



# Herramienta para la Evaluación de Calidad de Servicios Web Geográficos en Infraestructuras de Datos Espaciales

Informe final  
Proyecto de Grado

**Autores:**

Ramiro Sanchez  
Natalia Calle

**Tutores:**

MSc. Ing. Laura González  
MSc. Ing. Raquel Sosa

Diciembre 2017

Informe de Proyecto de Grado presentado al Tribunal Evaluador como requisito de graduación de la carrera Ingeniería en Computación. Montevideo, Uruguay, 2017



## Resumen

Actualmente los servicios web son muy utilizados con el propósito de intercambiar datos a través de la web y ofrecer servicios como procedimientos remotos. En particular, los servicios web geográficos son utilizados para el acceso a información geográfica, resultando de interés para sistemas o aplicaciones que deban localizar el lugar en donde se halla un elemento.

Los servicios web geográficos son particularmente importantes para las infraestructuras de datos espaciales (IDE). Las IDEs son plataformas que permiten el uso y publicación de información geográfica, y tienen como objetivo facilitar la disponibilidad y acceso a los datos geográficos, logrando mayor eficiencia en la gestión de los mismos. La información publicada en las IDEs debe poder ser accedida en forma adecuada, por lo que resulta de vital importancia establecer medidas de calidad que permitan evaluar cuán aceptable o confiable es un determinado servicio.

Este proyecto propone una solución para evaluar la calidad de servicios web geográficos en el contexto de las IDEs. Para la evaluación de los servicios, se utiliza un modelo de calidad existente organizado en dimensiones, factores y métricas de calidad. Además la solución permite extender el modelo de calidad con nuevas dimensiones, factores y métricas.

La solución propuesta fue implementada utilizando el lenguaje de programación Java, con Bootstrap como framework para el desarrollo web, PostgreSQL como sistema de base de datos, Wildfly como servidor de aplicaciones, JSF para el desarrollo de interfaces de usuario y PrimeFaces por sus componentes visuales. Por último, se desarrolló un caso de estudio en el marco de la IDEuy: la infraestructura de datos espaciales del Uruguay.

**Palabras claves:** Servicios web geográficos, IDEuy, modelo de calidad, OGC, QoS.



## Tabla de contenidos

<b>Resumen</b>	<b>3</b>
<b>1. Introducción</b>	<b>7</b>
Objetivos	7
Resultados y aportes del proyecto	8
Organización del informe	9
<b>2. Marco Conceptual</b>	<b>11</b>
Datos espaciales	11
Servicios web	13
OGC Web Services (OWS)	14
Arquitectura y tecnologías GIS	20
Infraestructura de datos espaciales	22
IDEuy	24
Calidad de Servicio	25
Modelo de calidad de servicio para IDEuy	26
Introducción	26
Objetos medibles del modelo	28
Dimensiones y factores del modelo	29
Métricas del modelo	30
Perfiles de evaluación	32
<b>3. Análisis</b>	<b>35</b>
Marco de trabajo	35
Análisis de requerimientos	36
Relevamiento de soluciones existentes	39
Resumen	45
<b>4. Planteo de la Solución</b>	<b>47</b>
Descripción general	47
Arquitectura	48
Métodos de medición	54
Extensibilidad de modelo	56
Decisiones tomadas	57
<b>5. Implementación de la Solución</b>	<b>59</b>
Herramientas utilizadas	59
Carga dinámica de métodos de evaluación	62
Procesamiento de esquemas XML	63
Componentes de interfaz de usuario	64
Publicación del sistema	66
Problemas encontrados	66

<b>6. Caso de Estudio y Pruebas</b>	<b>69</b>
Caso de estudio	69
Pruebas realizadas	74
<b>7. Conclusiones y Trabajo a Futuro</b>	<b>75</b>
Gestión de proyecto	75
Conclusiones generales	77
Trabajo a futuro	78
<b>Referencias</b>	<b>81</b>

# 1. Introducción

Actualmente los servicios web son muy utilizados con el propósito de intercambiar datos a través de la web y ofrecer servicios como procedimientos remotos, entre distintas aplicaciones sin depender de la plataforma ni del lenguaje de programación empleado [1]. En particular, los servicios web geográficos son utilizados para el acceso a información geográfica, resultando de interés para sistemas o aplicaciones que deban localizar el lugar en donde se halla un elemento. Ejemplos de datos geográficos son la ubicación y extensión de las distintas carreteras, la distribución de ríos y cursos de agua situados en un territorio o los límites fronterizos que conforman los países de la región. [2]

Los servicios web geográficos son particularmente importantes para las infraestructuras de datos espaciales (IDE). Las IDEs son plataformas que permiten el uso y publicación de información geográfica, y tienen como objetivo facilitar la disponibilidad y acceso a los datos geográficos, logrando mayor eficiencia en la gestión de los mismos.

La información publicada en las IDEs debe poder ser accedida en forma adecuada, y debido a que es publicada a través de servicios web geográficos, resulta de vital importancia establecer medidas de calidad que permitan evaluar cuán aceptable o confiable es un determinado servicio.

Para abordar esta problemática, en la Tesis de Maestría en Sistemas de Información y Tecnologías de Gestión de Datos por el Lic. De Los Reyes [3] se propuso un modelo para evaluar la calidad de los servicios web geográficos. Si bien en el marco de la tesis se implementó un prototipo que evalúa algunas de las medidas de calidad pertenecientes a determinados factores, no se desarrolló una solución integral que permita ponerlo en práctica y obtener indicadores de calidad sobre las infraestructuras de datos espaciales.

## 1.1. Objetivos

El objetivo general de este proyecto es diseñar e implementar una herramienta para la evaluación de la calidad de servicios web geográficos en el contexto de infraestructuras de datos espaciales tomando como referencia la Tesis del Lic. De Los Reyes [3].

Para alcanzar este objetivo se proponen los siguientes objetivos específicos:

- Estudiar el marco teórico y los conceptos relevantes para el proyecto, en particular el funcionamiento de los servicios web geográficos, las infraestructuras de datos espaciales, la calidad de servicio y el modelo de calidad desarrollado en la Tesis del Lic. De Los Reyes [3].

- Analizar los requerimientos del sistema a construir, describiendo su comportamiento general, las funcionalidades específicas y las soluciones existentes en el área de calidad de servicios web geográficos.
- Proponer y diseñar una solución que permita cumplir con los requisitos planteados.
- Implementar un prototipo que permita validar la factibilidad técnica de la solución.
- Desarrollar un caso de estudio en el marco de la realidad uruguaya que permita validar funcionalmente la solución.

## 1.2. Resultados y aportes del proyecto

Los principales aportes consisten en el diseño e implementación de una solución para evaluar la calidad de servicios web geográficos. A continuación se describen los resultados y aportes del proyecto:

- Análisis de requerimientos del sistema y relevamiento de soluciones existentes en el área de calidad de servicios geográficos. Los requerimientos funcionales se agrupan en las siguientes áreas: autenticación y gestión de usuarios, gestión de objetos medibles, gestión de perfiles de evaluación, gestión de modelos de calidad, evaluaciones y gestión de reportes. Mientras que los requerimientos no funcionales se centran en extensibilidad del modelo, desempeño, seguridad y mantenibilidad.
- Propuesta y diseño de una solución que cumple con los requisitos planteados. Se plantea una arquitectura en tres capas y se diseñan los principales componentes, teniendo en consideración la extensibilidad del modelo de calidad y los métodos de medición.
- Implementación de un prototipo que permite validar la factibilidad técnica de la solución. El prototipo fue implementado en Java, utilizando Wildfly como servidor de aplicaciones y PostgreSQL como motor de base de datos.
- Desarrollo de un caso de estudio. La herramienta fue utilizada exitosamente en un caso de estudio específico, el cual consistió en evaluar una IDE en particular: la infraestructura de datos espaciales del Uruguay (IDEuy).<sup>1</sup>

---

<sup>1</sup> <http://ide.uy/>

### 1.3. Organización del informe

A continuación se describe la organización del resto del documento:

En el Capítulo 2 se describe el marco conceptual del proyecto, detallando los conceptos teóricos, así como la infraestructura de datos espaciales del Uruguay y el modelo de calidad utilizado.

En el Capítulo 3 se introduce el contexto de trabajo, se detallan los principales requerimientos del sistema y se estudian las soluciones existentes.

En el Capítulo 4 se describe la solución propuesta junto a la arquitectura elegida, los métodos de medición, la extesibilidad del modelo y las decisiones tomadas a nivel de diseño.

El Capítulo 5 describe detalles acerca de la implementación de la solución. Además se presentan las herramientas utilizadas en la realización de la herramienta.

El Capítulo 6 describe el caso de estudio junto con las pruebas realizadas en las distintas etapas del proyecto.

Finalmente en el Capítulo 7 se exponen las conclusiones del proyecto, se detalla lo relativo a la gestión del proyecto y se evalúan posibles trabajos a futuro.



## 2. Marco Conceptual

En el presente capítulo se presentan y describen conceptos necesarios para la correcta comprensión de este documento, estableciendo un marco conceptual. Se comienza describiendo los conceptos de datos espaciales, servicios web, servicios web geográficos y la arquitectura y tecnología típica que suele haber por detrás de un sistema de información geográfica (GIS). Luego se describe el concepto de la infraestructura de datos espaciales (IDE) y se presenta la infraestructura de datos espaciales del Uruguay (IDEuy), que son conceptos esenciales para entender este documento. Por último, se describen los conceptos de calidad de servicios y de modelo de calidad.

### 2.1. Datos espaciales

Los datos espaciales o también llamados datos geográficos, son un conjunto de valores o elementos que al ser interpretados representan información geográfica. La información geográfica tiene una componente temática y una componente geográfica, que juntas conforman una unidad única de información geográfica. La componente geográfica hace referencia a la posición dentro de un sistema de referencia establecido, obteniéndose la localización junto con sus características geométricas. La componente temática establece la naturaleza del fenómeno localizado por la componente geográfica. Además en función de la componente temática la información geográfica se divide en capas, agrupando entidades geográficas de un mismo tipo que pueden ser combinadas o utilizadas de forma independiente. El concepto de capa es mostrado en la Figura 1 [4].

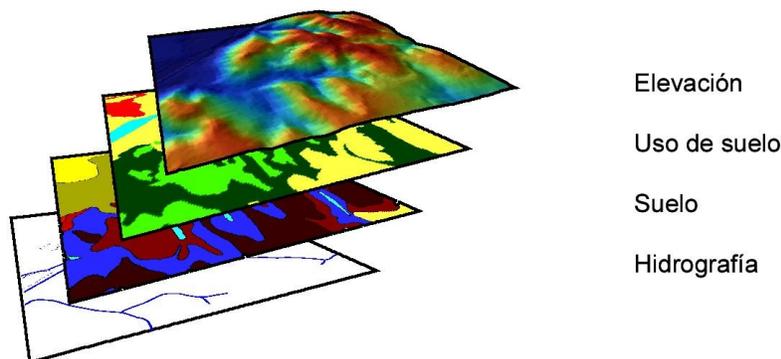


Figura 1: Concepto de capa de información geográfica [4]

Los datos geográficos son representados principalmente mediante el modelo ráster y mediante el modelo vectorial. Los modelos de datos geográficos pueden ser usados para producir mapas, realizar consultas o analizar los datos representados. Comúnmente son empleados en sistemas de información geográfica (GIS), como por ejemplo el sistema de información geográfica que gestiona la Intendencia Municipal de Montevideo<sup>2</sup>.

<sup>2</sup> <http://sig.montevideo.gub.uy/>

### Modelo de representación ráster

En el modelo ráster el espacio se divide sistemáticamente en unidades mínimas llamadas celdas, comúnmente de forma cuadrada y con valores asociados a cada celda. Los valores representan por medio de coloración algún atributo temático como ser temperatura, elevación o vegetación. La unidad mínima de elemento puede tomar formas diversas, siendo la forma cuadrada la más habitual aunque también puede ser rectangular, triangular o hexagonal.

### Modelo de representación vectorial

En el modelo vectorial se almacenan las entidades geográficas a través de las coordenadas de los puntos que la componen, definiendo una serie de elementos geométricos (puntos, líneas y polígonos) con valores asociados.

En la Figura 2 se muestra una comparación esquemática entre el modelo de representación vectorial y el modelo de representación ráster.

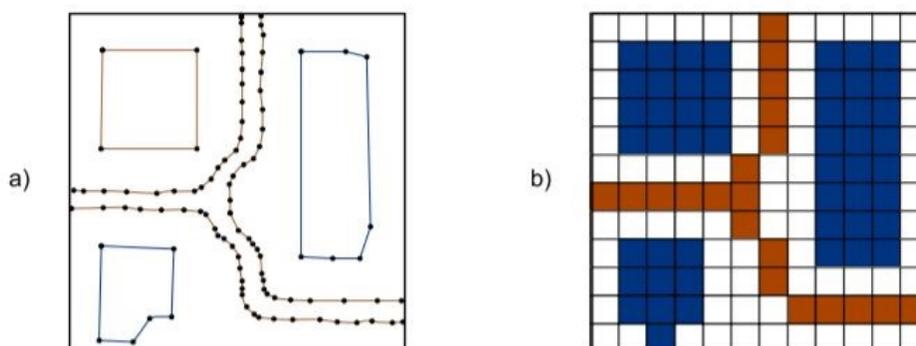


Figura 2: Comparación esquemática entre el modelo vectorial (a) y ráster (b) [4]

### Sistema de coordenadas

Existen distintos sistemas de coordenadas con el fin de poder localizar de forma precisa los puntos sobre la Tierra y en consecuencia los elementos que la conforman. Una de las formas de localización es el sistema de coordenadas geográficas, que es un sistema de coordenadas esféricas mediante el cual un punto se localiza con dos valores angulares llamados latitud y longitud. La latitud es el ángulo entre la línea que une el centro de la esfera (Tierra) con un punto de su superficie y el plano ecuatorial. La longitud es el ángulo formado entre dos de los planos que contienen a la línea de los polos. Las coordenadas geográficas son de gran utilidad al trabajar en grandes regiones [4].

Otra forma de localización es el sistema de coordenadas cartesianas, en el cual un punto se define mediante un par de medidas de distancia  $x$  e  $y$ . Las coordenadas cartesianas son representadas en un plano o superficie bidimensional, a diferencia de las coordenadas geográficas que son representadas en una superficie tridimensional [4].

## Estándares para representación y obtención de información geográfica

El uso de estándares para representar y obtener la información geográfica es vital para garantizar la correcta comunicación entre sistemas. Las organizaciones más relevantes que desarrollan estándares en el campo de la información geográfica son el Open Geospatial Consortium (OGC<sup>3</sup>) y la International Organization for Standardization (ISO<sup>4</sup>). Los principales estándares desarrollados en este ámbito son:

- **Simple Feature Interface Standard (SFS)**

El estándar SFS especifica un modelo para el almacenamiento y acceso a entidades geográficas en bases de datos relacionales y objeto-relacional. Las entidades geográficas que especifica el modelo son representadas mediante elementos vectoriales como lo son puntos, líneas y polígonos [5]. Además el estándar también define una serie de operaciones SQL<sup>5</sup> que operan sobre las entidades geográficas [6]

- **Geography Markup Language (GML)**

GML es un estándar basado en XML [7] que expresa entidades geográficas. GML es usado como un lenguaje de modelado, intercambio y almacenamiento de información geográfica. Como ocurre con la mayoría de las gramáticas basadas en XML, GML tiene dos partes diferenciadas. Una parte es el esquema que describe el documento y otra parte es el documento instanciado que contiene los datos [8].

- **Styled Layer Description (SLD)**

SLD es un estándar que permite la simbolización y coloración de entidades geográficas, capas y mapas. SLD permite extender las capacidades del estándar Web Map Service (WMS) que se explica más adelante [9].

- **Keyhole Markup Language (KML)**

KML es un estándar presentado por la compañía Google y adoptado como un estándar de OGC. Es un lenguaje basado en XML que se centra en las visualizaciones geográficas. Las visualizaciones geográficas incluyen no solo la presentación de datos geográficos sobre la superficie, sino también el control de la navegación del usuarios, en el sentido de a dónde ir y qué mirar [10].

## 2.2. Servicios web

Un servicio web es un módulo de software disponible a través de una red, que completa tareas, resuelve problemas o realiza transacciones solicitadas por un usuario o aplicación. Las aplicaciones que hacen uso de servicios web pueden intercambiar datos sin que importe el lenguaje o plataforma que utilicen. Los servicios web están basados en tecnología existente y en estándares comúnmente usados por la industria, como ser XML y

<sup>3</sup> <http://www.opengeospatial.org/>

<sup>4</sup> <https://www.iso.org/home.html>

<sup>5</sup> Structured Query Language. En español Lenguaje de Consulta Estructurada. Es un lenguaje estándar para almacenar, manipular y obtener datos que se encuentran en base de datos.

HTTP [11], proporcionando un mecanismo universal para la integración de procesos de negocio [2]. Las principales organizaciones abocadas a la elaboración de estándares de servicios en la web son la World Wide Web Consortium (W3C<sup>6</sup>) y la Organization for the Advancement of Structured Information Standards (OASIS<sup>7</sup>).

Existen dos grandes tendencias en el desarrollo de servicios web. Los servicios basados en Simple Object Access Protocol (SOAP) que estandarizan su información, y los servicios basados en Representational State Transfer (REST) que no estandarizan su información y son denominados como servicios RESTful.

### **SOAP**

Es un protocolo para el intercambio de información en un ambiente descentralizado y distribuido. Utiliza lenguaje XML para armado de mensajes y puede ser usado sobre múltiples protocolos, típicamente HTTP. La estructura básica de un mensaje SOAP consiste de un elemento “Envelope” que contiene un elemento opcional “Header” y un elemento requerido “Body” [12].

### **REST**

Es un estilo de arquitectura que se basa en cuatro principios [12]:

- Principio de identificación de recursos mediante Uniform Resource Identifier (URI) [13]. Las URIs proveen de un espacio de direccionamiento global de recursos y descubrimiento de servicios.
- Principio de interfaz uniforme para manipular los recursos mediante las operaciones PUT, GET, POST y DELETE.
  - La operación PUT crea un nuevo recurso.
  - La operación GET obtiene la representación de un recurso.
  - La operación POST modifica o actualiza un recurso.
  - La operación DELETE borra un recurso.
- Principio de mensajes auto descriptivos, donde los recursos están desacoplados de su representación para que su contenido pueda ser accedido en varios formatos como ser HTML, XML, texto plano, PDF, JPEG, entre otros.
- Principio de interacciones sin estado, debido a que no se guarda el estado de las solicitudes y por tanto cada solicitud es independiente.

## **2.3. OGC Web Services (OWS)**

Varios protocolos, especificaciones y estándares son desarrollados para ofrecer información geográfica a través de la web. El Open Geospatial Consortium (OGC) y la International Organization for Standardization (ISO/TC 211) han publicado conjuntamente

---

<sup>6</sup> <https://www.w3.org/>

<sup>7</sup> <https://www.oasis-open.org/>

una serie de especificaciones interoperables de servicios geográficos y codificación de mensajes. Dichas especificaciones conforman estándares que permiten a los desarrolladores construir servicios geográficos confiables y sostenibles. Los servicios web geográficos estandarizados, llamados OGC Web Services (OWS), son definidos por medio de estándares abiertos no propietarios de Internet, particularmente Hypertext Transfer Protocol (HTTP), Uniform Resource Locators (URLs), Multipurpose Internet Mail Extensions (MIME) y Extensible Markup Language (XML) [14].

Entre los estándares agrupados bajo OWS se encuentran el Web Map Service (WMS) [15], el Web Feature Service (WFS) [16], el Web Coverage Service (WCS) [17] y el Catalogue Service for the Web (CSW) [18]. Estos estándares deben de soportar obligatoriamente el método GET. El método POST es obligatorio para WFS y opcional para WMS. Además todos definen la operación GetCapabilities, que se detalla a continuación [19].

### **GetCapabilities** [19]

- La operación permite que el cliente conozca las capacidades del servidor.
- Devuelve la metadata del servicio.
- Permite la operación de “binding”.
- No soporta WSDL [20].

Los parámetros de la solicitud de la operación GetCapabilities son los siguientes:

- **Service**  
El parámetro especifica el tipo de servicio que se desea (ej. WMS). Un mismo servidor puede soportar varios servicios.
- **Request**  
El parámetro especifica el nombre de la operación (GetCapabilities, en este caso).
- **Version**  
El parámetro especifica el número de versión del protocolo que se está consultando. Las versiones del protocolo tienen el formato x.y.z.
- **AcceptVersions**  
El parámetro especifica las versiones del protocolo que soporta el cliente, en orden de preferencia. El cliente y el servidor negocian la versión del protocolo. El servidor elige una versión dentro del AcceptVersions y se la envía al cliente. Si no soporta ninguna de las versiones, envía una excepción.
- **Sections**  
El parámetro especifica las secciones de la metadata del servicio que se deben incluir en la respuesta (ServiceIdentification, ServiceProvider, OperationsMetadata, Contents, All).
- **UpdateSequence**  
El parámetro especifica si permite que el cliente averigüe si su caché es consistente con

el servidor.

- **AcceptFormats**  
El parámetro especifica los formatos de respuesta (tipos MIME) que soporta el cliente, en orden de preferencia. Por omisión es “text/XML”.

