatterns.org/candidate_patterns/service_virtualization. [Accedido: 06-mar-2016].
- [28] «Patrones de uso común», 17-mar-2016. [En línea]. Disponible en: https://www.ibm.com/support/knowledgecenter/api/content/nl/es/SS964W/com.ibm.wbpm.main.doc/topics/esbprog_patterns.html. [Accedido: 08-may-2016].
- [29] Erik Rissanen, «eXtensible Access Control Markup Language (XACML) Version 3.0». [En línea]. Disponible en: <http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.pdf>. [Accedido: 07-feb-2016].
- [30] AGESIC (Agencia para el Desarrollo del Gobierno de Gestión Electrónica y la Sociedad de la Información y el Conocimiento), «Descripción y guías de uso de la Plataforma de Gobierno Electrónico del Estado Uruguayo», 2011. [En línea]. Disponible en: http://www.agesic.gub.uy/innovaportal/file/1295/1/descripcion_y_guias_de_uso_de_la_plataforma_de_gobierno_electronico_del_estado_uruguayo.pdf.
- [31] El Senado y la Cámara de Representantes de la República Oriental del Uruguay, *Ley de Protección de Datos Personales y Acción «Habeas Data»*. .
- [32] «History of Microsoft Corporation – FundingUniverse». [En línea]. Disponible en: <http://www.fundinguniverse.com/company-histories/microsoft-corporation-history/>. [Accedido: 06-mar-2016].
- [33] «Biblioteca de Microsoft SQL Server». [En línea]. Disponible en: <https://msdn.microsoft.com/es-es/library/bb545450.aspx>. [Accedido: 06-mar-2016].
- [34] «Master Data Services». [En línea]. Disponible en: [https://msdn.microsoft.com/es-es/library/ee633763\(v=sql.120\).aspx](https://msdn.microsoft.com/es-es/library/ee633763(v=sql.120).aspx). [Accedido: 06-mar-2016].

- [35] «IBM - InfoSphere Master Data Management». [En línea]. Disponible en: <http://www-03.ibm.com/software/products/en/infosphere-master-data-management>. [Accedido: 19-mar-2016].
- [36] «IBM - InfoSphere Master Data Management Standard Edition». [En línea]. Disponible en: <http://www-03.ibm.com/software/products/es/infomastdatamanastanedit>. [Accedido: 06-mar-2016].
- [37] «IBM - InfoSphere Master Data Management Advanced Edition». [En línea]. Disponible en: <http://www-03.ibm.com/software/products/es/infomastdatamanaadvaedit>. [Accedido: 06-mar-2016].
- [38] Paul Mickelson, «Demystifying IBM InfoSphere MDM Products: Names and Product Bundles | Prolifics». [En línea]. Disponible en: <http://www.prolifics.com/blog/demystifying-ibm-infosphere-mdm-products-names-and-product-bundles>. [Accedido: 06-mar-2016].
- [39] «IBM - InfoSphere Master Data Management Enterprise Edition». [En línea]. Disponible en: <http://www-03.ibm.com/software/products/en/infomastdatamanaaenteedit>. [Accedido: 06-mar-2016].
- [40] «IBM Software IBM Rational - España». [En línea]. Disponible en: <http://www-01.ibm.com/software/es/rational/>. [Accedido: 06-mar-2016].
- [41] «IBM - WebSphere Application Server». [En línea]. Disponible en: <http://www-03.ibm.com/software/products/es/appserv-was>. [Accedido: 06-mar-2016].
- [42] «What Is Talend: Talend Software Overview & About Talend». [En línea]. Disponible en: <http://www.talend.com/about-us>. [Accedido: 06-mar-2016].
- [43] «Master Data Management - Talend». [En línea]. Disponible en: <https://www.talend.com/resource/master-data-management.html>. [Accedido: 06-mar-2016].
- [44] «Eclipse - The Eclipse Foundation open source community website.» [En línea]. Disponible en: <https://eclipse.org/>. [Accedido: 06-mar-2016].
- [45] «ETL Tools - Talend». [En línea]. Disponible en: <http://www.talend.com/resource/etl-tools.html>. [Accedido: 06-mar-2016].
- [46] «About | WSO2 Inc». [En línea]. Disponible en: <http://wso2.com/about/>. [Accedido: 06-mar-2016].
- [47] «“Balana” The Open source XACML 3.0 implementation – SOA Security». [En línea]. Disponible en: <http://xacmlinfo.org/2012/08/16/balana-the-open-source-xacml-3-0-implementation/>. [Accedido: 06-mar-2016].
- [48] «About Identity Server - Identity Server 5.1.0 - WSO2 Documentation». [En línea]. Disponible en: <https://docs.wso2.com/display/IS510/About+Identity+Server>. [Accedido: 06-mar-2016].
- [49] «Identity Server Features». [En línea]. Disponible en: <https://docs.wso2.com/display/IS510/Identity+Server+Features>.
- [50] «Creating a XACML Policy». [En línea]. Disponible en: <https://docs.wso2.com/display/IS510/Creating+a+XACML+Policy>.
- [51] «Publishing a XACML Policy». [En línea]. Disponible en: <https://docs.wso2.com/display/IS510/Publishing+a+XACML+Policy>.
- [52] «Enabling and Disabling a XACML Policy». [En línea]. Disponible en: <https://docs.wso2.com/display/IS510/Enabling+and+Disabling+a+XACML+Policy>.