Los parámetros y secciones de la respuesta de la operación GetCapabilities son:

- **Version**  
El parámetro especifica el número de versión del protocolo en que se envía la respuesta.
- **UpdateSequence**  
El parámetro informa al cliente si hubo actualizaciones.
- **ServiceIdentification**  
La sección presenta metadata del servidor (tipo de servicio, versión del servicio, título del servidor, descripción del servidor, palabras clave, costos, etc).
- **ServiceProvider**  
La sección presenta metadata del operador del servidor (nombre, sitio Web, información de contacto, etc.)
- **OperationsMetadata**  
La sección presenta metadata de cada operación definida para este servicio y que es soportada por el servidor, incluyendo las URLs de las mismas.
- **Contents**  
La sección presenta metadata de los juegos de datos (datasets) existentes y opcionalmente los *bounding boxes*<sup>8</sup> de los mismos. El servidor no tiene obligación de incluir información de todos sus juegos de datos, por lo que podría devolver una referencia a un servicio de catálogos (un CSW). Como ejemplo un WCS podría agregar miles de nuevos rasters por día, por lo que sería inapropiado incluirlos todos en la respuesta.

En la Figura 3, se muestra un ejemplo de una respuesta del método GetCapabilities del servicio Web Map Service (WMS).

---

<sup>8</sup> *Bounding boxes*: definido como un rectángulo delimitador mínimo en coordenadas geográficas.

```

<WMS_Capabilities xmlns="http://www.opengis.net/wms" xmlns:xlink="http://www.w3.org/1999/xlink"
version="1.3.0" updateSequence="4364" xsi:schemaLocation="http://www.opengis.net/wms
http://www.dinama.gub.uy/geoserver/schemas/wms/1.3.0/capabilities_1_3_0.xsd">
  <Service>
    <Name>WMS</Name>
    <Title>GeoServer Web Map Service</Title>
    <Abstract>
      A compliant implementation of WMS plus most of the SLD extension (dynamic styling).
    </Abstract>
    <KeywordList>
    <OnlineResource xlink:type="simple" xlink:href="http://geoserver.sourceforge.net/html/index.php"/>
    <ContactInformation>
    <Fees>NONE</Fees>
    <AccessConstraints>NONE</AccessConstraints>
  </Service>
  <Capability>
    <Request>
      <GetCapabilities>
        <Format>text/xml</Format>
        <DCPType>
        </GetCapabilities>
      <GetMap>
        <Format>image/png</Format>
        <Format>image/png; mode=8bit</Format>
        <DCPType>
        </GetMap>
      <GetFeatureInfo>
        <Format>text/html</Format>
        <Format>text/javascript</Format>
        <Format>application/json</Format>
        <DCPType>
        </GetFeatureInfo>
      </Request>
    </Exception>
    <Layer>
    </Capability>
  </WMS_Capabilities>

```

Figura 3: Respuesta del método GetCapabilities del servicio WMS

A continuación se detallan los estándares Web Map Service (WMS), Web Feature Service (WFS), Web Coverage Service (WCS) y Catalogue Service for the Web (CSW).

### Web Map Service (WMS) [19]

El estándar WMS devuelve información geográfica en formato imagen, como ser PNG o JPEG. El servicio WMS define tres operaciones básicas, dos de ellas obligatorias y la restante opcional. Las tres operaciones son:

- **GetCapabilities**

Es una operación obligatoria que describe el servicio, informando acerca de los mapas disponibles. Dentro de los elementos específicos de WMS en la respuesta de esta operación se incluyen los distintos parámetros, algunos de los más importantes son:

- **Layer**

El parámetros especifica un elemento con metadata sobre cada capa geográfica.

- **Name**  
El parámetro especifica el nombre de la capa.
- **CRS**  
El parámetro especifica el sistema de referencia de coordenadas de la capa. Puede haber varios por capa.
- **GetMap**  
Es una operación obligatoria que devuelve un mapa en forma de imagen de acuerdo con el pedido realizado.
- **GetFeatureInfo**  
Es una operación opcional, que permite al cliente pedir al servidor información particular sobre alguna entidad geográfica representada en el mapa.

### **Web Feature Service (WFS) [19]**

El estándar está relacionado con los datos de tipo vectorial y permite consultar y modificar información geográfica. Utiliza lenguaje GML que deriva de XML. Las operaciones que incluye el estándar son las siguientes: DescribeFeatureType, GetFeature, GetGmlObject, Transaction y LockFeature.

El servicio WFS es clasificado de acuerdo a las operaciones que soporta:

- **WFS Básico**  
Si soporta las operaciones DescribeFeatureType y GetFeature.
- **WFS con Xlink**  
Si soporta WFS básico y además la operación GetGmlObject.
- **WFS Transaccional**  
Si soporta WFS básico y además soporta Transaction.
- **WFS Transaccional con Xlink**  
Si soporta WFS Transaccional y GetGmlObject.

Las operaciones principales se detallan a continuación.

- **GetCapabilities**  
La operación describe los metadatos correspondientes al propio servicio WFS, incluyendo una descripción del contenido del servicio y los parámetros que acepta.
- **DescribeFeatureType**  
La operación solicita un esquema de descripción de un tipo de entidad geográfica en un determinado lenguaje, por ejemplo GML.

- **GetFeature**  
La operación devuelve un conjunto de entidades geográficas.
- **GetGmlObject**  
La operación permite obtener cualquier elemento GML a través de su identificador único (gml:id), que posee todo elemento GML.
- **Transaction**  
La operación permite realizar pedidos transaccionales, modificando las entidades geográficas ofrecidas por el servicio. Creado, modificación y borrado.

A partir del año 2009 existe una nueva especificación de WFS, la versión 2.0, que incorpora nuevas operaciones:

- **GetPropertyValue**  
La operación devuelve el valor de una propiedad de una o varias entidades geográficas.
- **GetFeatureWithLock**  
La operación al igual que GetFeature, devuelve un conjunto de entidades geográficas pero además aplica un bloqueo (LockFeature) sobre las entidades devueltas.
- **CreateStoredQuery y DropStoredQuery**  
Las operaciones crean y borran respectivamente una consulta almacenada.
- **ListStoredQueries y DescribeStoredQueries**  
Las operaciones listan y describen respectivamente las consultas almacenadas existentes.

### **Web Coverage Service (WCS) [21]**

El estándar define un protocolo para publicar coberturas <sup>9</sup>geográficas. A diferencia de WMS, no tienen sólo fines de visualización, sino que permite obtener información con su semántica completa, para realizar análisis, geoprosesos, etc. El estándar incluye las siguientes operaciones: GetCapabilities, DescribeCoverage y GetCoverage.

- **DescribeCoverage**  
La operación envía una lista de identificadores de coberturas y obtiene la descripción detallada de cada una de las coberturas especificadas.
- **GetCoverage**  
Por medio de la operación se obtiene una cobertura o una parte de la misma.

---

<sup>9</sup> Cobertura: modelo de datos georrelacionales que almacena datos vectoriales; contiene datos espaciales (ubicación) y de atributos (descriptivos) de entidades geográficas.

**Catalog Service for the Web (CSW) [22], [23]**

El estándar define un servicio base de catálogos de información geográfica. El catálogo permite a los proveedores de servicios de datos, publicar datos geográficos para ser localizados, describiendo detalladamente los datos ofrecidos. Las operaciones incluidas en el estándar son:

- **GetCapabilities**  
La operación devuelve metadata del servicio de catálogo.
- **DescribeRecord**  
La operación enumera y describe los perfiles de metadatos usados, así como la lista de elementos que pueden ser consultados.
- **GetRecords**  
La operación permite hacer una consulta sobre el catálogo de metadatos, solicitando que el servicio retorne todos los registros de metadatos que cumplan las condiciones impuestas al invocar la operación.
- **GetRecordByID**  
La operación retorna el registro de metadato correspondiente con el identificador único solicitado.
- **GetDomain**  
La operación devuelve los valores registrados para un determinado elemento de los metadatos.
- **Transaction**  
La operación permite insertar, borrar o actualizar un determinado registro identificado por su identificador único, gestionando los registros de metadatos almacenados en el catálogo.

## 2.4. Arquitectura y tecnologías GIS

Los sistemas de información geográfica o por sus siglas en inglés GIS (Geographical Information System) son sistemas que permiten manejar, analizar, almacenar, entregar y presentar información geográfica. Entre otras cosas, los GIS pueden ser usados para encontrar lugares dentro de una región o descubrir cierta relación entre distintos datos de acuerdo a su localización [24].

Tradicionalmente los sistemas de información geográfica han sido definidos como sistemas formados por componentes de hardware, software y datos para capturar, gestionar, analizar y desplegar toda forma de información geográficamente referenciada. Una arquitectura típica de un GIS consta de un servidor de base de datos geográfico (GeoDBMS), un servidor de mapas (Map Server) y un visualizador de mapas (Map Viewer) y/o de un editor de mapas (Map Editor) para proveer al usuario de una interfaz con la cual interactuar. En

las últimas décadas los GIS han evolucionado pasando de ser una aplicación de escritorio a ser la típica arquitectura en tres capas que es usualmente encontrada en aplicaciones empresariales, como la mostrada en la Figura 4 [24].

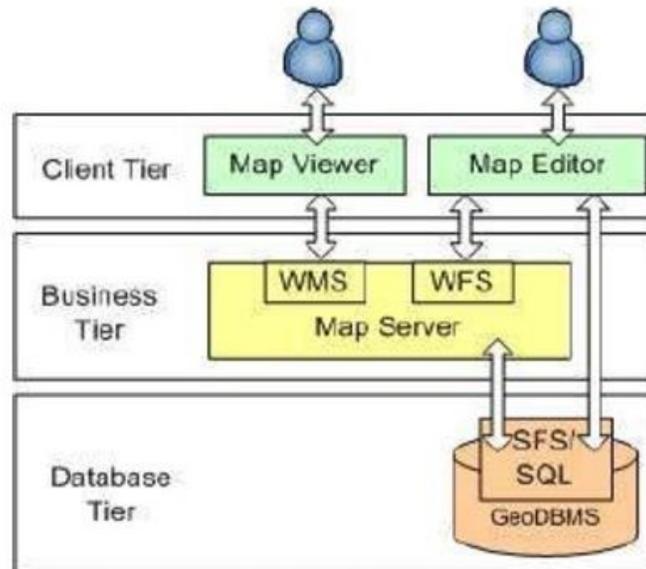


Figura 4: Principales componentes en una arquitectura GIS [24].

En la capa de base de datos (Database Tier) el GeoDBMS provee acceso a los datos geográficos almacenados en tablas. En la capa de negocio (Business Tier) el servidor de mapas (Map Server) publica los datos que son leídos y escritos en la base de datos geográfica. En la capa del cliente (Client Tier) puede ser usado, tanto un editor de mapas (Map Editor) como un visualizador de mapas (Map Viewer). Los componentes se describen a continuación:

### Sistema de gestión de base de datos geográficas, GeoDBMS

GeoDBMS es un sistema que soporta la lectura y escritura de entidades geográficas. Comúnmente los sistemas de base de datos relacionales suelen tener extensiones para convertirse en base de datos geográficas, como por ejemplo Oracle Spatial and Graph<sup>10</sup>, PostGIS<sup>11</sup> para PostgreSQL y MySQL Spatial Extensions<sup>12</sup>.

### Servidor de mapas

Un servidor de mapas es un componente de software que usualmente corre dentro de un servidor web como Apache HTTP Server<sup>13</sup>, un contenedor de servlets como Apache Tomcat<sup>14</sup> o dentro de un servidor de aplicaciones como WildFly<sup>15</sup>. La principal función de un servidor de mapas es la de proveer a los clientes de internet con mapas generados

<sup>10</sup> <http://www.oracle.com/technetwork/database/options/spatialandgraph/overview/index.html>

<sup>11</sup> <http://postgis.net/>

<sup>12</sup> <https://dev.mysql.com/doc/refman/5.7/en/spatial-extensions.html>

<sup>13</sup> <https://httpd.apache.org/>

<sup>14</sup> <http://tomcat.apache.org/>

<sup>15</sup> <http://wildfly.org/>

dinámicamente desde datos vectoriales o ráster.

### **Visualizador/Editor de mapas**

Un visualizador de mapas es una aplicación cliente que presenta mapas en una interfaz de usuario, para que la información geográfica pueda ser accedida y analizada. Un editor de mapas permite además, crear nuevas capas, modificar entidades geográficas, realizar tareas de geo-procesamiento avanzadas, entre otras.

## **2.5. Infraestructura de datos espaciales**

Una infraestructura de datos espaciales (IDE) se define como el conjunto de datos geográficos, tecnologías, normas y políticas con el objetivo de facilitar la disponibilidad y acceso a dichos datos, logrando mayor eficiencia en la gestión de los datos geográficos. A su vez las IDEs ofrecen documentación, catálogos y visualizadores de datos promoviendo la accesibilidad y transparencia de la información geográfica. [25]

Para lograr garantizar la interoperabilidad de la información geográfica se cumplen una serie de normas, estándares y especificaciones. También es necesario establecer un marco legal que asegure que los datos producidos por las instituciones serán compartidos por toda la administración y que se promueva su uso entre los ciudadanos. [26]

### **Estructura de una IDE**

Las IDEs surgen para coordinar la producción de información geográfica a nivel global. Para gestionar tal cantidad de información es necesario delegar en distintos niveles y establecer una estructura organizativa con el fin de distribuir responsabilidades y tareas, dando lugar a una estructura en forma de red con una serie de nodos interconectados. Cada nodo es administrado por un determinado organismo responsable. Es posible clasificar un nodo IDE en función de la posición en la jerarquía administrativa que ocupe el organismo responsable del nodo [4].

En la Figura 5 se muestra un ejemplo de estructura jerárquica, en cuyo nivel inferior (Ámbito local) aparece la IDE Pamplona<sup>16</sup>, luego la IDE Navarra<sup>17</sup>, seguida de la IDEE<sup>18</sup>, INSPIRE<sup>19</sup> y finalmente GSDI<sup>20</sup>.

---

<sup>16</sup> <http://ide.pamplona.es>

<sup>17</sup> <https://idena.navarra.es>

<sup>18</sup> <http://www.idee.es/>

<sup>19</sup> <https://inspire.ec.europa.eu/>

<sup>20</sup> <http://gsdiassociation.org/>



Figura 5: Ejemplo de estructura a distintos niveles para una IDE en particular (IDE de Pamplona) [4]

A medida que descendemos en la jerarquía se aumenta el nivel de detalle de la información geográfica y se disminuye el área geográfica gestionada. Los nodos de la IDE se relacionan entre sí. Cuando dos nodos de un mismo nivel se relacionan, entonces se dice que existe una relación horizontal. Cuando dos nodos de distinto nivel se relacionan, entonces se dice que existe una relación vertical [4].

### Componentes de la IDE [4]

La IDE se divide en distintos componentes que se describen a continuación:

- **Datos**  
Los datos deben de estar al alcance de los usuario con las restricciones de uso que el propietario de los datos decida.
- **Estándares**  
Los estándares de datos y servicios que hacen posible la interoperabilidad. Particularmente los especificados por las normas ISO, el OGC y las propias normas de un país o región.
- **Políticas**  
Las políticas definidas por los gobiernos para regular y fomentar el uso de la información geográfica.
- **Marco legal**  
El marco legal regula aspectos como qué información es oficial, qué ocurre con los

derechos de autor y las licencias de uso o qué organismos públicos tienen la obligación de publicar una cartografía dada.

- **Tecnología**

Las tecnologías sirven de base para hacer pública la información a través de internet.

- **Usuarios**

Existen varios tipos de usuarios con distintas necesidades y requerimientos, así como también un capital humano que se encargue del análisis de la información y la implementación de servicios web.

## 2.6. IDEuy

La infraestructura de datos espaciales del Uruguay (IDEuy) es una iniciativa del Poder Ejecutivo, que apunta a posicionarse como un instrumento fundamental en el apoyo a la toma de decisiones mediante el uso de información geográfica de calidad [27].

Se puede considerar que la creación de la infraestructura de datos espaciales del Uruguay (IDEuy) se remonta a la sanción del decreto [28] del Poder Ejecutivo del 16 de junio del año 2006 donde se considera la necesidad de crear un Plan Nacional de Catastro y la coordinación de la infraestructura de datos espaciales a nivel nacional. Luego en octubre del año 2013 y mediante los artículos 35 y 36 de la Ley 19.149 [29] se crea la infraestructura de datos espaciales como un órgano desconcentrado de la Presidencia de la República. La Ley fue reglamentada por el Decreto N° 390/014 [30].

La infraestructura de datos espaciales está conformada por una Comisión Directiva que se encarga de la conducción cotidiana y un Consejo Nacional Honorario de Información Geográfica, que diseña las líneas generales de acción de datos geográficos<sup>21</sup>.

La infraestructura de datos espaciales (IDE) tiene los siguientes cometidos<sup>22</sup>:

- Coordinar, planificar y promover la producción de información geográfica en el territorio nacional.
- Garantizar por medio de normas y estándares la interoperabilidad, actualización, calidad y acceso a la información geográfica nacional.
- Integrar la información geográfica de los diferentes niveles de la Administración.
- Construir el geoportal.
- Favorecer la eficiencia en el gasto público.
- Generar los ámbitos de discusión adecuados para las normas de ejecución.

En la Figura 6 se muestra un diagrama conceptual representando las entidades más relevantes de la IDEuy y cómo se relacionan.

---

<sup>21</sup> <http://ide.uy/estructura>

<sup>22</sup> <http://ide.uy/cometidos>

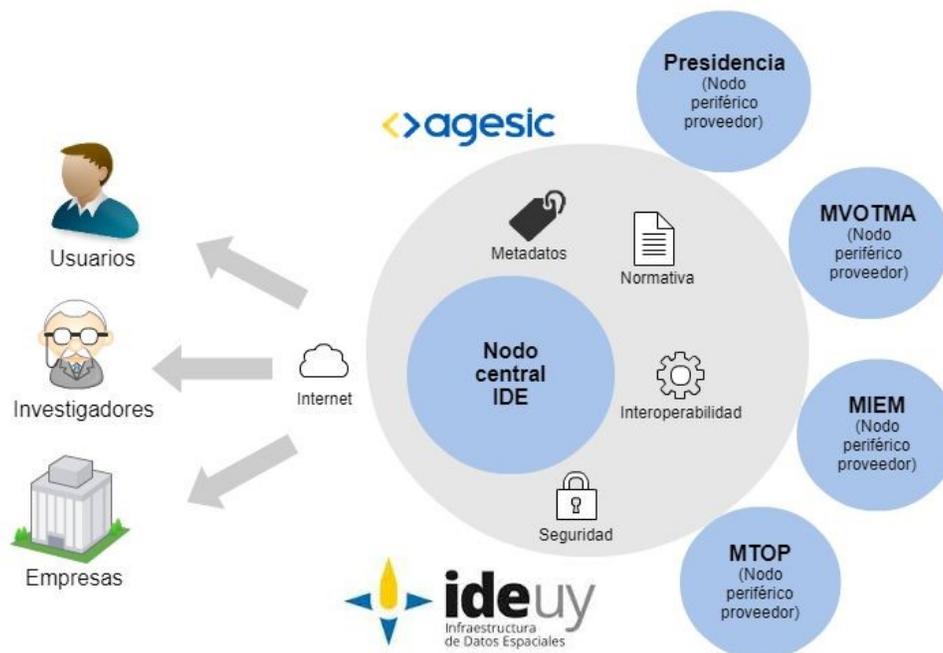


Figura 6: Diagrama conceptual de IDEuy

A la derecha de la Figura 6 se muestran algunos de los organismos que forman los nodos periféricos que proveen de información geográfica a la IDEuy. Como nodo central se ve a la IDEuy que publica la información geográfica reunida y que luego es accedida por diferentes actores por medio de internet haciendo uso del geoportal.

## 2.7. Calidad de Servicio

Un requerimiento fundamental para las aplicaciones con arquitecturas orientadas a servicios (Service Oriented Architecture, SOA [31]) es que operen de tal forma que funcionen de manera fiable y proporcionen servicios consistentes a una variedad de niveles, lo que requiere centrarse no solo en las propiedades funcionales sino también en las propiedades no funcionales del servicio. En este contexto la calidad de servicio (Quality of Service, QoS), está teniendo una alta prioridad para los prestadores de servicios y sus clientes [2].

QoS refiere a la habilidad de un servicio web de responder efectivamente a las solicitudes realizadas y llevarlas a cabo a un nivel acorde con las expectativas del proveedor y del cliente. Varias características de calidad de un servicio tales como disponibilidad, conectividad y altos niveles de respuesta, se vuelven fundamentales para mantener un negocio competitivo y viable debido al impacto que pueden tener en su operativa diaria [2]. La mayoría de los trabajos referidos a QoS identifican características de calidad similares aunque en algunos casos presentan diferencias en su semántica [3].

Una forma de tratar con calidad de servicio es a través de los modelos de calidad. Por ejemplo, el modelo de la OASIS define factores de calidad agrupados en las áreas de calidad empresarial y calidad de sistemas. Algunos de ellos son interoperabilidad, procesamiento de negocio, seguridad y calidad de medición del nivel de servicio, los mismos se pueden ver la Figura 7.