- [53] «About | MuleSoft». [En línea]. Disponible en: <https://www.mulesoft.com/about>. [Accedido: 06-mar-2016].
- [54] «What is Mule ESB? | MuleSoft». [En línea]. Disponible en: <https://www.mulesoft.com/resources/esb/what-mule-esb>. [Accedido: 06-mar-2016].
- [55] «Mule Concepts // MuleSoft Documentation». [En línea]. Disponible en: <https://docs.mulesoft.com/mule-fundamentals/v/3.7/mule-concepts>. [Accedido: 06-mar-2016].
- [56] «Elements in a Mule Flow // MuleSoft Documentation». [En línea]. Disponible en: <https://docs.mulesoft.com/mule-fundamentals/v/3.7/elements-in-a-mule-flow>. [Accedido: 06-mar-2016].
- [57] «Apache Tomcat - Welcome!» [En línea]. Disponible en: <http://tomcat.apache.org/>. [Accedido: 06-mar-2016].
- [58] «SoapUI | Functional Testing for SOAP and REST APIs». [En línea]. Disponible en: <https://www.soapui.org/>. [Accedido: 06-mar-2016].
- [59] «MySQL». [En línea]. Disponible en: <https://www.mysql.com/>. [Accedido: 06-mar-2016].
- [60] «PDP PEP Communication – how WSO2 Identity Server defines». [En línea]. Disponible en: <http://xacmlinfo.org/2012/06/13/pdp-pep-communication-wso2is/>.
- [61] «PEP client for WSO2 Identity Server PDP». [En línea]. Disponible en: <http://xacmlinfo.org/2012/06/14/pep-client-for-wso2is-pdp/>.
- [62] AGESIC (Agencia para el Desarrollo del Gobierno de Gestión Electrónica y la Sociedad de la Información y el Conocimiento), «Metadatos Modelo de Referencia Persona - Diccionario de Datos». 08-feb-2013.
- [63] «Cómo seleccionar la edición correcta de IBM InfoSphere MDM para un proyecto de Integración de Datos de Cliente». [En línea]. Disponible en: <http://www.ibm.com/developerworks/ssa/library/infosphere-mdm/>. [Accedido: 07-feb-2016].
- [64] «Master Data Management Software for Customers & Products». [En línea]. Disponible en: <https://www.talend.com/products/mdm>. [Accedido: 06-mar-2016].
- [65] «About Identity Server». [En línea]. Disponible en: <https://docs.wso2.com/display/IS510/About+Identity+Server>.

Apéndice 1. Políticas de ejemplo utilizadas

En este apéndice se presentan las políticas utilizadas en el caso de estudio.