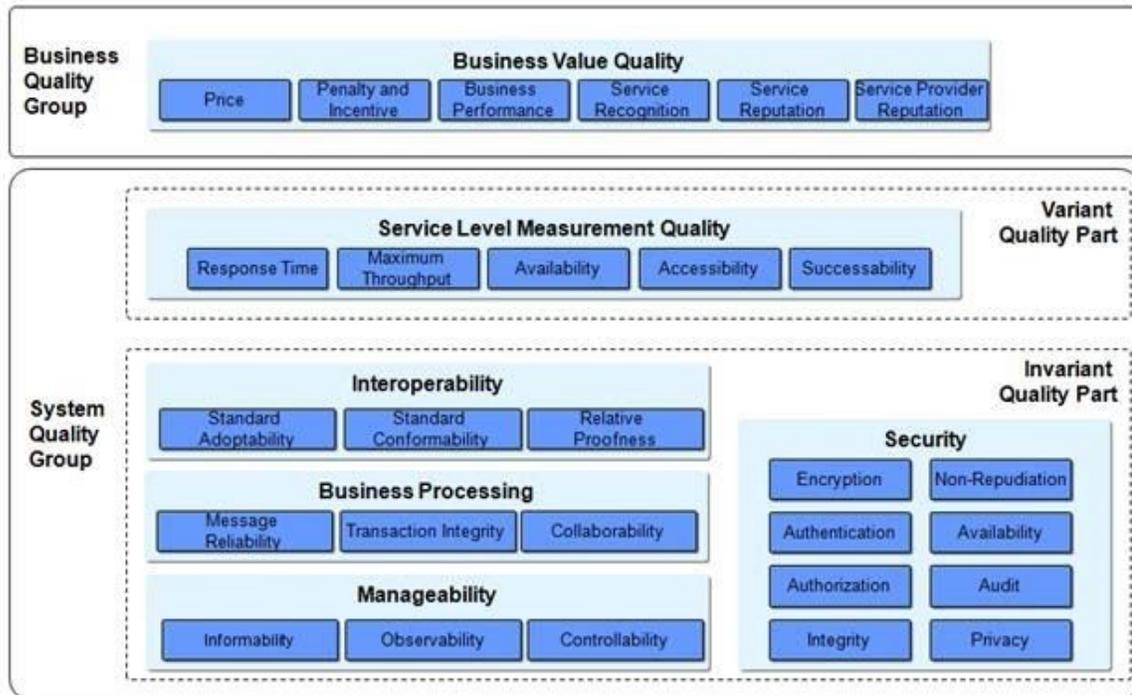


Figura 7: Factores de calidad de la OASIS [32]

Por todos estos motivos mencionados, la calidad de servicio es muy relevante, en particular en el área de servicios web geográficos y en el contexto de las IDEs.

## 2.8. Modelo de calidad de servicio para IDEuy

En esta sección se detallan los conceptos principales y componentes del modelo de calidad utilizado.

### 2.8.1. Introducción

Las características de calidad se organizan y documentan en modelos de calidad. En esta sección se presenta el modelo de calidad para la evaluación de servicios web geográficos de la IDEuy, utilizado en el contexto de este proyecto y que es descrito tanto en [3] como en [33]. El modelo se define mediante los siguientes componentes: objetos medibles del modelo, dimensiones y factores del modelo, métricas del modelo y métodos del modelo.

### **Objetos medibles**

Los objetos medibles son entidades a ser evaluadas y de las cuales se puede obtener un indicador de calidad.

### **Dimensiones y factores**

Una dimensión captura una faceta de calidad a alto nivel (p. ej. rendimiento). Un factor representa un aspecto particular de una dimensión (p. ej. tiempo de respuesta).

### **Métricas**

Una métrica es un instrumento utilizado para medir un factor de calidad (p. ej. tiempo de respuesta promedio diario). Cada métrica se describe en función de los siguientes atributos:

- Nombre  
Nombre descriptivo de la métrica.
- Unidad  
Unidad en que se mide la métrica.
- Granularidad  
Representa el objeto medible al que aplica.
- Semántica  
Descripción del procedimiento de medición.

### **Método**

Un método es un proceso que implementa una determinada métrica [33]. Los métodos pueden ser de medición o de agregación.

- Métodos de medición  
Los métodos de medición son los que calculan la calidad de un objeto midiéndola directamente sobre el objeto (p. ej. Un programa que invoca a un servicio para conocer su tiempo de respuesta) [3].
- Métodos de agregación  
Los métodos de agregación son los que calculan la calidad de un objeto compuesto como la agregación de los valores de calidad de partes de ese objeto (p. ej. el tiempo de respuesta de un servicio se podría calcular como el promedio de los tiempos de respuesta de sus operaciones) [3].

En la Figura 8 se muestra cómo se estructuran las dimensiones, factores, métricas y métodos del modelo.

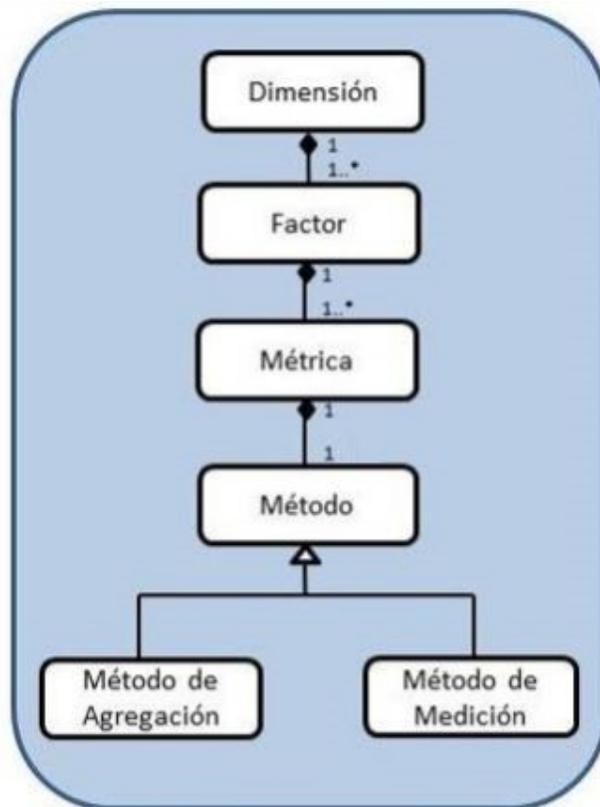


Figura 8: Estructura de dimensiones, factores, métricas y métodos [3].

### 2.8.2. Objetos medibles del modelo

Los objetos medibles identificados para el modelo de calidad en el contexto de la IDEuy son:

- **Institución**  
Representa una entidad proveedora de información geográfica en una IDE. Por ejemplo, la Intendencia de Montevideo (IM<sup>23</sup>), el Ministerio de Transporte y Obras Públicas (MTO<sup>24</sup>) y el Ministerio de Vivienda y Ordenamiento Territorial y Medio Ambiente (MVOTMA<sup>25</sup>).
- **Nodo**  
Representa un servidor que provee servicios web geográficos. Una institución puede tener varios nodos. Por ejemplo, el MVOTMA actualmente tiene dos nodos publicados en la IDEuy<sup>26</sup>.

<sup>23</sup> <http://sig.montevideo.gub.uy/>

<sup>24</sup> <http://geoportal.mtop.gub.uy/>

<sup>25</sup> <http://www.mvotma.gub.uy/>

<sup>26</sup> <http://ide.uy/nodos-ideuy>

- **Servicio**  
Representa un servicio web geográfico publicado en un nodo. Por ejemplo, un nodo puede publicar servicios del tipo WMS y WFS.
- **Método**  
Representa un método de un servicio web geográfico. Por ejemplo, el servicio WMS provee los métodos GetCapabilities y GetMap, entre otros.
- **Capa**  
Representa una capa geográfica publicada en un nodo. Por ejemplo, en el nodo MTOP se encuentra publicada una capa de rutas nacionales. Una capa puede accederse desde distintos servicios (p. ej. WMS, WFS).

Además de los objetos medibles mencionados y a los efectos de la solución planteada, se consideran como objetos medibles a las infraestructuras de datos espaciales, por ejemplo la IDEuy.

### 2.8.3. Dimensiones y factores del modelo

El modelo se estructura mediante las dimensiones y factores presentados en la Tabla 2.

<b>Dimensión</b>	<b>Factor</b>
Seguridad	Autenticación Autorización Integridad Protección Privacidad
Confiabilidad	Accesibilidad Disponibilidad Robustez Reputación
Rendimiento	Tiempo de respuesta Capacidad
Interoperabilidad	Soporte de estándares Sistema de referencia Estructura de datos
Publicación datos	Representación gráfica Datos abiertos Área de cobertura Formatos soportados Licencia datos Institución competente

Metadatos	Publicación catálogo Metadatos capa Metadatos servicio
Usabilidad	Facilidad de aprendizaje

Tabla 2 - Dimensiones y factores del modelo [33]

#### 2.8.4. Métricas del modelo

Las métricas del modelo son presentadas en las Tablas siguientes, agrupadas bajo factores y dimensiones.

Dimensión: Seguridad	
Factor	Métrica
Autenticación	Autenticación de clientes
Autorización	Autorización a nivel de servicio Autorización a nivel de método Autorización sobre los datos
Integridad	Integridad de datos
Protección	Expone métodos transaccionales Anti-DoS Información en excepciones
Privacidad	Devuelve datos sensibles

Tabla 3.1 - Métricas dimensión seguridad [33]

Dimensión: Confiabilidad	
Factor	Métrica
Accesibilidad	Accesibilidad diaria al nodo
Disponibilidad	Disponibilidad del servicio
Robustez	Errores diarios del servicio Tolerancia a parámetros nulos Tolerancia a parámetros largos
Reputación	Reputación del servicio Reputación del proveedor Veracidad de la capa Adecuación al uso

Tabla 3.2 - Métricas dimensión confiabilidad [33]

Dimensión: Rendimiento	
Factor	Métrica
Tiempo de respuesta	Promedio tiempo de respuesta diario
Capacidad	Máximo de invocaciones concurrentes Tope máximo de objetos Cantidad invocaciones por día Cantidad de información diaria

Tabla 3.3 - Métricas dimensión rendimiento [33]

Dimensión: Interoperabilidad	
Factor	Métrica
Soporte de estándares	Utilización de estándares OGC Adopción de estándares OGC Excepciones en formato OGC Soporta múltiples plataformas Estilo de capas en formato SLD
Sistema de referencia	Capas del servicio con CRS adecuado (IDEuy) Capa con CRS adecuado (IDEuy) Transformación de CRS
Estructura de Datos	Referencia domiciliaria

Tabla 3.4 - Métricas dimensión interoperabilidad [33]

Dimensión: Publicación datos	
Factor	Métrica
Representación gráfica	Soporta imagen simplificada Soporta imagen vacía Elementos gráficos adicionales Correctitud representación gráfica
Datos Abiertos	Grado de apertura de los datos
Área de cobertura	Jurisdicción del dato
Formatos soportados	Formato PNG Formato KML Formato text/html método GetFeatureInfo Formato text/xml método GetFeatureInfo Formato excepción vnd.ogc.se_inimage Formato excepción vnd.ogc.se_blank Cantidad de formatos soportados Cantidad de formatos de excepciones soportadas

Licencia datos	Licenciamiento Creative Commons
Institución competente	Publicación datos competentes

Tabla 3.5 - Métricas dimensión publicación de datos [33]

Dimensión: Metadatos	
Factor	Métrica
Publicación catálogo	Registrado en el catálogo metadatos IDE Tiene servicio CSW
Metadatos capa	Tiene metadatos Publicación de acuerdo al perfil Información de calidad Fecha del dato Compleitud de metadatos Plan de actualización
Metadatos servicio	Leyenda de la capa Especifica rango útil

Tabla 3.6 - Métricas dimensión metadatos [33]

Dimensión: Usabilidad	
Factor	Métrica
Facilidad de aprendizaje	Errores descriptivos Ejemplos Documentación

Tabla 3.7 - Métricas dimensión usabilidad [33]

### 2.8.5. Perfiles de evaluación

Para evaluar la calidad de los servicios web geográficos se propone la definición de perfiles. El modelo presenta dos perfiles de evaluación, el perfil de evaluación básico y el perfil de evaluación ponderado.

- Perfil de evaluación básico  
Un perfil de evaluación básico es un conjunto de métricas aplicables a un mismo objeto medible (p. ej. servicio) y rangos de valores esperados para dichas métricas según el objetivo de la evaluación.

El rango es un valor asociado a una métrica y representa el valor esperado contra el cual se debe comparar para obtener un resultado (éxito o fracaso) al realizar una evaluación.

### Evaluación con perfil básico

La evaluación de un objeto medible bajo un perfil básico devuelve como resultado un valor, que indica el cumplimiento o no de una métrica. En la Tabla 4 se presenta un ejemplo de perfil básico [3].

Perfil de calidad (Servicio)		
Métrica	Valor esperado	Dimensión
Disponibilidad diaria del servicio	> 70 %	Confiabilidad
Capas del servicio con CRS adecuado(IDEuy)	> 50 %	Interoperabilidad

Tabla 4 - Ejemplo de perfil básico para un servicio [3]

- Perfil de evaluación ponderado

Un perfil de evaluación ponderado es un conjunto de métricas aplicables al mismo objeto medible y un conjunto de ponderaciones según el objetivo de la evaluación. Las ponderaciones se aplican a todos los elementos de la jerarquía del modelo de calidad: Dimensiones, factores, métricas, rangos de valores que pueden tomar las métricas [3].

En la Tabla 5 se presenta la escala de valores utilizados para llevar a cabo la ponderación de los elementos del modelo [3].

Valor	Descripción	Recíproco
1	Igualmente preferido	1
2	De igualmente a moderadamente	1/2
3	Moderadamente preferido	1/3
4	De moderadamente a fuertemente	1/4
5	Fuertemente preferido	1/5
6	De fuertemente a muy fuerte	1/6
7	Muy fuertemente preferido	1/7
8	De muy fuerte a extremadamente	1/8
9	Muy recomendado	1/9

Tabla 5: Valores de ponderación [3]

**Evaluación con perfil ponderado**

La evaluación de un objeto medible bajo un perfil ponderado devuelve como resultado un índice de calidad, que indica el cumplimiento o no de una métrica. El cálculo del índice de calidad se realiza mediante la fórmula mostrada en la Figura 9.

$$ph = \text{Ponderación de elemento hijo}$$

$$pp = \text{Ponderación de elemento padre}$$

$$\text{Índice}(phi, ppj) = \sum_{i=1..n} \text{ponderaciónElemento}(phi.valores) * \text{ponderacionElementoPadre}(ppi)$$

Figura 9: Fórmula para el cálculo del índice de calidad [3]

Luego en la Figura 10 se muestra a modo de ejemplo, como el cálculo del índice de calidad es aplicado. En negrita se marcan los valores de las ponderaciones. Los recuadros celestes representan los elementos del modelo de calidad a ponderar: métrica, factor y dimensión. Luego debajo se muestra el cálculo de cada una de las ramas del árbol.

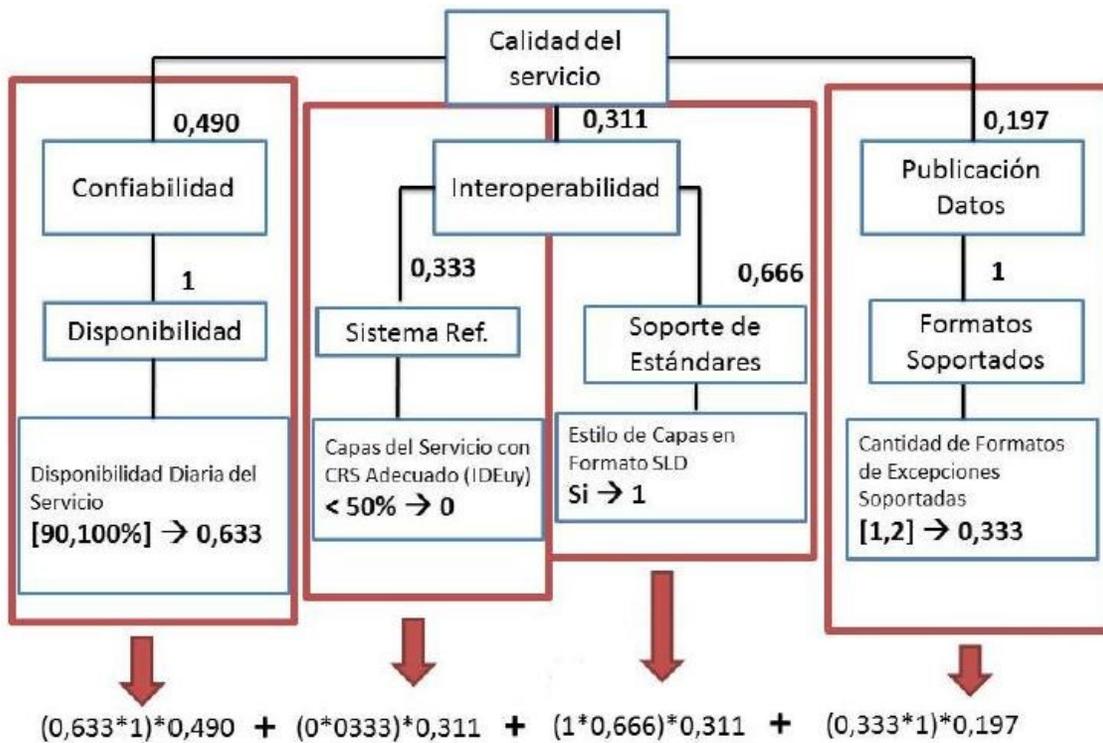


Figura 10: Ejemplo de cálculo de índice de calidad [3]

Como resultado se obtiene que el Índice de Calidad = 0,582.

## 3. Análisis

En este capítulo se analizan los requerimientos del sistema, describiendo su comportamiento general así como las funcionalidades específicas con las que debe de contar. Además se hace un análisis de las soluciones existentes para la evaluación de calidad de servicios geográficos.

### 3.1. Marco de trabajo

Muchas organizaciones centran parte de su operativa intercambiando información o ejecutando procesos a través de servicios web que brindan información geográfica. Particularmente entidades e instituciones de gobierno diariamente publican y consumen servicios web geográficos. Dentro del marco de trabajo, la IDEuy coordina y promueve la producción de información geográfica en el territorio nacional y le es de interés contar con indicadores de calidad de servicios web geográficos con el propósito de poder mejorarlos.

En el contexto de las IDEs, los estándares son los elementos que van a permitir interoperabilidad, e independencia de los formatos, definiendo el marco común de comunicación entre clientes y servidores. El OGC define algunos de los estándares más relevantes dentro de los sistemas de información geográfica, brindando estándares abiertos e interoperables, como parte de una arquitectura orientada al procesamiento e intercambio de información geográfica distribuida.

Si bien se toma como referencia el modelo de calidad planteado en la Tesis del Lic. De Los Reyes [3], también se considera que puede ser de utilidad utilizar otros modelos de calidad. A su vez, el modelo de referencia podría actualizarse en el futuro, cambiando las metodologías de medición. En base a estos factores, es necesario adaptarse a distintos modelos de calidad y a la actualización de los mismos para poder realizar una evaluación completa de la calidad de los servicios web geográficos.

En este contexto se identifican distintos tipos de usuarios con distintos niveles de acceso a la información, según las competencias sobre el área a la cual pertenecen y las funcionalidades a las que van a acceder. Concretamente se necesita diferenciar tres roles de usuario según su jerarquía: a nivel de institución, a nivel de IDE y a nivel general. Por otro lado, se necesita un usuario con rol técnico, que cuente con conocimientos en el área, y que tenga las responsabilidades de administración de IDEs y modelos de calidad en la solución.

En la Figura 11 se puede ver el diagrama conceptual del marco de trabajo.

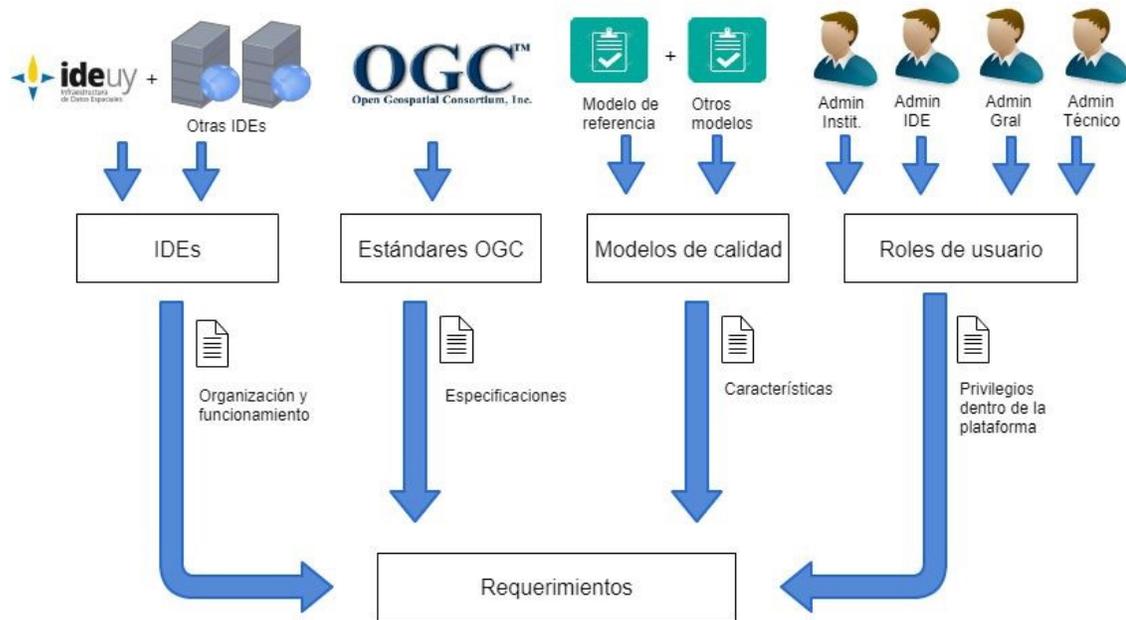


Figura 11: Diagrama conceptual del marco de trabajo

## 3.2. Análisis de requerimientos

Los requerimientos para la construcción de la herramienta fueron recabados en diferentes instancias de reunión con las tutoras de proyecto y tomando como referencia la Tesis del Lic. De Los Reyes [3]. Además se realizó una reunión en Presidencia<sup>27</sup> que contó con miembros de AGESIC, Presidencia, IDEuy y las tutoras de proyecto. En la reunión se obtuvieron nuevos requerimientos y se validaron conceptos. A continuación se detallan los requerimientos relevados, tomados como base para la construcción de la herramienta.