En primer lugar, la política que admite el acceso a todos los atributos públicos siempre que el propósito incluido en la solicitud sea *MDM_BusinessOriented* como se puede ver en las figuras Figura 65, Figura 66, Figura 67 y Figura 68.

```
<Policy xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
  PolicyId="P_PublicAttributes"
  RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:first-applicable"
  Version="1.0">
  <Target>
    <AnyOf>
      <AllOf>
        <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          <AttributeValue
            DataType="http://www.w3.org/2001/XMLSchema#string">MDM_BusinessOriented
          </AttributeValue>
          <AttributeDesignator
            AttributeId="urn:oasis:names:tc:xacml:1.0:purpose:purpose-id"
            Category="urn:oasis:names:tc:xacml:3.0:attribute-category:purpose"
            DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true">
          </AttributeDesignator>
        </Match>
        <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
            EstructuraPersona.Persona_Nacionalidad
          </AttributeValue>
          <AttributeDesignator
            AttributeId="urn:oasis:names:tc:xacml:1.0:resource:Target"
            Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
            DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true">
          </AttributeDesignator>
        </Match>
      </AllOf>
    </AnyOf>
  </Target>
  <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
    <AttributeValue
      DataType="http://www.w3.org/2001/XMLSchema#string">MDM_BusinessOriented
    </AttributeValue>
    <AttributeDesignator
      AttributeId="urn:oasis:names:tc:xacml:1.0:purpose:purpose-id"
      Category="urn:oasis:names:tc:xacml:3.0:attribute-category:purpose"
      DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true">
    </AttributeDesignator>
  </Match>
  <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
    <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
      EstructuraPersona.NombrePersona_NombrePersonaId
    </AttributeValue>
    <AttributeDesignator
      AttributeId="urn:oasis:names:tc:xacml:1.0:resource:Target"
      Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
      DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true">
    </AttributeDesignator>
  </Match>
</Policy>
```

Figura 65 - Política P_PublicAttributes (1/4)

```

<AllOf>
  <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
    <AttributeValue
      DataType="http://www.w3.org/2001/XMLSchema#string">MDM_BusinessOriented
    </AttributeValue>
    <AttributeDesignator
      AttributeId="urn:oasis:names:tc:xacml:1.0:purpose:purpose-id"
      Category="urn:oasis:names:tc:xacml:3.0:attribute-category:purpose"
      DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true">
    </AttributeDesignator>
  </Match>
  <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
    <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
      EstructuraPersona.NombrePersona_NombreCompleto
    </AttributeValue>
    <AttributeDesignator
      AttributeId="urn:oasis:names:tc:xacml:1.0:resource:Target"
      Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
      DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true">
    </AttributeDesignator>
  </Match>
</AllOf>
<AllOf>
  <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
    <AttributeValue
      DataType="http://www.w3.org/2001/XMLSchema#string">MDM_BusinessOriented
    </AttributeValue>
    <AttributeDesignator
      AttributeId="urn:oasis:names:tc:xacml:1.0:purpose:purpose-id"
      Category="urn:oasis:names:tc:xacml:3.0:attribute-category:purpose"
      DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true">
    </AttributeDesignator>
  </Match>
  <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
    <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
      EstructuraPersona.NombrePersona_NombreCorto
    </AttributeValue>
    <AttributeDesignator
      AttributeId="urn:oasis:names:tc:xacml:1.0:resource:Target"
      Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
      DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true">
    </AttributeDesignator>
  </Match>
</AllOf>
<AllOf>
  <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
    <AttributeValue
      DataType="http://www.w3.org/2001/XMLSchema#string">MDM_BusinessOriented
    </AttributeValue>
    <AttributeDesignator
      AttributeId="urn:oasis:names:tc:xacml:1.0:purpose:purpose-id"
      Category="urn:oasis:names:tc:xacml:3.0:attribute-category:purpose"
      DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true">
    </AttributeDesignator>
  </Match>
  <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
    <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
      EstructuraPersona.NombrePersona_TerminoCortesia
    </AttributeValue>
    <AttributeDesignator
      AttributeId="urn:oasis:names:tc:xacml:1.0:resource:Target"
      Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
      DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true">
    </AttributeDesignator>
  </Match>
</AllOf>