### 3.2.1. Requerimientos funcionales

A continuación se describen los requerimientos funcionales relevados para la construcción de la herramienta.

- **Roles de usuarios**

Se debe contar con control de acceso basado en roles, tanto a nivel funcional como a nivel de datos. Mediante los cuales cada usuario tendrá acceso a ciertas funcionalidades y permisos de evaluación sobre los objetos medibles que dispone el sistema. Los roles de usuario son: Administrador Técnico, Administrador General, Administrador IDE y Administrador Institucional.

La Tabla 6 presenta la clasificación de roles de usuario junto a sus funcionalidades.

<sup>27</sup> <https://www.presidencia.gub.uy/>

Rol de usuario	Funcionalidades
Administrador Técnico	Puede evaluar todos los objetos medibles del sistema. Puede realizar alta, baja y modificación de objetos medibles. Puede realizar alta, baja y modificación de usuarios. Puede realizar alta, baja y modificación de perfiles de evaluación. Puede realizar alta, baja y modificación de modelo de calidad. Tiene acceso a la vista de reportes del sistema.
Administrador General	Puede evaluar todos los objetos medibles del sistema. Tiene acceso a la vista de reportes del sistema.
Administrador IDE	Puede evaluar una IDE determinada así como los objetos medibles incluidos en ella. Tiene acceso a la vista de reportes del sistema.
Administrador Institucional	Puede evaluar una institución determinada así como los objetos medibles incluidos en ella. Tiene acceso a la vista de reportes del sistema.

Tabla 6 - Roles de usuarios

- **Login/logout de usuario**

Se debe contar con un módulo de autenticación para acceder a la herramienta.

- **Alta, baja y modificación de usuarios**

El usuario Administrador Técnico cuenta con los permisos necesarios para ingresar, editar y eliminar usuarios. Al dar de alta un usuario también se dan permisos sobre los objetos medibles que puede evaluar, dependiendo del tipo de usuario creado y la IDE e Institución a la que pertenezca.

- **Alta, baja y modificación de objetos medibles**

El usuario Administrador Técnico cuenta con los permisos necesarios para ingresar, editar y eliminar objetos medibles. Al crear un objeto medible se debe asignar permisos de evaluación sobre el mismo, y solamente para los usuarios que ya cuenten con permisos de evaluación sobre objetos medibles en un mismo nivel de jerarquía que el objeto medible creado.

- **Alta, baja y modificación de perfiles de evaluación**

El usuario Administrador Técnico puede crear, modificar y eliminar perfiles de evaluación. También puede asignar valores de rangos y agregar ponderaciones al perfil de evaluación en caso de ser ponderado.

- **Alta, baja y modificación de modelo de calidad**

El usuario Administrador Técnico puede ingresar nuevos modelos de calidad así como crear nuevas dimensiones, factores o métricas. También puede editar las ya existentes. En caso de ingresar una nueva métrica, se debe adjuntar un archivo Java estándar

(JAR) con la implementación de método de medición.

- **Evaluación de objeto medible**

Los usuarios pueden evaluar los objetos medibles, siempre que tengan permisos de evaluación sobre ellos. Para esto se debe seleccionar un objeto medible y un perfil de evaluación. Las evaluaciones se pueden realizar a demanda, cuando el usuario lo solicita, y de forma periódica cada cierto período de tiempo.

- **Soporte a distintas técnicas de medición**

Se deben implementar técnicas de medición que permitan la evaluación a demanda así como la evaluación planificada de un objeto medible, dependiendo de la métrica y método ejecutado. Cuando la evaluación sea a demanda, el método asociado a la métrica será ejecutado en el momento. Cuando la evaluación sea planificada, el método asociado a la métrica será ejecutado cada cierto periodo de tiempo.

- **Reportes**

Los usuarios tienen acceso a la vista de reportes del sistema que muestra información sobre las evaluaciones realizadas de manera gráfica y detallada de los resultados obtenidos. En la Figura 12 se muestran los roles de usuario junto a sus casos de uso.

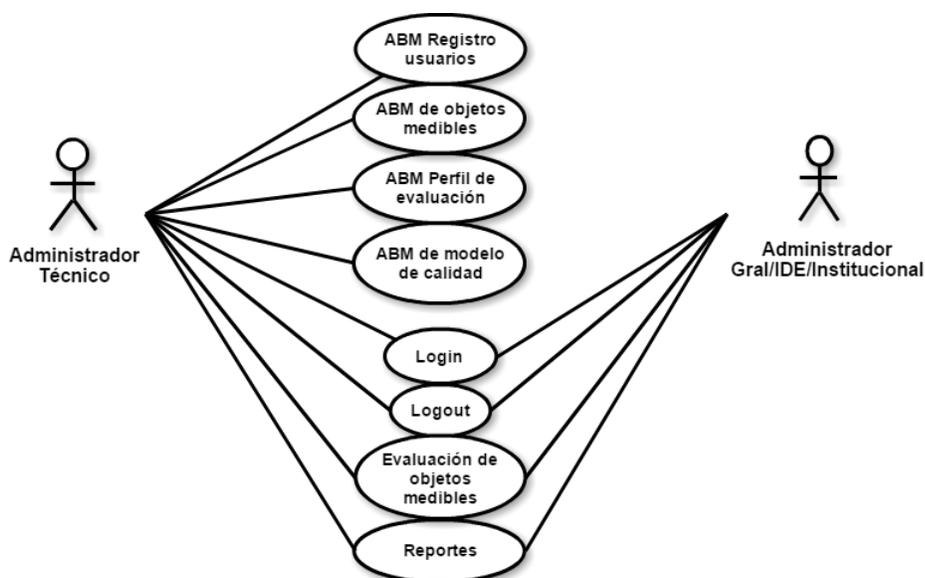


Figura 12: Modelo de casos de uso

### 3.2.2. Requerimientos no funcionales

A continuación se describen los requerimientos no funcionales reunidos para la construcción del sistema.

- **Extensibilidad del modelo de calidad**

Si bien se parte de un modelo de calidad previamente definido, el sistema debe permitir modificarlo así como ingresar nuevos modelos, con sus respectivas dimensiones, factores, métricas y métodos.

- **Desempeño**

Se debe optimizar la cantidad de pedidos a los servicios de cada IDE de forma de no sobrecargar el tráfico a los servidores.

- **Mantenibilidad**

El sistema debe ser fácilmente mantenible y debe utilizar estándares de código para su mejor entendimiento.

- **Seguridad**

Mediante un control de acceso basado en roles se restringe el ingreso a personas no registradas en el sistema.

- **Lenguaje de programación**

La solución deberá ser desarrollada en el lenguaje de programación Java.

### 3.3. Relevamiento de soluciones existentes

En esta sección se analizan herramientas y soluciones relacionadas a este proyecto. En particular, el análisis se centra en la resolución de dos dificultades: el monitoreo de servicios web geográficos y el procesamiento de esquemas OGC.

Para el relevamiento de soluciones se consultaron a los sitios de referencia en el área, y las principales organizaciones dedicadas a la investigación y estudio de la información geográfica, como lo son: OGC<sup>7</sup>, INSPIRE<sup>28</sup>, OSGeo<sup>29</sup>, FGDC<sup>30</sup>, GeoSur<sup>31</sup> y el IPGH<sup>32</sup>. Además se recurrió al Portal Timbó<sup>33</sup> para consultar trabajos académicos relacionados en esta temática.

Debido a la gran variedad de soluciones y las distintas metodologías para cumplir con los objetivos planteados, se dividen las soluciones en tres categorías principales:

- Evaluación de servicios web genéricos.
- Evaluación de servicios web geográficos.
- Simulación de fallos.

<sup>28</sup> <http://inspire.ec.europa.eu/>

<sup>29</sup> <http://www.osgeo.org/>

<sup>30</sup> <https://statuschecker.fgdc.gov/>

<sup>31</sup> <https://www.geosur.info/>

<sup>32</sup> <https://www.ipgh.org/>

<sup>33</sup> <http://www.timbo.org.uy/>

Para cada una de estas categorías, se caracterizaron los diferentes tipos de herramientas, mostrando las principales diferencias entre sí. Las características principales se basan en las funcionalidades soportadas, su licenciamiento y la compatibilidad con el proyecto.

Algunas de las características mencionadas fueron factores influyentes en los criterios de selección. En lo referente al licenciamiento de las herramientas, se seleccionaron únicamente las de licencias de código abierto, en cualquiera de sus categorías. En cuanto a la compatibilidad con el proyecto, se buscaron herramientas desarrolladas en java para que sea más fácil su integración.

### **Evaluación de servicios web genéricos**

Las herramientas de evaluación y monitoreo de servicios web, son sistemas para recopilar y almacenar información de estado sobre algún proceso, función u otros sistemas de software en general, para evaluar y controlar métricas de calidad basadas en Service Level Agreement (SLA), Service Level Objectives (SLO) o cualquier otro tipo de acuerdo de calidad. [34]

Existe una gran cantidad de herramientas de este tipo, por lo cual se realizó una selección con algunas de las más idóneas para esta categoría:

- **Fixd**<sup>34</sup>

Las funcionalidades más importantes son: creación de reglas de ruteo, utilización de sesiones de prueba, captura de request y verificación de asserts sobre los mismos, configuración asincrónica de respuestas HTTP y retraso de tiempos de respuesta. Para su utilización se debe descargar el archivo JAR y agregarlo al proyecto o agregar la dependencia Maven<sup>35</sup> correspondiente en el archivo pom.xml del proyecto.

El proyecto está publicado bajo licencia de Apache 2.0.

- **WebInject, Web Application and Web Services Test Tool**<sup>36</sup>

Herramienta gratuita para pruebas automatizadas de aplicaciones y servicios web. Puede ser utilizada para probar los componentes individuales del sistema que tienen interfaces HTTP incluyendo servicios XML / SOAP, o como instrumento para crear un conjunto de pruebas funcionales, de aceptación y regresión, obteniendo reportes de sus resultados.

Webinject se puede utilizar como framework de pruebas junto con la interfaz de usuario Webinject User Interface o como ejecutor de test independiente a integrar con otra aplicación. El proyecto es Open Source, publicado bajo licencia de GNU General Public License (GPL).

---

<sup>34</sup> <http://bigtesting.org/fixd/>

<sup>35</sup> <https://maven.apache.org/>

<sup>36</sup> <http://www.webinject.org/>

- **SoapUI, Functional Testing for SOAP and REST APIs**<sup>37</sup>

Solución de pruebas funcionales multiplataforma de código libre y abierto. Permite crear y ejecutar pruebas automatizadas funcionales, de regresión y de cumplimiento. Permite obtener cobertura de las pruebas realizadas. Es compatible con todos los protocolos y tecnologías estándar.

SoapUI se puede utilizar como framework de pruebas o integrarlo a proyectos a través de distintos plugins disponibles (actualmente cuenta con plugins para IntelliJ IDEA, Maven, Eclipse, NetBeans, and JBoss). El proyecto tiene dos versiones, una comunitaria y otra empresarial.

La Tabla 7 muestra un resumen comparativo de las herramientas disponibles.

Herramienta	Funcionalidades	Integración	Licencia
<b>Fixd</b>	<ul style="list-style-type: none"> <li>- Reglas de ruteo</li> <li>- Sesiones de prueba</li> <li>- Captura de request</li> <li>- Configuración asincrónica de respuestas HTTP</li> <li>- Retraso de tiempos de respuesta</li> </ul>	Soporte Maven y jar	Apache 2.0 [35]
<b>WebInject, Web Application and Web Services Test Tool</b>	<ul style="list-style-type: none"> <li>- Pruebas de componentes</li> <li>- Pruebas de regresión</li> <li>- Reportes de resultados</li> </ul>	Plugins y soporte con Nagios	GNU General Public License [35]
<b>SoapUI, Functional Testing for SOAP and REST APIs</b>	<ul style="list-style-type: none"> <li>- Pruebas de regresión</li> <li>- Pruebas de cumplimiento</li> <li>- Pruebas de carga rápida</li> <li>- Cobertura de pruebas</li> </ul>	Plugins para IntelliJ IDEA, Maven, Eclipse, NetBeans, and JBoss	European Union Public License [39]

Tabla 7 - Comparativa de herramientas de evaluación servicios web

Las funcionalidades presentes en las herramientas mencionadas pueden ser aplicadas en el monitoreo y la evaluación de calidad de los servicios web, se encontró una gran cantidad de información y trabajos académicos en el área, así como una gran variedad de herramientas disponibles.

### **Evaluación de servicios web geográficos**

En esta categoría, se investigó el uso de herramientas de testing enfocadas específicamente a la evaluación de servicios web geográficos. Como resultado se obtuvieron herramientas más específicas, ya que resuelven dificultades típicas en el geoprocesamiento de datos.

A continuación se detallan las herramientas mas relevantes para esta categoría:

<sup>37</sup> <https://www.soapui.org/>

- **TEAM Engine**<sup>38</sup>

Test, Evaluation, And Measurement Engine. Es una herramienta basada en Java para evaluación de Servicios Web de OGC. Ejecuta casos de prueba escritos en lenguaje OGC CTL, TestNG, entre otros. Es generalmente usado para verificar el cumplimiento de estándares en los Servicios Web, y es la herramienta oficial usada por la propia OGC. Su código está disponible en GitHub.

El proyecto está publicado bajo licencia de Apache 2.0.

- **Service Status Checker**<sup>39</sup>

Servicio que permite monitorear, alertar y analizar el estado de Servicios Web Geográficos, realizando varias pruebas de validación al servicio y estableciendo una valoración para el mismo. Genera informes con resumen de resultados y diagnóstico de las pruebas realizadas. Es posible evaluar los servicios en tiempo real, introduciendo la URL en cuestión. La herramienta permite integrar el servicio de pruebas con otras aplicaciones.

Para hacer uso del servicio, primeramente se debe de tener un usuario, luego crear un feed XML conteniendo los servicios a probar y almacenarlo en un WAF (web accesible folder) en el servidor web, para luego registrar la URL del feed en el Services Status Checker.

El proyecto está publicado bajo licencia GNU Lesser General Public License

- **OGC Schemas and Tools**<sup>40</sup>

Proyecto desarrollado en Java con el objetivo de procesar todos los esquemas OGC por medio de JAXB (Java Architecture for XML Binding) y brindar herramientas para la manipulación de documentos XML en entornos Java GIS.

El proyecto compila la mayoría de los esquemas de OGC y permite utilizar las clases generadas para trabajar con documentos XML que cumplan con las especificaciones de OGC a través de JAXB. A su vez, permite reutilizar archivos de enlace JAXB (archivos de extensión .xjb) pertenecientes al proyecto y compilar nuevos esquemas de interés.

El proyecto es Open Source, publicado bajo licencia de GNU General Public License.

- **GeoTools**<sup>41</sup>

Proyecto desarrollado en Java, que provee métodos para la manipulación de datos espaciales. Esta librería cumple con las especificaciones OGC. A su vez, proporciona acceso a datos en distintos formatos, un marco de geoprocésamiento, implementación de estándares como WMS, WFS, WCS, SLD y SE, un motor para la generación de

---

<sup>38</sup> <http://cite.opengeospatial.org/teamengine/>

<sup>39</sup> <https://statuschecker.fgdc.gov/>

<sup>40</sup> <http://confluence.highsource.org/display/OGCS/Home>

<sup>41</sup> <http://docs.geotools.org/>

imágenes, etc.

El proyecto es Open Source y esta publicado bajo licencia GNU Lesser General Public License.

La Tabla 8 muestra la tabla comparativa.

Herramienta	Funcionalidades	Integración	Licencia
<b>TEAM Engine</b>	- Testing de servicios web de OGC. - Ejecución de casos de prueba	Maven o Gradle	Apache 2.0 [35]
<b>Service Status Checker</b>	- Monitor de servicios web geográficos - Validación al servicio - Informes de resultados	Validación mediante servicio web	GNU Lesser General Public License [37]
<b>OGC Schemas and Tools</b>	- Procesamiento de esquemas OGC - Procesamiento mediante JAXB - Manipulación de documentos XML en entornos Java GIS - Soporte con gran mayoría de esquemas OGC	Maven o Gradle	GNU General Public License [36]
<b>GeoTools</b>	- Procesamiento de esquemas OGC - Procesamiento mediante JAXB - Manipulación de documentos XML en entornos Java GIS	Maven o Gradle	GNU Lesser General Public License [37]

Tabla 8 - Comparativa de herramientas de evaluación Servicios Web Geográficos

Si bien no hay una gran variedad de herramientas como en el área de servicios web, se encontraron herramientas interesantes en el área de evaluación de servicios web geográficos que pueden ser de gran utilidad para el monitoreo, validación y verificación de cumplimiento de estándares.

### Simulación de fallos

También se encontraron herramientas de simulación de fallos que permiten crear y reproducir escenarios más complejos de analizar. La mayoría de estas herramientas utilizan objetos simulados, con valores fijos similares a los obtenidos por los programas, para reproducir situaciones de errores y evaluar cómo reaccionan las aplicaciones.

- **WireMock**<sup>42</sup>

Biblioteca flexible para la simulación de servicios de terceros. Compatible con respuestas HTTP, verificación de respuestas, interceptación, grabación, reproducción de stubs e inyección de fallos. La misma se puede utilizar dentro de pruebas unitarias o deployado en un entorno de pruebas. A pesar de estar escrito en Java, dispone de una API JSON

<sup>42</sup> <http://wiremock.org/>

para poder utilizarlo con casi cualquier lenguaje.

El proyecto está publicado bajo licencia de Apache 2.0.

- **MockServer**<sup>43</sup>

Utilizado para simular cualquier sistema, servicio o sitio web vía HTTP o HTTPS. Sus funcionalidades principales son: devolver una respuesta simulada, reenviar o verificar una solicitud cuando una petición coincide con un formato determinado. Es ideal para escenarios de pruebas donde se podrán recrear distintos tipos de respuestas y configurar fácilmente respuestas simuladas de forma independiente para cada prueba. MockServer puede ser ejecutado de distintas formas: mediante Maven, archivo war, JUnit o por línea de comandos.

El proyecto está publicado bajo licencia de Apache 2.0.

- **MockWeb Server**<sup>44</sup>

Servidor web de script de comandos para clientes HTTP. Permite la creación de simulación de respuestas a través de la copia de respuestas reales. Permite probar situaciones difíciles de reproducir como gran cantidad de errores simultáneos o respuestas de carga lenta.

El proyecto está publicado bajo licencia de Apache 2.0.

- **Mockito**<sup>45</sup>

Framework de código abierto para pruebas en Java. El mismo permite la creación y configuración de objetos mock para pruebas unitarias automáticas, estos objetos imitan el comportamiento de objetos reales de forma controlada.

El proyecto está publicado bajo licencia MIT y puede ser utilizado agregando las dependencias correspondientes a través de Maven o Gradle.

Se muestra la Tabla 9 con las principales herramientas encontradas.

Herramienta	Funcionalidades	Integración	Licencia
<b>WireMock</b>	- Simulación de servicios de terceros. - Verificación de respuestas - Inyección de fallos	Maven o Gradle	Apache 2.0 [35]
<b>MockServer</b>	- Respuesta simulada - Reenvío y verificación de solicitud	Maven, JUnit, Gradle	Apache 2.0 [35]
<b>MockWeb Server</b>	- Simulación de respuestas a través de la copia de respuestas reales.	Maven o Gradle	Apache 2.0 [35]

<sup>43</sup> <http://www.mock-server.com/>

<sup>44</sup> <https://github.com/square/okhttp/tree/master/mockwebserver>

<sup>45</sup> <http://site.mockito.org/>

	- Permite probar gran cantidad de errores simultáneos o respuestas de carga lenta.		
<b>Mockito</b>	- Creación y configuración de objetos simulados para pruebas unitarias automáticas	Maven o Gradle	MIT [38]

Tabla 9 - Comparativa de servidores de simulación de fallas

Resulta interesante la idea de realizar una simulación de falla, con el propósito de poder reproducir escenarios de error y analizar el comportamiento de una aplicación ante dichas situaciones. También se nota un gran avance en esta área y se encontró una gran variedad de herramientas que pueden ser utilizadas con este fin.