```

Figura 66 - Política P_PublicAttributes (2/4)

```

<AllOf>
  <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
    <AttributeValue
      DataType="http://www.w3.org/2001/XMLSchema#string">MDM_BusinessOriented
    </AttributeValue>
    <AttributeDesignator
      AttributeId="urn:oasis:names:tc:xacml:1.0:purpose:purpose-id"
      Category="urn:oasis:names:tc:xacml:3.0:attribute-category:purpose"
      DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true">
    </AttributeDesignator>
  </Match>
  <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
    <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
      EstructuraPersona.NombrePersona_NombrePreferido
    </AttributeValue>
    <AttributeDesignator
      AttributeId="urn:oasis:names:tc:xacml:1.0:resource:Target"
      Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
      DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true">
    </AttributeDesignator>
  </Match>
</AllOf>
<AllOf>
  <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
    <AttributeValue
      DataType="http://www.w3.org/2001/XMLSchema#string">MDM_BusinessOriented
    </AttributeValue>
    <AttributeDesignator
      AttributeId="urn:oasis:names:tc:xacml:1.0:purpose:purpose-id"
      Category="urn:oasis:names:tc:xacml:3.0:attribute-category:purpose"
      DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true">
    </AttributeDesignator>
  </Match>
  <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
    <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
      EstructuraPersona.DocumentoPersona_DocumentoPersonaId
    </AttributeValue>
    <AttributeDesignator
      AttributeId="urn:oasis:names:tc:xacml:1.0:resource:Target"
      Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
      DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true">
    </AttributeDesignator>
  </Match>
</AllOf>
<AllOf>
  <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
    <AttributeValue
      DataType="http://www.w3.org/2001/XMLSchema#string">MDM_BusinessOriented
    </AttributeValue>
    <AttributeDesignator
      AttributeId="urn:oasis:names:tc:xacml:1.0:purpose:purpose-id"
      Category="urn:oasis:names:tc:xacml:3.0:attribute-category:purpose"
      DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true">
    </AttributeDesignator>
  </Match>
  <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
    <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
      EstructuraPersona.DocumentoPersona_IdPaisAlfa2
    </AttributeValue>
    <AttributeDesignator
      AttributeId="urn:oasis:names:tc:xacml:1.0:resource:Target"
      Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
      DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true">
    </AttributeDesignator>
  </Match>
</AllOf>

```

Figura 67 - Política P_PublicAttributes (3/4)

```

<AllOf>
  <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
    <AttributeValue
      DataType="http://www.w3.org/2001/XMLSchema#string">MDM_BusinessOriented
    </AttributeValue>
    <AttributeDesignator
      AttributeId="urn:oasis:names:tc:xacml:1.0:purpose:purpose-id"
      Category="urn:oasis:names:tc:xacml:3.0:attribute-category:purpose"
      DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true">
    </AttributeDesignator>
  </Match>
  <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
    <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
      EstructuraPersona.DocumentoPersona_IdPaisAlfa3
    </AttributeValue>
    <AttributeDesignator
      AttributeId="urn:oasis:names:tc:xacml:1.0:resource:Target"
      Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
      DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true">
    </AttributeDesignator>
  </Match>
</AllOf>
<AllOf>
  <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
    <AttributeValue
      DataType="http://www.w3.org/2001/XMLSchema#string">MDM_BusinessOriented
    </AttributeValue>
    <AttributeDesignator
      AttributeId="urn:oasis:names:tc:xacml:1.0:purpose:purpose-id"
      Category="urn:oasis:names:tc:xacml:3.0:attribute-category:purpose"
      DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true">
    </AttributeDesignator>
  </Match>
  <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
    <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
      EstructuraPersona.DocumentoPersona_TipoDocumento
    </AttributeValue>
    <AttributeDesignator
      AttributeId="urn:oasis:names:tc:xacml:1.0:resource:Target"
      Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
      DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true">
    </AttributeDesignator>
  </Match>
</AllOf>
<AllOf>
  <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
    <AttributeValue
      DataType="http://www.w3.org/2001/XMLSchema#string">MDM_BusinessOriented
    </AttributeValue>
    <AttributeDesignator
      AttributeId="urn:oasis:names:tc:xacml:1.0:purpose:purpose-id"
      Category="urn:oasis:names:tc:xacml:3.0:attribute-category:purpose"
      DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true">
    </AttributeDesignator>
  </Match>
  <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
    <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
      EstructuraPersona.DocumentoPersona_NúmeroDocumento
    </AttributeValue>
    <AttributeDesignator
      AttributeId="urn:oasis:names:tc:xacml:1.0:resource:Target"
      Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
      DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true">
    </AttributeDesignator>
  </Match>
</AllOf>
</AnyOf>
</Target>
<Rule Effect="Permit" RuleId="dd977ded-2d41-4b44-95d4-e900d27015e2"></Rule>
</Policy>

```

Figura 68 - Política P_PublicAttributes (4/4)