### 3.4. Resumen

En el presente capítulo se analizaron los requerimientos del sistema, describiendo su comportamiento general así como las funcionalidades específicas con las que debe contar. Además se hizo un análisis de las soluciones existentes para la evaluación de calidad de servicios geográficos

En primer lugar se realizó una descripción del marco general de trabajo del proyecto, donde se analizaron las estructuras de las IDEs, los estándares OGC, se estudiaron los modelos de calidad y los usuarios interesados en la herramienta. Del análisis de las estructuras de las IDEs se obtuvieron requerimientos y características, sobre el ingreso y administración de nuevas IDEs. Del análisis de modelos de calidad, se obtuvieron características sobre la metodología para el ingreso de los métodos de las métricas de los modelos de calidad. Del análisis de los usuarios interesados en la herramienta se obtuvieron los roles de usuarios con los que debe contar la solución.

En segundo lugar, se describieron los requerimientos funcionales y no funcionales que fueron recabados en diferentes instancias de reunión con las tutoras de proyecto y tomando como referencia la Tesis del Lic. De Los Reyes [3]. Los requerimientos también fueron obtenidos en una reunión en Presidencia que contó con miembros de AGESIC, Presidencia e IDEuy.

Por último, se realizó un relevamiento de soluciones existentes, analizando las principales herramientas y soluciones encontradas para resolver funcionalidades necesarias en la construcción de la solución. El análisis se focalizó en la resolución de dos dificultades centrales en la evaluación de los servicios web, el monitoreo de servicios web geográficos y el procesamiento de esquemas OGC. Las soluciones y herramientas encontradas fueron divididas en categorías según sus objetivos y se caracterizaron según sus funcionalidades soportadas, su licenciamiento y la compatibilidad con el proyecto.

Cabe destacar que ninguna de las herramientas y soluciones encontradas cumple con los objetivos de este proyecto, por lo que la construcción de una herramienta a medida es necesaria para poder evaluar la calidad de los servicios web geográficos, y en particular en el contexto de una IDE. Sin embargo, se encontraron herramientas de gran utilidad para procesar las respuestas obtenidas de los servicios web geográficos.

## 4. Planteo de la Solución

En este capítulo se describe el planteo de la solución y su funcionamiento, los métodos de medición, la extensibilidad del modelo de calidad, la arquitectura planteada y las decisiones tomadas.

### 4.1. Descripción general

En la Figura 13 se muestra un esquema conceptual de la solución planteada la cual resume visualmente la estructura del sistema.

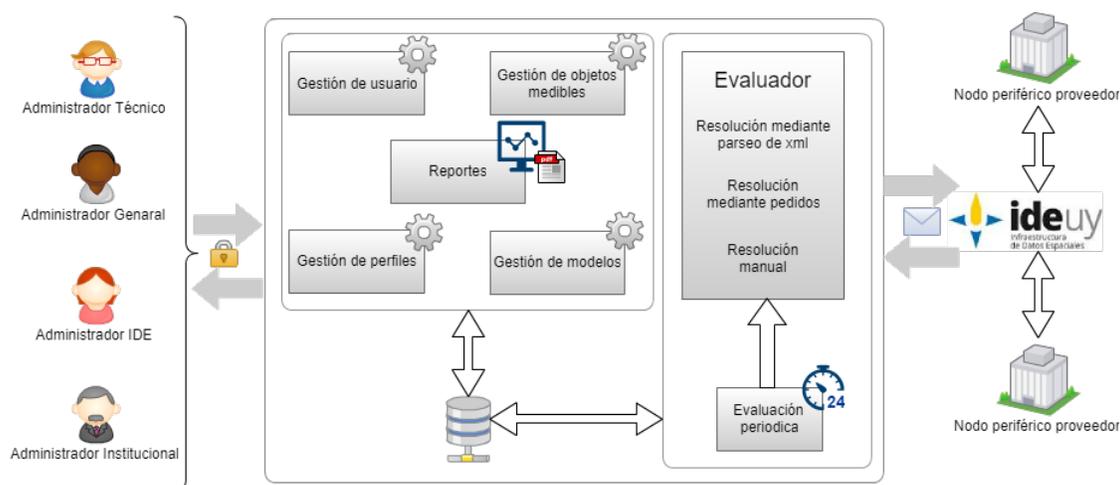


Figura 13: Esquema conceptual de la solución planteada

A la izquierda de la Figura 13, se muestran los distintos tipos de usuarios de la aplicación, los cuales inician sesión para hacer uso de las funcionalidades del sistema. En la parte central izquierda, se muestran los componentes de gestión para cada una de las entidades presentes. Luego en la parte central derecha se ve el evaluador, componente encargado de realizar las evaluaciones de las métricas y que realiza distintas tareas de acuerdo al tipo de resolución a aplicar. Debajo aparece el componente que registra las evaluaciones periódicas, que son ejecutadas cada veinticuatro horas. Estos componentes se comunican con la base de datos como lo indica la figura. Luego a la derecha se muestra la interacción con la IDEuy y sus nodos periféricos, a los cuales se les realizan pedidos sobre sus servicios web a evaluar.

El principal componente de la solución planteada consta de un evaluador de servicios web, el cual determina las tareas requeridas para la evaluación según la categoría de resolución de cada métrica a ejecutar. También realiza los pedidos a los servicios web correspondientes y examina la respuesta obtenida en busca de ciertas propiedades, aspectos o características que permitan evaluar las métricas asociadas.

Para la implementación de métricas que requieren de evaluaciones periódicas se desarrolló un módulo encargado de registrar y escuchar eventos de alarmas, planificadas para interrumpir periódicamente. Por medio de este mecanismo, al expirar el tiempo establecido se invocan los eventos registrados y se realizan las tareas de gestión de pedidos a los servicios web correspondientes, registrando los resultados obtenidos. En particular la métrica de “Disponibilidad diaria del servicio” utiliza este módulo para su evaluación y está planificada para invocarse cada 24 horas.

Toda la información obtenida de las evaluaciones realizadas es almacenada en la base de datos del sistema. Con dicha información se elaboran distintos reportes e indicadores que muestran al usuario medidas de calidad sobre los objetos medibles evaluados. En particular, el componente de reportes se encarga de exponer al usuario los resultados obtenidos con medidas e indicadores de calidad, mediante gráficas y en forma de tablas, estas últimas disponibles para ser descargadas en formato pdf.

Además se implementaron componentes encargados de gestionar las entidades de usuarios, objetos medibles, perfiles y modelos de calidad. En los puntos siguientes se explica con detenimiento aspectos y funcionalidades importantes del sistema construido.

## 4.2. Arquitectura

A continuación se describe la arquitectura del sistema construido, mostrando cómo se relacionan las partes que conforman el software.

### 4.2.1. Vista lógica

Se decidió utilizar una arquitectura en tres capas para lograr separación y desacoplamiento entre las partes que conforman el sistema. Las capas implementadas son:

- **Capa de Presentación**

La capa de presentación es la encargada de mostrar información e interactuar con el usuario a través del navegador web. Los datos ingresados son pasados a la capa de negocio.

- **Capa de Negocio**

La capa de negocio recibe la información y las peticiones de usuario que son enviadas desde la capa de presentación. Procesa los pedidos tomando decisiones lógicas y realizando cálculos. Finalmente devuelve al usuario la respuesta del proceso. Cuenta con los siguientes componentes:

- Gestión de usuarios.
- Gestión de objetos medibles.
- Gestión de modelo de calidad.
- Reportes.
- Gestión de perfiles.

- Gestión de evaluaciones.

La capa de negocio también se comunica con la capa de datos para persistir información u obtener datos de ella.

- **Capa de Datos**

La capa de datos encapsula el acceso a los datos, encargándose de persistir la información y de atender los pedidos de datos de la capa de negocio.

La división en capas también fomenta la escalabilidad y facilidad de mantenimiento en el sistema. En la Figura 14 se muestra la vista lógica con los componentes, sus interfaces y las dependencias entre ellos.

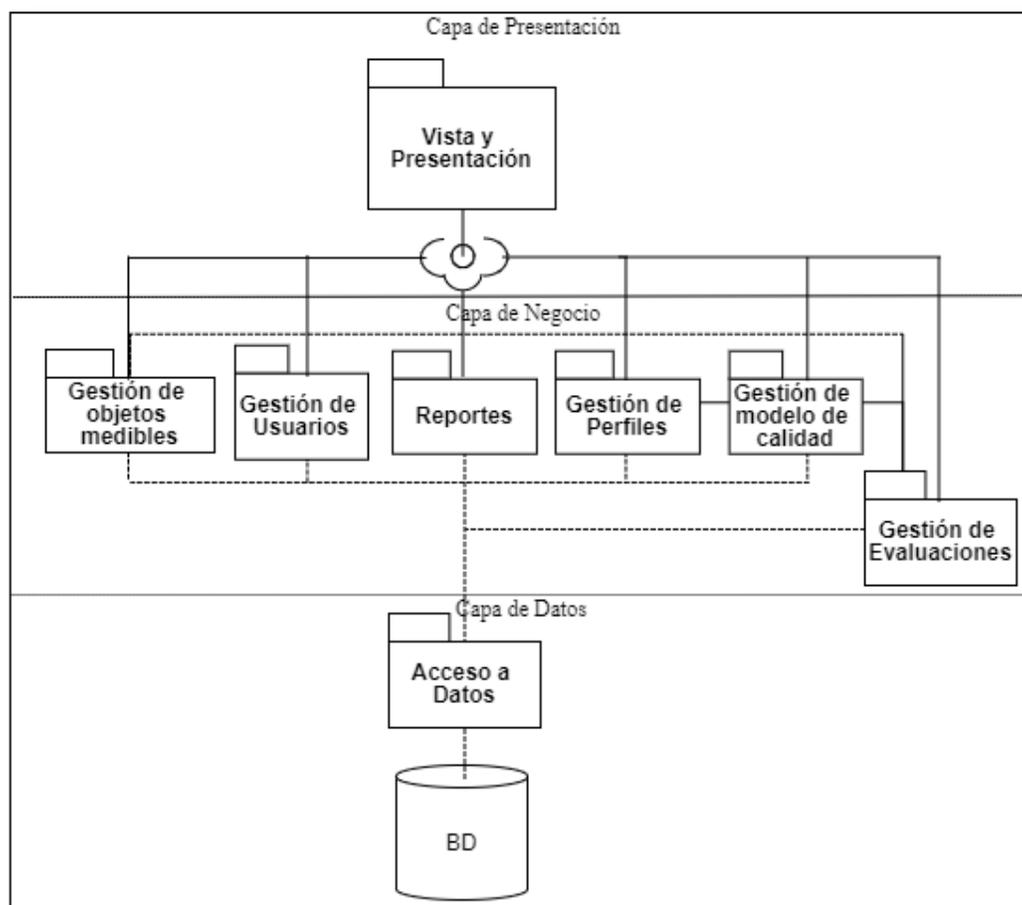


Figura 14: Vista lógica de componentes, interfaces y dependencias

#### 4.2.1.1. Componente de Gestión de Usuarios

Este componente se encarga del manejo de los usuarios del sistema, implementa los controles de autenticación y permisos, así como las funcionalidades de registro, login y logout.

Por razones de seguridad, solo el usuario Administrador Técnico cuenta con los permisos necesarios para ingresar, editar y eliminar usuarios con sus correspondientes permisos. Además se implementaron controles de autenticación por medio de nombre de usuario y contraseña y se agregaron controles de sesión en todo el sistema con redirección al login, estableciendo tiempos de expiración para la sesión de usuario.

En la Figura 15 se muestra el componente de gestión de usuarios.

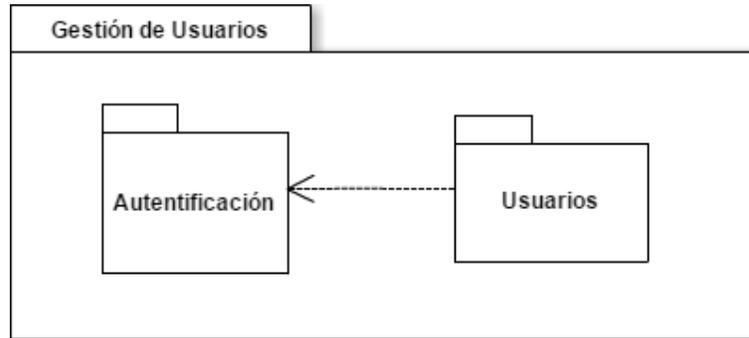


Figura 15: Componente de gestión de usuarios

#### 4.2.1.2. Componente de Gestión de Objetos Medibles

Este componente se encarga del manejo de las infraestructuras de datos espaciales y los elementos que la componen, ingresados en el sistema para su evaluación. Implementa las funcionalidades de altas, bajas y modificaciones.

En la Figura 16 se muestra el componente de gestión de objetos medibles.

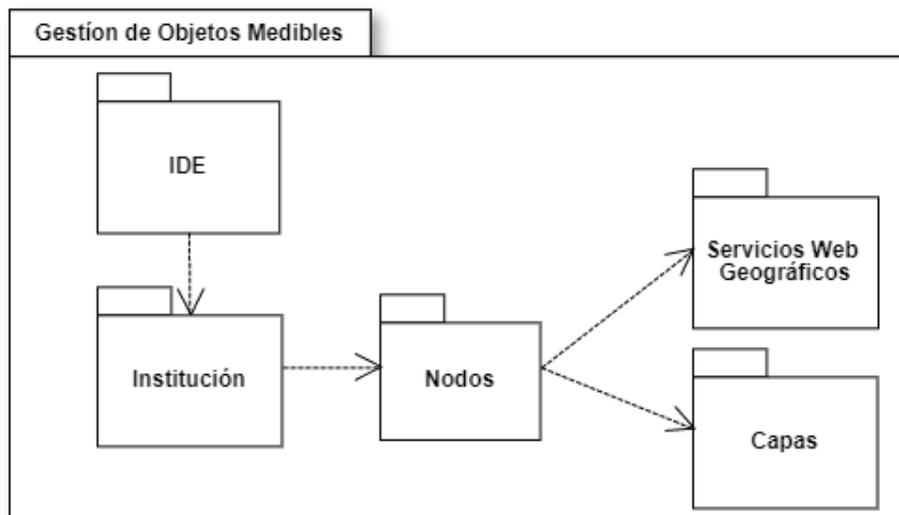


Figura 16: Componente de gestión de objetos medibles

#### 4.2.1.3. Componente de Modelo de Calidad

Este componente gestiona los elementos de los modelos de calidad existentes en el sistema, junto con los respectivos elementos (dimensiones, factores, métricas y métodos). Implementa las funcionalidades de altas, bajas y modificaciones. Es mostrado en la Figura 17.

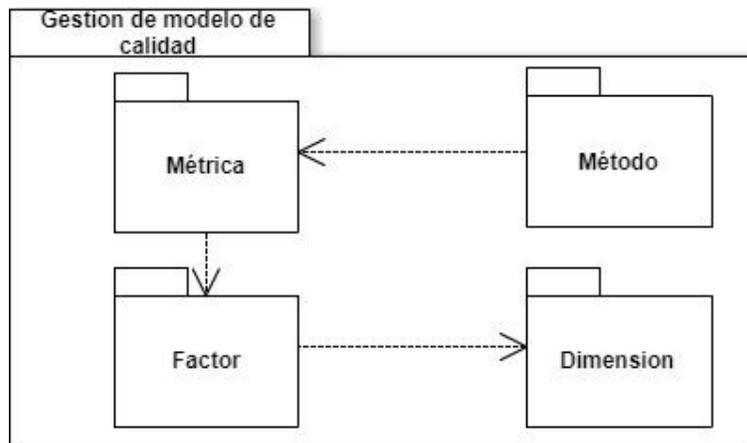


Figura 17: Componente de gestión de modelo de calidad

#### 4.2.1.4. Componente de Gestión Perfiles

Este componente gestiona los distintos perfiles existentes en el sistema, las métricas que incluye cada uno, los valores de rango dados para cada una de ellas así como las ponderaciones asignadas en caso de que el perfil sea ponderado.

En la Figura 18 se muestra el componente de gestión de perfiles.

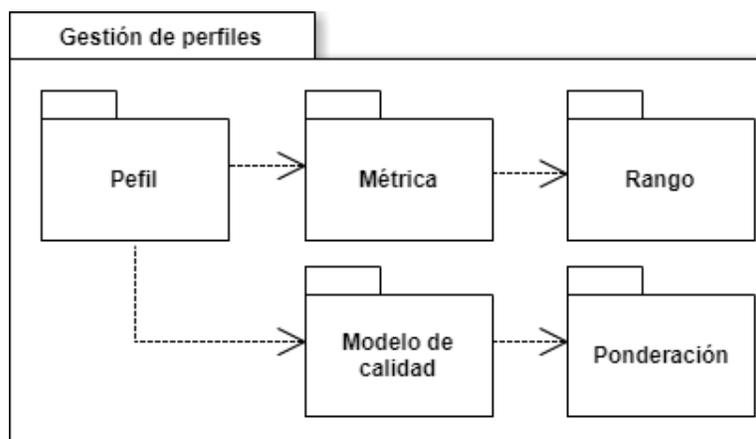


Figura 18: Componente de gestión de perfiles

Al ingresar un perfil se debe seleccionar la lista de métricas que contiene. A cada métrica se le asigna un valor de rango por defecto, el cual puede ser modificado si lo requiere el usuario. A su vez se debe indicar si el perfil es ponderado. En caso de ser ponderado, se asignan valores de ponderaciones por defecto a cada elemento del modelo de calidad. Los valores de ponderaciones de cada elemento pueden ser modificados si el usuario así lo requiere.

#### 4.2.1.5. Componente de Gestión de Evaluaciones

Este componente gestiona la evaluación de IDEs, instituciones, nodos, capas y servicios web geográficos. Evalúa las distintas dimensiones, factores y métricas con los rangos de valores y ponderaciones del perfil seleccionado. El componente también gestiona las evaluaciones que deben de ser realizadas de forma periódica.

En la Figura 19 se muestra el componente de gestión de evaluación.

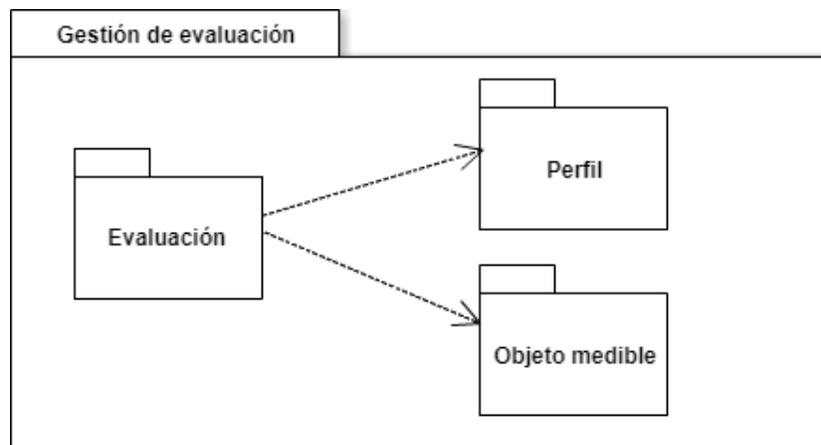


Figura 19: Componente de gestión de evaluaciones

Al momento de realizar una evaluación, se debe distinguir si se realiza con un perfil ponderado o no. Si el perfil no es ponderado entonces se verifica si el objeto medible cumple o no cumple con la métrica ejecutada. Si el perfil es ponderado entonces se realiza el cálculo del índice de calidad para el objeto medible, considerando las ponderaciones asignadas a los elementos del modelo de calidad para el perfil y el índice de calidad presentado en el modelo de referencia de la Tesis del Lic. De Los Reyes [3].

#### 4.2.1.6. Componente de Gestión de Reportes

Este componente gestiona los reportes de evaluaciones de los objetos medibles del sistema. El mismo se encarga de exponer los resultados obtenidos con medidas e indicadores de calidad, mediante gráficas y en forma de tablas, estas últimas disponibles para ser descargadas en formato pdf.

Los datos que se muestran en los reportes son obtenidos de la base de datos, al consultar el historial de evaluaciones realizadas para los objetos medibles consultados hasta el momento. Luego con dichos datos se elaboran distintos tipos de gráficos.

En la Figura 20 se muestra el componente de gestión de reportes.

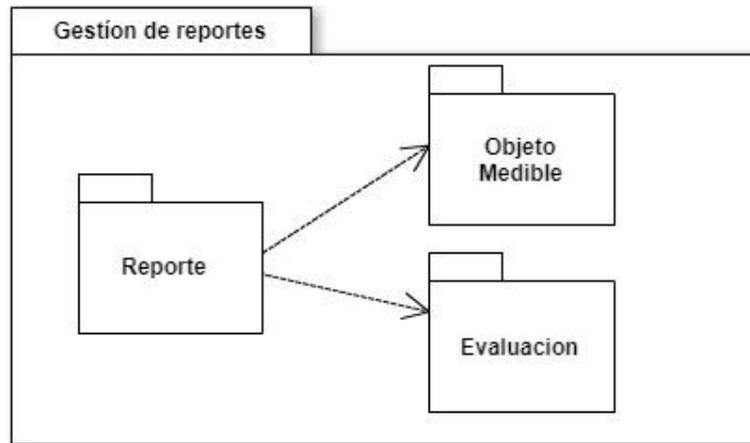


Figura 20: Componente de gestión de reportes

#### 4.2.1.7. Diagrama de secuencia de Evaluaciones

En la Figura 21 se muestra el diagrama de secuencia de la evaluación de un servicio web, el mismo contiene las interacciones entre los distintos componentes del sistema que participan en la evaluación.