Por otro lado, la política que habilita a AGESIC a ver el atributo sensible *TelNumero* se puede ver en la Figura 69.

```
<Policy xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
  PolicyId="P_12345678_Select_Business"
  RuleCombiningAlgId="urn:oasis:names:tc:xacml:3.0:rule-combining-algorithm:deny-unless-permit"
  Version="1.0">
  <Target>
    <AnyOf>
      <AllOf>
        <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          <AttributeValue
            DataType="http://www.w3.org/2001/XMLSchema#string">Select_for_use
          </AttributeValue>
          <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
            Category="urn:oasis:names:tc:xacml:3.0:attribute-category:action"
            DataType="http://www.w3.org/2001/XMLSchema#string"
            MustBePresent="true"/>
        </Match>
        <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
            MDM_BusinessOriented</AttributeValue>
          <AttributeDesignator
            AttributeId="urn:oasis:names:tc:xacml:1.0:purpose:purpose-id"
            Category="urn:oasis:names:tc:xacml:3.0:attribute-category:purpose"
            DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true">
          </AttributeDesignator>
        </Match>
      </AllOf>
    </AnyOf>
  </Target>
  <Rule Effect="Permit" RuleId="Rule_DGI">...</Rule>
  <Rule Effect="Permit" RuleId="Rule_AGESIC">
    <Target>
      <AnyOf>
        <AllOf>
          <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
              DataType="http://www.w3.org/2001/XMLSchema#string">AGESIC
            </AttributeValue>
            <AttributeDesignator
              AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
              Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
              DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true">
            </AttributeDesignator>
          </Match>
        </AllOf>
      </AnyOf>
    </Target>
    <Condition>
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:or">
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-is-in">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
            EstructuraPersona.Persona_TelNumero
          </AttributeValue>
          <AttributeDesignator
            AttributeId="urn:oasis:names:tc:xacml:1.0:resource:Target"
            Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
            DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true">
          </AttributeDesignator>
        </Apply>
      </Apply>
    </Condition>
  </Rule>
</Policy>
```

Figura 69 - Política P_12345678_Select_Business

Luego, la política que habilita a DNIC a compartir el dato *TelNumero* para el Propósito *MDM_BusinessOriented* y la Acción *Select_for_share* se puede ver en la Figura 70.

```
<Policy xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
  PolicyId="P_12345678_SelectShare_Business"
  RuleCombiningAlgId="urn:oasis:names:tc:xacml:3.0:rule-combining-algorithm:deny-unless-permit"
  Version="1.0">
  <Target>
    <AnyOf>
      <AllOf>
        <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
            Select_for_share
          </AttributeValue>
          <AttributeDesignator
            AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
            Category="urn:oasis:names:tc:xacml:3.0:attribute-category:action"
            DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true">
          </AttributeDesignator>
        </Match>
        <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
            MDM_BusinessOriented
          </AttributeValue>
          <AttributeDesignator
            AttributeId="urn:oasis:names:tc:xacml:1.0:purpose:purpose-id"
            Category="urn:oasis:names:tc:xacml:3.0:attribute-category:purpose"
            DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true">
          </AttributeDesignator>
        </Match>
      </AllOf>
    </AnyOf>
  </Target>
  <Rule Effect="Permit" RuleId="Rule_DNIC">
    <Target>
      <AnyOf>
        <AllOf>
          <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
              DNIC
            </AttributeValue>
            <AttributeDesignator
              AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
              Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
              DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true">
            </AttributeDesignator>
          </Match>
        </AllOf>
      </AnyOf>
    </Target>
    <Condition>
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:or">
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-is-in">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
            EstructuraPersona.Persona_TelNumero
          </AttributeValue>
          <AttributeDesignator
            AttributeId="urn:oasis:names:tc:xacml:1.0:resource:Target"
            Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
            DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true">
          </AttributeDesignator>
        </Apply>
      </Apply>
    </Condition>
  </Rule>
</Policy>
```