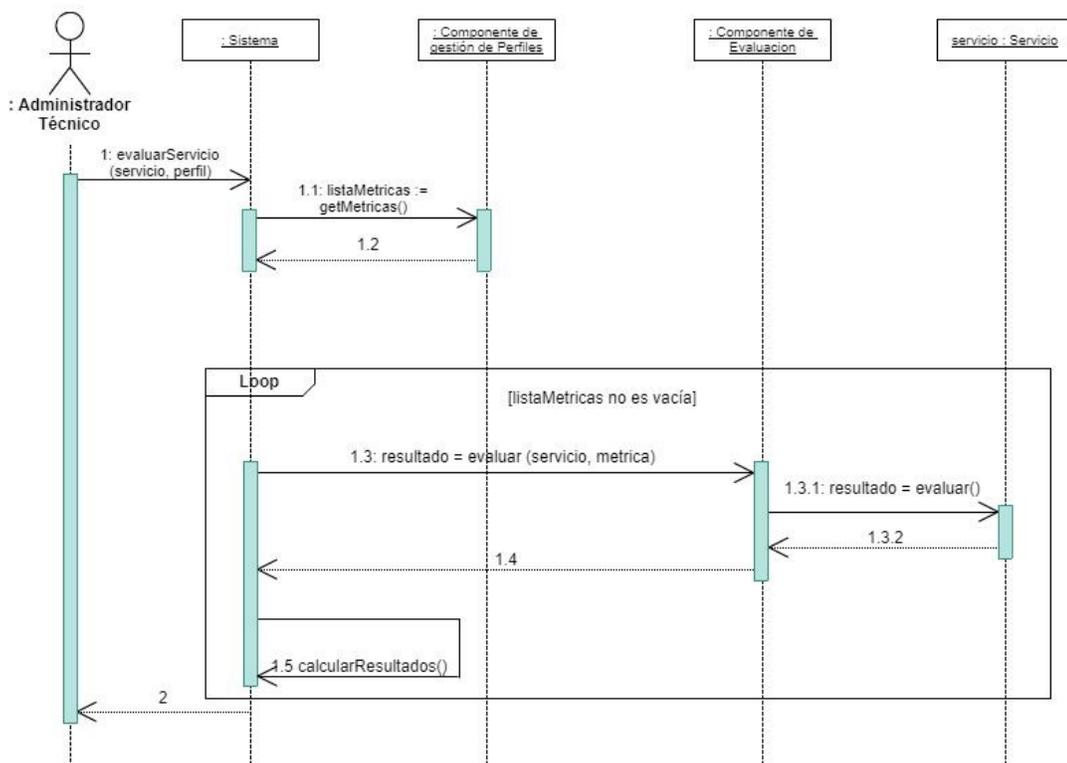


Figura 21: Diagrama de secuencia de evaluación

La secuencia para evaluar un nodo, una institución o una IDE es similar, pero con la diferencia que se debe realizar la evaluación mediante métodos de agregación, calculando

la calidad de un objeto compuesto como la agregación de los valores de calidad de partes de ese objeto.

La secuencia de operaciones que aparecen en la Figura 21 se describe a continuación:

- 1 - Evaluación del servicio  
La secuencia comienza una vez que el usuario inicia una evaluación de un servicio, para ello debe seleccionar previamente un perfil con las métricas requeridas y el servicio a evaluar. Por ejemplo, seleccionando un perfil que tenga la métrica “Información en excepciones”, y seleccionando uno de los servicios que brinda Presidencia.
- 1.1 - Listar métricas  
El componente de gestión de perfiles obtiene la colección de métricas asociadas al perfil seleccionado. Siguiendo el ejemplo, la métrica “Información en excepciones” formará parte de las métricas asociadas al perfil.

Para cada una de las métricas obtenidas en el paso anterior, se realizan las siguientes tareas:

- 1.3 - Evaluar servicio  
El sistema invoca al componente de evaluación con la métrica correspondiente. En el ejemplo, se invoca al componente de evaluación con la métrica “Información en excepciones”.
- 1.3.1 - Evaluación  
El componente de evaluación invoca al servicio, obtiene el método que implementa la métrica a evaluar y persiste el resultado obtenido. Siguiendo el ejemplo, se ejecuta el método asociado a la métrica “Información en excepciones” para evaluar el servicio seleccionado de Presidencia y se guarda el resultado obtenido.
- 1.5 - Calcular resultados  
Se calcula el resultado total de la evaluación con los resultados obtenidos anteriormente. Finalmente para el ejemplo propuesto, se verifica si el servicio cumple con la métrica y se prosigue a evaluar el resto de las métricas del perfil.

### 4.3. Métodos de medición

Para implementar los métodos de medición se realizó un análisis de las métricas planteadas en el modelo de calidad, identificando los pasos para su resolución y clasificando las métricas en distintas categorías según su metodología de medición. Para determinar dichas categorías se consideran los aspectos que se describen a continuación.

Muchas de las métricas analizadas requieren evaluar parámetros de aceptación, operaciones soportadas o listar las capas disponibles de los servicios WMS o WFS, siendo necesario consultar los atributos del método getCapabilities. Este es el caso de las métricas: Información en excepciones, Excepciones en formato OGC y Capas del Servicio con CRS Adecuado, entre otras. A su vez se debe tener en cuenta que la estructura y los

nombres de los atributos de la respuesta pueden ser distintos según la versión del servicio web disponible.

Otras métricas requieren determinar aspectos sobre la operativa de los servicios a evaluar, como el rendimiento o disponibilidad del servicio cada cierto período de tiempo, verificando que la respuesta a una solicitud realizada a un servicio sea obtenida en un tiempo de respuesta adecuado. Este es el caso de las métricas: Disponibilidad diaria del servicio, Promedio Tiempo de Respuesta Diario, entre otras.

Además ciertas métricas requieren consultar directamente al usuario sobre su cumplimiento, ya que no pueden ser resueltas de forma automática. Este tipo de evaluación se considera de resolución manual, y debe ser efectuada por el usuario en el momento. Este es el caso de las métricas: Errores Descriptivos, Documentación, entre otras.

Considerando estos aspectos se determinaron tres categorías para los métodos de medición:

- **CAT\_GET\_CAPABILITIES**  
Esta categoría implica que la resolución de la métrica es efectuada mediante el procesamiento de la operación getCapabilities para distintos métodos (p. ej. GetMap).
- **CAT\_PEDIDOS\_SERVICIO**  
Esta categoría implica que la resolución de la métrica es efectuada mediante la realización de pedidos a los servicios y verificación de su respuesta.
- **CAT\_RESPUESTA\_MANUAL**  
Esta categoría implica que la resolución de la métrica es efectuada manualmente, por parte de los usuarios.

Debido a la gran cantidad de métricas contenidas en el modelo de calidad y considerando el tiempo disponible para la ejecución del proyecto se realizó una selección de las métricas a implementar según su relevancia, categoría, factores y dimensiones. Las métricas fueron seleccionadas de modo de contar con al menos una de cada categoría de resolución.

La Tabla 10 contiene las métricas implementadas, junto a la clasificación según su método de resolución.

Dimensión	Factor	Métrica	Categoría
Seguridad	Protección	Información en excepciones	CAT_GET_CAPABILITIES
Confiabilidad	Disponibilidad	Disponibilidad diaria del servicio	CAT_PEDIDOS_SERVICIO
Interoperabilidad	Soporte de estándares	Excepciones en formato OGC	CAT_GET_CAPABILITIES

	Sistema de referencias	Capas del Servicio con CRS Adecuado (IDEuy)	CAT_GET_CAPABILITIES
		Capa con CRS Adecuado (IDEuy)	CAT_GET_CAPABILITIES
Publicación de Datos	Formatos soportados	Formato PNG	CAT_GET_CAPABILITIES
		Formato KML	CAT_GET_CAPABILITIES
		Formato text/html método getFeatureInfo	CAT_GET_CAPABILITIES
		Formato Excepción application/vnd.ogc.se_image	CAT_GET_CAPABILITIES
		Excepción application/vnd.ogc.se_blank	CAT_GET_CAPABILITIES
		Cantidad de formatos soportados	CAT_GET_CAPABILITIES
		Cantidad de formatos de excepciones soportadas	CAT_GET_CAPABILITIES
Metadatos	Metadatos Servicio	Leyenda de la Capa	CAT_GET_CAPABILITIES
		Especifica Rango Útil	CAT_GET_CAPABILITIES
Usabilidad	Facilidad de aprendizaje	Errores descriptivos	CAT_RESPUESTA_MANUAL

Tabla 10 - Métricas implementadas y su clasificación según el método de resolución

#### 4.4. Extensibilidad de modelo

Como se estableció anteriormente en los requerimientos no funcionales, el modelo de calidad aplicado en la solución debe ser extensible, y si bien se parte de un modelo de calidad de referencia, el sistema debe permitir modificarlo así como ingresar nuevos modelos, con sus respectivos métodos de evaluación.

Por tanto la extensibilidad del modelo es lograda mediante el ingreso de nuevos modelos de calidad a la solución o extendiendo el modelo existente. A continuación se analizan algunos aspectos centrales del proceso.

Para representar los modelos de calidad de forma uniforme y contemplar todos sus elementos, se toma la organización jerárquica definida en el metamodelo presentado en la

Tesis del Lic. De Los Reyes [3]: Dimensiones, Factores, Métricas y Métodos, como estructura base para representar todos los modelos.

El ingreso de la estructura del modelo de calidad, se realiza mediante el ingreso de todos sus componentes en forma jerárquica de mayor a menor nivel. De esta forma se asegura que todos los elementos respeten la organización del modelo.

El ingreso de los métodos de implementación se realiza mediante la carga dinámica de los métodos asociados a las métricas de medición ingresadas por los usuarios. De esta forma los usuarios pueden crear métricas adjuntando sus propios métodos de evaluación sin la necesidad de recompilar la herramienta.

## 4.5. Decisiones tomadas

Durante la realización del proyecto y en diferentes instancias se tomaron ciertas decisiones claves en la construcción de la herramienta, las cuales son mencionadas a continuación.

Para realizar mediciones sobre la operativa de los servicios evitando la sobrecarga de pedidos, se manejó la posibilidad de guardar registros en los servidores y luego enviar la información a la herramienta mediante la instalación de un plugin. Sin embargo, esta opción requiere la instalación en cada uno los servidores de las distintas organizaciones, lo cual compromete la independencia de la solución para realizar las evaluaciones. Por lo cual se decidió realizar las mediciones desde el lado de la solución e incluir controles para evitar la sobrecarga de pedidos en los servidores.

Luego de clasificar las métricas, se eligieron las que resultaron más relevantes para el propósito del proyecto y se optó por implementar al menos una métrica de cada categoría para facilitar el mantenimiento y ampliación del modelo a futuro. Las métricas seleccionadas son las definidas en la Tabla 10.

Para dar soporte a la creación y ampliación del modelo de calidad, se decidió implementar el ingreso de las métricas mediante la inclusión de un archivo JAR que será validado en el momento y cargado en tiempo de ejecución. De esta forma la implementación de una nueva métrica se puede añadir al sistema con facilidad, y el desarrollador debe concentrarse únicamente en implementar el método de forma correcta y no de cómo integrarlo a la solución.



## 5. Implementación de la Solución

En este capítulo se especifican detalles de implementación de la solución planteada. Se comienza presentando las herramientas utilizadas en la implementación. Luego se comentan detalles acerca de la extensibilidad del modelo y se explica cómo se realizó el procesamiento de esquemas XML. Seguidamente se describen los principales componentes de interfaz gráfica utilizados y se comentan detalles de la publicación del sistema. Finalmente se describen los problemas encontrados a lo largo de la implementación de la solución.

### 5.1. Herramientas utilizadas

El sistema fue implementado en el lenguaje de programación Java<sup>46</sup>, como se especificó en los requerimientos funcionales planteados anteriormente. Las herramientas fueron seleccionadas por su compatibilidad, así como también por el conocimiento y experiencias previas adquiridas por los integrantes del grupo. La Figura 22 reúne las herramientas utilizadas.

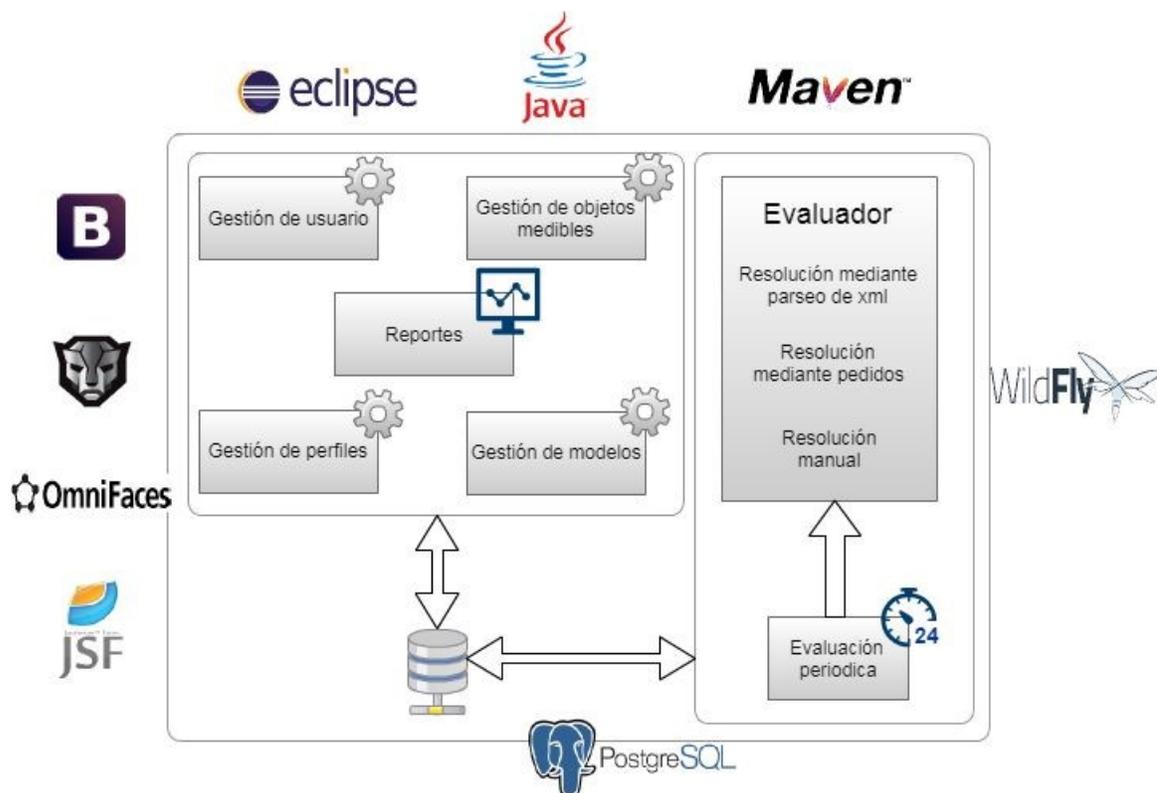


Figura 22: Herramientas utilizadas en la implementación de la solución

<sup>46</sup> <https://www.java.com/es/>

En la Tabla 11 se describen las herramientas utilizadas para el desarrollo del sistema.

Herramienta	Descripción
Bootstrap <sup>47</sup> 	<p>Es uno de los frameworks más utilizados para el desarrollo web y es un proyecto de código abierto. Creado originalmente por un diseñador y un desarrollador de la empresa Twitter.</p> <p>Bootstrap fue utilizado como framework para el desarrollo web debido a las facilidades que brinda para el diseño web y la posibilidad de aplicar estilos predeterminados.</p>
Apache Maven <sup>48</sup> 	<p>Maven es un software de administración de proyectos, que utiliza archivos de configuración de tipo Project Object Model (POM) y permite manejar la compilación del proyecto, dependencias y librerías. A su vez permite generar documentación y pruebas unitarias automáticas.</p> <p>Se utilizó Apache Maven 3.5.0 como gestor de proyectos y dependencias, para el manejo y descarga dinámica de librerías y plugins de distintos repositorios.</p>
Eclipse Neon <sup>49</sup> 	<p>Eclipse es uno de los entornos de desarrollo de programación más utilizados. Se basa en espacios de trabajos y cuenta con un sistema de plugins para personalizar el entorno. Eclipse está escrito principalmente en Java y su uso principal es para desarrollar aplicaciones Java, pero también puede ser usado para desarrollar aplicaciones en otros lenguajes de programación.</p> <p>Se utilizó el entorno de desarrollo Eclipse Neon 4.6.0, el mismo cuenta con soporte para Java EE, Maven, JSF y Git Team Provider, entre otros.</p>
PostgreSQL <sup>50</sup> 	<p>PostgreSQL es un sistema de base de datos relacional de código abierto, cuenta con más de 15 años de desarrollo activo y una arquitectura sólida que le brinda fiabilidad e integridad de datos.</p> <p>Se utilizó PostgreSQL 9.6 como motor de base de datos, debido a que es posible añadir soporte a base de datos geográficas.</p>
Wildfly <sup>51</sup> 	<p>Wildfly es un servidor de aplicaciones de código abierto, creado por JBoss y luego adquirido y desarrollado por Red Hat. Está escrito en Java e implementa la especificación Java Platform Enterprise Edition (Java EE), a su vez ofrece una plataforma de alto rendimiento para</p>

<sup>47</sup> <http://getbootstrap.com/>

<sup>48</sup> <https://maven.apache.org/>

<sup>49</sup> <http://www.eclipse.org/>

<sup>50</sup> <https://www.postgresql.org/>

<sup>51</sup> <http://wildfly.org/>

	<p>aplicaciones empresariales. Puede ser administrado mediante línea de comandos o a través de una consola web.</p> <p>Se utilizó Wildfly 9.0 como servidor de aplicaciones, debido al soporte de Enterprise JavaBeans<sup>52</sup> (EJB). EJB es la arquitectura de componentes del servidor para Java EE, permite un desarrollo rápido y simplificado de aplicaciones distribuidas, transaccionales, seguras y portátiles basadas en la tecnología Java.</p>
<p>JSF<sup>53</sup></p> 	<p>La tecnología Java Server Faces establece un estándar para el desarrollo de interfaces de usuario del lado del servidor. A su vez brinda facilidades para representar los componentes de la interfaz de usuario manteniendo las capas de interfaz y lógica de negocios bien definidas y separadas. También permite otras funcionalidades como manejo de beans, navegación entre páginas, y definición de eventos sobre la interfaz.</p> <p>Se utilizó el estándar JSF para desarrollar interfaces de usuario de los componentes web, manejo y validación de beans.</p>
<p>PrimeFaces<sup>54</sup></p> 	<p>Es una biblioteca de código abierto de componentes de interfaz de usuario para JSF. Su configuración es sencilla ya que simplemente consiste en agregar un único archivo jar como dependencia.</p> <p>Se utilizó la biblioteca de Primefaces por sus componentes visuales, ya que resultan útiles y fáciles de emplear. En particular se utilizaron los componentes de gráficos personalizados y tabla de datos en forma de árbol.</p>
<p>OmniFaces<sup>55</sup></p> 	<p>Es una librería utilitaria para JSF 2 que se centra en utilidades que facilitan el uso de JSP estándar y ofrece una solución a problemas recurrentes frecuentemente encontrados en el desarrollo con JSF. Además incluye varios convertidores, validadores de datos y manejadores de Facelets.</p> <p>OmniFaces fue utilizado debido a las facilidades que presta al trabajar con JSF, en particular en el uso de conversores<sup>56</sup> de datos.</p>
<p>Bitbucket<sup>57</sup></p> 	<p>Bitbucket es un repositorio para el control de versiones distribuido basado en Git y también compatible con Mercurial. Entre sus principales ventajas cuenta con repositorios públicos y privados ilimitados y no requiere pasos adicionales de configuración.</p>

<sup>52</sup> <http://www.oracle.com/technetwork/java/javaee/ejb/index.html>

<sup>53</sup> <http://www.oracle.com/technetwork/java/javaee/javaserverfaces-139869.html>

<sup>54</sup> <https://www.primefaces.org/>

<sup>55</sup> <http://omnifaces.org/>

<sup>56</sup> <http://showcase.omnifaces.org/converters/SelectItemsConverter>

<sup>57</sup> <https://bitbucket.org/product>

	Se utilizó Bitbucket debido a que facilita la colaboración en equipo y cuenta con repositorios privados ilimitados, además tiene versión gratuita para equipos pequeños de hasta cinco personas.
<p>Trello<sup>58</sup></p> 	<p>Trello es una herramienta colaborativa y flexible para la gestión de proyectos. Permite la administración de tareas para trabajar en equipo y utiliza un sistema basado en carteleras para la organización de información que ayuda a mantener altos niveles de producción.</p> <p>Se utilizó Trello para el seguimiento de errores, asignación de tareas y notificaciones de eventos, utilizando listados para categorizar las tareas según su importancia y nivel de avance, junto con etiquetas y referencias para ampliar información.</p>

Tabla 11 - Herramientas utilizadas

## 5.2. Carga dinámica de métodos de evaluación

La extensibilidad del modelo se logra mediante la carga dinámica de los métodos asociados a las métricas ingresadas por los usuarios. La carga se realiza en tiempo de ejecución por medio de la API Java Reflection<sup>59</sup>, que provee mecanismos para acceder e invocar métodos de objetos Java de manera segura. De esta forma los usuarios pueden crear métricas adjuntando sus propios métodos de evaluación sin la necesidad de recompilar la herramienta.

Cuando un usuario crea una nueva métrica usando la herramienta construida, debe adjuntar un archivo Java estándar (JAR) con la implementación del método de medición, el cual debe cumplir con el siguiente formato:

- Nombre de paquete  
El nombre del paquete debe ser: *"UserMetricPackage"*.
- Nombre de clase  
El paquete debe contener una clase de nombre: *"UserMetricClass"*.
- Firma de método  
La clase contenida en el paquete, debe tener un método de firma: *"public boolean userMetricMethod(String param1, String param2)"*.

Si el método de medición cumple con el formato especificado entonces el JAR es guardado en una carpeta local del servidor y se lo asocia a la métrica ingresada. Luego cuando un

<sup>58</sup> <https://trello.com/>

<sup>59</sup> <https://docs.oracle.com/javase/tutorial/reflect/>

usuario solicita la evaluación de dicha métrica, la librería se carga dinámicamente y se ejecuta el método definido para su evaluación.

Cuando un archivo JAR es asociado a una métrica satisfactoriamente, el archivo JAR que es guardado en el servidor conserva su nombre original concatenando además la fecha y hora de subida del archivo. De esta forma se logra evitar el reemplazo de archivos con un mismo nombre, dado que podría ocurrir el caso en que distintos usuarios nombren de igual forma un archivo JAR para crear métricas totalmente diferentes.

### 5.3. Procesamiento de esquemas XML

Cuando se realiza una evaluación de un objeto medible, se debe realizar el pedido a los servicios web correspondientes y si la métrica a evaluar lo requiere, entonces se debe procesar las respuestas obtenidas. Para ambas tareas (pedido a los servicios web y procesamiento de las respuestas obtenidas) se utiliza la librería OGC Schemas and Tools. Las respuestas obtenidas en formato XML son procesadas mediante Java Architecture for XML Binding (JAXB), en busca de ciertas etiquetas y valores dependiendo de la métrica ejecutada.

Por ejemplo, si se intenta evaluar un servicio web geográfico con la métrica “Cantidad de formatos de excepciones soportadas” se realizará los siguientes pasos:

- Primero se obtiene la clasificación de la métrica para saber cómo debe ser resuelta. En este caso es clasificada como CAT\_GET\_CAPABILITIES, cuya resolución es efectuada mediante el procesamiento de la operación getCapabilities
- Luego se realiza el pedido al servicio web geográfico.
- Con la respuesta obtenida se realiza el procesamiento del XML en busca de las etiquetas “*Exception*” y “*Format*” que indican los tipos de formatos de excepciones soportadas por el servicio web geográfico.
- Finalmente los valores obtenidos de procesar el XML son utilizados para compararlos contra los valores esperados para la métrica y perfil de evaluación.

En la Figura 23 se muestra las secciones buscadas al procesar el XML.

```

▼<WMS_Capabilities xmlns="http://www.opengis.net/wms"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:esri_wms="http://www.esri.com/wms"
  version="1.3.0" xsi:schemaLocation="http://www.opengis.net/wms
  http://schemas.opengis.net/wms/1.3.0/capabilities_1_3_0.xsd http://www.esri.com/wms
  http://gissrv.unasev.gub.uy/arcgis/services/UNASEV/srvUNASEVWFS/MapServer/WmsServer?
  version=1.3.0%26service=WMS%26request=GetSchemaExtension">
  ▶<Service>...</Service>
  ▼<Capability>
    ▶<Request>...</Request>
    ▼<Exception>
      <Format>application/vnd.ogc.se_xml</Format>
      <Format>application/vnd.ogc.se_inimage</Format>
      <Format>application/vnd.ogc.se_blank</Format>
      <Format>text/xml</Format>
      <Format>XML</Format>
    </Exception>
    ▶<Layer>...</Layer>
  </Capability>
</WMS_Capabilities>

```

Figura 23: Ejemplo de GetCapabilities con sección de formatos de excepciones soportados

## 5.4. Componentes de interfaz de usuario

En esta sección se describen los principales componentes de interfaz de usuario utilizados. La sección presenta ejemplos de gráficos y reportes, describe cómo se implementó la generación de PDFs y el componente utilizado para mostrar la estructura jerárquica de objetos medibles.

### 5.4.1. Gráficos

Los gráficos que se muestran en la herramienta, en la sección de Reportes y en el Home, son creados con datos obtenidos de la base de datos, al consultar el historial de evaluaciones realizadas para los objetos medibles consultados hasta el momento. Los gráficos son creados usando librerías desarrolladas por PrimeFaces, en particular se utilizaron gráficos de barras<sup>60</sup> y gráficos de torta<sup>61</sup>. En la Figura 24 se muestra un ejemplo de un gráfico de barras, obtenido de la herramienta construida.

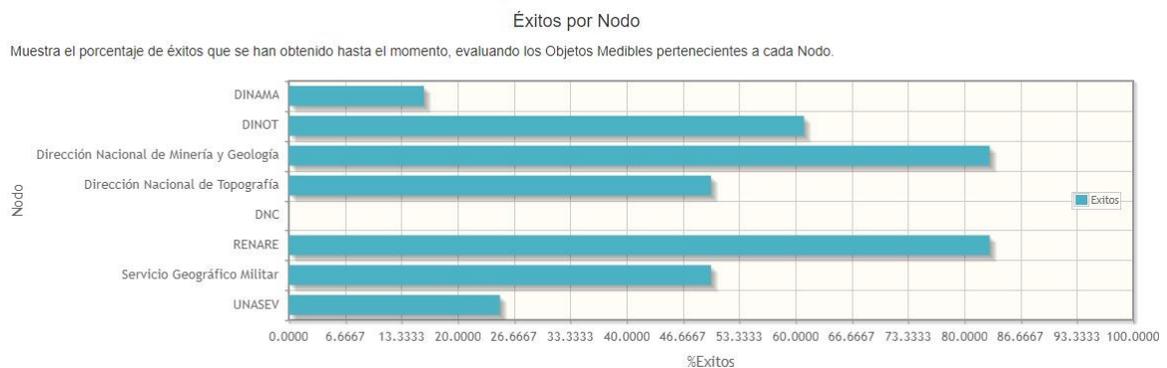


Figura 24: Ejemplo de gráfico de barras generado por la solución

<sup>60</sup> <https://www.primefaces.org/showcase/ui/chart/bar.xhtml>

<sup>61</sup> <https://www.primefaces.org/showcase/ui/chart/pie.xhtml>

### 5.4.2. Generación de archivos PDF

Las tablas de datos que se encuentran en las secciones de Reportes y Home de la herramienta construida, tienen la opción de ser descargadas en formato PDF. Para el armado de PDF sobre las tablas de datos se utilizó el exportador de datos<sup>62</sup> de PrimeFaces y la librería iText<sup>63</sup> que permite la creación y manipulación de archivos en formato PDF. La versión utilizada es la itext 2.1.7. En la Figura 25 se muestra a modo de ejemplo, una tabla de datos y su PDF correspondiente, conteniendo el total de evaluaciones realizadas por métrica.

Total de evaluaciones realizadas por Métrica.

Métrica	#Evaluaciones	Métrica	#Evaluaciones
Información en excepciones	91	Información en excepciones	91
Excepciones en formato OGC	91	Excepciones en formato OGC	91
Cantidad de formatos soportados	86	Cantidad de formatos soportados	86
Capas del servicio con CRS adecuado (IDEuy)	86	Capas del servicio con CRS adecuado (IDEuy)	86
Cantidad de formatos de excepciones soportadas	86	Cantidad de formatos de excepciones soportadas	86
Disponibilidad diaria del servicio	13	Disponibilidad diaria del servicio	13
Específica Rango Util	4	Específica Rango Util	4
Capa con CRS Adecuado (IDEuy)	4	Capa con CRS Adecuado (IDEuy)	4

Download

Figura 25: Ejemplo de tabla de datos y su PDF correspondiente

Se muestra a la izquierda, la tabla de datos con el total de evaluaciones realizadas por métrica, en un momento dado. Luego a la derecha se muestra el PDF generado a partir de la tabla de datos mostrada a la izquierda.

### 5.4.3. Estructura de árbol

Se buscaron componentes de interfaz de usuario que pudieran mostrar la estructura jerárquica de los objetos medibles de forma clara, pudiendo además cargar los datos contenidos a medida que son seleccionados. Finalmente se optó por una combinación de un componente selector en forma de árbol para los objetos medibles de mayor jerarquía (IDE, institución y nodo), y dos tablas para mostrar sus datos, una para el objeto seleccionado y otra para los objetos de menor jerarquía (servicios web y capas).

En la Figura 26 se muestra la representación de objetos medibles en forma de árbol y las tablas con sus datos.

<sup>62</sup> <https://www.primefaces.org/showcase/ui/data/dataexporter/basic.xhtml>

<sup>63</sup> <http://itextpdf.com/>

## Árbol de objetos medibles    Objeto seleccionado

The screenshot shows the IDE tool interface. On the left is a tree view of measurable objects under 'ide.uy'. The 'Presidencia' object is selected. On the right is a table with the following data:

ID	Nombre	Descripción	Acciones
97	Presidencia	Presidencia de la República	

Below the table is a section titled 'Servicios/Capas' with a table showing details for selected items:

ID	Descripción	Tipo	Tipo de Servicio	URL	Acciones
48	Siniestros Fatales	Servicio	WFS	Link al Servicio	
81		Capa	WMS	Link al Servicio	
107	Siniestros Fatales	Servicio	WMS	Link al Servicio	

At the bottom of the 'Servicios/Capas' section are buttons for '+ Nuevo' and 'Eliminar'.

Figura 26: Árbol de objetos medibles

## 5.5. Publicación del sistema

El sistema fue publicado en la nube en la plataforma Microsoft Azure<sup>64</sup>. Con tal motivo se creó una máquina virtual con Windows Server 2008 R2 Datacenter. En la máquina virtual se instalaron los componentes necesarios, se configuró el servidor de aplicaciones y se abrieron puertos para acceder al sistema desde fuera de la máquina virtual.

## 5.6. Problemas encontrados

Durante el desarrollo de la solución surgieron algunas dificultades relacionadas con la evaluación de los servicios geográficos y luego con la publicación de la solución. A continuación se describen los principales problemas encontrados:

- Distintos formatos según versiones del servicio  
Se debió considerar que la información devuelta por los servicios web consumidos presentan diferencias de acuerdo a la versión del servicio consultado. Este problema tiene la particularidad de que cada versión puede presentar diferencias en el formato y estructuración de la información devuelta, así como no contar con ciertos datos para versiones más antiguas. En consecuencia se realizó la carga previa de los servicios de la IDEUy tomando la última versión disponible para sacar mayor provecho de la evaluación y se agregaron controles en el ingreso de los servicios.
- Tipo de datos incorrectos  
Algunos métodos de los servicios evaluados pertenecientes a las IDEs contienen valores que no son los esperados según la especificación del estándar, lo cual provocó errores

<sup>64</sup> <https://azure.microsoft.com>

con la librería utilizada para el procesamiento de la información. En consecuencia, se debieron agregar controles adicionales sobre el tipo de datos en los métodos accedidos.

- **Publicación del sistema**

Se encontraron varias dificultades al publicar la solución, debiendo probar con distintos productos y plataformas como ser Heroku<sup>65</sup>, OpenShift<sup>66</sup> y Firebase<sup>67</sup>. La publicación en estas plataformas resultó dificultosa debido principalmente a problemas de incompatibilidad con la versión utilizada del servidor de aplicaciones WildFly, así como con la versión de Java y el sistema de base de datos PostgreSQL. En última instancia se optó por la plataforma Microsoft Azure. Particularmente se debió usar una versión paga de Microsoft Azure debido a las limitaciones que ofrece la cuenta de estudiantes.

---

<sup>65</sup> <https://www.heroku.com/>

<sup>66</sup> <https://www.openshift.com/>

<sup>67</sup> <https://firebase.google.com/>



## 6. Caso de Estudio y Pruebas

En este capítulo se describe la elaboración del caso de estudio, así como las pruebas de la solución en las distintas etapas del proyecto.

### 6.1. Caso de estudio

La elaboración de un caso de estudio específico permitió validar el funcionamiento de la herramienta y solución planteada. Desarrollar el caso de estudio implicó generar la estructura en la herramienta construida que permite evaluar la calidad de los servicios de la infraestructura de datos espaciales del Uruguay (IDEuy<sup>68</sup>). Se generó un juego de datos para los siguientes elementos:

- **Objetos medibles**

Los objetos medibles cargados para la elaboración del caso de uso son todos los servicios, capas, nodos e instituciones que forman la IDEuy. Los datos cargados fueron obtenidos del portal de la IDEuy, accediendo a cada uno de sus nodos publicados. En la Figura 27 se muestra una vista del árbol de objetos medibles del caso de estudio y algunos de los servicios que contiene la IDEuy.

Árbol de objetos medibles      Objeto seleccionado

ID	Nombre	Descripción	Acciones
68	Nodo DINAMIGE	Dirección Nacional de Minería y Geología	

Servicios/Capas

ID	Descripción	Tipo	Tipo de Servicio	URL	Acciones
20	Hidrogeología	Servicio	WMS	Link al Servicio	
59	Mapa Geológico	Servicio	WFS	Link al Servicio	
85	Distritos Mineros	Servicio	WFS	Link al Servicio	
101	Laboratorio	Servicio	WMS	Link al Servicio	
106	Distritos Mineros	Servicio	WMS	Link al Servicio	
119	Catastro Minero	Servicio	WFS	Link al Servicio	
125	Mapa Geológico	Servicio	WMS	Link al Servicio	
133	Catastro Minero	Servicio	WMS	Link al Servicio	

Figura 27: Se muestran los objetos medibles en la herramienta

<sup>68</sup> <http://ide.uy/>

- Usuarios

Se crearon usuarios con distintos roles. En particular se crearon usuarios de tipo Administrador técnico que pueden acceder a todas las funcionalidades de la herramienta sin restricción alguna. También se crearon usuarios de tipo Administrador general, los cuales pueden evaluar cualquier objeto medible disponible en la herramienta. Además se crearon usuario de tipo Administrador IDE, que puedan evaluar toda la IDEuy y finalmente se crearon usuarios de tipo Administrador institucional que puede evaluar únicamente una institución en particular. En la Figura 28 se muestra una vista de alguno de los usuarios creados para el caso de estudio.

### Datos de Usuarios

ID	Institución	Email	Tipo	Nombre	Apellido	Telefono	Acciones
13	Ministerio de Vivienda, Ordenamiento Territorial y Medio Ambiente	lsimaldone@mail.com	Administrador General	Luciano	Simaldone	98715432	
14	Ministerio de Defensa Nacional	fgonzalez@mail.com	Administrador Tecnico	Federico	Gonzalez	98876654	
16	Ministerio de Economía y Finanzas	aperez@mail.com	Administrador IDE	Augusto	Perez	99565543	
17	MTOP	dsuarez@mail.com	Administrador Institucional	Diego	Suarez	98789564	

Figura 28: Vista de la herramienta, mostrando usuarios creados

- Modelo de calidad

El modelo de calidad cargado contiene todas las dimensiones, factores y métricas que fueron desarrolladas y que se listan en la Tabla 10 del Capítulo 4. En la Figura 29 se muestra la estructura del modelo de calidad cargado en la herramienta construida.

Árbol de modelos de calidad			
Nombre	Granularidad	Unidad	
▼ ModeloIDEuy			/
▼ Publicación de Datos			/
▼ Formatos soportados			/
Formato Excepcion application/vnd.ogc.se	Método	Boleano	/
Formato PNG	Método	Boleano	/
Formato KML	Método	Boleano	/
Formato Excepcion application/vnd.ogc.se	Método	Boleano	/
Cantidad de formatos de excepciones sop	Servicio	Entero	/
Formato text/html metodo getFeatureInfo	Método	Boleano	/
Cantidad de formatos soportados	Servicio	Entero	/
▼ Usabilidad			/
▼ Facilidad de aprendizaje			/
Errores descriptivos	Servicio	Boleano	/
▼ Seguridad			/
▼ Protección			/
Información en excepciones	Servicio	Boleano	/
▼ Interoperabilidad			/
▼ Sistema de referencias			/
Capas del servicio con CRS adecuado (ID	Servicio	Porcentaje	/
Capa con CRS Adecuado (IDEuy)	Capa	Boleano	/
▼ Soporte de estándares			/
Excepciones en formato OGC	Servicio	Boleano	/
▼ Metadatos			/
▼ Metadatos Servicio			/
Específica Rango Util	Capa	Boleano	/
Leyenda de la Capa	Servicio	Boleano	/
▼ Confiabilidad			/
▼ Disponibilidad			/
Disponibilidad diaria del servicio	Servicio	Porcentaje	/

Figura 29: Vista de la herramienta, mostrando el modelo de calidad

- **Perfiles de evaluación**

Seis perfiles de evaluación fueron creados, cada uno con distinta granularidad o conjunto de métricas, con el fin de tener varias posibilidades al momento de evaluar. La Figura 29 muestra los perfiles creados en la herramienta construida y la lista de métricas que contiene uno de los perfiles creados.

ID	Nombre	Granularidad	Ponderado	Acciones
<input type="radio"/> 1	Perfil Servicio Básico	Servicio	false	
<input checked="" type="radio"/> 2	Perfil Servicio Avanzado	Servicio	false	
<input type="radio"/> 3	Perfil Capa Basico	Capa	false	
<input type="radio"/> 4	Perfil Nodo Basico	Nodo	false	
<input type="radio"/> 5	Perfil Institucion Basico	Institución	false	
<input type="radio"/> 6	Perfil Ide Basico	Ide	false	

+ Nuevo    Eliminar

Lista de Métricas disponibles para el Perfil seleccionado.

Métricas Booleanas.

Métrica	Unidad	Valor Booleano	Acciones
<input type="radio"/> Información en excepciones	Booleano	true	
<input type="radio"/> Excepciones en formato OGC	Booleano	true	

Métricas Enteras.

Métrica	Unidad	Valor Entero	Acciones
<input type="radio"/> Cantidad de formatos soportados	Entero	1	
<input type="radio"/> Cantidad de formatos de excepciones soportadas	Entero	1	

Métricas Enumeradas.

Métrica	Unidad	Valor Enumerado	Acciones
No hay métricas de tipo enumeradas asociadas al perfil seleccionado.			

Métricas Porcentaje.

Métrica	Unidad	Valor Porcentaje	Acciones
<input type="radio"/> Disponibilidad diaria del servicio	Porcentaje	50	
<input type="radio"/> Capas del servicio con CRS adecuado (IDEuy)	Porcentaje	20	

Figura 30: Vista de la herramienta con perfiles de evaluación

En la parte superior de la Figura 30 vemos los seis perfiles creados, estando seleccionado el perfil de nombre “Perfil Servicio Avanzado”. La granularidad indica el tipo de objeto medible que puede ser evaluado con el perfil, en este caso servicio. Luego debajo se muestra la lista de métricas que pertenecen al perfil, divididas en cuatro categorías según la unidad de medida. Por ejemplo, la métrica “Cantidad de formatos soportados” es de unidad entera y se muestra en la lista de métricas enteras.

Luego de tener la estructura del caso de estudio cargada, se procedió a la evaluación de los objetos medibles con los perfiles y usuarios creados. A modo de ejemplo se muestra en la Figura 31 el resultado de una evaluación. En particular se muestra el resultado obtenido de evaluar un servicio de la UNASEV con el perfil de nombre “Perfil Servicio Básico”, cuyo resultado fue del 50% de aprobación. Se ve como el servicio evaluado no cumple con la métrica de nombre “Información en excepciones” y si cumple con la métrica de nombre “Excepciones en formato OGC”.

Resultado de la evaluación general del perfil:

50 % de aprobación

Detalle de la evaluación realizada:

Ide/Institución/Nodo	Descripción del Servicio	Métrica	Modelo	Perfil	Fecha	Completa	Resultado
Nodo UNASEV	Siniestros Fatales	Información en excepciones	ModeloIDEuy	Perfil Servicio Básico	2017-11-13	Si	No cumple
Nodo UNASEV	Siniestros Fatales	Excepciones en formato OGC	ModeloIDEuy	Perfil Servicio Básico	2017-11-13	Si	Cumple

Figura 31: Vista de la herramienta mostrando los resultados

Luego en la Figura 32 se muestran los resultados de varias evaluaciones y algunos de los indicadores obtenidos. En la parte superior vemos los cinco servicios con mejor disponibilidad, de los servicios que han sido evaluados hasta el momento. Abajo a la izquierda se muestra un gráfico de torta indicando el porcentaje de éxitos contra el porcentaje de fracasos, del total de evaluaciones realizadas hasta el momento. A la derecha se muestra un gráfico de barras presentando el porcentaje de éxitos por perfil de evaluación.