Figura 70 - Política P_12345678_SelectShare_Business

Finalmente, el Policy Set del ciudadano 12345678 que contiene la referencia a las políticas determinadas por sus consentimientos, se puede ver en la Figura 71.

```

<PolicySet
  xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
  PolicyCombiningAlgId="urn:oasis:names:tc:xacml:3.0:policy-combining-algorithm:deny-unless-permit"
  PolicySetId="PS_CII12345678" Version="1.0">
  <Target>
    <AnyOf>
      <AllOf>
        <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-regexp-match">
          <AttributeValue
            DataType="http://www.w3.org/2001/XMLSchema#string">12345678
          </AttributeValue>
          <AttributeDesignator
            AttributeId="urn:oasis:names:tc:xacml:1.0:resource:NumeroDocumento"
            Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
            DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true">
          </AttributeDesignator>
        </Match>
        <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-regexp-match">
          <AttributeValue
            DataType="http://www.w3.org/2001/XMLSchema#string">CI
          </AttributeValue>
          <AttributeDesignator
            AttributeId="urn:oasis:names:tc:xacml:1.0:resource:TipoDocumento"
            Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
            DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true">
          </AttributeDesignator>
        </Match>
      </AllOf>
    </AnyOf>
  </Target>
  <PolicyIdReference>P_12346578_Select_Business</PolicyIdReference>
  <PolicyIdReference>P_12346578_SelectShare_Business</PolicyIdReference>
  <PolicyIdReference>P_12345678_Update_Maintain</PolicyIdReference>
</PolicySet>

```

Figura 71 - Conjunto de Políticas PS_12345678

Apéndice 2. Mensajes XACML

En este apéndice se muestra un ejemplo de solicitud y respuesta XACML.

En primer lugar, se detalla la solicitud XACML. Esta incluye el Propósito *MDM_BusinessOriented* y Acción *Select_for_use* para los cuales el Sujeto *DGI* solicita los atributos *Género* y *Sexo* del Recurso identificado por el tipo de documento *CI* y número de documento *12345678* como se puede ver en la Figura 72.

```

<Request CombinedDecision="false" ReturnPolicyIdList="false">
  <Attributes
    Category="urn:oasis:names:tc:xacml:3.0:attribute-category:action">
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
      IncludeInResult="false">
      <AttributeValue
        DataType="http://www.w3.org/2001/XMLSchema#string">Select_for_use
      </AttributeValue>
    </Attribute>
  </Attributes>
  <Attributes
    Category="urn:oasis:names:tc:xacml:3.0:attribute-category:purpose">
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:purpose:purpose-id"
      IncludeInResult="false">
      <AttributeValue
        DataType="http://www.w3.org/2001/XMLSchema#string">MDM_BusinessOriented
      </AttributeValue>
    </Attribute>
  </Attributes>
  <Attributes
    Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject">
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
      IncludeInResult="false">
      <AttributeValue
        DataType="http://www.w3.org/2001/XMLSchema#string">DGI
      </AttributeValue>
    </Attribute>
  </Attributes>
  <Attributes
    Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource">
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:NumeroDocumento"
      IncludeInResult="true">
      <AttributeValue
        DataType="http://www.w3.org/2001/XMLSchema#string">12346578
      </AttributeValue>
    </Attribute>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:TipoDocumento"
      IncludeInResult="true">
      <AttributeValue
        DataType="http://www.w3.org/2001/XMLSchema#string">CI
      </AttributeValue>
    </Attribute>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:Target"
      IncludeInResult="true">
      <AttributeValue
        DataType="http://www.w3.org/2001/XMLSchema#string">EstructuraPersona.Persona_Genero
      </AttributeValue>
    </Attribute>
  </Attributes>
  <Attributes
    Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource">
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:NumeroDocumento"
      IncludeInResult="true">
      <AttributeValue
        DataType="http://www.w3.org/2001/XMLSchema#string">12346578
      </AttributeValue>
    </Attribute>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:TipoDocumento"
      IncludeInResult="true">
      <AttributeValue
        DataType="http://www.w3.org/2001/XMLSchema#string">CI
      </AttributeValue>
    </Attribute>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:Target"
      IncludeInResult="true">
      <AttributeValue
        DataType="http://www.w3.org/2001/XMLSchema#string">EstructuraPersona.Persona_Sexo
      </AttributeValue>
    </Attribute>
  </Attributes>
  <Attributes
    Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource">
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:NumeroDocumento"
      IncludeInResult="true">
      <AttributeValue
        DataType="http://www.w3.org/2001/XMLSchema#string">12346578
      </AttributeValue>
    </Attribute>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:TipoDocumento"
      IncludeInResult="true">
      <AttributeValue
        DataType="http://www.w3.org/2001/XMLSchema#string">CI
      </AttributeValue>
    </Attribute>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:Target"
      IncludeInResult="true">
      <AttributeValue
        DataType="http://www.w3.org/2001/XMLSchema#string">EstructuraPersona.Persona_Ascendencia
      </AttributeValue>
    </Attribute>
  </Attributes>
</Request>