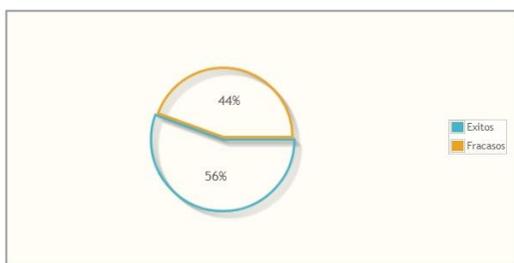
### Servicios destacados

Servicios evaluados con la mejor disponibilidad.

	Descripción	Tipo	Disponibilidad
●	Hidrogeología	Servicio	100%
●	INSTRUMENTOS DE O.T EN ELABORACIÓN	Servicio	100%
●	Cartas de Suelos	Servicio	100%
●	Distritos Mineros	Servicio	100%
●	NORMATIVA DE ORDENAMIENTO TERRITORIAL	Servicio	100%

#### Éxitos vs Fracasos

Muestra en cantidades porcentuales el total de evaluaciones con resultado exitoso contra el porcentaje de evaluaciones con resultado fallido.



#### Éxitos por Perfil

Muestra el porcentaje de éxitos que se han obtenido hasta el momento, evaluando los distintos Perfiles existentes.

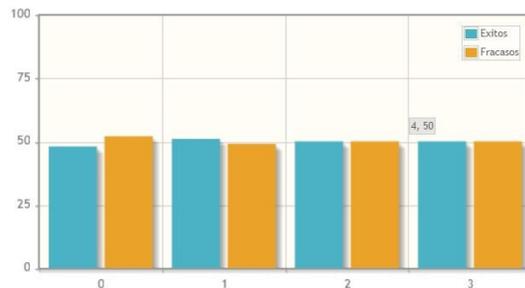


Figura 32: Vista de la herramienta mostrando indicadores

## 6.2. Pruebas realizadas

En primera instancia se realizaron pruebas unitarias de cada módulo desarrollado. Conforme la aplicación fue creciendo, se debió realizar pruebas de integración entre los distintos módulos construidos.

Para la etapa de desarrollo del prototipo se realizaron pruebas de cada una de las componentes implementadas. Luego para la versión final, se realizaron pruebas de regresión para asegurar el correcto funcionamiento de las funcionalidades construidas en el prototipo, así como pruebas funcionales sobre los últimos elementos integrados para la versión final de la herramienta.

Durante la ejecución de pruebas del software, se realizaron pruebas de calidad, detectando mejoras de usabilidad que fueron aplicadas en la herramienta. Por ejemplo, se mejoró la forma en que un objeto medible es seleccionado para ser evaluado, mostrando un árbol de objetos medibles desplegable y seleccionable, junto con dos tablas que muestran datos del objeto seleccionado. Además se realizaron pruebas no funcionales de seguridad, restringiendo el acceso a usuarios sin permisos a la aplicación, pruebas de robustez, reaccionando adecuadamente ante entradas inválidas de los usuarios y sin que el desempeño del sistema se vea afectado. Como por ejemplo un servicio no existente, o una dirección de correo electrónico mal formada.

Finalmente una vez que la solución fue publicada, se dio acceso a distintas personas involucradas en el área para obtener opiniones sobre la herramienta construida y poder mejorar la experiencia de usuario. Particularmente el sistema fue probado por las tutoras del proyecto, por Martín de los Reyes y por Bruno Rienzi. Las devoluciones recibidas sirvieron para valorar el funcionamiento y realizar ajustes al sistema. Además la herramienta fue presentada en la exposición de ciencia y tecnología Ingeniería de Muestra<sup>69</sup>, realizada los días jueves 19, viernes 20 y sábado 21 de octubre del año 2017. Las personas que concurrieron a la exposición también pudieron hacer uso de la herramienta, validando su usabilidad.

---

<sup>69</sup> <https://www.fing.edu.uy/ingenieriademuestra>

## 7. Conclusiones y Trabajo a Futuro

En este capítulo se presentan las conclusiones del proyecto y las posibles mejoras a realizar como trabajo a futuro, para esto comenzamos describiendo la gestión del proyecto en la sección 7.1.

### 7.1. Gestión de proyecto

La planificación del proyecto fue realizada mediante diagrama de Gantt, elegido por ser un método gráfico que representa adecuadamente el avance del proyecto, reflejando las distintas etapas así como los hitos a cumplir. Además como herramienta de gestión de tareas se utilizó Trello, por su flexibilidad y facilidad de uso.

La modalidad de entrega y evaluación del proyecto requería de la implementación de un prototipo de la herramienta dentro de las 20 semanas de comenzado el proyecto. Es por ello que se dividieron las funcionalidades a implementar en dos categorías: funcionalidades básicas y avanzadas. Se incluyeron las primeras en el prototipo, dejando el resto para la última etapa.

Básicas:

- Login de usuarios.
- ABM de usuarios.
- ABM de objetos medibles de tipo servicio.
- ABM de perfiles de granularidad servicio y no ponderados.
- Evaluación de servicios.
- Reportes de servicios evaluados.
- Implementación de las métricas más relevantes.

Avanzadas:

- ABM de cualquier tipo de objeto medible.
- ABM de perfiles de todas las granularidades.
- ABM de perfiles ponderados.
- Evaluación de todos los tipos de objetos medibles.
- Reportes, indicadores y gráficos.
- Implementación del resto de las métricas.

El prototipo fue validado ante las partes interesadas luego de una reunión en la Torre Ejecutiva - Presidencia de la República, de la cual participaron distintos miembros de AGESIC, Presidencia e IDEuy, así como las tutoras de proyecto. Posteriormente se prosiguió con el desarrollo de las funcionalidades avanzadas, ya mencionadas.



Parte de los desvíos en la planificación se debieron a la ampliación de la etapa de desarrollo de la solución y caso de estudio, lo que generó un corrimiento en el comienzo de la etapa de construcción del informe final del proyecto. Además fue necesario añadir la tarea de publicación de la aplicación web, que insumió al equipo más tiempo de lo pensado (tres semanas de trabajo).

## 7.2. Conclusiones generales

La herramienta construida cumple con el objetivo general planteado para el proyecto de brindar indicadores de calidad sobre los distintos servicios web geográficos evaluados, así como desarrollar el caso de estudio de una IDE en particular, la IDEuy.

La primera etapa consistió en el estudio del marco teórico y los conceptos relevantes para el desarrollo del proyecto. En particular se estudió el funcionamiento de los servicios web geográficos, las infraestructuras de datos espaciales, el concepto de calidad de servicio y el modelo de calidad desarrollado en la Tesis del Lic. De Los Reyes [3].

Seguidamente se realizó el análisis de requerimientos del sistema identificando los requerimientos funcionales que se agrupan en las siguientes áreas: autenticación y gestión de usuarios, gestión de objetos medibles, gestión de perfiles de evaluación, gestión de modelos de calidad, evaluaciones y gestión de reportes. Mientras que los requerimientos no funcionales se centran en extensibilidad del modelo, desempeño, seguridad y mantenibilidad.

Luego se realizó un análisis de las herramientas y soluciones existentes en el mercado. Si bien no se encontraron soluciones que cumplan con los objetivos generales del proyecto, se identificaron librerías a utilizar para el procesamiento de las respuestas de los servicios web geográficos.

Una vez terminado el análisis de requerimientos y de soluciones existentes, se realizó una propuesta y diseño de solución que cumple con los requerimientos reunidos. La solución planteada consta de una arquitectura en tres capas con varios componentes. Además como parte del planteo de la solución se realizó un análisis y clasificación de las métricas comprendidas en el modelo de calidad, identificando los pasos para su resolución. Las métricas fueron clasificadas en tres categorías:

### **CAT\_GET\_CAPABILITIES**

Esta categoría implica que la resolución de la métrica es efectuada mediante el procesamiento de la operación getCapabilities para distintos métodos.

### **CAT\_PEDIDOS\_SERVICIO**

Esta categoría implica que la resolución de la métrica es efectuada mediante la realización

de pedidos a los servicios y verificación de su respuesta.

### **CAT\_RESPUESTA\_MANUAL**

Esta categoría implica que la resolución de la métrica es efectuada manualmente, por parte de los usuarios.

Luego se llevó a cabo un prototipo con las funcionalidades identificadas como básicas y necesarias, el cual permitió validar la solución propuesta, su viabilidad y las herramientas utilizadas. En cuanto a la librería OGC Schemas and Tools, utilizada para el procesamiento de esquemas xml, creemos que tiene un gran potencial para la manipulación de documentos XML en entornos Java GIS.

Finalmente se extendió el prototipo implementando el resto de las funcionalidades y se realizaron las pruebas planificadas, para validar el funcionamiento y corregir errores.

Además se logró cumplir con el requerimiento de extensibilidad del modelo de calidad, mediante la carga dinámica de los métodos asociados a las métricas ingresadas por los usuarios. De esta forma se permite asignar implementaciones al método de evaluación de las métricas, siguiendo las pautas de implementación pero sin detenerse en cómo las métricas son integradas con el resto del sistema, facilitando su incorporación.

Cabe destacar que la herramienta fue utilizada exitosamente en el caso de estudio planteado, el cual consistió en evaluar una IDE en particular, la infraestructura de datos espaciales del Uruguay (IDEuy).

Como conclusión general del proyecto, se puede determinar que es factible la implementación de una herramienta de estas características, cumpliendo con los objetivos planteados, pudiendo ser de gran utilidad para lograr una mayor eficiencia en el cumplimiento de estándares y en la mejora de los servicios geográficos ofrecidos por las IDEs. Además la temática tratada a lo largo del proyecto resultó interesante, debido en parte, a las posibilidades de mejora en el área en lo que respecta a nuestro país.

## **7.3. Trabajo a futuro**

A continuación se plantean algunos puntos interesantes identificados como posibles mejoras en la herramienta:

- Reportes de usuario a medida.

Actualmente el usuario tiene la posibilidad de descargar los reportes generados en formato PDF con toda la información obtenida. Una mejora podría ser brindar la posibilidad de que el usuario pueda elaborar sus propios reportes, seleccionando solamente las medidas que considere útil.

- Personalización de página de inicio.  
Si bien la herramienta cuenta con una cartelera de inicio donde se muestra la información más relevante, se podría agregar la funcionalidad de personalización de la misma para cada usuario, de forma de acceder rápidamente a la información más relevante para él.
- Ampliación de métricas disponibles.  
La implementación de métricas que no fueron tenidas en cuenta dentro del alcance del proyecto (Tabla 10: Métricas implementadas y su clasificación según el método de resolución), pertenecientes al modelo de calidad propuesto en la Tesis de referencia por el Lic. De Los Reyes [3], ya que aportan indicadores de calidad útiles al usuario.
- Envío de notificaciones.  
Brindar la posibilidad a los usuarios Administradores Técnicos de registrarse para recibir notificaciones por email, informando de determinados eventos de su interés. Por ejemplo cuando determinados servicios dejan de estar disponibles.
- Soporte a múltiples métodos por métrica.  
El sistema fue implementado de tal forma que exista un único método por métrica, pero se podría quitar esta restricción y permitir la implementación de más de un método por métrica.
- Adicionar controles sobre los objetos medibles a evaluar.  
En caso de evaluar un objeto medible con una métrica no aplicable al mismo, se podría alertar al usuario de esto, o bien no permitir que se realice la evaluación. Esto ocurre, por ejemplo, al intentar evaluar un servicio WFS con la métrica “formato KML”, que indica si el método getMap soporta formato KML. Como WFS no cuenta con dicho método, entonces la evaluación siempre resulta en fracaso. En este caso, la métrica debería ser aplicada a servicios WMS que si cuentan con el método getMap. Actualmente el usuario debe tener el conocimiento suficiente para evaluar los objetos medibles que apliquen a los perfiles seleccionados.



## 8. Referencias

- [1] Open Geospatial Consortium Inc, “OpenGIS® web services architecture description”. OpenGIS Best Practices Paper, 17-11-2005.
- [2] M. I. Papazoglou, “Web Services: Principles and Technology”. 1st ed. Prentice Hall, 09-2007.
- [3] Lic. M. De los Reyes, “Modelo de Calidad para la Evaluación de Servicios Web Geográficos en la Infraestructura de Datos Espaciales de Uruguay”. Tesis de Maestría, Universidad de la República, 2016.
- [4] V. Olaya, “Sistemas de Información Geográfica”. S.I.: CreateSpace Independent Publishing Platform, 2016.
- [5] J. R. Herring, “OpenGIS Implementation Standard for Geographic information, Simple feature access - Part 1: Common architecture, OGC”. Open Geospatial Consortium Inc, 28-05-2011 [Online]  
Disponible: <http://www.opengeospatial.org/standards/sfa> [Accedido: 05-09-2017].
- [6] J. R. Herring, “OpenGIS Implementation Standard for Geographic information, Simple feature access - Part 2: SQL Option, OGC”. Open Geospatial Consortium Inc, 04-08-2010 [Online]  
Disponible: <http://www.opengeospatial.org/standards/sfs> [Accedido: 05-09-2017].
- [7] “Extensible Markup Language, W3C”. [Online]  
Disponible: <https://www.w3.org/standards/xml/> [Accedido: 05-09-2017].
- [8] “OGC Geography Markup Language (GML), Extended schemas and encoding rules”. Open Geospatial Consortium, 07-02-2012. [Online]  
Disponible: <http://www.opengeospatial.org/standards/gml> [Accedido: 05-09-2017].
- [9] M. Lupp, “Styled Layer Descriptor profile of the Web Map Service Implementation Specification”. Open Geospatial Consortium, 29-06-2007 [Online]  
Disponible: <http://www.opengeospatial.org/standards/sld> [Accedido: 05-09-2017].
- [10] D. Burggraf, “OGC KML 2.3”. Open Geospatial Consortium, 04-08-2015 [Online]  
Disponible: <http://www.opengeospatial.org/standards/kml> [Accedido: 05-09-2017].
- [11] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, “Hypertext Transfer Protocol, HTTP/1.1”. The Internet Society, 06-1999. [Online]  
Disponible: <https://tools.ietf.org/html/rfc2616> [Accedido: 05-09-2017].

[12] C. Pautasso, O. Zimmermann, F. Leymann, "Restful web services vs. big web services: making the right architectural decision", Proceedings of the 17th international conference on World Wide Web, Beijing, China, 21-04-2008.

[13] T. Berners-Lee, R. Fielding, L. Masinter. "Uniform Resource Identifier (URI): generic syntax". IETF RFC 3986, 01-01-2005.

[14] S. Supavetch, S. Chunithipaisan, "Interface Independent Geospatial Services Orchestration". Information Technology Journal, 2011.

[15] "OpenGIS Web Map Server Implementation Specification". Open Geospatial Consortium Inc. 15-03-2006. [Online]  
Disponible: <http://www.opengeospatial.org/standards/wms> [Accedido: 03-09-2017].

[16] A. Vretanos, "OGC Web Feature Service 2.0 Interface Standard - With Corrigendum". Open Geospatial Consortium Inc. 10-07-2014. [Online]  
Disponible: <http://docs.opengeospatial.org/is/09-025r2/09-025r2.html> [Accedido: 03-09-2017].

[17] "Web Coverage Service, OGC". Open Geospatial Consortium Inc. 12-07-2012. [Online]  
Disponible: <http://www.opengeospatial.org/standards/wcs> [Accedido: 03-09-2017].

[18] D. Nebert, U. Voges, L. Bigagli, "OGC® Catalogue Services 3.0 - General Model". Open Geospatial Consortium Inc. 10-06-2016 [Online]  
Disponible: <http://docs.opengeospatial.org/is/12-168r6/12-168r6.html> [Accedido: 03-09-2017].

[19] "Web Services Geoespaciales", Taller de Sistemas de Información Geográfica, Facultad de Ingeniería, Universidad de la República, Montevideo, Uruguay. [Online]  
Disponible: <https://eva.fing.edu.uy/mod/resource/view.php?id=47130> [Accedido: 03-09-2017].

[20] R. Chinnici, J. Moreau, A. Ryman, S. Weerawarana, "Web Services Description Language (WSDL)". W3C, 26-06-2007 [Online] Disponible: <https://www.w3.org/TR/wsdl>. [Accedido: 03-09-2017].

[21] "Web Services Geográficos II", Taller de Sistemas de Información Geográfica, Facultad de Ingeniería, Universidad de la República, Montevideo, Uruguay. [Online] Disponible: <https://eva.fing.edu.uy/mod/resource/view.php?id=55690> [Accedido: 03-09-2017].

[22] R. Sosa, "Web Services Geográficos y Gobierno Electrónico", Reporte Técnico, PEDECIBA Informática, Facultad de Ingeniería, Universidad de la República, Montevideo, Uruguay, 04-2011. [Online] Disponible: <https://www.fing.edu.uy/inco/pedeciba/bibliote/reptec/TR1104.pdf> [Accedido: 03-09-2017].

[23] C. M. López-Vázquez, M. A. Bernabé-Poveda, "Fundamentos de las Infraestructuras de Datos Espaciales", 1 edition. UPM Press, 2012.

[24] B. Rienzi, R. Sosa, P. Foti, L. González. "Benefits and challenges of using geographic information systems to enhance social security services". 6th International Policy and Research Conference on Social Security. Luxemburgo, 09-2010. [Online] Disponible: [http://www.academia.edu/26151031/Benefits\\_and\\_challenges\\_of\\_using\\_geographic\\_information\\_systems\\_to\\_enhance\\_social\\_security\\_services](http://www.academia.edu/26151031/Benefits_and_challenges_of_using_geographic_information_systems_to_enhance_social_security_services). [Accedido: 08-09-2017].

[25] "Infraestructura de datos espaciales (IDE)", Scripta Nova, Revista electrónica de geografía y ciencias sociales, Joan Capdevila i Subirana, Instituto Geográfico Nacional, Universidad de Barcelona, 01-08-2004.

[26] "Introducción a las IDE." Infraestructura de Datos Espaciales de España, España, 02-2017 [Online]. Disponible: [http://idee.es/resources/documentos/Introduccion\\_IDEE.pdf](http://idee.es/resources/documentos/Introduccion_IDEE.pdf). [Accedido: 12-11-2017].

[27] "IDE, AGESIC" AGESIC [Online].

Disponible: <https://www.agesic.gub.uy/innovaportal/v/665/1/agesic/IDE.html>. [Accedido: 06-06-2017].

[28] "Asunto 1405. Resolución Presidencial." Ministerio de Economía y Finanzas, República Oriental del Uruguay, 16-06-2006 [Online].

Disponible: [http://ide.uy/sites/default/files/resolucion\\_presidencial\\_16062006.pdf](http://ide.uy/sites/default/files/resolucion_presidencial_16062006.pdf). [Accedido: 06-06-2017].

[29] "Ley N° 19149. Aprobación de rendición de cuentas y balance de ejecución presupuestal", Presidencia de la República, 11-11-2013 [Online].

Disponible: <http://www.impo.com.uy/bases/leyes/19149-2013>. [Accedido: 06-06-2017].

[30] "Decreto N° 390/014. Reglamentación del art. 75 de la ley 18.362 relativo a las pautas para la producción, documentación, acceso y uso de la información geográfica que se produce en el estado", Presidencia de la República, 12-01-2015 [Online].

Disponible: <http://www.impo.com.uy/bases/decretos/390-2014>. [Accedido: 06-06-2017].

[31] Metodología del Proceso Extensión SOA, "Service Oriented Architecture (SOA)", Instituto de Computación, Facultad de Ingeniería, Universidad de la República, Montevideo, Uruguay, 2005

[32] E. Kim, Y. Lee, Y. Kim, H. Park, J. Kim, B. Moon, J. Yun, G. kang, "Web Services Quality Factors Version 1.0", OASIS, 01-07-2010. [Online].

Disponible: <http://docs.oasis-open.org/wsrm/wsrf/v1.0/WS-Quality-Factors-v1.0-cd02.html>.

[Accedido: 06-06-2017]

- [33] M. de los Reyes, L. González, R. Sosa, “Quality Model for the Evaluation of Geospatial Web Services within the Spatial Data Infrastructure of Uruguay”, XLIII Conferencia Latinoamericana en Informática (CLEI), Córdoba, Argentina, 2017.
- [34] O. Cabrera, X. Franch, “A quality model for analysing web service monitoring tools”, 2012 Sixth International Conference on Research Challenges in Information Science (RCIS), 16-05-2012.
- [35] “Apache License 2.0.” Open Source Initiative, 01-2004 [Online].  
Disponible: <https://opensource.org/licenses/Apache-2.0>. [Accedido: 06-06-2017].
- [36] “GNU General Public License.” Open Source Initiative, 06-2007 [Online].  
Disponible: <https://opensource.org/licenses/gpl-license>. [Accedido: 06-06-2017].
- [37] “GNU LGPL.” Open Source Initiative, 06-1991 [Online].  
Disponible: <https://opensource.org/licenses/lgpl-license>. [Accedido: 06-06-2017].
- [38] “The MIT License.” Open Source Initiative, 1988 [Online].  
Disponible: <https://opensource.org/licenses/MIT>. [Accedido: 06-06-2017].
- [39] “European Union Public License, version 1.2 (EUPL-1.2).” Open Source Initiative, 2006. [Online]. Disponible: <https://opensource.org/licenses/EUPL-1.1>. [Accedido: 06-06-2017].