```

Figura 72 - Ejemplo solicitud XACML

Luego, se detalla la respuesta XACML retornada. Esta incluye la decisión que se determina para cada atributo solicitado como se puede ver en la Figura 73.

```

<Result>
  <Decision>Permit</Decision>
  <Status>
    <StatusCode Value="urn:oasis:names:tc:xacml:1.0:status:ok"/>
  </Status>
  <Attributes
    Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource">
    <Attribute
      AttributeId="urn:oasis:names:tc:xacml:1.0:resource:TipoDocumento"
      IncludeInResult="true">
    <AttributeValue
      DataType="http://www.w3.org/2001/XMLSchema#string">CI
    </AttributeValue>
    </Attribute>
    <Attribute
      AttributeId="urn:oasis:names:tc:xacml:1.0:resource:NumeroDocumento"
      IncludeInResult="true">
    <AttributeValue
      DataType="http://www.w3.org/2001/XMLSchema#string">12345678
    </AttributeValue>
    </Attribute>
    <Attribute
      AttributeId="urn:oasis:names:tc:xacml:1.0:resource:Target"
      IncludeInResult="true">
    <AttributeValue
      DataType="http://www.w3.org/2001/XMLSchema#string">EstructuraPersona.Persona_Genero
    </AttributeValue>
    </Attribute>
    </Attributes>
  </Result>
<Result>
  <Decision>Permit</Decision>
  <Status>
    <StatusCode Value="urn:oasis:names:tc:xacml:1.0:status:ok"/>
  </Status>
  <Attributes
    Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource">
    <Attribute
      AttributeId="urn:oasis:names:tc:xacml:1.0:resource:TipoDocumento"
      IncludeInResult="true">
    <AttributeValue
      DataType="http://www.w3.org/2001/XMLSchema#string">CI
    </AttributeValue>
    </Attribute>
    <Attribute
      AttributeId="urn:oasis:names:tc:xacml:1.0:resource:NumeroDocumento"
      IncludeInResult="true">
    <AttributeValue
      DataType="http://www.w3.org/2001/XMLSchema#string">12345678
    </AttributeValue>
    </Attribute>
    <Attribute
      AttributeId="urn:oasis:names:tc:xacml:1.0:resource:Target"
      IncludeInResult="true">
    <AttributeValue
      DataType="http://www.w3.org/2001/XMLSchema#string">EstructuraPersona.Ascendencia
    </AttributeValue>
    </Attribute>
    </Attributes>
  </Result>
<Result>
  <Decision>Deny</Decision>
  <Status>
    <StatusCode Value="urn:oasis:names:tc:xacml:1.0:status:ok"/>
  </Status>
  <Attributes
    Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource">
    <Attribute
      AttributeId="urn:oasis:names:tc:xacml:1.0:resource:TipoDocumento"
      IncludeInResult="true">
    <AttributeValue
      DataType="http://www.w3.org/2001/XMLSchema#string">CI
    </AttributeValue>
    </Attribute>
    <Attribute
      AttributeId="urn:oasis:names:tc:xacml:1.0:resource:NumeroDocumento"
      IncludeInResult="true">
    <AttributeValue
      DataType="http://www.w3.org/2001/XMLSchema#string">12345678
    </AttributeValue>
    </Attribute>
    <Attribute
      AttributeId="urn:oasis:names:tc:xacml:1.0:resource:Target"
      IncludeInResult="true">
    <AttributeValue
      DataType="http://www.w3.org/2001/XMLSchema#string">EstructuraPersona.Sexo
    </AttributeValue>
    </Attribute>
    </Attributes>
  </Result>

```

Figura 73 - Ejemplo respuesta XACML

Apéndice 3. Glosario

ABAC	Attribute Based Access Control (Control de acceso basado en atributos)
ESB	Enterprise Service Bus
HTTP	HyperText Transfer Protocol
JDBC	Java DataBase Connectivity
JMS	Java Message System
MDC	Modelo de Datos Canónico
MDM	Master Data Management
PAP	Policy Administration Point
PDP	Policy Decision Point
PEP	Policy Enforcement Point
PIP	Policy Information Point
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
XACML	eXtensible Access Control Markup Language
XML	eXtensible Markup Language
XSLT	eXtensible Stylesheet Language Transformations
WSDL	Web Service Definition Language